

# GUÍA DE ADAPTACIÓN A WCAG 2.1 DESDE WCAG 2.0

*(Versión 2, abril 2023)*



GOBIERNO  
DE ESPAÑA

MINISTERIO  
DE ASUNTOS ECONÓMICOS  
Y TRANSFORMACIÓN DIGITAL

MINISTERIO  
DE HACIENDA  
Y FUNCIÓN PÚBLICA

## Guía de adaptación a WCAG 2.1 desde WCAG 2.0

(Versión 2, abril 2023)

Elaboración y coordinación de contenidos:  
Secretaría General de Administración Digital (SGAD)

Publicación disponible en el Portal de Administración Electrónica (PAe)  
<http://administracionelectronica.gob.es>

### Edita:

© Ministerio de Asuntos Económicos y Transformación Digital  
Secretaría General Técnica  
Centro de Publicaciones

© Ministerio de Hacienda y Función Pública  
Subdirección de Información, Documentación y Publicaciones  
Secretaría General Técnica  
Centro de Publicaciones

Colección: *Administración electrónica*

**NIPO (MINECO):** 094-23-046-8

**NIPO (MHFP):** 137-23-056-X



El presente documento está bajo la licencia Creative Commons Reconocimiento-No comercial-Compartir Igual versión 4.0 España.

Usted es libre de:

- Copiar, distribuir y comunicar públicamente la obra.
- Hacer obras derivadas.

Bajo las condiciones siguientes:

- Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciadador (pero no de forma que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).
- Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, solo puede distribuir la obra generada bajo una licencia idéntica a esta.

Al reutilizar o distribuir la obra debe dejar claros los términos de la licencia.

Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor.

Nada en esta licencia menoscaba o restringe los derechos morales del autor.

Esta descripción es un resumen legible por humanos del texto legal (licencia completa) disponible en:

## ÍNDICE

---

<b>ÍNDICE DE IMÁGENES .....</b>	<b>5</b>
<b>1. INTRODUCCIÓN.....</b>	<b>6</b>
<b>2. OBJETIVO DE LA GUÍA .....</b>	<b>7</b>
<b>3. CAMBIOS PRINCIPALES RESPECTO A LAS WCAG 2.0 .....</b>	<b>8</b>
<b>4. NUEVOS REQUISITOS .....</b>	<b>11</b>
4.1. Orientación .....	11
4.2. Identificar el propósito de la entrada.....	11
4.3. Reajuste de texto.....	13
4.4. Contraste de contenido no textual .....	15
4.5. Espaciado del texto .....	17
4.6. Contenido con hover o focus.....	18
4.7. Atajos con teclas de caracteres .....	20
4.8. Gestos del puntero .....	20
4.9. Cancelación del puntero .....	21
4.10. Etiqueta en el nombre .....	22
4.11. Activación mediante movimiento .....	24
4.12. Mensajes de estado.....	25
<b>5. ANEXO GUÍA RÁPIDA WCAG 2.1.....</b>	<b>28</b>

**Tabla 1 Control de cambios**

Versión 2	Descripción de cambios
Enero 2023	Actualización de UNE-EN 301549:2019 por UNE-EN 301549:2022 Actualización de EN 301549 v2.1.2 (2018-08) por EN 301549 v3.2.1 (2021-03)
Abril 2023	Actualización del formato de estilos del documento.

## ÍNDICE DE IMÁGENES

---

Ilustración 1. Ejemplo de uso de media queries y CSS grid .....	15
Ilustración 2. En una gráfica de líneas cada línea debe tener contraste y en una gráfica de tarta se debe poder diferenciar cada parte de la gráfica (imágenes de ejemplo obtenidas de la documentación de las WCAG 2.1) .....	17
Ilustración 3. Al ajustar las características del texto no se pueden producir solapamientos, cortes o desbordamientos del texto que provoquen pérdida de contenido o funcionalidad.....	18
Ilustración 4. Se han de proporcionar mecanismos de interacción alternativos que no impliquen gestos complejos como botones para manejar y desplazarse por un mapa. ....	21
Ilustración 5. Guía rápida de aplicación de las WCAG 2.1 .....	30

## 1. INTRODUCCIÓN

---

El 20 de septiembre de 2018, un día después de su publicación en el Boletín Oficial del Estado, entró en vigor el [Real Decreto 1112/2018, de 7 de septiembre, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público](#), que traspone al ordenamiento jurídico español la [Directiva \(UE\) 2016/2102](#).

En el Real Decreto se obliga a que todas las páginas web de la Administración Pública o dependientes de ésta sean accesibles para las personas con discapacidad. Aplica a todos los sitios web del sector público, desde aquellos pertenecientes a las administraciones, tribunales y servicios de policía a los de los hospitales, universidades y bibliotecas públicas, haciendo que sean accesibles a todos los ciudadanos, especialmente a aquellos con dificultades auditivas, visuales o funcionales.

En el Real Decreto se establecen como requisitos de accesibilidad a cumplir los indicados en la norma armonizada publicada por la Comisión Europea el 21 de diciembre de 2018 mediante la [Decisión de Ejecución \(UE\) 2018/2048, sobre la norma armonizada aplicable a los sitios web y a las aplicaciones para dispositivos móviles](#). El 11 de agosto de 2021 la Comisión Europea publicó **la Decisión de Ejecución (UE) 2021/1339**, por la que se modifica la Decisión de Ejecución (UE) 2018/2048 en lo que atañe a la norma armonizada, pasando a ser la norma **EN 301549 v3.2.1** "Requisitos de accesibilidad de productos y servicios TIC" que en España se corresponde con la **UNE-EN 301549:2022**.

**Nivel de conformidad a cumplir:** en **la tabla A.1 del anexo A de la norma EN 301549 v3.2.1** se recogen todos los requisitos esenciales de la Directiva (UE) 2016/2102. Desde el apartado 9.1 hasta el apartado 9.6 se recogen la práctica totalidad de los criterios de conformidad A y AA de las WCAG 2.1. Además, la tabla A.1 hace referencia a otros requisitos de accesibilidad que no provienen de las WCAG (requisitos de los apartados 5, 6, 7, 9.6, 10, 11 y 12).

Hasta la publicación de este nuevo Real Decreto las Administraciones Públicas debían cumplir los requisitos de nivel A y nivel AA de la UNE 139803:2012 que son equivalentes a las WCAG 2.0.

Por tanto, en esta guía se describirán las novedades a nivel técnico más importantes introducidas por las WCAG 2.1.

## 2. OBJETIVO DE LA GUÍA

---

El objetivo de esta guía es ofrecer, a aquellos responsables de sitios web que ya son conformes con respecto a las WCAG 2.0, una guía resumen útil que les sirva para la evolución o adaptación de sus sitios o aplicaciones web a las nuevas WCAG 2.1.

### 3. CAMBIOS PRINCIPALES RESPECTO A LAS WCAG 2.0

---

Las WCAG 2.1 surgen con el objetivo de mejorar la accesibilidad principalmente de tres grupos de usuarios:

- Personas con discapacidad cognitiva o del aprendizaje
- Personas con baja visión
- Personas con discapacidad que acceden desde dispositivos móviles

Se consideraba que las WCAG 2.0 no cubrían como era de esperar las necesidades de estos grupos de personas. Asimismo, en los últimos años las tecnologías y las formas de acceder al contenido web han variado de forma significativa dando lugar a situaciones no contempladas originalmente en las WCAG 2.0.

Las WCAG 2.1 pretenden cubrir un mayor conjunto de recomendaciones para hacer la web más accesible. Estas Pautas aspiran a mejorar la accesibilidad del contenido web tanto en entornos de escritorio y portátiles como en tablets y dispositivos móviles.

Entre las novedades que encierra esta nueva versión con respecto a su antecesor están:

- Se han añadido 17 nuevos criterios de conformidad (5 nuevos criterios de conformidad de nivel A, 7 de nivel AA y 5 de nivel AAA)
- Se han añadido nuevos términos al glosario
- Se añade una nueva pauta (2.5 "*Modalidades de entrada*") al segundo principio (Operabilidad), pasando de tener 12 pautas a 13
- Se añade en el enunciado del criterio 1.3.3 específicamente el color y se quita en consecuencia la nota del criterio 1.4.1
- Se añade una nota adicional en el requisito 5.2.2 "*Páginas completas*"
- Se añade un punto adicional en "*Componentes opcionales de la declaración de conformidad*" de la sección de conformidad

Cabe mencionar que en esta nueva versión no se han modificado los criterios de conformidad de las WCAG 2.0, ni siquiera su numeración. Las WCAG 2.1 se



pueden considerarse como un superconjunto que contiene a las WCAG 2.0. Por lo tanto, como las WCAG 2.1 extienden las WCAG 2.0, no existen requisitos que sean incompatibles entre una versión y la otra.

Esta estrategia de ampliación ayuda a dejar claro que si una página web es conforme a las WCAG 2.1 también será conforme a las WCAG 2.0. Y en sentido contrario, si una página web es conforme a las WCAG 2.0 se podrá actualizar a las WCAG 2.1 sin incompatibilidades.

Las WCAG 2.1 introducen 17 nuevos criterios de conformidad, 12 de los cuales tienen un nivel de conformidad A o AA. Estos son:

- Pauta 1.3 Adaptable:
  - 1.3.4 Orientación (AA)
  - 1.3.5 Identificar el Propósito de la Entrada (AA)
  - 1.3.6 Identificar el Propósito (AAA)
- Pauta 1.4 Distinguible:
  - 1.4.10 Reajuste del texto (AA)
  - 1.4.11 Contraste de contenido no textual (AA)
  - 1.4.12 Espaciado de texto (AA)
  - 1.4.13 Contenido con Hover o Focus (AA)
- Pauta 2.1 Teclado Accesible:
  - 2.1.4 Atajos con teclas de caracteres (A)
- Pauta 2.2 Tiempo Suficiente:
  - 2.2.6 Tiempos de inactividad (AAA)
- Pauta 2.3 Convulsiones y reacciones físicas:
  - 2.3.3 Animación iniciada desde Interacciones (AAA)
- Pauta 2.5 Modalidades de entrada:
  - 2.5.1 Gestos del puntero (A)

- 2.5.2 Cancelación del puntero (A)
- 2.5.3 Etiqueta en el Nombre (A)
- 2.5.4 Activación mediante movimiento (A)
- 2.5.5 Tamaño del objetivo (AAA)
- 2.5.6 Mecanismos de entrada simultáneos (AAA)
- Pauta 4.1 Compatible:
  - 4.1.3 Mensajes de estado (AA)

Los nuevos criterios de conformidad se incluyen entre las Pautas ya existentes excepto para la pauta 2.5 "*Modalidades de entrada*", la cual es una nueva pauta introducida en las WCAG 2.1. Está dedicada específicamente a facilitar la interacción a través de diferentes dispositivos de entrada más allá de la interacción mediante teclado, especialmente desde dispositivos móviles.

En esta guía se detallan los aspectos técnicos más relevantes para aplicar los nuevos criterios de conformidad de las WCAG 2.1 para un nivel de conformidad AA.

## 4. NUEVOS REQUISITOS

---

### 4.1. ORIENTACIÓN

#### *CRITERIO DE CONFORMIDAD 1.3.4 (NIVEL AA)*

El contenido **no puede restringir su visualización y funcionalidad a una única orientación de la pantalla** (horizontal o vertical) a no ser que dicha orientación específica sea esencial.

Algunos sitios web o aplicaciones están diseñados y configurados de forma que requieren que el dispositivo se use con una determinada orientación, vertical u horizontal. Sin embargo, existen usuarios que tienen los dispositivos anclados en una posición fija (p. ej. sobre una silla de ruedas) y no pueden modificar su orientación. Por ejemplo, un usuario que vaya en silla de ruedas y use una tablet anclada en el reposa manos en posición vertical no podrá acceder a un sitio web que sólo se muestre correctamente en posición horizontal.

Por tanto, las maquetaciones de los sitios y aplicaciones web se han de realizar empleando técnicas (CSS - Cascading Style Sheets) que no restrinjan su visualización a una determinada orientación y sin impedir o dificultar su uso en otras posibles orientaciones.

Se consideran como **excepción** a este requisito aquellos sitios o aplicaciones en los que es **esencial mostrar su contenido en una determinada orientación**. Entre los ejemplos que se documentan en las pautas de accesibilidad se incluye el caso de un cheque bancario, una aplicación de piano, una presentación para mostrarse en un proyector o televisión o un contenido de realidad virtual donde no son aplicables múltiples posibles orientaciones.

Es necesario distinguir entre la responsabilidad de los autores de contenido web respecto a no restringir la orientación del contenido y los mecanismos específicos de dichos dispositivos empleados para bloquear la orientación. Si un usuario bloquea su dispositivo en una determinada orientación se espera que todas las aplicaciones respeten dicha selección y se muestren de forma acorde.

### 4.2. IDENTIFICAR EL PROPÓSITO DE LA ENTRADA

#### *CRITERIO DE CONFORMIDAD 1.3.5 (NIVEL AA)*

En las WCAG 2.1 es necesario identificar el **propósito de los campos** de introducción de datos de forma que se pueda **determinar mediante programación** la finalidad de aquellos que solicitan información acerca de las personas. De esta forma los agentes de usuario podrán obtener esta información

y facilitar la interacción a los usuarios de tecnologías de apoyo. Por ejemplo, autocompletando los campos cuyos datos son conocidos.

Para aplicar el marcado semántico correspondiente en algunos casos puede parecer suficiente con los nuevos tipos de elementos `INPUT` de HTML5 (tipo *tel*, *email*, etc.). Sin embargo, aunque estos campos aportan cierta información sobre la naturaleza del tipo de dato a introducir, las categorías que se pueden emplear son demasiado genéricas. Por ejemplo, sirven para indicar que un campo se trata de un email o un teléfono, pero no aclaran cuál es el dato concreto al que se refieren (¿teléfono o email del usuario o de otra persona?).

Por lo tanto es preciso emplear técnicas alternativas como el [atributo `autocomplete` de HTML 5.2](#) junto con un valor que indique el tipo de información solicitada. Por ejemplo, algunos de los valores posibles para el atributo `autocomplete` son *name* para indicar el nombre completo, *honorific-prefix* para el título ("Sr.", "Sra.", "Dr.", etc.), *given-name* para el nombre propio, *family-name* para el apellido, *nickname* para el apodo, *street-address* para la dirección, etc.).

### Ejemplo de código 1

```
<form method="post" action="paso2">
  <div>
    <label for="nombre">Nombre</label>
    <input id="nombre" type="text" autocomplete="given-name" ... >
  </div>
  <div>
    <label for="apellidos">Apellidos:</label>
    <input id="apellidos" type="text" autocomplete="family-name"
      ... >
  </div>
  <div>
    <label for="direccion">Dirección:</label>
    <input type="text" id="direccion" autocomplete="street-
address"
      ... >
  </div>
  <div>
    <input type="submit" value="Continuar..." >
  </div>
</form>
```

Con el atributo `autocomplete` se establece una relación programática entre el elemento `<INPUT>` y un término de una taxonomía predefinida que identifica su finalidad. De esta forma el campo de edición dispondrá de un identificador común y estándar asociado al mismo, independientemente del idioma de la página o de los textos visibles empleados para etiquetar el campo.

Esto permite que el campo de edición sea identificado mediante programación por los agentes de usuario y tecnologías de apoyo que serán capaces de reconocer su finalidad. Esto no sólo es útil y práctico para autocompletar los campos con valores anteriormente introducidos por los usuarios sino que también permite otras mejoras para facilitar la interacción como, por ejemplo, incluir automáticamente etiquetas más familiares o iconos que aclaren el propósito de los campos a las personas con problemas cognitivos (p. ej. icono de un teléfono en campos de tipo `tel`).

Este requisito sólo aplica a aquellos campos de formulario que solicitan información acerca del usuario y sólo es necesario usar el atributo `autocomplete` en campos cuya finalidad se corresponda con alguna de las definidas en la especificación de HTML5.

### 4.3. REAJUSTE DE TEXTO

#### CRITERIO DE CONFORMIDAD 1.4.10 (NIVEL AA)

El contenido de los sitios web debe poder presentarse sin pérdida de información o funcionalidad y sin necesidad de realizar scroll en dos dimensiones en áreas de visualización reducidas (320x256 píxeles CSS). Si bien este criterio está pensado inicialmente para mejorar la experiencia de las personas con baja visión asegurando que se pueda hacer zoom hasta al menos un 400% del tamaño original sin que se produzca un doble scroll (partiendo de una resolución de 1280x1024), en la práctica e implícitamente también supone que la página se debe "adaptar" correctamente a diferentes tamaños de ventana.

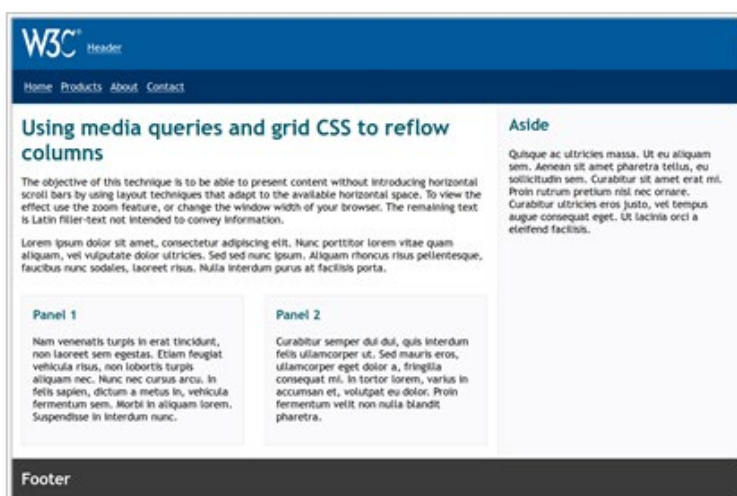
En las WCAG 2.1 se indica que permitir el reajuste del contenido se conoce también como *Diseño Web Adaptable (Responsive Web Design)* y se considera como la forma más efectiva para conseguir cumplir este criterio. Así, entre las técnicas indicadas para CSS (en español "hojas de estilos en cascada") está el uso de *media queries* (en español "consultas de medios") para establecer puntos de ruptura y reformatear el contenido para diferentes anchos de visualización, definiendo unos estilos que solo se aplicarán en el caso de que se cumplan unas condiciones definidas. Estos puntos de ruptura se aplicarán de igual forma tanto si se reduce el tamaño de la ventana como si se hace zoom sobre el contenido.

Por tanto, la maquetación de las plantillas empleadas en los sitios web ha de respetar este requisito. Deben poder adaptarse a diferentes tamaños de ventana o niveles de zoom sin que se pierda contenido o funcionalidad y sin que se produzca un doble scroll. Para ello se pueden emplear diferentes técnicas como la **maquetación fluida**, mediante **media queries** y **grid layout** de CSS o maquetación con el **modelo Flexible Box** de CSS (Flexbox).

## Ejemplo de código 2

```
/* Ejemplo de uso de media queries y grid layout */
@media all and (min-width: 576px) {
  /*
   Estilos para todo tipo de dispositivo con un
   ancho mínimo de ventana de 576px.
  */
  .subgrid {
    grid-template-columns: minmax(0, 1fr) minmax(0, 1fr);
    margin-bottom: 1rem;
  }
  .subgrid-item {
    padding-bottom: 0.25rem;
  }
}

@media all and (min-width: 992px) {
  /*
   Estilos para todo tipo de dispositivo con un
   ancho mínimo de ventana de 992px.
  */
  .grid {
    grid-template-areas:
      'header header header'
      'main main aside'
      'footer footer footer';
    grid-template-columns: minmax(0, 1fr) minmax(0, 1fr) minmax(0,
1fr);
  }
}
```



### Ilustración 1. Ejemplo de uso de media queries y CSS grid

Puede ver el [ejemplo en funcionamiento del uso de media queries y CCS grid](#).

Adicionalmente, aunque el contenido se vea correctamente en un dispositivo móvil no se debe bloquear la posibilidad de hacer zoom sobre el mismo. Se considera un error emplear, por ejemplo, elementos meta con `"maximum-scale"` o `"minimum-scale"` o bien con `"user-scalable=no"` o `"user-scalable=0"`.

No obstante, es importante tener en cuenta que este requisito contempla como **excepciones aquellas partes del contenido que requieren de una presentación y maquetación en dos dimensiones** para poder transmitir su significado o para poder emplearse de forma correcta.

Dentro de esta excepción se pueden incluir las **imágenes y contenidos multimedia** ya que por naturaleza tienen dos dimensiones, aunque es posible su redimensión hasta adaptarlos al tamaño de la ventana. Una excepción más clara es el caso de las **tablas de datos**, en las que las relaciones entre las celdas se establecen en un contexto bidimensional. Por lo tanto, las tablas de datos, en especial las tablas de datos complejas, están fuera del alcance de este requisito.

Otro contenido que se puede considerar como una excepción son aquellas **interfaces de usuario complejas** como las que proporcionan barras de herramientas para editar contenidos que se deben mostrar simultáneamente junto con la barra de herramientas (editores de texto, editores gráficos, etc.). Por otra parte, aunque no se mencione de forma explícita en las WCAG 2.1, se pueden considerar también como excepción los **formularios o trámites de notoria complejidad**, tanto a nivel de interacción como de visualización, de forma que no son razonablemente operables desde dispositivos móviles.

NOTA: Esta última excepción acerca de los formularios o trámites de notoria complejidad no está incluida explícitamente en las WCAG 2.1. Por tanto, desde un punto de vista estricto sí deberían poder adaptarse a dispositivos móviles. Sin embargo, desde el Observatorio de Accesibilidad se considera que existen ciertos trámites de la Administración Electrónica cuya complejidad de comprensión, operación e interacción desde dispositivos móviles iguala o supera a los casos excepcionales detallados en las WCAG 2.1.

## 4.4. CONTRASTE DE CONTENIDO NO TEXTUAL

### CRITERIO DE CONFORMIDAD 1.4.11 (NIVEL AA)

En las WCAG 2.0 el requisito de asegurar un mínimo de contraste únicamente aplicaba a la información textual, bien incluida directamente como texto o bien como texto mostrado en imágenes.

Sin embargo, en las WCAG 2.1 se amplía el **requisito de contraste también al contenido no textual**. Se debe asegurar un ratio mínimo de **contraste de al menos 3:1** en los colores adyacentes de:

- **Componentes de la interfaz de usuario:** la información visual necesaria para identificar los componentes y los estados de la interfaz de usuario, excepto para los componentes inactivos o cuando la apariencia del componente la determina el agente de usuario y no la modifica el autor de la página.
- **Objetos gráficos:** todas las partes de los gráficos que sean necesarias para comprender el contenido, excepto cuando una presentación particular de gráficos es esencial para la información que se transmite.

El contraste se mide como la diferencia de luminosidad entre dos colores y el algoritmo para determinar dicho ratio está especificado en las WCAG. Para su verificación podemos hacer uso de herramientas que permiten seleccionar dos colores y calcular automáticamente su ratio de contraste. Algunas de estas herramientas, como ***Colour Contrast Analyser*** o ***WCAG Contrast Checker***, están explicadas en la "***Guía de Validación de Accesibilidad***" disponible en la sección de documentación del Portal de la Administración Electrónica.

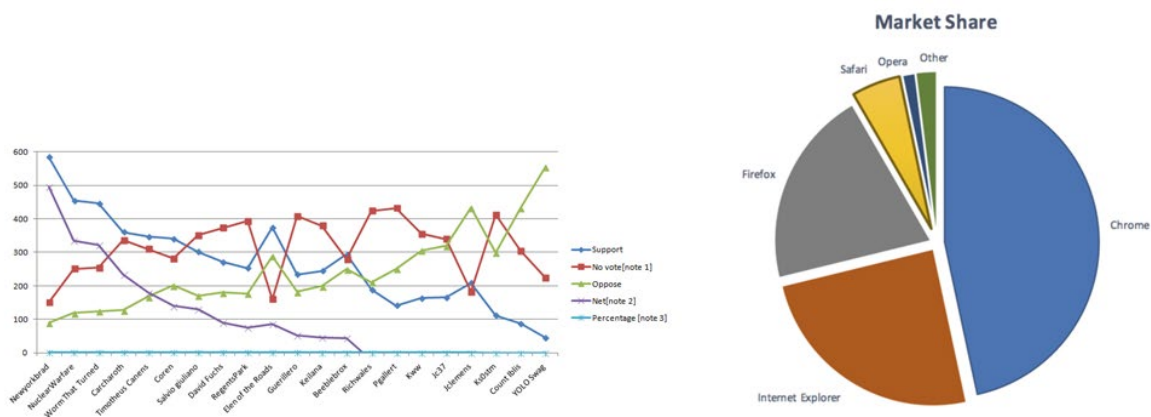
Entre los **componentes de la interfaz** de usuario se contemplan los botones y los controles de formulario. Cualquier **información visual** proporcionada que sea necesaria para que los usuarios identifiquen la existencia de los controles y cómo operar con ellos debe cumplir el mínimo de contraste respecto al color de fondo adyacente. Por ejemplo, el color del botón respecto al color de fondo o los bordes de un campo de edición respecto al color de fondo.

Asimismo, el **indicador del foco** de un componente debe tener un mínimo de contraste respecto al color de fondo adyacente cuando el componente recibe el foco. Los componentes de la interfaz que estén inactivos (no disponibles para la interacción con el usuario) están exentos de cumplir el mínimo de contraste.

El término **objetos gráficos** aplica a cualquier icono independiente como los iconos para imprimir (sin texto) así como a aquellas partes de imágenes o diagramas más complejos como las líneas en una gráfica de líneas, las diferentes barras en una gráfica de barras, los iconos o diferentes elementos que forman parte de una infografía, etc. Por ejemplo, en el caso de iconos sencillos de un



único color la imagen completa se considera un *objeto gráfico*. En el caso de imágenes formadas por múltiples líneas, colores y formas cada una de estas partes se considera un *objeto gráfico* que es necesario reconocer.



**Ilustración 2. En una gráfica de líneas cada línea debe tener contraste y en una gráfica de tarta se debe poder diferenciar cada parte de la gráfica (imágenes de ejemplo obtenidas de la documentación de las WCAG 2.1)**

Se consideran como **excepciones** los objetos gráficos cuya **presentación visual es esencial** para transmitir la información. Es decir, cuando no hay forma de presentar el recurso gráfico con el suficiente contraste sin alterar su significado. Es el caso, por ejemplo, de los **logotipos** o de las **banderas**; de las **imágenes de la vida real** como fotografías de personas o escenarios; o de las **imágenes que representan otras cosas** como capturas de pantallas para mostrar la apariencia de una aplicación o sitio web, diagramas médicos que usan colores estandarizados en biología o gradientes de color que representan medidas como mapas de calor.

Tampoco es aplicable a aquellos recursos gráficos o parte de los mismos que no necesitan percibirse para comprender el significado de la información. Es el caso, por ejemplo, de aquellos recursos gráficos usados con **finés puramente decorativos** o aquellos cuya **información también se está transmitiendo de forma alternativa** (p. ej. gráfica con una tabla de datos a continuación).

## 4.5. ESPACIADO DEL TEXTO

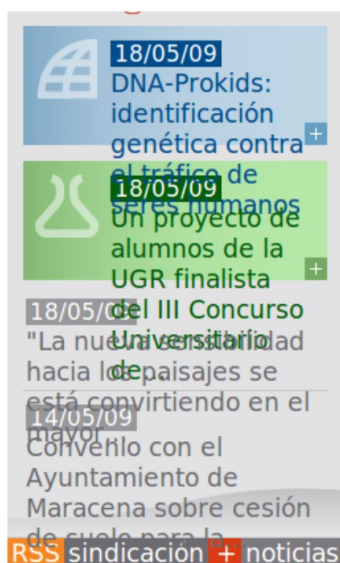
### CRITERIO DE CONFORMIDAD 1.4.12 (NIVEL AA)

Respecto a la presentación del texto, la maquetación del sitio o aplicación web ha de realizarse de forma que **no se pierda contenido o funcionalidad cuando los usuarios modifiquen algunas de las características del texto** como el espaciado entre líneas, párrafos, letras y palabras. Por ejemplo, algunos usuarios

con problemas visuales emplean hojas de estilo personalizadas para **adaptar la presentación del texto a sus necesidades**. Para facilitar el acceso al contenido a este tipo de usuarios, en las pautas de accesibilidad se exige que estos puedan:

- Ajustar el alto de la línea hasta al menos 1.5 veces el tamaño de la fuente
- Espaciar los párrafos hasta al menos el doble del tamaño de la fuente
- Ajustar el espacio entre letras hasta al menos 0.12 veces el tamaño de la fuente
- Ajustar el espaciado entre palabras hasta al menos 0.16 veces el tamaño de la fuente.

Este requisito no implica que el contenido debe tener dicha presentación inicialmente, sino que no se impida ajustarlo para obtener dicha presentación. Por ejemplo, que los usuarios puedan aplicar dichos estilos sin que se produzcan desbordamientos o solapamientos de contenidos que dificulten o impidan su visualización.



**Ilustración 3. Al ajustar las características del texto no se pueden producir solapamientos, cortes o desbordamientos del texto que provoquen pérdida de contenido o funcionalidad.**

## 4.6. CONTENIDO CON HOVER O FOCUS

### CRITERIO DE CONFORMIDAD 1.4.13 (NIVEL AA)

Si cuando un elemento recibe o pierde el puntero (p. ej. *hover* del ratón) o el foco del teclado (*focus*) se muestra u oculta algún contenido adicional, entonces se debe cumplir lo siguiente:

- Debe haber disponible un mecanismo que permita **descartar el contenido adicional** sin necesidad de mover el puntero o cambiar el foco del teclado a no ser que el contenido adicional informe acerca de un error en la entrada de datos o no tape o reemplace otro contenido.
- Si el puntero puede mostrar el contenido adicional, entonces el **puntero se puede mover sobre el contenido adicional** sin que éste desaparezca.
- **El contenido adicional debe permanecer visible** hasta que se retire el puntero o el foco del teclado o bien el usuario lo descarte (por el primer punto) o su información no siga siendo válida.

Este requisito se entiende mejor con ejemplos. Hace referencia a los contenidos que se muestran u ocultan con la interacción del usuario al situar el puntero sobre algunos elementos o cuando reciben el foco del teclado como es el caso de menús desplegados, pop-ups no modales, tooltips, etc. El objetivo es asegurar que los usuarios puedan tanto percibir el contenido mostrado como descartarlo si no lo requieren y sin que perjudique la experiencia de uso. Las razones para esto es porque ciertos usuarios, como aquellos que emplean magnificadores de pantalla y sólo perciben visualmente un fragmento de la página y no todo su contenido de forma simultánea se pueden encontrar con situaciones no deseadas como:

- Haber mostrado el contenido adicional de forma involuntaria. Por ejemplo, al pasar el puntero sobre un menú desplegable mientras lo desplazaba hacia otra zona de la página, sin querer mostrar el menú.
- No percatarse de que el contenido adicional se ha mostrado. Por ejemplo, si el contenido mostrado está fuera de su rango de visión y fuera del alcance del puntero de ratón.
- El contenido adicional interfiere con la habilidad del usuario para realizar una tarea, si se muestra tapando u ocultando otro contenido al que necesite acceder.

Se consideran como **excepciones** aquellos contenidos que se muestran u ocultan cuya presentación visual está controlada por la aplicación de usuario y no

por los desarrolladores como, por ejemplo, el tooltip que muestra el contenido de los atributos *title* (título).

## 4.7. ATAJOS CON TECLAS DE CARACTERES

### CRITERIO DE CONFORMIDAD 2.1.4 (NIVEL A)

En el caso de que para activar los elementos de interacción (enlaces, botones, etc.) se empleen **atajos de teclado usando una única letra**, signo de puntuación, número o símbolo entonces se debe cumplir al menos una de las siguientes condiciones:

- Existe un mecanismo que permite **desactivar** el atajo de teclado
- Existe un mecanismo que permite **reasignar** el atajo de teclado para emplear en su lugar otra tecla no imprimible (ej, *Ctrl*, *Alt*, etc.)
- El atajo de teclado sólo se puede activar cuando el componente tiene el foco del teclado.

El objetivo es evitar que las personas que interactúan mediante entradas de voz activen los elementos de interacción de forma accidental y no intencionada cuando pronuncien la letra correspondiente por otros motivos diferentes.

Esto no aplica a las teclas de acceso rápido (*accesskey*) sino a los atajos de teclado personalizados y programados de forma específica para emplear un único carácter para su activación. Las teclas de acceso rápido requieren de la pulsación de una tecla modificadora para su activación (*Ctrl*, *Shift*, *Alt*, etc.) de forma que no se activan accidentalmente al emplear entrada por voz.

## 4.8. GESTOS DEL PUNTERO

### CRITERIO DE CONFORMIDAD 2.5.1 (NIVEL A)

Toda la funcionalidad que emplee gestos multipunto o dependientes de la trayectoria realizada también se debe poder realizar empleando **un único punto de contacto y sin trazos**, mediante una pulsación sencilla con un dedo o con un puntero.

Este requisito no impide el uso de este tipo de interacción más compleja sino que en caso de emplearse se ha de proporcionar un método de interacción alternativo en el que sólo se requieran pulsaciones sencillas. Asimismo, aplica al contenido web que interpreta los gestos del puntero pero no es aplicable a las acciones

necesarias para operar el dispositivo, aplicación de usuario o producto de apoyo (están fuera de la responsabilidad del autor del sitio web).

Por ejemplo, si es posible hacer un gesto de pinza con dos dedos para hacer zoom sobre un mapa o un movimiento lineal para desplazarse sobre el mismo entonces también debe haber un par de botones que permitan hacer zoom y otros botones (p. ej. rueda de cuatro puntos) que permitan el desplazamiento. El objetivo de este criterio es asegurar que el contenido se puede operar desde un gran número de dispositivos de entrada sencillos de forma que las personas con problemas de movilidad los puedan realizar.



**Ilustración 4. Se han de proporcionar mecanismos de interacción alternativos que no impliquen gestos complejos como botones para manejar y desplazarse por un mapa.**

Ejemplos de **gestos complejos** son el de pinza con dos dedos (p. ej. para zoom), pulsación con un dedo mientras otro se mantiene pegado a la pantalla, pulsación múltiple con varios dedos o desplazamientos de uno o varios dedos. Por el contrario, ejemplos de **gestos sencillos** incluyen pulsaciones únicas, dobles pulsaciones o pulsaciones largas. En el caso del ratón, trackpad o dispositivo similar esto equivale a clicks sencillos, clicks dobles y clicks prolongados.

Este requisito tiene como **excepción** aquel contenido para el que los gestos de puntero multipunto o basados en trazos sean esenciales. Por ejemplo, cuando hay que incluir la firma en un dispositivo táctil o en una aplicación de dibujo a mano alzada.

## 4.9. CANCELACIÓN DEL PUNTERO

### CRITERIO DE CONFORMIDAD 2.5.2 (NIVEL A)

Toda funcionalidad que se pueda operar mediante un puntero sencillo (*single pointer*) debe cumplir al menos una de las siguientes condiciones:

- El **evento *down* del puntero no se emplea** para ejecutar ninguna parte de la funcionalidad.
- La función se completa con el evento *up* del puntero y existe un **mecanismo para cancelar la función** antes de que se complete **o para deshacerla** una vez completada.
- El **evento *up* del puntero deshace** cualquier consecuencia del evento *down* previo.
- Completar la función con el **evento *down* es esencial** (p. ej. se considera esencial un emulador de teclado)

Este requisito se refiere a los métodos de interacción con puntero, como puede ser el empleo del ratón o bien, en una pantalla táctil, la interacción con el dedo.

El objetivo es prevenir interacciones accidentales por parte de los usuarios al pulsar sobre algún componente sin querer. Por ejemplo, personas con problemas de movilidad (falta de precisión, temblores, etc.) pueden pulsar o tocar la pantalla sin querer con resultados no deseados. La mejor forma de evitar esto es lanzar las acciones con el evento *up* en vez de con el evento *down*.

El evento *up* sólo se dispara cuando se levanta el dedo o se libera el clic del ratón dentro de los límites del elemento de interacción. Así, si los usuarios se dan cuenta de que han pulsado un botón o enlace sin querer (al cambiar el indicador visual del foco, por ejemplo) aún tienen la posibilidad de deslizar y levantar el puntero fuera del área de interacción, cancelando así la acción.

## 4.10. ETIQUETA EN EL NOMBRE

### CRITERIO DE CONFORMIDAD 2.5.3 (NIVEL A)

En los componentes de la interfaz de usuario (como controles de formulario, botones, enlaces o componentes personalizados) **con etiquetas visibles que incluyen texto o imágenes de texto**, el nombre del componente (nombre programático o nombre accesible) debe contener el texto que se presenta visualmente. Esto es:

- **Nombre accesible = etiqueta visible + yyyyyyy**

Entendemos por **nombre accesible** o **nombre programático** el que emplean los agentes de usuario y las tecnologías de apoyo para identificar e informar a los usuarios acerca de los componentes, así como para interactuar con los mismos.

El objetivo de este criterio es asegurar que las **palabras que etiquetan visualmente** un componente **sean también** las **palabras asociadas programáticamente** con el componente. Esto ayuda a garantizar que las **personas con discapacidades puedan confiar en etiquetas visibles** como un medio **para interactuar** con los componentes. Así, una persona con una discapacidad motriz que pueda ver la página web en un navegador gráfico, pero interactúa mediante entrada por voz podrá leer el texto visible de la etiqueta para accionar o seleccionar el componente. Si la etiqueta visible y el nombre accesible no coinciden entonces la interacción resulta mucho más complicada para este tipo de personas y se abre la posibilidad además de accionar componentes accidentalmente.

El **nombre accesible** puede ser derivado de una propiedad visible (por ejemplo el texto de un botón) o no visible (por ejemplo el texto alternativo de una imagen) de un elemento de la interfaz de usuario. Es fundamental saber cómo se construye el nombre accesible; **cada tipo de elemento HTML tiene sus propias reglas para formar el nombre accesible ([HTML Accessibility API Mappings](#))**. Tanto `aria-label` como `aria-labelledby` tienen prioridad en el cálculo del nombre accesible, anulando el texto visible como nombre accesible incluso cuando la etiqueta de texto visible está asociada mediante programación con el control. Por esta razón, cuando ya existe una etiqueta visible, debe evitarse o usarse con cuidado `aria-label`, y debe usarse con cuidado como suplemento `aria-labelledby`. Por ejemplo, si utiliza el atributo `aria-label` será el nombre que va a leer un lector de pantalla cuando informe acerca del componente y el nombre que empleará un sistema de reconocimiento de voz para activar el componente, ya que el nombre del componente que era su texto visible ha sido reemplazado al utilizar el atributo `aria-label`.

El **nombre accesible debe contener sin modificación el texto visible**. Se considera como error el hecho de que el nombre accesible no contenga el texto visible de la etiqueta o que el nombre accesible contenga al texto visible pero en un orden diferente o con otras palabras intercaladas. Se recomienda no utilizar los atributos de WAI-ARIA relacionados con el nombre accesible cuando el componente de la interfaz tiene un texto visible que actúa como su etiqueta, ya que sería redundante e innecesario.

Si aplicamos todas las indicaciones de las pautas a la hora de proporcionar un etiquetado correcto para los campos de formulario u otros elementos de interacción ya estamos cumpliendo este requisito. Es decir, el atributo o mecanismo nativo de HTML para asignar un nombre a un componente es el nombre accesible que emplearán los lectores de pantalla o sistemas de reconocimiento de voz.

Por ejemplo, si en los formularios estamos empleando la asociación explícita entonces ya se está cumpliendo este requisito de coincidencia entre la etiqueta visible y el nombre accesible. El nombre accesible del campo será el texto de la etiqueta LABEL.

### Ejemplo de código 3

```
<label for="nombre">Nombre: <input type="text" id="nombre" >  
</label>
```

En el caso de los INPUT de tipo *submit* el nombre accesible por defecto es el valor de su atributo *value*.

### Ejemplo de código 4

```
<input type="submit" value="Buscar">
```

En el caso de los botones el nombre accesible es su contenido textual.

### Ejemplo de código 5

```
<button type="submit">Buscar</button>
```

En el caso de que estemos en otras situaciones, como cuando tenemos controles personalizados realizados mediante scripts, entonces podemos hacer uso de los atributos de **WAI-ARIA** *aria-label*, *aria-labelledby* o *aria-describedby* para asignar un nombre accesible a los componentes recordando que éste debe contener el texto de la etiqueta visible.

### Ejemplo de código 6

```
<div role="button" aria-label="Buscar Tramitaciones">Buscar</div>
```

## 4.11. ACTIVACIÓN MEDIANTE MOVIMIENTO

### CRITERIO DE CONFORMIDAD 2.5.4 (NIVEL A)

Cualquier **funcionalidad que se pueda operar mediante el movimiento** del dispositivo también se debe poder **realizar a través del interfaz de usuario** y también se debe poder **desactivar dicha operación** mediante el movimiento para evitar acciones no deseadas.

Esto beneficia a las personas que tienen sus dispositivos anclados en posiciones fijas (p. ej. una silla de ruedas) y no pueden interactuar mediante el movimiento.



También se evita que personas con problemas de movilidad ejecuten ciertas acciones accidentalmente al realizar movimientos no controlados.

Este requisito es aplicable a las acciones que se pueden activar mediante sensores que responden directamente a movimientos como la inclinación, el balanceo o sacudidas del dispositivo. Es el caso, por ejemplo, de activar acciones como "deshacer" mediante la sacudida del dispositivo, el avance o retroceso por páginas de un documento al inclinar el dispositivo hacia un lado u otro o el desplazamiento por una fotografía de 360º mediante el movimiento del dispositivo. Si existen este tipo de interacciones se han de proporcionar mecanismos alternativos (p. ej. botones) y se han de poder desactivar las acciones basadas en el movimiento.

Por el contrario, **no es aplicable al movimiento espacial de los usuarios** como el relacionado con la geolocalización u otros eventos controlados por el dispositivo que no tengan que ver con gestos intencionados del usuario.

## 4.12. MENSAJES DE ESTADO

### CRITERIO DE CONFORMIDAD 4.1.3 (NIVEL AA)

Los **mensajes de estado** deben poder ser **identificados por software a través de sus roles o propiedades** de forma que se puedan transmitir a los usuarios sin necesidad de recibir el foco. Es decir, que los lectores de pantalla y los tecnologías de apoyo similares puedan identificar y reconocer automáticamente estos mensajes de estado para que así se los puedan transmitir a los usuarios automáticamente sin que éstos tengan que desplazarse hasta su ubicación para leer su contenido. El objetivo es que los usuarios sean conscientes de cualquier cambio relevante en el contenido aunque no tenga el foco en ese momento.

Se entiende por un **mensaje de estado** aquellos mensajes que proporcionan información al usuario acerca del éxito o el resultado de una acción, del estado de espera de una aplicación, del progreso de un proceso o de la existencia de errores y dichos mensajes se proporcionan dinámicamente sin que se produzca un cambio de contexto (p. ej. se muestran en la misma página y no en una página nueva).

No toda la información que se añade en la página entra dentro de la definición de un mensaje de estado. Por ejemplo, en los resultados de una búsqueda el propio listado con los resultados no son un mensaje de estado. Sin embargo, un texto que informe acerca de la situación de la búsqueda ("*buscando...*") o acerca del número de resultados obtenidos ("*9 resultados obtenidos*", "*Sin resultados*", etc.) sí se considera un mensaje de estado si éste no recibe el foco.

Para hacer que los mensajes de estado sean reproducidos por los lectores de pantalla hay que emplear **roles de WAI-ARIA** para identificar la finalidad del componente. Algunos roles tienen la característica de que los lectores de pantalla leen su contenido automáticamente cuando éste cambia o cuando se muestra en la página. Así, dependiendo de la situación podemos emplear diferentes tipos de roles:

Si se trata de un mensaje de estado avisando del **éxito o del resultado de una acción**, con el atributo `role="status"` con un texto informando acerca del envío correcto o del resultado obtenido.

#### Ejemplo de código 7

```
<div role="status">5 resultados obtenidos.</div>
```

#### Ejemplo de código 8

```
<p role="status">  
    
  <span id="cart">0</span> elementos  
</p>
```

Si se trata de una sugerencia o **aviso acerca de la existencia de un error**, con el atributo `role="alert"` junto con alguna técnica de las empleadas para informar acerca de errores en la entrada de datos (identificar los campos obligatorios no completados, informar acerca de los errores en el formato o valor de los datos, dar sugerencias de corrección del error, etc.

#### Ejemplo de código 9

```
<p id="errors" role="alert">  
  El campo email es obligatorio.  
</p>
```

Si se trata de un mensaje de estado acerca del **progreso en un proceso**, con `role="progressbar"`, con `role="log"` (para identificar actualizaciones secuenciales de información) o con `role="status"` y un texto explicativo.

#### Ejemplo de código 10

```
<div role="progressbar" aria-valuenow="20" aria-valuemin="0" aria-  
valuemax="100">20 %</div>
```

### Ejemplo de código 11

```
<div id="chatRegion" role="log">  
  <h4>Chat</h4>  
  <ul id="conversation">  
    <li>Primer mensaje del chat</li>  
    <li>... </li>  
    <li>Último mensaje del chat</li>  
  </ul>  
</div>
```

### Ejemplo de código 12

```
<p role="status">Se encuentra en el paso 3 de 5. Por favor,  
incluya sus datos personales.</p>
```

*Nota:* cuando un contenido se marca con `role="log"` el lector de pantalla únicamente leerá las nuevas entradas que se añadan al final del mismo. Así, en los comentarios de un chat como el del ejemplo, el lector de pantalla leerá en voz alta el último mensaje añadido. El `role="log"` también es aplicable a un histórico de mensajes, un log de un juego o un log de errores.

**Importante:** para que funcione correctamente el `role` debe estar fijo en el código HTML, es decir debe existir el contenedor que tiene el `role` al cargar la página y después insertar dinámicamente el contenido o texto a mostrar.

## 5. ANEXO GUÍA RÁPIDA WCAG 2.1

A continuación se muestra una tabla con el resumen de los criterios de las WCAG 2.1 y las técnicas a aplicar.

**Tabla 2. Guía rápida de aplicación de las WCAG 2.1**

CRITERIO	TÉCNICAS
<b>1.3.4 ORIENTACIÓN (AA)</b> – El contenido se ve correctamente en orientación vertical y horizontal	<ul style="list-style-type: none"> <li>□ Aplica CSS para establecer la orientación.</li> <li>□ Utiliza controles para acceder al contenido en diferentes orientaciones.</li> </ul>
<b>1.3.5 IDENTIFICAR EL PROPÓSITO DE LA ENTRADA (AA)</b> – La asignación de tipos de los campos de información posibilita el autor relleno	<ul style="list-style-type: none"> <li>□ Aplicando los atributos de autocompletar HTML 5.2.</li> </ul>
<b>1.4.10 REAJUSTE DE TEXTO (AA)</b> – Aumentando el tamaño del texto en un 400% no existe pérdida de información ni scroll horizontal	<ul style="list-style-type: none"> <li>□ Uso de media queries y CSS grid para ajustar las columnas.</li> <li>□ Utilización de CSS Flexbox.</li> <li>□ Dar opciones en el contenido para el cambio de layout.</li> </ul>
<b>1.4.11 CONTRASTE CONTENIDO NO TEXTUAL (AA)</b> – Contraste mínimo de 3:1 en elementos como botones, campos de formulario, gráficos, iconos ...	<ul style="list-style-type: none"> <li>□ Contrastar gráficos, p.ej. incluyendo líneas de contraste entre colores contiguos.</li> <li>□ Contrastar el foco visible Contrastar texto sobre imágenes.</li> </ul>
<b>1.4.12 ESPACIADO DE TEXTO (AA)</b> – Al ajustar las características del texto no se pueden producir solapamientos, cortes o desbordamientos del texto que provoquen pérdida de contenido.	<ul style="list-style-type: none"> <li>□ Uso del espaciado de letras CSS para controlar el espaciado dentro de una palabra.</li> <li>□ Implementar el tamaño de los contenedores de texto utilizando unidades em.</li> </ul>
<b>1.4.13 CONTENIDO CON HOVER O FOCUS (AA)</b> – Al mostrar contenido mediante el puntero o el foco, los usuarios pueden percibirlo y descartarlo sin que perjudique la experiencia de uso.	<ul style="list-style-type: none"> <li>□ Utilizando Aria - role="tooltip"</li> <li>□ Uso de CSS: Pseudoclasas hover y focus</li> </ul>
<b>2.1.4 ATAJOS CON TECLAS DE CARACTERES (A)</b> – Si utilizas atajos de teclado usando una única letra... Cuidado!	<ul style="list-style-type: none"> <li>□ Los usuarios tienen una manera de desactivar o modificar los accesos directos de una sola tecla.</li> <li>□ Las accesskey no suponen un problema.</li> </ul>

CRITERIO	TÉCNICAS
<p><b>2.5.1 GESTOS DEL PUNTERO (A) –</b> Todo gesto multipunto o dependientes de trayectoria debe poder realizarse empleando un único punto de contacto y sin trazos.</p>	<ul style="list-style-type: none"> <li>□ Proporciona una alternativa sencilla a cualquier interacción compleja.</li> <li>□ Ejemplo. Zoom con dos dedos &gt; Botón para ampliar y otro para reducir.</li> </ul>
<p><b>2.5.2 CANCELACIÓN DE PUNTERO (A)</b> – Previene interacciones accidentales de los usuarios al pulsar sobre algún componente sin querer.</p>	<ul style="list-style-type: none"> <li>□ No utilizar el evento down del puntero para ejecutar ninguna parte de la funcionalidad.</li> <li>□ Utilizar el evento up para completar la acción.</li> </ul>
<p><b>2.5.2 ETIQUETA EN EL NOMBRE (A) -</b> La etiqueta de un componente del interfaz de usuario, que sirve para reconocerlos visualmente, también forma parte de su nombre accesible.</p>	<ul style="list-style-type: none"> <li>□ El nombre accesible debe coincidir con el texto visible o al menos comenzar con el texto visible.</li> <li>□ Al aplicar aria-label, aria-labelledby éste debe contener el texto de la etiqueta visible.</li> </ul>
<p><b>2.5.4 ACTIVACIÓN MEDIANTE MOVIMIENTO (A) -</b> Cualquier funcionalidad operable mediante el movimiento del dispositivo también se debe poder realizar a través del interfaz de usuario.</p>	<ul style="list-style-type: none"> <li>□ Al incorporar acciones como "deshacer" mediante la sacudidas del dispositivo, es obligatorio también un botón "Deshacer".</li> <li>□ No aplica a la geolocalización.</li> </ul>
<p><b>4.1.3 MENSAJES DE ESTADO (AA) -</b> Los mensajes de estado deben poder identificarse por software a través de sus roles o propiedades sin necesidad de recibir el foco.</p>	<ul style="list-style-type: none"> <li>□ Usar role=status para mostrar mensajes de estado.</li> <li>□ Usar role=alert para informar acerca de mensajes de error.</li> </ul>

A continuación se muestra una ilustración de la Guía rápida de aplicación de las WCAG 2.1 con el mismo contenido que en la tabla anterior:

# WCAG 2.1

## Guía rápida de aplicación

<p><b>1.3.4 ORIENTACIÓN</b> - El contenido se ve correctamente en orientación vertical y horizontal</p> <p><b>AA</b></p>	<ul style="list-style-type: none"> <li>● Aplica CSS para establecer la orientación</li> <li>● Utiliza controles para acceder al contenido en diferentes orientaciones</li> </ul> <p></p>
<p><b>1.3.5 IDENTIFICAR EL PROPÓSITO DE LA ENTRADA</b> - La asignación de tipos de la los campos de información posibilita el autorrellenado</p> <p><b>AA</b></p>	<ul style="list-style-type: none"> <li>● Aplicando los atributos de autocompletar HTML 5.2</li> </ul> <p></p>
<p><b>1.4.10 REAJUSTE</b> - Aumentando el tamaño del texto en un 400% no existe pérdida de información ni scroll horizontal</p> <p><b>AA</b></p>	<ul style="list-style-type: none"> <li>● Uso de media queries y CSS grid para ajustar las columnas</li> <li>● Utilización de CSS Flexbox</li> <li>● Dar opciones en el contenido para el cambio de layout</li> </ul> <p></p>
<p><b>1.4.11 CONTRASTE CONTENIDO NO TEXTUAL</b> - Contraste mínimo de 3:1 en elementos como botones, campos de formulario, gráficos, iconos ..</p> <p><b>AA</b></p>	<ul style="list-style-type: none"> <li>● Contrastar gráficos, p.ej. incluyendo líneas de contraste entre colores contiguos</li> <li>● Contrastar el foco visible</li> <li>● Contrastar texto sobre imágenes</li> </ul> <p></p>
<p><b>1.4.12 ESPACIADO DE TEXTO</b> - Al ajustar las características del texto no se pueden producir solapamientos, cortes o desbordamientos del texto que provoquen pérdida de contenido</p> <p><b>AA</b></p>	<ul style="list-style-type: none"> <li>● Uso del espaciado de letras CSS para controlar el espaciado dentro de una palabra</li> <li>● Implementar el tamaño de los contenedores de texto utilizando unidades em</li> </ul> <p></p>
<p><b>1.4.13 CONTENIDO CON HOVER O FOCUS</b> - Al mostrar contenido mediante el puntero o el foco, los usuarios pueden percibirlo y descartarlo sin que perjudique la experiencia de uso</p> <p><b>AA</b></p>	<ul style="list-style-type: none"> <li>● Utilizando Aria - role="tooltip"</li> <li>● Uso de CSS: Pseudoclasas hover y focus</li> </ul> <p></p>
<p><b>2.1.4 ATAJOS CON TECLAS DE CARACTERES</b> - Si utilizas atajos de teclado usando una única letra... Cuidado!</p> <p><b>A</b></p>	<ul style="list-style-type: none"> <li>● Los usuarios tienen una manera de desactivar o modificar los accesos directos de una sola tecla</li> <li>● Las accesskey no suponen un problema</li> </ul> <p></p>
<p><b>2.5.1 GESTOS DEL PUNTERO</b> - Todo gesto multipunto o dependientes de trayectoria debe poder realizarse empleando un único punto de contacto y sin trazos</p> <p><b>A</b></p>	<ul style="list-style-type: none"> <li>● Proporciona una alternativa sencilla a cualquier interacción compleja</li> <li>● Ejemplo. Zoom con dos dedos &gt; Botón para ampliar y otro para reducir</li> </ul> <p></p>
<p><b>2.5.2 CANCELACIÓN DE PUNTERO</b> - Previene interacciones accidentales de los usuarios al pulsar sobre algún componente sin querer</p> <p><b>A</b></p>	<ul style="list-style-type: none"> <li>● No utilizar el evento down del puntero para ejecutar ninguna parte de la funcionalidad</li> <li>● Utilizar el evento up para completar la acción</li> </ul> <p></p>
<p><b>2.5.3 ETIQUETA EN EL NOMBRE</b> - La etiqueta de un componente del interfaz de usuario, que sirve para reconocerlos visualmente, también forma parte de su nombre accesible</p> <p><b>A</b></p>	<ul style="list-style-type: none"> <li>● El nombre accesible debe coincidir con el texto visible o al menos comenzar con el texto visible</li> <li>● Al aplicar aria-label, aria-labelledby éste debe contener el texto de la etiqueta visible.</li> </ul> <p></p>
<p><b>2.5.4 ACTIVACIÓN MEDIANTE MOVIMIENTO</b> - Cualquier funcionalidad operable mediante el movimiento del dispositivo también se debe poder realizar a través del interfaz de usuario</p> <p><b>A</b></p>	<ul style="list-style-type: none"> <li>● Al incorporar acciones como "deshacer" mediante la sacudidas del dispositivo, es obligatorio también un boton "Deshacer"</li> <li>● No aplica a la geolocalización</li> </ul> <p></p>
<p><b>4.1.3 MENSAJES DE ESTADO</b> - Los mensajes de estado deben poder identificarse por software a través de sus roles o propiedades sin necesidad de recibir el foco</p> <p><b>AA</b></p>	<ul style="list-style-type: none"> <li>● Usar role=status para mostrar mensajes de estado</li> <li>● Usar role=alert para informar acerca de mensajes de error</li> </ul> <p></p>

**Ilustración 5. Guía rápida de aplicación de las WCAG 2.1**