

# Accesibilidad en páginas web

Ana María Martínez Gómez  
Daniel Báscones García  
Víctor A. Gallego Alcalá  
Iñigo Zunzunegui Monterrubio

17 de enero de 2016

## Índice

<b>1. Manual de la programación web accesible</b>	<b>2</b>
1.1. Pautas generales	2
1.2. Pautas para aplicaciones no dirigidas	4
1.2.1. Manejo de la aplicación	4
1.2.2. Gráficos, enlaces gráficos y botones	4
1.2.3. Textos	5
1.2.4. Formularios	5
1.2.5. Vídeos	6
1.2.6. Tablas	6
1.3. Pautas para aplicaciones autodirigidas	6
<b>2. Herramientas de testing</b>	<b>7</b>
2.1. Ratio de contraste	7
2.1.1. WebAIM Color Contrast Checker	7
2.1.2. WAVE	7
2.1.3. Accessibility Developer Tools	8
2.1.4. Capybara-accessible	8
2.1.5. T.A.W.	8
2.2. Test de usabilidad	10
2.2.1. Tab navigate	10
2.2.2. Jaws	10
2.2.3. Voice Over	10

## Índice de figuras

1. Página web de la Universidad de Cambridge	3
2. Tabulación circular. Página de la Social Security de USA	3
3. Captcha sonoro	5
4. Errores y warnings proporcionados por WAVE	7
5. Errores con Capybara-accessible	8
6. Auditoría de la página de Amazon con T.A.W.	9
7. Revisores de pantalla más populares	10

## Resumen

La presente guía, que está disponible bajo una licencia [Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional](#), ha surgido de la convicción de que la accesibilidad de las aplicaciones informáticas son un requisito indispensable para su universalización. Será por tanto nuestro objetivo que se difunda entre los desarrolladores una metodología para la creación de aplicaciones web accesibles. Nos hemos centrado en aplicaciones de escritorio aunque muchas de las pautas que vamos a presentar son perfectamente válidas para cualquier dispositivo móvil.

Hemos dividido el documento en dos partes, en la primera hemos pretendido crear una guía metodológica del buen diseño tanto de aplicaciones no dirigidas como de las autodirigidas, mientras que en la segunda parte presentamos una serie de herramientas que permiten la evaluación de nuestras creaciones en lo que a accesibilidad se refiere.

Queríamos agradecer a Raquel Hervás y a Pablo Moreno Ger, profesores de la facultad de Informática de la UCM por su ayuda para enfocar el problema, a José Antonio Muñoz Sevilla, técnico del CIDAT<sup>1</sup> que nos transmitió parte de sus conocimientos acerca de la [tiflotecnología](#) y a María José García López, terapeuta ocupacional de la ONCE que derrochó paciencia y generosidad en un intento de que aprendiésemos los rudimentos del revisor de pantalla Jaws. La difusión de estas prácticas es fundamental en la creación de una conciencia de programación accesible, por esto queremos agradecer a Manuel Freire Morán y Pablo Moreno Ger, profesores de Ingeniería web y Aplicaciones web respectivamente, por su compromiso de divulgar y fomentar entre sus alumnos este conocimiento.

Hemos dejado conscientemente fuera de este trabajo tanto las *Normas, criterios y buenas prácticas para el desarrollo de aplicaciones accesibles*, como todo lo relativo a la *Interacción persona-computador* para personas con discapacidad visual. Animamos a que otros recojan el testigo y profundicen en esos temas.

## 1. Manual de la programación web accesible

Es necesario recalcar que estas pautas no se enumeran en orden de importancia o prioridad, sino que la accesibilidad se conseguirá en mayor medida cuanto más pautas se consideren a la hora de hacer las especificaciones y diseño de la aplicación

### 1.1. Pautas generales

A continuación listaremos una serie de directrices comunes a los dos tipos de aplicaciones, tanto las no dirigidas como las autodirigidas:

- La aplicación ha de poder manejarse exclusivamente con el teclado, lo que no implica la anulación del ratón, más aún ambas modalidades deben coexistir.
- Es conveniente tener un *Menú Principal* que aparezca en todas las secciones y desde el que se pueda acceder a cualquier apartado de la aplicación. Un ejemplo de página en la que se ha tenido en cuenta esta pauta podemos verla en la [Figura 1](#).
- La configuración de colores de la aplicación debe poder cambiarse (fondos, textos, etc) o, al menos hacer que la aplicación utilice la que el usuario está empleando en el sistema operativo.
- La aplicación debe disponer de una opción que permita al usuario seleccionar las opciones de visualización de textos, opciones de colores de la aplicación, y opciones de impresión, en tinta o en braille, o si se van a imprimir pantallas que posteriormente se adaptarán con un horno Fúser.
- La elección de las teclas que se usan para la navegación en la pantalla y en los menús ha de seguir un orden lógico y aquella debe ser circular, volviendo al primer elemento de la navegación después del último. Un ejemplo de buena aplicación de este principio podemos verlo en la [Figura 2](#)

---

<sup>1</sup>Centro de Investigación, Desarrollo y Aplicación Tiflotecnología de la ONCE

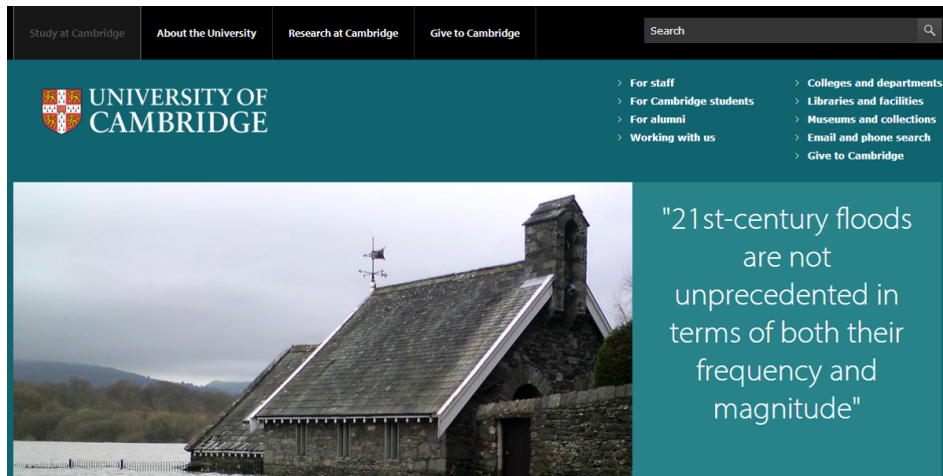


Figura 1: Página web de la Universidad de Cambridge



Figura 2: Tabulación circular. Página de la Social Security de USA

## 1.2. Pautas para aplicaciones no dirigidas

En esta sección vamos a desarrollar las pautas a seguir para conseguir la accesibilidad de aplicaciones que se van a manejar con la ayuda de un revisor de pantalla.

### 1.2.1. Manejo de la aplicación

- Permitir al usuario la posibilidad de ejecutar la aplicación a pantalla completa y ampliar campos de la misma.
- En la medida de lo posible se recomienda usar controles estándar del sistema operativo para el que se desarrolle la aplicación, facilitando el uso integral con el teclado como se ha mencionado anteriormente.
- En aplicaciones complejas se debe permitir que el usuario acceda a las acciones más críticas o habituales mediante el uso de las teclas rápidas del revisor de pantalla.
- Es aconsejable realizar el diseño visual de la aplicación para una configuración de pantalla de 800 × 600 píxeles, ya que es la más comúnmente utilizada por las personas ciegas y deficientes visuales.
- No sobrecargar las pantallas de la aplicación con excesivos enlaces a otras secciones (salvo en el caso de índices). Se recomienda que no haya más de cinco o seis en cada pantalla. En este sentido se recomienda eliminar enlaces redundantes dentro de la misma página o pantalla. De nuevo la **Figura 1** es una buena muestra de cumplimiento de esta pauta.

### 1.2.2. Gráficos, enlaces gráficos y botones

- Todos los enlaces gráficos deben tener un texto alternativo descriptivo de la acción que realizan.
- Deben tener un tamaño grande para ser fácilmente identificables en la pantalla.
- Es aconsejable que los enlaces aumenten su tamaño al recibir el foco.
- Los botones o enlaces que realizan la misma acción deben ser iguales en todas las pantallas o páginas de la aplicación, por ejemplo Volver, ir a la página principal, imprimir, etc.
- La forma de los botones y enlaces gráficos debe ser sencilla, preferiblemente formas geométricas básicas.
- Deben tener destacados los contornos de los diferentes elementos
- El color del botón o enlace gráfico debe contrastar con el color de fondo de la pantalla en la que se encuentra.
- Si el botón contiene una imagen representativa de la acción que desempeña, esta debe contrastar con el color de fondo del botón.
- No utilizar colores como la única forma de indicar el significado de un elemento (Un botón rojo y otro verde no significan nada para un usuario ciego o deficiente visual).



Figura 3: Captcha sonoro

### 1.2.3. Textos

- No sobreimprimir textos sobre imágenes. El texto debe presentarse sobre fondos lisos de un único color.
- Permitir el empleo de magnificadores de pantalla o, en su defecto, utilizar tamaños de letra grandes (mínimo: 14) y colores adecuados.
- Los textos deben ser “editables”, para permitir su lectura por frases cortas, palabra a palabra e incluso letra a letra.
- Para textos extensos, es preferible la presentación en única columna, recurriendo a la lectura mediante desplazamiento vertical. Un deficiente visual se pierde al cambiar de línea si ésta es muy larga.
- Las fórmulas matemáticas, de Física o Química y las frases musicales precisan de una edición especial por “Línea braille”, con un editor adecuado. De no ser posible, deben ser considerados como elemento gráfico, con su texto explicativo como se ha comentado con anterioridad
- Aquellos elementos y aspectos gráficos con fines de estructuración y resaltado de textos (recuadros, fondos, cambios de color o tipográficos, etc.) no tienen por qué reflejarse en la edición por línea braille o describirse vía audio, salvo que se les confiera un carácter fundamental; en cuyo caso, es preferible la ilustración sonora a la mera descripción.

### 1.2.4. Formularios

- Asociar a cada elemento del formulario su etiqueta correspondiente.
- Separar un elemento del formulario de la etiqueta de otro.
- Las listas desplegadas deben tener un botón asociado para ejecutar la acción asociada a la opción seleccionada en la lista.
- Evitar en la medida de lo posible las listas de selección múltiple.
- Es conveniente encerrar el formulario dentro de un cuadro de un color que contraste con el de fondo de la pantalla, para su fácil localización.
- Sé inteligente con tus **Captchas**. No servirá de nada que el usuario ciego utilice su tiempo en rellenar un formulario si al final le colocas un captcha no accesible. Utiliza un captcha con descripción sonora como el de la **Figura 3** para permitirles completar el formulario.

### 1.2.5. Vídeos

- El vídeo debe tener un botón asociado para comenzar su visualización, y no comenzar automáticamente.
- Los vídeos deben tener un tamaño grande en la pantalla.
- Una locución sonora (audiodescripción) que describa, en sincronía con la imagen, la representación o variaciones que se van produciendo en el vídeo.
- Posibilidad de ralentizar la proyección/verbalización.
- Posibilidad de repetir la proyección/verbalización.

### 1.2.6. Tablas

Estas pautas están basadas en las recomendaciones del **WAI**<sup>2</sup> para diseño de tablas accesibles en HTML.

- No utilizar tablas si no es estrictamente necesario para estructurar la información de forma comprensible.
- Distinguir entre las celdas de los encabezados y las de datos propiamente dichos.

## 1.3. Pautas para aplicaciones autodirigidas

Las *aplicaciones autodirigidas* son aquellas que, mediante mensajes sonoros, dirigen al ciego o deficiente visual por los menús de la misma sin necesidad de un revisor de pantalla.

- El acceso a la aplicación debe ser inmediato o con el menor recorrido posible desde el arranque, y la salida de la misma sencilla, aunque haya de ser verificada.
- Para el manejo de la aplicación debe utilizarse el menor número de teclas posible. Asimismo debe procurarse que éstas sean fácilmente localizables (cursores, teclado numérico, Enter y barra espaciadora son siempre buenas opciones).
- Las pantallas han de tener todas un título identificativo que deberá verbalizarse al iniciarse.
- Cualquier cambio que se produzca en la pantalla, automáticamente o por acción del usuario debe ser informado mediante un sonido o verbalmente.
- Hay que dar la opción al usuario de escuchar tantas veces como desee los mensajes de texto y la información relevante que aparece en la pantalla.
- Las imágenes deben tener asociado un fichero que describa su contenido. Igualmente los vídeos deben tener un fichero con la narración de lo que está ocurriendo.
- La finalización de una acción debe informarse al usuario mediante un sonido, sea cual sea el resultado de esta.
- Los elementos comunes a todas las pantallas deben tener la misma localización en cada una de ellas.

---

<sup>2</sup> *Web Accessibility Initiative*: Se trata de una actividad desarrollada por el **W3C**, cuyo objetivo es facilitar el acceso de las personas con discapacidad, desarrollando pautas de accesibilidad, mejorando las herramientas para la evaluación y reparación de accesibilidad Web, llevando a cabo una labor educativa y de concienciación en relación a la importancia del diseño accesible de páginas Web, y abriendo nuevos campos en accesibilidad a través de la investigación en este área.

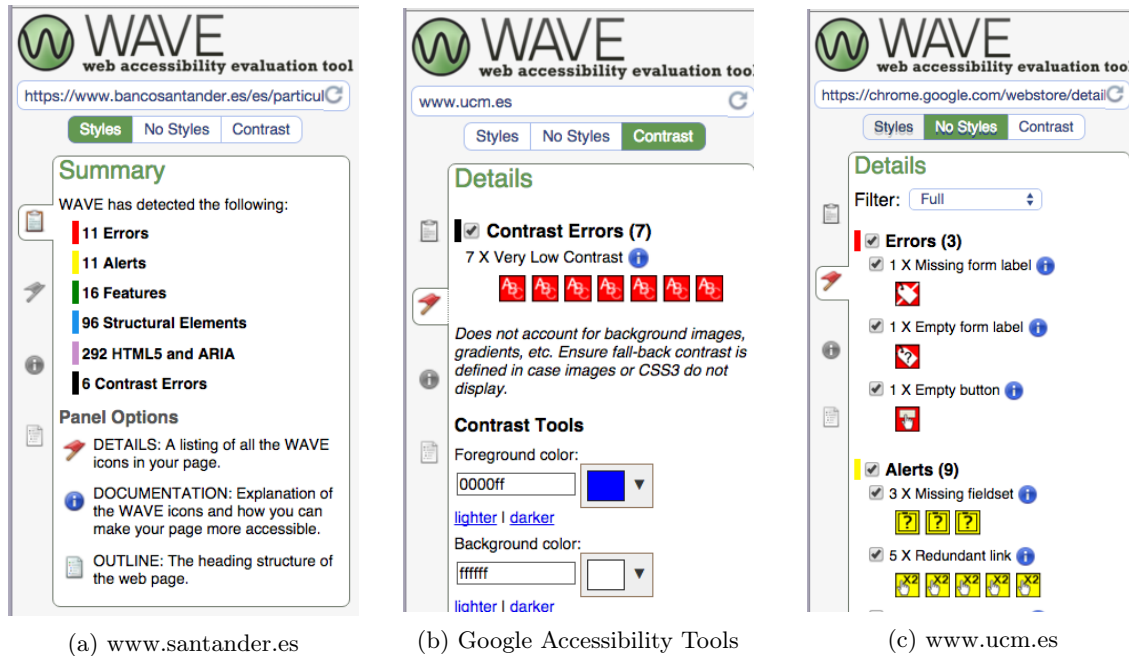


Figura 4: Errores y warnings proporcionados por WAVE

## 2. Herramientas de testing

El desarrollador implicado en la accesibilidad web para personas ciegas o con baja visión tiene a su disposición múltiples herramientas para ayudarle a testear su trabajo.

### 2.1. Ratio de contraste

Como hemos comentado en las anteriores pautas el contraste de nuestra página será fundamental para que las personas con baja visión puedan identificar los elementos de una página web.

#### 2.1.1. WebAIM Color Contrast Checker

**WebAIM Color Contrast Checker** evalúa el contraste de los elementos de nuestra página así como la idoneidad del tamaño de letra. Para ello tendremos que introducir el RGB (formato hexadecimal) tanto del texto como del fondo de los elementos que queramos testear, pudiendo modificar tanto el color como la luminosidad con un selector de color.

- **ColorZilla** Es una herramienta perfecta para obtener el valor del color de cualquier elemento de una página

#### 2.1.2. WAVE

**WAVE** Puede analizar los ratios de contraste de todos los elementos de una página a la vez. Instalamos una extensión para el navegador y visitando cualquier página nos da un informe detallado de los errores y warnings de los elementos de la página, del porqué de su importancia y de cómo solucionarlos. Podemos ver un ejemplo de diagnóstico del contraste y del estilo de tres páginas en la **Figura 4**. Al usar la extensión, no se envía ninguna información a los servidores de WAVE, pero si



```

skinner :: ~/workspace/whiteboard <inaccessible> » rspec spec/features/standups_spec.rb
F

Failures:

  1) standups creates new standups
     Failure/Error: click_link('New Standup')
     Capybara::Accessible::InaccessibleError:
       Found at http://127.0.0.1:57852/standups/new

     *** Begin accessibility audit results ***
     An accessibility audit found
     Errors:
     Error: AX_TEXT_01 (Controls and media elements should have labels) failed on the following element:
     #standup_ip_key
     See https://code.google.com/p/accessibility-developer-tools/wiki/AuditRules#AX_TEXT_01:_Controls_and_media_elements_should_have_labels for more information.

     Warnings:
     Warning: AX_IMAGE_01 (Meaningful images should not be used in element backgrounds) failed on the following element:
     #new_standup > .block:nth-of-type(5) > .btn.btn-primary.btn-block

```

Figura 5: Errores con Capybara-accessible

esto no nos supone ningún problema y la web está online, podemos realizar el análisis directamente sin instalar el plugin entrando en el enlace anterior.

### 2.1.3. Accessibility Developer Tools

**Accessibility Developer Tools** es un proyecto que surge desde el equipo de Chrome de Google. Es un conjunto de asertos escritos en Javascript y encapsulados en un plugin de Chrome, que puedes obtener en la **Chrome Web Store**.

Cuando lo ejecutas, lanza la consola de desarrollador del navegador, con la cual cualquier desarrollador web estará familiarizado. Además integra y señala fallos y advertencias en la página. Haciendo click en cualquiera de ellos abre el elemento en el inspector para ver en qué parte del código se encuentra el problema.

Además da sugerencias sobre cómo arreglarlo, y gracias a la vista previa activa del navegador se pueden ir viendo los arreglos en directo. Por ejemplo para contraste de color hay unos asertos incluidos en la biblioteca que permiten hacer click en un botón y cambiar el color del texto en el navegador, para hacerte una idea de cómo se vería más claro, oscuro, de otro color...

### 2.1.4. Capybara-accessible

**Capybara-accessible** es una gema de Ruby on Rails que introduce tests de accesibilidad en los test de integración. Las auditorías se hacen automáticamente con *Google's Accessibility Developer Tools*, que muestra los errores de accesibilidad dando valiosos detalles con enlaces a documentación para cómo solventarlos, tal y como se puede ver en la **Figura 5**.

### 2.1.5. T.A.W.

**T.A.W.** es una familia de herramienta para el análisis de la accesibilidad de sitios web. Nos aporta



dos tipos de errores:

- Automáticos: Problemas de accesibilidad que la herramienta detecta por si sola y que deben ser solucionados.
- Manuales: La herramienta señala la existencia de un posible problema que el evaluador debe confirmar o descartar.

Si lo deseas, puedes poner en tu sitio web el logotipo que indica que has validado la accesibilidad con TAW y el grado alcanzado. Existe uno por cada nivel de accesibilidad (A, doble A y triple A ) y puedes elegir como logo la alternativa que más te guste (CSS, gráfico grande y gráfico pequeño).

No necesita instalación, es fácil de usar y se puede ver una muestra en la **Figura 6**.

**t.a.w.**  
CTIC Centro Tecnológico

*Vista Detalle*

Información del análisis

Recurso: <http://www.amazon.es/>  
 Fecha: 12/12/2015 13:17  
 Pautas: WCAG 2.0  
 Nivel del análisis: AA  
 Tecnologías: HTML, CSS

Tipología	Comprobación
<b>1.1.1 - Contenido no textual</b>	
Imágenes	Imágenes sin atributo alt <b>f</b>
	Elemento area sin equivalente textual <b>f</b>
	Imágenes que pueden requerir descripción larga <b>f</b>
	Imágenes con alternativas extensas <b>f</b>
	Imágenes con alt vacío <b>f</b>
Formularios	Controles de formulario sin etiquetar <b>f</b>
Navegación	Enlaces consecutivos de texto e imagen al mismo recurso <b>f</b>
<b>1.3.1 - Información y relaciones</b>	
Tablas	Existencia de resumen para las tablas de datos <b>f</b>
Formularios	Controles de formulario sin etiquetar <b>f</b>

Figura 6: Auditoría de la página de **Amazon** con T.A.W.



(a) Jaws for Windows



(b) VoiceOver

Figura 7: Revisores de pantalla más populares

## 2.2. Test de usabilidad

El siguiente paso en la evaluación de nuestra página web será la usabilidad. Para poder medir cómo de usable es nuestra página solo podemos recomendar al programador que la pruebe. Son parte esencial los revisores de pantalla que se muestran en la [Figura 7](#).

### 2.2.1. Tab navigate

No se trata de una herramienta de testing como tal sino tan solo un reto para el programador: ser capaz de navegar por su página sin hacer uso del ratón. Es la mejor prueba que puede hacerse al respecto.

### 2.2.2. Jaws

[Jaws](#), con más de 20 años de existencia, es probablemente el mejor revisor de pantalla para Windows del mercado y sin duda uno de los más extendidos entre los afiliados a la ONCE. Su precio es muy alto, pero ofrece una versión de prueba durante unos días. El programador puede sentir rechazo a navegar su página con este revisor de pantalla por el desconocimiento de la aplicación. Puede aprender lo básico de su uso en esta [breve guía](#).

### 2.2.3. Voice Over

[Voice Over](#) es el revisor de pantalla que instala Apple en todos sus dispositivos. Recomendamos probar la página web con esta herramienta debido a que es la herramienta de Apple que llevan instaladas todos sus dispositivos. Si se ha optado por desarrollar en un Mac es sin duda la mejor herramienta que podemos recomendarle por su facilidad de uso y porque es gratuita.