

La única versión normativa de este documento es la versión original en inglés que se encuentra en el sitio web del W3C <http://www.w3.org/TR/2000/NOTE-WCAG10-HTML-TECHS-20001106/>.

Ninguna parte del presente documento en castellano es normativa aunque se especifique lo contrario.

Copyright © 1999-2000 W3C® (MIT, INRIA, Keio), Todos los Derechos Reservados. Son aplicables las reglas del W3C sobre obligaciones, marcas registradas, utilización de documentos y licencias de software.

Traductores: Carlos Egea, Alicia Sarabia, Alan Chuter

[\[indice\]](#) [\[indice.html\]](#)



Técnicas HTML para las Pautas de Accesibilidad al Contenido de la Web 1.0

Nota de la W3C de 6 de noviembre de 2000.

Versión original en inglés:

<http://www.w3.org/TR/2000/NOTE-WCAG10-HTML-TECHS-20001106/>
([texto plano](#), [PostScript](#), [PDF](#), [archivo gzip tar del HTML](#), [archivo zip del HTML](#))

Última versión (en inglés):

<http://www.w3.org/TR/WCAG10-HTML-TECHS/>

Versión anterior:

<http://www.w3.org/TR/2000/NOTE-WCAG10-HTML-TECHS-20000920/>

Editores:

Wendy Chisholm, [W3C](#);
Gregg Vanderheiden, [Trace R & D Center](#) Universidad de Wisconsin -- Madison;
Ian Jacobs, [W3C](#)

[Copyright](#) ©1999 - 2000 [W3C](#)® ([MIT](#), [INRIA](#), [Keio](#)), todos los derechos reservados. Se aplican las normas concernientes a [responsabilidad](#), [marca](#), [uso de documento](#) y [licencia de software](#) W3C.

Resumen

Este documento describe técnicas para la creación de contenidos accesibles en el Lenguaje de Marcación de Hipertexto (HTML véase HTML 4.01 [\[HTML4\]](#)). El propósito de este documento es servir de ayuda a los creadores de contenidos web que desean manifestar su conformidad con las "Pautas de Accesibilidad de los Contenidos Web 1.0" ([\[WCAG10\]](#)). Aunque las técnicas descritas en este documento ayudará a crear HTML conforme con las "Pautas de Accesibilidad a los Contenidos Web 1.0", estas técnicas no garantizan la conformidad y no son la única manera en que un desarrollador puede crear contenidos conformes a las Pautas.

Este documento forma parte de una serie de documentos sobre técnicas para la creación de contenidos Web accesibles. Para información sobre los otros documentos de la serie, véase "Techniques for Web Content Accessibility Guidelines 1.0" [\[WCAG10-TECHS\]](#).

Nota: Este documento contiene algunos ejemplos que ilustran soluciones accesibles en CSS pero también ejemplos desaconsejados que ilustran lo que los desarrolladores de contenidos no deben hacer. Los ejemplos desaconsejados están resaltados y los lectores los deben leer con precaución -- se incluyen únicamente a título de ilustración.

Estatus de este documento

Esta versión se publica para corregir algunos vínculos rotos en la versión anterior.

La versión de este documento del 6 de noviembre del 2000 es una Nota dentro de una serie de Notas redactadas y aprobadas por el [Grupo de Trabajo sobre la Accesibilidad de los Contenidos Web](#) (WCAG WG). Esta Nota no ha sido revisada ni aprobada por los miembros del W3C. La serie de documentos reemplaza el documento único [5 May 1999 W3C Note Techniques for Web Content Accessibility Guidelines 1.0](#) (Técnicas para las Pautas de Accesibilidad al Contenido en la Web 1.0, Nota W3C de 5 de mayo de 1999). Los temas tratados en el documento anterior se han separado en documentos específicos para cada tecnología que pueden evolucionar de forma independiente. Los documentos mas pequeños, específicos para cada tecnología permitirán a los autores concentrarse en una tecnología concreta.

Mientras que la Recomendación "Pautas de Accesibilidad a los Contenidos Web" [\[WCAG10\]](#) es un documento estable, se espera que esta serie de documentos de acompañamiento evolucione a medida que cambien las tecnologías y los desarrolladores de contenidos descubran técnicas más efectivas para diseñar contenidos web accesibles.

Están disponibles el [historial de los cambios a esta serie de documentos](#) y la [lista de temas abiertos y cerrados](#). Se anima a los lectores a formular sus comentarios sobre el documento y a proponer soluciones a los temas que trata. Por favor, los comentarios sobre este documento deben enviarse a w3c-wai-gl@w3.org; Existe también un [archivo histórico público](#).

La versión en lengua inglesa de esta especificación es la única versión normativa. Pueden estar disponibles [Traducciones de este documento](#).

La lista de errores conocidos de este documento está disponible en "[Errata in Web Content Accessibility Guidelines](#)." Por favor, los errores que detecte en este documento deben enviarse a wai-wcag-editor@w3.org.

La [Web Accessibility Initiative \(WAI\)](#) del World Wide Web Consortium (W3C) ofrece una variedad de recursos sobre accesibilidad de la Web. Las Pautas de Accesibilidad de la WAI se producen como parte de la [actividad técnica de la WAI](#). Los objetivos del Grupo de Trabajo sobre Pautas de Accesibilidad del Contenido Web se describen en [los estatutos](#).

Tabla de Contenidos

Resumen
Estatus de este documento
1 Estructura y metadatos del documento
1.1 Metadatos
1.2 Agrupación estructural
2 Información sobre el idioma
2.1 Señalar cambios de idioma
2.2 Señalar el idioma principal
3 Etiquetado de texto
3.1 Énfasis
3.2 Acrónimos y abreviaturas
3.3 Citas
3.4 Etiquetas y hojas de estilo mejor que imágenes: el ejemplo de las matemáticas
3.5 Otros ocho elementos estructurales (para identificar citas, fragmentos de código, texto eliminado, y otros)
4 Listas

4.1	Uso de hojas de estilo para cambiar las viñetas de una lista
5	Tablas
5.1	Tablas de datos
5.2	Tablas para maquetar
5.3	Alineación de las Tablas
5.4	Cuestiones de compatibilidad con versiones anteriores para tablas
6	Vínculos
6.1	Texto de los vínculos
6.2	Agrupar y saltar vínculos
6.3	Acceso desde el teclado
6.4	Vínculos y objetivos
7	Imágenes y mapas de imagen
7.1	Equivalentes textuales breves para imágenes ("texto alt")
7.2	Descripciones largas de imágenes
7.3	Dibujos con caracteres (ASCII art)
7.4	Mapas de imagen
7.5	Color en las imágenes
7.6	Imágenes animadas
8	Applets y otros objetos programados
8.1	Equivalentes textuales y no textuales para applets y otros objetos programados
8.2	Applets accesibles directamente
9	Audio y video
9.1	Información sonora
9.2	Equivalentes textuales para multimedia
9.3	Objetos multimedia incrustados
10	Marcos (frames)
10.1	Proporcionar un título para cada marco
10.2	Describir las relaciones entre marcos
10.3	Crear páginas para navegadores que no soportan FRAME
10.4	Contenidos fuente para marcos
10.5	Usar objetivos (target) para FRAME
10.6	Alternativas a los marcos
11	Formularios
11.1	Acceso a los formularios mediante el teclado
11.2	Agrupar los controles dentro del formulario
11.3	Etiquetas para los controles de formulario
11.4	Botones gráficos
11.5	Técnicas para controles de formulario específicos
11.6	Compatibilidad con versiones anteriores para los formularios
12	Scripts
12.1	Transformación airosa de los scripts
12.2	Scripts que causan destello
12.3	Scripts que causan movimiento y parpadeo
12.4	Scripts accesibles directamente
12.5	Presentación alternativa de scripts
12.6	Actualización de la página, ventanas nuevas
13	Índice de elementos y atributos HTML
	Elementos
	Atributos
14	Referencias
15	Recursos
15.1	Otras pautas
15.2	Las aplicaciones de usuario y otras herramientas
15.3	Recursos de accesibilidad
16	Agradecimientos

1 Estructura y metadatos del documento

Los desarrolladores de contenidos deberían usar marcación estructural y utilizarla según la especificación. Los elementos y atributos estructurales (consultar el [índice de elementos y atributos HTML](#) para identificarlos) fomentan la coherencia en los documentos y proveen información a otras herramientas (por ejemplo, herramientas de indexación, motores de búsqueda, programas que extraen datos de tablas a

bases de datos, herramientas de navegación que usan elementos de encabezamiento y programas de traducción automática que traducen el texto de un idioma a otro).

1.1 Metadatos

Puntos de verificación en esta sección:

- [13.2](#) Proporcione metadatos para añadir información semántica a las páginas y sitios. [Prioridad 2]

Algunos elementos estructurales proporcionan información sobre el propio documento. Esto es denominado "metadatos" sobre el documento (Metadato es información sobre datos). Los metadatos bien contruidos pueden proporcionar a los usuarios importante información orientativa. Los elementos HTML que proporcionan información útil sobre un documento incluyen:

1.1.1 TITLE: El título del documento.

Observe que el elemento (preceptivo) TITLE, que sólo aparece una vez en un documento, es diferente del atributo "title", que se aplica a casi todos los elementos HTML 4.01. Los desarrolladores de contenido deberían usar el atributo "title" de acuerdo con las especificaciones HTML 4.01. Por ejemplo, "title" debería ser usado con vínculos para proporcionar información sobre el objetivo del vínculo.

1.1.2 El elemento ADDRESS

Puede usarse para proporcionar información sobre el creador de la página.

1.1.3 El elemento META

Puntos de verificación en esta sección:

- [7.4](#) Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener las actualizaciones, no cree páginas que se actualicen automáticamente de forma periódica. [Prioridad 2]
- [7.5](#) Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener la redirección automático, no utilice marcadores para redirigir las páginas automáticamente. En su lugar, configure el servidor para que ejecute esta posibilidad. [Prioridad 2]

Este elemento puede especificar metadatos para un documento, que incluyen palabras clave, e información sobre el autor. Consulte la sección sobre [actualización automática de página](#) para obtener información sobre por qué META **no debe** usarse para redirigir a otras páginas o actualizar páginas automáticamente.

Los que siguen, son ejemplos **desaconsejados** en HTML. El primero cambia la página del usuario a intervalos regulares. Los desarrolladores de contenidos no deberían usar esta técnica para simular tecnología puntera de servidor. Los desarrolladores no pueden saber de antemano cuanto tiempo necesitará el usuario para leer una página. Una actualización antes de tiempo desorienta al usuario. Los desarrolladores de contenidos deberían evitar la actualización periódica y permitir a los usuarios elegir cuándo quieren actualizar la información.

Ejemplo desaconsejado.

```
<META http-equiv="refresh" content="60">
<BODY>
<P>...Información...
</BODY>
```

El siguiente ejemplo de HTML (que emplea el elemento META) dirige al usuario de una página a otra al acabar cierto tiempo. No obstante, las páginas **no deben** redirigir a los usuarios con estas etiquetas por que no es estándar, desorienta al usuario, y puede alterar el historial de páginas en el navegador.

Ejemplo desaconsejado.

```
<HEAD>
<TITLE>¡No utilice esto!</TITLE>
<META http-equiv="refresh" content="5;
      http://www.ejemplo.com/paginanueva">
</HEAD>
<BODY>
<P>Si su navegador soporta Refresh
(actualización automática), será redirigido a nuestro
<A href="http://www.ejemplo.com/paginanueva">sitio nuevo</A>
en 5 segundos; si no, active el vínculo manualmente.
</BODY>
```

1.1.4 La declaración !DOCTYPE

Puntos de verificación en esta sección:

- [3.2](#) Cree documentos que estén validados por las gramáticas formales publicadas. [Prioridad 2]

Validar contra una gramática formal y declarar esa validación al principio del documento permite al usuario saber que la estructura del documento es fiable. También permite al usuario saber dónde buscar la semántica si fuera necesario. [El Servicio de Validación W3C](#) valida un documento contra [toda una lista de gramáticas publicadas](#).

Es preferible validar contra las gramáticas del W3C. Véase las [Tecnologías Revisadas para la Accesibilidad](#).

1.1.5 El elemento LINK y herramientas de navegación

Puntos de verificación en esta sección:

- [13.9](#) Proporcione información sobre las colecciones de documentos (por ejemplo, los documentos que comprendan múltiples páginas). [Prioridad 3]

Los desarrolladores de contenido deberían usar el elemento [LINK](#) y los tipos de vínculos (consultar [\[HTML4\]](#), sección 6.12) para describir los mecanismos y organización de la navegación del documento. Algunas aplicaciones de usuario pueden sintetizar herramientas de navegación o permitir la impresión ordenada de un grupo de documentos basados en esta etiqueta.

Ejemplo.

Los siguientes elementos [LINK](#) podrían ser incluidos en el encabezamiento del capítulo 2 de un libro:

```
<LINK rel="Next" href="capítulo3">
<LINK rel="Prev" href="capítulo1">
<LINK rel="Start" href="portada">
<LINK rel="Glossary" href="glosario">
```

1.1.6 El elemento LINK y documentos alternativos

Puntos de verificación en esta sección:

- [6.5](#) Asegúrese de que los contenidos dinámicos son accesibles o proporcione una página o presentación alternativa [Prioridad 2]

También se puede emplear el elemento LINK para apuntar a documentos alternativos. Los navegadores deberían cargar la página alternativa de forma automática según las preferencias y el tipo de navegador del usuario. Por ejemplo, se puede emplear el elemento LINK como sigue:

Ejemplo.

Las aplicaciones de usuario que soportan LINK cargarán la página alternativa para aquellos usuarios cuyos navegadores se identifican como uno de los que pueden mostrar páginas "aural", "braille", o "tty".

```
<HEAD>
<TITLE>Bienvenidos al Centro Comercial Virtual!</TITLE>
<LINK title="Versión solo texto"
      rel="alternate"
      href="solo_texto"
      media="aural, braille, tty">
</HEAD>
<BODY><P>...</BODY>
```

1.2 Agrupación estructural

Puntos de verificación en esta sección:

- [12.3](#) Divida los bloques largos de información en grupos más manejables cuando sea natural y apropiado. [Prioridad 2]

Los siguientes grupos de mecanismos HTML 4.01 agrupan el contenido y facilitan su comprensión:

- Utilice [FIELDSET](#) para [agrupar controles de formulario en unidades semánticas](#) y describa el grupo con el elemento [LEGEND](#).
- Utilice [OPTGROUP](#) para [organizar las listas largas de opciones de menú en grupos más pequeños](#).
- [Utilice tablas para datos tabulares](#) y describa la tabla con [CAPTION](#).
- [Agrupe las filas y columnas de las tablas](#) con [THEAD](#), [TBODY](#), [TFOOT](#) y [COLGROUP](#).
- [Anide listas](#) con [UL](#), [OL](#) y [DL](#).
- Use los encabezamientos de sección ([H1](#) - [H6](#)) para crear documentos estructurados y separar tramos largos de texto. Consulte la sección siguiente para más información.
- Separe las líneas de texto en párrafos (con el elemento [P](#)).
- Agrupe vínculos relacionados entre sí. Consulte la sección [Agrupar y saltar vínculos](#).

Todos estos mecanismos de agrupación deberían ser usados cuando sean apropiados y naturales, por ejemplo, cuando la información se dé en grupos lógicos. Los desarrolladores de contenido no deberían crear grupos de forma aleatoria, puesto que pueden confundir a los usuarios.

1.2.1 Encabezamientos de sección

Puntos de verificación en esta sección:

- [3.5](#) Utilice elementos de encabezado para transmitir la estructura lógica y utilícelos de acuerdo con la especificación. [Prioridad 2]

Los documentos largos se suelen dividir en una serie de capítulos, los capítulos tienen apartados, los apartados se dividen en distintas secciones, las secciones en párrafos, etc. Estos trozos semánticos de información constituyen la estructura del documento.

Las secciones deberían iniciarse con los elementos de encabezamiento HTML ([H1](#)-[H6](#)). Otras etiquetas pueden complementar estos elementos para mejorar la presentación (por ejemplo, el elemento [HR](#) para crear una línea divisoria horizontal), pero la presentación visual no es suficiente para identificar las secciones de un documento.

Puesto que algunos usuarios ojean un documento navegando sus encabezamientos, es importante usarlos apropiadamente para expresar correctamente la estructura del documento. Los desarrolladores deberían ordenar los elementos de encabezamiento de forma apropiada. Por ejemplo, en HTML, los elementos [H2](#) deberían seguir a los elementos [H1](#), los [H3](#) deberían seguir a los elementos [H2](#), etc. Los desarrolladores de contenido no deberían "saltarse" niveles (por ejemplo, pasar directamente de [H1](#) a [H3](#)). No utilice encabezamientos para crear efectos de fuentes. Use [hojas de estilo para cambiar los estilos de fuentes](#),

por ejemplo.

Observe que en HTML, los elementos de encabezamiento (H1 - H6) sólo inician secciones y no las contienen como contenido del elemento. El siguiente etiquetado HTML muestra cómo pueden usarse las hojas de estilo para controlar la presentación de un encabezamiento y el contenido que sigue:

Ejemplo.

```
<HEAD>
<TITLE>Técnicas de cocina</TITLE>
<STYLE type="text/css">
  /* Sangra el encabezamiento y el contenido
    siguiente */
  DIV.sección2 { margin-left: 5% }
</STYLE>
</HEAD>
<BODY>
<H1>Técnicas de cocina</H1>
... algún texto aquí ...
<DIV class="sección2">
<H2>Cocinar con aceite</H2>
... texto para esta sección ...
</DIV>

<DIV class="sección2">
<H2>Cocinar con mantequilla</H2>
... texto para esta sección ...
</DIV>
```

[Consulte también la sección sobre vínculos.](#)

2 Información sobre el idioma

2.1 Señalar los cambios de idioma

Puntos de verificación en esta sección:

- [4.1](#) Identifique claramente los cambios en el idioma del texto del documento y en cualquier texto equivalente (por ejemplo, los subtítulos). [Prioridad 1]

Si utiliza varios idiomas en una página, asegure que cualquier cambio de idioma esté claramente identificado, mediante el uso del atributo `lang`:

Ejemplo.

```
<P>y con un cierto<SPAN lang="fr">je ne sais quoi</SPAN>,
ella entró tanto en la habitación como en su vida para siempre.
<Q>Mi nombre es Natasha</Q>dijo ella.
<Q lang="it">Piacere,</Q> respondió él en impecable
italiano, cerrando la puerta.
```

Es importante identificar los cambios de idioma por varias razones:

1. Los usuarios que leen el documento en braille podrán insertar los códigos de control adecuados (etiquetas) donde ocurren los cambios de idioma, para asegurar que el programa de traducción a braille generará los caracteres correctos (por ejemplo, caracteres acentuados). Estos códigos de control previenen también que se generen contracciones braille, que confundirían aun más al usuario. Las contracciones braille combinan grupos de caracteres comúnmente utilizados, que usualmente aparecen en celdas múltiples, en una sola celda. Por ejemplo, "ección", que habitualmente ocupa seis celdas (una para cada letra) puede contraerse en dos..
2. De forma similar, los sintetizadores de voz que "hablan" varios idiomas, serán capaces de generar el texto con el acento y la pronunciación adecuados. Si los cambios no están señalados, el sintetizador

- tratará de pronunciarlos en el idioma original del programa. Así, la palabra francesa para coche, "voiture", será pronunciada "voiture".
3. Los usuarios que no saben idiomas, obtendrán la traducción de los textos de idiomas desconocidos mediante programas de traducción.

2.2 Señalar el idioma principal

Puntos de verificación en esta sección:

- [4.3](#) Identifique el idioma principal de un documento. [Prioridad 3]

Es, así mismo, una buena práctica identificar el idioma principal de un documento, bien con una etiqueta (como se muestra abajo) o bien a través de encabezamientos HTTP.

Ejemplo.

```
<HTML lang="fr">
...resto de un documento HTML escrito en francés...
</HTML>
```

3 Etiquetado de texto

Las secciones siguientes tratan los modos de agregar estructura a trozos de texto.

3.1 Énfasis

Puntos de verificación en esta sección:

- [3.3](#) Utilice hojas de estilo para controlar la maquetación y la presentación. [Prioridad 2]

Para remarcar el énfasis, utilice los elementos HTML apropiados: **EM** y **STRONG**. No deberían usarse los elementos **B** e **I**, ya que se usan para crear un efecto visual de presentación. Los elementos EM y STRONG fueron diseñados para indicar un énfasis estructural, que puede ser plasmado en varios modos (cambios de estilo de fuente, cambios de inflexión del discurso, etc.).

3.2 Acrónimos y abreviaturas

Puntos de verificación en esta sección:

- [4.2](#) Especifique la expansión de cada abreviatura o acrónimo cuando aparezcan por primera vez en un documento. [Prioridad 3]

Marque las abreviaturas y acrónimos con **ABBR** y **ACRONYM** y utilice "title" para indicar la expansión:

Ejemplo.

```
<P>¡Bienvenido a la <ACRONYM title="World Wide Web">WWW</ACRONYM>!
```

Esto también es aplicable a las frases recortadas que se emplean para los encabezados de filas o columnas de tablas. Si el texto de encabezado ya está abreviado, proporcione una expansión con ABBR. Si el texto es largo, puede proporcionar una abreviatura, tal y como se describe en [Tablas de datos](#).

Ejemplo.

```
...
<TH>Nombre</TH>
<TH><ABBR title="Número del Documento Nacional de Identidad">Nº DNI</ABBR>
```


3.3 Citas

Puntos de verificación en esta sección:

- [3.7](#) Marque las citas. No utilice el marcador de citas para efectos de formato tales como sangrías. [Prioridad 2]

Los elementos [Q](#) y [BLOCKQUOTE](#) marcan citas en línea y en bloques, respectivamente.

Ejemplo.

Este ejemplo marca una cita larga con [BLOCKQUOTE](#):

```
<BLOCKQUOTE cite="http://www.ejemplo.com/trabajosamorperdido">
  <P>¡Recompensa! ¡Oh! Esa es la palabra latina para tres peniques.
    --- William Shakespeare (Trabajos de amor perdido).
  </P>
</BLOCKQUOTE>
```

3.4 Etiquetas y hojas de estilo mejor que imágenes: el ejemplo de las matemáticas

Puntos de verificación en esta sección:

- [3.1](#) Cuando exista un marcador apropiado, use marcadores en vez de imágenes para transmitir la información. [Prioridad 2]

El uso de etiquetas (y hojas de estilo) donde sea posible, mejor que imágenes (por ejemplo, en una ecuación matemática), promueve la accesibilidad por las siguientes razones:

- El texto puede ser ampliado o interpretado como habla o braille.
- Los mecanismos de búsqueda pueden usar la información del texto.

Como ejemplo, considere estas técnicas para presentar matemáticas en la Web:

- Asegúrese de que el usuario sabe lo que representan las variables, por ejemplo, en la ecuación " $F = m \cdot a$ ", indique que F = fuerza, m = masa y a = aceleración.
- Para ecuaciones simples utilice caracteres, como en " $x + y = z$ ".
- Para ecuaciones más complejas, márquelas con MathML ([\[MATHML\]](#)) o TeX. **Nota:** MathML puede ser usado para crear documentos muy accesibles, pero en la actualidad no es tan ampliamente soportado y usado como TeX.
- Proporcione una descripción de texto de la ecuación y, donde sea posible, use entidades HTML para crear símbolos matemáticos. Debe proporcionarse un texto equivalente si la ecuación está representada por una o más imágenes.

TeX se usa habitualmente para crear documentación técnica que después se convierte a HTML para su publicación en la Web. Puesto que los convertidores tienden a generar imágenes, utilizan etiquetas desaconsejadas y utilizan tablas para maquetar, los responsables del contenido deberían en consecuencia:

1. Hacer el documento original TeX (o LaTeX) disponible en la Web. Existe un sistema denominado "AsTeR" ([\[ASTER\]](#)) que puede crear una representación auditiva de documentos TeX y LaTeX. También IBM proporciona un plug-in para Netscape e Internet Explorer que lee documentos TeX/LaTeX y parte de MathML (consulte [\[HYPERMEDIA\]](#)). **Nota.** Estas herramientas funcionan con el idioma inglés, y puede que no funcionen adecuadamente con sintetizadores de habla cuyo idioma primario no sea inglés.
2. Asegúrese de que el HTML creado en el proceso de conversión es accesible. Proporcione una sola descripción de la ecuación (mejor que texto "alt" de cada imagen generada, puesto que puede haber pequeñas imágenes para partes y trozos de la ecuación).

3.5 Otros ocho elementos estructurales (para identificar citas, fragmentos de código, texto eliminado, y otros)

La especificación HTML 4.01 define los siguientes elementos estructurales para diversas necesidades de etiquetado:

CITE

Contiene una cita o referencia a otras fuentes.

DFN

Indica que es la definición del término que contiene.

CODE

Designa un fragmento de código informático.

SAMP

Designa una muestra de datos de salida de un programa, script, etc.

KBD

Indica el texto que debe introducir el usuario.

VAR

Indica un ejemplo de una variable o argumento de un programa.

INS

Indica un texto insertado en un documento.

DEL

Indica un texto suprimido de un documento.

4 Listas

Puntos de verificación en esta sección:

- [3.6](#) Marque correctamente las listas y los ítems de las listas. [Prioridad 2]

Los elementos de listado HTML [DL](#), [UL](#) y [OL](#) deberían ser usados únicamente para crear listas, no para formatear efectos tales como sangría. Consulte la información sobre [CSS y tablas para disposición de página](#) en el documento sobre Técnicas CSS [\[WCAG10-CSS-TECHNIQUES\]](#).

Las listas ordenadas ayudan a navegar a los usuarios no videntes. Los usuarios no visuales pueden "sentirse perdidos" en las listas, especialmente en las anidadas y aquellas que no especifican el nivel de anidamiento para cada ítem de la lista. Hasta que las aplicaciones de usuario proporcionen un medio para identificar claramente el contexto dentro de una lista (por ejemplo, soportando el pseudo-elemento ":before" en CSS2), los desarrolladores de contenido deberían incluir pistas contextuales en sus listas.

Para listas numeradas, los números compuestos son más informativos que los simples. Así, una lista numerada "1, 1.1, 1.2, 1.2.1, 1.3, 2, 2.1, ..." proporciona más contexto que la misma lista sin números compuestos, la cual se formatearía así:

1.
 - 1.
 2.
 - 1.
 - 3.
2.
 - 1.

y sería narrada como "1, 1, 2, 1, 3, 2, 1", sin aportar información sobre la profundidad dentro de la lista.

[\[CSS1\]](#) y [\[CSS2\]](#) permiten a los usuarios controlar los estilos de números (para todos los tipos de lista, no sólo las ordenadas) mediante las hojas de estilo del usuario.

Ejemplo.

La siguiente hoja de estilo CSS2 muestra cómo especificar números compuestos para listas anidadas creadas tanto con elementos UL como OL. Los ítems están numerados como "1", "1.1", "1.1.1", etc.

```
<STYLE type="text/css">
  UL, OL { counter-reset: item }
  LI { display: block }
  LI:before { content: counters(item, "."); counter-increment: item }
</STYLE>
```

Hasta que CSS2 sea ampliamente soportado o las aplicaciones de usuario permitan al usuario controlar la interpretación de las listas a través de otros medios, los autores deberían considerar el proporcionar pistas contextuales en las listas anidadas no numeradas. Los usuarios no videntes pueden tener dificultades para saber dónde está el comienzo y el fin de una lista y dónde comienza cada ítem de la misma. Por ejemplo, si un apartado de la lista envuelve a la siguiente línea en la pantalla, puede parecer ser dos ítems separados en la lista. Esto puede suponer un problema para los lectores de pantalla de anteriores generaciones.

4.1 Uso de hojas de estilo para cambiar las viñetas de una lista

Para cambiar el estilo de viñeta de una lista de ítem sin ordenar creada con el elemento `LI`, utilice hojas de estilo. En CSS, es posible especificar un estilo de viñeta por defecto (por ejemplo, 'disc') si no se encuentra la imagen especificada para la viñeta.

Ejemplo.

```
<HEAD>
<TITLE>Uso de hojas de estilo para cambiar viñetas</TITLE>
<STYLE type="text/css">
  UL { list-style: url(estrella.gif) disc }
</STYLE>
</HEAD>
<BODY>
<UL>
  <LI>Audrey
  <LI>Laura
  <LI>Alicia
</UL>
```

Para asegurar más aún que los usuarios comprenden las diferencias entre los ítems de la lista indicados visualmente, los desarrolladores de contenido deberían proporcionar una etiqueta de texto antes o después de la frase del ítem de la lista:

Ejemplo.

En este ejemplo, la nueva información es comunicada a través del texto ("Nuevo"), estilo de fuente (negrita) y el color (viñeta amarilla, texto rojo sobre fondo amarillo).

```
<HEAD>
<TITLE>Ejemplo de estilo de viñetas</TITLE>
<STYLE type="text/css">
  .txtnuevo { font-weight: bold;
              color: red;
              background-color: yellow }
  .bolonuevo { list-style : url(amarillo.gif) disc }
</STYLE>
</HEAD>
<BODY>
<UL>
  <LI class="bolonuevo">Fondo de pensiones Roth <SPAN class="txtnuevo">Nuevo</SPAN></LI>
  <LI> 401(k)</LI>
</UL>
</BODY>
```

4.1.1 Imágenes utilizadas como viñetas

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]

Evite el uso de imágenes como viñetas de listados de definiciones creadas con [DL](#), [DT](#), y [DD](#). No obstante, si usa este método, asegúrese de proporcionar un texto equivalente para las imágenes.

Ejemplo desaconsejado.

```
<HEAD>
<TITLE>Ejemplo desaconsejado que usa imágenes en listas DL</TITLE>
</HEAD>
<BODY>
<DL>
  <DD><IMG src="estrella.gif" alt="*" ">Audrey
  <DD><IMG src="estrella.gif" alt="*" ">Laura
  <DD><IMG src="estrella.gif" alt="*" ">Alicia
</DL>
```

Los desarrolladores de contenido deberían evitar los estilos de lista donde las viñetas proporcionen información (visual) adicional. No obstante, si se hace, asegúrese de proporcionar un [texto equivalente](#) que describa el significado de la viñeta:

Ejemplo desaconsejado.

```
<DL>
<DD><IMG src="rojo.gif" alt="Nuevo:">Fondo de pensiones Roth</DD>
<DD><IMG src="amarillo.gif" alt="Antiguo:">Fondo 401(k)</DD>
</DL>
```

5 Tablas

Esta sección trata la accesibilidad de las tablas y elementos que se pueden poner en un elemento [TABLE](#). Se tratan dos tipos de tablas: tablas usadas para organizar datos y tablas usadas para crear una disposición visual de la página.

5.1 Tablas de datos

Los desarrolladores de contenidos pueden hacer las tablas de datos HTML 4.01 más accesibles de varias maneras:

- Proporcionar información de resumen
- Identificar la información en filas y columnas

5.1.1 Proporcionar información de resumen

Puntos de verificación en esta sección:

- [5.5](#) Proporcione resúmenes de las tablas. [Prioridad 3]
- [5.6](#) Proporcione abreviaturas para las etiquetas de encabezamiento. [Prioridad 3]
- Proporcione un título mediante el elemento [CAPTION](#). El título de una tabla describe la naturaleza de la tabla en una a tres oraciones. Dos ejemplos:
 1. "Tazas de café consumidas por cada senador."
 2. "¿Quién invierte más en la limpieza de la contaminación?"

No es siempre necesario un título.

- Si no se proporciona un título mediante CAPTION, utilice el atributo "title" del elemento TABLE para describir la naturaleza de la tabla en pocas palabras.
- Proporcione un resumen a través del atributo "summary". Los resúmenes son especialmente útiles para los lectores no videntes. Un resumen de las relaciones entre las celdas es especialmente importante para las tablas con encabezamientos anidados, celdas que ocupan varias columnas o filas, u otras relaciones que no sean evidentes en un análisis de la estructura de la tabla, pero que sí se aprecian cuando la tabla se muestra visualmente. Un resumen también puede describir como una tabla encaja en el contexto del documento actual. Si no se proporciona ningún título es mas importante aún proporcionar un resumen. Dos ejemplos:
 1. "Esta tabla esquematiza el número de tazas de café consumidas por cada senador, el tipo de café (descafeinado o normal) y si se ha tomado con azúcar."
 2. "Total exigido por las normas de control de la contaminación a 1 de enero 1971. La categoría Comercial incluye almacenes, compañías de seguros y bancos. La tabla se divide en dos columnas. La columna izquierda es 'Inversión total necesaria en billones de dólares'. La columna derecha es 'Inversión' y se divide en tres subcolumnas. La primera columna se titula '1970 real en millones de dólares', la segunda es '1971 previsto en millones de dólares', y la tercera es 'Variación porcentaje, 1970 sobre 1971.' Las filas representan las industrias." [NBA, 1996].
- Proporcione subtítulos breves para las etiquetas de encabezamiento con el atributo "abbr" de TH. Estas serán particularmente útiles para las futuras tecnologías con habla que puedan leer las etiquetas de fila y columna para cada celda. Las abreviaturas reducen la repetición y el tiempo de lectura.

5.1.2 Identificar información de filas y columnas

Puntos de verificación en esta sección:

- [5.1](#) En las tablas de datos, identifique los encabezamientos de fila y columna. [Prioridad 1]
- [5.2](#) Para las tablas de datos que tienen dos o más niveles lógicos de encabezamientos de fila o columna, utilice marcadores para asociar las celdas de encabezamiento y las celdas de datos. [Prioridad 1]
- Identifique grupos estructurales de filas ([THEAD](#) para encabezamientos de tabla que se repiten, [TFOOT](#) para pies de tabla que se repiten y [TBODY](#) para otros grupos de fila) y grupos de columnas ([COLGROUP](#) y [COL](#)).
- Etiquete los elementos de tabla con los atributos "scope", "headers", y "axis" de forma que los futuros navegadores y ayudas técnicas sean capaces de seleccionar datos de una tabla filtrando por categorías.
- No utilice [PRE](#) para crear una disposición tabular de texto (utilice el elemento TABLE de forma que las ayudas técnicas puedan reconocer que es una tabla).
- Para información sobre los encabezamientos de tabla, consulte el algoritmo encabezamiento de tablas y el debate sobre la Recomendación HTML 4.01 ([HTML4](#)), sección 11.4.3).

Ejemplo.

Este ejemplo muestra cómo asociar celdas de datos (creadas con ID) con sus correspondientes encabezamientos a través del atributo "headers". El atributo "headers" especifica una lista de celdas de encabezamiento (etiquetas de fila y columna) asociadas con la celda de datos actual. Esto requiere que cada encabezamiento de celda tenga un atributo "id" que lo identifique.

```
<TABLE border="1"
  summary="Esta tabla esquematiza el número de
           tazas de café consumidas por cada senador,
           el tipo de café (descafeinado o normal)
           y si se ha tomado con azúcar.">
  <CAPTION>Tazas de café consumidas por cada senador</CAPTION>
  <TR>
    <TH id="header1">Nombre</TH>
    <TH id="header2">Tazas</TH>
    <TH id="header3" abbr="Tipo">Tipo de café</TH>
    <TH id="header4">¿Azúcar?</TH>
```

```

<TR>
  <TD headers="header1">T. Sexton</TD>
  <TD headers="header2">10</TD>
  <TD headers="header3">Expreso</TD>
  <TD headers="header4">No</TD>
<TR>
  <TD headers="header1">J. Dinnen</TD>
  <TD headers="header2">5</TD>
  <TD headers="header3">Descaf</TD>
  <TD headers="header4">Si</TD>
</TABLE>

```

Un sintetizador de voz podría leer esta tabla como sigue:

Título: Tazas de café consumidas por cada senador
 Resumen: Esta tabla esquematiza el número de tazas de café consumidas por cada senador, el tipo de café (descafeinado o normal) y si se ha tomado con azúcar.
 Nombre: T. Sexton, Tazas: 10, Tipo: Expreso, Azúcar: No
 Nombre: J. Dinnen, Tazas: 5, Tipo: Descaf, Azúcar: Si

Una aplicación de usuario visual podría mostrar la tabla como sigue:

TAZAS DE CAFÉ CONSUMIDAS POR CADA SENADOR

NOMBRE	TAZAS	TIPO DE CAFÉ	AZÚCAR?
T. Sexton	10	Expreso	No
J. Dinnen	5	Descafeinado	Sí

[\[Descripción de la tabla de café\]](#)

El siguiente ejemplo asocia las mismas celdas de encabezamiento (**TH**) y datos (**TD**) como antes, pero esta vez utiliza el atributo "**scope**" en lugar de "**headers**". "Scope" debe tener uno de los siguientes valores: "row", "col", "rowgroup" o "colgroup". "Scope" especifica la batería de celdas de datos que han de asociarse con la celda de encabezamiento correspondiente. Este método es particularmente útil para tablas simples. Debería tenerse en cuenta que la versión hablada de esta tabla podría ser idéntica a la del ejemplo anterior. La elección entre los atributos "headers" y "scope" depende de la complejidad de la tabla. No afecta al resultado en la medida en que en la etiqueta haya quedado clara la relación entre el encabezamiento y las celdas de datos.

Ejemplo.

```

<TABLE border="1"
  summary="Esta tabla esquematiza ..."
  <CAPTION>Tazas de café consumidas por cada senador</CAPTION>
  <TR>
    <TH scope="col">Nombre</TH>
    <TH scope="col">Tazas</TH>
    <TH scope="col" abbr="Tipo">Tipo de Café</TH>
    <TH scope="col">¿Azúcar?</TH>
  <TR>
    <TD>T. Sexton</TD> <TD>10</TD>
    <TD>Expreso</TD> <TD>No</TD>
  <TR>
    <TD>J. Dinnen</TD> <TD>5</TD>
    <TD>Descaf</TD> <TD>si</TD>
  </TABLE>

```

El siguiente ejemplo muestra cómo crear categorías en una tabla usando el atributo "**axis**" (eje).

Ejemplo.

```

<TABLE border="1">
  <CAPTION>Liquidación de gastos de viaje</CAPTION>
  <TR>
    <TH></TH>
    <TH id="header2" axis="gastos">Comidas
    <TH id="header3" axis="gastos">Hotel
    <TH id="header4" axis="gastos">Transportes
    <TD>subtotales</TD>
  <TR>
    <TH id="header6" axis="lugar">San Jose
    <TH> <TH> <TH> <TD>
  <TR>
    <TD id="header7" axis="fecha">25-ago-97
    <TD headers="header6 header7 header2">37,74
    <TD headers="header6 header7 header3">112,00
    <TD headers="header6 header7 header4">45,00
    <TD>
  <TR>
    <TD id="header8" axis="fecha">26-ago-97
    <TD headers="header6 header8 header2">27,28
    <TD headers="header6 header8 header3">112,00
    <TD headers="header6 header8 header4">45,00
    <TD>
  <TR>
    <TD>subtotales
    <TD>65,02
    <TD>224,00
    <TD>90,00
    <TD>379,02
  <TR>
    <TH id="header10" axis="lugar">Seattle
    <TH> <TH> <TH> <TD>
  <TR>
    <TD id="header11" axis="fecha">27-ago-97
    <TD headers="header10 header11 header2">96,25
    <TD headers="header10 header11 header3">109,00
    <TD headers="header10 header11 header4">36,00
    <TD>
  <TR>
    <TD id="header12" axis="date">28-ago-97
    <TD headers="header10 header12 header2">35,00
    <TD headers="header10 header12 header3">109,00
    <TD headers="header10 header12 header4">36,00
    <TD>
  <TR>
    <TD>subtotales
    <TD>131,25
    <TD>218,00
    <TD>72,00
    <TD>421,25
  <TR>
    <TH>Totales
    <TD>196,27
    <TD>442,00
    <TD>162,00
    <TD>800,27
</TABLE>

```

Esta tabla lista los gastos de viaje en dos lugares, San José y Seattle, por fecha y categoría (comidas, hoteles y transporte). La siguiente imagen muestra cómo la mostraría una aplicación de usuario visual.

INFORME DE GASTOS DE VIAJE

	COMIDAS	HOTELES	TRANSPORTE	SUBTOTALES
San Jose				
25-08-97	37.74	112.00	45.00	
26-08-97	27.28	112.00	45.00	
subtotales	65.02	224.00	90.00	379.02
Seattle				
27-08-97	96.25	109.00	36.00	
28-08-97	35.00	109.00	36.00	
subtotales	131.25	218.00	72.00	421.25
Totales	196.27	442.00	162.00	800.27

[[Descripción de la tabla de viajes](#)]

5.2 Tablas para maquetar

Puntos de verificación en esta sección:

- [5.3](#) No utilice tablas para maquetar a menos que la tabla tenga sentido cuando se alinee. Por otro lado, si la tabla no tiene sentido, proporcione una alternativa equivalente (la cual debe ser una versión alineada). [Prioridad 2]
- [5.4](#) Si se utiliza una tabla para maquetar, no utilice marcadores estructurales para realizar efecto visual de formato. [Prioridad 2]

Los autores deberían utilizar hojas de estilo para [maquetar y posicionar](#). De cualquier modo, cuando es necesario usar una tabla para maquetar, el orden de la información en la tabla debe ser comprensible después de transformarse en forma lineal. Cuando se alinea una tabla, los contenidos de las celdas se convierten en series de párrafos (es decir, de arriba hacia abajo) uno tras otro. Las celdas deben tener sentido cuando se leen en orden horizontal y deben incluir elementos estructurales (que creen párrafos, encabezamientos, listas, etc.) de modo que la página tenga sentido al ser alineada.

Igualmente, cuando se utilicen tablas para maquetar, no utilice etiquetas estructurales para crear formatos visuales. Por ejemplo, el elemento TH se muestra visualmente centrado y en negrita. Si una celda no es realmente el encabezamiento de una fila o columna de datos, utilice hojas de estilo o atributos de formateo del elemento.

5.3 Alineación de las Tablas

Puntos de verificación en esta sección:

- [10.3](#) Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten correctamente los textos contiguos, proporcione un texto lineal alternativo (en la página actual o en alguna otra) para todas las tablas que maquetan texto en paralelo en columnas de palabras. [Prioridad 3]

Las tablas utilizadas para maquetar páginas y en lo que en una celda el texto se extiende a través de más de una línea, plantean problemas para los lectores de pantalla antiguos, que no interpretan el código fuente HTML o para los navegadores que no permiten la navegación por celdas individuales de tablas. Estos lectores de pantalla leerán la página por líneas completas, leyendo las frases de toda una línea pero de cada columna como si fueran una sola frase.

Por ejemplo, si una tabla aparece así en la pantalla::

Hay un 30% de posibilidad de que
llueva esta mañana, pero ellos
deberían parar antes del fin de semana.

Las clases de la Universidad de Wisconsin se
reanudarán el 3 de septiembre.

El lector de pantalla la leería como:

Hay un 30% de posibilidad de que las clases de la Universidad de Wisconsin se llueva esta mañana, pero ellos reanudarán el 3 de septiembre. deberían parar antes del fin de semana.

Normalmente es fácil ver el texto lineal de una tabla empleada para maquetar la página - simplemente se quitan las etiquetas de tabla. Existen varias herramientas para realizar esta tarea, y cada vez más los lectores de pantalla y algunos navegadores alinean las tablas.

Sin embargo, la alineación de las [tablas de datos](#) requiere una estrategia diferente. Puesto que las celdas de datos dependen de la información proporcionada por las celdas que las rodean y las de encabezado, la información sobre las relaciones entre ellas proporcionada visualmente debe ser trasladada a la tabla alineada.

Por ejemplo, para especificar la orden de la disposición de las columnas. La dirección de escritura del idioma puede afectar la disposición de las columnas y por lo tanto el orden de la tabla alineada. El atributo "[dir](#)" especifica la secuencia de la disposición de las columnas (por ejemplo, `dir="rtl"` especifica una disposición de derecha a izquierda).

Este etiquetado también ayudará a los navegadores a alinear las tablas (también llamado "serialización" de tablas). Se puede crear una versión lineal basada en filas empezando con el encabezado de la fila, y también precediendo cada celda con el encabezado de columna de la celda. O, la alineación puede basarse en la columna. Los navegadores y las ayudas técnicas futuras podrán traducir automáticamente las tablas a secuencias lineales o navegar por la tabla celda por celda si los datos se etiquetan adecuadamente. El grupo de trabajo de la WAI sobre la Evaluación y Reparación está siguiendo el progreso de las herramientas y desarrolla sus propias herramientas que permitirán que los usuarios alineen o que naveguen las tablas celda por celda. Véase [\[WAL-ER\]](#).

¡Test rápido! Para entender mejor cómo un lector de pantalla leería una tabla, coloque un trozo de papel sobre la página y lea la tabla línea a línea.

5.4 Cuestiones de compatibilidad con versiones anteriores para tablas

Con los navegadores diseñados para HTML 3.2, las filas de un elemento [TFOOT](#) aparecerán antes que el cuerpo de la tabla.

6 Vínculos

6.1 Texto de los vínculos

Puntos de verificación en esta sección:

- [13.1](#) Identifique claramente el objetivo de cada vínculo [Prioridad 2].

Un buen texto para un vínculo de no debería ser demasiado general; no utilice frases como "pinche aquí". Esta frase no sólo es específica a un dispositivo (implica un dispositivo de apuntamiento) sino que no indica nada acerca de lo que se encontrará si se sigue el vínculo. En lugar de "pinche aquí", el texto del vínculo debería indicar la naturaleza del objetivo del vínculo, como en "más información sobre los leones marinos" o "versión sólo-texto de esta página". Tenga en cuenta que, para este último caso (y otros documentos con formato o idioma específicos), se insta a los desarrolladores de contenido a usar la [negociación del contenido](#) como alternativa, de forma que los usuarios que prefieran las versiones de texto las obtengan automáticamente.

Además de textos claros en los vínculos, los desarrolladores de contenido pueden especificar un valor del atributo "title" que clara y concisamente describa el objetivo del vínculo.

Si hay más de un vínculo en una página con el mismo texto, todos estos vínculos deben remitir al mismo recurso. Esta coherencia ayudará al diseño de la página tanto como a la accesibilidad.

Si dos o más vínculos van referidos a objetivos diferentes pero comparten el mismo texto, distinga los vínculos especificando un valor diferente para el atributo "[title](#)" de cada elemento [A](#).

Los "usuarios auditivos" -- personas ciegas, con dificultades de visión o que utilizan mecanismos con pantallas pequeñas o sin pantalla -- no pueden revisar rápidamente una página con sus ojos. Para tener una visión rápida de la página o encontrar rápidamente un vínculo, estos usuarios a menudo saltarán de un vínculo a otro con la tecla tabulador o revisarán una lista de vínculos disponibles en una página.

Así pues, para una serie de vínculos relacionados, incluya información introductoria en el primer vínculo e información distintiva en los vínculos siguientes. Ello proporcionará información contextual para los usuarios que los leen en orden secuencial.

Ejemplo.

```
<A href="my-doc.html">Mi documento está disponible en HTML</A> ,  
<A href="my-doc.pdf" title="Mi documento en PDF">PDF</A> ,  
<A href="my-doc.txt" title="Mi documento en formato texto">texto</A>
```

6.1.1 Texto para imágenes utilizadas como vínculos

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]

Cuando se use una imagen como contenido de un vínculo, especifique un [texto equivalente](#) para la imagen.

Ejemplo.

```
<A href="rutas.html">  
  <IMG src="topo.html"  
    alt="Rutas actuales en Gimnasio de escalada Boulders">  
</A>
```

O, si proporciona un texto para el vínculo, use un espacio como el valor del atributo "alt" del elemento IMG. Tenga en cuenta que este texto aparecerá en la página junto a la imagen.

Ejemplo.

```
<A href="rutas.html">  
  <IMG src="topo.html" alt=" ">  
    Rutas actuales en Gimnasio de escalada Boulders  
</A>
```

6.2 Agrupar y saltar vínculos

Puntos de verificación en esta sección:

- [13.6](#) Agrupe los vínculos relacionados, identifique el grupo (para las aplicaciones de usuario) y, hasta que las aplicaciones de usuario lo hagan, proporcione una manera de evitar el grupo [Prioridad 3].
- [10.5](#) Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten claramente los vínculos contiguos, incluya caracteres imprimibles (rodeados de espacios), que no sirvan como vínculo, entre los vínculos contiguos [Prioridad 3]

Cuando los vínculos se agrupan en conjuntos lógicos (por ejemplo, en una barra de navegación que aparezca en todas las páginas de un sitio), deberían estar etiquetados como una unidad. Las barras de

navegación son normalmente lo primero que uno encuentra en una página. Para los usuarios con sintetizador de voz, ello implica tener que escuchar una serie de vínculos en todas las páginas antes de llegar a los contenidos interesantes de una página. Hay varias maneras para permitir a los usuarios saltar un grupo de vínculos (tal y como hacen los usuarios videntes cuando ven el mismo comienzo en cada página):

- Incluya un vínculo que permita al usuario saltar sobre el conjunto de vínculos de navegación.
- Proporcione una hoja de estilo que permita a los usuarios ocultar el conjunto de vínculos de navegación.
- Use el elemento MAP de HTML 4.01 para agrupar los vínculos, e identifique el grupo con el atributo "title".

En el futuro, las aplicaciones de usuario podrán permitir a los usuarios saltar sobre elementos como barras de navegación.

Ejemplo.

En este ejemplo, el elemento **MAP** agrupa un conjunto de vínculos. El atributo "title" lo identifica como barra de navegación, y un vínculo al principio del grupo sirve de vínculo a un marcador después del grupo. Observe que los vínculos se separan con caracteres imprimibles, que no forman parte de un vínculo (rodeados de espacios).

```
<BODY>
  <MAP title="Barra de navegación">
    <P>
      [<A href="#como">Salta barra de navegación</A>]
      [<A href="home.html">Página principal</A>]
      [<A href="search.html">Buscar</A>]
      [<A href="new.html">Novedades y destacados</A>]
      [<A href="mapasitio.html">Mapa del sitio</A>]
    </P>
  </MAP>
  <H1><A name="como">Cómo utilizar nuestro sitio</A></H1>
  >
  <!-- contenido de la página -->
</BODY>
```

6.3 Acceso desde el teclado

Puntos de verificación en esta sección:

- [9.4](#) Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos. [Prioridad 3]
- [9.5](#) Proporcione atajos de teclado para los vínculos más importantes (incluidos los de los mapas de imagen de cliente), los controles de formulario y los grupos de controles de formulario. [Prioridad 3]

El acceso a los elementos activos de una página es importante para muchos usuarios que no pueden manejar un dispositivo de apuntamiento. Las aplicaciones de usuario pueden incorporar características que permitan a los usuarios asignar acciones de teclado a ciertas acciones. HTML 4.01 permite a los desarrolladores de contenidos especificar atajos de teclado en sus documentos mediante el atributo "accesskey".

Ejemplo.

En este ejemplo, si el usuario activa la tecla "C", se seguirá el vínculo.

```
<A accesskey="C" href="doc.html" hreflang="en"
  title="Página de la empresa XYZ">
  Página de la empresa XYZ</A>
```

Nota: Hasta que las aplicaciones de usuario proporcionen un resumen de las asignaciones de teclas

disponibles, proporcione información sobre las asignaciones de teclas.

6.4 Vínculos y objetivos

Puntos de verificación en esta sección:

- [10.1](#) Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario. [Prioridad 2]

7 Imágenes y mapas de imagen

Los apartados siguientes tratan de la accesibilidad de las imágenes (incluyendo animaciones sencillas tales como animaciones GIF) y mapas de imagen.

Para información sobre las matemáticas presentadas en forma de imágenes, consulte la sección sobre [cómo usar etiquetas con texto y hojas de estilo en vez de imágenes](#).

7.1 Equivalentes textuales breves para imágenes ("texto alt")

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]

Cuando utilice [IMG](#), proporcione un breve [equivalente textual](#) con el atributo "[alt](#)". **Nota.** El valor de este atributo es denominado "texto alt".

Ejemplo.

```
<IMG src="lupa.gif" alt="Buscar">
```

Cuando utilice [OBJECT](#), proporcione un [equivalente textual](#) en el contenido del elemento OBJECT:

Ejemplo.

```
<OBJECT data="lupa.gif" type="image/gif">
  Buscar
</OBJECT>
```

7.2 Descripciones largas de imágenes

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]

Cuando un breve texto equivalente no es suficiente para transmitir adecuadamente la función o propósito de una imagen, proporcione información adicional en un archivo designado por el atributo "[longdesc](#)":

Ejemplo.

```
<IMG src="ventas97.gif" alt="Ventas en 1997"
      longdesc="ventas97.html">
```

En ventas97.html:

Un gráfico muestra cómo evolucionaron las ventas en 1997. Es un gráfico de barras que muestra el aumento de las ventas por meses en porcentaje. Las ventas en enero aumentaron un 10% sobre las de diciembre 1996. En febrero las ventas cayeron en un 3%, ..

Para las aplicaciones de usuario que no soportan "longdesc", proporcione también un [vínculo a una descripción](#) al lado del gráfico:

Ejemplo.

```
<IMG src="97sales.gif" alt="Ventas en 1997" longdesc="ventas.html">
<A href="ventas.html" title="Descripción de las cifras de ventas en 1997">[D]</A>
```

Cuando utilice OBJECT, especifique un texto equivalente más extenso en el contenido del elemento:

Ejemplo.

```
<OBJECT data="97ventas.gif" type="image/gif">
    Las ventas en 1997 bajaron como consecuencia de nuestra
    <A href="anticipado.html">compra anticipada</A> ...
</OBJECT>
```

Observe que el contenido de OBJECT, a diferencia del texto en "alt", puede incluir etiquetas. Así pues, los desarrolladores de contenido pueden proporcionar un vínculo hacia información adicional desde el elemento OBJECT:

Ejemplo.

```
<OBJECT data="97sales.gif" type="image/gif">
    Gráfico de nuestras ventas en 1997.
    Está disponible una
    <A href="desc.html">descripción textual</A>.
</OBJECT>
```

7.2.1 Vínculos-d invisibles

Nota. Los vínculos-d invisibles están desaconsejados. Se debe de utilizar el atributo "longdesc"..

Un [vínculo-d](#) invisible es una imagen pequeña (1 píxel) o transparente cuyo valor del atributo "[alt](#)" es "vínculo-D" o "D" y que forma parte del contenido de un elemento [A](#). Al igual que otros vínculos-d, apunta a un equivalente textual de la imagen asociada. Al igual que otros vínculos, el usuario puede desplazarse a él con la tecla tabulador. De esta manera, los vínculos-d invisible proporcionan una solución (temporal) para los diseñadores que desean evitar los vínculos-d invisibles por motivos de estética.

7.3 Dibujos con caracteres (ASCII art)

Puntos de verificación en esta sección:

- [13.10](#) Proporcione un medio para saltar sobre un ASCII art de varias líneas. [Prioridad 3]

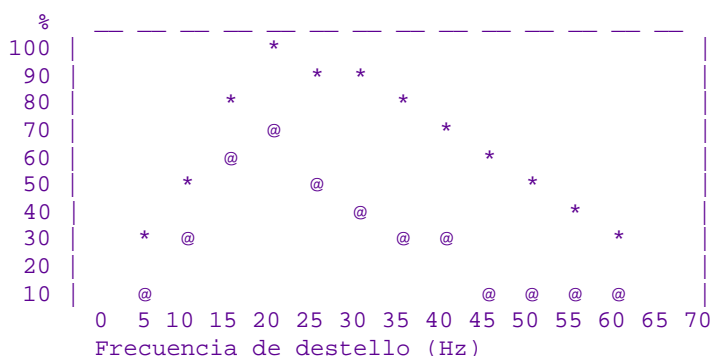
No utilice dibujos con caracteres, y utilice en su lugar imágenes reales, puesto que es más fácil proporcionar [texto equivalente](#) para imágenes. No obstante, si tiene que utilizar dibujos con caracteres, proporcione un vínculo para saltar sobre el dibujo, tal y como se muestra a continuación:

Ejemplo.

```
<P>
<a href="#despues">Saltar dibujo</a>
<!-- Dibujo aquí -->
<a name="después">Título del dibujo</a>
```

Los dibujos con caracteres se pueden etiquetar como sigue [[saltar dibujo con caracteres](#) o consultar una [descripción del gráfico](#)]:

Ejemplo.



Otra opción para los dibujos más pequeños con caracteres es emplear un elemento [ABBR](#) con un ["title"](#).

Ejemplo.

```
<P><ABBR title="carita con sonrisa en caracteres">:-)</ABBR>
```

Si el dibujo es complejo, asegúrese de que el equivalente textual lo describe adecuadamente.

Otra manera de sustituir los dibujos con caracteres es emplear sustitutos de lenguaje natural. Por ejemplo, `<guiño>` puede servir para sustituir una sonrisa con guiño ;-). O las palabras "por tanto" pueden sustituir a las flechas realizadas con guiones y el signo "mayor que" (por ejemplo: --->), y la palabra "dados" para la poco común abreviatura "da2".

7.4 Mapas de imagen

Un mapa de imagen es una imagen que tiene "zonas activas". Cuando el usuario selecciona una de las regiones, ocurre una acción -- se puede activar un vínculo, se pueden enviar datos al servidor, etc. Para hacer accesible un mapa de imagen, los desarrolladores de contenidos deben asegurarse de que cada acción asociada con una zona visual pueda ser activada sin ningún dispositivo de puntero.

Los mapas de imagen se crean con el elemento [MAP](#). HTML permite dos tipos de mapa de imagen: lado-cliente (el navegador del usuario procesa un URI) y lado-servidor (el servidor procesa las coordenadas del punto seleccionado). Para todos los mapas de imagen, los desarrolladores de contenidos deben proporcionar un [equivalente textual](#).

Los desarrolladores deben crear mapas de imagen lado cliente (con `"usemap"`) y no mapas de lado servidor (con `ismap`), porque los mapas lado servidor precisan un dispositivo de entrada específico. Si resulta imprescindible utilizar un mapa lado servidor (por ejemplo, porque la geometría de la zona no se puede representar mediante valores del atributo `shape`), los autores deben proporcionar la misma funcionalidad o información en una forma alternativa, accesible. Una manera de proporcionar esta

alternativa es de proporcionar un vínculo de texto para cada zona activa para permitir acceder a cada vínculo mediante el [teclado](#). Si resulta imprescindible emplear un mapa de imagen lado servidor, véase la sección sobre [mapas de imagen lado servidor](#).

7.4.1 Equivalentes textuales para mapas de imagen de cliente

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]

Proporcione [equivalentes textuales](#) para los mapas de imagen si transmiten información visual. Igual que otros vínculos, el texto debe tener sentido cuando se lee fuera de su contexto. Véase [la sección sobre el texto de los vínculos](#) para información sobre cómo redactar buen texto para vínculos. Los usuarios pueden desear teclas de acceso rápido para vínculos de uso frecuente. Véase [la sección sobre el acceso a los vínculos mediante teclado](#).

Si emplea [AREA](#) para crear el mapa, use el atributo "[alt](#)":

Ejemplo.

```
<IMG src="bienvenido.gif" alt="Mapa de imagen de las zonas de la biblioteca"
    usemap="#map1">
<MAP name="map1">
  <AREA shape="rect" coords="0,0,30,30"
        href="consulta.html" alt="Consulta">
  <AREA shape="rect" coords="34,34,100,100"
        href="medios.html" alt="Laboratorio audiovisual">
</MAP>
```

El siguiente ejemplo ilustra la misma idea, pero emplea [OBJECT](#) en vez de [IMG](#) para insertar la imagen para proporcionar más información sobre la imagen:

Ejemplo.

```
<OBJECT data="bienvenido.gif" type="image/gif" usemap="#map1">
  La biblioteca tiene varias zonas, entre las cuales están
  la <A href="consulta.html">Sección de Consulta</A> y el
  <A href="medios.html">Laboratorio Audiovisual</A>.
</OBJECT>
<MAP name="map1">
  <AREA shape="rect" coords="0,0,30,30"
        href="consulta.html" alt="Consulta">
  <AREA shape="rect" coords="34,34,100,100"
        href="medios.html" alt="Laboratorio Audiovisual">
</MAP>
```

7.4.2 Vínculos redundantes de texto para mapas lado cliente

Puntos de verificación en esta sección:

- [1.5](#) Hasta que las aplicaciones de usuario interpreten el texto equivalente para los vínculos de los mapas de imagen de cliente, proporcione vínculos texto redundantes para cada zona activa del mapa de imagen de cliente [Prioridad 3].

Además de un equivalente textual, proporcione vínculos textuales redundantes. Si emplea el elemento [A](#) en vez de [AREA](#), el diseñador puede describir las regiones activas y proporcionar vínculos redundantes al mismo tiempo:

Ejemplo.

```
<OBJECT data="barranav1.gif" type="image/gif" usemap="#map1">
<MAP name="map1">
  <P>Navegar por el sitio.
  [<A href="guia.html" shape="rect"
    coords="0,0,118,28">Guía de acceso</A>]
  [<A href="atajos.html" shape="rect"
    coords="118,0,184,28">Ir</A>]
  [<A href="buscar.html" shape="circle"
    coords="184,200,60">Buscar</A>]
  [<A href="top10.html" shape="poly"
    coords="276,0,276,28,100,200,50,50,276,0">
    Diez principales</A>]
</MAP>
</OBJECT>
```

Observe que en el ejemplo anterior, el elemento MAP es el contenido del elemento OBJECT, de forma que los vínculos alternativos sólo son visibles si el mapa de imagen (barranav1.gif) no lo es.

Observe también que los vínculos están separados por corchetes ([]). Esto se hace para impedir que los lectores de pantalla antiguos lean vínculos adyacentes como si fueran un solo vínculo, así como para ayudar a los usuarios videntes a distinguir visualmente los diferentes vínculos.

Los desarrolladores de contenidos deberían asegurarse de incluir caracteres imprimibles (tales como corchetes o una barra vertical (|) separados por espacios en blanco entre los vínculos textuales adyacentes. El problema no ocurre si los vínculos son imágenes; el texto alternativo no se lee como un vínculo único debido a las imágenes de marcación de emplazamiento empleadas por los navegadores gráficos cuando no se muestran las imágenes. Para mayor información, véase la sección sobre [Agrupación y salto de vínculos](#).

7.4.3 Mapas de imagen lado cliente y lado servidor

Puntos de verificación en esta sección:

- [9.1](#) Proporcione mapas de imagen controlados por el cliente en lugar de por el servidor, excepto donde las zonas sensibles no puedan ser definidas con una forma geométrica. [Prioridad 1]

7.4.4 Mapas de imagen de servidor

Puntos de verificación en esta sección:

- [1.2](#) Proporcione vínculos redundantes en formato texto para cada zona activa de un mapa de imagen del servidor. [Prioridad 1]

Cuando es necesario utilizar un mapa imagen lado servidor, los desarrolladores de contenido deben proporcionar una lista alternativa de las opciones del mapa de imagen. Hay tres técnicas:

- Incluya los vínculos alternativos en el cuerpo de un elemento [OBJECT](#) (consulte el [ejemplo anterior que muestra cómo incluir vínculos en el elemento OBJECT](#)).
- Si se utiliza [IMG](#) para insertar la imagen, proporcione una lista alternativa de vínculos después de la imagen, e indique la existencia y ubicación de la lista alternativa (por ejemplo, con el atributo "[alt](#)").

Ejemplo.

```
<A href="http://www.ejemplo.com/cgi-bin/imagemap/mi-mapa">
<IMG src="bienvenido.gif" alt="¡Bienvenido! (Vínculos textuales mas abajo)" :
</A>

<P>[<A href="consulta.html">Consulta</A>]
  [<A href="medios.html">Laboratorio Audiovisual</A>]
```

- Si no hay otra manera de hacer el mapa de imagen accesible, cree una [página alternativa](#) que sí es accesible.

Los mapas de imagen del lado servidor y del lado cliente pueden ser usados como botones de envío en formularios. Para más información consulte la sección sobre [botones gráficos](#).

7.5 Color en las imágenes

Puntos de verificación en esta sección:

- [2.2](#) Asegúrese de que las combinaciones de los colores de fondo y primer plano tengan suficiente contraste para que sean percibidas por personas con deficiencias de percepción de color o en pantallas en blanco y negro. [Prioridad 2 para las imágenes. Prioridad 3 para el texto]

7.6 Imágenes animadas

Puntos de verificación en esta sección:

- [7.3](#) Hasta que las aplicaciones de usuario permitan congelar el movimiento de los contenidos, evite los movimientos en las páginas. [Prioridad 2]

8 Applets y otros objetos programados

Aunque los applets pueden incluirse en un documento tanto con el elemento [APPLET](#) como con [OBJECT](#), [OBJECT](#) es el método preferido.

8.1 Equivalentes textuales y no textuales para applets y objetos programados

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]
- [6.2](#) Asegúrese de que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico. [Prioridad 1]
- [6.3](#) Asegúrese de que las páginas sigan siendo utilizables se desconecten o no se soporten los scripts, applets u otros objetos programados. Si esto no es posible, proporcione información equivalente en una página alternativa accesible. [Prioridad 1]

Si se utiliza [OBJECT](#), proporcione un [texto equivalente](#) en el contenido del elemento:

Ejemplo.

```
<OBJECT classid="java:Press.class" width="500" height="500">
  A medida que la temperatura aumenta, las moléculas del globo...
</OBJECT>
```

Un ejemplo más complejo aprovecha el hecho de que los elementos OBJECT pueden ser anidados para proporcionar representaciones alternativas de la información:

Ejemplo.

```
<OBJECT classid="java:Press.class" width="500" height="500">
  <OBJECT data="Presion.mpeg" type="video/mpeg">
    <OBJECT data="Presion.gif" type="image/gif">
      A medida que la temperatura aumenta, las moléculas del globo...
    </OBJECT>
  </OBJECT>
</OBJECT>
```

Si utiliza [APPLET](#), proporcione un [equivalente textual](#) con el atributo "alt" y dentro del contenido del elemento APPLET. Esto permite una transformación satisfactoria del contenido para aquellas aplicaciones de usuario que sólo soportan uno de los dos mecanismos ("alt" o contenido).

Ejemplo desaconsejado.

```
<APPLET code="Press.class" width="500" height="500"
      alt="Applet Java: cómo la temperatura afecta a la presión">
  A medida que la temperatura aumenta, las moléculas del globo...
</APPLET>
```

8.2 Applets accesibles directamente

Puntos de verificación en esta sección:

- [8.1](#) Haga los elementos de programación tales como scripts y applets directamente accesibles o compatibles con las ayudas técnicas. [Prioridad 1 si la funcionalidad es importante y no se presenta en otro lugar; de otra manera Prioridad 2]

Si un applet (creado con [OBJECT](#) o con [APPLET](#)) precisa una interacción con el usuario (por ejemplo, la capacidad de manipular un experimento físico) que no se puede reproducir en un formato alternativo, se debe hacer el applet accesible directamente.

Si un applet causa movimiento, el diseñador debe proporcionar un mecanismo para parar este movimiento (por ejemplo, véase [\[TRACE\]](#)). También, consúltese el apartado siguiente para aprender cómo hacer accesibles las presentaciones de audio y vídeo.

Para mayor información sobre el desarrollo de applets accesibles, consulte [\[JAVAACCESS\]](#) y [\[IBMJAVA\]](#). Estas empresas están desarrollando un API para la accesibilidad así como haciendo accesibles las clases de Java Swing.

Puntos de verificación relacionados:

- [3.4](#) Utilice unidades relativas en lugar de absolutas al especificar los valores en los atributos de los marcadores de lenguaje en los valores de las propiedades de las hojas de estilo. [Prioridad 2],
- [6.4](#) Para los scripts y applets, asegúrese de que los manejadores de evento sean independientes del dispositivo de entrada. [Prioridad 2],
- [6.5](#) Asegúrese de que los contenidos dinámicos son accesibles o proporcione una página o presentación alternativa. [Prioridad 2],
- [7.1](#) Hasta que las aplicaciones de usuario permitan controlarlo, evite provocar destellos en la pantalla. [Prioridad 1],
- [7.2](#) Hasta que las aplicaciones de usuario permitan controlarlo, evite el parpadeo del contenido (por ejemplo, cambio de presentación en periodos regulares, así como el encendido y apagado). [Prioridad 2],
- [7.3](#) Hasta que las aplicaciones de usuario permitan congelar el movimiento de los contenidos, evite los movimientos en las páginas. [Prioridad 2],
- [7.4](#) Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener las actualizaciones, no cree páginas que se actualicen automáticamente de forma periódica. [Prioridad 2],
- [9.2](#) Asegúrese de que cualquier elemento que tiene su propia interfaz pueda manejarse de forma independiente del dispositivo [Prioridad 2],
- y [10.1](#) Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario. [Prioridad 2].
- [1.4](#) Para toda presentación multimedia tempodependiente (por ejemplo, una película o animación) sincronice alternativas equivalentes (por ejemplo, subtítulos o descripciones de la banda visual) con la presentación [Prioridad 1].

9 Audio y vídeo

9.1 Información sonora

9.2 Equivalentes textuales para multimedia

Puntos de verificación esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]

Cuando sea necesario, se debe proporcionar un [equivalente textual](#) de la información visual para permitir la comprensión de la página. Por ejemplo, se puede considerar la inclusión de una animación repetitiva que muestra la nubosidad y la precipitación como parte del boletín meteorológico. Puesto que la animación suplementa el resto del boletín (que se presenta en lenguaje natural, es decir, texto), una descripción breve es suficiente. No obstante, si la animación se emplea en un contexto pedagógico, en donde los alumnos aprenden acerca de la formación de las nubes en relación con la masa terrestre, entonces la animación precisa una descripción para los que no pueden ver la animación pero igualmente desean aprender la lección.

9.3 Objetos multimedia incrustados

Otros objetos, por ejemplo los que precisan un plug-in, también deben incrustarse mediante el elemento [OBJECT](#). No obstante, para asegurar la compatibilidad con versiones anteriores de navegadores Netscape, se debe emplear el elemento propietario EMBED anidado dentro del elemento OBJECT como sigue:

Ejemplo desaconsejado.

```
<OBJECT classid="clsid:A12BCD3F-GH4I-56JK-xyz"
codebase="http://ejemplo.com/content.cab" width=100 height=80>
<PARAM name="Película" value="nombrepeli.swf">
  <EMBED src="nombrepeli.swf" width=100 height=80
  pluginspage="http://ejemplo.com/shockwave/download/">
  </EMBED>

  <NOEMBED>
    <IMG alt="Imagen estática de la película"
    src="nombrepeli.gif" width=100 height=80>
  </NOEMBED>

</OBJECT>
```

Para mayor información, véase [\[MACROMEDIA\]](#).

10 Marcos (frames)

Para los usuarios videntes, los marcos sirven para organizar la página en zonas diferentes. Para los usuarios invidentes, las relaciones entre los contenidos en los distintos marcos (por ejemplo, cuando un marco contiene el índice y otro el contenido principal) debe transmitirse de otra manera.

Los marcos tal y como se implementan en la actualidad (con los elementos [FRAMESET](#), [FRAME](#), y [IFRAME](#)) son problemáticos por varios motivos:

- Sin *scripts*, los marcos tienden a degradar la funcionalidad del botón "página anterior" del navegador.
- Es imposible hacer referencia al "estado actual" de un conjunto de marcos con URI; una vez que cambien los contenidos de un conjunto de marcos, el URI original ya no es válido.
- Al abrir el contenido de un marco en una ventana nueva puede desorientar al usuario o simplemente molestarle.

En las secciones siguientes, tratamos sobre cómo hacer los marcos más accesibles. También proponemos

algunas [alternativas a los marcos](#) que emplean el HTML 4.01 y CSS e intentamos superar las limitaciones de las implementaciones actuales de los marcos.

10.1 Proporcionar un título para cada marco

Puntos de verificación en esta sección:

- [12.1](#) Titule cada marco para facilitar su identificación y navegación. [Prioridad 1]

Ejemplo.

Emplear el atributo "title" para dar nombre a los marcos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<HTML>
<HEAD>
<TITLE>Documento con un conjunto de marcos sencillo</TITLE>
</HEAD>
<FRAMESET cols="10%, 90%"
          title="Nuestra biblioteca de documentos electrónicos">
  <FRAME src="nav.html" title="Barra de navegación">
  <FRAME src="doc.html" title="Documentos">
</NOFRAMES>
  <A href="lib.html" title="Enlace a biblioteca">
    Seleccionar para ir a la biblioteca electrónica</A>
</NOFRAMES>
</FRAMESET>
```

10.2 Describir las relaciones entre marcos

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]
- [12.2](#) Describa el propósito de los marcos y cómo éstos se relacionan entre sí, si no resulta obvio solamente con el título del marco. [Prioridad 2]

Ejemplo.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<HTML>
<HEAD>
  <TITLE>Las noticias de hoy</TITLE>
</HEAD>

<FRAMESET cols="10%,*,10%">

  <FRAMESET rows="20%,*">
    <FRAME src="promo.html" name="promo" title="promociones">
    <FRAME src="barranavega.html" name="barranavega"
          title="Barra de navegación global del sitio"
          longdesc="frameset-desc.html#barranavega">
  </FRAMESET>

  <FRAME src="historia.html" name="noticia"
        title="Noticia seleccionada - contenido principal"
        longdesc="frameset-desc.html#noticia">

  <FRAMESET rows="*,20%">
    <FRAME src="titulares.html" name="indice">
```

```

        title="Índice de otros titulares nacionales"
        longdesc="frameset-desc.html#titulares">
<FRAME src="anuncio.html" name="espacioanuncio" title="Publicidad">
</FRAMESET>

<NOFRAMES>
    <p><a href="noframes.html">Versión sin marcos</a></p>
    <p><a href="frameset-desc.html">Descripciones de los marcos.</a></p>
</NOFRAMES>

</FRAMESET>
</HTML>

```

El texto de la página frameset-desc.html puede ser algo como:

```

#barranavega - este marco contiene vínculos a las secciones
               más importantes de este sitio: Noticias del mundo,
               Noticias nacionales, Noticias locales, Noticias tecnología,
               y Noticias del ocio.

#noticia      - este marco contiene la historia actualmente seleccionada.

#indice       - este marco contiene vínculos a las noticias principales de hoy
               dentro de esta sección.

```

Observe que si el contenido de un marco cambia, el equivalente textual ya no será aplicable. Además, los vínculos a las descripciones de los marcos deben incluirse junto con otro contenido alternativo en el elemento [NOFRAMES](#) del [FRAMESET](#).

10.3 Crear páginas para navegadores que no soportan FRAME

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]
- [6.5](#) Asegúrese de que los contenidos dinámicos son accesibles o proporcione una página o presentación alternativa. [Prioridad 2]

Ejemplo.

En este ejemplo, si el usuario lee "arriba.html":

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<HTML>
<HEAD>
<TITLE>Este es arriba.html</TITLE>
</HEAD>
<FRAMESET cols="50%, 50%" title="Nuestro documento largo">
    <FRAME src="principal.html" title="Donde se muestra el contenido">
    <FRAME src="tabla_de_contenido.html" title="Tabla de contenidos">
    <NOFRAMES>
        <A href="tabla_de_contenido.html">Tabla de contenidos.</A>
        <!-- otros vínculos de navegación visibles en principal.html
             también son visibles aquí. -->
    </NOFRAMES>
</FRAMESET>
</HTML>

```

y la aplicación de usuario no muestra los marcos, el usuario podrá acceder (mediante un vínculo) a una versión sin marcos de la misma información.

10.4 Contenidos fuente para marcos

Puntos de verificación en esta sección:

- [6.2](#) Asegúrese de que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico. [Prioridad 1]

Los diseñadores de contenidos deben proporcionar un equivalente textual de los marcos para asegurar que los contenidos de los mismos y las relaciones entre los marcos tienen sentido. Observe que cuando cambia el contenido del marco, entonces debe cambiar la descripción también. Esto no es posible si se inserta una IMG directamente dentro del marco. Por este motivo, el diseñador de contenidos siempre debe hacer que el origen ("src") del marco sea un archivo HTML. Las imágenes se pueden insertar en el archivo HTML y sus alternativas textuales cambiarán correctamente.

Ejemplo.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<HTML>
<HEAD>
<TITLE>Un documento de marcos correcto</TITLE>
</HEAD>
<FRAMESET cols="100%" title="Conjunto de marcos que evoluciona">
<FRAME name="marcobueno" src="manzanos.html" title="Manzanos">
</FRAMESET>
</HTML>

<!-- En manzanos.html -->
<P><IMG src="manzanos.gif" alt="Manzanos">
```

El siguiente ejemplo desaconsejado no debe emplearse puesto que inserta una [IMG](#) en un marco directamente:

Ejemplo desaconsejado.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<HTML>
<HEAD>
<TITLE>Un documento de marcos incorrecto</TITLE>
</HEAD>
<FRAMESET cols="100%" title="conjunto de marcos estático">
  <FRAME name="marcomalo"
    src="manzanos.gif" title="Manzanos">
</FRAMESET>
</HTML>
```

Observe que si, por ejemplo, un vínculo causa la inserción de una nueva imagen en el marco:

```
<P>Visite un huerto de
<A target="marcomalo" href="naranjos.gif"
  title="Naranjos">naranjos</A>
```

el título inicial del marco ("Manzanos") ya no corresponderá con el contenido actual del marco ("Naranjos").

10.5 Usar objetivos (target) para FRAME

Puntos de verificación en esta sección:

- [10.1](#) Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario. [Prioridad 2]

Los diseñadores de contenidos deben evitar de especificar una ventana nueva como objetivo (target) de un marco con target="_blank".

10.6 Alternativas a los marcos

Uno de los usos más comunes de los marcos es para dividir la ventana del navegador en dos partes: una ventana de navegación y una de contenidos. Como alternativa a los marcos, se recomienda probar lo siguiente:

1. Crear un documento para el mecanismo de navegación (que llamamos "nav.html"). Un documento separado permite compartir el mecanismo de navegación entre múltiples documentos.
2. En cada documento que requiere un mecanismo de navegación, incluirlo al final del documento con el código siguiente (o similar) de [OBJECT](#):

Ejemplo.

```
<P>
    <OBJECT data="nav.html">
    Ir al <A href="nav.html">índice de contenidos</A>
    </OBJECT>
```

Se incluye el mecanismo de navegación al final del documento para dar al usuario con hojas de estilo desactivadas, acceso a la información más importante en primer lugar.

3. Use hojas de estilo para colocar el mecanismo de navegación en la posición deseada dentro de la ventana. Por ejemplo, con la siguiente regla CSS la barra de navegación se coloca en el lateral izquierdo de la página y ocupa 25% del espacio horizontal disponible:

```
OBJECT { float: left; width: 25% }
```

Con la siguiente regla CSS se fija el mecanismo de navegación al final de la página a la izquierda y lo mantiene en esa posición aunque el usuario se desplace hacia abajo en la página:

```
OBJECT { position: fixed; left: 0; bottom: 0 }
```

Nota. Los mecanismos de navegación y otro contenidos se pueden insertar en el documento mediante 'includes' en el servidor.

10.6.1 Fijar el tamaño de los marcos con unidades relativas

Puntos de verificación en esta sección:

- [3.4](#) Utilice unidades relativas en lugar de absolutas para especificar los valores en los atributos de los marcadores de lenguaje y en los valores de las propiedades de las hojas de estilo. [Prioridad 2]

En los ejemplos anteriores, observe que los tamaños de los marcos se especifican como porcentajes. Cuando el usuario cambia el tamaño de la ventana los marcos se ajustan también y siguen siendo legibles.

11 Formularios

Esta sección trata de la accesibilidad de los formularios y los controles que se pueden poner dentro de un elemento [FORM](#).

11.1 Acceso a los formularios mediante el teclado

Puntos de verificación en esta sección:

- [9.4](#) Cree una orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos. [Prioridad 3]
- [9.5](#) Proporcione atajos de teclado para los vínculos más importantes (incluidos los de los mapas de imagen de cliente), los controles de formulario y los grupos de controles de formulario. [Prioridad 3]

En el ejemplo siguiente, especificamos una secuencia de navegación entre elementos (en secuencia, "campo2", "campo1", "enviar") con "[tabindex](#)":

Ejemplo.

```
<FORM action="submit" method="post">
<P>
<INPUT tabindex="2" type="text" name="campo1">
<INPUT tabindex="1" type="text" name="campo2">
<INPUT tabindex="3" type="submit" name="enviar">
</FORM>
```

Este ejemplo asigna la letra "U" como tecla de acceso (con "[accesskey](#)"). Al teclear "U" el enfoque pasa a la etiqueta, que a su vez pasa el enfoque al control de entrada de texto, para permitir al usuario teclear el texto.

Ejemplo.

```
<FORM action="submit" method="post">
<P>
    <LABEL for="usuario" accesskey="U">nombre</LABEL>
    <INPUT type="text" id="usuario">
</FORM>
```

11.2 Agrupar los controles dentro del formulario

Puntos de verificación en esta sección:

- [12.3](#) Divida los bloques largos de información en grupos más manejables cuando sea natural y apropiado. [Prioridad 2]

Los diseñadores de contenidos deben [agrupar la información](#) donde esto sea natural y apropiado. Cuando los controles de formulario se pueden agrupar en unidades lógicas, utilice el elemento [FIELDSET](#) y aplique una etiqueta a esas unidades con el elemento [LEGEND](#):

Ejemplo.

```
<FORM action="http://ejemplo.com/nuevousuario" method="post">
  <FIELDSET>
    <LEGEND>Datos personales</LEGEND>
    <LABEL for="nombre">Nombre: </LABEL>
    <INPUT type="text" id="nombre" tabindex="1">
    <LABEL for="apellidos">Apellidos: </LABEL>
    <INPUT type="text" id="apellidos" tabindex="2">
    ...mas datos personales...
  </FIELDSET>
  <FIELDSET>
    <LEGEND>Historial médico</LEGEND>
    ...datos del historial médico...
  </FIELDSET>
</FORM>
```

11.2.1 Agrupar las opciones de menú

Los desarrolladores de contenido deben [agrupar la información](#) donde esto sea natural y apropiado. Para

las listas largas de selecciones en los menús (en las cuales puede resultar difícil orientarse), los diseñadores de contenidos deben agrupar los elementos [SELECT](#) (definidos con [OPTION](#)) en una jerarquía con el elemento [OPTGROUP](#). Especifique una etiqueta para el grupo de opciones con el atributo [label](#) en el OPTGROUP.

Ejemplo.

```
<FORM action="http://ejemplo.com/prog/algunprog" method="post">
<P>
<SELECT name="ComOS">
  <OPTGROUP label="PortMaster 3">
    <OPTION label="3.7.1" value="pm3_3.7.1">PortMaster 3 con ComOS 3.7.1
    <OPTION label="3.7" value="pm3_3.7">PortMaster 3 con ComOS 3.7
    <OPTION label="3.5" value="pm3_3.5">PortMaster 3 con ComOS 3.5
  </OPTGROUP>
  <OPTGROUP label="PortMaster 2">
    <OPTION label="3.7" value="pm2_3.7">PortMaster 2 con ComOS 3.7
    <OPTION label="3.5" value="pm2_3.5">PortMaster 2 con ComOS 3.5
  </OPTGROUP>
  <OPTGROUP label="IRX">
    <OPTION label="3.7R" value="IRX_3.7R">IRX con ComOS 3.7R
    <OPTION label="3.5R" value="IRX_3.5R">IRX con ComOS 3.5R
  </OPTGROUP>
</SELECT>
</FORM>
```

11.3 Etiquetas para los controles de formulario

Puntos de verificación en esta sección:

- [12.4](#) Asocie explícitamente las etiquetas con sus controles. [Prioridad 2]
- [10.2](#) Hasta que las aplicaciones de usuario soporten explícitamente la asociación entre control de formulario y etiqueta, para todos los controles de formularios con etiquetas asociadas implícitamente, asegúrese de que la etiqueta está colocada adecuadamente. [Prioridad 2]

La sección anterior muestra un ejemplo de una etiqueta ([LABEL](#)) con ["for"](#) en HTML 4.01.

Una etiqueta se asocia implícitamente con su control de formulario bien mediante las marcas o por su posición en la página. El ejemplo siguiente muestra cómo una etiqueta y un control de formulario pueden asociarse implícitamente mediante marcadores.

Ejemplo.

```
<LABEL for="nombre">Nombre:
  <INPUT type="text" id="nombre" tabindex="1">
</LABEL>
```

11.4 Botones gráficos

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de ["alt"](#), ["longdesc"](#) o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), ["applets"](#) y objetos programados, ["ascii art"](#), marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]

El uso de las imágenes para decorar los botones permite a los diseñadores hacer sus formularios diferentes y mas fáciles de comprender. El uso de una imagen para un botón (por ejemplo, con los elementos [INPUT](#) o [BUTTON](#)) no es en sí inaccesible - siempre que se proporcione un equivalente textual

para la imagen.

No obstante, un botón gráfico de envío de formulario creada con `INPUT`, `type="image"` crea un especie de mapa de imagen lado servidor. Cuando se pincha con el ratón, las coordenadas `x` e `y` se transmiten al servidor con los datos de formulario.

La sección sobre [Imágenes y mapas de imagen](#) trata de por qué se deben de evitar las imágenes lado-servidor, y sugiere el uso de mapas de imagen lado-cliente como alternativa. En HTML 4.01, los botones gráficos ahora pueden ser mapas de imagen lado-cliente. Para conservar la funcionalidad proporcionada por el servidor, los autores tienen las siguientes opciones, tal y como se describe en la Recomendación HTML 4.01 ([\[HTML4\]](#), sección 17.4.1):

*Si el servidor ejecuta acciones diferentes según la zona seleccionada con el ratón, los usuarios con navegadores no gráficos serán discriminados.
Por este motivo, los creadores de contenidos deben considerar otras técnicas:*

- El uso de botones de envío (`submit`) múltiples (cada uno con su propia imagen) en vez de un solo botón de envío (`submit`) gráfico. Los creadores de contenido pueden emplear hojas de estilo para controlar la ubicación de estos botones.
- Emplear un mapa de imagen lado cliente conjuntamente con `scripts`.

11.5 Técnicas para controles de formulario específicos

Puntos de verificación en esta sección:

- [10.4](#) Hasta que las aplicaciones de usuario manejen correctamente los controles vacíos, incluya caracteres imprimibles (rodeados de espacios), que no sirvan como vínculo, entre los vínculos contiguos. [Prioridad 3]

Ejemplo.

Algunas ayudas técnicas antiguas requieren un texto inicial en los controles de formulario tales como `TEXTAREA` para funcionar correctamente.

```
<FORM action="http://ejemplo.com/prog/text-read" method="post">
  <P>
    <TEXTAREA name=tunombre rows="20" cols="80">
      Teclee su nombre aquí.
    </TEXTAREA>
    <INPUT type="submit" value="Enviar"><INPUT type="reset">
  </P>
</FORM>
```

Proporcione un [equivalente textual](#) para las imágenes empleadas como botones "enviar":

Ejemplo.

```
<FORM action="http://ejemplo.com/prog/text-read" method="post">
  <P>
    <INPUT type="image" name=submit src="button.gif" alt="Enviar">
  </FORM>
```

Consulte también la sección sobre [acceso con teclado](#) puesto que es también de aplicación para controles de formulario.

11.6 Compatibilidad con versiones anteriores para formularios

En algunos navegadores HTML 3.2,

- El elemento `BUTTON` no aparece

- [INPUT](#) con type="button" se muestra como un campo de entrada de texto

12 Scripts

Esta sección trata sobre la accesibilidad de los scripts incluidos en un documento mediante el elemento [SCRIPT](#).

12.1 Transformación airosa de los scripts

Puntos de verificación en esta sección:

- [6.5](#) Asegúrese de que los contenidos dinámicos son accesibles o proporcione una página o presentación alternativa. [Prioridad 2]

Los diseñadores de contenido deben asegurarse de que las páginas sean accesibles con el soporte para scripts desactivado o con navegadores que no soporten scripts.

- Evite la creación de contenido sobre la marcha (on the fly) en el lado del cliente. Si el navegador del usuario no ejecuta scripts, el contenido no se generará o no se mostrará. Esto es distinto que mostrar o ocultar contenido ya existente mediante una combinación de scripts y hojas de estilo; si no se ejecuta el script, siempre se mostrará el contenido. Esto tampoco excluye la generación de páginas en el servidor para enviarlas al cliente.
- Evite crear vínculos que emplean "javascript" como URI. Si el usuario no ejecuta scripts, entonces no funcionarán los vínculos, puesto que el navegador no podrá crear el contenido del vínculo.

Ejemplo de uso desaconsejado. Este vínculo no llega a ningún sitio para un cliente que no ejecuta scripts o no los carga.

```
<A href="javascript:">...</A>
```

12.2 Scripts que causan destello

Puntos de verificación en esta sección:

- [7.1](#) Hasta que las aplicaciones de usuario permitan controlar los destellos, evite hacer destellar la pantalla. [Prioridad 1]

12.3 Scripts que causan movimiento y parpadeo

Puntos de verificación en esta sección:

- [7.2](#) Hasta que las aplicaciones de usuario permitan controlarlo, evite el parpadeo del contenido (por ejemplo, cambio de presentación en periodos regulares, así como encendido y apagado). [Prioridad 2]
- [7.3](#) Hasta que las aplicaciones de usuario permitan congelar el movimiento de los contenidos, evite los movimientos en las páginas. [Prioridad 2]

12.4 Scripts directamente accesibles

Puntos de verificación en esta sección:

- [6.4](#) Para los scripts y applets, asegúrese de que los manejadores de evento sean independientes del dispositivo de entrada. [Prioridad 2]
- [6.3](#) Asegúrese de que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, applets u otros objetos programados. Si esto no es posible, proporcione información equivalente en una página alternativa accesible. [Prioridad 1]
- [8.1](#) Haga los elementos de programación, tales como scripts y applets, directamente accesibles o compatibles con ayudas técnicas [Prioridad 1 Si la funcionalidad es importante y no se presenta en otro lugar; de otra manera, Prioridad 2.]
- [9.3](#) Para los scripts, especifique manejadores de evento lógicos en vez de manejadores de evento

dependientes de dispositivos. [Prioridad 2]

Un manejador de eventos es un script que se invoca cuando ocurre un evento concreto (por ejemplo, se mueve el ratón, se aprieta una tecla, se carga el documento, etc.). En HTML 4.01, los manejadores de eventos se asocian a los elementos mediante los [atributos de manejador de evento](#) (los atributos que comienzan con "on", por ejemplo en "onkeyup").

Algunos manejadores de eventos, cuando se invocan, causan efectos puramente decorativos tales como resaltar una imagen o cambiar el color del texto de un elemento. Otros manejadores de eventos causan efectos más importantes, tales como realizar un cálculo, proporcionar información importante al usuario, o enviar un formulario. Para manejadores de eventos que hacen más que cambiar la apariencia de un elemento, los desarrolladores de contenidos deben hacer lo siguiente:

1. Emplear disparadores al nivel de aplicación en vez de disparadores de nivel de interacción con el usuario. En HTML 4.01, los atributos al nivel de aplicación son "onfocus", "onblur" (lo opuesto a "onfocus"), y "onselect". Note que estos atributos son diseñados para ser independiente del dispositivo, pero en los navegadores actuales se implementan como eventos específicos del teclado.
2. Si tiene que emplear atributos específicos de un dispositivo, proporcione mecanismos redundantes de interacción (es decir, especifique dos manejadores para un mismo elemento):
 - Utilice "onmousedown" con "onkeydown".
 - Utilice "onmouseup" con "onkeyup".
 - Utilice "onclick" con "onkeypress".

Observe que en HTML 4.01 no existe equivalente de teclado para doble-click ("ondblclick").

3. No emplee manejadores de eventos que dependan de las coordenadas de ratón porque esto impide la interacción independiente del dispositivo.

12.5 Presentación alternativa de scripts

Puntos de verificación en esta sección:

- [1.1](#) Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). *Esto incluye:* imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos. [Prioridad 1]
- [6.2](#) Asegúrese de que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico. [Prioridad 1]

Una manera para conseguirlo es con el elemento [NOSCRIPT](#). El contenido de este elemento se muestra cuando los scripts no están habilitados.

Ejemplo.

```
<SCRIPT type="text/tcl">
...un script Tcl para mostrar un resumen de resultados deportivos...
</SCRIPT>
<NOSCRIPT>
  <P>Resultados de los partidos de ayer:</P>
  <DL>
    <DT>Bulls 91, Sonics 80.
    <DD><A href="bullsonic.html">Bulls vs. Sonics resumen del partido</A>
    ...más resultados...
  </DL>
</NOSCRIPT>
```

12.6 Actualizaciones de la página y ventanas nuevas

Puntos de verificación en esta sección:

- [7.4](#) Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener las actualizaciones, no cree páginas que se actualicen automáticamente de forma periódica. [Prioridad 2]
- [7.5](#) Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener el redireccionamiento automático, no utilice marcadores para redirigir las páginas automáticamente. En su lugar, configure el servidor para ejecute esta posibilidad. [Prioridad 2]
- [10.1](#) Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario. [Prioridad 2]

13 Índice de elementos y atributos HTML

Puntos de verificación en esta sección:

- [11.2](#) Proporcione vínculos redundantes en formato texto para cada zona activa de un mapa de imagen del servidor. [Prioridad 2]

Elementos

[Versión lineal del índice de elementos HTML 4.01.](#)

Este índice lista todos los elementos de HTML 4.01. La primera columna de esta tabla vincula a la definición del elemento en la especificación HTML 4.01 ([\[HTML4\]](#)). [[NdT](#)]. Los elementos que están desaconsejados en HTML 4.01 van seguidos de un asterisco (*). No aparecen en esta tabla los elementos que están obsoletos para HTML 4.01 o no existen en una especificación W3C de HTML (2.0, 3.2, 4.01).

La segunda columna indica otras especificaciones W3C de HTML que incluyen cada elemento. La tercera columna indica la función del elemento.

La última columna lista las secciones del presente documento en las que se trata el elemento. La entrada "N/A" significa que el elemento no se trata en este documento.

Nombre del elemento	Definido también en	Función	Técnicas
A	2.0, 3.2	Estructura	N/A
ABBR		Estructura	N/A
ACRONYM		Estructura	N/A
ADDRESS	2.0, 3.2	Metadato	N/A
APPLET*	3.2	Reemplazado	N/A
AREA	3.2	Estructura	N/A
B	2.0, 3.2	Presentación	N/A
BASE	2.0, 3.2	Procesamiento	N/A
BASEFONT*	3.2	Presentación	N/A
BDQ		Procesamiento	N/A
BIG	3.2	Presentación	N/A
BLOCKQUOTE	2.0, 3.2	Estructura	N/A
BODY	2.0, 3.2	Estructura	N/A
BR	2.0, 3.2	Presentación	N/A
BUTTON		Estructura	N/A
CAPTION	3.2	Estructura	N/A
CENTER*	3.2	Presentación	N/A
CITE	2.0, 3.2	Estructura	N/A
CODE	2.0, 3.2	Estructura	N/A
COL		Estructura	N/A
COLGROUP		Estructura	N/A

DD	2.0, 3.2	Estructura	N/A
DEL		Metadato	N/A
DEFN	3.2	Estructura	N/A
DIR*	2.0, 3.2	Estructura	N/A
DIV	3.2	Estructura	N/A
DL	2.0, 3.2	Estructura	N/A
DT	2.0, 3.2	Estructura	N/A
EM	2.0, 3.2	Estructura	N/A
FIELDSET		Estructura	N/A
FONT*	3.2	Presentación	N/A
FORM	2.0, 3.2	Estructura	N/A
FRAME		Reemplazado	N/A
FRAMESET		Presentación	N/A
H1	2.0, 3.2	Estructura	N/A
HEAD	2.0, 3.2	Estructura	N/A
HR	2.0, 3.2	Presentación	N/A
HTML	2.0, 3.2	Estructura	N/A
I	2.0, 3.2	Presentación	N/A
IFRAME		Reemplazado	N/A
IMG	2.0, 3.2	Reemplazado	N/A
INPUT	2.0, 3.2	Estructura	N/A
INS		Metadato	N/A
ISINDEX*	2.0, 3.2	Estructura	N/A
KBD	2.0, 3.2	Estructura	N/A
LABEL		Estructura	N/A
LEGEND		Estructura	N/A
LI	2.0, 3.2	Estructura	N/A
LINK	2.0, 3.2	Metadato	N/A
MAP	3.2	Estructura	N/A
MENU*	2.0, 3.2	Estructura	N/A
META	2.0, 3.2	Metadato	N/A
NOFRAMES		Alternativa	N/A
NOSCRIPT		Alternativa	N/A
OBJECT		Reemplazado	N/A
OL	2.0, 3.2	Estructura	N/A
OPTGROUP		Estructura	N/A
OPTION	2.0, 3.2	Estructura	N/A
P	2.0, 3.2	Estructura	N/A
PARAM	3.2	Procesamiento	N/A
PRE	2.0, 3.2	Presentación	N/A
Q		Estructura	N/A
S*		Presentación	N/A
SAMP	2.0, 3.2	Estructura	N/A
SCRIPT	3.2 (DTD)	Procesamiento	N/A
SELECT	2.0, 3.2	Estructura	N/A
SMALL	3.2	Presentación	N/A
SPAN		Estructura	N/A

STRIKE*	3.2	Presentación	N/A
STRONG	2.0, 3.2	Estructura	N/A
STYLE	3.2 (DTD)	Procesamiento	N/A
SUB	3.2	Presentación	N/A
SUP	3.2	Presentación	N/A
TABLE	3.2	Estructura	N/A
TBODY		Estructura	N/A
TD	3.2	Estructura	N/A
TEXTAREA	2.0, 3.2	Estructura	N/A
TFOOT		Estructura	N/A
TH	3.2	Estructura	N/A
THEAD		Estructura	N/A
TITLE	2.0, 3.2	Metadato	N/A
TR	3.2	Estructura	N/A
TT	2.0, 3.2	Presentación	N/A
U*	3.2	Presentación	N/A
UL	2.0, 3.2	Estructura	N/A
VAR	2.0, 3.2	Estructura	N/A

Atributos

[Versión lineal del índice de atributos de HTML 4.01.](#)

Este índice lista algunos atributos de HTML 4.01 que afectan a la accesibilidad y los elementos a los que se aplican. La primera columna de esta tabla vincula a la definición del atributo en la especificación HTML 4.01 ([\[HTML4\]](#)). Los atributos y elementos desaconsejados en HTML 4.01 ([\[HTML4\]](#)) van seguidos de un asterisco (*). Los atributos y elementos que están obsoletos para HTML 4.01 o no existen en una especificación HTML de la W3C (2.0, 3.2, 4.01) no aparecen en esta tabla. Los atributos que se aplican a la mayoría de los elementos de HTML 4.01 se indican como tales; consulte la especificación HTML 4.01 para la relación exacta de elementos con este atributo.

La segunda columna indica otras especificaciones de HTML que incluyen cada atributo. La tercera columna indica los elementos que toman este atributo. La cuarta columna indica la función del atributo.

La última columna lista las secciones en el presente documento donde se trata el atributo. La entrada "N/A" significa que el elemento no se trata en este documento.

Nombre del atributo	Aplicable a los elementos	Función	Técnicas
abbr	TD , TH	Alternativa	N/A
accesskey	A , AREA , BUTTON , INPUT , LABEL , LEGEND , TEXTAREA	Interfaz de usuario	N/A
alt	APPLET , AREA , IMG , INPUT	Alternativa	N/A
axis	TD , TH	Estructura	N/A
class	Mayoría de elementos	Estructura	N/A
dir	Mayoría de elementos	Procesamiento	N/A
for	LABEL	Estructura	N/A
headers	TD , TH	Estructura	N/A
hreflang	A , LINK	Metadato	N/A
id	Mayoría de elementos	Estructura	N/A
label	OPTION	Alternativa	N/A

lang	Mayoría de elementos	Metadato	N/A
longdesc	IMG , FRAME , IFRAME	Alternativa	N/A
scope	TD , TH	Estructura	N/A
style	Mayoría de elementos	Procesamiento	N/A
summary	TABLE	Alternativa	N/A
tabindex	A , AREA , BUTTON , INPUT , OBJECT , SELECT , TEXTAREA	Interfaz de usuario	N/A
title	Mayoría de elementos	Metadato	N/A
usemap	IMG , INPUT , OBJECT	Procesamiento	N/A

La siguiente es una lista de atributos HTML 4.01 no relacionados directamente con la accesibilidad. Los desarrolladores de contenido deberían usar hojas de estilo en lugar de atributos de presentación. Para los atributos de manejadores de eventos, consulte la sección "[manejadores de eventos independientes del dispositivo](#)" para más detalles.

Otros atributos estructurales:

start*, value*, rowspan, colspan, span

Otros atributos de presentación:

align*, valign*, clear*, nowrap*, char, charoff, hspace*, vspace*, cellpadding, cellspacing, compact*, face*, size*, background*, bgcolor*, color*, text*, link*, alink*, vlink*, border, noshade*, rules, size (deprecated according to element), marginheight, marginwidth, frame, frameborder, rows, cols

Otros atributos del proceso de instrucción:

ismap, coords, shape

Otros atributos de la interfaz de usuario:

target, scrolling, noresize

Otros atributos de metadatos:

type, cite, datetime

Atributos de los manejadores de eventos:

onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onload, onunload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onreset, onselect, onsubmit, onunload

14 Referencias

Para ver la última versión de cualquier especificación W3C, consulte la lista de [Informes Técnicos de W3C](#) en <http://www.w3.org/TR>.

[CSS1]

["CSS, level 1 Recommendation"](#), B. Bos, H. Wium Lie, editores, 17 de Diciembre de 1996, revisada el 11 enero de 1999. Esta Recomendación CSS1 es <http://www.w3.org/TR/1999/REC-CSS1-19990111>. La [última versión de CSS1](#) está disponible en <http://www.w3.org/TR/REC-CSS1>.

[CSS2]

["CSS, level 2 Recommendation"](#), B. Bos, H. Wium Lie, C. Lilley, and I. Jacobs, editores, 12 de mayo de 1998. Esta Recomendación CSS2 es <http://www.w3.org/TR/1998/REC-CSS2-19980512/>. La [última versión de CSS2](#) está disponible en <http://www.w3.org/TR/REC-CSS2>.

[HTML4]

["HTML 4.01 Recommendation"](#), D. Raggett, A. Le Hors, and I. Jacobs, editores, 24 de diciembre de 1999. Esta Recomendación HTML 4.01 es <http://www.w3.org/TR/1999/REC-html401-19991224/>.

[MATHML]

["Mathematical Markup Language"](#), P. Ion and R. Miner, editores, 7 de abril de 1998, revisada 7 de julio de 1999. Esta Recomendación MathML 1.0 es <http://www.w3.org/TR/1998/REC-MathML-19990707/>. La [última versión de MathML 1.0](#) está disponible en <http://www.w3.org/TR/REC-MathML>.

[WCAG10]

["Web Content Accessibility Guidelines 1.0"](#), W. Chisholm, G. Vanderheiden, y I. Jacobs, editores, 5 de mayo de 1999. Esta Recomendación WCAG 1.0 es <http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/>.

[WCAG10-CSS-TECHNIQUES]

["CSS Techniques for Web Content Accessibility Guidelines 1.0"](http://www.w3.org/TR/WCAG10-CSS-TECHS/), W. Chisholm, G. Vanderheiden, y I. Jacobs, editores. La última versión de este documento está disponible en <http://www.w3.org/TR/WCAG10-CSS-TECHS/>.

[WCAG10-TECHS]

["Techniques for Web Content Accessibility Guidelines 1.0"](http://www.w3.org/TR/WCAG10-CSS-TECHS/), W. Chisholm, G. Vanderheiden, I. Jacobs, editores. Este documento explica como implementar los puntos de verificación definidos en las "Pautas de Accesibilidad al Contenido en la Web 1.0". El ultimo borrador de las Técnicas está disponible en <http://www.w3.org/TR/WCAG10-TECHS/>.

15 Recursos

Nota: W3C no garantiza la estabilidad de cualquiera de las siguientes referencias, que están ajenas a su control. Estas referencias se incluyen para su comodidad. Cualquier referencia a un producto no es una recomendación de ese producto.

15.1 Otras pautas

[IBMJAVA]

[IBM Guidelines for Writing accesible Applications Using 100% Pure Java](#) proporcionadas por IBM Special Needs Systems.

[JAVAACCESS]

Información sobre [Java Accessibility and Usability](#), proporcionada por Trace R&D Center.

[MACROMEDIA]

[Flash OBJECT and EMBED Tag Syntax](#) de Macromedia.

[UWSAG]

["The Unified Web Site Accessibility Guidelines"](#), G. Vanderheiden, W. Chisholm, editores. Estas Pautas Unificadas de Accesibilidad para Sitios Web han sido compiladas por el [Trace R & D Center](#) de la Universidad de Wisconsin con fondos del National Institute on Disability and Rehabilitation Research (NIDRR), del Departamento de Educación de los Estados Unidos.

15.2 Las aplicaciones de usuario y otras herramientas

El sitio web de la WAI mantiene una lista de [navegadores Web alternativos](#) (ayudas técnicas y otras aplicaciones de usuario diseñadas para la accesibilidad).

[ASTER]

Para información sobre ASTER, un "Sistema Sonoro para Lecturas Técnicas", consúlte [la página de T. V. Raman](#).

[HYPERMEDIA]

[El navegador de hipermedios Techexplorer de IBM](#).

15.3 Recursos de accesibilidad

[TRACE]

El [Centro de I+D de Trace](#). Consulte este sitio para obtener una variada información sobre la accesibilidad, que incluye un [applet Java con barras de desplazamiento que pueden ser inmovilizadas por el usuario](#).

[WAI-ER]

El [Grupo de trabajo sobre Evaluación y Reparación de WAI](#)

16 Agradecimientos

Co-directores del Grupo de Trabajo de las Pautas de Contenido en la Web:

[Jason White](#), Universidad de Melbourne

[Gregg Vanderheiden](#), Trace Research and Development

Persona de contacto del equipo W3C:

[Wendy Chisholm](#)

Nuestro agradecimiento a las siguientes personas que han contribuido con su tiempo y sus valiosos comentarios a dar forma a estas pautas:

Harvey Bingham, Kevin Carey, Chetz Colwell, Neal Ewers, Geoff Freed, Al Gilman, Larry Goldberg, Jon Gunderson, Eric Hansen, Phill Jenkins, Leonard Kasday, George Kerscher, Marja-Riitta Koivunen, Josh Krieger, Chuck Letourneau, Scott Luebking, William Loughborough, Murray

Maloney, Charles McCathieNevile, MegaZone (Livingston Enterprises), Masafumi Nakane, Mark Novak, Charles Oppermann, Mike Paciello, David Pawson, Michael Pieper, Greg Rosmaita, Liam Quinn, Dave Raggett, T.V. Raman, Robert Savellis, Jutta Treviranus, Steve Tyler, and Jaap van Lelieveld



[\[contents\]](#) [\[html index\]](#)