

# Aplicaciones técnicas de usabilidad y accesibilidad en el entorno cliente

UF1843

Xabier Ganzábal García



**CERTIFICADO DE PROFESIONALIDAD  
DESARROLLO DE APLICACIONES CON TEC  
MF0491\_3>UF1843**

Paraninfo

# Aplicaciones técnicas de usabilidad y accesibilidad en el entorno cliente

Xabier Ganzábal García

# Índice

1. Accesibilidad web	1
1.1. Definición de accesibilidad web	3
1.2. Ventajas y dificultades en la implementación de la accesibilidad web	7
1.3. Ventajas y dificultades en la implementación de la accesibilidad web	9
1.3.1. Organismos reguladores de la accesibilidad web	9
1.3.2. Comparativa de normas y estándares	10
1.4. Guías para el cumplimiento de normativas y estándares	12
1.4.1. Descripción de las pautas principales (imágenes, enlaces, video, etc.)	12
1.4.2. Pautas para una navegación accesible	24
1.5. Descripción del proceso de conformidad en accesibilidad web	25
1.6. Tecnologías donde la accesibilidad es aplicable	26
1.6.1. XHTML	26
1.6.2. CSS	27
1.6.3. JavaScript	28
1.6.4. Flash	28
1.6.5. PDF	29
1.6.6. XML/XSL	29
1.6.7. Reproducción multimedia	32
1.6.8. Otras tecnologías	32
1.7. Herramientas para la validación de la accesibilidad	33
1.7.1. Basadas en navegador	33
1.7.2. Mediante aplicaciones de escritorio	38
1.7.3. Mediante servicios web externos	38
1.8. Evolución de la accesibilidad. Nuevas tendencias	40
2. Usabilidad web	45
2.1. Definición de usabilidad	47
2.2. Importancia del diseño web centrado en el usuario	47
2.3. Diferencias entre accesibilidad y usabilidad	47
2.4. Ventajas y problemas en la combinación de accesibilidad y usabilidad	48
2.5. Ventajas y dificultades en la implantación de sitios web usables	48
2.6. Métodos de usabilidad	49
2.7. Análisis de requerimientos de usuario	52
2.8. Principios del diseño conceptual. Creación de prototipos orientados al usuario	55
2.9. Pautas para la creación de sitios web usables	57
2.10. Evaluación de la usabilidad	62
Referencias	71

# 1. Accesibilidad web

## Introducción

La accesibilidad es un área habitualmente olvidada en los manuales sobre diseño y desarrollo web. Aun así, es importante para muchos proyectos y obligatoria para casi todos los relacionados con las administraciones públicas.

A lo largo de este capítulo iremos viendo las pautas y técnicas que hay que utilizar para desarrollar páginas web accesibles. Primero definiremos qué se entiende por accesibilidad y las dificultades que pueden presentar los usuarios en acceder al contenido web. También veremos las tecnologías de asistencia de las que disponen los usuarios. Es un aspecto importante para entender cómo acceden los usuarios con discapacidad a una página y poder adaptarla a sus necesidades. En la sección 1.4 se tratan las pautas para una navegación accesible, con atención especial a las que define el W3C (World Wide Web Consortium), que son la base de muchas de las leyes sobre accesibilidad web en Europa, y concretamente en España.

Aunque la mayoría de las técnicas que veremos están relacionadas con HTML y CSS, la accesibilidad abarca todas las tecnologías y lenguajes utilizados en una página web. En la sección 1.6 veremos cómo afecta a documentos PDF y contenido Flash entre otros. En las secciones finales del capítulo conoceremos una serie de herramientas útiles para evaluar la accesibilidad de las páginas y hablaremos de las nuevas tendencias en accesibilidad.

# Contenido

- 1.1. Definición de accesibilidad web
- 1.2. Ventajas y dificultades en la implementación de la accesibilidad web
- 1.3. Normativa y estándares sobre accesibilidad web
- 1.4. Guías para el cumplimiento de normativas y estándares
- 1.5. Descripción del proceso de conformidad en accesibilidad web
- 1.6. Tecnologías donde la accesibilidad es aplicable
- 1.7. Herramientas para la validación de la accesibilidad
- 1.8. Evolución de la accesibilidad. Nuevas tendencias

## 1.1. Definición de accesibilidad web

La accesibilidad web se ocupa de hacer posible que personas con discapacidad puedan acceder a la web en igualdad de condiciones, sin pérdida de información o funcionalidad. Incluye problemas de visión, audición, movilidad y cognitivos. También a las personas mayores a medida que pierden facultades.

- **Ceguera.** Como la web es un medio visual, muchas de las directrices sobre accesibilidad están dirigidas a usuarios ciegos o con problemas de visión. Los usuarios ciegos requieren tecnología de asistencia que les permita acceder a una versión solo texto de la página, como lectores de pantalla o líneas braille. También necesitan medios específicos para interactuar con la página.
- **Falta de visión parcial.** Muchos usuarios tienen problemas de visión sin llegar a ser ciegos. Pueden tener problemas para ver fuentes pequeñas. Estas personas acceden a la página usando magnificadores de pantalla, hojas de estilo propias para sobrescribir el estilo de la página o ampliando el texto usando la función correspondiente del navegador.
- **Daltonismo.** Es la incapacidad para distinguir algunos colores. Afecta aproximadamente al 7 % de la población, especialmente a los hombres. Puede ser un problema para interpretar gráficos o información indicada por el uso del color. Por ejemplo, si el campo de un formulario se marca en verde si es correcto y en rojo si hay algún error, algunos usuarios no podrán distinguirlo.
- **Fotosensibilidad.** Algunas personas pueden sufrir dolores de cabeza, convulsiones o ataques epilépticos ante cambios rápidos en la intensidad de la luz que perciben. Es muy conocido el caso de la serie de dibujos Pokemon, uno de cuyos episodios causó ataques epilépticos a más de 600 niños.
- **Auditiva.** La mayoría de la información en una página web es de tipo visual, pero aun así la sordera, total o parcial, es un problema para acceder al audio o al video con audio. También es un problema si se usa algún sonido para el funcionamiento de la página, como un sonido de error al introducir un dato incorrecto.
- **Motora.** Los usuarios con problemas de movilidad tienen dificultades para usar el ratón o el teclado. Por ejemplo, hay personas que no tienen suficiente movilidad como para usar un ratón con precisión pero sí pueden manejar un teclado, normal o adaptado. Pensando en estos usuarios, hay que realizar páginas que se puedan utilizar sin ratón.

La movilidad reducida también afecta a la velocidad con la que un usuario puede navegar por página e interactuar con la interfaz de usuario. Esto hay



que tenerlo en cuenta a la hora de establecer límites de tiempo para realizar un proceso.

- **Dislexia.** Los disléxicos pueden tener dificultades para leer un texto. Hay fuentes que les resultan más fáciles de leer que otras.
- **Problemas de atención y comprensión.** Hay personas con problemas para mantener la atención en una tarea incluso durante periodos relativamente cortos. En estos casos, el contenido en movimiento o las ventanas emergentes suponen un problema adicional.

Hay que tener en cuenta que un mismo usuario puede presentar problemas en más de un área.

### Tecnologías de asistencia

La tecnología de asistencia ayuda a los usuarios con las tareas que no pueden realizar o les resultan dificultosas. Un bastón o unas gafas son un ejemplo común de tecnología de asistencia. Sin gafas, muchas personas no podrían conducir o usar un ordenador cómodamente.

**Lectores de pantalla** Para los ordenadores, una de las tecnologías más usadas son los lectores de pantalla, apropiadas para personas ciegas o con visión reducida. Generan una versión de solo texto de la página y la transmiten al usuario mediante síntesis de voz o una línea braille.

Es habitual que los sistemas operativos tengan un lector de pantalla integrado. Además, hay aplicaciones tanto gratuitas como de pago disponibles. Para Chrome existe un lector de pantalla gratuito, el complemento ChromeVox. La siguiente tabla muestra algunos de los lectores de pantalla más extendidos.

Tabla 1. Lectores de pantalla.

Lector de pantalla	Plataformas	Licencia
JAWS Job Access With Speech	Windows	De pago, la licencia normal cuesta más de 800 dólares
NVDA (Non Visual Desktop Access)	Windows	Gratuito
Orca	Sistemas tipo Unix	Gratuito y de código abierto
VoiceOver	Mac	Integrado en todos los dispositivos Apple
ChromeVox	Es un complemento de Chrome	Gratuito
Narrator	Windows	Integrado de Windows

BRLTTY	Consola de sistemas tipo Unix y DOS	Gratuito y de código abierto
--------	-------------------------------------	------------------------------

Entre las características habituales de los lectores de pantallas están:

- Leer los textos alternativos de las imágenes.
- Crear una lista de encabezados para informar al usuario de la estructura del documento.
- Crear una lista de los vínculos presentes en una página para permitir una navegación rápida.
- Crear una lista de las tablas de la página.
- Incluyen una serie de atajos de teclado que permiten obtener información sobre la página y navegar por ella rápidamente.
- La velocidad de lectura es configurable. Los usuarios con experiencia utilizan hasta 300 palabras por minuto.

Tabla 1.2 Algunos atajos de teclado de JAWS.

Tecla (o combinación)	Función
Insert + Flecha arriba	Leer la línea actual
Insert + F6	Lista de encabezados de la página
1-6	Encabezados h1-h6
Insert + F7	Lista de vínculos de la página
Tabulador	Pasar al siguiente vínculo o elemento de formulario
Mayúsculas + Tabulador	Pasar al anterior vínculo o elemento de formulario
Espacio	Seleccionar o deseleccionar casillas de verificación
Control + Alt + Flecha	Moverse por las celdas de una tabla
Control + Alt + Mayúsculas + Flecha izquierda	Ir a la primera celda de la fila
Control + Alt + Mayúsculas + Flecha arriba	Ir a la primera celda de la columna
Insert + Mayúsculas + Flecha arriba	Leer la fila actual de la tabla

Al ser una de las tecnologías de asistencia más extendidas es recomendable que los desarrolladores los usen para comprender bien cómo funcionan. Veamos un ejemplo usando una página sencilla.



```

<!DOCTYPE html>
<html>
  <head>
    <title>Página básica</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Bienvenido</h1>
    Nos hemos mudado, visita nuestra
    <a href="http://www.nuevapagina.com">nueva página</a>
    
  </body>
</html>

```

Ejemplo 1. Ejemplo sencillo para probar un lector de pantalla.

JAWS la leerá como:

La página tiene un encabezado y un enlace. Página básica Encabezado nivel uno. Bienvenido. Nos hemos mudado visita nuestra página gráfico dibujo de un camión de mudanzas.

Las palabras que no aparecen en la página están marcadas en cursiva. El lector de pantalla identifica ciertos tipos de contenido antes de leerlos. A lo largo de este capítulo iremos viendo más ejemplos.

- **Líneas braille** Las líneas braille muestran en alfabeto braille el contenido de una página web. Las usan las personas ciegas y sordociegas. Tiene una serie de elementos móviles que suben y bajan para dibujar los caracteres braille. Son dispositivos bastante caros, en general por encima de los mil euros.



Ilustración 1. Línea Braille.

- **Pulsadores (switch).** Las personas con movilidad reducida pueden usar distintos tipos de pulsadores. Los hay de mano, que suelen ser botones bastante grandes, y de pie, con forma de pedal. Dentro de los pulsadores se

engloban también los dispositivos que detectan si el usuario ha parpadeado o hablado.

- Los joysticks y la trackballs resultan útiles como alternativa al ratón. Los joysticks tienen una palanca que se puede mover con cualquier parte del cuerpo. Suelen incluir uno o más botones. La trackball son esferas que giran sobre sí mismas.
- Dispositivos de inhalar y soplar Estos dispositivos son una varilla o tubo que el usuario maneja con la boca. Moviendo la varilla se realiza el movimiento del ratón. Soplar o inhalar es equivalente a pulsar el ratón. Algunos dispositivos permiten diferenciar entre inhalaciones y soplos fuertes o débiles.
- Dispositivos de seguimiento de la mirada Para usuarios con movilidad muy reducida. Siguen la dirección de la mirada del usuario para simular el movimiento de un ratón. Normalmente utilizan una cámara situada sobre el monitor.
- Dispositivos de generación de discurso Para personas con dificultades cognitivas o que se mueven muy lentamente. Tienen un teclado que puede tener letras, palabras o imágenes. El usuario selecciona la tecla que desea directamente o mediante un proceso de escaneo. Es decir, el dispositivo va pasando de imagen a imagen hasta que el usuario elige una mediante un pulsador. Normalmente, primero recorre fila a fila y luego, cuando el usuario selecciona una, recorre las columnas hasta que llega a la que el usuario desea. Una vez el usuario ha seleccionado varias palabras, el dispositivo las lee en voz alta.
- Amplificadores de pantalla Pueden ser elementos de hardware, como una lupa o cristal amplificador que se sitúa sobre la pantalla, o software, como el programa Lupa de Windows, para amplificar una parte de la pantalla.

## 1.2. Ventajas y dificultades en la implementación de la accesibilidad web

A la hora de comenzar una página web, el desarrollador web o quién le contrata pueden preguntarse qué ventajas tiene desarrollar una página web accesible y si merece la pena preocuparse por la accesibilidad. Hay una serie de motivos que justifican la creación de páginas accesibles:

- La accesibilidad web es en muchos casos un requerimiento legal. En España, así como en otros muchos países las páginas desarrolladas para las

administraciones públicas deben ser accesibles. Cada país define en su legislación qué características debe tener una página para considerarse accesible y en qué casos debe serlo. En la sección 1.4 se trata la normativa aplicable.

- Si una página web no es accesible deja fuera a muchos posibles usuarios o clientes. Según diversas estadísticas el porcentaje de personas con algún tipo de discapacidad está en torno al 18 %, pero esta cifra incluye discapacidades que no influyen a la hora de visitar una web. Si nos fijamos exclusivamente en dificultades visuales y auditivas, afectan aproximadamente al 7 % de los adultos y la cifra crece hasta el 20 % para los mayores de 65 años.

Pensando en el beneficio del propietario de la web, dejar fuera a esa cantidad de usuarios es muy mala idea. Además, al ser personas con discapacidad es lógico que tengan una tendencia mayor a hacer compras y gestiones por Internet en lugar de desplazarse a una tienda física.

- Una página realizada con la accesibilidad como objetivo será más fácil de usar para todos los posibles visitantes.
- Para que una web sea accesible su contenido debe estar organizado correctamente y hay que utilizar las etiquetas semánticamente apropiadas. Una página bien escrita siempre será más fácil de entender y mantener.

El desarrollo de páginas accesibles también presenta algunas particularidades:

- La excusa habitual para no preocuparse por los aspectos relativos a la accesibilidad web es el sobrecoste en el desarrollo en que se incide. Aunque adaptar una página ya creada para que sea accesible puede resultar laborioso, si desde el inicio del desarrollo se tiene la accesibilidad como objetivo no representa una gran sobrecarga de trabajo. La atención a la accesibilidad se debe integrar en el proceso de desarrollo de la empresa u organización.
- El equipo de desarrollo tiene que estar formado adecuadamente. Muchos diseñadores web con experiencia no están al corriente de los temas de accesibilidad. Si una empresa u organización quiere desarrollar páginas web accesibles deberá preocuparse de que los desarrolladores reciban la formación adecuada.
- También es necesario realizar pruebas con usuarios con discapacidad para poder estar seguros de que una web es accesible. Los desarrolladores pueden hacer pruebas usando tecnologías de asistencia, como los lectores de pantalla, pero es difícil ponerse en el lugar de una persona con problemas de accesibilidad si no se tienen sus mismas características.

- Si se trabaja con clientes de diversos países habrá que conocer la normativa aplicable en cada caso y estar al corriente de los posibles cambios.

### 1.3. Normativa y estándares sobre accesibilidad web

Hasta ahora, hemos definido la accesibilidad web en términos genéricos. Pero como hemos dicho, en algunos casos la accesibilidad es un requisito legal y por tanto debe quedar bien definido qué se espera de una página accesible. Hay una serie de normas y estándares que definen qué requisitos debe cumplir una web para poder considerarse accesible.

#### 1.3.1. Organismos regulatorios de la accesibilidad web

El organismo más relevante en el desarrollo de estándares sobre accesibilidad es el W3C. A través de la Iniciativa para una Web Accesible (Accessibility Initiative, WAI) ha desarrollado las Pautas para un Contenido Web Accesible (Content Accessibility Guidelines, WCAG) 2.0. La versión 2.0 es una recomendación oficial desde 2008 y es actualmente el estándar más extendido. En lo que sigue nos referiremos a ellas por sus siglas en inglés, WCAG. Si no se especifica lo contrario, se trata de la versión 2.0.

Son muchos los países que han aprobado legislación relativa a la accesibilidad web, entre ellos España. En general estas leyes obligan a que las páginas de la administración pública sean accesibles. Actualmente, la normativa española fija las pautas WCAG 2.0 como referencia, al igual que ocurre en Italia, Brasil e Israel (en este último caso, con ciertos matices).

La sección 508 de ley de rehabilitación (Rehabilitation Act) de Estados Unidos es otra de las referencias habituales en temas de accesibilidad. Es normal referirse a ella simplemente como sección 508. No se aplica solo a las páginas web, también se ocupa de:

- Aplicaciones y sistemas operativos.
- Aplicaciones web para Internet e intranets.
- Productos de telecomunicaciones, como teléfonos.
- Vídeos y productos multimedia.
- Productos cerrados (en los que los usuarios no pueden usar sus tecnologías de asistencia, como un fax).
- Ordenadores de sobremesa y portátiles.



Es de obligado cumplimiento solo para webs de los organismos federales de Estados Unidos, o webs que reciban fondos federales.

### 1.3.2. Comparativa de normas y estándares

La sección 508 marca unos requisitos mínimos, mientras que las WCAG 2.0 diferencian tres niveles de conformidad. Aunque ambas normas son parecidas en líneas generales las WCAG representan un nivel de accesibilidad mayor que la sección 508.

Las pautas WCAG 2.0 están organizadas en principios, pautas y criterios. Los criterios están clasificados en tres niveles, A, AA y AAA, ordenado de menos a más exigencia. Además, para criterio se incluyen una serie de técnicas para cumplirlo. Las veremos en detalle en la sección 1.4.2.

Además de las WCAG, que se ocupan del contenido web, el W3C ha desarrollado pautas para aplicaciones de Internet enriquecidas (RIA), herramientas de autor y agentes de usuario.

Pautas de accesibilidad para las herramientas de autor: **Authoring Tool Accessibility Guidelines, ATAG**. Las herramientas de autor son las aplicaciones que se usan para generar contenido web, por ejemplo:

- Gestores de contenido como Joomla o Moodle.
- Editores visuales que generan HTML, como Dreamweaver.
- Herramientas para convertir otros formatos, como un documento de texto, a HTML. Entre estas se pueden incluir los procesadores de texto, como Word o Libre Office, que permiten salvar un documento como HTML.
- Herramientas para crear contenido multimedia.
- Páginas que permiten a los usuarios generar contenido como blogs, foros o redes sociales.

Hay dos versiones de ATAG, la 1.0 y la 2.0. La versión oficial actualmente es ATAG 1.0. ATAG 2.0 no es todavía una recomendación oficial, pero se espera que lo sea pronto y sustituya a la versión 1.0. El borrador actual está bastante avanzado y no se prevén grandes cambios.

Las pautas ATAG tienen dos partes:

- La parte A se ocupa de que las propias herramientas sean accesibles para que las personas con discapacidad puedan usarlas.
- La parte B se ocupa de que el contenido generado con estas herramientas sea accesible.

Como las WACG, se organizan en principios, pautas y criterios. Cada criterio está asociado a un nivel de cumplimiento (A, AA y AAA).

**Pautas para la accesibilidad de los agentes de usuario (UAAG, User Agent Accessibility Guidelines).** Por agente de usuario se entiende cualquier programa que obtiene o muestra al usuario contenido web, como los navegadores web y sus plugin adaptadores, como los que reproducen contenido multimedia. Un agente de usuario accesible:

- Permite el acceso a todo el contenido.
- Permite que el usuario configure cómo se muestra el contenido.
- Permite al usuario controlar la interfaz.
- Implementa interfaces de programación estandarizadas para facilitar su uso con tecnologías de asistencia.

Hay dos versiones de UAAG, la 1.0 y la 2.0. La versión 1.0 se aprobó en 2002 y es la versión vigente. La 2.0 no es todavía una versión oficial pero se espera que lo sea pronto sin cambios relevantes.

**WAI-ARIA (Accessible Rich Internet Applications)** es una recomendación del W3C que se centra en la accesibilidad de las páginas con contenido dinámico y controles de usuario avanzados desarrolladas con HTML, JavaScript y AJAX.

WAI-ARIA utiliza una serie de atributos que permiten identificar la funcionalidad de los elementos, su estado y las interacciones entre ellos. Por ejemplo, es posible designar una región como menú, contenido primario o lista. En la sección 1.4.1 veremos dos de estos atributos.

La versión 1.0 es una recomendación oficial. Se ha publicado un borrador de la versión 1.1, con pocos cambios respecto de la anterior.

Tabla 1.3 Guías sobre accesibilidad del W3C.

Siglas	Nombre oficial	Descripción
WCAG	Web Content Accessibility Guidelines	Pautas para la accesibilidad del contenido web
ATAG	Authoring Tool Accessibility Guidelines	Herramientas y programas que se usan para crear páginas web y contenido para las mismas
UAAG	User Agent Accessibility Guidelines	Agentes de usuario: navegadores y plugins



## 1.4. Guías para el cumplimiento de normativas y estándares

Cuando se desarrolla una web accesible es habitual que el cliente exija la adherencia a ciertas pautas, por ejemplo WCAG 2.0. Aun así, no hay que perder de vista el significado general de la accesibilidad y centrarse solo en el cumplimiento de ciertas reglas.

### 1.4.1. Descripción de las pautas principales (imágenes, enlaces, vídeo, etc.)

En esta sección veremos cómo hacer accesibles los elementos más habituales de las páginas webs. Empezamos conociendo algunos principios básicos de accesibilidad y técnicas útiles para varios tipos de contenidos.

#### Alternativas de texto

Las alternativas de texto son un elemento básico para la accesibilidad. Todo el contenido no textual debe tener una alternativa de texto, que será la que usen los lectores de pantalla. El elemento debe quedar asociado con su descripción de manera que los lectores de pantalla (u otros programas) puedan determinar la asociación. En ocasiones esto se puede conseguir usando solo atributos y elementos HTML:

- El atributo `alt` de la elemento `<img>` representa precisamente el texto alternativo de una imagen. Este atributo no se representa en la pantalla (salvo que no se pueda cargar la imagen correspondiente).
- El elemento `<figcaption>` se puede usar para añadir una explicación a cualquier imagen o gráfico. Este texto sí se muestra en pantalla, además de estar disponible para la tecnología de asistencia.
- El elemento `<caption>` añade un texto a una tabla. También se muestra en pantalla.

En otras ocasiones puede que no haya un atributo o elemento apropiado. Por ejemplo en el caso de una transcripción de vídeo o audio. En este caso se puede añadir un vínculo a una página con el texto o relacionar el vídeo con alguna sección de la página con el atributo `described-by` del que hablamos más ade-

lante. También veremos cómo hacer que ciertos contenidos solo estén disponibles para los lectores de pantalla.

### Uso del teclado

Hay usuarios no pueden usar un ratón y acceden a la web usando solo el teclado. Hay que asegurar que toda la información y funcionalidad de la web esté disponible también para ellos. Estos usuarios tienen que poder navegar por la página e interactuar con ella sin limitaciones. Hay que prestar atención a varios factores:

- El usuario no debe poder llegar mediante el teclado a ninguna página o situación de la que no pueda salir mediante el teclado.
- Si en la página se usa algún efecto parecido a arrastrar y soltar (*drag-and-drop*), habrá que implementar también alguna alternativa que se puede usar desde el teclado.

### Ocultar elementos

A veces resulta útil que ciertos elementos estén solo presentes para los usuarios que usan un lector de pantalla. Por ejemplo, para permitir que el usuario se salte una tabla que no le interesa o la barra de navegación. Las barras de navegación suelen repetirse en todas las páginas de un mismo sitio, pero es probable que un usuario con lector de pantalla no quiera oír la lista de vínculos cada vez que accede a una nueva página. Si se utilizan las propiedades CSS

```
display: none;
```

o

```
visibility: hidden;
```

los lectores de pantalla lo ignorarán. Una opción válida es situar el elemento fuera de la pantalla, por ejemplo con la siguiente regla de CSS, que afecta a los elementos de clase oculto.

```
.oculto{  
    position: relative;  
    left: -1000px;  
}
```

Ejemplo 1. Propiedades de estilo para ocultar un elemento.

A continuación se utiliza este método para ocultar un vínculo que permite saltar una barra de navegación.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Vínculo para saltar la barra de navegación</title>
    <link rel = 'stylesheet' href= 'vinculos_salvar_nav.css'></link>
    <meta charset='UTF-8'>
  </head>
  <body>
    <!--vínculo para saltar el menú de navegación-->
    <a class = 'oculto' href = '#contenido'>Saltar navegación</a>
    <!--menú de navegación-->
    <nav id = 'barraNav'>
      <a href='inicio.html'>Inicio</a>
      <a href='futbol.html'>Fútbol</a>
      <a href='baloncesto.html'>Baloncesto</a>
      <a href='fl.html'>Fórmula 1</a>
      <a href='moto.html'>Motociclismo</a>
      <a href='tenis.html'>Tenis</a>
      <a href='contacto.html'>Contacto</a>
    </nav>
    <!-- destino del vínculo-->
    <article id = 'contenido'></article>
  </body>
</html>
```

Ejemplo 1.3. Vínculo para saltar una tabla.

### Atributos labelled-by y described-by (WAI-ARIA)

Estos atributos sirven para asociar una etiqueta o una descripción a un elemento. Son bastante parecidos. La etiqueta describe brevemente el elemento, mientras que la descripción aporta información adicional.

El atributo `labelled-by` se usa para etiquetar un elemento. Por ejemplo `widget`, un grupo de elementos o una región de la página. Contiene `ids` de los elementos que hacen de etiqueta. Si un usuario de lector de pantalla llega a elemento por medio del teclado, leerá las etiquetas asociadas. Es habitual usarlo para campos y botones en formularios.

```

<form>
  <div id = 'etiquetaCampo'>
    Dirección alternativa
  </div>
  <input type = text labelled-by='etiquetaCampo'>
</form>

```

Ejemplo 1.4 Atributo ARIA labelled-by.

El atributo `described-by` se usa para asociar una descripción de texto a un elemento. El valor del atributo debe ser el id del elemento que contiene el texto. Como con `labelled-by` puede haber más de uno.

```

<form>
  <input type = 'button' value = 'Cancelar' described-by='descBoton'>
  <div id = 'descBoton'>
    Este botón cancela el proceso, toda la información introducida se perderá
  </div>
</form>

```

Ejemplo 1.5 Atributo ARIA described-by.

## Imágenes

El método básico para hacer una imagen accesible es usar el atributo `alt` que además es obligatorio. Este texto solo se muestra en pantalla cuando la imagen no está disponible, pero los lectores de pantalla lo leen siempre.

```
<img src = 'tigre.jpg' alt = 'Fotografía de un tigre'>
```

Lo cierto es que hay ocasiones en que realidad una imagen tiene una finalidad puramente decorativa y no tiene ningún contenido informativo. Por ejemplo, si se usa una imagen para un botón. En este caso hay dos opciones:

- Dejar el atributo `alt` vacío.

```
<img src = 'boton.jpg' alt = ''>
```

Cuando los lectores de pantalla encuentran una imagen así, la ignoran.

- Usar la hoja de estilo. En lugar de poner la imagen directamente se puede poner un `<div>` del tamaño deseado y poner una imagen de fondo con la

imagen. Al no aparecer dentro del fichero HTML el lector de pantalla no percibirá la imagen.

```
#boton { background-image: url( 'boton.jpg' ) }
```

En HTML5 hay nuevos elementos para asociar una imagen (u otro tipo de contenido) y un texto descriptivo. Se trata de `<figure>` y `<figcaption>`. El texto contenido en `<figcaption>` se muestra como pie de imagen en la página, pero no siempre tiene que ser una descripción de la misma. Además demostrarse a todos los usuarios, el lector de pantalla también lo leerá, así que se puede usar en combinación `alt`.

```
<figure>  
  <img src = 'vistas.jpg' alt = 'Imagen del valle con el mar de fondo'>  
  <figcaption>Vistas desde la habitación</figcaption>  
</figure>
```

Ejemplo 1. Imagen con `alt` y `figcaption`.

Si se desea incluir una descripción más larga se puede usar el atributo `aria-describedby`. Se puede ocultar esta descripción adicional para que solo esté disponible para los lectores de pantalla, como vimos en las páginas.

## Imágenes de CAPTCHA

Es muy habitual usar imágenes CAPTCHA para verificar que el usuario de la página es una persona real y no un programa. Para evitar que sean fáciles de descifrar, presentan una serie de letras deformadas y a veces con poco contraste con el fondo. Por su propia naturaleza, no tiene sentido ofrecer texto alternativo o una versión más accesible que facilite entender el texto. Sí existe la posibilidad de ofrecer un CAPTCHA de audio. Otra opción es hacer preguntas que sean difíciles de resolver para un programa pero fáciles para cualquier persona. Estas preguntas se mostrarían en texto plano.

## Enlaces

Los enlaces son una pieza fundamental en el HTML y por tanto hacerlos accesibles es prioritario. Hay una serie de pautas básicas que hay que tener siempre en cuenta.

- Hay que usar texto descriptivo en los vínculos.

Los navegadores permiten saltar de vínculo en vínculo pulsando el tabulador. Si se pulsa enter, se sigue el vínculo. Al navegar por una página de esta



manera los lectores de pantalla ignoran el contenido entre vínculos. Por otro lado, es habitual que los lectores de pantalla ofrezcan una lista de todos los vínculos presentes en la página. Así pues es importante que el texto de los vínculos sea descriptivo sin necesidad de leer el contenido que lo rodea.

Por ejemplo, estos vínculos no son descriptivos.

Pulse aquí para leer las instrucciones de instalación.  
Puede aprender a usar la aplicación aquí y aquí.

Ejemplo 1. Vínculos poco accesibles.

La siguiente alternativa es mucho más descriptiva.

Instrucciones de instalación de la aplicación.  
También disponemos de tutoriales sobre uso de filtros y conversión de formatos.

Ejemplo 1. Vínculos accesibles.

Especialmente inapropiado es usar una URL como texto del vínculo. Aunque la URL sea descriptiva el lector de pantalla la leerá letra por letra, lo que seguramente no servirá de nada para el usuario.

- Avisar de los vínculos de descarga.

Si el vínculo sirve para descargar un fichero conviene indicarlo en el texto del propio vínculo. Si se usa un icono para indicar el tipo de contenido que se va a descargar (el icono de Word o de PDF) es importante explicarlo apropiadamente en el atributo alt. Los iconos añadidos automáticamente mediante CSS son ignorados por los lectores de pantalla y por tanto deben ser evitados.

```
<a href = 'fichero.zip' download>Descarga de ficheros históricos</a>
```

- Vínculos que se abren en nueva ventana.

Conviene evitar este tipo de vínculos para no liar al lector de pantalla. En el caso de que considere necesario es recomendable indicarlo en texto del vínculo.

```
<a href = 'personal.html' >¿Quiénes somos? (se abre en ventana nueva)</a>
```

- Vínculos para activar código JavaScript.

Es habitual usar vínculos que no llevan a ninguna parte para llamar a una función JavaScript.



```
<a href="#" onclick="mostrarDatos()">Datos</a>
```

Seguir uno de estos vínculos puede ser desconcertante si el usuario no puede ver el efecto que ha causado en la página. En la sección 1. 6. 3 se explica cómo hacer accesible la funcionalidad generada con JavaScript.

## Vídeos

Los vídeos presentan problemas para las personas con dificultades visuales y auditivas. Para el segundo caso la solución es añadir subtítulos. Al añadir subtítulos a un vídeo, conviene tener en cuenta lo siguiente:

- Los subtítulos deben incluir no sólo las voces, sino también información sobre ruidos relevantes para entender el vídeo.
- Según el color de la imagen puede ser buena idea incluir un fondo de color tras los subtítulos para que se lean bien.
- Hay que tener cuidado para que los subtítulos no tapen partes importantes del vídeo.

Si se trata de un vídeo largo añadir los subtítulos puede ser una tarea tediosa, pero es sencilla. Sólo hay que generar un fichero de texto con los subtítulos y el momento en que deben aparecer e incrustarlas en el vídeo con una herramienta apropiada. Hay varias herramientas gratuitas que permiten hacerlo, algunas de ellas online.

Para los usuarios con problemas de visión hay que ofrecer descripción, o al menos un resumen, del vídeo. Se puede hacer con el atributo `aria-describedby` incluyendo un enlace a un texto con la descripción.

Además de ocuparse de la accesibilidad del vídeo en sí es importante utilizar un reproductor de vídeo accesible. Trataremos el tema en la sección 1.4.5.

## Audio

Si se incluye algún audio en una página web hay que asegurarse de incluir también una transcripción del mismo para los usuarios con problemas auditivos. Lo más sencillo es incluir un vínculo a la misma junto a los controles del audio (si los hay). Se puede ocultar el vínculo para que no aparezca en pantalla pero los lectores de pantalla sí lo lean, como vimos al principio de la sección.

También se pueden utilizar sonidos para indicar alguna condición en la página, por ejemplo:

- Un dato mal introducido en un formulario.

- Indicar que el tiempo para acabar algún procedimiento se acaba. Un caso típico es realizar una reserva de vuelo u hotel, que suelen tener un tiempo determinado para realizarse, o un aviso de cierre de sesión por inactividad en la página de un banco.

En estos casos, hay que añadir también la información de forma visual, ya sea con una alerta o con un icono o mensaje dentro de la página.

## Tablas

Una tabla, si está bien diseñada, es fácil de interpretar a simple vista. Pero para una persona que no ve y dispone de un lector de pantalla puede ser bastante difícil seguir toda la información de la tabla, sobre todo si es grande. Un elemento que no debe faltar es `caption`, para describir la tabla. También es recomendable utilizar los `thead`, `tbody` y `tfoot` para agrupar las filas de encabezado, cuerpo y pie de la tabla respectivamente.

Como las tablas pueden contener mucha información puede ser molesto tener que esperar a que el lector de pantalla la lea entera, sobre todo si el usuario no está interesado en ella. Conviene poner un vínculo que permita saltarlas.

Veamos un ejemplo sencillo de una tabla con dos filas y dos columnas.

```
<table>
  <caption>Tabla de películas y directores</caption>
  <thead>
    <tr>
      <th>Película</th>
      <th>Director</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Los otros</td>
      <td>Alejandro Amenábar</td>
    </tr>
    <tr>
      <td>Los pájaros</td>
      <td>Alfred Hitchcock</td>
    </tr>
  </tbody>
</table>
```

Ejemplo 1. Tabla con `caption`.

La tabla se leerá así:

“Tabla de películas y directores”  
Tabla con dos columnas y tres filas.  
Películas. Director. Los otros. Alejandro Amenábar. Los pájaros. Alfred Hitchcock.”

Tabla de películas y directores

Película	Director
Los otros	Alejandro Amenábar
Los pájaros	Alfred Hitchcock

Ilustración 1.2 Aspecto de la tabla.

Al tratarse de una tabla pequeña es posible oír la y entenderla, pero si tuviera más filas y columnas puede resultar bastante difícil de seguir.

La primera opción para mejorar la accesibilidad de las tablas es usar los atributos `id` y `header`. Se asigna un atributo a cada una de las celdas de cabecera de la tabla. Para las celdas que no son parte de la cabecera se usa el atributo `headers`, que debe contener uno o más de `id` que se han usado en las cabeceras. De esta manera quedan asociadas cabeceras y celdas. La tabla anterior quedaría así.

```
<table>
  <caption>Tabla de películas y directores</caption>
  <tr>
    <th id = "peli">Película</th>
    <th id = "dir">Director</th>
  </tr>
  <tr>
    <td headers = "peli">Los otros</td>
    <td headers = "dir">Alejandro Amenábar</td>
  </tr>
  <tr>
    <td headers = "peli">Los pájaros</td>
    <td headers = "dir">Alfred Hitchcock</td>
  </tr>
</table>
```

Ejemplo 1.10 Atributo `headers`.

A la hora de elegir `id`s:

- Cada columna debe tener un `id` diferente.
- Es recomendable, aunque no necesario, usar un `id` que represente el contenido de la columna.

Si un lector de pantalla lee la tabla entera se leerá como en el caso anterior, pero si se recorre celda a celda el resultado será:

“Tabla de películas y directores. Tabla con dos columnas y tres filas. Películas. Director. Película. Los otros. Director. Alejandro Amenábar. Película. Los pájaros. Director. Alfred Hitchcock.”

Otra opción bastante parecida es usar el atributo `scope`. En HTML5 no se puede usar con los elementos `td`, solo con `th`. Este método es más sencillo y hay que escribir menos, pero no es tan versátil como el anterior.

```
<table>
  <caption>Tabla de películas y directores</caption>
  <tr>
    <th scope = "col">Película</th>
    <th scope = "col">Director</th>
  </tr>
  <tr>
    <td>Los otros</td>
    <td>Alejandro Amenábar</td>
  </tr>
  <tr>
    <td>Los pájaros</td>
    <td>Alfred Hitchcock</td>
  </tr>
</table>
```

Ejemplo 1.1 Atributo `scope`

Esta tabla se leerá igual que la anterior:

“Tabla de películas y directores. Películas. Director. Película. Los otros. Director. Alejandro Amenábar. Película. Los pájaros. Director. Alfred Hitchcock.”

### Organización de la página y etiquetas semánticas

Una página semánticamente correcta es más fácil de interpretar para la tecnología de asistencia. En este sentido es importante:

- Crear una jerarquía de encabezados consistente utilizando los elementos `h1` a `h6`.
- Utilizar las etiquetas semánticas de HTML5 para organizar las secciones de la página `nav`, `header`, `section`, en lugar de `div`, que no dice nada respecto del tipo de información que contiene.

- Escribir la página de manera que se pueda leer correctamente si se prescindiese de las hojas de estilo.

## El texto

El texto es elemento básico de la mayoría de las páginas web. Hay que usar una fuente apropiada y fácil de leer en pantalla y además escoger las combinaciones de colores adecuadas. También hay que asegurarse de que los usuarios que lo necesiten puedan ampliar el texto sin que este salga de la pantalla o se solape con otros elementos. Esto último es posible conseguirlo usando unidades relativas para el texto y las secciones de la página en la hoja de estilo.

Un grupo de usuarios especialmente sensible al formato del texto de una página son los disléxicos. Se han realizado múltiples estudios sobre sus preferencias al leer una página y se han identificado las siguientes:

- Las personas disléxicas encuentran algunas fuentes más fáciles de leer que otras. Prefieren las tipografías sin remates (sans-serif) y monoespaciadas, es decir en las que todas las letras ocupan la misma anchura. Entre las mejor valoradas están Arial, Comic Sans, Century Gothic y Verdana. También se han creado varias fuentes específicas, algunas de ellas de código abierto como OpenDislexic.
- Prefieren el texto no justificado. Al justificar el texto se modifica el espacio entre letras y palabras lo que dificulta la lectura.
- El contraste entre color del fondo y el texto también puede causar problemas. El texto negro contra fondo claro, pero no blanco, es la mejor combinación.
- La fuente debería tener al menos un tamaño de 12 pt.

## Colores y contraste

Como se comentó en la sección 1.1 las personas daltónicas no pueden distinguir bien los colores. El caso más habitual es la dificultad para distinguir entre rojo y verde, pero en menor medida también hay personas que tienen problemas con otros colores. Una pequeña parte de los daltónicos no pueden distinguir ningún color y ven en blanco y negro.

Para evitar problemas a estos usuarios hay que asegurarse de que el color no es la única manera de transmitir determinada información. Un ejemplo habitual es un gráfico de tarta. Cada “trozo” de la tarta tiene un color diferente, y se añade una clave que relaciona los colores con lo que representan. Para alguien con problemas para distinguir colores este tipo de gráfico puede resultar inservible.



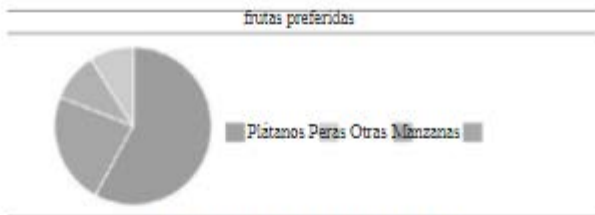


Ilustración 1.3 Gráfico de tarta no accesible

Una solución sería añadir a cada color una textura como puntos o rayas en diferentes direcciones o poner etiquetas adicionales, como se puede observar en el siguiente ejemplo.

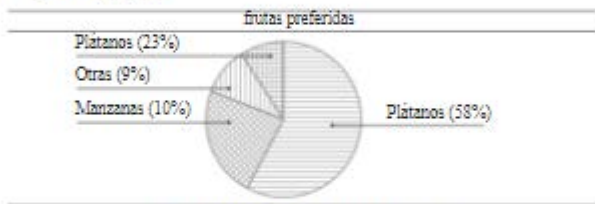


Ilustración 1.4 Gráfico de tarta accesible.

Otro aspecto a tener en cuenta es el contraste entre el color del texto y el del fondo. Esto también se aplica a los colores de las imágenes. Si se usa una combinación con poco contraste algunos usuarios tendrán dificultad para distinguir el contenido.

### Fotosensibilidad

Hay que evitar los vídeos y animaciones con cambios de intensidad intermitentes o parpadeantes, como cambios rápidos y repetidos de un color brillante a uno oscuro. Pueden causar dolores de cabeza, convulsiones o ataques epilépticos a algunos usuarios. En la sección 1.7 veremos una herramienta para valorar si un vídeo presenta estos riesgos.

### Límites de tiempo

Algunos procesos tienen un límite de tiempo para poder llevarlos a cabo. Por ejemplo, al reservar un vuelo hay un tiempo máximo transcurrido el cual el pro-



ceso de reserva se anula y hay que volver al principio. Esto es así porque la plaza queda reservada cuando el usuario la selecciona, dejando de estar disponible para otros. Si el proceso de compra no se realiza en cierto periodo de tiempo la plaza queda libre y vuelve a estar disponible.

Para algunos usuarios esto puede ser un problema. Ya sea por problemas de movilidad o por necesitar más tiempo para leer o entender el contenido pueden no ser capaces de completar la tarea de manera suficientemente rápida.

A la hora de establecer un límite de tiempo hay que tener en cuenta lo siguiente:

- Solo hay que establecerlo si es realmente necesario. Si se establece hay que hacer que sea lo más amplio que el procedimiento permita.
- Conviene avisar al usuario del límite antes de comenzar el proceso.
- Avisar al usuario cuando el tiempo empiece a acabarse.
- Si es posible, hay que dar a los usuarios que lo necesiten la posibilidad de ampliar el plazo.

#### 1.4.2. Pautas para una navegación accesible

Según las pautas WCAG 2.0, para ser accesible el contenido web debe ser:

- Accesible
- Operable.
- Comprensible.
- Robusto.

A partir de estos cuatro principios se definen 12 pautas para la creación de contenido accesible. Para cada pauta establece una serie de criterios de éxito. Además, para cada criterio se incluyen una serie de técnicas para cumplirlo. Cada criterio lleva asociado un nivel A, AA o AAA.

A continuación se enumeran las 12 pautas WCAG 2.0. Se han traducido de Web Content Accessibility Guidelines (WCAG) recomendación del W3C del 11 de diciembre de 2008. La única versión normativa de este documento se encuentra en <http://www.w3.org/TR/WCAG20> esta traducción puede contener errores.

##### Principio 1. Perceptible.

1. Proporcionar alternativas de texto para todo contenido no textual de manera que se pueda transformar a otros formatos que la gente necesite como texto ampliado, braille, voz, símbolos o lenguaje más sencillo.

2. Proporcionar alternativas para los medios tempodependientes.
3. Crear contenido que se pueda presentar de diferentes maneras (por ejemplo, una maquetación más sencilla) sin perder información o estructura.
4. Facilitar a los usuarios ver y oír el contenido, incluyendo separar el primer plano del fondo.

#### Principio 2. Operable.

1. Hacer que toda la funcionalidad esté disponible mediante teclado.
2. Proporcionar a los usuarios suficiente tiempo para leer y usar el contenido.
3. No diseñar contenido de manera que se sabe que puede provocar convulsiones.
4. Proporcionar maneras de ayudar a los usuarios a navegar, encontrar contenido y determinar dónde están.

#### Principio 3. Comprensible.

1. Hacer el contenido textual legible y entendible.
2. Hacer que las páginas web aparezcan y funcionen de manera predecible.
3. Ayudar a los usuarios a evitar y corregir errores.

#### Principio 4. Robusto.

1. Maximizar la compatibilidad con los agentes de usuario actuales y futuros, incluyendo las tecnologías de asistencia.

### 1.5. Descripción del proceso de conformidad en accesibilidad web

El proceso de conformidad consiste en evaluar si se cumplen los requisitos de accesibilidad deseados, por ejemplo, las pautas WCAG 2.0. No se aplica solo a HTML y CSS, sino a todo el contenido de la web que se esté evaluando: videos, animaciones o documentos de texto. Hay herramientas que pueden ayudar en el proceso, como veremos en la sección 1.7, pero algunos criterios deben ser evaluados por una persona con conocimientos en la materia.

Hay tres niveles de conformidad para las pautas WCAG 2.0:

- Nivel A si una página cumple todos los criterios de nivel A o se proporciona una versión alternativa que los cumpla. Este es nivel mínimo de conformidad.

- Nivel AA si se cumplen todos los criterios de nivel A y AA o se proporciona una versión alternativa que los cumpla.
- Nivel AAA si se cumplen todos los criterios de nivel A, AA y AAA o se proporciona una versión alternativa que los cumpla.

La conformidad se refiere páginas completas, no puede aplicarse si solo una parte de la página cumple los criterios. Además, si una página forma parte de una serie de páginas que se utilizan para completar un proceso, todas las páginas involucradas en este deben cumplir el nivel de conformidad declarado.

El W3C proporciona tres logos para indicar el nivel de conformidad. Es habitual verlos en las páginas accesibles. El código HTML necesario para usarlos se puede descargar de la página del W3C e incluye un vínculo a una descripción detallada de lo que significan. Para el nivel A hay que incluir este código:

```
<a href="http://www.w3.org/WAI/WCAG2A-Conformance"
title="Explanation of WCAG 2.0 Level A Conformance">

</a>
```

Ejemplo 1.1 Código para incluir el logo de conformidad nivel A con las WCAG 2.0.

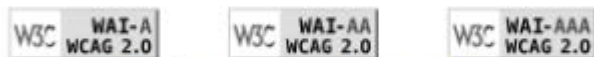


Ilustración 1.5 Logos de conformidad con las WCAG 2.0.

## 1.6. Tecnologías donde la accesibilidad es aplicable

En las páginas web se usan varias tecnologías y lenguajes. Las tres básicas son HTML, CSS y JavaScript, pero no es raro introducir animaciones con Flash, documentos en XML y PDF y una variedad de formatos multimedia. Hay que preocuparse de la accesibilidad de todo el contenido web.

### 1.6.1. (X)HTML

Como lenguaje básico de la web el HTML (y el XHTML), una gran parte de las cuestiones de accesibilidad están relacionadas con él. Como vimos en el apartado 1.4, el código HTML debe ser semánticamente correcto, estar bien jerarquizado y aparecer en el mismo orden en que debe ser mostrado.

### 1.6.2. CSS

Las hojas de estilo CSS permiten separar la información de la forma en que se presenta. Se utilizan para maquetar la página y para dar formato a los elementos de la misma. Algunas propiedades de estilo relacionadas con la accesibilidad son:

- El color del texto y el fondo de las diversas secciones. El cumplimiento de los criterios de accesibilidad relativos al contraste dependen, salvo para las imágenes, de las hojas de estilo.
- El tipo de fuente utilizado. Ya hemos visto que esto es importante para los usuarios disléxicos.
- El tamaño de la fuente, de los elementos HTML y de las secciones en las que se divide la página. Conviene utilizar unidades de medida relativas, como los porcentajes `em`. Ayudará a los usuarios que necesiten agrandar la fuente para poder leer.
- Un uso habitual de CSS es para ocultar contenido de manera que no se vea en pantalla pero sí esté disponible para los lectores de pantalla. Por ejemplo, si se utiliza un vínculo para saltar el menú de navegación o una tabla pensando en los usuarios con lector de pantalla.

Hay una serie de malas prácticas con las hojas de estilo que pueden causar problemas de accesibilidad y por tanto deben ser evitadas:

- Con las hojas de estilo es posible situar los elementos donde se desee con independencia del orden en el que aparezcan en el HTML. Los lectores de pantalla leerán el contenido en el orden en que lo encuentren, así que la página se debe poder leer bien sin la hoja de estilo.
- Utilizar la propiedad de imagen de fondo `background-image` para incluir imágenes. Los lectores de pantalla ignorarán una imagen incluida de esta manera. Solo es apropiado si la imagen es puramente decorativa. Si es parte del contenido debería utilizarse la etiqueta `img` para que los lectores de pantalla la tengan en cuenta.
- La propiedad `content` permite crear contenido mediante la hoja de estilo. Igual que en el caso anterior, solo debe hacerse si el contenido generado es decorativo. Si es importante para el significado de la página debe aparecer en el HTML.
- No se debe usar la hoja de estilo para sustituir a las etiquetas HTML apropiadas. Por ejemplo, en lugar de usar las etiquetas `h1`, ..., `h6` para marcar los encabezados de la página sería posible, pero incorrecto, utilizar párra-



fos y modificar su tamaño mediante la hoja de estilo. Por supuesto, si el aspecto por defecto de los encabezados no es el deseado, se puede modificar su estilo.

### 1.6.3. JavaScript

La tercera tecnología básica en el desarrollo web es JavaScript. Actualmente se usa intensivamente en gran cantidad de páginas. JavaScript puede ayudar a mejorar la accesibilidad de una página (por ejemplo, para avisar de que el tiempo de sesión está a punto a acabar), pero también puede causar problemas.

Una posible fuente de problemas son los eventos. En JavaScript es habitual asociar funciones a eventos, algunos de los cuales están relacionados con acciones de usuario, como pulsar el botón de un formulario o mover el ratón sobre cierto elemento de la página. Hay eventos que están asociados a un dispositivo, como `onmouseover` al ratón y `onkeydown` al teclado, lo que será un problema para los usuarios que no usen ratón o teclado, respectivamente. Conviene usar en su lugar eventos que no necesiten ningún dispositivo concreto, como `onfocus` y `onblur`. El evento `onclick` aunque en principio está relacionado con el ratón, se produce también cuando se pulsa el botón `enter` si el elemento asociado tiene el foco.

También es habitual usar JavaScript para conseguir efectos avanzados, entre ellos:

- Arrastrar y soltar. Lógicamente un usuario sin ratón, o que no pueda usarlo con mucha precisión, no podrá arrastrar elementos por la página. Habrá que preocuparse de ofrecer otra forma de llevar a cabo las tareas.
- Actualizar automáticamente algunas secciones de la página usando AJAX. Mediante JavaScript es posible realizar operaciones periódicamente y actualizar una sección de la página con el resultado de las mismas. El contenido cambiante puede causar problemas a los lectores de pantalla y otras tecnologías de asistencia y también a usuarios con problemas para concentrarse o entender la página.
- Mover, ocultar y mostrar elementos y en general alterar sus propiedades de estilo.

El W3C ha publicado una serie de pautas específicas para las interfaces avanzadas con JavaScript y AJAX bajo la denominación WAI-ARIA.

### 1.6.4. Flash

Flash siempre ha tenido fama de ser poco accesibles, pero en las últimas versiones ha ido introduciendo mejoras en este aspecto, entre ellas:

- Una serie de controles de interfaz gráfica (botones, casillas de verificación...) accesibles.
- Posibilidad de especificar un orden de lectura para los lectores de pantalla.
- Manejo mediante teclado.
- Reproductor de Flash accesible.
- Subtítulos.
- Control sobre el audio.

### 1.6.5. PDF

En muchas páginas web se muestra información en formato PDF. Por ejemplo, la carta de un restaurante, un catálogo o un manual de instrucciones. Hay que ocuparse de que el contenido del PDF sea también accesible.

El formato PDF admite el uso de etiquetas para crear documentos accesibles. Las etiquetas describen la estructura del documento sin alterar su apariencia y están disponibles para los lectores de pantalla. En muchos casos tienen nombres parecidas a las de HTML. Algunas de las disponibles son:

- H1 a H6, para diferentes tipos de encabezado.
- P para párrafos.
- Document, Part, Div, Art y Sect para estructurar el contenido.
- Table y L para tablas y listas respectivamente.

Para crear estos PDF accesibles se puede usar el programa Acrobat de Adobe y también algunos procesadores de texto habituales como Libre Office o Word. El documento original tiene que tener la información necesaria, como textos alternativos para las imágenes y estilos apropiados para los encabezados.



Ilustración 1.6 Cúadro de diálogo con opciones para salvar como PDF en Microsoft Word.

### 1.6.6. XML/XSL

Es habitual incluir contenido XML en las páginas web. Se suele transformar en HTML usando una transformación XSL. ~~El lenguaje~~ Extensible Stylesheet Language



es un lenguaje basado en XML. Está formado por tres lenguajes: XSLT, XPath y XSL-FO. XSLT (Extensible Stylesheet Language Transformation) permite transformar un fichero XML en otro formato, como HTML.

Veamos un ejemplo. Una página sobre coches podría mostrar información almacenada en un fichero XML como este:

```
<?xml version="1.0" encoding="UTF-8"?>
<coches>
  <coche categoria="berlina">
    <marca>BMW</marca>
    <modelo>500</modelo>
    <cilindrada>2000</cilindrada>
    <plazas>5</plazas>
  </coche>
  <coche categoria="utilitario">
    <marca>Nissan</marca>
    <modelo>Micra</modelo>
    <cilindrada>900</cilindrada>
    <plazas>4</plazas>
  </coche>
  <coche categoria="SUV">
    <marca>BMW</marca>
    <modelo>X7</modelo>
    <cilindrada>4000</cilindrada>
    <plazas>5</plazas>
  </coche>
</coches>
```

Ejemplo 1.1 Fichero XML con información sobre coches

Se puede integrar en el HTML de la siguiente con la siguiente transformación, que transforma el fichero XML en una tabla:

```
<table>
  <tr>
    <th>Marca</th>
    <th>Modelo</th>
    <th>Tipo</th>
  </tr>
  <xsl:for-each select="coches/coche">
```

```

<tr>
<td> <xsl value-of select="marca"> </td>
<td> <xsl value-of select="modelo"> </td>
<td> <xsl value-of select="cilindrada"> </td>
<tr>
<xsl for-each>
</table>

```

Ejemplo 1.14 Transformación XSL para el fichero de coches.

El resultado de la transformación será la siguiente tabla de HTML:

```

<table>
<tr>
<th>Marca</th>
<th>Modelo</th>
<th>Tipo</th>
<tr>
<tr>
<td>BMW</td>
<td>500</td>
<td>2000</td>
<tr>
<tr>
<td>Nissan</td>
<td>Micra</td>
<td>900</td>
<tr>
<tr>
<td>BMW</td>
<td>X7</td>
<td>4000</td>
<tr>
</table>

```

Ejemplo 1.15 Resultado de aplicar la transformación del ejemplo 1.14 al fichero del ejemplo 1.13.

Como se puede observar que se genera una tabla HTML que contiene una fila por cada elemento coche del fichero XML. Hay que asegurarse de que el contenido generado a partir de fichero XML cumpla también las pautas de accesibilidad.

### 1.6.7. Reproducción multimedia

Los reproductores que incluyen los navegadores más extendidos están lejos de poder ser considerados accesibles. Hay varios proyectos gratuitos de reproductor accesible que podemos incluir en cualquier página:

- [Video.js](#), de código abierto.
- Accessible HTML5 Video Player, desarrollado por PayPal.
- JW Player, con los controles accesibles desarrollados por la Universidad de Ohio.

Algunas de las características que debe tener un reproductor accesible son:

- Los controles del reproductor se pueden manejar desde el teclado.
- Posibilidad de añadir subtítulos y descripciones de audio.
- Permitir a los lectores de pantalla acceder al volumen y al tiempo de reproducción.
- Desplazamientos por el fichero en función del tamaño del mismo. Por ejemplo, avanzar el 5 % de la duración de un vídeo pulsando la tecla apropiada.

### 1.6.8. Otras tecnologías

- Silverlight, la alternativa a Flash de Microsoft, también tienen características relacionadas con la accesibilidad, entre otras: subtítulos, posibilidad de especificar el orden de los elementos al usar el teclado y plantillas de alto contraste.
- Java. Otro de los medios habituales para incrustar aplicaciones en una página es usar un applet de Java. Las applets se desarrollan usando componentes SWING, que cuentan con mecanismos para mejorar su accesibilidad. El componente Java Access Bridge permite a los lectores de pantalla acceder a la información de accesibilidad de los componentes dentro de la máquina virtual de Java (JVM).
- SVG ( ScalableVector Graphics). SVG es un lenguaje de marcas basado en XML para gráficos vectoriales. A partir de HTML5 se puede integrar sin problemas en cualquier página web. La característica principal de los gráficos vectoriales es que describen la imagen a partir de las formas que la componen, no como un conjunto de píxeles (por ejemplo, para una elipse se indican su centro, sus radios, y el color de relleno). Esto hace posible que la imagen se pueda ampliar sin pérdida de definición, una característica muy

deseable desde el punto de la accesibilidad ya que beneficia a los usuarios que necesiten ampliar la pantalla.

Dentro de una imagen SVG se pueden utilizar los elementos `title` y `desc` para darle un título y descripción apropiados.

```
<svg version="1.1" width="600" height="250">
  <title>Elipse</title>
  <desc>Elipse azul con bordes amarillos</desc>
  <ellipse cx="200" cy="80" rx="100" ry="50"
    style="fill:blue;stroke:yellow;stroke-width:3" />
</svg>
```

Ejemplo 1.16 Fragmento de SVG accesible

## 1.7. Herramientas para la validación de la accesibilidad

Hay bastantes programas y servicios web que ayudan a comprobar si una página se puede considerar o no accesible. Hay herramientas para desactivar la hoja de estilo, generar una versión solo texto de la página, comprobar el contraste de colores y también para verificar que se cumplan las pautas WCAG o la sección 508.

La validación de las pautas WCAG no puede basar solo en estas herramientas, porque algunas no se pueden comprobar de manera automática y es necesario que un humano valore si cierto criterio se cumple o no. Por ejemplo, valorar qué imágenes no necesitan un texto alternativo por ser puramente decorativas.

Además, cada herramienta de validación tiene sus puntos fuertes y sus debilidades así que se recomienda usar varias y comparar sus resultados.

### 1.7.1. Basadas en navegador

**Fangs** (<http://www.standards-schmandards.com/projects/fangs/>) Esta extensión gratuita para Firefox emula el funcionamiento de un lector de pantalla. Genera una versión de solo texto de manera similar a como lo haría uno de estos programas. También genera una lista de vínculos y encabezados. No evalúa la accesibilidad o el cumplimiento de estándares, pero puede ser muy útil para los desarrolladores y resulta bastante más práctico que usar un lector de pantalla normal durante el desarrollo.

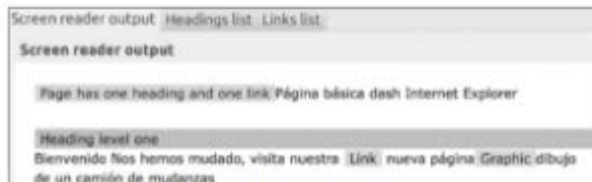


Ilustración 1. Salida de Fanga para el ejemplo 1.1.

Barra de herramientas WAVE (<http://wave.webaim.org/toolbar/>) Disponible para Firefox y Chrome, creada por la organización WebAim. Agrupa varias herramientas útiles en una barra del navegador, por lo que es bastante cómoda de usar mientras se desarrolla. Hay herramientas para señalar problemas de accesibilidad, generar un resumen o una versión de solo texto de la página y deshabilitar las hojas de estilo.



Ilustración 1.8 Barra de herramientas WAVE.

Para probar las herramientas de la barra usaremos un ejemplo. La siguiente página podría ser un principio de resumen de los problemas habituales de accesibilidad.



Ilustración 1.9 Página de ejemplo para probar las herramientas de la barra WAVE.

La página presenta algunos problemas de accesibilidad:

- No tiene título.
- La imagen de la sección superior izquierda no tiene texto alternativo.



- La tabla no tiene elementos `th`, `caption`, `tbody`, `tfoot` o `thead`.
- En la sección superior derecha se ha usado una imagen de fondo con el texto 'PROBLEMA GRAVE!!!!' y fondo rojo. Ni el texto ni el color estarán disponible para los que no puedan ver la imagen. Además, la imagen se ha introducido mediante CSS, así que no aparece en el HTML ni hay posibilidad de usar un texto alternativo.
- El encabezado de la sección inferior izquierda es más pequeño que los del resto porque está mal elegido. Está marcado como `h4` en lugar de `h2`, que es lo que le correspondería.

Para crear esta página se ha usado el siguiente código HTML:

```
<html>
<head>
  <meta charset='UTF-8'>
  <link rel='stylesheet' href='ejFangs.css'>
</head>
<body>
  <h1>Problemas comunes de accesibilidad</h1>
  <section id='multimedia' class='izquierda'>
    <h2>Problemas con elementos multimedia</h2>
    <h3>Problemas con imágenes</h3>
    <p class='izquierda'>Imágenes sin texto alternativo </p>
    <img src='nube.jpg' class='derecha'>
    <h3>Problemas con elementos de audio</h3>
    <p>En algunas ocasiones pueden resultar molestos</p>
  </section>
  <section class='derecha'>
    <h2>Problemas con las hojas de estilo</h2>
    <p id='estilo'>Imágenes de fondo y orden de los elementos de la página</p>
  </section>
  <section class='izquierda'>
    <h2>Problemas con tablas</h2>
    <table>
      <tr>
        <td>Ciudad</td>
        <td>Distancia</td>
      </tr>
      <tr>
```

```

        <td>Madrid</td>
        <td>350 km</td>
    </tr>
</table>
</section>
<section class= 'derecha'>
    <h4>Problemas con el texto</h4>
    <p>Tipo y tamaño de fuentes y contraste</p>
</section>
</body>
</html>

```

Ejemplo 1.1 HTML para la página de la ilustración 1.9.

La hoja de estilo referenciada en la cabecera es la siguiente:

```

section {
    background-color: lightgray;
    margin: 1 %;
    padding: 1 %;
    width: 46 %;
    border-radius: 1em;
}
izquierda { float: left;
            clear: left; }
derecha { float: right; }
#estilo{
    background-image: url('problemasgrave.jpg');
    height: 4em;
}
table, td{
    border: 1px solid black;
    border-collapse: collapse;
}
h3{ clear: both }

```

Ejemplo 1.1 CSS para la página de la ilustración 1.9.

Veamos qué se obtiene al aplicar las herramientas de la barra WAVE a esta página:

- Errores, características, alertas. Con este botón se muestra información sobre accesibilidad mediante una serie de iconos superpuestos sobre la propia página. En la siguiente ilustración se pueden ver iconos rojos, que in-

dicen error, amarillos, que son advertencias, y azules, que son simplemente informativos.



Ilustración 1.10Página de la ilustración 1.9, con información sobre accesibilidad superpuesta.

- Resumen de la página. Esta opción genera un resumen de la página utilizando los encabezados a partir de las etiquetas `h1`, ..., `h6` presentes en la página. El símbolo amarillo que aparece justo antes del último encabezado advierte del error en la jerarquía.

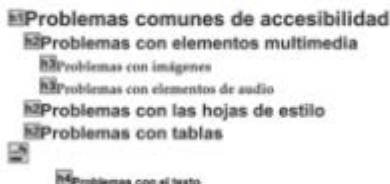


Ilustración 1.11Resumen de la página obtenido con la barra WAVE.

- Versión solo texto o sin estilo.

Problemas comunes de accesibilidad  
Problemas con elementos multimedia  
Problemas con imágenes  
Imágenes sin texto alternativo  
Problemas con elementos de audio  
En algunas ocasiones pueden resultar molestos

Problemas con las hojas de estilo  
Imágenes de fondo y orden de los elementos de la página  
Problemas con tablas  
Ciudad  
Distancia  
Madrid  
350 km  
Problemas con el texto  
Tipo y tamaño de fuentes y contraste

Ilustración 1.12 Versión de la página con el estilo desactivado.

### 1.7.2. Mediante aplicaciones de escritorio

Color Contrast Analyzer(<http://www.paciellogroup.com/resources/contrast-analyzer/>). Pequeña aplicación para comprobar el contraste de colores. Permite saber si cierta combinación de colores pasa los criterios WCAG 2.0. Disponible para Windows y MAC.

TotalValidator(<https://www.totalvalidator.com/>). Además de validar contra WCAG y la sección 508 incluye validación de (X)HTML y CSS. Hay una versión gratuita y una de pago. Si se instala cualquiera de las dos se puede instalar también una extensión para Firefox y Chrome.

PDF Accessibility Checker(<http://www.access-for-all.ch/en/pdf-lab/pdf-accessibility-checker-pac.html>). Es una aplicación gratuita para evaluar la accesibilidad de documentos PDF.

PEAT, Photosensitive Epilepsy Analysis(<http://trace.wisc.edu/peat/>). Esta herramienta gratuita permite comprobar si vídeos y animaciones suponen un riesgo para personas fotosensibles. Solo está disponible para Windows.

### 1.7.3. Mediante servicios web externos

CynthiaSays(<http://www.cynthiasays.com/>). Permite probar una página para ver si cumple con la sección 508 o las pautas WCAG 2.0. Con la versión gratuita solo se pueden probar páginas disponibles en Internet a través de su URL. Para usarlo en una intranet o con páginas en desarrollo hay una versión de pago.

AChecker([achecker.ca](http://achecker.ca)). Permite probar usando la URL, subiendo el archivo o copiando el texto en un formulario. Comprueba WCAG 1 y 2, niveles A, AA y AAA y la sección 508. También incluye validación de HTML y CSS.

Si lo usamos con el ejemplo 1.17 para comprobar las WCAG 2.0 se obtienen los siguientes errores:

- La página no tiene título.
- La imagen no tiene texto alternativo.
- El idioma de la página no está especificado.

Contrast-Finder (<http://contrast-finder.tanaguru.com/>). Esta página permite comprobar si una combinación de colores cumple un cierto ratio de contraste. En el caso de que no lo cumpla ofrece soluciones con colores parecidos a los originales.

Valoración de la legibilidad. Hay herramientas que permiten valorar si un texto es fácil o difícil de entender. Se basan en métricas desarrolladas desde el campo de la lingüística. Para determinarlo tienen en cuenta factores como longitud de las frases y la cantidad de palabras largas que haya en el texto. Un ejemplo es el test de facilidad de lectura de Flesch (FRES, Flesch Reading Ease Score) cuya fórmula es:

$$206.835 - 1.01 (L) - 84.6 (S)$$

Un resultado por encima de 90 significa el texto es apropiado para niños de once años. Un resultado por debajo de 30 significa que el texto es difícil de leer. Hay que señalar que los coeficientes de la fórmula están pensados para textos en inglés, como también ocurre en la mayoría de las herramientas disponibles. Legibilidad M (<http://www.legibilidadmu.com/>) permite evaluar textos en español.

Herramienta	Utilidad
Barra WAVE	Desactivar los estilos, crear una versión solo texto o un resumen de la página. Advertir de problemas de accesibilidad
Fangs	Crea una versión de solo texto de la página de la misma manera que lo haría un lector de pantalla.
TotalValidator	Comprueba el cumplimiento de las WCAG 1.0 y 2.0 (en los tres niveles). También valida HTML, CSS y comprueba los vínculos.
Color Contrast Analyzer	Comprueba si una combinación de colores cumple los criterios de WCAG 2.0.
PDF Accessibility Checker	Comprueba la accesibilidad en documentos PDF.
Photosensitive Epilepsy Analysis Tool	Comprueba si un vídeo o animación puede causar problemas a usuarios fotosensibles (ataques epilépticos o dolores de cabeza)



Cynthia Says	Comprueba el cumplimiento de la las WCAG 1.0 y 2.0 y la sección 508
AChecker	Comprueba el cumplimiento de la las WCAG 1.0 y 2.0 y la sección 508
Contrast-Finder	Comprueba una combinación de colores tiene suficiente contraste y ofrece alternativas si es necesario.
Legibilidad Mú	Valora la dificultad de entender un texto

Tabla 1.4 Resumen de herramientas de accesibilidad

## 1.8. Evolución de la accesibilidad. Nuevas tendencias

Como todo lo relacionado con el desarrollo web la accesibilidad está sujeta a continuos cambios. La legislación sobre el tema es todavía reciente en muchos países, y la tecnología de asistencia, tanto hardware como software, mejora rápidamente. Además, las innovaciones en la web, como el contenido dinámico, los nuevos dispositivos y la proliferación de contenido generado por el usuario añaden nuevas complicaciones para el desarrollo accesible. Es previsible que en los próximos años se den avances en los siguientes campos:

**Homogeneización de la legislación sobre accesibilidad** Como vimos en la sección 1.4, no todos los países exigen lo mismo para considerar una web accesible. Es de esperar que con el tiempo acerquen posiciones, como ya está ocurriendo con las pautas WCAG 2.0 y varios países europeos.

**Contenido generado por el usuario** Con el auge de las redes sociales ha habido un crecimiento del contenido generado por los propios usuarios en Internet. Los desarrolladores de la red social pueden ocuparse de hacer una web accesible, pero sus usuarios suben fotos y vídeos a la red y en la mayoría de los casos no están al corriente de los problemas de accesibilidad de otros usuarios y cómo evitarlos.

**Tecnologías de asistencia** Esta es una de las áreas que puede deparar más avances, en concreto en la mejora y estandarización de las tecnologías existentes y en el desarrollo de tecnologías experimentales. Por ejemplo, el uso de electrodos para poder manejar un ordenador simplemente pensando, puede cambiar radicalmente la manera en que muchas personas utilizan un ordenador.

Por otro lado, como suele ocurrir con la tecnología, es de esperar que los dispositivos de asistencia bajen de precio y más personas que los necesitan se los puedan permitir.

# Ejercicios propuestos

## 1.1. El proceso de conformidad:

- a) Es automatizable en parte, pero algunas pautas requieren verificación por parte de un experto.
- b) Es automatizable en su totalidad.
- c) Es automatizable en su totalidad sólo para el nivel de conformidad A.

## 1.2. Señala la opción correcta sobre la accesibilidad web:

- a) Es una obligación legal para todas las páginas desarrolladas en la Unión Europea.
- b) Se ocupa de que las páginas sean accesibles desde dispositivos móviles.
- c) Beneficia a los usuarios con discapacidad, pero también al resto de usuarios.

## 1.3. La accesibilidad web:

- a) Solo es aplicable para el HTML
- b) Es aplicable a todo el contenido de un sitio web.
- c) Solo es aplicable para HTML y CSS.

## 1.4. Si una imagen es puramente decorativa y no aporta información:

- a) Se usa el atributo `alt` con valor 'puramente decorativa'
- b) No se usa atributo `alt`
- c) Se usa el atributo `alt`, pero se deja vacío `alt = ''`

## 1.5. Escoge la respuesta que consideres incorrecta sobre la accesibilidad de los vínculos:

- a) Tiene que tener un texto representativo.
- b) Si es un vínculo de descarga, avisa al usuario.
- c) Deben abrirse siempre en ventana nueva.

## 1.6. Escoge la respuesta que consideres incorrecta sobre la accesibilidad de las tablas:

- a) Conviene poner un vínculo que permita saltarlas.
  - b) No hay que usarlas, porque los lectores de pantalla no las pueden leer.
  - c) Hay que usar caption
- 1.7. Escoge la respuesta que consideres incorrecta sobre la accesibilidad de los videos:
- a) Basta con añadir subtítulos integrados.
  - b) Hay que utilizar también un reproductor accesible.
  - c) Hay que añadir una descripción en texto.
- 1.8. Escoge la respuesta que consideres incorrecta sobre las hojas de estilo y la accesibilidad:
- a) Las páginas deben mantener el orden y la estructura sin la hoja de estilo.
  - b) Pueden servir de ayuda para la accesibilidad, pero también causar problemas.
  - c) Los lectores de pantalla las ignoran, así que son irrelevantes para la accesibilidad.
- 1.9. El desarrollo web accesible:
- a) En general resulta mucho más caro que si no se valora la accesibilidad.
  - b) No se aplica a las páginas para teléfonos móviles y tabletas.
  - c) Se aplica también a las páginas para teléfonos móviles y tabletas.
- 1.10. Escoge la respuesta que consideres incorrecta sobre el uso de JavaScript y la accesibilidad:
- a) JavaScript puede ayudar a mejorar la accesibilidad de las páginas.
  - b) Hay que escoger eventos que sean independientes del dispositivo.
  - c) Hay que evitar a toda costa el uso de JavaScript.

# Actividades

1.1. Escoge una página de las que visites frecuentemente y comprueba su accesibilidad mediante las herramientas de la sección 1.8. Comprueba tanto los errores de la barra WAVE como su conformidad con las pautas WCAG 2.0.

1.2. Repite el ejercicio con alguna página propia que tenga al menos:

- Imágenes.
- Varios encabezados.
- Tablas.
- Elementos multimedia.
- Vínculos.

Haz los cambios necesarios para que cumpla con los criterios WCAG 2.0 nivel A.

1.3. Intenta crear un PDF accesible con Libre Office o Word. Comprueba su accesibilidad con la herramienta PDF Accessibility Checker.

1.4. Visita la página de algún organismo público e investiga si tiene los logos de conformidad.

1.5. Investiga en Internet los precios de las tecnologías de asistencia de la sección 1.1.

1.6. Instala algún lector de pantalla y utilízalo para visitar alguna página que conozcas.

## 2. Usabilidad web

### Introducción

En el diseño web la usabilidad es absolutamente necesaria. Si los usuarios de una página no pueden encontrar la información que buscan fácilmente o les resulta complicado completar la tarea por la accedieron a la página, como puede ser reservar un vuelo, buscarán una alternativa que ofrezca las mismas funciones pero resulte más sencilla de usar. Si se trata de una aplicación web para los empleados de una empresa, un diseño fácil de usar hará que sean más eficientes en su trabajo.

En este capítulo veremos qué se entiende por usabilidad y los métodos que permiten desarrollar páginas web usables.

### Contenido

- 2.1. Definición de usabilidad
- 2.2. Importancia del diseño web centrado en el usuario
- 2.3. Diferencias entre accesibilidad y usabilidad
- 2.4. Ventajas y problemas en la combinación de accesibilidad y usabilidad
- 2.5. Ventajas y dificultades en la implantación de sitios web usables
- 2.6. Métodos de usabilidad
- 2.7. Análisis de requerimientos de usuario
- 2.8. Principios del diseño conceptual. Creación de prototipos orientados al usuario
- 2.9. Pautas para la creación de sitios web usables
- 2.10. Evaluación de la usabilidad



## 2.1. Definición de usabilidad

La usabilidad es la facilidad de uso y aprendizaje de un producto. El término no se aplica solo a programas informáticos y páginas web, también a máquinas, herramientas o procesos. En la norma ISO 9241-210:2019 *Human-centered design for interactive systems* se define como 'la medida en que un sistema, producto o servicio se puede usar por los usuarios especificados con efectividad, eficiencia y satisfacción en un contexto de uso especificado'.

Abarca varios aspectos del producto, entre ellos:

- Facilidad de aprendizaje.
- Eficiencia. La velocidad con la que pueden realizarse las tareas.
- Errores. Cuantos se cometen y si es fácil solucionarlos.
- Satisfacción. La valoración subjetiva del usuario.

## 2.2. Importancia del diseño web centrado en el usuario

Para conseguir un producto usable es fundamental pensar en los usuarios desde el inicio del proceso de análisis y diseño. Hay que tener en cuenta a qué tipo de usuarios está dirigida la web, las tareas que tienen que realizar, sus capacidades, conocimientos, y el entorno desde el que accederán a la página.

El diseño centrado en el usuario incorpora actividades centradas en el usuario en todas las etapas de desarrollo. Al inicio del proyecto, se analizan los requerimientos de usuario mediante entrevistas y otros métodos. En las siguientes fases, los usuarios valoran y prueban las alternativas de diseño. A partir de las opiniones y resultados de los usuarios los diseñadores van refinando sus propuestas en un proceso iterativo.

## 2.3. Diferencias entre accesibilidad y usabilidad

La usabilidad se ocupa de la facilidad de uso de un sitio web. Los diseños usables son sencillos y fáciles de usar. Por el contrario, lo que determina que un sitio web sea accesible es que los usuarios con discapacidad pueden usar la página sin restricciones.

Es posible que un diseño sea intuitivo y fácil de usar para cualquiera que pueda verlo y usar un ratón, pero sea imposible de usar para alguien que accede con un lector de pantalla y un teclado. Igualmente, un sitio puede ser accesible pero no usable. Por ejemplo, en una página completamente accesible en que los con-

tenidos estén mal organizados, todos los usuarios tendrán problemas para encontrar la información que buscan.

## 2.4. Ventajas y problemas en la combinación de accesibilidad y usabilidad

Como ya se comentó en el capítulo 1, las páginas web accesibles son más fáciles de usar para todos los usuarios ya que en parte accesibilidad y usabilidad tienen objetivos comunes. Muchas de las pautas que vimos benefician a todos los usuarios, entre otras:

- La elección de colores y contraste facilitan la lectura. Igualmente, la posibilidad de ampliar el texto es muy útil para la mayoría de los usuarios a partir de los cincuenta años.
- Los subtítulos servirán también a los usuarios que no dispongan de altavoces y a los que accedan desde un lugar en el que no puedan utilizar sonido.
- Los vínculos con texto representativo son más fáciles de interpretar para todos.

Por otro lado, a veces no es fácil lograr ambas cosas. Gran parte de la usabilidad está relacionada con los controles de la interfaz de usuario. Algunas interfaces avanzadas pueden presentar dificultades de accesibilidad, como puede ser el efecto de arrastrar y soltar o el contenido dinámico. En estos casos puede ser necesario desarrollar una alternativa más accesible, con el consiguiente trabajo extra.

## 2.5. Ventajas y dificultades en la implantación de sitios web usables

La usabilidad de un sitio web mejora la experiencia de usuario y tiene beneficios económicos tangibles:

- Mayor productividad. En muchos casos, como podría ser una aplicación web de gestión, un sitio web es una herramienta de trabajo. Si está bien diseñada los usuarios serán capaces de realizar sus tareas más rápido.
- Menores costes en formación y soporte. Una de las características de la usabilidad es la facilidad de aprendizaje. Una página usable requerirá una inversión menor en formación y soporte de los empleados.
- Más visitas y ventas. Si una web es agradable y fácil de usar es más probable que los usuarios vuelvan a visitarla. Además, si se trata de una página

en la que se venden productos y servicios, una página usable facilitará que los clientes puedan encontrar lo que buscan o necesitan rápidamente.

- **Costes de mantenimiento reducidos.** El diseño centrado en el usuario reduce los costes de mantenimiento, que representa una gran parte del coste total de una aplicación.

Queda claro que la usabilidad de un sitio web es un objetivo irrenunciable, pero conseguirlo requiere conocimientos y experiencia. Hay que conocer a los usuarios o posibles usuarios de la web e identificar sus objetivos y necesidades. Si se trata de una aplicación de trabajo, los usuarios están claramente identificados, ya que son los trabajadores de la empresa. Para una web disponible para el público en general hay que identificar el público objetivo y sus características, de manera similar a como se hace en los análisis de mercado.

En cualquier caso, una característica del diseño centrado en el usuario es, precisamente, que el usuario participa de una manera u otra en todas las fases del desarrollo. Esto puede ocasionar dificultades si los usuarios y el equipo de desarrollo no están cerca unos de otros.

## 2. 6. Métodos de usabilidad

El diseño centrado en el usuario cuenta con una serie de métodos bien definidos para cada una de las fases. La siguiente tabla muestra algunos métodos habituales y el momento del proyecto en se aplican:

Tabla 2 IMétodos de usabilidad.

Tarea	Método
Planificación y viabilidad	Análisis de requerimientos Análisis de sistemas existentes
Análisis de requerimientos de usuario	Entrevistas Grupos de trabajo Encuestas Personas Casos de uso Análisis de tareas
Arquitectura de la información	Ordenación de tarjetas Inventario de contenidos
Diseño de la interfaz	Diseño en paralelo Wireframes
Evaluación de la usabilidad	Pruebas de usabilidad Pruebas en remoto Evaluación heurística y revisión por expertos Seguimiento de la mirada

## Planificación y viabilidad

**Análisis de requerimientos** Como en cualquier proyecto de desarrollo de software, el análisis de requerimientos es uno de los primeros pasos que hay que dar al desarrollar un sitio web. Se trata de definir las funciones y características que debe tener el sitio. El objetivo de esta fase es llegar a una especificación de requerimientos, un documento que servirá de base para el diseño de la aplicación.

Los requerimientos pueden ser de varios tipos. Una clasificación habitual es:

- De negocio. Los objetivos que el cliente quiere cumplir con la aplicación.
- De usuario. Lo que el usuario espera de la aplicación.
- Funcionales. Las características y funciones que debe tener la aplicación, es decir, qué debe hacer.
- De rendimiento. Restricciones a la velocidad o capacidad de la aplicación. Por ejemplo, el número de usuarios que se conectarán a la vez, o el tiempo máximo de respuesta en las consultas a la base de datos.
- De implementación. Se refieren a cómo debe desarrollarse la aplicación, como puede ser el cumplimiento de estándares.

En la sección 2.7 veremos en detalle los métodos para el análisis de requerimientos de usuario.

**Análisis de sistemas existentes** Evaluar versiones previas o páginas de competidores ya presentes en el mercado puede proporcionar información interesante para el proyecto. Sirve para detectar problemas de usabilidad y para establecer una referencia con la que poder hacer comparaciones.

## Arquitectura de la información

La arquitectura de la información se ocupa de la estructura de la página web. El objetivo es organizar los contenidos y la navegación de manera que se faciliten las tareas del usuario. Los componentes principales de la arquitectura de la información son:

- Categorías en las que se organiza el contenido.
- Nomenclatura de las categorías.
- Sistemas de navegación (barras, menús...)
- Sistemas de búsqueda.

Una arquitectura de la información adecuada facilitará que los usuarios encuentren los contenidos y pueden navegar por la página de manera intuitiva. En



esta sección vamos a ver dos métodos que pueden ayudar a desarrollar una arquitectura de la información efectiva:

- Ordenación de tarjetas.
- Inventario de contenidos.

**Ordenación de tarjetas** Este método sirve para ayudar a las categorías del sitio web tomando en cuenta la forma en que los usuarios agrupan los contenidos.

Se pide a un grupo de usuarios que agrupe una serie de tarjetas. Cada tarjeta tiene escrito el nombre de un tema o concepto relacionado con el sitio. Los usuarios agrupan las tarjetas de la manera que les parezca más lógica y apropiada. Hay dos variantes:

- Ordenación abierta. Los usuarios tienen que asignar el nombre que les parezca más apropiado a cada grupo de tarjetas.
- Ordenación cerrada. Se da a los usuarios una lista con los posibles nombres de grupos.

Los resultados se tienen en cuenta para:

- Organizar las barras de navegación y los menús.
- Decidir qué categorías tendrá el sitio y cómo repartir los contenidos.

**Inventario de contenidos** El inventario de contenidos es una lista de todo el contenido de un sitio web. Incluye texto, imágenes, documentos, audio y vídeo. Se recopila información para cada elemento. Los datos básicos son:

- Identificador del elemento.
- URL.
- Tipo de contenido y formato.
- Descripción y palabras clave.
- Categoría a la que está asignado.
- Fecha de creación y última modificación.

Antes de empezar con el inventario hay que definir cuál es el objetivo que se pretende conseguir haciéndolo, por ejemplo:

- Decidir qué contenidos deben ser eliminados.
- Comprobar si están actualizados o hay que revisarlos.
- Identificar contenidos necesarios para que la página sea más completa.
- Revisar si todos los contenidos están en la categoría apropiada.



## 2.7. Análisis de requerimientos de usuario

Conocer al usuario es el principio fundamental de la usabilidad, así que el análisis de requerimientos de usuario debe ser completo y riguroso. Se efectúa al inicio del proyecto y la información obtenida se usa como base para el diseño, junto con el resto de requerimientos.

Hay que identificar las características que definen a los usuarios del sistema. Entre ellas: su nivel de formación, su edad y sus objetivos al utilizar la página. También hay que analizar el entorno del usuario. Por ejemplo, una aplicación que se va a usar en un entorno ruidoso no debería usar un pitido para indicar una alerta.

Otro aspecto importante es su familiaridad con páginas o aplicaciones similares o con versiones previas de la página. Cuando un usuario se acostumbra a un sitio web, los cambios en la estructura del mismo pueden resultarle muy incómodos. Es habitual que los sitios web cambien de aspecto para tener una imagen más moderna y actual, una opción lógica pensando en atraer nuevos visitantes. En cambio, los usuarios acostumbrados a la interfaz anterior se encontrarán con una web que antes manejaban sin problemas y que ahora tienen que aprender a manejar de nuevo, sobre todo si los cambios afectan a la estructura de las páginas de las que se compone el sitio, a los elementos de navegación y la organización de los menús. En estos casos es habitual ofrecer la posibilidad de utilizar el aspecto antiguo al menos durante un periodo considerable de tiempo.

En esta sección se describen algunos de los métodos principales para esta fase de desarrollo:

- Entrevistas y encuestas.
- Personas.
- Casos de uso.
- Análisis de tareas.

### Entrevistas y encuestas

Las entrevistas pueden servir para analizar los objetivos y necesidades de los usuarios. Pueden realizarse en persona, por teléfono o videoconferencia. Se pregunta al usuario por sus expectativas, objetivos y experiencias. El propósito general es conocer mejor al usuario, pero conviene definir con precisión qué información se quiere obtener en cada caso para poder plantear una serie de preguntas apropiadas y seleccionar a usuarios representativos. El principal in-

conveniente es que entrevistar a un número considerable de usuarios puede llevar mucho tiempo.

También se pueden realizar sesiones con grupos de usuarios, normalmente entre cinco y diez. En comparación con la entrevista individual, permite recoger más opiniones en el mismo tiempo y el trabajo en grupo puede ser enriquecedor. Pero también pueden surgir dinámicas de grupo que desvíen la conversación o dejen fuera a alguno de los participantes, así que hace falta un moderador experimentado.

Con las encuestas online se puede llegar a un número mayor de usuarios con un esfuerzo mucho menor que con las entrevistas. Si a los usuarios se les ofrece solo respuestas cerradas, se podrá realizar un análisis estadístico automático para conocer los resultados. Como contrapartida, es difícil saber si quien responde está tomando el tiempo necesario para entenderla y responder apropiadamente.

## Personas

Las personas son una herramienta habitual a la hora analizar a los usuarios. Una persona es un personaje de ficción que representa un grupo de usuarios con características comunes. Para poder definirlos apropiadamente es necesario haber realizado un análisis de usuarios previo. Hay que centrarse solo en los segmentos principales que representan a la mayoría de los usuarios. Se pueden describir con un nivel detalle variable. Entre otras cosas, para cada persona se puede especificar:

- Razón por la que visita la página.
- Sus características personales: edad, género, educación, familia.
- Datos económicos, profesión y experiencia.
- Sus conocimientos técnicos. Hábitos de navegación.
- En qué contexto (lugar, hora) se encuentra cuando accede a la página.
- Objetivos, motivaciones y necesidades.
- Otras páginas similares que pueda visitar.

A cada persona se le asignan algunos datos personales inventados para darles realismo.

## Casos de uso

Un caso de uso describe la interacción del usuario con el sistema. Representa el proceso que tiene que realizar el usuario paso a paso para llevar a cabo a una

tarea. En UML, muy utilizado para representar casos de uso, al usuario se le denomina actor. Son muy utilizados como herramienta de análisis del sistema. No entran en detalles de implementación ni sobre la interfaz de usuario. Como en el caso de las personas, el nivel de detalle es variable. Es habitual incluir al menos:

- Título.
- Actores implicados.
- Precondiciones.
- Disparadores.
- Lista de pasos del escenario de éxito.
- Caminos alternativos si hay algún problema.
- Casos de uso relacionados.

Un ejemplo sencillo de caso de uso podría ser el siguiente:

Elemento	Descripción
Identificador	CU-1
Nombre	Procesar pedido
Descripción	El usuario paga los productos de su carro de la compra. Introduce la información de pago y de envío.
Actores	Cliente
Precondiciones	Tiene que haber algún producto en el carro de la compra
Disparadores	El usuario pulsa el botón de procesar pedido
Escenario principal	<ul style="list-style-type: none"> <li>• El usuario pulsa el botón del carro de la compra</li> <li>• Revisa los productos que está pidiendo</li> <li>• Introduce la información de envío</li> <li>• Se le informa de los gastos de envío y el precio total</li> <li>• Introduce la información de pago</li> <li>• Se procesa el pago</li> <li>• El usuario recibe la confirmación</li> </ul>
Flujos alternativos	<ul style="list-style-type: none"> <li>• Si el usuario no ha iniciado sesión, se le redirige al formulario de login</li> <li>• Si el usuario no acepta los gastos de envío, se cancela el pedido</li> <li>• Si el pago no puede procesarse, se cancela el pedido.</li> </ul>

Ejemplo 2. Caso de uso sencillo.

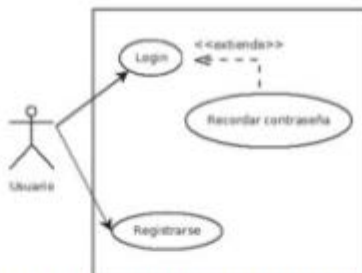


Ilustración 2.1 Fragmento de un diagrama de casos de uso en UML.

**Análisis de tareas** Consiste en observar al usuario mientras usa la página web. El objetivo es entender cómo realizan sus tareas. Puede ayudar a identificar requerimientos del sitio web, a definir los modos de navegación y la organización de los contenidos y a entender el entorno en el que trabajan los usuarios. Este método también puede usarse para evaluar la usabilidad.

## 2.8. Principios del diseño conceptual. Creación de prototipos orientados al usuario

Con los datos recogidos en la fase de análisis los diseñadores ya pueden crear propuestas de diseño. Es un proceso iterativo en el que se van realizando prototipos cada vez más complejos. El cliente da su opinión sobre cada nuevo prototipo y se incorporan sus sugerencias para la siguiente iteración. Este enfoque es útil:

- Para identificar requisitos.
- Para el diseño de la interfaz. En este caso, no hace falta implementar realmente todas las funcionalidades en los prototipos.
- Para detectar errores de planteamiento en fases tempranas.

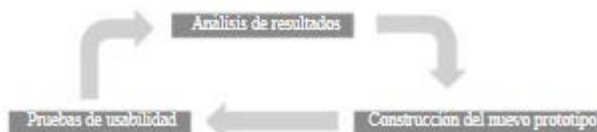


Ilustración 2.2 Diseño iterativo de prototipos.

En las primeras iteraciones se pueden utilizar prototipos poco funcionales para evaluar la arquitectura de la información y el posicionamiento de las secciones en la página. A medida que se avanza en el proceso, se le añaden más funcionalidades y detalles. Cada prototipo se evalúa con los métodos de la sección 2.10.

En esta fase pueden aplicar los siguientes métodos:

**Diseño en paralelo** Consiste en desarrollar varios diseños alternativos a la vez. Cada diseño lo hace un grupo de personas diferente. Los grupos trabajan de manera independiente, ya que el objetivo es explorar la mayor cantidad de alternativas. Posteriormente, se ponen en común las propuestas desarrolladas y se elige el diseño final. Puede ser uno de los propuestos o uno nuevo que incorpore ideas de varios de ellos. Este método permite evaluar múltiples alternativas de diseño de manera rápida y económica. Es habitual en el diseño de interfaces.

**Wireframing** Esta técnica consiste en realizar un boceto de la página web. Se trata de una representación que se centra en los aspectos generales de la interfaz. Los wireframes se usan para:

- Determinar el posicionamiento de los elementos más importantes, la posición de las barras de navegación y los menús dentro de la página.
- Jerarquizar los contenidos mediante su posicionamiento en la pantalla.
- Representar las funcionalidades disponibles.
- Vincular el diseño visual con la arquitectura de la información de la página.
- Punto de partida de los prototipos.

Suelen incluir:

- Encabezados.
- Logo del sitio.
- Barra de búsqueda.
- Medios de navegación (barras, menús).
- Controles de usuario (botones, campos de texto...).
- Información de contacto.
- Pie de página.

Los wireframes pueden hacerse a mano o por ordenador con un programa apropiado. También varían en nivel de detalle:



- **Baja fidelidad.** Representaciones con poco nivel detalle, utilizando rectángulos y etiquetas para representar las secciones principales. Se pueden realizar a mano, son rápidos de crear y sirven como herramienta de colaboración entre los miembros del equipo en las etapas iniciales del diseño.
- **Alta fidelidad.** En este caso el nivel de detalle es mayor, por lo que tardan más en realizarse. Por el mismo motivo, son más útiles como documentación. Incluyen información sobre cada elemento mostrado, como sus dimensiones y comportamiento. Suelen realizarse con aplicaciones específicas, algunas de las cuales permiten incluir elementos interactivos. Hay varias opciones gratuitas y en línea (entre otras: [Flockinbird](#), [Vireframe.cc](#), [Lumzy](#) y [Mockflow](#)).



Ilustración 2.3 Ejemplo de boceto realizado con Wireframe.cc.

Su objetivo no es desarrollar el estilo visual de la página, así que no deben incluir:

- Los colores que se usarán como tema para la página.
- Imágenes.
- Fuentes de texto concretas.

## 2.9. Pautas para la creación de sitios web usables

En este apartado veremos una serie de pautas que facilitan el desarrollo de páginas usables. La mayoría son principios generales, algunos de los cuales se pueden poner en práctica con los métodos que hemos visto en las secciones

anteriores. Otras son consejos muy concretos sobre el estilo y las buenas prácticas de ciertos elementos. No son reglas rígidas y puede haber situaciones en las que no sea conveniente usarlas.

### Navegación por el sitio web y arquitectura de la información

La organización de un sitio es fundamental para que los usuarios puedan llevar a cabo sus tareas fácilmente. La mayoría de los sitios web tienen una barra de navegación con vínculos a las secciones de la página en la parte superior. No es raro que en cada vínculo se despliegue un menú con varias opciones y submenús.

- Debe haber una forma sencilla y clara de volver a la página inicial. El típico vínculo 'Home'.
- Los menús de navegación deben estar organizados de manera lógica y sencilla de entender.
- Los menús tienen que tener un texto adecuado y explicativo.
- Los menús de navegación deben estar organizados de manera que faciliten las tareas del usuario.
- No es conveniente que los menús tengan muchos subniveles. Es preferible que haya más categorías principales.
- Hay que informar al usuario de la sección de la página en la que se encuentra. Una de las formas habituales de hacerlo es conigas de pan (del inglés breadcrumbs). En su versión más sencilla se trata de una sola línea que describe la ruta de la página actual dentro de la estructura de categorías del sitio. Los pasos intermedios de la ruta son vínculos para facilitar que el usuario pueda volver atrás.

Usted está aquí: [Home](#) > [Fútbol](#) > [Primera división española](#)

Ilustración 2.4 Las migas de pan (breadcrumbs) indican al usuario dónde se encuentra.

También es habitual resaltar la sección en la que se encuentra el usuario dentro de la barra de navegación principal.

[Inicio](#) **[Fútbol](#)** [Baloncesto](#) [Fórmula 1](#) [Motociclismo](#) [Tenis](#) [Contacto](#)

Ilustración 2.5 Resaltar la sección actual en la barra de navegación indica al usuario dónde se encuentra.

- Es buena idea crear un mapa del sitio. El vínculo al mapa debe estar presente en todas las páginas.
- En sitios web grandes hay que implementar la función de búsqueda. El campo de texto y el botón de búsqueda deben estar en una posición visible en todas las páginas del sitio. Normalmente es la esquina superior derecha.
- Si se usan pestañas para organizar el contenido hay que poner un texto apropiado y explicativo en cada pestaña. También hay que darles el aspecto apropiado para que el usuario las identifique como pestañas. La siguiente ilustración utiliza *widget* de pestañas de jQuery UI.

Ilustración 2.6 Pestañas con etiquetas representativas.

## Vínculos

- Los vínculos deben tener un texto descriptivo. Además el texto de los vínculos no debe repetirse en vínculos diferentes.
- Es mejor usar texto que imágenes para los vínculos. El texto es más fácil de interpretar.
- Hay que dejar claro qué elementos son vínculos y cuáles no (y más en general, qué elementos son *clickable*).
- Todos los vínculos deben tener el mismo aspecto, diferenciando entre vínculos visitados y no visitados. Los elementos que no sean vínculos no deben parecerlo. Por ejemplo, si se subraya un texto en azul, el estilo por defecto de un vínculo sin visitar, la mayoría de los usuarios lo interpretará como un vínculo.

## Formularios y entrada de datos

Los formularios permiten que los usuarios introduzcan información, normalmente para enviarla al servidor web. En ocasiones son largos y complejos, además de ser parte de procesos importantes como transferencias bancarias. Hay que esforzarse para que los usuarios entiendan fácilmente qué tienen que poner en cada campo.

- Los campos de entrada deben tener valores por defecto útiles si es posible.
- Tanto los campos de entrada como los botones y otros controles deben estar correctamente etiquetados, de manera que no quede duda de su función. Si hace falta, se puede incluir un texto descriptivo adicional.
- Se debe indicar qué campos del formulario son obligatorios y diferenciarlos claramente de los opcionales.
- Si es posible, es mejor usar casillas de verificación, botones de radio y listas en lugar de campos de texto. Minimiza la posibilidad de errores por parte del usuario.
- Si se espera que el usuario introduzca un dato según un formato específico, hay que indicarle cual o dar un ejemplo, como 'dd/mm/aaaa' para fechas o '12345678-A' para el NIF.
- El formulario se debe organizar de manera lógica, agrupando los campos de entrada y con un título adecuado para cada grupo. Por ejemplo, si se trata de un formulario de matrícula en una universidad, el formulario contendrá datos personales del usuario y datos de los estudios en los que quiera matricularse. El formulario será más fácil de entender si los dos grupos de datos están separados y diferenciados.
- Si se introduce un dato erróneo, hay que señalarlo de manera clara y ofrecer al usuario información sobre cómo solucionarlo.
- Hay que situar el foco en un campo apropiado al cargar la página. Si hay que corregir un error, el foco se sitúa en el campo correspondiente.

## Hardware y software

Como en el resto del diseño web, hay que tener presente que nuestras páginas se usarán desde dispositivos muy diferentes en tamaño, capacidades y software.

- Diseñar para los navegadores más comunes. Es habitual que existan diferencias en cómo muestran la misma página diferentes navegadores. Estas diferencias se pueden minimizar si la página se desarrolla siguiendo estrictamente los estándares de HTML, CSS y JavaScript, pero hoy por hoy sigue siendo necesario probar la página en cada navegador en que queramos asegurarnos de que se vea bien.

Aunque hay decenas de navegadores en el mercado, la mayor parte de los usuarios se reparten entre cinco: Mozilla Firefox, Google Chrome, Internet Explorer, Safari y Opera. También hay que tener en cuenta que muchos



usuarios no tienen actualizado su navegador, así que no hay que trabajar solo para las últimas versiones.

Si un usuario se encuentra con problemas al usar una página puede ser que decida que está mal hecha y no se le ocurra utilizar otro navegador.

- Diseñar para los sistemas operativos más habituales. Igual que con los navegadores, los diseñadores deben tener en cuenta también los sistemas operativos más habituales y las posibles diferencias entre ellos, especialmente si algún contenido de la página requiere algún adaptador para poder reproducirlo.
- Diseñar para la velocidad de conexión habitual. Como comentaremos más adelante la velocidad de carga es un factor fundamental en la usabilidad de una página web. Esta depende de los contenidos de la página y de la velocidad de conexión del usuario. Hay que tener en cuenta la velocidad de conexión habitual entre los usuarios de la página. Como ejemplo, la velocidad media en Japón es más del triple que en España y seis veces la de Brasil.
- Diseñar para resoluciones y tamaños de pantalla habituales. Con la aparición de tabletas y smartphones los diseñadores web han visto como su carga de trabajo aumentaba considerablemente. Ahora tienen que ocuparse de que sus diseños se vean bien tanto en un monitor de más de veinte pulgadas como en teléfono de cuatro.

En muchos casos no es suficiente con hacer un diseño que se pueda ampliar o reducir sin que la maquetación se descuadre. La diferencia en tamaño hace que la información de la página se tenga que mostrar de maneras diversas según el dispositivo. Además, hay que tener en cuenta que tabletas y teléfonos se usan habitualmente con la mano, lo que hace necesario que los controles sean relativamente grandes.

### Tiempo de espera

- Minimizar el tiempo de carga de la página. A los usuarios no les gusta esperar en absoluto. Un tiempo de carga de diez segundos por cada página puede hacer que muchos de ellos la abandonen y busquen una página alternativa.
- Avisar al usuario cuando el tiempo de espera para un proceso o una petición puede ser largo. Si es posible, informarle del progreso del proceso y del tiempo restante.
- También hay que avisar a los usuarios si una descarga puede llevar mucho tiempo.



En muchas ocasiones los usuarios prefieren imprimir el contenido de la página, especialmente si se trata de un resguardo o de un artículo largo. Si se usa la función de imprimir del navegador es resultado seguramente no será el deseado. Aparecerán la barra de navegación, los menús y todos los elementos de la página. Hay que preparar una versión especial para imprimir.

## 2.10. Evaluación de la usabilidad

Para cerrar el capítulo nos centramos en los métodos de evaluación de la usabilidad. Dentro del proceso iterativo de diseño la evaluación de los prototipos es necesaria para identificar las áreas en las que hay que hacer cambios o mejoras. Los datos recogidos en la evaluación serán usados por los desarrolladores en la siguiente iteración. Se han propuesto diversos métodos y métricas para evaluar la usabilidad de una aplicación o página web. Pueden fijarse en parámetros subjetivos, como la opinión y grado de satisfacción de usuario, o en medidas objetivas, como el tiempo de respuesta o la cantidad de errores cometidos por los usuarios al realizar una tarea.

**Pruebas de usabilidad** En estas pruebas se convoca a una serie de usuarios representativos y se les pide que usen la página web para sus tareas habituales. Mientras trabajan, los responsables de las pruebas les observan y tratan de obtener datos cualitativos y cuantitativos. Los objetivos de una prueba de usabilidad son:

- Comprobar si los usuarios son capaces de llevar a cabo las tareas correspondientes.
- Medir el tiempo que necesitan para hacerlo.
- Identificar los problemas que tienen de cara a implementar mejoras en el diseño.
- Medir la satisfacción de los usuarios con el sistema.

Un factor esencial en las pruebas de usabilidad es una selección apropiada de usuarios. Las personas que hayan participado en el desarrollo del sistema son una mala elección, ya que están familiarizados con el mismo y por tanto saben usarlo. Hay que escoger a personas ajenas al equipo de desarrollo y que sean representativas de los usuarios del sistema.

Antes de realizar la prueba hay que planificarla detalladamente. El plan debe incluir:

- **Ámbito.** Hay que definir qué se va a probar: la navegación, los contenidos, algunas tareas concretas.
- **Propósito.** Cuáles son los objetivos de la prueba. Por ejemplo: ver si los usuarios encuentran la información que buscan o si cometen errores al rellenar cierto formulario.
- **Participantes.** El número y tipo de usuarios convocados a la prueba.
- **Número de sesiones y duración de las mismas.**
- **Equipamiento necesario:** tipo de dispositivo (ordenador de sobremesa, móvil), sistema operativo, tamaño y resolución de pantalla.
- **Escenarios.** Las tareas concretas que realizarán los usuarios.
- **Métricas subjetivas.** Hay que preparar los cuestionarios previos o posteriores a las pruebas.
- **Métricas objetivas.** Los datos objetivos que se recogerán en la prueba: porcentaje de éxito, número de errores, tiempo medio para llevar a cabo cierta tarea.

Algunas de las métricas habituales son:

- **Porcentaje de usuarios capaces de completar la tarea.**
- **Tiempo invertido.**
- **Errores críticos.** Porcentaje de usuarios que no pueden llevar a cabo la tarea.
- **Errores no críticos.** Porcentaje de usuarios que cometen algún error pero pueden solventarlo y acaban completando la tarea correctamente.
- **Porcentaje de tareas completadas sin errores.**
- **Medidas subjetivas.** A través de cuestionarios, los usuarios opinan sobre la facilidad de uso, la organización de la información o el diseño de la web.
- **Comentarios de los usuarios sobre los que les ha gustado, lo que no y sus sugerencias para mejorar el sitio.**

Las preguntas que se le pueden hacer un usuario son muy amplias. Una primera aproximación es la Escala de Usabilidad del Sistema (System Usability Scale, SUS), desarrollada por John Brooke en 1986. Se trata de diez afirmaciones, muy generales, que los usuarios deben valorar usando una escala de cinco valores, de 'completamente de acuerdo' a 'completamente en desacuerdo':

- Creo que me gustaría usar este sistema habitualmente.
- Encuentro el sistema innecesariamente complejo.

- Encuentro el sistema fácil de usar.
- Creo que necesitaría la ayuda de personal técnico para usar este sistema.
- Creo que las funciones del sistema están bien integradas.
- Creo que el sistema no es consistente.
- Creo que la mayoría de las personas aprenderían a usar el sistema rápidamente.
- Encuentro el sistema incómodo de usar.
- Me siento seguro usando el sistema.
- Tuve que aprender muchas cosas antes de poder manejarlo con el sistema.

La respuesta a cada pregunta se traduce a un número de 0 a 4. La suma de todas las respuestas se multiplica por 2.5 y se obtiene un resultado de 0 a 100. La mejor manera de utilizar este número es para establecer comparaciones entre sistemas.

**Pruebas en remoto** En estas pruebas el usuario permanece en su lugar de trabajo y es monitorizado a distancia.

Como ventajas, evita desplazamientos y permite realizar muchas pruebas simultáneamente. Además los usuarios están en su entorno habitual. En la parte negativa, la falta de contacto directo dificulta el diagnóstico y pueden surgir dificultades que los usuarios no sean capaces de solucionar sin ayuda (como instalar o configurar programas).

**Evaluación heurística y revisión por expertos** En este caso la evaluación no se lleva a cabo por usuarios sino por expertos en usabilidad. Se han definido muchas heurísticas diferentes. Entre las más conocidas están las definidas por Nielsen:

- Visibilidad del estado del sistema.
- Correspondencia entre el sistema y el mundo real.
- Control del usuario y libertad.
- Consistencia y estándares.
- Prevención de errores.
- Reconocer mejor que recordar.
- Flexibilidad y eficiencia.
- Proporcionar ayuda a los usuarios a reconocer, diagnosticar y recuperarse de un error.

- Ayuda y documentación.

Estas evaluaciones tienen que ser llevadas a cabo por un experto que sea capaz de valorar cada heurística apropiadamente. Complementan, pero no sustituyen a las pruebas de usuario.

**Seguimiento de la mirada***(eye-tracking)*. En estas pruebas se registra la posición de la pantalla a la que mira el usuario.

- Usando los datos de varios usuarios se genera un mapa de calor en los que se puede apreciar qué zonas reciben más atención y cuáles pasan desapercibidas.
- Se puede saber si el usuario está leyendo el contenido o recorriendo la página rápidamente.

# Ejercicios propuestos

- 2.1. ¿Cuál de las siguientes no es una característica relacionada con la usabilidad?
- a) Facilidad de aprendizaje.
  - b) Posibilidad de manejo desde el teclado.
  - c) Interfaz de usuario intuitiva.
- 2.2. Marca la afirmación que consideres correcta:
- a) La usabilidad incluye accesibilidad. Para que una página se pueda considerar usable debe ser accesible.
  - b) La accesibilidad incluye usabilidad. Para que una página se pueda considerar accesible debe ser usable.
  - c) Una página puede ser accesible sin ser usable y viceversa.
- 2.3. Marca la afirmación que consideres incorrecta sobre el diseño centrado en el usuario:
- a) Es un proceso iterativo.
  - b) Solo cuenta con los usuarios en las fases iniciales del proyecto.
  - c) Aspira a conocer al usuario para poder responder a sus expectativas.
- 2.4. Los wireframes o bocetos de un sitio no deben incluir información sobre:
- a) Modos de navegación.
  - b) Colores y tipografía.
  - c) Categorías.
- 2.5. El método de ordenación de tarjetas se usa para:
- a) Definir la arquitectura de la información de un sitio web.
  - b) Identificar los requerimientos funcionales de la aplicación.
  - c) Probar los prototipos de la interfaz de usuario.
- 2.6. Marca la afirmación que consideres incorrecta sobre la evaluación de la usabilidad:
- a) La hacen exclusivamente los usuarios.



- b) Incluye evaluación por parte de usuarios y de expertos en usabilidad.
  - c) Se basa en factores objetivos y subjetivos.
- 2.7. Marca la que no consideres una buena pauta de usabilidad en un formulario:
- a) Ofrecer valores por defecto.
  - b) Poner un color de fondo diferente a cada campo para diferenciarlos.
  - c) Ofrecer ejemplos de formato para fechas.
- 2.8. Marca la que no consideres una buena pauta de usabilidad:
- a) Informar de la categoría a la que pertenece el contenido que está viendo el usuario.
  - b) Crear menús de navegación con pocas categorías principales y muchos subniveles.
  - c) Usar texto descriptivo en los vínculos.
- 2.9. Marca la opción que no es un método para el análisis de requerimientos de usuario:
- a) Personas.
  - b) Casos de uso.
  - c) Inventario de contenidos.
- 2.10. Los participantes es una prueba de usabilidad:
- a) Tienen que ser expertos en sistemas informáticos.
  - b) Son preferiblemente los desarrolladores del sistema.
  - c) Deben ser representativos de los usuarios del sistema.

# Actividades

- 2.1. Piensa en proyecto de página web y en quienes podrían ser sus usuarios. Intenta definir los grupos de usuario más relevantes y para cada uno de ellos enumeras sus objetivos, las tareas que quiere llevar a cabo en la página y sus capacidades y características.  
Algunas ideas son:
  - Una aplicación de correo web.
  - Una tienda online.
  - Noticias sobre deporte.
- 2.2. Piensa en los posibles casos de uso de la página anterior y descríbelos.
- 2.3. Estructura el contenido de la página en categorías y diseña los medios de navegación, especialmente la barra de navegación principal.
- 2.4. Realiza un boceto (wireframe) de la página del ejercicio 1. Cuando tengas una primera versión utiliza alguna herramienta gratuita online para hacer otro boceto más detallado.
- 2.5. Escoge una página que visites habitualmente y examínala para ver si cumple las pautas de la sección 2.9.

# Aplicaciones técnicas de usabilidad y accesibilidad en el entorno cliente

UF1843

El presente libro trata sobre el desarrollo de sitios web accesibles y usables. A través de sus páginas, veremos cómo accesibilidad y usabilidad son áreas diferentes, pero con puntos en común: la accesibilidad se centra en asegurar que los usuarios con algún tipo de discapacidad puedan acceder a los contenidos y funciones de las páginas en igualdad de condiciones; además de permitir llegar a más usuarios, en muchos casos se trata de un requerimiento legal. Por usabilidad se entiende la facilidad de uso de una página; una página usable está bien organizada, tiene una interfaz intuitiva y usarla es agradable y eficiente. El diseño centrado en el usuario ayuda a desarrollar páginas con estas características, como veremos a lo largo de este manual; los ejercicios prácticos que también se aportan contribuirán de manera decisiva al afianzamiento de los conceptos aprendidos, las soluciones están disponibles en [www.paraninfo.es](http://www.paraninfo.es).

Los contenidos de esta obra se corresponden con los de la UF 1843 Aplicaciones técnicas de usabilidad y accesibilidad en el entorno cliente, que se encuentra dentro del MF 0491\_3 Programación web en el entorno cliente y pertenece al certificado de profesionalidad Desarrollo de aplicaciones con tecnologías web, que está regulado por el RD 1531/2011 de 31 de octubre, y modificado a su vez por el RD 628/2013 de 2 de agosto.

Xabier Ganzoál García es funcionario de carrera y ejerce su labor docente en la Comunidad de Madrid.

Paraninfo  
[www.paraninfo.es](http://www.paraninfo.es)

[www.certificadoprofesionalidad.com.es](http://www.certificadoprofesionalidad.com.es)

ISBN: 978-84-281-9781-0



9 788428 197810 >