

ACCESIBILIDAD WEB



Accesibilidad y Usabilidad Web

Este curso se basa en trabajo del [W3C](#), principalmente el [Curriculum de WCAG 1.0](#) y en las [Pautas de Accesibilidad al Contenido Web](#), y se encuentra por tanto sujeto al siguiente aviso de copyright:

[Copyright](#) ©1999, 2000 [W3C](#) ® ([MIT](#), [ERCIM](#), [Keio](#)), Todos los derechos reservados. Son aplicables las reglas de [responsabilidad](#), [marcas registradas](#), [utilización de documentos](#) y [licencias de software](#).

Índice

ÍNDICE	3
INTRODUCCIÓN A LA ACCESIBILIDAD WEB	5
1 ¿QUÉ ES LA ACCESIBILIDAD WEB?.....	6
2 DISEÑO PARA TODOS-AS	8
3 ORGANISMOS	9
4 TIPOS DE DISCAPACIDAD	13
5 AYUDAS TÉCNICAS.....	20
6 VENTAJAS DE LA ACCESIBILIDAD WEB	28
7 LEGISLACIÓN.....	29
8 BIBLIOGRAFÍA.....	32
INTRODUCCIÓN A LAS PAUTAS DE ACCESIBILIDAD AL CONTENIDO WEB	33
1 ORGANIZACIÓN DE LAS PAUTAS	34
2 PRIORIDADES Y NIVELES DE ADECUACIÓN	35
3 DOCUMENTOS COMPLEMENTARIOS	36
4 ORGANIZACIÓN DEL CURSO	37
IMÁGENES Y MULTIMEDIA	38
1 IMÁGENES	39
2 MAPAS DE IMAGEN	46
3 MULTIMEDIA	51
OBJETOS PROGRAMADOS.....	54
1 EQUIVALENTES ACCESIBLES	55
2 CONTENIDO DINÁMICO	59
3 INDEPENDENCIA DE DISPOSITIVO	62
MARCOS.....	66
1 EQUIVALENTES TEXTUALES	67
TABLAS.....	72
1 TABLAS DE DATOS	73
2 TABLAS DE MAQUETACIÓN	86
FORMULARIOS.....	91
1 JAVASCRIPT Y FORMULARIOS	92
2 ASOCIACIÓN ENTRE ETIQUETAS Y CONTROLES.....	99
3 INDEPENDENCIA DE DISPOSITIVO	102
4 OTROS ASPECTOS SOBRE ACCESIBILIDAD	106
ESTÁNDARES.....	109
1 GRAMÁTICAS FORMALES	110
2 TECNOLOGÍAS W3C	114
3 ELEMENTOS DESACONSEJADOS.....	116
4 PÁGINA ALTERNATIVA ACCESIBLE.....	118
ESTRUCTURA Y PRESENTACIÓN.....	120
1 HOJAS DE ESTILO.....	121
2 COLOR Y CONTRASTE	130

3	ENCABEZAMIENTOS	132
4	LISTAS	134
5	CITAS.....	145
6	DIVIDIR LA INFORMACIÓN.....	147
7	MARCADORES MEJOR QUE IMÁGENES	148
NAVEGACIÓN, LENGUAJE Y COMPRENSIÓN.....		151
1	LENGUAJE Y COMPRENSIÓN	152
2	ACTUALIZACIONES AUTOMÁTICAS Y REDIRECCIONAMIENTO	159
3	ENLACES.....	161
4	MECANISMOS DE NAVEGACIÓN	167
5	METADATOS	171
6	INDEPENDENCIA DE DISPOSITIVO	180
HERRAMIENTAS DE EVALUACIÓN Y REPARACIÓN.....		185
1	ANÁLISIS PREVIO	186
2	EVALUACIÓN AUTOMÁTICA.....	188
3	EVALUACIÓN MANUAL.....	198
4	HERRAMIENTAS QUE FACILITAN LA EVALUACIÓN MANUAL.....	205
5	PRUEBAS DE USUARIO-A	210
6	MONITORIZACIÓN CONTINUADA.....	211
7	HERRAMIENTAS DE REPARACIÓN	212
8	BIBLIOGRAFÍA	215

Introducción a la Accesibilidad Web

El poder de la Web está en su universalidad. Un acceso a la Web para todos independientemente de su discapacidad es un aspecto esencial.

Tim Berners-Lee

1 ¿Qué es la accesibilidad Web?

La Web (World Wide Web) se creó como una red universal de conocimiento que ha supuesto un salto cualitativo y cuantitativo en cuanto a la adquisición y tratamiento de información.

Sin embargo, y debido a diferentes motivos, se ha convertido también en una barrera para una parte numerosa de la ciudadanía, sobre todo para las personas con discapacidad.

La **accesibilidad** es la posibilidad de que un sitio o servicio Web pueda ser visitado y utilizado de forma satisfactoria por el mayor número posible de personas, independientemente de sus propias limitaciones o de las derivadas de su entorno.

Del mismo modo que existen barreras arquitectónicas, también existen barreras en la Web que impiden o entorpecen el acceso a la misma a las personas con discapacidad.

En la sociedad de la información se hace imprescindible eliminar estas barreras para favorecer así la igualdad de oportunidades; si no se eliminan, se está impidiendo a las personas con discapacidad el acceso a la información y servicios electrónicos.

La accesibilidad web beneficia a los siguientes colectivos:

- **Personas mayores**
- **Personas con discapacidad:**
 - Físicas, psíquicas, sensoriales
 - Cognitivas y de lenguaje

También existen otras dificultades de acceso a la Web, como son las derivadas del **entorno** (ruido, mala iluminación, deficiencia tecnológica).

Las principales barreras que encuentran los usuarios-as para acceder a los sitios web son:

- **Contenidos visuales**, como imágenes, sin texto alternativo.
- **Elementos multimedia** sin subtítular o sin describir su contenido visual.
- **Mapas de imagen** sin texto alternativo para las diferentes zonas activas del mismo.
- Uso inadecuado de **elementos estructurales** o con fines sólo de presentación.
- **Tablas** usadas para maquetación y difíciles de interpretar cuando se interpretan por tecnologías asistivas.
- Falta de **contenido alternativo** para los marcos (frames), scripts, u otros elementos programados, impidiendo la navegación a usuarios-as cuyo navegador no los soporte.

La accesibilidad Web se entiende desde tres **áreas** bien diferenciadas:

Accesibilidad y Usabilidad Web

- **Sitios Web y aplicaciones.** Relativo a la accesibilidad del **contenido** de los sitios Web, permite que las personas con discapacidad interactúen adecuadamente con la Web.
- **Aplicaciones de usuario-a.** Estudia cómo mejorar la accesibilidad en las aplicaciones software que se emplean para acceder al contenido de la Web, como navegadores y reproductores multimedia, para que las personas con discapacidad puedan utilizarlos correctamente.
- **Herramientas de autor.** Sobre las aplicaciones usadas por los desarrolladores-as para la creación de las páginas Web se trabaja en dos líneas:
 - Que las herramientas de autor produzcan contenido Web accesible.
 - Que puedan ser utilizadas por personas con discapacidad.

2 Diseño para todos-as

La accesibilidad se sustenta en una serie de premisas llamadas *principios del diseño para todos-as* o **diseño universal**. Tienen como objetivo el diseño de productos y entornos de fácil uso para el mayor número posible de personas, sin la necesidad de adaptarlos o rediseñarlos de una forma especial.

Estos siete principios son:

Igualdad de uso

El diseño debe ser fácil de usar y adecuado para todas las personas independientemente de sus capacidades y habilidades. Debe proporcionar las mismas maneras de uso para todos los usuarios-as: idénticas cuando es posible, equivalentes cuando no lo sea.

Flexibilidad

El diseño debe poder adecuarse a un amplio rango de preferencias y habilidades individuales. Por ejemplo, permitiendo al usuario elegir el mecanismo de interacción o adaptándose al ritmo de uso.

Simple e intuitivo

El diseño debe ser fácil de entender, independientemente de la experiencia, conocimientos, habilidades o el nivel de concentración del usuario-a. Esto elimina la complejidad innecesaria así como prioriza la entrega de información acorde a su importancia.

Información fácil de percibir

El diseño debe ser capaz de intercambiar información con el usuario-a, independientemente de sus capacidades sensoriales o de las condiciones ambientales. Algunos ejemplos de esto es la presentación por medios redundantes (texto, voz...), la mejora de la legibilidad de la información esencial, la compatibilidad con las ayudas técnicas, etc.

Tolerante a errores

El diseño debe minimizar las acciones accidentales o fortuitas que puedan tener consecuencias fatales o no deseadas. Proactividad en cuanto a los posibles errores que el usuario pueda cometer en su interacción con la web procurando minimizarlos en diseño.

Escaso esfuerzo físico

El diseño debe poder ser usado eficazmente y con el mínimo esfuerzo posible. Cuando hablamos de este concepto, intentamos evitar las acciones repetitivas, por ejemplo.

Dimensiones apropiadas

Los tamaños y espacios deben ser apropiados para la manipulación y uso por parte del usuario-a, independientemente de su tamaño, posición, y movilidad.

3 Organismos

La accesibilidad Web surge como una iniciativa dentro del **World Wide Web Consortium** (W3C).

El **W3C**, a través de su iniciativa **WAI** (Web Accessibility Initiative), busca que la Web sea accesible para todos-as independientemente de su cultura, idioma, recursos materiales y discapacidad de los usuarios-as.

3.1.W3C



El [W3C](#) fue creado en octubre de 1994 para guiar la Web a su máximo potencial mediante el desarrollo de estándares y protocolos comunes que promoviesen su evolución y asegurasen su interoperabilidad.

La Interoperabilidad es la capacidad de un sistema o producto de trabajar en conjunción con otros sistemas o productos sin un esfuerzo especial por parte del usuario. Los usuarios esperan poder acceder al contenido de la Web con el software de su elección (navegador, lectores de pantalla, teléfonos móviles, etc.). El W3C promueve la interoperabilidad desarrollando estándares y protocolos abiertos (no privados).

El W3C está formado por casi 400 [organizaciones miembro](#) con el objetivo de dirigir el desarrollo de la futura World Wide Web.

Tim Berners-Lee, inventor de la Web y **Premio Príncipe de Asturias de Investigación 2002**, fundó el W3C en el Instituto de Tecnología de Massachusetts. El W3C se creó en colaboración con el CERN (Laboratorio Europeo de Partículas Físicas) y con el apoyo de DARPA (Agencia Norteamericana de Investigación Avanzada en Proyectos de Defensa) y de la Comisión Europea. La primera sede europea del W3C se creó en 1995 en el INRIA (Instituto Nacional de Investigación en Informática y Automática), siendo sustituida en 2003 por ERCIM (Consortio Europeo para la Investigación en Informática y Matemáticas).

Los principales [objetivos del W3C](#) son:

Acceso universal

Hacer la Web accesible a todos promocionando tecnologías que tengan en cuenta las enormes diferencias en cultura, idiomas, educación, capacidad, recursos materiales, dispositivos de acceso y limitaciones físicas de los usuarios en todos los continentes.

Web Semántica

Desarrollar un entorno software que permita a cada usuario-a hacer el mejor uso de los recursos disponibles en la Web.

Web de confianza

Guiar el desarrollo de la Web con cuidadosa atención ante los nuevos aspectos legales, comerciales y sociales generados por esta tecnología.

Para cumplir con sus objetivos, el W3C se apoya en **recomendaciones o estándares**; hasta el momento ha desarrollado más de 40. Entre ellas destacan las siguientes:

HTML

Es el lenguaje utilizado para representar documentos en la Web. Es un lenguaje de etiquetado (HyperText Markup Lenguaje) para la creación de hipertextos. Por hipertexto entendemos texto con inclusión de elementos multimedia e hiperenlaces que permiten relacionar el documento actual con otros documentos. La última especificación es [HTML 4.01](#) de diciembre de 1999.

XHTML

La primera versión ([XHTML 1.0](#)), de enero de 2000, ofrece la semántica de HTML 4.01 utilizando la sintaxis de XML. [XHTML 1.1](#)., la versión modularizada de XHTML, fue publicada en mayo de 2001.

CSS

Las hojas de estilo en cascada definen una serie de instrucciones para controlar la apariencia, presentación y estilo de los elementos de un documento (X)HTML o XML, facilitando la administración del sitio y promoviendo la accesibilidad Web al permitir la separación de contenido y presentación. CSS1 fue publicada en diciembre de 1996 y [CSS2](#) en mayo de 1998.

XML 1.0

[XML](#) (eXtensible Markup Language o Lenguaje Extensible de Marcas), es una especificación para la creación de lenguajes de marcas usados para guardar datos en documentos de texto y poder transmitirlos y procesarlos automáticamente. Se publicó en febrero de 1998.

Por otra parte, dentro de la iniciativa [WAI](#) (Web Accessibility Initiative) se publicaron en mayo de 1999 las [Directrices de Accesibilidad para el Contenido Web 1.0](#), que explican cómo deben crearse páginas y sitios Web accesibles y que son el objeto de estudio del presente curso.

El W3C colabora con organizaciones locales de diferentes países para albergar sus oficinas, las cuales actúan como puntos de contacto locales y aseguran que el W3C y sus especificaciones son conocidos en su zona.

En Octubre de 2003 se inauguró la [oficina española del W3C](#) ubicada en el Parque Científico y Tecnológico de Gijón. La oficina colabora en la promoción de los idiomas locales y en la ampliación de la base geográfica del W3C. La oficina está hospedada por la [Fundación CTIC](#) (Centro Tecnológico para la Información y la Comunicación) cuya misión es la difusión y desarrollo de la Sociedad de la Información y las TIC. Gracias a la creación de la oficina española ha aumentado la representación de la comunidad española en el W3C (Ver [nota de prensa](#)).

3.2.WAI



El compromiso del W3C de guiar la Web a su máximo potencial incluye alcanzar un alto grado de funcionalidad para personas con discapacidades. De ahí que desarrollase la **Iniciativa de Accesibilidad Web** ([WAI](#))

La Iniciativa de Accesibilidad Web pretende alcanzar la accesibilidad de la web a través de cinco **áreas principales de trabajo**:

- Tecnología.
- Directrices.
- Herramientas.
- Educación y difusión.
- Investigación y desarrollo.

El WAI colabora con organizaciones de todo el mundo de la industria, organizaciones de discapacitados, gobiernos, escuelas, universidades, etc.

La **actividad técnica** del WAI se divide en varios **grupos de trabajo** cuyo objetivo es aumentar la accesibilidad de la Web para las personas con discapacidad:

- **Grupo de Trabajo de Protocolos y Formatos** ([PFWG](#)): asegura que las tecnologías del W3C soporten la accesibilidad. El principal objetivo de este grupo es la colaboración con otras áreas del W3C que desarrollan otras especificaciones, para asegurar que su trabajo soporte la accesibilidad. Este grupo revisa las especificaciones, generalmente en el estadio de borrador de trabajo, y genera una declaración de requerimientos.
- Varios grupos de trabajo para el desarrollo de directrices o pautas de accesibilidad Web, agentes de usuario, herramientas de autor y XML. De estos equipos se derivan las:
 - **Directrices de Accesibilidad para el Contenido Web 1.0** ([WCAG 1.0](#)): normas para hacer el contenido de los sitios Web accesible para las personas con discapacidades.
 - **Directrices de Accesibilidad para Herramientas de Autor 1.0** ([ATAG 1.0](#)): directrices para los desarrolladores-as de software para hacer que las herramientas y aplicaciones, y el contenido que generen, sean más accesible para las personas con discapacidades.
 - **Directrices de Accesibilidad para Agentes de Usuario 1.0** ([UAAG 1.0](#)): indican a los programadores y diseñadores de herramientas de usuario cómo hacer sus productos más accesibles para las personas con discapacidad y para aumentar la usabilidad para todas las personas en general.
 - **Directrices de Accesibilidad para XML** ([XAG](#)): pautas para el diseño de aplicaciones XML que eliminen las barreras de accesibilidad para las personas con discapacidades.

Accesibilidad y Usabilidad Web

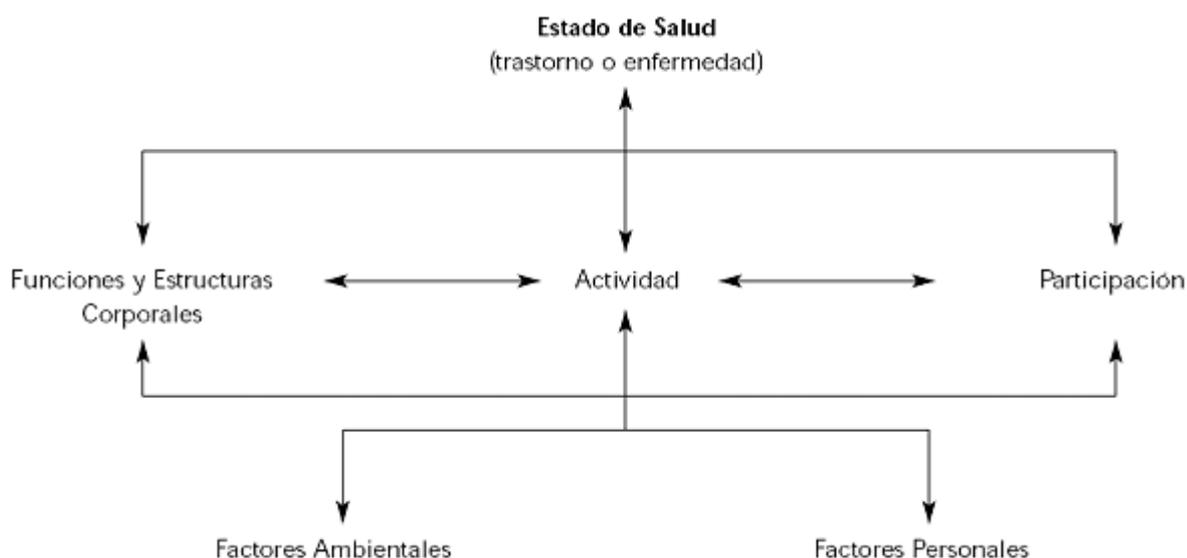
- **Grupo de Trabajo de Herramientas de Evaluación y Reparación (ERTWG)**: desarrolla técnicas y herramientas para la evaluación de la accesibilidad de los sitios Web e insta a los sitios Web a ser más accesibles. También se dedica a las Herramientas de Reparación de la accesibilidad. Actualmente, su principal trabajo es el desarrollo del Lenguaje para Evaluación e Informe ([EARL](#), **Evaluation And Report Language**).

Además de la actividad técnica, el WAI también realiza labores de formación y difusión para promover la accesibilidad Web así como en investigación y desarrollo.

4 Tipos de discapacidad

La **OMS (Organización Mundial de la Salud)** cuenta con un grupo de trabajo dedicado a la clasificación y evaluación de conceptos relacionados con la salud. En mayo de 2001 la OMS elabora la Clasificación Internacional de Funcionamiento, de la Discapacidad y de la Salud (**CIF**).

La **CIF** tiene como objetivo "*proporcionar un lenguaje unificado y estandarizado que sirva como punto de referencia para la descripción de la salud y los estados relacionados con la salud*". En esta clasificación se establece que la **discapacidad** "*engloba las deficiencias, las limitaciones en la actividad y las restricciones en la participación*".



La **salud en forma positiva** está descrita por las **funciones y estructuras corporales**, la **actividad** y la **participación**.

La **salud en su forma negativa** se describe por las **deficiencias**, las **limitaciones en la actividad** y las **restricciones en la participación**.

4.1. Tipos de discapacidad que afectan a la accesibilidad Web

En el ámbito de la **accesibilidad Web**, el término **discapacidad** se utiliza en un sentido muy amplio. Este concepto engloba las discapacidades relacionadas con la salud, tanto crónicas como transitorias, las relacionadas con el envejecimiento o las derivadas de una situación desfavorable determinada.

La accesibilidad Web también beneficia a las personas con limitaciones derivadas del entorno. Por ejemplo, una persona que acceda a la Web desde un monitor de pantalla pequeña o monocromo, con un navegador sólo de texto o que no disponga de ratón para navegar.

A continuación, se realiza una descripción de las distintas clases generales de discapacidad o limitaciones que pueden afectar el acceso a la Web.

4.1.1. Discapacidades visuales

Este tipo de discapacidades afectan a la percepción de estímulos visuales.

1. Ceguera

El concepto de ceguera tiene sobre todo una finalidad legal y social ya que muy pocas personas son totalmente ciegas.

Legalmente, se considera que una persona es ciega cuando la visión es inferior a 20/200 (considerado sobre el mejor ojo y con la mejor corrección) o con un campo visual inferior a 20° sea cual sea su nivel de visión.

Según la **OMS**, se considera ceguera aquella visión inferior a 20/400.

Algunas de las **barreras** que tienen las personas ciegas al acceder al contenido de la Web son:

- Imágenes sin un **texto alternativo** que describa su contenido.
- Imágenes complejas, como gráficos de barras o estadísticas, sin una **descripción detallada**.
- **Elementos multimedia** (vídeos, animaciones...) sin descripción textual o sonora.
- **Tablas** cuyo contenido resulta incomprensible cuando se leen de forma secuencial (celda a celda según el orden en que aparecen en el código HTML o en líneas completas según se presentan en la pantalla).
- **Marcos** (frames) sin un contenido alternativo adecuado para los navegadores que no los soporten.
- Falta de independencia de dispositivo que no permite usar correctamente la Web con dispositivos de entrada diferentes del ratón (por ejemplo, el teclado). El ratón es un dispositivo de apuntamiento imposible de usar por las personas que no pueden ver dónde está el cursor.
- **Formatos no estándar** de documentos que pueden ser problemáticos para un lector de pantalla.

Las personas ciegas usan diferentes **ayudas técnicas** para acceder a la Web. Estas ayudas técnicas son dispositivos hardware y/o software para facilitar el acceso al contenido de las páginas Web.

A continuación se detallan algunas de las ayudas técnicas y estrategias empleadas por las personas afectadas por este tipo de discapacidad.

- **Lectores de pantalla**: aplicaciones que leen el texto de la pantalla y lo transmiten mediante un sintetizador de voz (vía auditiva) o bien lo envían a una línea braille para que el usuario-a lo pueda leer con los dedos.
- **Navegadores de voz**: navegadores que leen el contenido de las páginas Web con un sintetizador de voz incorporado. Son una alternativa a la combinación de un navegador convencional y un lector de pantalla.
- **Teclado como dispositivo de entrada**: mediante el teclado utilizan el tabulador para recorrer los enlaces y controles de formulario o las teclas de

acceso rápido para ir a una zona determinada del contenido directamente, en vez de secuencialmente.

En el apartado siguiente (*5. Ayudas Técnicas*) se describirán con más detalle éstas y otras ayudas técnicas.

2. Baja Visión

Esta discapacidad supone una incapacidad en la función visual, incluso con las mejores medidas o tratamientos correctivos. Sin embargo, las personas con baja visión tienen una capacidad visual suficiente para la planificación y ejecución de diferentes tareas.

Existen muchos tipos de baja visión como la falta de agudeza visual (visión poco nítida), distrofia macular (solo se ve una parte reducida del campo visual) o la visión borrosa.

Las personas con **baja visión** encuentran las siguientes **barreras** en el acceso a la Web:

- **Tamaño de letra** con medidas absolutas que no permiten redefinirlo.
- **Diseño de páginas** que, al modificar el tamaño de fuente, estropea la maquetación y hace difícil la navegación.
- **Poco contraste** en las imágenes o texto que no se puede cambiar fácilmente usando una hoja de estilos de usuario.
- **Texto añadido mediante imágenes** en vez de directamente que dificulta aumentar el tamaño para facilitar su lectura.

Algunas de las barreras explicadas para los ciegos también lo son para las personas con baja visión.

Las **ayudas técnicas y estrategias** que suelen emplear son:

- **Pantallas grandes.**
- **Amplificadores de pantalla.**
- **Combinaciones específicas** de colores de texto y fondo.
- **Tipos de letra** más legibles

3. Daltonismo

El **daltonismo** es una alteración congénita que dificulta la percepción de ciertos colores. Las formas más comunes de daltonismo son la dificultad para distinguir entre rojo y verde o entre amarillo y azul.

Las barreras que las personas con daltonismo pueden encontrar en la Web son:

- Uso del color para resaltar texto sin usar otro **elemento de formato adicional** (como cursiva, negrita o subrayado).
- **Poco contraste** en las imágenes o entre el texto y el color de fondo de la página.

- Navegadores que no soportan el uso de **hojas de estilo** definidas por el usuario-a.

Las personas con daltonismo pueden emplear sus propias **hojas de estilo** para modificar los colores de las fuentes y del fondo de las páginas, en vez de usar los definidos por el diseñador-a.

4.1.2. Discapacidades auditivas

Las discapacidades auditivas son las alteraciones cuantitativas de la correcta percepción de los sonidos. Así, se puede distinguir entre una pérdida total de la capacidad auditiva y una disminución más o menos relevante de la misma.

Sordera o cofosis

La sordera es la **pérdida total de la audición** que imposibilita la percepción del lenguaje por vía auditiva, teniendo que acceder a esa información a través de la vista. El idioma principal de algunas personas con sordera es el lenguaje de signos y es posible que no hablen o lean con fluidez otro idioma.

Hipoacusia

La hipoacusia es aquella **disminución de la capacidad auditiva** que aún permite recibir el lenguaje oral por vía auditiva. También se puede denominar *sordera ligera o moderada*.

Las **barreras** que encuentran las personas con **sordera** o **hipoacusia** son:

- Falta de **subtítulos** o de **transcripciones** de los contenidos sonoros.
- Falta de **imágenes** que ayuden la comprensión del contenido de las páginas. Las páginas con demasiado texto y sin imágenes pueden entorpecer la comprensión a personas cuyo idioma principal es una lengua de signos en lugar del lenguaje escrito o hablado.
- Necesidad de **entrada de voz** en algunos sitios Web.

4.1.3. Discapacidades motrices

Las **discapacidades motrices** son aquellas que afectan a la correcta movilidad de las personas. Algunas de las enfermedades que afectan al aparato locomotor son:

Distrofia muscular

Grupo de enfermedades hereditarias caracterizadas por provocar una debilidad progresiva y un deterioro de los músculos esqueléticos, o voluntarios, que controlan el movimiento.

Distonía muscular

Contracciones involuntarias permanentes de los músculos de una o más partes del cuerpo debido a una disfunción del sistema nervioso. Puede ser hereditaria o adquirida.

Enfermedad de Parkinson

Enfermedad degenerativa crónica que se caracteriza por los siguientes síntomas: temblores (en manos, brazos, piernas, mandíbula y cara), rigidez en

las extremidades y el tronco, lentitud de movimientos e inestabilidad postural. Generalmente no suele ser hereditaria.

Algunas de las **barreras** que afectan a personas con discapacidad motriz son:

- Iconos, botones, enlaces y otros **elementos de interacción demasiado pequeños** que dificultan su uso a personas con poca destreza en sus movimientos.
- **Falta de independencia de dispositivo** que no permite usar correctamente la Web con el teclado en vez de con el ratón.
- **Tiempos de respuesta limitados** para interactuar con la página.

Las personas con este tipo de discapacidades necesitarán diferentes y variados **dispositivos de entrada**, según cual sea su discapacidad concreta:

- **Ratones especiales.**
- **Teclados alternativos** con una disposición de teclas adecuada a la capacidad de movimiento de las manos.
- **Ratón de cabeza, licornio o apuntador de boca.**
- Software de **reconocimiento de voz.**
- Sistemas de **seguimiento de ojos.**

4.1.4. Discapacidades neurológicas y cognitivas

Las **discapacidades neurológicas** y cognitivas más frecuentes son las siguientes:

- Relacionadas con el **aprendizaje**: dislexia, discalculia...
- Transtornos de **déficit de atención.**
- Dificultad para comprender **conceptos complejos.**
- Falta de **memoria.**
- **Transtornos emocionales** que dificultan la concentración.
- **Epilepsia.**

Algunas de las **barreras** que encuentran las personas con discapacidades cognitivas y neurológicas son:

- Falta de acceso a la información de forma alternativa (como por ejemplo, un texto que pueda convertir la información visual en sonido).
- Elementos visuales o sonoros que no se puedan desactivar cuando se desee y que puedan distraer a las personas con déficit de atención.
- Falta de una organización clara y coherente de la información que ayude a las personas con problemas de memoria o con escasa capacidad cognitiva.
- Lenguaje complejo.
- Ausencia de gráficos en los sitios Web que complementen la información textual.
- Tamaño de letra fijo que no se puede aumentar.

- Destellos o parpadeos con altas frecuencias que pueden provocar ataques de epilepsia.

Algunas de las **ayudas técnicas** utilizadas por las personas con discapacidades neurológicas o cognitivas son:

- Lector de pantalla para facilitar la comprensión a las personas con dificultades para la lectura.
- Subtítulos que faciliten la comprensión de un contenido sonoro a las personas con dificultades para procesar la información auditiva.
- Desactivación de los elementos multimedia (animaciones gráficos y sonido) para concentrarse en el contenido.
- Los enfermos de epilepsia pueden desactivar animaciones, sonidos y todos los objetos que puedan provocar ataques.

Por norma general, este tipo de usuarios-as no usan ayudas técnicas para acceder a la información, sino que dependen sobre todo del diseño del sitio Web.

4.1.5. Discapacidades relacionadas con el envejecimiento

El envejecimiento lleva asociado una **pérdida paulatina de capacidades** que puede traducirse en una disminución de la vista, de la capacidad auditiva, la memoria, coordinación y destreza física. Por tanto, este tipo de usuarios-as no se pueden clasificar dentro de un único tipo de discapacitados-as, pues pueden pertenecer a varios grupos al tener **múltiples limitaciones**.

Cualquiera de las barreras explicadas en las discapacidades anteriores puede afectar a una persona mayor.

4.1.6. Limitaciones derivadas del entorno

Las limitaciones derivadas del entorno no pueden considerarse discapacidades en sentido estricto, sino que son condiciones del entorno que **restringen las posibilidades en el acceso** a Internet.

Los usuarios-as con este tipo de limitaciones utilizan métodos alternativos de interacción con el equipo informático que dificultan el acceso a la Web encontrándose con barreras similares a las explicadas anteriormente para algunas discapacidades.

Algunas **limitaciones** derivadas del entorno son:

- **Navegadores antiguos** con escaso soporte para nuevas tecnologías. Se deberían proporcionar contenidos alternativos para este tipo de elementos.
- **Navegadores de texto** usados en ordenadores antiguos o en terminales con una limitada capacidad gráfica. Debe haber un equivalente textual para todos los elementos no textuales (imágenes, contenido visual y sonoro, etc.).
- **Conexiones lentas** que hace que algunos usuarios-as desactiven las imágenes y elementos multimedia para reducir el tiempo de descarga de las páginas Web.
- **Pantallas pequeñas**, que dificultan la visualización de páginas Web diseñadas para resoluciones mayores.

Accesibilidad y Usabilidad Web

- **Monitores monocromos** o en blanco y negro que impiden percibir la información que se sustenta sólo en el color.
- **Entornos de trabajo** que no permiten la percepción del contenido sonoro de las páginas Web (elevado nivel de ruido ambiental, etc). Para solventar esta limitación es necesario que se proporcionen transcripciones o subtítulos.
- Ambientes con **mala iluminación** o escasas condiciones de visibilidad que requieren aumentar el tamaño de la letra, el zoom, el contraste o cambiar los estilos de las páginas Web.
- **Ausencia de ratón** para usar el ordenador, teniendo que usar el teclado. Las páginas Web deberían estar diseñadas para permitir la independencia de dispositivo.

5 Ayudas técnicas



Las **ayudas técnicas** son dispositivos diseñados para ayudar a las personas con discapacidad a realizar las tareas comunes de la actividad diaria que de otra manera no podrían realizar fácilmente. Así, una silla de ruedas, una máquina lectora o un audífono son ayudas técnicas.

Dentro del mundo de la informática, las ayudas técnicas son aplicaciones software, dispositivos hardware o una combinación de ambos.

A continuación se presenta una breve descripción de algunas de las ayudas técnicas empleadas por las personas con discapacidad para el uso de los ordenadores y, en particular, para el acceso al contenido de la Web.

5.1. Braille y líneas braille

El **Braille** es un código de lectura y escritura inventado por el profesor francés Louis Braille en el siglo XIX para que las personas invidentes tuviesen acceso a la información a través de las puntas de los dedos.

El código Braille original se basa en una celda, denominada "**celda Braille**", formada por una matriz de seis puntos de relieve para representar todas las letras, números y caracteres de puntuación. El número total de caracteres que se pueden conseguir combinando los seis puntos es de 64.

Este número era insuficiente para el uso de ordenadores y por eso surgió el "**Braille computerizado**" que añade dos puntos más a los seis del código "**Braille integral**" (o clásico).

Con un código Braille de 8 puntos se pueden representar hasta 256 caracteres y se consigue una compatibilidad mayor con los sistemas informáticos (permite representar todo el código ASCII).

Las **líneas Braille** son dispositivos basados en mecanismos electromagnéticos que permiten la presentación de caracteres braille mediante puntos o clavijas que suben y bajan dinámicamente a través de unos agujeros realizados sobre una superficie plana.

En la siguiente imagen podemos ver un ejemplo de línea Braille.



5.2. Lectores de pantalla

Los **lectores de pantalla** se usan para permitir a las personas con discapacidad visual la navegación y la interacción con el sistema operativo y sus aplicaciones, incluyendo los navegadores Web.

Este tipo de software interpreta el texto que aparece en la pantalla y lo dirige a un dispositivo de salida en un formato que puede ser entendido por una persona ciega (por ejemplo, información sonora mediante síntesis de voz o información táctil por medio de una línea de Braille).

Con respecto al acceso a páginas Web, los **lectores de pantalla modernos** se basan en la estructura del documento (el código analizado del documento) para leer la pantalla, con lo que pueden seguir la estructura lógica e interpretar bien las tablas.

Los lectores de pantalla más usados son:

JAWS for Windows

[JAWS for Windows](#) (desarrollado por [Freedom Scientific](#)) es uno de los lectores de pantalla más conocidos. Usa un sintetizador de voz incorporado y permite dirigir la salida a una línea braille. Tiene una gran cantidad de funciones especiales y comandos. Por contra, su uso es complicado y requiere de un largo periodo de aprendizaje para conocer todas sus funcionalidades.

Window-Eyes

[Window-Eyes](#) (desarrollado por [GW Micro](#)) es posiblemente el segundo lector de pantalla más usado, después de JAWS. Con características similares, también es complicado de usar y la curva de aprendizaje es similar a la de JAWS.

5.3. Navegadores de voz

Los **navegadores de voz** son aplicaciones similares a los lectores de pantalla pero enfocados sólo al ámbito de la accesibilidad Web.

Un lector de pantalla es un tipo de software que funciona *sobre* el resto de aplicaciones y el sistema operativo, leyendo el contenido de la pantalla.

Los navegadores de voz, en cambio, son aplicaciones independientes y funcionan de forma autónoma para permitir el acceso a Internet. En vez de usar un navegador convencional junto con un lector de pantalla, los navegadores de voz son aplicaciones que integran los dos tipos de software en uno solo (navegador + sintetizador de voz).

El coste de los navegadores de voz suele ser inferior al de los lectores de pantalla y son una buena alternativa a estas ayudas más potentes y versátiles, pero también mucho más caras. El navegador de voz más conocido y usado es **Home Page Reader** de IBM.

Home Page Reader

[Home Page Reader](#) está desarrollado por [IBM](#) y está disponible en varios idiomas. Esta ayuda dispone de un sintetizador de voz para la lectura del contenido de las páginas Web para las personas invidentes y además, permite a los usuarios-as con escasa visión redimensionar las ventanas o las fuentes, cambiar los colores de fondo, etc. También tiene características especiales de navegación por tablas que facilita a los usuarios-as la comprensión de tablas complejas.

5.4. Navegadores de texto

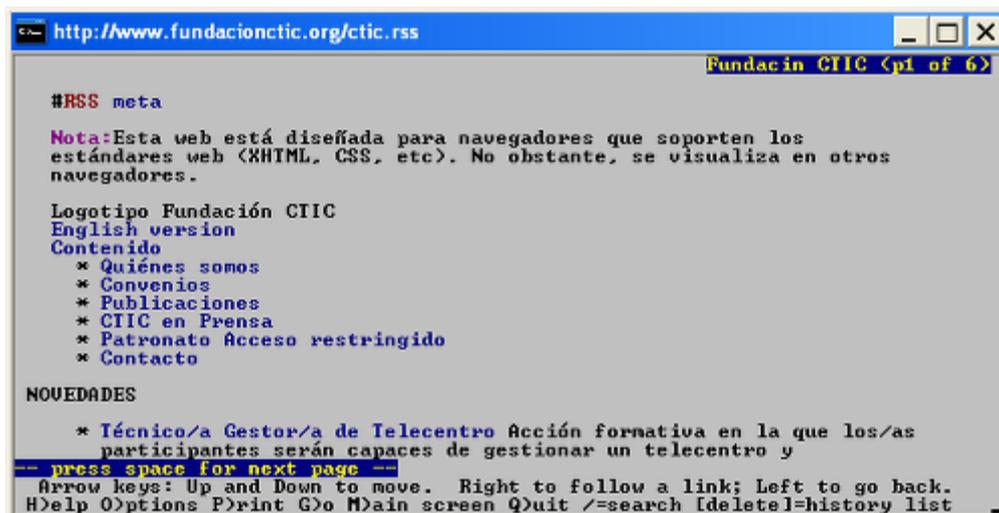
Algunos usuarios-as ciegos-as utilizan los navegadores de texto en conjunción con algún sintetizador de voz.

Por lo general, estos navegadores son demasiado sencillos y, aunque se usen con un sintetizador de voz, no se acercan a la funcionalidad que se puede encontrar en lectores de pantalla como JAWS o Windows Eyes o en navegadores de voz como Home Page Reader. También los pueden usar personas con conexiones lentas o que utilizan dispositivos de pantalla con escasa capacidad gráfica.

A continuación te mostramos algunos navegadores de texto disponibles:

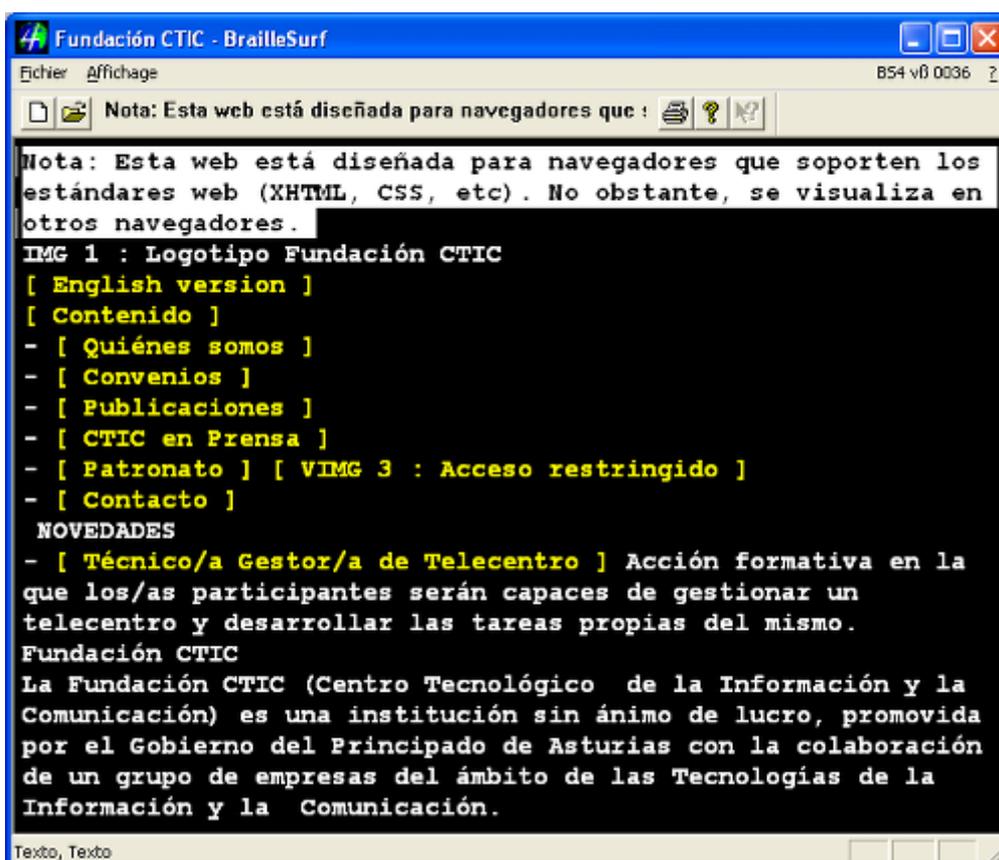
LYNX

El navegador de textos más usado es [Lynx](#). Es un navegador gratuito y está disponible en varios sistemas operativos como VMS, Windows, Linux o Unix.



BrailleSurf

Es un navegador de texto pensado como ayuda para las personas con discapacidades visuales. [BrailleSurf](#) muestra la información disponible en la Web en un formato de texto que puede ser interpretado por una línea braille o un lector de pantalla.



5.5. Ampliadores de pantalla

Los **ampliadores de pantalla** son aplicaciones software que permiten aumentar una parte de la pantalla para facilitar la visualización de la información a las personas con baja visión.

El inconveniente de los ampliadores de pantalla es que, al aumentar la imagen, también reducen el campo de visión o parte visible de la pantalla, perdiéndose la información del contexto de lo que se ve. Para paliar este problema, algunos ampliadores de pantalla proporcionan dos vistas de la pantalla, una vista ampliada y otra con el tamaño por defecto para la navegación.

Se puede encontrar gran cantidad de información sobre ampliadores de pantalla en el sitio Web [The Screen Magnification Home Page](#). En la [sección de descargas de ampliadores de pantalla](#) hay una clasificación por sistemas operativos de los ampliadores de pantalla disponibles.

5.6. Reconocimiento de voz

Estos mecanismos permiten a los usuarios-as con alguna discapacidad física, lesiones en manos o antebrazos (crónicas o temporales) manejar el ordenador e introducir información sin necesidad de usar otros dispositivos como el teclado o el ratón.

Estos sistemas de **reconocimiento de voz** están formados por una parte hardware (un micrófono) y un software que interpreta la voz y la traduce en texto.

El **reconocimiento de voz** también puede ser útil para personas que sufren trastornos de aprendizaje (dificultadas para la lectura y la escritura), así como para aquellos usuarios-as que, por las características del entorno o de su trabajo, tienen las manos ocupadas y necesitan una alternativa para interactuar con el ordenador.



Los programas más importantes de reconocimiento de voz son:

- [Dragon Naturally Speaking](#)
- [IBM ViaVoice](#)
- [Philips FreeSpeech](#)

5.7. Teclados alternativos y conmutadores

Los **teclados alternativos** son ayudas técnicas diseñadas específicamente para usuarios-as con determinadas discapacidades (generalmente físicas o motrices) o para ciertas condiciones de uso impuestas por el entorno.

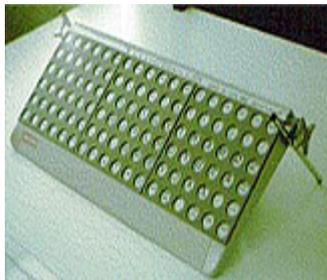
Estos teclados (y sus teclas) suelen ser más grandes o más pequeños que los teclados convencionales, con una disposición de teclas diferente o pensados para trabajar con una sola mano.

Además de los teclados alternativos, hay una serie de conmutadores y dispositivos que permiten usar el ordenador sin necesidad de recurrir al ratón o teclado estándar.

A continuación, se muestran algunos ejemplos de este tipo de ayudas técnicas:

Teclados ampliados

Las teclas se encuentran muy separadas para impedir que se pulsen dos teclas al mismo tiempo. Éstos teclados resultan muy útiles para personas con poca precisión de movimientos.



Teclados reducidos

Tienen dimensiones más pequeñas que un teclado normal y requieren menor amplitud de movimientos.



Teclados para una sola mano

Con una disposición especial de las teclas que requieren menos amplitud de movimientos para poder usarse con una sola mano.



Teclados en pantalla

Muestran un teclado en la pantalla y el usuario-a puede pulsar las teclas por medio de un dispositivo de apuntamiento (ratón, joystick...).

Estas aplicaciones suelen tener la opción de detección, en la que las teclas se van iluminando secuencialmente permitiendo que el usuario-a seleccione la

tecla que desee apretando un botón (este tipo de ayuda también se denomina *software de rastreo*).



Punteros de cabeza o licornios

Son cascos con una varilla metálica incorporada. En el extremo de la varilla se puede colocar un puntero para golpear las teclas mediante movimientos de la cabeza.



Varillas bucales

Similares a los licornios pero se usan sujetándolas con la boca.



Pantallas táctiles

Dispositivos que se adaptan al monitor (o son parte del mismo) y que detectan la presión al pulsar sobre ellos. Permiten seleccionar opciones o teclas directamente al pulsar sobre el monitor. Estas pantallas son útiles para personas con discapacidades motrices, para los usuarios-as de punteros de cabeza o con problemas de aprendizaje, pues su uso es más sencillo e intuitivo que el teclado y ratón.

Sistemas de reconocimiento del movimiento del ojo

Mediante un dispositivo (gafas especiales, cámaras situadas delante del usuario-a...) estos sistemas detectan hacia dónde está mirando el ojo y permiten dirigir el cursor sobre la pantalla. Son útiles para personas con una movilidad muy restringida y se pueden emplear conjuntamente con otras ayudas técnicas, como los teclados en pantalla.

Detectores del movimiento de la cabeza

Funcionan de forma similar a los sistemas de reconocimiento del ojo pero se basan en el movimiento de la cabeza.

Conmutadores de aspiración y soplado.

Estas ayudas se activan con la respiración del usuario-a.

5.8. Software de rastreo

El **software de rastreo** es una aplicación que muestra en pantalla una serie de opciones, enlaces, textos, frases...de forma secuencial (una a una), de forma que el usuario-a, por medio de algún dispositivo de entrada (botón, pulsador...) pueda seleccionar el elemento que desee cuando esté resaltado.

5.9. Navegación con tabulador y teclas de acceso rápido

La navegación mediante tabulador y teclas de acceso rápido no es una ayuda técnica, sino una **estrategia de adaptación** de las personas con discapacidad para acceder a la Web.

El **tabulador** permite acceder a todos los enlaces de una página Web y a los controles de formulario de forma **secuencial**. Al pulsar la tecla de tabulación se sitúa el foco sobre el siguiente enlace o control de la página Web. Esta técnica se utiliza con frecuencia con el único inconveniente que para acceder a un enlace hay que recorrer previamente todos los anteriores.

Las **teclas de acceso rápido** son combinaciones de teclas que permiten un **acceso directo** a un enlace, un control de formulario, o cualquier zona predefinida de la página. Un ejemplo de teclas de acceso rápido son las combinaciones usadas en la mayoría de los programas asociadas a las acciones de copiar (Control+C), cortar (Control+X) o pegar (Control+V).

Los sitios Web que se pueden usar completamente mediante el **teclado**, sin ratón, generalmente también son **accesibles** para otros modos alternativos de interacción o ayudas técnicas.

Estas técnicas de navegación o estrategias no son exclusivas de las personas con discapacidad, sino que pueden ser empleadas por **todos los usuarios**.

6 Ventajas de la accesibilidad Web

La accesibilidad Web no solo presenta ventajas para los usuarios con discapacidades físicas, psíquicas o sensoriales, **beneficia** también a **todos los usuarios en general**.

Las principales **ventajas** de la accesibilidad Web son:

Posibilita el acceso de las personas con discapacidad y de las personas mayores.

La accesibilidad nace para permitir y mejorar el acceso de las personas con discapacidad a los contenidos Web. Sin embargo, su aplicación conlleva también ventajas para las personas mayores. Con la edad se pierden destrezas y capacidades cognitivas que genera dificultades para, por ejemplo, entender ciertos contenidos o para manejar con facilidad ciertos dispositivos.

Aumenta la usabilidad.

Tanto la usabilidad como la accesibilidad son dos conceptos interrelacionados. Conceptos como facilidad de navegación, ergonomía de interfaces, facilidad de manejo, sencillez y eficiencia, se manejan en ambas disciplinas. Así, se puede decir que los sitios Web accesibles son por definición más *usables*.

Simplifica el desarrollo.

Ciertas condiciones y requisitos técnicos que recomienda la accesibilidad dan como resultado mejoras en los procesos de desarrollo. Recomendaciones como el uso de CSS (hojas de estilo en cascada) para separar la presentación del contenido o el uso de estándares Web posibilita mantenimientos posteriores de la Web más rápidos.

Mejora la indexación.

En general, y para posibilitar la búsqueda posterior, los contenidos Web se indexan mediante agentes. Un ejemplo son los contenidos multimedia. Al ser textualizados se enriquece la información de la Web y las búsquedas que sobre ella se realicen. Una imagen con un texto alternativo puede ser indexada más efectivamente por los buscadores.

Facilita la independencia de dispositivo y la interoperabilidad.

En general, se puede decir que una Web que alcanza un buen nivel de accesibilidad, puede verse de forma similar en un ordenador doméstico, en un dispositivo de mano o en una televisión, por citar algún ejemplo.

Mejora del acceso en general.

Como hemos comentado, las mejoras de usabilidad, de navegación, de estructuración... asociadas a la accesibilidad, constituyen valores que benefician en sí mismos a todos los usuarios-as de la Web. Por tanto, no hay que ver la accesibilidad como una serie de requisitos aislados para un colectivo concreto sino como opciones de mejora de la calidad de la Web en general.

Asimismo, no obliga al usuario-a a interactuar con un dispositivo dado (multimodalidad), por ejemplo el ratón, sino que permite otros medios de interacción como el teclado. El hecho de no utilizar el ratón puede ser una cuestión de imposibilidad física, limitación del entorno, o una elección personal.

7 Legislación

La **accesibilidad Web** no sólo es una recomendación, sino que en algunos casos se trata de una obligación marcada por la legislación existente.

Cada vez más, los países incorporan una **legislación sobre la accesibilidad Web** e incluso algunas organizaciones de diferente tipo (educativas, comerciales...) tienen entre sus premisas que sus sitios Web sean accesibles.

La creciente importancia de la Sociedad de la Información y de las Tecnologías de la Información y la Comunicación (TIC) ha aumentado la necesidad de una política y legislación que las regule y que garantice el acceso a la misma a todos los ciudadanos independientemente de sus discapacidades.

7.1. Europa

A **nivel europeo**, la sensibilización en estos temas ya se significó en la Comunicación [e-Europe](#) del 9 de diciembre de 1999 (Comunicación e-Europe "Una sociedad de la información para todos") en la que una de sus diez áreas prioritarias era *la participación de las personas con discapacidad en la cultura electrónica*.

En este informe se sugiere la adopción de las normas WAI del W3C para sitios Web públicos. De ahí que los estados miembro y la Comisión Europea deberán exigir que los proyectos y actividades u organizaciones que reciban financiación pública cumplan las directrices de accesibilidad en relación con el diseño de sitios Web. Asimismo, estas instituciones recomiendan la adopción de la recomendación del WAI doble A para los portales de la administración pública.

A continuación se enumeran los principales materiales de referencia sobre **e-Europe**:

- Comunicación de la Comisión al Consejo, al Parlamento Europeo, al Comité Económico y Social y al Comité de las Regiones de 25 de septiembre de 2001: "[eEurope 2002: Accesibilidad de los sitios Web públicos y de su contenido](#)".
- [Informe al Parlamento Europeo](#) sobre la Comunicación de la Comisión de Industria, Comercio Exterior, Investigación y Energía del 24 de abril de 2002: "eEurope 2002: Accesibilidad de los sitios Web públicos y de su contenido".
- [Resolución del Consejo \(PDF\)](#) sobre "Accesibilidad electrónica" del 14 de enero de 2003.

7.2. España

La legislación específica sobre accesibilidad en **España** es la siguiente:

LSSICE (34/2002)

Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico: [LSSICE](#).

Extracto:

Disposición adicional quinta. *Accesibilidad para las personas con discapacidad y de edad avanzada a la información proporcionada por medios electrónicos.*

Uno. *Las Administraciones Públicas adoptarán las medidas necesarias para que la información disponible en sus respectivas páginas de Internet pueda ser accesible a personas con discapacidad y de edad avanzada de acuerdo con los criterios de accesibilidad al contenido generalmente reconocidos antes del 31 de diciembre de 2005. Asimismo, podrán exigir que las páginas de Internet cuyo diseño o mantenimiento financien, apliquen los criterios de accesibilidad antes mencionados.*

Dos. *Igualmente, se promoverá la adopción de normas de accesibilidad por los prestadores de servicios y los fabricantes de equipos y software, para facilitar el acceso de las personas con discapacidad o de edad avanzada a los contenidos digitales.*

LIONDAU (51/2003)

Ley 51/2003, de 2 de diciembre, de Igualdad de Oportunidades, No Discriminación y Accesibilidad Universal de las personas con discapacidad: [LIONDAU](#) publicada en el [BOE del 03/12/2003](#). En la disposición final 7ª se establecen las condiciones básicas de accesibilidad y no discriminación para el acceso y utilización de las tecnologías, productos y servicios relacionados con la sociedad de la información y medios de comunicación social.

Extracto:

Disposición final 7ª, *Condiciones básicas de accesibilidad y no discriminación para el acceso y utilización de las tecnologías, productos y servicios relacionados con la sociedad de la información y medios de comunicación social.*

En el plazo de dos años desde la entrada en vigor de esta ley, el Gobierno aprobará unas condiciones básicas de accesibilidad y no discriminación para el acceso y utilización de las tecnologías, productos y servicios relacionados con la sociedad de la información y de cualquier medio de comunicación social, que serán obligatorias en el plazo de cuatro a seis años desde la entrada en vigor de esta ley para todos los productos y servicios nuevos, y en el plazo de ocho a diez años para todos aquellos existentes que sean susceptibles de ajustes razonables.

En el plazo de dos años desde la entrada en vigor de esta ley, el Gobierno deberá realizar los estudios integrales sobre la accesibilidad a dichos bienes o servicios que se consideren más relevantes desde el punto de vista de la no discriminación y accesibilidad universal.

7.3.Estados Unidos

La legislación sobre accesibilidad Web aplicable en **Estados Unidos** es la [Sección 508](#) perteneciente al Acta de Rehabilitación (Rehabilitation Act.) de 1973. El Acta de Rehabilitación es una legislación federal para autorizar programas de rehabilitación, apoyo en el empleo e independencia de las personas con discapacidad, así como actividades de investigación administradas por el Instituto Nacional de Rehabilitación y

Accesibilidad y Usabilidad Web

Discapacidad. El acta también incluye diferentes informes centrados en los derechos, defensa y protección de las personas con discapacidad.

En la **Sección 508**, enmienda aprobada en 1998, se detallan los requisitos sobre accesibilidad que deben tener los productos informáticos que se quieran vender a la Administración Federal. Entre ellos se incluyen los **requisitos para el desarrollo de sitios Web accesibles**, tomando como referencia las Pautas de Accesibilidad al Contenido Web del W3C.

8 Bibliografía

- [*How People with Disabilities Use the Web*](#). W3C Working Draft, 8 Julio de 2004. Judy Brewer.
- [*Web Content Accessibility Guidelines 1.0*](#). Recomendación W3C, 5 de Mayo de 1999. Wendy Chisholm, Gregg Vanderheiden, Ian Jacobs.
- [*Alternative Web Browsing*](#). Peter Boshier y Judy Brewer (2001)
- [*Web Accessibility Initiative \(WAI\)*](#). Iniciativa de accesibilidad Web.
- [*W3C Oficina Española*](#). Oficina española del W3C.
- [*What is Universal Design?*](#). NC State University, The Center for Universal Design (1997).
- [*Clasificaciones de la OMS sobre discapacidad*](#). Carlos Egea García y Alicia Sarabia Sánchez (2001).
- [*LEY 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico*](#) (en PDF). LSSICE, publicada en el Boletín Oficial del Estado. 12/07/2002. nº166.
- [*LEY 51/2003, de 2 de diciembre, de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad*](#) (otro enlace). LIONDAU, publicada el BOE del 03/12/2003, nº 289.
- [*eEurope 2002: Accesibilidad de los sitios Web públicos y de su contenido*](#) Comunicación e-Europe del 9 de diciembre de 1999 (Comunicación e-Europe "Una sociedad de la información para todos").
- [*Informe al Parlamento Europeo*](#), sobre la Comunicación de la Comisión de Industria, Comercio Exterior, Investigación y Energía.
- [*Discapnet, Discapacidades y Deficiencias*](#). Portal Web sobre la salud y la discapacidad
- [*Clases de productos de tecnología de asistencia*](#), Microsoft Accessibility Technology for Everyone.
- [*Tecnum, Catálogo de Ayudas Técnicas*](#).

Introducción a las Pautas de Accesibilidad al Contenido Web

1 Organización de las pautas

Las Pautas de accesibilidad al Contenido en la Web ([WCAG 1.0](#)), publicadas por la Iniciativa de Accesibilidad Web ([WAI](#)) del [W3C](#), pasaron a ser una recomendación en Mayo de 1999. Las pautas explican cómo hacer accesible el contenido de las páginas Web a personas con discapacidad, aunque también benefician a todos los usuarios-as en general. Están dirigidas tanto a los desarrolladores-as Web como a los desarrolladores-as de herramientas de autor.

El documento está formado por 14 **Pautas Generales** sobre el diseño accesible:

1. Proporcione alternativas equivalentes para el contenido visual o auditivo.
2. No se base solo en el color.
3. Utilice marcadores y hojas de estilo y hágalo adecuadamente.
4. Identifique el idioma usado.
5. Cree tablas que se transformen correctamente.
6. Asegúrese de que las páginas que incorporan nuevas tecnologías se transformen correctamente.
7. Asegure al usuario el control sobre los cambios de los contenidos tempo-dependientes.
8. Asegure la accesibilidad directa de las interfaces de usuario incrustadas.
9. Diseñe para la independencia de dispositivo.
10. Utilice soluciones provisionales.
11. Utilice las tecnologías y pautas del W3C.
12. Proporcione información de contenido y presentación.
13. Proporcione mecanismos claros de navegación.
14. Asegúrese de que los documentos sean claros y simples.

Cada pauta a su vez está dividida en varios **puntos de verificación** o aspectos concretos sobre la aplicación de la pauta. En total hay 65 puntos de verificación.

2 Prioridades y niveles de adecuación

Cada **punto de verificación** tiene asignada una **prioridad** que indica cómo afecta a la accesibilidad de un sitio Web si dicho punto de verificación no se cumple. Es decir, según cuál sea su impacto en la accesibilidad.

Prioridad 1

Un desarrollador de contenidos de páginas Web **tiene** que satisfacer este punto de verificación. De otra forma, uno o más grupos de usuarios encontrarán imposible acceder a la información del documento. Satisfacer este punto de verificación es un requerimiento básico para que algunos grupos puedan usar los documentos Web.

Prioridad 2

Un desarrollador de contenidos de páginas Web **debe** satisfacer este punto de verificación. De otra forma, uno o más grupos encontrarán dificultades en el acceso a la información del documento. Satisfacer este punto de verificación eliminará importantes barreras de acceso a los documentos Web.

Prioridad 3

Un desarrollador de contenidos de páginas Web **puede** satisfacer este punto de verificación. De otra forma, uno o más grupos de usuarios encontrarán alguna dificultad para acceder a la información del documento. Satisfacer este punto de verificación mejorará la accesibilidad de los documentos Web.

Existen tres **niveles de adecuación** que indican el grado de cumplimiento de los puntos de verificación por un determinado sitio Web.

Adecuación de nivel A (A)

Se satisfacen todos los puntos de verificación de prioridad 1.

Adecuación de nivel Doble A (AA)

Se satisfacen todos los puntos de verificación de prioridad 1 y 2.

Adecuación de nivel Triple A (AAA)

Se satisfacen todos los puntos de verificación de prioridad 1, 2 y 3.

Así, por ejemplo, si queremos que nuestro sitio Web tenga un nivel de adecuación AA tendremos que cumplir todos los puntos de verificación de prioridad 1 y de prioridad 2.

3 Documentos complementarios

Además de las Pautas de Accesibilidad, el WAI ha desarrollado una serie de documentos y anexos que sirven como complemento a las WCAG.

3.1. Documentos de Técnicas

Los **documentos de técnicas** describen cómo aplicar los puntos de verificación definidos en las "[Pautas de accesibilidad para el Contenido Web](#)".

- "[Técnicas para las Pautas de Accesibilidad al Contenido en la Web 1.0](#)", el documento índice de los documentos de técnicas descritos a continuación.
- "[Técnicas fundamentales para las Pautas de Accesibilidad al Contenido en la Web 1.0](#)", que expone los aspectos fundamentales de accesibilidad y las técnicas generales
- "[Técnicas HTML para las Pautas de Accesibilidad al Contenido en la Web 1.0](#)": con ejemplos de técnicas de programación en HTML para hacer el contenido accesible.
- "[Técnicas CSS para las Pautas de Accesibilidad al Contenido en la Web 1.0](#)": similar al anterior, pero con ejemplos y estrategias para crear hojas de estilo en cascada que sirvan para crear páginas Web accesibles.

3.2. Tabla de puntos de verificación

Además de las pautas de accesibilidad y de los documentos de técnicas, el WAI ha creado un documento denominado "[Tabla de puntos de verificación](#)". En dicha tabla se enumeran todos los puntos de verificación **organizados por prioridades y por conceptos**, de forma que la tabla sirva como documento base para hacer revisiones de accesibilidad. Por cada punto de verificación dispone de tres celdas en las que se puede marcar si dicho punto se satisface, si no se satisface, o si no es aplicable.

La tabla se divide en tres partes bien diferenciadas, cada una de ellas agrupando todos los puntos de verificación de una prioridad. Dentro de cada prioridad, se organizan los puntos de verificación según varios conceptos, como imágenes, multimedia, tablas, marcos, formularios, applets y scripts, o puntos de carácter general.

4 Organización del curso

El presente curso proporciona una **organización más didáctica** a las pautas y puntos de verificación. La organización usada se asemeja más a la empleada en la tabla de puntos de verificación o en los documentos de técnicas, que la propia organización de las WCAG.

Se han ordenado los puntos de verificación por **conceptos** y dentro de cada concepto, por **prioridades**. Por ejemplo, en la unidad de formularios están listados, ordenados por prioridad, todos los puntos de verificación aplicables, directa o indirectamente, a los formularios.

Las diferentes categorías en las que se han dividido los puntos de verificación, correspondientes con una unidad del curso, son las siguientes:

- Imágenes y multimedia
- Objetos programados
- Marcos
- Tablas
- Formularios
- Estándares
- Estructura y presentación
- Navegación, lenguaje y comprensión.

Se han agrupado los puntos de verificación teniendo en cuenta su uso, y no su situación dentro de las pautas. Así, si quieres saber qué tienes que hacer para crear un formulario accesible sólo tienes que consultar la unidad correspondiente.

Sin embargo, con esta estructura es posible que se repitan algunos puntos de verificación que sean aplicables a diferentes conceptos. Es decir, algún punto de verificación aplicable a imágenes y multimedia también puede ser aplicable a objetos programados (applets, scripts) y por tanto aparecerá en las dos unidades didácticas. Esto es porque se pretende que las unidades didácticas sean cerradas, que no haya que consultar otra unidad en busca de puntos de verificación aplicables a los conceptos explicados en la unidad actual.

Imágenes y Multimedia

*El texto se considera accesible para prácticamente todos los usuarios si puede ser manejado por lectores de pantalla, navegadores no visuales y lectores braille. Puede ser presentado visualmente, agrandado, sincronizado con un video para crear un subtítulo, etc. Durante el diseño de un documento que contenga información no textual complemente esa información con **textos equivalentes** siempre que sea posible.*

*Cuando se presente al usuario un equivalente textual, éste debe cumplir esencialmente la misma función (en la medida de lo posible) que el contenido original. Para **contenidos simples**, un equivalente textual puede sólo necesitar describir la **función o propósito** del contenido. Para **contenidos complejos** (gráficas, gráficos, etc.) el texto equivalente puede ser más largo e incluir **información descriptiva**.*

Deben proporcionarse textos equivalentes para los logotipos, fotografías, botones de envío, viñetas en listas, ASCII art y en todos los vínculos contenidos en un mapa de imagen, así como en las imágenes invisibles usadas para maquetar una página.

Pautas WCAG del WAI

1 Imágenes

Prioridad 1

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.

Equivalente textual [1.1 - Prioridad 1]

Todas las imágenes deben disponer de un **equivalente textual** que proporcione la misma información o que cumpla la misma función que la imagen.

El **texto alternativo** indica a los dispositivos de usuario-a qué es lo que deben hacer en caso de no mostrar las imágenes (porque estén desactivadas, porque se trate de un navegador en modo de texto o un lector de pantalla, etc.). Por tanto, el texto alternativo debe proporcionar la **misma información o función** que aporta la imagen.

En una fotografía o en un logotipo, por ejemplo, puedes describir de forma sencilla su contenido, de forma que las personas que no vean la imagen se hagan una idea de la misma.

Evita la utilización de textos alternativos que incumplan con su función, como poner el nombre de la imagen, su tamaño, su resolución o palabras como "imagen", "foto", etc. Todos estos ejemplos no aportan ninguna información sobre lo que se ve en la imagen, o de la información que aporta la imagen, que es el objetivo del texto alternativo.

A la variedad de imágenes que se pueden incluir en un sitio web hay que sumarle las diferentes formas de incluirlas y funcionalidades que pueden tener, con lo que a veces no es fácil decidir cuál es el texto alternativo más adecuado. A continuación te mostramos diferentes casos que te puedes encontrar a la hora de incluir imágenes en la web.

1. Imágenes sencillas

Muchas de las imágenes que se incluyen en las páginas Web son imágenes sencillas que no requieren de una descripción de masiado extensa. Para este tipo de imágenes basta con proporcionar una **breve descripción** mediante el atributo "alt".

Para las imágenes sencillas proporciona una **breve descripción** mediante el atributo "alt".



Ejemplo.

```

```

2. Imágenes con OBJECT

Las imágenes también se pueden añadir al documento HTML mediante el elemento **OBJECT** en vez de IMG.

El elemento OBJECT se caracteriza porque el contenido que se añade dentro de su cuerpo sólo se muestra en caso de que el navegador no pueda mostrar el propio objeto. De esta forma, a las imágenes añadidas con el elemento OBJECT se les puede proporcionar un equivalente textual simplemente incluyéndolo en su contenido.

Ejemplo.

```
<object data="casa.gif" type="image/gif"
height="100" width="100">
  En este lugar se ubica la descripción
  de texto (breve, larga o enlazando a un
  archivo externo).
</object>
```

3. Imágenes decorativas

Las imágenes que se usan con fines decorativos y apenas aportan información no necesitan texto alternativo (por ejemplo, las imágenes usadas como espaciadores, invisibles o transparentes, imágenes como líneas o como bordes de determinados elementos...).

En estos casos el texto alternativo es innecesario, pero como el uso del atributo "alt" es obligatorio debes poner un texto alternativo vacío. Un texto alternativo vacío indica a los dispositivos de usuario-a que, si es imposible mostrar la imagen, no deben mostrar nada, ya que carece de importancia.

Ejemplo.

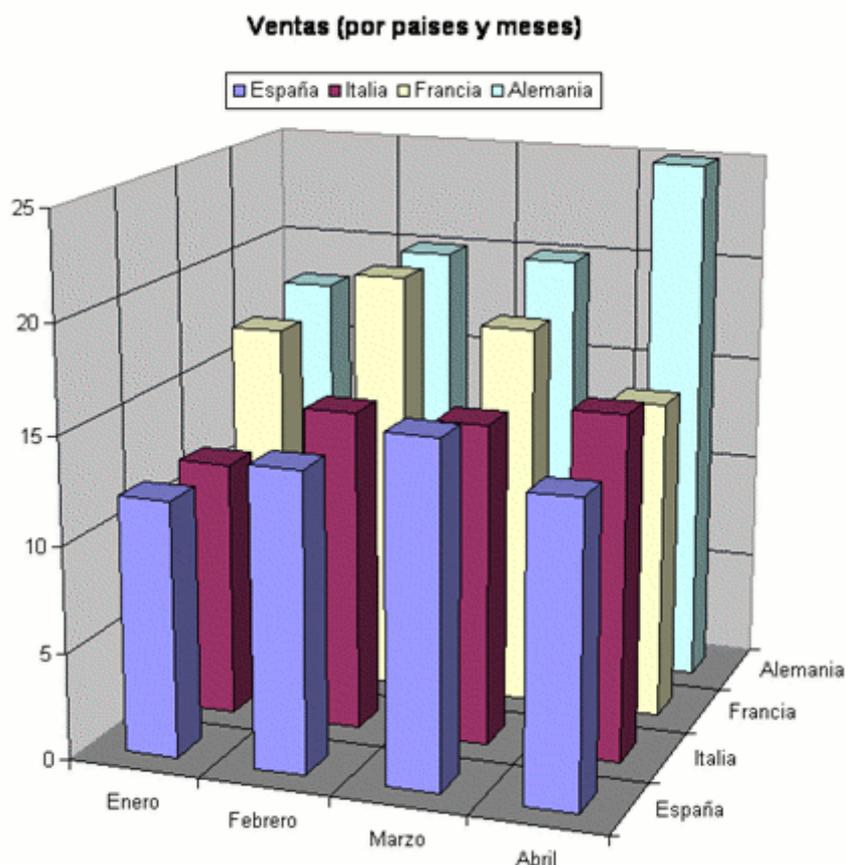
```

```

4. Imágenes complejas

Si la información aportada por la imagen es importante y precisa de una descripción más detallada debes usar el atributo "**longdesc**".

El atributo "longdesc" del elemento IMG proporciona la dirección de una página HTML en la que se incluye una descripción detallada de la imagen.



Ejemplo.

```

```

En la página indicada en el atributo "longdesc" se describiría detalladamente la imagen. Por ejemplo, para la imagen de la figura se podría aportar la misma información en una tabla de datos (elemento TABLE).

Otra forma de proporcionar descripciones largas es mediante el uso de **enlaces descriptivos** (en inglés "D-links").

Los **enlaces descriptivos** son enlaces normales a la página en la que se proporciona la descripción de la imagen y deben estar situados de forma visible cerca de la imagen.

Generalmente se usa la letra D ("Descripción") como texto del enlace. Se recomienda el uso de este tipo de enlaces debido al escaso soporte que dan los navegadores al atributo "longdesc" en la actualidad.

Ejemplo.

```
<a href="grafico1.html"
title="Descripción larga de la imagen">
  [D]
</a>
```

5. Imágenes que son texto

Para las imágenes que contienen texto introduce en el atributo "alt" el **mismo texto** que el que aparece en la imagen.



Ejemplo.

```

```

Se debería evitar la utilización de imágenes para incluir texto. El punto de verificación 3.1 aconseja el uso de marcadores apropiados en lugar de imágenes para transmitir información.

En el caso de las imágenes con texto, incluye el texto en el marcado HTML (como un párrafo, un DIV...) y dale formato mediante las hojas de estilo.

6. Imágenes como botones de formulario

Las imágenes que funcionan como botones de formulario, añadidas mediante el elemento **INPUT de tipo "image"**, deben contener un texto alternativo siguiendo las mismas indicaciones que para el resto de imágenes.

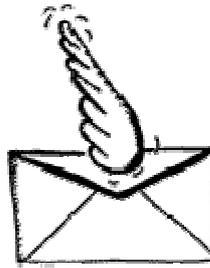
Nombre: **Enviar**

Ejemplo.

```
<input type="image" src="img/enviar.gif"
alt="Enviar formulario">
```

7. Imágenes con animaciones

En las imágenes animadas, como en aquellas de tipo GIF animado, habría que describir escuetamente la imagen y la animación.



Ejemplo.

```

```

8. Imágenes como enlaces

Cuando se usa una imagen como enlace existen dos posibilidades para proporcionar un texto alternativo.

1. Si la **imagen es el único elemento que funciona como enlace**, entonces especifica claramente el destino del enlace en el texto alternativo de la imagen.

En ese caso describe el enlace y evita la descripción de la imagen. Si la imagen necesita una descripción utiliza el atributo "longdesc".



Ejemplo.

```
<a href="capitulo2.html">
  
</a>
```

2. Si, **además de la imagen, el enlace dispone de un texto que funciona como vínculo** y el contenido de la misma es irrelevante, entonces evita especificar el destino del enlace en el texto alternativo, dejándolo vacío. Así evitarás que los lectores de pantalla lean el mismo texto dos veces seguidas (el texto del enlace y el texto alternativo de la imagen con el destino del enlace).



Anterior

Ejemplo.

```
<a href="capitulo2.html">
  
  Anterior
</a>
```

Prioridad 3

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

ASCII Art [13.10 - Prioridad 3]

Utiliza imágenes reales en vez de gráficos creados mediante caracteres (ASCII art) ya que es más fácil proporcionar un texto alternativo para éstas.

Por otra parte, los dibujos con caracteres son muy molestos para los usuarios-as de lectores de pantalla ya que los leerán sin ningún sentido. Por tanto, conviene que proporciones un texto descriptivo y una manera de saltar los ASCII art.

Por ejemplo:

([Saltar dibujo con caracteres](#) de una cara con un sombrero)

```

MMMMMMM
WWWWW   WWWWW
*****
 |o/    o/|
   |
   |
  -----/
   vvvv
    vv
     v
    
```

En este caso el código de la figura sería:

Ejemplo.

```

<p>
  (<a href="#post-art">
    Saltar dibujo con caracteres
  </a>
  de una cara con un sombrero)
  <!-- Dibujo aquí -->
  <a name="post-art">
    Texto después de gráfico...
  </a>
</p>
    
```

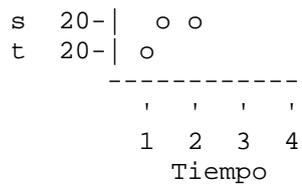
Si el dibujo tiene información importante se puede proporcionar un enlace a una página con una descripción larga del mismo, de forma similar al atributo "longdesc" del elemento IMG.

Por ejemplo:

([Saltar dibujo con caracteres](#) o consultar una [descripción del gráfico](#))

```

c 40- |      o
o 30- |      o o
    
```



Descripción del ascii art: [gráfica de ejemplo con información estadística.](#)

Texto después del gráfico...

El texto de después del gráfico puede ser el título del mismo. En el caso de dibujos pequeños puedes emplear el elemento ABBR y el atributo "title" para proporcionarles un texto alternativo. Este puede ser el caso de los típicos *emoticones* y similares.

Ejemplo.

```

<abbr title="cara sonriente (hecha con caracteres)">
  :-)
</abbr>
  
```

2 Mapas de imagen

Prioridad 1

[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.

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

[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.

Tipos de mapas de imagen

Los mapas de imagen los podemos clasificar en dos categorías: mapas de imagen del lado del servidor y mapas de imagen del lado del cliente.

- **Mapas de servidor:** Cuando el usuario-a pulsa sobre una zona de la imagen, se envían las coordenadas del punto seleccionado al servidor. Éste procesa y determina con qué acción o enlace (URL) se corresponden las coordenadas seleccionadas. La principal forma de crear mapas de imagen de servidor es con el elemento **IMG** y el atributo **"ismap"** o bien mediante el elemento **INPUT de tipo "image"** (<INPUT type="image">).
- **Mapas de cliente:** El navegador procesa las coordenadas seleccionadas y la URL correspondiente a cada zona activa del mapa de imagen. El servidor no determina qué zona del mapa de imagen ha sido activada. Los mapas de imagen de cliente se crean con el elemento **IMG** con el atributo **"usemap"**.

Los **mapas de imagen** son **accesibles** si cada acción asociada con una zona activa del mismo puede activarse sin necesidad del ratón u otro dispositivo de puntero.

Por tanto, es necesario proporcionar un **equivalente textual** para los enlaces de un mapa de imagen.

Equivalentes textuales en mapas de imagen de cliente [1.1 - Prioridad 1]

Para que un mapa de imagen de cliente sea directamente accesible debe llevar un texto alternativo para cada región activa del mismo.

De ese modo un lector de pantalla puede acceder a la información sin ninguna dificultad. El texto alternativo identificará claramente el objetivo del enlace y tendrá sentido cuando se lea fuera de su contexto.

Para proporcionar un texto alternativo a cada enlace de un mapa de imagen de cliente utiliza el atributo "alt" del elemento **AREA**.



Ejemplo.

```

<map name="map1">
  <area shape="rect" coords="0,0,120,75" href="#"
    alt="¿Quiénes somos?">
  <area shape="rect" coords="121,0,240,75" href="#"
    alt="Unidad de Accesibilidad Web">
  <area shape="rect" coords="0,76,120,150" href="#"
    alt="Formación">
  <area shape="rect" coords="121,76,240,150" href="#"
    alt="W3C">
</map>
```

También puedes crear mapas de imagen de cliente usando el elemento **OBJECT** en vez de **IMG**.

Ejemplo.

```
<object data="img/mapaimagen.gif "
  type="image/gif" usemap="#map1"
  width="240" height="150">
  <p>
    [<a href="#">¿Quiénes Somos?</a>]
    [<a href="#">Unidad de Accesibilidad Web</a>]
    [<a href="#">Formación</a>]
    [<a href="#">W3C</a>]
  </p>
</object>
```

Vínculos redundantes en mapas de imagen de servidor [1.2 - Prioridad 1]

En un mapa de imagen de servidor es el servidor quien procesa las coordenadas que pulsa el usuario-a y determina con qué acción o URL se corresponden.

Accesibilidad y Usabilidad Web

Como todo el proceso se realiza del lado del servidor, el navegador no aporta información al usuario-a sobre la acción relacionada con cada zona del mapa de imagen.

Al contrario que con los mapas de imagen de cliente, es imposible hacer directamente accesible un mapa de imagen de servidor.

En los **mapas de imagen de servidor** debes proporcionar la misma funcionalidad mediante **enlaces redundantes** en formato de texto para cada zona activa del mismo, e indicar la existencia y ubicación de dichos enlaces mediante el atributo "**alt**" del elemento **IMG**.



[\[¿Quiénes Somos?\]](#) [\[Unidad de Accesibilidad Web\]](#) [\[Formación\]](#) [\[W3C\]](#)

Ejemplo.

```
<a href="../../../cgi-bin/imagemap/mapa">
  
</a>
<p>
  [¿Quiénes Somos?]
  [Unidad de Accesibilidad Web]
  [Formación]
  [W3C]
</p>
```

El atributo "alt" se usa para indicar la presencia y ubicación de los enlaces redundantes. Si fuese necesario describir la imagen utiliza el atributo "longdesc".

Mapas de imagen de cliente mejor que de servidor [9.1 - Prioridad 1]

Los mapas de imagen de servidor deben sustituirse por mapas de imagen de cliente que pueden hacerse directamente accesibles al proporcionar equivalentes textuales para cada zona activa.

Prioridad 3

1.5 Hasta que las aplicaciones de usuario interpreten el texto equivalente para los vínculos de los mapas de imagen de cliente, proporcione vínculos de texto redundantes para cada zona activa del mapa de imagen de cliente.

Vínculos redundantes en mapas de imagen de cliente [1.5 - Prioridad 3]

El mapa de imagen de cliente se puede hacer directamente accesible mediante los textos alternativos para cada enlace. De hecho puedes proporcionar enlaces redundantes de varias maneras.



[[¿Quiénes Somos?](#)] [[Unidad de Accesibilidad Web](#)] [[Formación](#)] [[W3C](#)]

Ejemplo.

```

<map name="map1">
  <area shape="rect" coords="0,0,120,75" href="#"
  alt="¿Quiénes somos?">
  <area shape="rect" coords="121,0,240,75" href="#"
  alt="Unidad de Accesibilidad Web">
  <area shape="rect" coords="0,76,120,150" href="#"
  alt="Formación">
  <area shape="rect" coords="121,76,240,150" href="#"
  alt="W3C">
</map>
<p>
  [¿Quiénes Somos?]
  [Unidad de Accesibilidad Web]
  [Formación]
  [W3C]
</p>
```

Si empleas el elemento **A** en vez de **AREA**, junto con el elemento **OBJECT**, puedes describir las regiones activas y proporcionar vínculos redundantes al mismo tiempo:

Ejemplo.

```
<object data="img/mapaimagen.gif"
type="image/gif" usemap="#map1"
width="240" height="150">
  <map name="map1">
    [<a href="#" shape="rect" coords="0,0,120,75">
      ¿Quiénes Somos?</a>]
    [<a href="#" shape="rect" coords="121,0,240,75">
      Unidad de Accesibilidad Web</a>]
    [<a href="#" shape="rect" coords="0,76, 120,150">
      Formación</a>]
    [<a href="#" shape="rect" coords="121,76,240,150">
      W3C</a>]
  </map>
</object>
```

3 Multimedia

Prioridad 1

[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.

[1.3](#) Hasta que las aplicaciones de usuario puedan leer en voz alta automáticamente el texto equivalente de la banda visual, proporcione una descripción auditiva de la información importante de la banda visual de una presentación multimedia.

[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.

Equivalentes textuales para sonidos [1.1 - Prioridad 1]

También debes proporcionar un equivalente textual alternativo a los sonidos.

- Si es un **sonido corto** puedes colocar el texto alternativo en el atributo "alt" de una imagen usada como enlace del archivo de audio.

Ejemplo.

```
<a href="charla.wav">  
    
</a>
```

- Si se trata de un **sonido largo** proporciona una **transcripción textual** del contenido. De hecho, puedes incluir la transcripción en el mismo documento o mediante un enlace a un archivo externo.

Pistas de audio en vídeos [1.4 - Prioridad 1]

Tal y como señalan las directrices WCAG:

Las presentaciones sonoras deben ir acompañadas por transcripciones del texto, equivalentes textuales de los eventos sonoros. Cuando estas transcripciones se presentan de forma sincronizada con la presentación visual, se denominan subtítulos y son utilizados por las personas que no pueden escuchar la banda sonora del material visual.

Algunos formatos de medios (por ejemplo, QuickTime 3.0 y SMIL) permiten añadir subtítulos y descripciones de las imágenes a los clips multimedia. SAMI permite que se

añadan subtítulos. Los subtítulos deberían incluir los diálogos y otros sonidos ambientales que ayuden a los espectadores a entender lo que está ocurriendo.

Hasta que el formato que está usando soporte bandas alternativas, podría ofrecer dos versiones de la película, una con subtítulos y descripción de las imágenes y otra sin ellos. Algunas tecnologías, como SMIL y SAMI, permiten archivos sonoros y visuales separados para combinarlos con los archivos de texto a través del archivo de sincronización para crear audio y películas subtitulados.

- [SAMI](#) (Microsoft Synchronized Accessible Media Interchange).
- [SMIL](#) (Synchronized Multimedia Integration Language).

Por ejemplo, a continuación verás la transcripción de una conversación en formato HTML+TIME, que es la implementación realizada por Microsoft de HTML+SMIL, definido en la especificación [SMIL 2.1](#) (en estado de borrador).

Ejemplo.

```
<html xmlns:t="urn:schemas-microsoft-com:time">
<head>
  <?import namespace="t"
  implementation="#default#time2">
  <style>
    .t {
      behavior: url(#default#time2)
    }
    body{
      font-family: "Verdana", "Arial", sans-serif;
      font-size: 1em;
    }
    .narrador{
      font-style: italic;
    }
    .dialogo{
      font-style: normal;
    }
  </style>
</head>
<body>
  <par>
    <t:audio src="play.wav"
    repeatCount="indefinite" type="wav" />
    <t:seq repeatCount="indefinite">
      <p class="t narrador" dur="3s">
        Narrador: Dos niños, Manuel y Jorge,
        juegan sentados en el suelo.
      </p>
      <p class="t dialogo" dur="2s">
        Manuel: ¡Dámelo, es mío!
      </p>
      <p class="t dialogo" dur="3s">
        Jorge: De eso nada. Lo gané yo jugando,
        así que olvídalos.
      </p>
    </t:seq>
  </par>
```

```
</body>  
</html>
```

En este ejemplo se muestran en la página web, los subtítulos correspondientes a la voz del narrador y a los diálogos. La secuencia sería la siguiente:

Narrador: Dos niños, Manuel y Jorge, juegan sentados en el suelo (*se muestra durante 3 segundos*).

Manuel: ¡Dámelo, es mío! (*2 segundos*)

Jorge: De eso nada. Lo gané yo jugando, así que olvídale (*3 segundos*).

Vídeo [1.3 - Prioridad 1]

Las pautas WCAG, en relación con las **descripciones auditivas**, proponen que:

"Las descripciones sonoras de la banda visual proporcionan una narración de los elementos visuales claves sin interferir con el sonido o el diálogo de una película. Los elementos visuales clave incluyen acciones, escenarios, lenguaje corporal, gráficos y el texto mostrado. Las descripciones auditivas son utilizadas, primordialmente, por las personas ciegas para seguir la acción y la información no auditiva en el material visual."

En relación con las **transcripciones textuales completas** el WAI aconseja lo siguiente:

"Las transcripciones textuales completas permiten el acceso de las personas con discapacidades tanto visuales como auditivas. También proporcionan a cualquier persona la posibilidad de indexar y buscar información contenida en materiales audiovisuales."

Las transcripciones textuales completas incluyen diálogos hablados, así como cualquier otro sonido significativo, aparezca o no en pantalla: música, risas, aplausos. etc. En otras palabras, todo el texto que aparece en subtítulos, así como las descripciones que se proporcionan en la narración sonora."

Ejemplo:

Transcripción textual completa de una escena

Narrador: Dos niños, Manuel y Jorge, juegan sentados en el suelo.

Manuel: ¡Dámelo, es mío!.

Jorge: De eso nada. Lo gané yo jugando, así que olvídale.

Objetos Programados

"Si bien se alienta a los desarrolladores de contenidos a usar nuevas tecnologías que superen los problemas que proporcionan las tecnologías existentes, deberán saber cómo hacer para que sus páginas funcionen con navegadores más antiguos, y para quienes decidan desconectar esta características."

Pautas WCAG del WAI

El **contenido dinámico**, a diferencia del contenido estático, es aquel contenido que se genera, cambia o actualiza sin la intervención directa de una persona (programador-a) sobre el código de la página Web.

El contenido dinámico se genera:

- Bien en el servidor por medio de lenguajes como PHP, ASP o JSP, generalmente mediante accesos a bases de datos.
- Bien en el cliente gracias a lenguajes de script como JavaScript o a objetos incrustados en las páginas web con tecnologías como Flash o los Applets de Java.

1 Equivalentes accesibles

Prioridad 1

[1.1](#) Proporcione un texto equivalente para todo elemento no textual.

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

[6.3](#) Asegure 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.

Desactivación de objetos programados [1.1; 6.3 - Prioridad 1]

Todos los **objetos programados** deben llevar equivalentes accesibles.

Como el texto se considera accesible para casi todos los usuarios-as, se debe proporcionar un texto equivalente para todos los elementos no textuales.

Sin embargo, no es necesario que los equivalentes accesibles usados en primer lugar para los objetos programados sean textuales.

Por ejemplo, para una animación Flash puedes usar como alternativa una imagen, siempre y cuando sea accesible, es decir, tenga un texto alternativo; también, en caso de que el objeto flash funcione como un mecanismo de navegación con diversos enlaces, podrías utilizar como alternativa un mapa de imagen accesible.

Recuerda que siempre debes incluir un **equivalente textual final**, pero la primera alternativa para los objetos programados puede ser **cualquier elemento que sea accesible**.

A continuación, describiremos los equivalentes necesarios para los diferentes elementos programados:

Applets

Cuando programes applets pon el texto alternativo en el atributo "**alt**" y dentro del contenido del elemento APPLET (de forma similar al elemento OBJECT). De esta forma todos los navegadores tendrán acceso al texto alternativo.

Ejemplo.

```
<applet code="nombreapplet.class"
width="300" height="300"
alt="Applet Java: descripción del applet">
  Aquí va la descripción del applet.
</applet>
```

Sin embargo, el elemento APPLET está **desaconsejado** desde HTML 4.0. y en su lugar deberías usar el elemento OBJECT.

Objetos añadidos con OBJECT

Si utilizas el elemento OBJECT para añadir los objetos programados, puedes proporcionar el equivalente textual en su contenido, ya que los navegadores muestran, en primer lugar, el objeto y en segundo lugar, su contenido.

A continuación te mostramos un ejemplo que describe cómo incluir un applet con OBJECT y proporcionarle un texto alternativo:

Ejemplo.

```
<object classid="java:nombreadpplet.class"
width="300" height="300">
  Aquí va la descripción del applet
</object>
```

La especificación de HTML permite anidar elementos OBJECT para proporcionar varias representaciones alternativas. Puedes comprobarlo en el siguiente código:

Ejemplo.

```
<object classid="java:nombreadpplet.class"
width="300" height="300">
  <object data="video.mpeg" type="video/mpeg">
    <object data="imagen.gif" type="image/gif">
      Aquí va la descripción de la imagen.
    </object>
  </object>
</object>
```

De ese modo, si el navegador no puede presentar el applet intentaría mostrar el vídeo en su lugar. En caso de que tampoco pudiese presentar el vídeo, usaría la imagen y, si tampoco pudiese, entonces mostraría el texto alternativo de la imagen (el contenido del último elemento OBJECT).

Scripts

El desarrollador-a puede proporcionar una descripción o funcionalidad alternativa a los scripts con el elemento NOSCRIPT. En caso de que el navegador no pueda mostrar los scripts, por no soportarlos o por estar desactivados, mostrará el contenido del elemento NOSCRIPT.

Por ejemplo, si utilizas un script para generar una barra de navegación puedes proporcionar los mismos enlaces en el elemento NOSCRIPT. De esta forma, si no se puede ejecutar el script, no se perderían los enlaces de navegación.

Ejemplo.

```
<script type="text/javascript"
  src="script/barranavegacion.js">
</script>
<noscript>
  <ul>
    <li><a href="#">Enlace 1</a></li>
    <li><a href="#">Enlace 2</a></li>
  </ul>
</noscript>
```

Objetos añadidos con EMBED

La etiqueta NOEMBED funciona de forma similar a la etiqueta NOSCRIPT. Sirve para proporcionar un contenido alternativo para los elementos incluidos mediante la etiqueta EMBED.

El elemento EMBED es propietario y está desaconsejado por el W3C.

Actualización de los equivalentes del contenido dinámico [6.2 - Prioridad 1]

Actualiza los equivalentes del contenido dinámico si quieres evitar que el documento deje de ser accesible.

El contenido alternativo o equivalente tiene que aportar la misma información o funcionalidad que el contenido original.

Por ejemplo, mediante un lenguaje de script de servidor, como PHP, se puede crear un contador de visitas sin que se actualice el equivalente textual, tal y como se muestra a continuación:



Ejemplo desaconsejado.

```
<p>
  Numero de visitas:
  "
    alt="Contador">
</p>
```

Con <?include("contador.php")?> se añade el código en php que se encarga de generar el nombre de la imagen. Este contador, tal y como está hecho, presenta varios problemas:

- El texto alternativo no aporta la misma información que la imagen.
- Cuando cambia la imagen no se actualiza su texto alternativo.

Accesibilidad y Usabilidad Web

Para solucionarlo habría que colocar en el texto alternativo el mismo número que aparece en la imagen y actualízalo cada vez que cambie.

Ejemplo.

```
<p>  
  Numero de visitas: <?include("contador.php")?>  
</p>
```

Ahora, en contador php se generaría tanto la imagen como su texto alternativo correspondiente indicando el mismo número.

2 Contenido dinámico

Prioridad 1

[7.1](#) Hasta que las aplicaciones de usuario permitan controlarlo, evite provocar destellos en la pantalla.

No provocar destellos en la pantalla [7.1 - Prioridad 1]

Los desarrolladores-as deben **evitar los destellos** en la pantalla hasta que las aplicaciones de usuario-a proporcionen mecanismos adecuados para controlar este efecto potencialmente peligroso para cierto grupo de personas.

Las personas con **epilepsia fotosensitiva** pueden sufrir ataques desencadenados por parpadeos o destellos que oscilen entre los 4 y los 59 destellos por segundo (hertzios), con un nivel máximo de sensibilidad a los 20 destellos por segundo, así como con los cambios rápidos de oscuridad a iluminación (como las luces estroboscópicas).

Algunos elementos que pueden provocar destellos en la pantalla son:

- Scripts.
- Spplets.
- Elementos añadidos con OBJECT (Flash, applets, etc.).
- Imágenes en formato GIF.
- Refrescos rápidos de contenido.
- Objetos incrustados con el elemento desaconsejado EMBED.

Prioridad 2

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

[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).

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

Contenido dinámico accesible [6.5 - Prioridad 2]

Las páginas deben ser accesibles cuando los navegadores tengan desactivados (o no soporten) los scripts u otros elementos dinámicos.

Contenido generado

Evita generar contenido sobre la marcha (*on the fly*) en el lado de cliente mediante scripts, ya que si el navegador no los ejecuta no se mostrará el contenido generado.

Sin embargo, puedes mostrar u ocultar el contenido del documento HTML combinando scripts y hojas de estilo. De esta manera, se puede seguir accediendo al contenido aunque no se ejecuten los scripts.

También puedes usar scripts de servidor para generar contenido, pues el agente de usuario-a recibe el contenido HTML tal y como ha sido generado en el mismo.

Scripts en los enlaces

Al mismo tiempo, evita usar javascript como URI en los enlaces. En caso de no soportarse los scripts, o de tenerlos desactivados, los enlaces dejarían de funcionar.

Evita los enlaces como el que mostramos en el ejemplo:

Ejemplo desaconsejado.

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

Usa JavaScript en los manejadores de eventos como onclick, onfocus, onkeypress... en lugar de en el atributo "href" de los enlaces.

Por ejemplo, si vas a utilizar JavaScript para abrir un enlace en una nueva ventana, hazlo de forma similar a la que te mostramos a continuación:

Ejemplo.

```
<script type="text/javascript">
  var v=null;
  function abrirURL(url) {
    if (v) v.close();
    v = window.open(url, 'v',
      "height=400, width=500,
      resizable=1, scrollbars=1");
    v.focus();
    return false;
  }
</script>
...
<a target="_blank"
  href="http://www.fundacionctic.org"
  onclick="return abrirURL(this.href);"
  title="El enlace se abre en nueva ventana:
  Página de la Fundación CTIC">
  Fundación CTIC
</a>
```

De esta forma, el enlace funcionará al tener un "href" correcto y se abrirá en una nueva ventana aunque no se soporte JavaScript, gracias al atributo target="blank".

Sin embargo, en XHTML 1.0 "Strict" y en XHTML 1.1 no se podría abrir una nueva ventana sin usar JavaScript al no estar definido el atributo "target".

Evitar el parpadeo del contenido [7.2 - Prioridad 2]

El **parpadeo** del contenido puede **dificultar la lectura** a las personas con **discapacidades cognitivas o visuales**.

Algunos elementos que pueden provocar parpadeo del contenido son

- Scripts.
- Applets.
- Elementos añadidos con OBJECT
- Imágenes en formato GIF.
- Refrescos rápidos de contenido.
- También los elementos desaconsejados o propietarios EMBED y BLINK.

El elemento BLINK se usa para resaltar un texto haciéndolo intermitente y se puede sustituir por STRONG, EM, SPAN con propiedades CSS o elementos de encabezado, entre otros.

Si ves absolutamente necesario que el elemento parpadee, utiliza en la hoja de estilos la propiedad "text-decoration: blink" para que el usuario-a elimine el parpadeo desactivando la CSS.

Movimiento del contenido [7.3 - Prioridad 2]

Como puedes leer en las WCAG:

Algunas personas con discapacidades cognitivas o visuales son incapaces de leer textos que se mueven con la suficiente rapidez o en absoluto. El movimiento puede también distraer de tal manera que el resto de la página se vuelve ilegible para las personas con discapacidades cognitivas. Los lectores de pantalla son incapaces de leer textos móviles. Las personas con discapacidades físicas podrían no ser capaces de moverse tan rápida o certeramente como para interactuar con objetos móviles.

Los **scripts**, **applets**, elementos añadidos con **OBJECT** o las imágenes en formato **GIF** pueden provocar movimiento del contenido así como los elementos desaconsejados o propietarios **EMBED** y **MARQUEE**.

El elemento **MARQUEE** se utiliza para resaltar un texto y se puede sustituir por **STRONG**, **EM**, **SPAN** con propiedades CSS o elementos de encabezado, entre otros.

Si introduces contenido en movimiento ofrece una forma de pararlo y dejarlo estático.

Si, por ejemplo, utilizas Javascript para el movimiento puedes pararlo desactivandolo en el navegador.

3 Independencia de dispositivo

Prioridad 2

[6.4](#) Para los scripts y applets, asegúrese de que los manejadores de eventos sean independientes del dispositivo de entrada.

[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].

[9.2](#) Asegúrese de que cualquier elemento que tiene su propia interfaz pueda manejarse de forma independiente del dispositivo.

[9.3](#) Para los "scripts", especifique manejadores de evento lógicos mejor que manejadores de evento dependientes de dispositivos.

Independencia de dispositivo de entrada [6.4 - Prioridad 2]

La **independencia de dispositivo** garantiza que los usuarios-as interactúen con el documento independientemente del dispositivo de entrada usado.

Un documento o aplicación con el que sólo se pueda interactuar mediante el ratón, estará restringiendo el acceso a todas las personas que no lo puedan usar, bien porque tengan alguna discapacidad que les incapacite para ello o porque no lo tengan disponible.

Por ejemplo, las personas con discapacidades visuales no pueden utilizar dispositivos de apuntamiento y recurren a otro tipo de recursos como el teclado o la voz.

Las páginas que permiten la interacción a través del teclado suelen ser también accesibles a través de una entrada de voz u otro tipo de dispositivos diferentes del ratón.

Elementos con interfaz independiente de dispositivo [9.2 - Prioridad 2]

Como desarrollador-a debes valorar que muchas personas no utilizan el ratón ni navegadores visuales. Si diseñas una página para que sea accesible mediante teclado, también debe serlo para muchos otros tipos de dispositivos.

Algunos **consejos** que proporcionan mayor funcionalidad son:

- **Equivalentes textuales** en los mapas de imagen de cliente y enlaces redundantes en los de servidor.
- **Atajos de teclado** que permitan, mediante combinaciones de teclas, navegar por los enlaces o controles de formulario.
- **Órdenes de tabulación** adecuadas que describan un orden de navegación entre los enlaces o los controles de formulario.

- Los **elementos incrustados** cuya interfaz no se controle mediante HTML (como applets, flash o reproductores multimedia) se deben hacer directamente accesibles o se debe proporcionar una alternativa accesible.

Objetos programados directamente accesibles [8.1 - Prioridad 2]

Siempre que puedas construye los elementos de programación (scripts, applets, flash...) directamente accesibles o compatibles con las ayudas técnicas usando las características de accesibilidad que proporcionen esas características.

Esto es especialmente relevante siempre que la funcionalidad aportada sea importante y no se presente en otro lugar.

Por ejemplo, si un applet precisa una interacción con el usuario que es imposible reproducir en un formato alternativo, ha de hacerse directamente accesible.

Algunas tecnologías como los Applets o Flash incluyen ciertas características de accesibilidad que deben emplearse a la hora de utilizarlas en un sitio Web accesible.

Muchas de las recomendaciones para la creación de objetos programados directamente accesibles son similares a las elaboradas para construir sitios Web accesibles.

Por ejemplo, Flash permite proporcionar texto alternativo a las imágenes; descripciones textuales de la banda visual o en formato de audio; mecanismos para que el usuario detenga las animaciones a voluntad; interacción independiente de dispositivo permitiendo el uso de teclado; teclas de acceso rápido; etc.

A continuación te mostramos dos enlaces interesantes:

- Información sobre la [accesibilidad en Flash y otros productos de Macromedia](#).
- Información sobre la [accesibilidad en Java](#).

Si quieres ver cómo puedes crear scripts directamente accesibles lee el siguiente apartado sobre el uso de manejadores de eventos lógicos.

Manejadores de eventos lógicos en los scripts [9.3 - Prioridad 2]

Como desarrollador-a te recomendamos que emplees manejadores de eventos a nivel de aplicación en vez de eventos de interacción con el usuario-a.

En HTML 4.01, los atributos al nivel de aplicación son "**onfocus**", "**onblur**" (lo opuesto a "onfocus"), y "**onselect**". Estos atributos son independientes del dispositivo, pero en los navegadores actuales se implementan como eventos específicos del teclado.

A continuación te mostramos cómo puedes reemplazar algunos eventos de ratón:

- "**onMouseOver**", reemplazar con "**onFocus**".
- "**onMouseOut**", reemplazar con "**onBlur**".

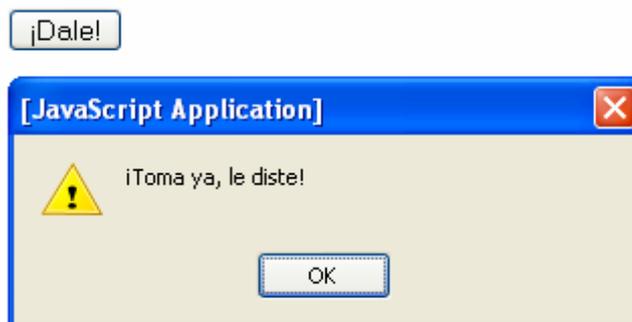
Accesibilidad y Usabilidad Web

Si es necesario que utilices atributos específicos de un dispositivo, proporciona mecanismos redundantes de interacción, es decir, especifica varios manejadores de eventos. Por ejemplo:

- "onMouseDown" usarlo con "onKeyDown".
- "onMouseUp" usarlo con "onKeyUp".
- "onClick" usarlo con "onKeyPress".

Evita el uso de **ondblclick** dado que no dispone de equivalente. Además, huye de la utilización de eventos dependientes de las coordenadas donde se encuentre el ratón.

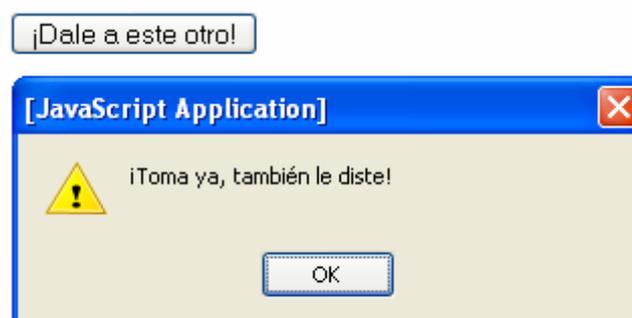
En el siguiente ejemplo puedes ver cómo mediante el evento onfocus se consigue que el botón se active, tanto si se pulsa con el ratón como si se accede a él por medio del tabulador:



Ejemplo.

```
<script type="text/javascript">
  function pop1() {
    alert ("¡Toma ya, le diste!") }
</script>
...
<button onfocus="pop1()">¡Dale!</button>
```

En el siguiente ejemplo se utilizan mecanismos redundantes de interacción, uno para el ratón y otro para el teclado. El botón se activa si se pulsa sobre él con el ratón o si se accede a él con el tabulador y se presiona la tecla de retorno.



Ejemplo.

```
<script type="text/javascript">
  function pop2() {
    alert ("¡Toma ya, también le diste!")
  }
</script>
...
<button onclick="pop2()" onkeypress="pop2()">
  ¡Dale a este otro!
</button>
```

Marcos

La actual implementación de los **marcos** (con los elementos **FRAMESET**, **FRAME**, e **IFRAME**) hace que sea problemática por varios motivos:

- 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 una **URI**. Esto es porque al navegar por una página con marcos y cambiar el contenido de los mismos la dirección Web no varía. Una vez que cambian los contenidos de un conjunto de marcos, el **URI** original ya no es válido.
- Se obliga al usuario-a a entrar por la declaración de frames, ya que en caso de no hacerlo sólo se vería la página interior sin los marcos. Al no mostrarse los marcos exteriores pueden **perderse los elementos de navegación** del sitio Web.
- Los **buscadores Web** puede que no entiendan las relaciones entre los marcos e indexarán las páginas de forma normal, sin atender si se trata de una página interior a un marco o no. Como consecuencia de esto, las páginas indexadas del sitio Web perderían su presentación original o sus elementos de navegación.
- La **apertura de enlaces** a páginas externas en los marcos de la página actual puede resultar demasiado molesto a los usuarios-as, pues da la sensación de que se está limitando la navegación o que impide que se abandone el sitio actual.
- En ocasiones, se produce **desorientación** o malestar en el usuario-a al abrir el contenido de un marco en una ventana nueva.
- Algunos **navegadores no soportan los marcos**, aunque en la actualidad es poco frecuente que ocurra esto.
- Algunos navegadores pueden tener problemas en caso de querer añadir la página a los **favoritos**.

1 Equivalentes textuales

Prioridad 1

- 1.1 Proporcione un texto equivalente para todo elemento no textual.
- 6.2 Asegúrese de que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico.
- 12.1 Titule cada marco para facilitar su identificación y navegación.

Contenido alternativo para marcos [1.1 - Prioridad 1]

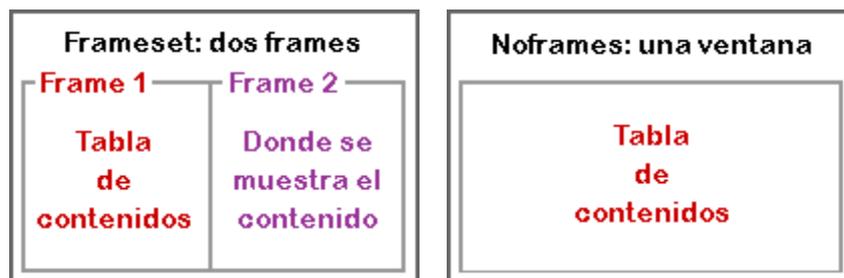
Algunos navegadores no soportan los marcos o están configurados para no mostrarlos, por lo que debes ofrecer una **alternativa** a los mismos.

Las alternativas a los marcos se realizan con el elemento **NOFRAMES**. En el contenido de este elemento debes incluir los elementos de navegación necesarios para moverse por el sitio Web así como los enlaces a las descripciones de los marcos.

Cuando utilices el elemento NOFRAMES evita informar al usuario-a sobre la antigüedad de su navegador ni le sugieras que lo actualice a una versión que sí soporte frames.

Los agentes de usuario-a que soportan marcos mostrarán los contenidos del elemento NOFRAMES cuando se configuren para no mostrar marcos, mientras que aquellos que no soporten marcos siempre mostrarán el contenido del elemento NOFRAMES.

En el siguiente código puedes ver un ejemplo.



Ejemplo

```
<frameset cols="50%, 50%"
title="Documento principal">
  <frame src="contenido.html"
  title="Donde se muestra el contenido">
  <frame src="toc.html"
  title="Tabla de contenidos">
</frameset>
<p>
  <a href="toc.html">Tabla de contenidos.</a>
</p>
<p>
```

```

    <a href="noframes.html">
    Versión sin marcos
    </a>
</p>
<p>
    <a href="descripcion_frames.html">
    Descripciones de los marcos.</a>
</p>
<!-- otros enlaces de navegación disponibles
en contenido.html también deberían estar
disponibles. -->
</noframes>
</frameset>

```

En el caso de los marcos en línea, insertados dentro de un bloque de texto con el elemento **IFRAME**, incluye el equivalente accesible dentro del contenido, tal y como se realiza en el elemento OBJECT.

El contenido del elemento IFRAME sólo se mostrará por los agentes de usuario-a que no soporten marcos o estén configurados para no mostrarlos.

Ejemplo

```

<iframe src="condiciones.html"
width="400" height="600" scrolling="auto">
<!-- Contenido alternativo para los navegadores
que no soporten o no muestren los marcos -->
<p>
    Antes de aceptar debe leer detenidamente las
    <a href="condiciones.html"
    title="Condiciones de contrato">
    condiciones del contrato.
    </a>
</p>
</iframe>

```

Títulos para los marcos [1.1 y 12.1 - Prioridad 1]

El atributo **"title"** describe de forma escueta la función o contenido de los marcos.

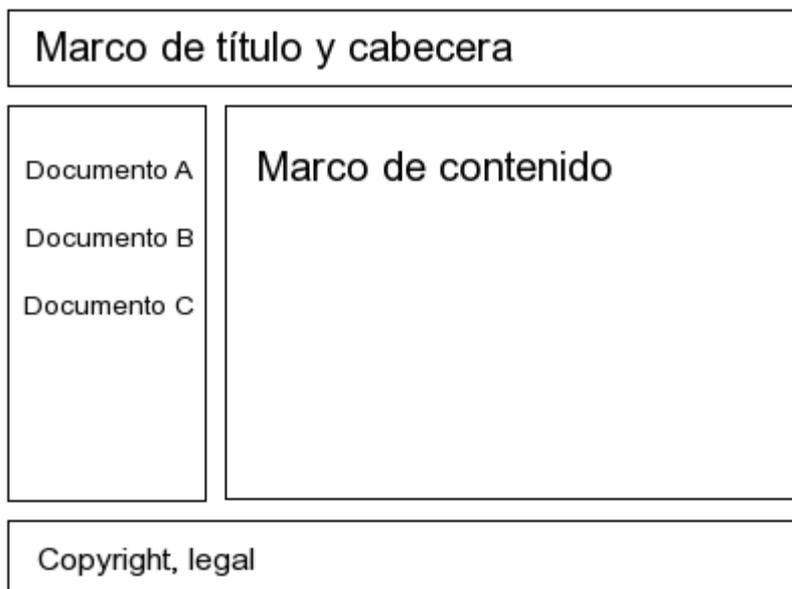
Ejemplo

```

<frameset cols="20%, 80%">
    <frame src="barra_nav.html"
    title="Barra de navegación">
    <frame src="contenido.html"
    title="Contenido">
</frameset>

```

Un ejemplo de una estructura con marcos más compleja que la anterior podría ser ésta:



Ejemplo

```
<frameset rows="20%,*,30%">
  <frame src="f1.htm"
    title="Marco de título y
    navegación principal">
  <frameset cols="20%,*,">
    <frame src="f2.htm"
      title="Menú de opciones principales">
    <frame src="f3.htm"
      title="Marco de contenido">
  </frameset>
  <frame src="f4.htm"
    title="Copyright y navegación final">
</frameset>
```

Actualización de los equivalentes textuales [6.2 - Prioridad 1]

Cuando cambies el contenido del marco modifica también su descripción. Ésto es imposible si se incluye directamente una imagen dentro del marco, por lo que el valor del atributo **src** debe ser siempre un **archivo HTML válido**.

Algunos ejemplos de documentos válidos son: ".htm", ".html", ".shtm", ".shtml", ".cfm", ".cfml", ".asp", ".php", ".cgi", ".pl", ".smil" o cualquier documento de destino que tenga un identificador público conocido al principio del mismo.

A continuación te mostramos un ejemplo de mala práctica:

Ejemplo desaconsejado

```
<frame name="marcoimagen" src="casa.gif"
title="Imagen de una casa">
```

Si un enlace inserta una nueva imagen en el marco, el título inicial deja de corresponderse con el contenido de la misma y ésta queda sin descripción al no proporcionarse un texto alternativo.

Ejemplo

```
<a target="marcoimagen" href="coche.gif">
  Ver coche
</a>
```

Una imagen puede ser el contenido del marco siempre que la incluyas en un documento HTML y hagas referencia a ese documento.

Ejemplo

```
<frame name="marcoimagen" src="casa.html"
title="Imágenes">

<!-- En casa.html -->

```

De esta forma el título del marco sigue siendo coherente a la vez que se puede proporcionar una descripción de la imagen.

Prioridad 2

[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.

Descripción de la relación entre los marcos [12.2 - Prioridad 2]

Si la función o la relación entre los marcos no queda clara con el título, proporciona una descripción de los mismos y de la relación entre ellos en una nueva página a la que se acceda a través del atributo **"longdesc"** de, al menos, uno de los marcos del FRAMESET.

Ejemplo

```
<frameset cols="20%, 80%">
  <frame src="barra_nav.html"
    title="Barra de navegación"
    longdesc=" frameset-desc.html">
  <frame src="contenido.html" title="Contenido">
</noframes>
  <!-- Contenido alternativo en caso de
    no soportarse los frames -->
  ...
</noframes>
</frameset>
```

Accesibilidad y Usabilidad Web

La descripción de los marcos del ejemplo anterior, en la página frameset-desc.html, podría ser:

- Barra de navegación: este marco contiene la barra de navegación con los enlaces más importantes del sitio Web.
- Contenido: en este marco es donde va situado el contenido principal de la página.

A continuación te presentamos otro ejemplo:



En este caso podrías describir los marcos de la siguiente manera:

Hay 4 frames. El frame superior contiene el título y la navegación principal. El frame lateral izquierdo contiene una lista de links a los apartados del nivel. El frame central es donde se presenta el contenido de la página. El frame inferior presenta el copyright, y la navegación secundaria.

Tablas

"Las tablas deberían utilizarse solamente para marcar la información tabular ("tablas de datos"). Los desarrolladores de contenidos deberían evitar usarlas para maquetar páginas ("tablas de composición"). Usar tablas para cualquier finalidad crea también especiales dificultades para los usuarios de lectores de pantalla.

Algunas aplicaciones de usuario permiten a los usuarios navegar entre las celdas de las tablas y acceder a los encabezamientos y otras informaciones de las celdas. A menos que marquemos apropiadamente las tablas, éstas no proporcionaran a la aplicación de usuario la información necesaria para ello.

Los siguientes puntos de verificación beneficiarán directamente a las personas que accedan a la tabla por medios auditivos (por ejemplo un lector de pantalla o un PC de automóvil), o a aquellos que sólo visualicen una parte de la página cada vez (Por ejemplo, los usuarios ciegos o de escasa visión que utilicen un sistema auditivo o un dispositivo braille u otros usuarios de dispositivos con pantallas pequeñas, etc.)."

Pautas WGAC del WAI

1 Tablas de datos

Prioridad 1

- 5.1 En las tablas de datos, identifique los encabezamientos de fila y columna.
- 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.

Identificar encabezamientos [5.1 - Prioridad 1]

Cuando presentes datos en forma tabular identifica los encabezamientos de fila y columna mediante el elemento **TH**. Revisa el siguiente ejemplo:

Medallero Olímpico

	Medallas de oro	Medallas de plata	Medallas de bronce
EEUU	22	26	18
Rusia	18	24	20
China	15	19	19

Ejemplo.

```
<table border="1">
<caption>Medallero Olímpico</caption>
<tr>
  <th></th>
  <th>Medallas de oro</th>
  <th>Medallas de plata</th>
  <th>Medallas de bronce</th>
</tr>
<tr>
  <th>EEUU</th>
  <td>22</td>
  <td>26</td>
  <td>18</td>
</tr>
<tr>
  <th>Rusia</th>
  <td>18</td>
  <td>24</td>
  <td>20</td>
</tr>
<tr>
  <th>China</th>
  <td>15</td>
  <td>19</td>
  <td>19</td>
</tr>
</table>
```

Tablas de datos complejas [5.2 - Prioridad 1]

En las tablas de datos complejas, con dos o más niveles lógicos de encabezamiento, debes asociar las celdas de encabezamiento con las de datos y dividir estructuralmente la información cuando sea necesario. Para ello dispones de los siguientes elementos o atributos:

- **THEAD** identifica grupos de encabezamientos (TH).
- **TFOOT** identifica pies de tabla.
- **TBODY** identifica grupos de filas de datos.
- **COLGROUP** y **COL** identifican grupos de columnas.
- Los atributos "**scope**", "**headers**" y "**axis**" sirven para etiquetar los elementos de tabla para que los futuros navegadores o ayudas técnicas puedan seleccionar datos filtrando por categorías.

A continuación profundizaremos en los elementos enumerados.

Elementos THEAD, TFOOT, TBODY

THEAD define un grupo de filas de encabezamiento de una tabla (TH). Sólo puedes emplear un elemento THEAD por tabla y debes situarlo después del elemento CAPTION y antes de TFOOT (si lo hubiese) o TBODY.

TFOOT delimita un grupo de filas de pie de tabla. Como en el caso de THEAD, sólo puedes utilizar un elemento TFOOT por tabla y debes colocarlo después de THEAD y antes de TBODY.

TBODY identifica grupos de filas de datos en una tabla. Este elemento divide las filas de datos en varios grupos de filas que comparten ciertas características. Su uso es opcional, pero si usas los elementos THEAD o TFOOT, entonces debes implementarlo obligatoriamente.

THEAD, TBODY y TFOOT permiten que, al imprimir tablas largas, los navegadores puedan presentar los encabezados y pies de página en todas las páginas.

Además, los navegadores podrían presentar tablas largas de forma que la cabecera y el pie se mantienen estáticos en pantalla mientras se desplaza el contenido de TBODY (las celdas de datos).

TBODY permite agrupar las filas de datos en varios grupos y darles un estilo diferente, facilitando así su visualización.

El elemento TFOOT añade notas aclaratorias en el pie de la tabla.

En el siguiente código te mostramos un ejemplo sencillo de utilización de estos elementos, basándonos en el ejemplo anterior.

Medallero Olímpico

	Medallas de oro	Medallas de plata	Medallas de bronce
EEUU	22	26	18
Rusia	18	24	20
China	15	19	19 (*)
(*) Medallas pendientes de revisión por un comité antidopaje			

Ejemplo.

```
<table border="1">
<caption>Medallero Olímpico</caption>
<thead>
  <tr>
    <th></th>
    <th>Medallas de oro</th>
    <th>Medallas de plata</th>
    <th>Medallas de bronce</th>
  </tr>
</thead>
<tfoot>
  <tr>
    <td colspan="4">(*) Medallas pendientes de
      revisión por un comité antidopaje</td>
  </tr>
</tfoot>
<tbody>
  <tr>
    <th>EEUU</th>
    <td>22</td>
    <td>26</td>
    <td>18</td>
  </tr>
  <tr>
    <th>Rusia</th>
    <td>18</td>
    <td>24</td>
    <td>20</td>
  </tr>
  <tr>
    <th>China</th>
    <td>15</td>
    <td>19</td>
    <td>19 (*)</td>
  </tr>
</tbody>
</table>
```

Elementos COLGROUP, COL

COLGROUP define un grupo de columnas en una tabla y debes situarlo después de **CAPTION** y antes de **THEAD**. Además identifica más fácilmente estilos para columnas o grupos de columnas.

El atributo "**span**" especifica el número de columnas que contendrá el grupo. Por otra parte, **COLGROUP** puede contener elementos **COL** que especifiquen atributos para columnas individuales. También indican el número de columnas que contendrá el grupo. Por lo tanto, si empleas elementos **COL** no uses el atributo "span".

Por ejemplo, para la tabla anterior, puedes usar **COLGROUP** para especificar la alineación del texto en las columnas de datos (en vez de estilos sobre TD):

```
<COLGROUP span="3" align="center"/>
```

Si, además, deseas dar estilos visuales adicionales a cada columna, usa elementos **COL**, como muestra el siguiente ejemplo. El primer **COLGROUP** hace referencia a la primera columna de encabezado y el segundo a las columnas de datos.

Medallero Olímpico

	Medallas de oro	Medallas de plata	Medallas de bronce
EEUU	22	26	18
Rusia	18	24	20
China	15	19	19
(*) Medallas pendientes de revisión por un comité antidopaje			

El código de esta tabla quedaría de la siguiente forma:

Ejemplo.

```
<table border="1">
<caption>Medallero Olímpico</caption>
<colgroup align="left">
<colgroup align="center">
  <col style="width:15em">
  <col style="width:10em">
  <col style="width:5em">
</colgroup>
<thead>
<tr>
  <th></th>
  <th>Medallas de oro</th>
  <th>Medallas de plata</th>
  <th>Medallas de bronce</th>
</tr>
</thead>
<tfoot>
<tr>
  <td colspan="4">(*) Medallas pendientes de
```

```

    revisión por un comité antidopaje</td>
  </tr>
</tfoot>
<tbody>
  <tr>
    <th>EEUU</th>
    <td>22</td>
    <td>26</td>
    <td>18</td>
  </tr>
  <tr>
    <th>Rusia</th>
    <td>18</td>
    <td>24</td>
    <td>20</td>
  </tr>
  <tr>
    <th>China</th>
    <td>15</td>
    <td>19</td>
    <td>19</td>
  </tr>
</tbody>
</table>

```

A continuación te mostramos un ejemplo más complejo del uso de COLGROUP y los elementos THEAD y TBODY.

PAGINAS DE CODIGOS SOPORTADAS POR MICROSOFT WINDOWS

ID Code-Page	Nombre	ACP OEMCP	Windows	Windows	Windows
			NT 3.1	NT 3.51	95
1200	Unicode (BMP de ISO/IEC-10646)		X	X	*
1250	Windows 3.1 Europa del Este	X	X	X	X
1251	Windows 3.1 Cirílico	X	X	X	X
1252	Windows 3.1 US (ANSI)	X	X	X	X
1253	Windows 3.1 Griego	X	X	X	X
1254	Windows 3.1 Turco	X	X	X	X
1255	Hebreo	X			X
1256	Árabe	X			X
1257	Báltico	X			X
1361	Coreano (Johab)	X		**	X
437	MS-DOS Estados Unidos	X	X	X	X
708	Árabe (ASMO 708)	X			X
709	Árabe (ASMO 449+, BCON V4)	X			X
710	Árabe (Árabe Transparente)	X			X
720	Árabe (ASMO Transparente)	X			X

Ejemplo.

```

<table border="2" frame="hsides" rules="groups"

```

```
summary="Páginas de códigos soportadas por las diferentes
versiones de MS Windows.">
<caption>PAGINAS DE CODIGOS SOPORTADAS POR
MICROSOFT WINDOWS</caption>
<colgroup align="center">
<colgroup align="left">
<colgroup align="center" span="2">
<colgroup align="center" span="3">
<thead valign="top">
  <tr>
    <th>ID<br>Code-Page</th>
    <th>Nombre</th>
    <th>ACP</th>
    <th>OEMCP</th>
    <th>Windows<br>NT 3.1</th>
    <th>Windows<br>NT 3.51</th>
    <th>Windows<br>95</th>
  </tr>
</thead>
<tbody>
  <tr><td>1200</td><td>Unicode (BMP de ISO/IEC-10646)</td>
    <td></td><td></td><td>X</td><td>X</td><td>*</td></tr>
  <tr><td>1250</td><td>Windows 3.1 Europa del Este</td>
    <td>X</td><td></td><td>X</td><td>X</td><td>X</td></tr>
  <tr><td>1251</td><td>Windows 3.1 Cirílico</td><td>X</td>
    <td></td><td>X</td><td>X</td><td>X</td></tr>
  <tr><td>1252</td><td>Windows 3.1 US (ANSI)</td>
    <td>X</td><td></td><td>X</td><td>X</td><td>X</td></tr>
  <tr><td>1253</td><td>Windows 3.1 Griego</td>
    <td>X</td><td></td><td>X</td><td>X</td><td>X</td></tr>
  <tr><td>1254</td><td>Windows 3.1 Turco</td>
    <td>X</td><td></td><td>X</td><td>X</td><td>X</td></tr>
  <tr><td>1255</td><td>Hebreo</td>
    <td>X</td><td></td><td></td><td></td><td>X</td></tr>
  <tr><td>1256</td><td>Árabe</td>
    <td>X</td><td></td><td></td><td></td><td>X</td></tr>
  <tr><td>1257</td><td>Báltico</td><td>X</td>
    <td></td><td></td><td></td><td>X</td></tr>
  <tr><td>1361</td><td>Coreano (Johab)</td>
    <td>X</td><td></td><td></td><td></td><td>*</td><td>X</td></tr>
</tbody>
<tbody>
  <tr><td>437</td><td>MS-DOS Estados Unidos</td>
    <td></td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
  <tr><td>708</td><td>Árabe (ASMO 708)</td>
    <td></td><td>X</td><td></td><td></td><td>X</td></tr>
  <tr><td>709</td><td>Árabe (ASMO 449+, BCON V4)</td>
    <td></td><td>X</td><td></td><td></td><td>X</td></tr>
  <tr><td>710</td><td>Árabe (Árabe Transparente)</td>
    <td></td><td>X</td><td></td><td></td><td>X</td></tr>
  <tr><td>720</td><td>Árabe (ASMO Transparente)</td>
    <td></td><td>X</td><td></td><td></td><td>X</td></tr>
</tbody>
</table>
```

El atributo *rules="groups"* usado en el elemento TABLE indica que sólo se dibujen los bordes de tabla correspondientes a grupos. De esta forma la información está más estructurada, facilitando así su comprensión.

Por otra parte, el atributo *frame="hsides"* indica que solo se dibujen las líneas horizontales del marco exterior de la tabla, es decir, el borde superior y el inferior.

Atributos "headers", "scope"

Estos atributos asocian las celdas de encabezamiento y las de datos.

El atributo "headers" se usa en las celdas de datos (TD) e identifica la lista de celdas de encabezamiento que proporcionan información a la celda de datos actual.

Para referirte a las celdas de encabezamiento, emplea el valor del atributo "id" usado en los elementos TH correspondientes.

A continuación te proponemos el ejemplo del medallero olímpico para mostrar cómo se relacionan las celdas de datos y de encabezamiento mediante "id" y "headers".

Ejemplo.

```
<table border="1">
<caption>Medallero Olímpico</caption>
<colgroup align="left">
<colgroup span="3" align="center">
<thead>
  <tr>
    <th></th>
    <th id="h1">Medallas de oro</th>
    <th id="h2">Medallas de plata</th>
    <th id="h3">Medallas de bronce</th>
  </tr>
</thead>
<tfoot>
  <tr>
    <td colspan="4">(*) Medallas pendientes de
    revisión por un comité antidopaje</td>
  </tr>
</tfoot>
<tbody>
  <tr>
    <th id="h4">EEUU</th>
    <td headers="h4 h1">22</td>
    <td headers="h4 h2">26</td>
    <td headers="h4 h3">18</td>
  </tr>
  <tr>
    <th id="h5">Rusia</th>
    <td headers="h5 h1">18</td>
    <td headers="h5 h2">24</td>
    <td headers="h5 h3">20</td>
  </tr>
  <tr>
    <th id="h6">China</th>
    <td headers="h6 h1">15</td>
    <td headers="h6 h2">19</td>
    <td headers="h6 h3">19</td>
  </tr>
</tbody>
</table>
```

De esta forma, si utilizásemos un lector de pantalla, podría leer los datos de forma relacionada:

Ejemplo.

```
Caption: Medallero Olímpico
EEUU , Medallas de oro = 22 ,
EEUU , Medallas de plata = 26 ,
EEUU , Medallas de bronce = 18
Rusia , Medallas de oro = 18 ,
Rusia , Medallas de plata = 24 ,
Rusia , Medallas de bronce = 20
China , Medallas de oro = 15 ,
China , Medallas de plata = 19 ,
China , Medallas de bronce = 19
```

Si no identificamos los encabezamientos ni relacionamos los datos, los lectores de pantalla leen la tabla linealmente, haciendo muy difícil o imposible su interpretación:

Ejemplo.

```
Caption: Medallero Olímpico
Medallas de oro Medallas de plata Medallas de bronce
EEUU 22 26 18 Rusia 18 24 20 China 15 19 19
```

El orden usado para identificar los encabezamientos influye en el orden de lectura de los datos. En el ejemplo anterior se usa el siguiente orden:

Ejemplo.

```
<th id="h4">EEUU</th>
<td headers="h4 h1">22</td>
<td headers="h4 h2">26</td>
<td headers="h4 h3">18</td>
```

También se podría haber especificado otro orden para identificar los encabezamientos:

Ejemplo.

```
<th id="h4">EEUU</th>
<td headers="h1 h4">22</td>
<td headers="h2 h4">26</td>
<td headers="h3 h4">18</td>
```

De esta forma, un lector de pantalla podría leer los encabezados en el orden indicado en headers:

Ejemplo.

```
Medallas de oro , EEUU = 22 ,
Medallas de plata , EEUU = 26 ,
Medallas de bronce , EEUU = 18
```

Por otra parte, el atributo "**scope**", definido para encabezamientos, se utiliza en tablas de datos sencillas en lugar de los atributos "id" y "headers". Este elemento indica el conjunto de celdas de datos para las cuales el elemento de encabezado actual proporciona información. Los valores que puede tomar son:

- **row** (fila): Información de encabezado para el resto de la fila que la contiene.
- **col**: Información de encabezado para el resto de la columna.
- **rowgroup**: Información de encabezado para el resto del grupo de filas.
- **colgroup**: Información de encabezado para el resto del grupo de columnas.

A continuación puedes ver el ejemplo del medallero olímpico con el uso de "scope":

Ejemplo.

```
<table border="1">
<caption>Medallero Olímpico</caption>
<colgroup align="left">
<colgroup span="3" align="center">
<thead>
<tr>
<th></th>
<th scope="col">Medallas de oro</th>
<th scope="col">Medallas de plata</th>
<th scope="col">Medallas de bronce</th>
</tr>
</thead>
<tfoot>
<tr>
<td colspan="4">(*) Medallas pendientes de
revisión por un comité antidopaje</td>
</tr>
</tfoot>
<tbody>
<tr>
<th scope="row">EEUU</th>
<td>22</td>
<td>26</td>
<td>18</td>
</tr>
<tr>
<th scope="row">Rusia</th>
<td>18</td>
<td>24</td>
<td>20</td>
</tr>
<tr>
<th scope="row">China</th>
<td>15</td>
<td>19</td>
<td>19</td>
</tr>
```

```
</tbody>
</table>
```

Atributo "axis"

Este atributo identifica categorías conceptuales, como ejes en un espacio n-dimensional. En el siguiente ejemplo te presentamos los datos de los gastos de un viaje estructurados en varias categorías:

- Categoría "gastos": comidas, hotel, transportes
- Categoría "lugar": San José, Seattle.
- Categoría "fecha": 25-ago-97; 26-ago-97; 27-ago-97; 28-ago-97

Cada categoría funcionaría como un eje de un espacio tridimensional. Fíjate en la tabla que te mostramos a continuación:

Gastos de viaje

	Comidas	Hotel	Transportes
San Jose			
25-ago-97	37,74	112,00	45,00
26-ago-97	27,28	112,00	45,00
Seattle			
27-ago-97	96,25	109,00	36,00
28-ago-97	35,00	109,00	36,00

El código sería el siguiente:

Ejemplo.

```
<table border="1">
<caption>Gastos de viaje</caption>
<tr>
  <th></th>
  <th id="header1" axis="gastos">Comidas</th>
  <th id="header2" axis="gastos">Hotel</th>
  <th id="header3" axis="gastos">Transportes</th>
</tr>
<tr>
  <th id="header4" axis="lugar">San Jose</th>
  <th></th> <th></th>
  <th></th>
</tr>
<tr>
  <td id="header5" axis="fecha">25-ago-97</td>
  <td headers="header4 header5 header1">37,74</td>
  <td headers="header4 header5 header2">112,00</td>
  <td headers="header4 header5 header3">45,00</td>
</tr>
<tr>
```

```

<td id="header6" axis="fecha">26-ago-97</td>
<td headers="header4 header6 header1">27,28</td>
<td headers="header4 header6 header2">112,00</td>
<td headers="header4 header6 header3">45,00</td>
</tr>
<tr>
<th id="header7" axis="lugar">Seattle</th>
<th></th> <th></th>
<th></th>
</tr>
<tr>
<td id="header8" axis="fecha">27-ago-97</td>
<td headers="header7 header8 header1">96,25</td>
<td headers="header7 header8 header2">109,00</td>
<td headers="header7 header8 header3">36,00</td>
</tr>
<tr>
<td id="header9" axis="date">28-ago-97</td>
<td headers="header7 header9 header1">35,00</td>
<td headers="header7 header9 header2">109,00</td>
<td headers="header7 header9 header3">36,00</td>
</tr>
</table>

```

De ese modo, un lector de pantalla podría leer la información de la siguiente manera:

Ejemplo.

Caption: Gastos de viaje

```

San Jose = 25-ago-97 ,
  San Jose , 25-ago-97 , Comidas = 37,74 ,
  San Jose , 25-ago-97 , Hotel = 112,00 ,
  San Jose , 25-ago-97 , Transportes = 45,00
San Jose = 26-ago-97 ,
  San Jose , 26-ago-97 , Comidas = 27,28 ,
  San Jose , 26-ago-97 , Hotel = 112,00 ,
  San Jose , 26-ago-97 , Transportes = 45,00
Seattle = 27-ago-97 ,
  Seattle , 27-ago-97 , Comidas = 96,25 ,
  Seattle , 27-ago-97 , Hotel = 109,00 ,
  Seattle , 27-ago-97 , Transportes = 36,00
Seattle = 28-ago-97 ,
  Seattle , 28-ago-97 , Comidas = 35,00 ,
  Seattle , 28-ago-97 , Hotel = 109,00 ,
  Seattle , 28-ago-97 , Transportes = 36,00

```

Si quieres profundizar en el uso de las tablas, sus elementos y atributos puedes revisar la sección sobre [tablas en la especificación HTML](#) del W3C.

Prioridad 3

[5.5](#) Proporcione resúmenes de las tablas.

[5.6](#) Proporcione abreviaturas para las etiquetas de encabezamiento.

Proporcionar resúmenes de las tablas [5.5 - Prioridad 3]

Proporciona información sobre el contenido o la naturaleza de una tabla mediante el atributo **"title"** del elemento TABLE o con el elemento **CAPTION**, el cual debe ir situado después de la etiqueta <TABLE>.

La diferencia entre "title" y CAPTION es que el primero asigna un título a la tabla que estará disponible para los lectores de pantalla pero que no se mostrará en los navegadores visuales (en algunos casos se muestra como tooltip o mensaje emergente). Por su parte, con el elemento CAPTION el título sí se mostrará, generalmente en un texto centrado sobre la tabla.

En los ejemplos anteriores de las tablas puedes revisar la utilización del elemento CAPTION. Además recuerda que es innecesario usar "title" y CAPTION a la vez, con usar uno de los dos elementos es suficiente.

También es necesario ampliar la información dada mediante CAPTION o "title", proporcionando un resumen con el atributo **"summary"**. Los resúmenes son especialmente útiles para los usuarios con discapacidades visuales.

En el resumen incluye información sobre las relaciones entre las celdas, sobre todo en las tablas con una estructura compleja, encabezamientos anidados o celdas que ocupan varias filas o columnas.

Por ejemplo, para la tabla del medallero olímpico podrías elaborar un pequeño resumen como éste:

Ejemplo.

```
<table summary="Esta tabla muestra el número y
tipo de medallas conseguidas por cada país en las
últimas olimpiadas.">
```

Proporcionar abreviaturas en los encabezamientos [5.6 - Prioridad 3]

En los elementos TH puedes usar el atributo **"abbr"** para proporcionar abreviaturas a los encabezamientos. De esta forma, los lectores de pantalla mostrarán las abreviaturas de la cabecera correspondiente a una celda (si se ha usado "id", "header" o "scope") en vez de leer el texto completo de la cabecera.

Así, reducimos el tiempo de lectura de los datos de una tabla y evitamos las repeticiones tediosas de textos largos.

Por ejemplo, para la tabla del ejemplo del medallero, puedes emplear las siguientes abreviaturas:

Ejemplo.

```
<tr>
  <th></th>
  <th id="h1" abbr="Oro">Medallas de oro</th>
  <th id="h2" abbr="Plate">Medallas de plata</th>
  <th id="h3" abbr="Bronce">Medallas de bronce</th>
</tr>
```

Así, cuando un lector de pantalla lea la información correspondiente a las celdas leerá "Oro", "Plata" o "Bronce" y el valor de la celda, en vez de "Medallas de Oro", "Medallas de Plata", "Medallas de Bronce".

Ejemplo.

```
EEUU , Oro = 22 ,
EEUU , Plata = 26 ,
EEUU , Bronce = 18
```

2 Tablas de maquetación

Prioridad 2

[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).

[5.4](#) Si se utiliza una tabla para maquetar, no utilice marcadores estructurales para realizar un efecto visual de formato.

Evita la utilización de tablas para maquetar [5.3 - Prioridad 2]

Cuando las aplicaciones de usuario-a soporten la colocación mediante hojas de estilo, evita la utilización de tablas para maquetar. Como en la actualidad todos los navegadores gráficos soportan el posicionamiento mediante CSS, deberías usar las tablas principalmente para presentar datos tabulares.

Los marcadores estructurales y hojas de estilo te permitirán controlar la presentación y maquetación del documento.

En ese sentido te recomendamos que maquetes todo el documento mediante CSS. Pero, si ésto es imposible, intenta combinar el uso de CSS con las principales tablas.

Evita caer en un anidamiento excesivo y emplea las tablas estrictamente necesarias. Por ejemplo, podrías usar una tabla para lograr una maquetación principal como la de la siguiente imagen:



El orden de lectura de una tabla con esta maquetación sería el siguiente:

- Cabecera
- Barra de navegación
- Lateral izquierdo
- Contenido principal
- Lateral derecho
- Pie de página

Sin embargo, puedes elaborar una maquetación como ésta mediante hojas de estilo de forma sencilla.

Las hojas de estilo proporcionan una serie de propiedades que, mediante el empleo de capas y su posicionamiento relativo o absoluto, controlan la situación de cualquier elemento del código. Revísalo en el código que te mostramos como ejemplo.

Ejemplo.

```
<html>
<head>
  <title>Tableless</title>
  <style type="text/css">
    body{
      font-size: 1em;
      font-family: verdana,arial,Helvetica;
      text-align: center;}
  </style>
</head>
```

```

#cabecera{background-color:#eedddd;
  height: 4em;}
#barra_navegacion{background-color:#dddddd;
  margin-top: 0.1em;
  margin-bottom: 0.1em;}
#izq{
  background-color:#ddeedd;
  width: 20%;
  float:left;
  min-height:20em;}
#centro{
  background-color:#ffffff;
  width: 60%;
  float:left;
  min-height:20em;}
#dcha{
  background-color:#ddddee;
  width: 20%;
  float:left;
  min-height:20em;}
#pie_pagina{
  clear:both;
  color: white;
  background-color:#666;}
</style>
</head>
<body>
  <div id="cabecera">CABECERA</div>
  <div id="barra_navegacion">Barra de navegacion</div>
  <div id="izq">Lateral izquierdo</div>
  <div id="centro">Contenido Principal</div>
  <div id="dcha">Lateral derecho</div>
  <div id="pie_pagina">Pie de página</div>
</body>
</html>

```

Si quieres **crear un nuevo portal**, evita el uso de tablas de maquetación. En este caso, empiezas desde cero y puedes realizar el portal con unos criterios y objetivos claros en cuanto a accesibilidad y usabilidad.

Sin embargo, si quieres **rediseñar un portal existente**, adecuando dicho portal a un determinado nivel de accesibilidad, puede resultarte demasiado complejo o costoso.

Si empleas las tablas para maquetar, asegúrate de que se alineen correctamente, es decir, que la información siga siendo comprensible cuando se lea de forma lineal. Las celdas deben incluir elementos estructurales (que creen párrafos, encabezamientos, listas, etc.) de modo que la página tenga sentido al ser alineada.

Una forma sencilla de ver cómo se alinea una tabla de maquetación es eliminando del código HTML todas las etiquetas relativas a la tabla (TABLE, TR, TD...). Cuando una tabla se alinea, las celdas se leen en el orden en el que aparecen en el código HTML.

Además puedes recurrir a [Tablin](#), un servicio gratuito de W3C para comprobar cómo se alinean las tablas, tanto si son de maquetación como de datos. Hay disponible una versión online y una descargable que puedes utilizar en línea de comandos.

El resultado de la utilización de **Tablin** para linealizar la tabla de la página del ejemplo anterior es el siguiente:

CABECERA Barra de navegación Lateral izquierdo Contenido principal Lateral derecho Pie de página

No usar marcadores estructurales para dar formato [5.4 - Prioridad 2]

Cuando utilices tablas para maquetar, no uses elementos estructurales para conseguir efectos visuales o dar formato.

Por ejemplo, evita recurrir a la utilización del elemento TH para conseguir que el texto quede centrado y en negrita cuando en realidad no se trata de un encabezado de fila o columna de datos. En su lugar, emplea las hojas de estilo como vemos en el siguiente código:

Ejemplo.

```
<td>
  <div style="font-weight: bold">Texto en negrita</div>
</td>
```

Recuerda que sólo deberías usar TH para las cabeceras de las tablas, por lo que el siguiente código ejemplifica una mala práctica:

Ejemplo desaconsejado.

```
<th>Texto en negrita</th>
```

Prioridad 3

[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.

Texto lineal alternativo para tablas [10.3 - Prioridad 3]

Algunos lectores de pantalla antiguos que no reconocen el código HTML tienen problemas para leer el contenido de las tablas. En vez de leer el contenido celda a celda, siguen una línea de texto en la pantalla. Por ejemplo, en la siguiente tabla cada celda contiene un conocido refrán:

1 Cuando las barbas de tu vecino veas pelar	2 Más vale pájaro en mano	3 No hay mal que por bien
---	---------------------------------	---------------------------------

pon las tuyas a remojar	que ciento volando	no venga
----------------------------	--------------------	----------

Un lector de pantalla antiguo podría leerlo de esta forma:

Ejemplo.

```
1 2 3
Cuando las barbas Más vale No hay mal
de tu vecino veas pelar pájaro en mano que por bien
pon las tuyas que ciento volando no venga
a remojar.
```

En estos casos, proporciona el texto lineal alternativo en el orden correcto de lectura, tal y como vemos a continuación:

Ejemplo.

```
1 Cuando las barbas de tu vecino veas pelar pon las
tuyas a remojar.
2 Más vale pájaro en mano que ciento volando.
3 No hay mal que por bien no venga.
```

Formularios

"El contenido estático presentado a los usuarios de su sitio Web puede ser entendido perfectamente por una persona que use un lector de pantalla, o una línea Braille, suponiendo que haya usado marcadores estructurales y semánticos (como H1 - H6 para encabezamientos). Sin embargo, leer solo el contenido puede ser una actividad demasiado pasiva para alguien que visite su sitio Web, ya que pueden quedarse o irse indistintamente. Pero con un formulario los usuarios pueden haber encontrado un elemento con el que quieran interactuar con lo que el escenario ha cambiado, es una comunicación en dos direcciones. En esta situación se hace muy importante el marcar adecuadamente las páginas si quiere conseguir la información correcta."

Accessible HTML/XHTML Forms (The Web Standards Project)

1 JavaScript y formularios

Prioridad 1

[6.3](#) Asegure 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.

En realidad, no existe ningún punto de verificación de **prioridad 1** relacionado directamente con los formularios. Sin embargo, es muy habitual la utilización de **lenguajes de script** para realizar acciones sobre formularios como la validación de los datos.

Validación del formulario mediante JavaScript

La principal utilidad de JavaScript en los formularios es la comprobación de que todos los datos se han introducido correctamente.

En este sentido puedes verificar que no falte algún dato requerido o que los datos introducidos son correctos (e-mail bien formado, DNI correcto, etc.).

Con JavaScript puedes realizar esa **validación en el navegador**, del lado del cliente, antes de que los datos sean enviados al servidor.

La validación de datos del lado del cliente tiene una **ventaja** clara: reduce el tráfico por la red y por tanto, la respuesta es inmediata. Gracias a esta velocidad en la comprobación de datos, este tipo de validación es más cómoda para el usuario-a.

Sin embargo, presenta los **inconvenientes** inherentes al uso de JavaScript y de cualquier lenguaje de script en el lado del cliente:

- Algunos navegadores no soportan lenguajes de script como JavaScript. Los navegadores antiguos, navegadores de texto o de voz no tienen soporte para JavaScript o sólo soportan las primeras versiones.
- Los motores de búsqueda no soportan JavaScript y por tanto, sólo pueden acceder a la información disponible en el documento, y no a la información proporcionada mediante JavaScript.
- Aunque la mayoría de los navegadores más importantes soportan JavaScript, sus implementaciones suelen ser incompatibles. Es difícil conseguir que un código se ejecute correctamente en varios navegadores diferentes.
- El soporte de script por parte de los navegadores puede ser desactivado por los usuarios-as. El uso de elementos de programación en los navegadores puede generar problemas de seguridad que algunas personas o empresas prefieren evitar.

La otra opción es la **validación en el lado del servidor**.

Este tipo de comprobación provoca más tráfico en la red si existen datos incorrectos ya que tienen que enviarse los datos y las páginas de respuesta entre el navegador y el servidor hasta comprobar que todo esté correcto.

La lentitud de respuesta y el formato de la comunicación hacen que sea más incomoda para los usuarios-as. Sin embargo, es mucho más efectiva, ya que esta comprobación siempre se realiza, al no depender de variables externas (navegadores, soporte de JavaScript, etc.).

Según las pautas de accesibilidad, es necesario que las páginas Web sigan teniendo la misma funcionalidad y proporcionen la misma información, aunque el navegador no soporte JavaScript.

Por tanto, cuando uses JavaScript en los formularios asegúrate de que éstos sigan funcionando correctamente cuando JavaScript esté desactivado o no se soporte. Si empleas JavaScript para validar los datos del formulario, la única forma de asegurarte de que son correctos es mediante la validación en el lado del servidor.

Nunca confíes en que el código JavaScript se ejecutará correctamente.

Siempre que necesites realizar una validación de datos hazlo en el servidor y, si lo deseas, usa la validación en el lado de cliente como mejora para reducir el tráfico por la red y dar más comodidad a los usuarios-as.

Algunos autores dudan de la conveniencia de duplicar la validación en el cliente y en el servidor y aconsejan usar sólo la validación de servidor. La utilización de los dos tipos de validación implica duplicar el esfuerzo en la programación y en el mantenimiento, pudiendo ser fuente de inconsistencias si no se tiene el cuidado necesario.

El siguiente código de ejemplo usa JavaScript de forma incorrecta para validar los datos del formulario:

Ejemplo desaconsejado.

```
<script type="text/javascript"><!--
  function comprobar(){
    if (document.forms.formulario.nombre.value==""){
      alert("Comprobacion: Debe introducir un nombre");
    }else {
      document.forms.formulario.submit();
    }
  }
//--></script>
...
<form name="formulario" method="get"
action="submitpage.php">
<div>
  <label for="nombre">Nombre:</label>
  <input name="nombre" id="nombre" type="text">
  <input type="button" onclick="comprobar()"
    value="Enviar">
</div>
</form>
```

La función en JavaScript del ejemplo, además de comprobar los datos también se encarga de enviar el formulario. La llamada a dicha función se realiza mediante el evento **"onclick"** del botón, de forma que si no funciona el JavaScript, tampoco lo hace esa función y no se enviarán los datos del formulario. De hecho, sin el soporte de JavaScript el formulario deja de funcionar.

Si deseas hacerlo de forma correcta, llama a la función de JavaScript desde el evento **"onsubmit"** del formulario y cambia el botón a un botón de envío (type="submit"). Aunque no se soporte JavaScript el formulario sigue funcionando y enviando los datos pero puede que no se realice la comprobación. En este caso, la comprobación final de los datos se hace en el servidor.

Ejemplo.

```
<script type="text/javascript"><!--
  function comprobar(){
    if (document.forms.formulario.nombre.value==""){
      alert("Comprobacion: Debe introducir un nombre");
      return false;
    }else
      return true;
  }
//--></script>
...
<form name="formulario"
method="get" action="submitpage.php"
onsubmit="return comprobar(this)">
<div>
  <label for=" nombre">Nombre:</label>
  <input name="nombre" id="nombre" type="text">
  <input type="submit" value="Enviar">
</div>
</form>
```

Informe de errores en el lado del cliente

Existen muchas aproximaciones para mejorar la usabilidad de las validaciones en el lado de cliente mediante JavaScript.

Una de las opciones es presentar una lista con todos los problemas en una ventana de aviso. Cuando se trate de formularios extensos, donde el número de problemas puede ser elevado, esta técnica resulta poco útil.

La siguiente imagen te ofrece un ejemplo de este tipo de recurso.

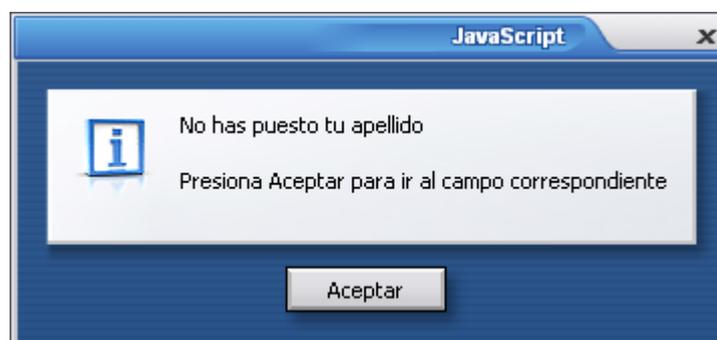


Otra opción, es mostrar una ventana con el primer error encontrado en el documento y situar el cursor sobre el campo de formulario que presenta el error para facilitar su corrección, y así sucesivamente.

De esta forma, se corregirán los errores uno a uno sin necesidad de mostrar una larga lista de ellos.

En este caso puedes usar JavaScript para mejorar la usabilidad haciendo que, cada vez que se muestre un aviso, al aceptar y cerrar la ventana, se coloque el cursor sobre el campo de formulario correspondiente.

Si lo deseas, puedes otorgar valores por defecto o resaltarlos visualmente con un color de fondo distinto y con un borde o tipo de letra diferente, etc. A continuación te mostramos un ejemplo de esta opción.



Al dar al botón de aceptar se pondría un texto por defecto y se seleccionaría, de forma que al escribir un nuevo carácter en el campo de formulario desapareciese el texto por defecto sin falta de que el usuario tuviese que borrarlo:

Ejemplo.

```
document.forms["datospersonales"].apellidos.value="Introduce tu apellido";
document.forms["datospersonales"].apellidos.select();
```

Informe de errores en el lado del servidor

En la validación de servidor puedes presentar los errores en una lista al comienzo de la página y resaltar visualmente los campos de formulario donde hay un error o falta un dato. Esta técnica puede desorientar a los usuarios-as de lectores de pantalla al recibir como respuesta la misma página del formulario.

Para evitar este problema, avisa en el **título de la página** (elemento TITLE), y en el encabezado principal de la misma (elemento H1), que se trata de una página de informe de errores en los datos del formulario:

Ejemplo.

```
<title>
  Faltan algunos datos en el formulario que
  acaba de enviar o algunos datos son erróneos.
</title>
```

En la lista inicial de errores puedes proporcionar enlaces a los campos de formulario que necesitan ser rellenados. De esta forma, evitas que el usuario-a deba tabular sobre todo el formulario hasta llegar al campo correspondiente.

Hay 4 errores en los datos que has enviado que necesitan ser corregidos:

1. [Falta el nombre](#)
2. [Falta el apellido](#)
3. [Falta el Código Postal](#)
4. [El e-mail no es correcto](#)

Otra opción es que únicamente muestres en la página aquellos campos que deben rellenarse, bien porque no se introdujeron los datos o porque éstos eran erróneos. Para ello, debes capturar y almacenar previamente los datos correctos.

Si muestras sólo un subconjunto de los campos evitas que el usuario-a se mueva por todo el formulario para seleccionar los campos con errores.

Select usados como menús de navegación

Algunos elementos **SELECT** se usan como **menús de navegación**, donde se encuentran las páginas a las que puedes acceder entre sus opciones.

Por ejemplo:

Ir a:

- Indice
- Capítulo 1
- Capítulo 2
- Capítulo 3

Ejemplo desaconsejado.

```
<script type="text/javascript"><!--
  function enviar(menu_ir){
    var opcion = menu_ir.options[menu_ir.selectedIndex].value;
    self.location = opcion;
  }
//--></script>
...
<form name="menu" method="get"
action="submitpage.php">
<div>
  <label for="ir">Ir a:</label>
  <select name="ir" id="ir"
  onchange="enviar(this)">
    <option value="index.html">
      Indice</option>
    <option value="capitulo1.html">
      Capitulo 1</option>
    <option value="capitulo2.html">
      Capitulo 2</option>
    <option value="capitulo3.html">
      Capitulo 3</option>
  </select>
</div>
</form>
```

Al seleccionar una opción del **SELECT**, el navegador abrirá automáticamente la página correspondiente. El problema de usar un código como el anterior es que, al navegar entre las opciones mediante el teclado, se dispara el evento "**onchange**" y se abre la página.

Los usuarios-as que navegan con el teclado no podrán escoger qué página quieren ver y sólo podrán ir a la primera de la lista. Además, en caso de que no se soporte JavaScript, el menú de navegación dejaría de funcionar.

La solución a este problema es tan sencilla como añadir un botón de envío para que sólo se visite una página una vez que se pulse ese botón. Esta opción es un poco más incómoda para los usuarios-as de ratón, que tendrán que dar dos clicks, pero deja de ser inaccesible para las personas que navegan con teclado u otro dispositivo similar.



Ejemplo.

```

<script type="text/javascript"><!--
  function enviar(formulario_menu){
    var opcion =
formulario_menu.ir.options[formulario_menu.ir.selectedIndex].value;
    self.location = opcion;
    return false;
  }
//--></script>
...
<form name="menu"
method="get" action="submitpage.php"
onsubmit="return enviar(this)">
<div>
  <label for="ir">Ir a:</label>
  <select name="ir" id="ir">
    <option value="index.html">
      Indice</option>
    <option value="capitulo1.html">
      Capítulo 1</option>
    <option value="capitulo2.html">
      Capítulo 2</option>
    <option value="capitulo3.html">
      Capítulo 3</option>
  </select>
  <input type="submit" value="Ir">
</div>
</form>

```

Además, para asegurarte de que sigue funcionando aún cuando no se soporte JavaScript, deberías usar un script de servidor similar al siguiente (situado en la página de destino, *submitpage.php*):

Ejemplo.

```

<?PHP if(isset($_POST['ir'])){
  header('Location: '.$_POST['ir']);
}?>

```

2 Asociación entre etiquetas y 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.

[12.4](#) Asocie explícitamente las etiquetas con sus controles.

Asociación explícita [12.4 - Prioridad 2]

La asociación explícita entre controles de formulario y sus etiquetas sirve de ayuda contextual para los lectores de pantalla a la vez que mejora la capacidad de interacción con los formularios para todos los usuarios-as.

Si un navegador soporta la asociación explícita, generalmente también permite interactuar con las etiquetas, de forma que, si una persona pulsa sobre una etiqueta, inmediatamente situará el cursor en el control de texto asociado, o se seleccionará el *radio button* o *checkbox* correspondiente.

Sólo puedes asociar cada elemento **LABEL** con un control de formulario.

Para asociar explícitamente una etiqueta con un control, identifica éste con el atributo **"id"** y asocia la etiqueta con el atributo **"for"** del elemento **LABEL**.



Opciones Especiales

- Frenos Antibloqueo
- Elevavunas eléctrico
- Techo Solar

Ejemplo.

```
<form action="#">
<div>
  <fieldset>
    <legend>Opciones Especiales</legend>
    <input id="opt1" type="checkbox" value="abs">
    <label for="opt1">Frenos Antibloqueo</label><br>
    <input id="opt2" type="checkbox" value="ee">
    <label for="opt2">Elevavunas eléctrico</label><br>
    <input id="opt3" type="checkbox" value="ts">
    <label for="opt3">Techo Solar</label>
  </fieldset>
</div>
</form>
```

Algunos navegadores no soportan la asociación explícita entre el control de formulario y su etiqueta, por lo que hay que asociarlos también implícitamente mediante su colocación en el código (ver "**Asociación implícita**", punto de verificación 10.2).

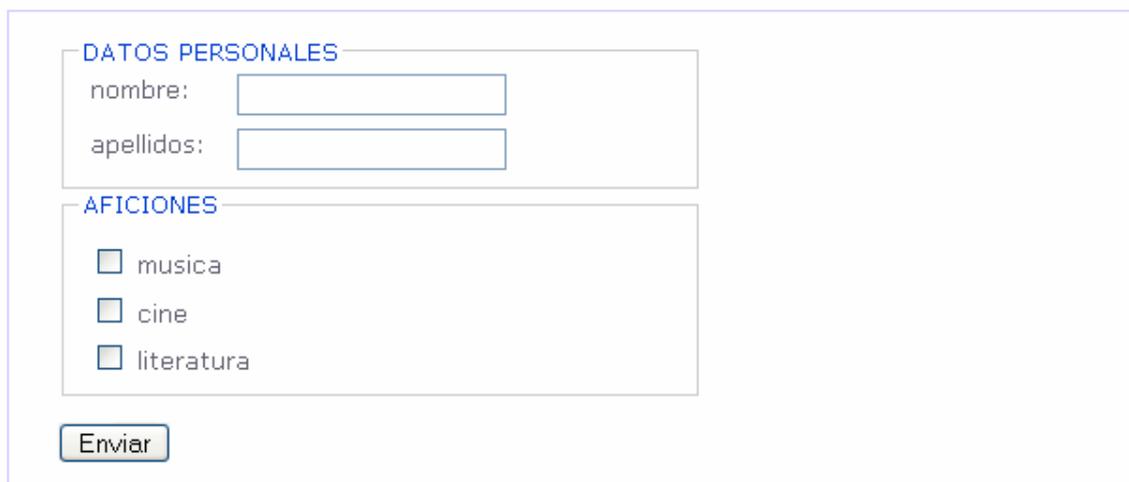
Asociación implícita [10.2 - Prioridad 2]

Algunos navegadores no soportan las asociaciones explícitas entre etiquetas y formularios por lo que debes relacionar las etiquetas y sus controles también por su colocación en el código.

Se puede asociar implícitamente una etiqueta con su control de formulario de dos formas:

- Por su posición en la página
- Mediante marcadores.

Si deseas asociar una etiqueta con su control mediante su **posición en la página**, sitúala inmediatamente antes o después del campo del formulario, dependiendo del tipo de control que sea. El texto de la etiqueta debe ser el primero antes o después del control de formulario.



Ejemplo.

```
<form action="#">
<div>
<fieldset>
  <legend>DATOS PERSONALES</legend>
  <div>
    <label for="nombre">nombre: </label>
    <input type="text" id="nombre" name="nombre">
  </div>
  <div>
    <label for="apellidos">apellidos:</label>
    <input type="text" id="apellidos" name="apellidos">
  </div>

```

```

</fieldset>
</div>
<div>
<fieldset>
  <legend>AFICIONES</legend>
  <div>
    <input type="checkbox" id="musica" name="musica">
    <label for="musica">musica</label>
  </div>
  <div>
    <input type="checkbox" id="cine" name="cine">
    <label for="cine">cine</label>
  </div>
  <div>
    <input type="checkbox" id="literatura" name="literatura">
    <label for="literatura">literatura</label>
  </div>
</fieldset>
</div>
<div>
  <input type="submit" value="Enviar">
</div>
</form>

```

Para la asociación mediante marcadores sitúa el texto de la etiqueta tal como te indicamos anteriormente, pero además, incluye el propio control de formulario dentro del contenido de la etiqueta LABEL.

Etiqueta asociada implícitamente mediante marcadores:

Ejemplo.

```

<label for="nombre">
  Nombre:
  <input type="text" id="nombre" name="nombre" tabindex="1">
</label>

```

Como puedes apreciar, la asociación implícita mediante marcadores no se puede usar cuando emplees tablas para maquetar y sitúes la etiqueta en una celda y el control de formulario en otra. Al tener cada elemento en una celda diferente, no podrías incluir el control dentro del contenido de la etiqueta LABEL..

Etiqueta no asociada implícitamente mediante marcadores:

Ejemplo.

```

<tr>
  <td><label for="nombre">Nombre:</label></td>
  <td><input type="text" size="20" name="nombre" id="nombre"></td>
</tr>

```

3 Independencia de dispositivo

Prioridad 3

9.4 Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos.

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.

Tabindex [9.4 - Prioridad 3]

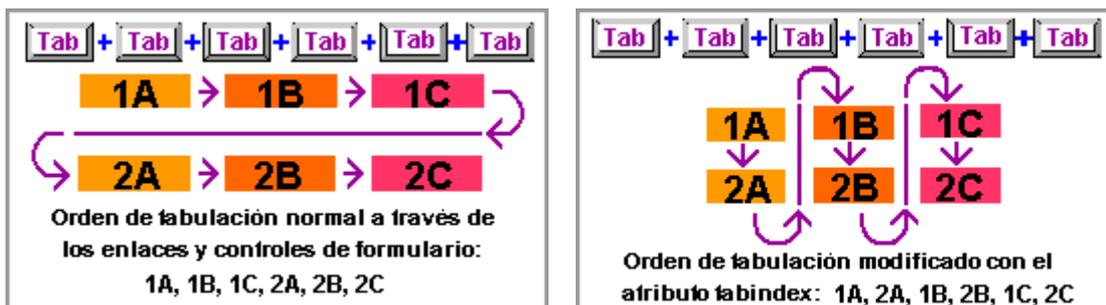
Uno de los principios básicos de la accesibilidad es el diseño **independiente de dispositivo**, por ejemplo, permitiendo la navegación por teclado y no sólo con el ratón.

El acceso mediante teclado es el principal medio de interacción de las páginas accesibles ya que una gran parte de las ayudas de usuario-a se basan en el uso del teclado o emulan su funcionamiento.

El orden de tabulación (por defecto) empieza en el primer enlace o control de formulario del documento y continúa por los siguientes enlaces o controles en el orden en el que aparecen en el código HTML.

En ocasiones este orden es inapropiado, por lo que podría ser necesario redefinirlo.

Para especificar un nuevo orden, usa el atributo "**tabindex**" (que funciona con los elementos: **A, AREA, BUTTON, INPUT, OBJECT, SELECT y TEXTAREA**) para especificar el orden de tabulación del elemento al que pertenece.



Sin embargo, no conviene especificar un orden de tabulación si éste es innecesario (si puede servir el orden por defecto). Procura que el orden de tabulación por defecto sea el correcto mediante una buena disposición lógica del contenido.

A continuación te mostraremos un ejemplo de una mala práctica.

(tabindex 6)

¿Cuál es tu color favorito?

Rojo (tabindex 3)

Verde (tabindex 4)

Azul (tabindex 5)

Empresa (tabindex 2)

(tabindex 7)

[Saltar a una alternativa más accesible](#) (tabindex 1)

Una buena práctica es la que te detallamos a continuación, porque los elementos se encuentran distribuidos de forma lógica para que el orden de tabulación por defecto sea el adecuado:

[Saltar a una alternativa más accesible](#)

Empresa

¿Cuál es tu color favorito?

Rojo

Verde

Azul

Ejemplo.

```
<form action="#">
<p>
  <a href="form-alt.htm">
    Saltar a una alternativa más accesible
  </a>
</p>
<p>
  <label for="emp">Empresa</label>
  <input type="text" id="emp">
</p>
<p>¿Cuál es tu color favorito?</p>
<div>
  <input type="checkbox" id="rojo">
  <label for="rojo">Rojo</label>
</div>
<div>
```

```

<input type="checkbox" id="verde">
<label for="verde">Verde</label>
</div>
<div>
<input type="checkbox" id="azul">
<label for="azul">Azul</label>
</div>
<p>
<input type="submit" value="Enviar">
<input type="reset" value="Deshacer">
</p>
</form>

```

Accesskey [9.5 - Prioridad 3]

La especificación HTML 4.01 permite definir, mediante el atributo "**accesskey**", combinaciones de teclas para acceder directamente a los enlaces y controles de formulario de una página Web.

Los elementos para los que puedes definir una tecla de acceso rápido son: **A, AREA, BUTTON, INPUT, LABEL, LEGEND, TEXTAREA.**

El valor del atributo "accesskey" será un único carácter con el que se asociará el control de formulario. La combinación de teclas necesaria para acceder al "accesskey" depende del navegador y de la plataforma.

Al usar teclas de acceso rápido debes tener en cuenta las siguientes consideraciones:

- Selecciona teclas que no causen conflicto con los navegadores.
- Informa sobre la existencia de teclas de acceso rápido e indica cuáles son.
- No abuses de estas teclas ya que, si son muchas, dejan de ser útiles.

En el siguiente ejemplo hemos asignado atajos de teclado para los controles del formulario:

[Saltar a una alternativa más accesible](#)

Emperesa

¿Cuál es tu color favorito?

Rojo

Verde

Azul

Ejemplo.

```
<form action="#">
<p>
  <a href="form-alt.htm">
    Saltar a una alternativa más accesible
  </a>
</p>
<p>
  <label for="emp">
    E<span style="text-decoration:underline">m</span>presa
  </label>
  <input type="text" id="emp">
</p>
<p>¿Cuál es tu color favorito?</p>
<div>
  <input type="checkbox" id="rojo">
  <label for="rojo">
    <span style="text-decoration:underline">R</span>ojo
  </label>
</div>
<div>
  <input type="checkbox" id="verde">
  <label for="verde">
    <span style="text-decoration:underline">V</span>erde
  </label>
</div>
<div>
  <input type="checkbox" id="azul">
  <label for="azul">
    A<span style="text-decoration:underline">z</span>ul
  </label>
</div>
<p>
  <input type="submit" value="E nviar">
  <input type="reset" value="D eshacer">
</p>
</form>
```

Para resaltar cuál es la letra que funciona como atajo de teclado puedes utilizar un subrayado o, si se trata del texto de los botones, separar la letra del resto y ponerla en mayúscula. El resaltado de la letra se ha de hacer de forma manual, pues los navegadores no resaltan las teclas de acceso rápido.

Puedes encontrar más información en el apartado "**Navegación, Lenguaje y Comprensión - Independencia de dispositivo**".

4 Otros aspectos sobre accesibilidad

Prioridad 2

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

Agrupar la información [12.3 - Prioridad 2]

Por ejemplo, puedes usar **OPTGROUP** para **organizar las listas largas de opciones** de menú en grupos más pequeños.



El código se construiría tal y como aparece a continuación:

Ejemplo.

```
<form action="#">
<div>
  <select name="Razas de Animales">
    <optgroup label="Perros">
      <option value="p_cocker">Cocker</option>
      <option value="p_bobtail">Bobtail</option>
      <option value="p_dalmata">Dálmata</option>
    </optgroup>
    <optgroup label="Gatos">
      <option value="p_siames">Siamés</option>
      <option value="p_persa">Persa</option>
    </optgroup>
  </select>
</div>
</form>
```

También puedes emplear **FIELDSET** para **agrupar controles** de formulario en unidades semánticas y **describir el grupo** con el elemento **LEGEND**.



En este caso el código sería el siguiente:

Ejemplo.

```

<form action="#">
<div>
<fieldset>
  <legend>Seleccione las opciones</legend>
  <fieldset>
    <legend>Color</legend>
    <div>
      <input type="radio" name="color" id="color1"
        value="rojo" checked="checked" tabindex="10">
      <label for="color1">Rojo</label>
    </div>
    <div>
      <input type="radio" name="color" id="color2"
        value="azul" tabindex="20">
      <label for="color2">Azul</label>
    </div>
  </fieldset>
  <fieldset>
    <legend>Motor</legend>
    <div>
      <input type="radio" name="power" id="power1"
        value="4cil" checked="checked" tabindex="30">
      <label for="power1">4 cilindros</label>
    </div>
    <div>
      <input type="radio" name="power" id="power2"
        value="6cil" tabindex="40">
      <label for="power2">6 cilindros</label>
    </div>
  </fieldset>
</fieldset>
</div>
</form>

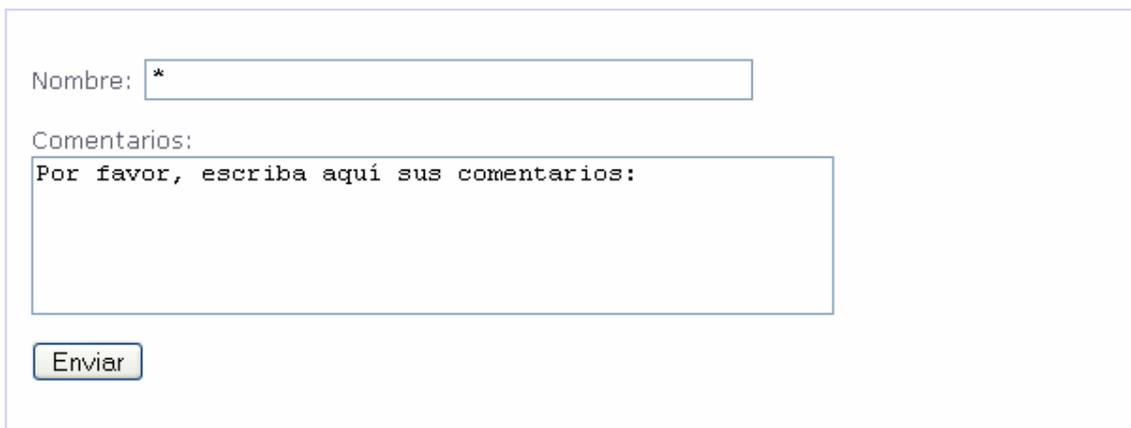
```

Prioridad 3

10.4 Hasta que las aplicaciones de usuario manejen correctamente los controles vacíos, incluya caracteres por defecto en los cuadros de edición y áreas de texto.

Caracteres por defecto en controles vacíos [10.4 - Prioridad 3]

Algunos navegadores antiguos no permiten al usuario-a utilizar la tecla de **tabulación** para ir a un campo de formulario vacío. Por tanto, hay que incluir **caracteres por defecto** en los cuadros de edición y áreas de texto.



Nombre: *

Comentarios:
Por favor, escriba aquí sus comentarios:

Enviar

Ejemplo.

```
<form action="#">
<div>
  <p>
    <label for="nombre">Nombre:
      <input type="text" name="nombre" id="nombre"
        size="50" value=" *">
    </label>
  </p>
  <p>
    <label for="comentarios">
      Comentarios:<br>
      <textarea name="comentarios" id="comentarios"
        rows="4" cols="50">
        Por favor, escriba aquí sus comentarios:
      </textarea>
    </label>
  </p>
  <p>
    <input type="submit" value="Enviar">
  </p>
</div>
</form>
```

Estándares

"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."

Pautas WCAG del WAI

1 Gramáticas formales

Prioridad 2

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

Validar los documentos contra gramáticas formales [3.2 - Prioridad 2]

Procura escribir el código siguiendo una gramática formal, identificar el tipo de documento mediante **!DOCTYPE** y validarlo con alguna de las herramientas disponibles, como el [servicio de validación](http://validator.w3.org/) (<http://validator.w3.org/>) proporcionado por el W3C.



The W3C Markup Validation Service is a free service that checks Web documents in formats like HTML and XHTML for conformance to W3C Recommendations and other standards.

VALIDATE YOUR MARKUP

VALIDATE BY URL

Address:

Enter the URL of the page you want to check. Advanced options are available from the [Extended Interface](#).

VALIDATE BY FILE UPLOAD

Local File:

Select the file you want to upload and check. Advanced options are available from the [Extended File Upload Interface](#).

Note: file upload may not work with Internet Explorer on some versions of Windows XP Service Pack 2, see our [information page](#) on the W3C QA Website.

Inicialmente, el **HTML** surge como un lenguaje para el intercambio de documentos científicos pero su éxito y popularidad hicieron que evolucionase demasiado rápido y de forma desordenada, a veces sin criterios claros.

Los diferentes navegadores, con la intención de aportar ventajas sobre sus competidores, crearon una serie de elementos o atributos nuevos que, en algunos casos, pasaron a formar parte del estándar. En otros casos, quedaron como elementos propietarios de esos navegadores. Todo esto repercutió en la compatibilidad de los documentos HTML.

El **W3C** definió las nuevas **gramáticas formales** y desaconsejó muchos de esos elementos. Este organismo limitó principalmente los elementos relacionados con la presentación y el aspecto visual del documento, con la intención de lograr la **separación entre contenido y presentación**. Así, se definieron una serie de gramáticas formales frente a las que poder **validar el código** por medio de una **declaración del tipo de documento**.

Por tanto, si empleas **HTML** que **no valide** frente a una gramática formal, te encontrarás con una serie de limitaciones y **desventajas** que debes valorar:

- **Limitación en la compatibilidad del código.** Las gramáticas del W3C nos aseguran que el código es estándar y que, por lo tanto, va a ser compatible con todos los navegadores.

Si empleamos código sin validar podemos usar elementos o atributos propietarios de los diferentes navegadores que, o bien no funcionen en el resto, o bien con el tiempo acaben desapareciendo y dejen de soportarse, ya que los navegadores tienden a cumplir los estándares.

- **Inseguridad sobre la compatibilidad futura.** Además de los elementos propietarios, en la evolución de HTML hay una serie de elementos y atributos que antes formaban parte de la especificación pero que, actualmente, están desaconsejados. Muchos de los navegadores no soportarán esos elementos en un futuro más o menos cercano.
- **Falta de separación entre contenido y presentación.** Con frecuencia creamos código en el que se mezcla el contenido de las páginas con su presentación. En el lenguaje HTML existen etiquetas que sirven para definir la estructura del documento, organizar su contenido y definir su presentación. Ésto genera un código complejo y difícil de mantener y modificar.

La separación de contenido y presentación facilita la accesibilidad, la gestión y la portabilidad.

Por su parte **XHTML** es una reformulación de HTML que cumple con la especificación de **XML**.

El lenguaje XHTML tiene como objetivo definir el contenido de los documentos logrando una separación completa de contenido y presentación y, a su vez, aportar a HTML todas las características y ventajas de XML.

Las principales **ventajas** que aporta XHTML frente a HTML son:

- Los documentos XHTML, al estar creados conforme a la especificación XML, pueden ser validados, procesados y **manipulados por aplicaciones** para el tratamiento de XML.
- XHTML se creó con la intención de que sea **modular**. Gracias a las características de XML, que permite la definición de nuevos elementos, podemos definir nuevos subconjuntos de XHTML con un vocabulario específico para una determinada necesidad. Los navegadores podrán implementar unos módulos u otros de XHTML.

Las **gramáticas formales** publicadas para HTML 4.01 y XHTML son las siguientes:

- **HTML "strict"**. Da prioridad a la estructura frente a la presentación. Te aconsejamos que uses esta gramática siempre que puedas, si quieres elaborar un código limpio de elementos propios de presentación. Incluye todo lo referente a la maquetación y presentación del documento en las hojas de estilo.

Declaración de tipo de documento.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

- **HTML "Transitional"**. Con menos restricciones que HTML "strict" permite incluir elementos de presentación, así como otros desaconsejados (deprecated). Esta gramática puedes utilizarla cuando sean necesarios los elementos y atributos de presentación.

Por ejemplo, úsalo si resulta demasiado costoso adaptar un portal existente a una gramática formal más estricta o cuando sepas que los visitantes potenciales de un portal no disponen de navegadores con soporte de CSS y deseas mantener el aspecto visual en dichos navegadores.

Declaración de tipo de documento.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

- **HTML "Frameset"**. Esta versión es igual que la "transitional" pero permite el uso de frames.

Declaración de tipo de documento.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD  
HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

- **XHTML "strict"**. Esta versión de XHTML contiene un elevado número de restricciones sobre los elementos, atributos, y anidamientos posibles de etiquetas. No permite la utilización de elementos de presentación.

Declaración de tipo de documento.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- **XHTML "Transitional"**. Esta opción es más flexible y menos restrictiva que la versión "strict". En este caso sí permite el uso de elementos de presentación.

Declaración de tipo de documento.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- **XHTML "Frameset"**. Similar a la versión Transitional pero posibilita la utilización de frames.

Declaración de tipo de documento.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

En función de esta información **terecomendamos** que uses gramáticas formales modernas como **XHTML**, aunque a veces necesites llegar a un compromiso entre la recomendación y la compatibilidad con desarrollos anteriores.

Si pretendes actualizar un portal existente el cambio de HTML a XHTML puede resultarte demasiado costoso por la cantidad de código que debes corregir.

Sin embargo, si necesitas crear un **nuevo portal** opta por el uso de **XHTML** debido a la escasa dificultad extra de codificar respecto a HTML.

Algo parecido ocurre a la hora de escoger entre la utilización de un DTD "Strict" o uno menos restrictivo, como el "Transitional" o "Loose". Te recomendamos que, si va a crear un nuevo portal, uses el DTD "**Strict**". El DTD "Frameset" sólo debes emplearlo si vas a usar frames.

Recuerda: El W3C **desaconseja** la utilización de los **frames** y, por tanto, del DTD "Frameset".

2 Tecnologías W3C

Prioridad 2

11.1 Utilice tecnologías W3C cuando estén disponibles y sean apropiadas para la tarea y use las últimas versiones que sean soportadas.

Utilizar tecnologías W3C (11.1 - Prioridad 2)

Los **estándares Web** son tecnologías establecidas por el **W3C** para crear una base común para el desarrollo Web.

El principal **objetivo** de los estándares Web es el de asegurar la máxima compatibilidad futura de los documentos así como su correcto funcionamiento para la mayoría de los usuarios-as, independientemente de la plataforma o navegador usado.

Te recomendamos que uses este tipo de tecnologías ya que han sido creadas, entre otras cosas, teniendo en cuenta la accesibilidad. Algunas de las tecnologías actuales del W3C son:

- **HTML, XHTML, XML:** Documentos estructurados.
- **CSS y XSL:** Definición de hojas de estilo.
- **RDF:** Metadatos.
- **SMIL.** Presentaciones multimedia.
- **XSLT:** Transformaciones de estilo.
- **PNG:** Gráficos.
- **MathML:** Ecuaciones matemáticas

Los últimos estándares o tecnologías más significativas son **XHTML, XML** y las hojas de estilo en cascada **CSS**.

Algunas de las principales razones que motivan la utilización de estas tecnologías son:

- **Facilidad de desarrollo y mantenimiento.**
- Asegura la **compatibilidad futura** de los documentos.
- Aumentan la **compatibilidad actual** al no usar tecnologías propietarias.
- Las páginas tendrán un **menor tamaño** y se cargarán con **mayor rapidez**.

En la actualidad, la práctica totalidad de los navegadores modernos ya soportan correctamente XHTML 1.0 y CSS de nivel 1. No ocurre lo mismo con las hojas de estilo en cascada de nivel 2, ya que varios navegadores aún no las soportan.

Aunque el uso de estos estándares parezca una tarea difícil, en realidad no implica tantos problemas. Si tienes conocimientos de HTML la migración a XHTML es prácticamente inmediata. De hecho XHTML puede considerarse una versión simplificada y más restrictiva de HTML.

Accesibilidad y Usabilidad Web

En función de este punto de verificación, y teniendo en cuenta las capacidades de los navegadores actuales, te **recomendamos** que diseñes las páginas Web con **XHTML 1.0 con el DTD "Strict"**.

Además, procura eliminar las tablas para maquetar (en la medida de lo posible) y los elementos de presentación para trasladarlos a las hojas de estilo. En su defecto, emplea la gramática XHTML "Transitional".

3 Elementos desaconsejados

Prioridad 2

[11.2](#) Evite características desaconsejadas por las tecnologías W3C.

No usar características desaconsejadas [11.2 - Prioridad 2]

En HTML existen una serie de **elementos o atributos desaconsejados**. Estos elementos han quedado **obsoletos** y desaparecerán en futuras versiones de HTML.

Muchas de las características desaconsejadas en HTML son elementos o atributos específicos de presentación o propietarios de determinados navegadores.

En el punto de verificación 3.2 ya te mostramos algunas de las **ventajas** que tiene la utilización de una gramática formal:

- Compatibilidad con navegadores actuales y futuros.
- Separación de contenido y presentación.
- Aumento de la accesibilidad.

Algunos de los elementos desaconsejados son:

- APPLET
- FONT
- BASEFONT
- ISINDEX
- CENTER
- MENU
- STRIKE
- U
- DIR
- S

El W3C **desaconseja** la mayoría de los **atributos de presentación**. Generalmente, éstos se prohíben al usar una definición de tipo de documento HTML "Strict". Puedes consultar la [lista de elementos](#) y la [lista de atributos](#) de HTML para saber cuáles están desaconsejados.

Por ejemplo, evita usar características de presentación en el código HTML, tal y como aparece en el siguiente código.

Ejemplo desaconsejado.

```
<p><b><font face="Verdana" color="#FF0000" size="3">  
En un lugar de La Mancha de cuyo...</font></b></p>
```

Puedes trasladar todas esas características a las hojas de estilo para seguir las recomendaciones del W3C:

Ejemplo.

```
p {  
  font-family: Verdana;  
  font-size: 1.2em;  
  font-weight: bold;  
  color: #FF0000;  
}
```

```
<p>En un lugar de La Mancha de cuyo... </p>
```

4 Página alternativa accesible

Prioridad 1

11.4 Si, después de los mayores esfuerzos, no puede crear una página accesible, proporcione un vínculo a una página alternativa que use tecnologías W3C, sea accesible, tenga información (o funcionalidad) equivalente y sea actualizada tan a menudo como la página (original) inaccesible.

Proporcionar alternativas accesibles [11.4 - Prioridad 1]

En alguna ocasión es imposible lograr que toda la información de una página sea accesible. En estos casos puedes usar una **página alternativa** cuyo contenido sí sea accesible.

Para solventarlo proporciona **enlaces** al inicio de la página original y de la alternativa que permitan a los usuarios-as moverse entre ellas. Los enlaces se han de situar de forma que sean bien visibles.

Otra opción es usar la **meta-información** (elementos LINK) de forma que los navegadores descarguen el documento original o el alternativo en función del navegador usado o las preferencias del usuario-a.

Sólo debes usar páginas alternativas como último recurso en caso de que fallen todos los intentos por hacer la página accesible.

Si construyes una página alternativa, debes actualizarla de forma periódica, al igual que lo haces en la página original.

A continuación te mostramos un ejemplo:

[Alternativas accesibles al formulario](#)

Nombre

Te gusta el curso?

Es una pasada

Muchísimo

Mucho

En este caso el enlace "*Alternativas accesibles al formulario*" dirige a una página donde se mostraría esta información:

Si prefieres no utilizar el formulario on-line, puedes:

- Enviar un e-mail message to nos@uaw.org con tu nombre y opinión del curso.

Accesibilidad y Usabilidad Web

- Enviar un FAX al 985 00 0000.
- Llamar al 985 11 0000 e indicar tu opinión.

Estructura y Presentación

"Usando marcadores de forma inapropiada (es decir, no de acuerdo con las especificaciones) se dificulta la accesibilidad. El mal uso de marcadores para una presentación (por ejemplo, utilizando una tabla para maquetar o un encabezado - etiqueta H - para cambiar el tamaño de la fuente) dificulta que los usuarios con software especializado entiendan la organización de la página o cómo navegar por ella. Más aún, utilizando los marcadores de presentación en lugar de marcadores estructurales para transmitir estructura (por ejemplo, construir lo que parece una tabla de datos con un elemento HTML PRE) se hace difícil interpretar una página de forma inteligible a otros dispositivos..."

"Si el color por sí mismo se usa para transmitir información, las personas que no puedan diferenciar ciertos colores, y los usuarios que no tengan pantallas en color o utilicen dispositivos de salida no visuales, no recibirán la información. Cuando los colores de primer plano y de fondo tienen un tono similar, pueden no proporcionar suficiente contraste en las pantallas monocromáticas, así como a las personas con diferentes tipos de deficiencias de percepción de los colores..."

"Algunas personas con discapacidades cognitivas o visuales son incapaces de leer textos que se mueven con la suficiente rapidez o en absoluto. El movimiento puede también distraer de tal manera que el resto de la página se vuelve ilegible para las personas con discapacidades cognitivas. Los lectores de pantalla son incapaces de leer textos móviles. Las personas con discapacidades físicas podrían no ser capaces de moverse tan rápida o certeramente como para interactuar con objetos móviles."

Pautas WGAC del WAI

1 Hojas de estilo

En primer lugar definiremos los conceptos siguientes:

Contenido

Toda aquella información que se transmite al usuario-a, bien como texto, imágenes, vídeos, sonidos, etc.

Estructura

Organización lógica del documento. Algunos ejemplos de elementos estructurales en HTML son las listas (OL, UL, DL, LI, DD) o los encabezados (H1, H2, H3...).

Presentación

Forma visual de mostrar el contenido. Algunos elementos de presentación son FONT, CENTER, B, etc.

Prioridad 1

[6.1](#) Organice el documento de forma que pueda ser leído sin hoja de estilo. Por ejemplo, cuando un documento HTML es interpretado sin asociarlo a una hoja de estilo, tiene que ser posible leerlo.

Lectura sin hojas de estilo [6.1 - Prioridad 1]

Las **hojas de estilo** aumentan la accesibilidad de los documentos al **separar la estructura de la presentación**. Sin embargo, hay que comprobar que el documento sigue siendo **legible sin hojas de estilo**.

El contenido importante debe estar presente en el código HTML ya que el texto o imágenes generadas desde las hojas de estilo (con propiedades como "background-image", "list-style", "content", "cue", "cue-before", "cue-after", y los pseudo-elementos ":before" y ":after") no están disponibles para las ayudas técnicas.

Las hojas de estilo permiten variar la **posición en pantalla** de todos los bloques y elementos del código HTML (por ejemplo, mediante capas y posicionamiento con las propiedades "float", "position", "top", "right", "bottom" y "left").

Si el contenido del documento está desordenado en el código HTML y el orden lógico de lectura sólo se consigue mediante hojas de estilo, al desactivarlas se perderá ese orden de lectura.

Asegúrate de que el contenido esté bien estructurado y en su correcto orden de lectura dentro del código HTML.

Por ejemplo, imagina que quieres crear un formulario en el que se usan hojas de estilo para controlar la maquetación.

Nombre:

Dirección:

Teléfono:

Carnet:

El código sería el que aparece a continuación:

Ejemplo desaconsejado.

```

<head>
  <title>CSS con sentido</title>
  <style type="text/css">
    body{
      font-family: verdana, serif;
      font-size: 0.9em;
    }
    .etiquetas {
      line-height: 1.65em;
      padding-left: 0;
      width: 5em;
      float:left;
      font-height:1.2em;
    }
    .campos {
      padding-left: 0px;
      float:left;
    }
  </style>
</head>
<body>
  ...
  <div class="etiquetas">
    Nombre:<br>
    Dirección:<br>
    Teléfono:<br>
    Carnet:
  </div>
  <div class="campos">
    <input type="text" name="nom" id="nom">
    <br>
    <input type="text" name="dir" id="dir">
    <br>
    <input type="text" name="tel" id="tel">
    <br>
    <input type="text" name="car" id="car"
      size="3">
  </div>
  ...
</body>

```

Al desactivar las hojas de estilo, el formulario quedaría como se muestra a continuación, perdiendo la relación entre las etiquetas y los campos de formulario.

Nombre:
Dirección:
Teléfono:
Carnet:

Este requerimiento obliga a que el contenido permanezca comprensible al desactivar o no soportar las hojas de estilo.

Esto no quiere decir que deba mantener una presentación similar a la original. Al desactivar las hojas de estilo es posible que la presentación sea diferente y que la maquetación de la página varíe mientras el contenido sigue siendo comprensible.

Prioridad 2

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

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

Utilizar hojas de estilo [3.3 - Prioridad 2]

Hasta la aparición de las hojas de estilo, para dar forma a un documento se usaban las etiquetas y atributos que aportaba HTML o algunos elementos propietarios de diversos navegadores.

Sin embargo, el lenguaje HTML presenta una serie de limitaciones e inconvenientes a la hora de controlar la maquetación y presentación de un sitio Web.

Cuando crees un sitio Web, antes de aplicar estilos y definir su presentación, define su estructura y sus contenidos.

La utilización de hojas de estilo, especialmente las externas, que permiten separar el contenido de la presentación, proporciona importantes **ventajas**:

Mayor facilidad para modificar el estilo de un sitio Web.

De esta forma, al modificar las hojas de estilo cambias la presentación de todas las páginas que hagan uso de dichas hojas.

Si existe una buena separación entre el contenido y la presentación, puedes cambiar el estilo visual de un portal Web sólo con modificar sus hojas de estilo.

Código más claro y sencillo.

El código libre de etiquetas de presentación, usando hojas de estilo externas, es más compacto, claro y sencillo de leer, con lo que su mantenimiento es más sencillo. En el código HTML sólo estaría presente la parte de contenido y estructura del documento, dejando toda la presentación para las hojas de estilo.

La estructura del código es muy clara, reduciéndose casi por completo a un contenido textual dividido en bloques, títulos, textos, listas, enlaces, imágenes, tablas... El número de las diferentes etiquetas y atributos HTML se reduce significativamente.

Código más compacto.

Al trasladar todos los aspectos de presentación y maquetación a las hojas de estilo, el código creado es mucho más compacto. A veces se puede reducir el código a la mitad o incluso menos.

Reducción de los tiempos de carga de las páginas.

Como el código es mucho más compacto, reduces los tiempos de carga de las páginas, facilitando a los-as visitantes el acceso a las mismas.

Ten en cuenta que las hojas de estilo asociadas sólo se descargan en el navegador la primera vez que se necesitan. A partir de ahí, cada vez que una página requiere una hoja de estilo que ya haya sido utilizada, el navegador la busca en su caché en vez de descargarla de la red.

Código más fácil de mantener.

El mantenimiento de las páginas que presentan separación de contenido y presentación es más sencillo y productivo, al tener un código más compacto.

Personalización del estilo.

Algunos usuarios-as, por elección o necesidad, requieren cambiar la presentación de las páginas. Si usas las hojas de estilo facilitas a los usuarios-as la aplicación de sus propias hojas prediseñadas, como por ejemplo estilos de alto contraste o de accesibilidad.

Mayor compatibilidad con diferentes dispositivos.

Separar el contenido y la estructura de la presentación facilita el acceso desde una amplia variedad de dispositivos (pantallas pequeñas, PDAs, móviles de última generación, etc.). Así evitarás crear diferentes versiones de las páginas en función del dispositivo, con el consiguiente ahorro de tiempo y dinero.

Aun así, si quieres controlar más la presentación de las páginas, define varias hojas de estilo diferentes y aplica unas u otras en función del dispositivo del usuario-a. CSS permite definir diferentes estilos según el tipo de dispositivo sobre el que se visualice. Es lo que se denomina "**control de medios**".

Por tanto, usa hojas de estilos para controlar la maquetación y la presentación de una página Web.

A continuación te detallamos unas **pautas básicas** que debes seguir y los **errores más comunes** que debes evitar:

- Elimina del código HTML aquellos elementos que sólo sirven para cambiar el aspecto visual de presentación. Por ejemplo, para marcar el énfasis utiliza los

elementos **EM** y **STRONG**, que aportan información estructural, en vez de B e I, que sólo aportan aspecto visual (negrita y cursiva).

Recuerda: Las gramáticas estrictas desapruaban los elementos de presentación como B, I, CENTER, etc.

- No utilices elementos estructurales para dar formato al texto. Por ejemplo, en vez de usar los elementos de marcado de citas **BLOQUOTE** o **Q** para lograr el efecto de texto indentado, sírvete de las propiedades de **CSS** para tal efecto ("**text-indent**").
- Utiliza **CSS para dar estilo al texto**, en lugar de representarlo mediante imágenes. Un texto incluido en una imagen deja de estar disponible para aquellas personas que no pueden verla. Si usas esta técnica añade el mismo texto que aparece en la imagen al texto alternativo de la misma (atributo "**alt**").
- Evita la utilización de **tablas para maquetar**. En su lugar usa marcadores estructurales y las **hojas de estilo** para controlar la maquetación del documento. Recuerda lo comentado sobre tablas de maquetación en la unidad 4 y el ejemplo de maquetación mediante CSS.
- Evita añadir espacios en blanco o mediante la entidad ** **; en las **celdas vacías** de las tablas para poder proporcionarles bordes. Usa la propiedad "**empty-cells**".
- No añadas espacios (** **) para conseguir **márgenes** al contenido de un elemento. En su lugar, recurre a las **propiedades CSS** "margin", "margin-top", "margin-right", "margin-bottom" y "margin-left".
- Emplea la propiedad CSS "**text-align:center**" en lugar del elemento no estándar **CENTER**.

Por ejemplo, evita construir un código como el siguiente.

Ejemplo desaconsejado.

```
<FONT size="+5" color="Green">
Esto está en verde y con letra grande,
</FONT>
<FONT size="+3" color="Navy">
y esto en azul y es más pequeño.
</FONT>
```

Y opta por la utilización de hojas de estilo como la que aparece a continuación.

Esto está en verde y con letra grande, y esto en azul y es más pequeño.

Ejemplo.

```
<style type="text/css">
.para {
font-size: 130%;
color: Green;}
.lineados {
font-size: 70% ;
font-style: italic ;
```

```

    color: Navy }
</style>
...
<p class="para">
  Esto está en verde y con letra grande,
  <span class="lineados">
  y esto en azul y es más pequeño.
  </span>
</p>

```

Utilizar unidades relativas [3.4 - Prioridad 2]

Debes usar unidades relativas en vez de unidades absolutas tanto en la hojas de estilo como en el código HTML.

Unidades de medida en las Hojas de Estilo

Las **unidades de medida relativas** especifican tamaños relativos a los usados en otros elementos o a la resolución de la pantalla.

Las unidades de medida relativas en CSS son:

- **em** (ems, "font-size" o tamaño de la fuente actual)
- **ex** (x-height, altura de la letra "x" minúscula de la fuente actual)
- **px** (pixels, unidad relativa a la resolución de pantalla)

Las unidades "**em**" y "**ex**" definen tamaños relativos al tamaño de la fuente actual, excepto cuando se usan para definir el tamaño de la fuente, en cuyo caso hacen referencia al tamaño de la fuente del elemento padre.

La unidad "**px**", aunque no sea una unidad absoluta (su tamaño real depende de la resolución usada) tiene un funcionamiento es similar, al no permitir el redimensionado ni establecer relaciones entre los diferentes tamaños usados.

A continuación puedes ver algunos ejemplos del uso de estas unidades:

Ejemplo.

```

body {font-size: 1em}
h1 {margin: 0.5em}
p.especial {font-size: 0.8ex;
border-width: 2px}

```

En este ejemplo, en BODY definimos el tamaño de la fuente principal para toda la página. Al encabezado H1 le asignamos un margen de la mitad del tamaño de la fuente actual y al párrafo le definimos un tamaño de fuente en el que la "x" minúscula tendrá el 80% del tamaño de la "x" minúscula de la fuente del elemento padre. Al párrafo también le definimos un borde con un ancho de dos píxeles.

Otro tipo de medida relativa son los **porcentajes**. Funcionan de forma similar a las unidades "em" y "ex" pero especificando los tamaños con porcentajes. Cada propiedad que admite porcentajes especifica el valor que toma como referencia, que puede ser otra propiedad del mismo elemento, del elemento padre, etc. Por ejemplo:

Ejemplo.

```
p { font-size: 10pt }
p { line-height: 120%
/* 120% de 'font-size' */}
```

Las **ventajas** del uso de **unidades relativas** ("em", "ex" y **porcentajes**) es que permiten el diseño de sitios Web independientemente de la resolución usada por los usuarios-as.

La resolución más usada en la actualidad es de 800x600 y muchos sitios Web se diseñan pensando sólo en esa resolución. Esta práctica perjudica tanto a los usuarios-as con navegadores gráficos de menor resolución (móviles, PDAs, pantallas pequeñas) como a los usuarios-as que usen resoluciones mayores.

En el primer caso, obliga a tener que desplazarse horizontalmente para ver todo el contenido, y en el segundo caso las páginas Web se verían en una esquina del monitor o en una estrecha franja situada el centro del mismo, desaprovechando en gran medida su capacidad.

Te recomendamos que diseñes páginas Web que se muestren correctamente en monitores pequeños y grandes, que se adapten y transformen adecuadamente sea cual sea la resolución usada y el tamaño de fuente preferida por el usuario-a. Esto es lo que se denomina "**diseño fluido**" o "**diseño líquido**".

Un buen diseño basado en unidades relativas permite un fácil redimensionado de las fuentes y de la presentación de la página, facilitando así la navegación a los usuarios-as con problemas de visión, que suelen ampliar el tamaño de la fuente para facilitar la lectura.

Por otra parte, las **unidades de medida absolutas** son independientes de otros tamaños usados por el usuario-a o de la resolución y dependen exclusivamente de su propio sistema de medición.

- **in** (pulgadas; 1in=2.54cm)
- **cm** (centímetros; 1cm=10mm)
- **mm** (milímetros)
- **pt** (puntos tipográficos; 1pt=1/72in)
- **pc** (picas; 1pc=12pt)

Sólo recomendamos el uso de **unidades absolutas** y de **píxeles** (px) para medio impreso porque interesa controlar de forma precisa su presentación, para definir anchos y altos en elementos con un tamaño conocido como en las imágenes, objetos de Flash, Applets, etc., o para los anchos de los bordes.

Además de las unidades de longitud relativas y absolutas también existen unos **tipos especiales** o **palabras clave** para definir el tamaño de la fuente.

Palabras clave absolutas:

[**xx-small** | **x-small** | **small** | **medium** | **large** | **x-large** | **xx-large**]

Con estas palabras clave se especifican una serie de tamaños definidos por el navegador. La especificación CSS2 recomienda un factor de crecimiento de 1.2 entre un tipo y el siguiente. Esta tabla de tamaños puede diferir de una fuente a otra y de un navegador a otro.

Por ejemplo:

Ejemplo.

```
p {font-size: medium}
h1 {font-size: large}
span {font-size: small}
```

Si el navegador usa un tamaño de 12pt para el valor "medium", entonces el valor para "small" podría ser de 10pt y el valor para "large" de 14.4pt.

Palabras clave relativas:

[**larger** | **smaller**]

Definen tamaños relativos al del elemento padre, de forma que puedes definir un tamaño que sea inmediatamente mayor ("larger") o menor ("smaller") en la tabla de tamaños. Por ejemplo, si el elemento padre tiene un tamaño de "medium", un valor de "larger" hace que el tamaño de la fuente del elemento actual sea "large". Si el elemento padre no tiene un valor en la tabla de tamaños entonces el navegador podrá interpolar o extrapolar el valor correspondiente.

Por ejemplo:

Ejemplo.

```
p {font-size: 12pt}
p span {font-size: smaller}
```

En este ejemplo, el texto dentro de un elemento SPAN que se encuentre dentro de un párrafo (P) podría tener un tamaño de 10pt.

Finalmente, te hacemos las siguientes **recomendaciones**:

- Usa la unidad "em" para fijar el tamaño de letra.
- Utiliza **unidades de medida relativas** y **porcentajes**. CSS permite utilizar unidades relativas incluso en posiciones absolutas. Por tanto, se puede colocar una imagen que será alejada por "3em" desde el comienzo del elemento que la

contenga. Esta distancia es fija pero relativa al tamaño de letra actual, con lo cual se escalará adecuadamente.

- Emplea **medidas absolutas de longitud** sólo cuando las características físicas del medio de salida sean conocidas, por ejemplo con imágenes de mapa de bits.

Por ejemplo, puedes definir un tamaño para el BODY sin pensar en una resolución determinada, sino en unos márgenes relativos al tamaño de la pantalla. También puedes especificar el tamaño de la fuente por defecto y luego asignar los demás tamaños relativos al tamaño por defecto.

Ejemplo.

```
body {
  margin-left: 15%;
  margin-right: 15%;
  font-size: 1em;
}
p.noticia{
  font-size: 1.2em;
}
```

En este ejemplo el tamaño de la fuente principal, definida para BODY, es de 1 "em" y el contenido del documento ocupa el 70% del ancho y está centrado con un margen del 15% a cada lado. El párrafo de la clase "noticia" tiene un tamaño de letra un 20% mayor (1.2em) que el tamaño de la fuente principal

Unidades de medida relativas en elementos HTML

A la hora de definir unidades de medida en elementos HTML como tablas (TABLE, TD, TH, ...), marcos (FRAMESET, IFRAME), utiliza unidades de medida relativas en lugar de absolutas.

Las medidas relativas que puedes utilizar en HTML son:

Porcentajes

Por ejemplo, la etiqueta `<table width="20%">` indica que el ancho de la tabla ocupará un 20% del ancho del elemento que la contiene, generalmente el ancho de la página si la tabla no está anidada.

Proporcionales

Por ejemplo, la etiqueta `<FRAMESET cols="1* , 3*">` indica que el primer marco ocupa una cuarta parte del total y el segundo marco las tres cuartas partes restantes.

La recomendación es que intentes dar formato a través de hojas de estilo.

2 Color y contraste

Prioridad 1

2.1 Asegúrese de que toda la información transmitida a través de los colores también esté disponible sin color, por ejemplo mediante el contexto o por marcadores.

No transmitir información solo por el color [2.1 - Prioridad 1]

Por ejemplo, si quieres ofrecer los resultados de un test evita aportar información sólo a través del color diciendo *"la opción correcta se marca en azul"*.

- A
- B
- C

En su lugar emplea una forma alternativa que no se base sólo en el color. Por ejemplo: *"la opción correcta se marca en azul y aparece entre paréntesis"*.

- A
- B
- (C)

En el primer ejemplo, la "opción C" sólo aporta información basada en el color y un usuario-a que no pueda percibir esa diferencia (deficiencias visuales, pantallas en blanco y negro, monitores monocromo, navegadores de texto, etc.) no sabrá qué opción es la correcta.

Sin embargo, en el segundo ejemplo las personas pueden diferenciar la opción correcta por el texto además de por el color.

La información transmitida mediante color también la debes proporcionar de otra forma, mediante texto o marcadores de fuente (STRONG, EM, etc.), por ejemplo.

Los enlaces nos proporcionan un buen ejemplo de ello. Si no subrayas los vínculos apenas se distinguirán del texto que les rodea en monitores monocromo o en blanco y negro.

Enlace normal

En color: Puedes visitar la [página del W3C](#) para más información.

En blanco y negro: Puedes visitar la [página del W3C](#) para más información.

Enlace que sólo se diferencia por el color

En color: Puedes visitar la [página del W3C](#) para más información.

En blanco y negro: Puedes visitar la [página del W3C](#) para más información.

En los ejemplos anteriores puedes ver cómo un enlace normal se seguiría identificando en monitores en blanco y negro, mientras que un enlace sin subrayar es imposible de diferenciarlo del texto que le rodea.

Prioridad 2

[2.2](#) Asegúrese de que las combinaciones de los colores de fondo y primer plano tengan el 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 los textos].

Usar un contraste suficiente [2.2 - Prioridad 2]

Establecer un buen **contraste entre fondo y primer plano** constituye una buena práctica del diseño web.

En contra de lo que cabría esperar, el contraste es más importante para las **imágenes (prioridad 2)** que para el **texto (prioridad 3)** ya que éste se puede modificar con hojas de estilo de usuario-a o ser leído mediante lectores de pantalla.

Cuando especifiques un color de primer plano establece también uno de fondo y viceversa.

A continuación puedes ver una imagen que apenas contrasta con el color de fondo y otra que tiene un buen contraste (Prioridad 2).



Recuerda que es importante también establecer un buen contraste en el texto (prioridad 3):

- **Texto con buen contraste**
- **Texto con poco contraste**

3 Encabezamientos

Prioridad 2

3.5 Utilice elementos de encabezado para transmitir la estructura lógica y utilícelos de acuerdo con la especificación.

Transmitir la estructura lógica [3.5 - Prioridad 2]

Utiliza los elementos de encabezado (H1-H6) para marcar las diferentes secciones que componen la **estructura** de un documento. Ten en cuenta las siguientes consideraciones:

- Evita la utilización de los marcadores de encabezamiento para conseguir efectos de formato (dar estilo al texto). Para este fin puedes emplear las hojas de estilo.
- Tampoco simules encabezamientos mediante la aplicación de estilos directamente sobre el texto por medio de etiquetas y atributos HTML. En este caso pierdes la información estructural del documento al no especificar que se trata de un encabezamiento. Un lector de pantalla no distinguiría entre el título y el texto del documento.

Por ejemplo, algunos lectores de pantalla permiten al usuario conocer los encabezados y la estructura de una página.

A continuación te mostramos un mal ejemplo de simulación de un encabezado mediante estilo:

Ejemplo desaconsejado.

```
<p>
  <font size="3" face="Verdana" color="#074A88">
    <b>Introducción</b>
  </font>
</p>
```

La forma correcta de este código sería la siguiente:

Ejemplo.

```
<style type="text/css">
  h1{
    font-family: Verdana;
    font-size: 1.5em;
    font-weight: bold;
    color: #074A88;
  }
</style>
...
<h1>Introducción</h1>
```

Además, recuerda que debes **anidar los encabezamientos de forma correcta**. Evita aumentar más de un nivel de profundidad de los encabezamientos de cada vez.

Es decir, después de un encabezamiento H1 sólo puede ir o otro H1 o un H2, pero no un H3. Sin embargo, puedes decrementar a cualquier nivel. Así, después de un H4 sólo podrías aumentar a un H5, pero sí podrías bajar a cualquier nivel entre H1 y H3 o poner otro H4.

Un ejemplo de mal anidamiento de encabezados es el siguiente:

Ejemplo desaconsejado.

```
<h1>Música</h1>
  <h3>Rock</h3>
  <h3>Clásica</h3>
```

La forma correcta de anidamiento de ese ejemplo es esta:

Ejemplo.

```
<h1>Música</h1>
  <h2>Tipos de música</h2>
    <h3>Rock</h3>
    <h3>Clásica</h3>
```

4 Listas

Prioridad 1

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.

Imágenes como viñetas [1.1 - Prioridad 1]

No es recomendable el uso de imágenes para simular viñetas. Se han de usar los marcadores HTML apropiadamente. Si quieres diseñar una lista con una viñeta personalizada utiliza los marcadores de listas **OL**, **UL** o **DL** y propiedades **CSS**.

Con las hojas de estilo puedes cambiar el aspecto visual de las listas. Entre otras cosas permite la modificación de la imagen usada por defecto como viñeta, la situación de la viñeta o el formato de la lista. Revisa el siguiente código:

Ejemplo.

```
ul {  
  list-style-image: url(path_imagen);  
  list-style-position: inside;  
}
```

En él, se especifica que la imagen indicada en el path se utiliza como viñeta y que la viñeta está dentro del marco que envuelve al texto y no fuera. Si el elemento LI está formado por varias líneas éstas se alineen con la viñeta y no con la primera línea.

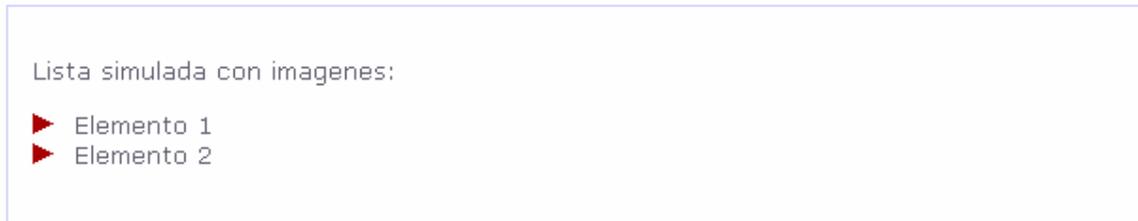
A continuación te mostramos algunas **propiedades CSS** que puedes aplicar a las listas:

Propiedades CSS.

```
display: inline | list-item  
list-style-type:  
  disc | circle | square | decimal |  
  lower-roman | upper-roman | lower-alpha  
list-style-image: url(path_de_la_imagen)  
list-style-position: inside | outside.
```

La propiedad "display" especifica si los ítems de la lista se renderizarán cada uno en una línea ("list-item") o seguidos en la misma línea ("inline"). La propiedad "list-style-type" define el tipo de imagen o numeración usada por defecto (círculo, cuadrado, decimales, números romanos, etc.).

En el siguiente ejemplo se usan imágenes en el código HTML para simular viñetas.

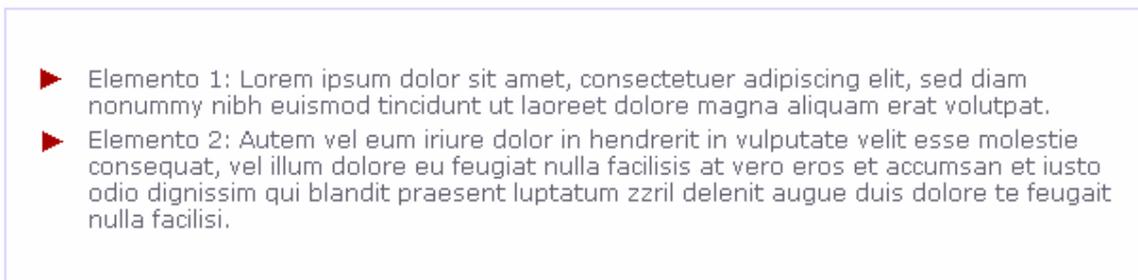


Recuerda que se trata de un ejemplo claro de mala práctica.

Ejemplo desaconsejado.

```
<p>Lista simulada con imagenes:</p>
<div>
  
  Elemento 1
</div>
<div>
  
  Elemento 2
</div>
```

Muchas veces empleamos la maquetación con tablas para conseguir el efecto de lista y que las imágenes queden alineadas y fuera del texto tal y como muestra la siguiente práctica:



Ejemplo desaconsejado.

```
<table>
<tr>
  <td></td>
  <td>Elemento 1: ...</td>
</tr>
<tr>
  <td></td>
  <td>Elemento 2: ...</td>
</tr>
</table>
```

Ten en cuenta que la forma correcta de hacer esto mismo es mucho más sencilla.

Lista realizada mediante marcadores de lista :

- ▶ Elemento 1
- ▶ Elemento 2

Ejemplo.

```
ul {
  list-style-image: url(img/bullet.png);
  list-style-position: outside;
  /*No haria falta, por defecto*/
}
...
<p>
  Lista realizada mediante marcadores de lista:
</p>
<ul>
  <li>Elemento 1</li>
  <li>Elemento 2</li>
</ul>
```

Si a pesar de la recomendación en contra utilizas imágenes como viñetas recuerda que sigue siendo necesario el atributo **"alt"** para añadir un texto alternativo a la imagen.

Por ejemplo:

Ejemplo.

```
<p>
  Lista simulada con imágenes con
  texto alternativo:
</p>
<div>
  
  Elemento 1
</div>
<div>
  
  Elemento 2
</div>
<div>
  
  Elemento 3
</div>
```

Si la lista está formada por pocos elementos puedes utilizar la palabra "Item" como texto alternativo de la imagen para indicar la presencia de un nuevo ítem de la lista.

También puedes usar algún carácter de uso común como "*" o "-" para simular viñetas de listas y que sirven para marcar la presencia de un ítem de lista.

En una lista con muchos elementos es recomendable que dejes el texto alternativo vacío para evitar que el lector de pantalla lea continuamente el texto alternativo de las viñetas (bien sean palabras o caracteres). En la siguiente imagen se muestra cómo aparecen los diferentes tipos de listas en un navegador de texto como **Lynx**.

```

Lista simulada con imagenes sin texto alternativo:
[bullet.png] Elemento 1
[bullet.png] Elemento 2

Lista realizada mediante marcadores de lista :
* Elemento 1
* Elemento 2

Lista simulada con imagenes con texto alternativo (*):
* Elemento 1
* Elemento 2

```

Si utilizas imágenes como viñetas, añadidas desde el código HTML (desaconsejado) o desde las hojas de estilo (recomendado), evita que éstas tengan un significado especial.

Por ejemplo, no uses una bola de color verde para indicar ítems nuevos en una lista y una bola de color naranja para indicar los antiguos.

En el caso de utilizar viñetas que transmitan información visual, debes presentarla o bien como texto alternativo de la imagen (si usas imágenes) o bien en el propio texto del ítem (si añades las viñetas desde el CSS). Por ejemplo:

```

Noticias (imágenes usadas como viñetas):
▶ En la última reunión...
▶ Según el informe...
▶ Inaugurado el...

```

Ejemplo.

```

<p>
  Noticias (imágenes usadas como viñetas):
</p>
<div>
  
  En la última reunión...
</div>
<div>
  
  Según el informe...
</div>
<div>
  

```

```
Inaugurado el...  
</div>
```

Noticias: (viñetas modificadas mediante CSS):

- ▶ Nueva: En la última reunión...
- ▶ Nueva: Según el informe...
- ▶ Inaugurado el...

Ejemplo.

```
<style type="text/css">  
  li {  
    list-style-image: url(img/bullet.png);  
  }  
  li.nueva{  
    list-style-image: url(img/bullet_verde.png);  
  }  
</style>  
...  
<p>  
  Noticias (viñetas modificadas mediante CSS):  
</p>  
<ul>  
  <li class="nueva">  
    Nueva: En la última reunión...  
  </li>  
  <li class="nueva">  
    Nueva: Según el informe...  
  </li>  
  <li>  
    Inaugurado el...  
  </li>  
</ul>
```

En el caso de utilizar las hojas de estilo es todavía más importante puesto que las imágenes incorporadas mediante CSS no llevan texto alternativo.

Prioridad 2

3.6 Marque correctamente las listas y los ítems de las listas.

Marcar correctamente las listas [3.6 - Prioridad 2]

Los elementos de marcado de listas especificados en HTML (DL, UL y OL) debes emplearlos sólo para crear listas y no con fines de formato o visuales, como dar sangría al texto.

Listas numeradas

En las listas numeradas (**OL**) usa los números compuestos mejor que los simples, ya que aportan más información sobre el anidamiento de las listas.

Una lista que esté realizada con números simples (como la del siguiente ejemplo) puede ser leída por un lector de pantalla como "1, 1, 2, 1, 2, 3, 2, 1, 2", sin aportar información sobre el anidamiento o profundidad de la lista.

- ```

1. Tarjetas Gráficas
 1. PCI
 2. AGP
 1. AGP-4x
 2. AGP-8x
 3. PCI-Express
2. Monitores
 1. TFT
 2. CRT

```

Sin embargo, una lista realizada con **números compuestos**, del tipo "1, 1.1, 1.2, 1.2.1, 1.2.2, 1.3, 2, 2.1, 2.2,..." sí proporciona información sobre el nivel de anidamiento de los ítems de la lista.

- ```

1. Tarjetas Gráficas
  1.1 PCI
  1.2 AGP
    1.2.1 AGP-4x
    1.2.2 AGP-8x
  1.3 PCI-Express
2. Monitores
  2.1 TFT
  2.2 CRT
    
```

Además, puedes modificar el tipo de numeración de las listas mediante las propiedades de las hojas de estilo.

Por ejemplo, el siguiente código muestra cómo especificar números compuestos para listas anidadas, creadas tanto con elementos UL como OL:

Ejemplo.

```
<style type="text/css">
  ul, ol {
    counter-reset: item
  }
  li {
    display: block;
  }
  li:before {
    content: counters(item, ".");
    counter-increment: item
  }
</style>
...
<ol>
  <li>Tarjetas Gráficas
    <ol>
      <li>PCI</li>
      <li>AGP
        <ol>
          <li>AGP-4x</li>
          <li>AGP-8x</li>
        </ol>
      </li>
      <li>PCI-Express</li>
    </ol>
  </li>
  <li>Monitores
    <ol>
      <li>TFT</li>
      <li>CRT</li>
    </ol>
  </li>
</ol>
```

La numeración compuesta mediante hojas de estilo solo la pueden realizar aquellos navegadores que tengan un buen soporte de CSS de nivel 2. Para aquellos navegadores que no lo soporten, actualmente la mayoría, la solución es usar listas desordenadas (UL) e incluir la numeración dentro de los elementos LI.

Ejemplo.

```
<ul>
  <li>1. Tarjetas Gráficas
    <ul>
      <li>1.1 PCI</li>
      <li>1.2 AGP
        <ul>
          <li>1.2.1 AGP-4x</li>
```

```

        <li>1.2.2 AGP-8x</li>
      </ul>
    </li>
    <li>1.3 PCI-Express</li>
  </ul>
</li>
<li>2 - Monitores
  <ul>
    <li>2.1 TFT</li>
    <li>2.2 CRT</li>
  </ul>
</li>
</ul>

```

Pistas contextuales

En las listas anidadas no numeradas (**UL**) debes proporcionar pistas contextuales ya que los usuarios-as invidentes pueden tener dificultades para distinguir el comienzo y fin de una lista y el inicio de cada ítem.

En el siguiente ejemplo te presentamos cómo ocultar el final de una lista cuando estén activadas las hojas de estilo. Si las hojas de estilo están desactivadas, no se soportan, o predomina la hoja de estilo del usuario-a, entonces se mostrará el final de la lista.

Suministros de oficina:

- Papelería:
 - Sobres
 - Blocs de notas
 - Papel con membrete
 - Papel carteles *(Fin de papelería)*
- Lápices:
 - Bolígrafos azules
 - Rotuladores pizarra blanca *(Final de lápices)*
- Sujetapapeles:
 - clips
 - grapas
 - cuerda. *(Fin de sujetapapeles)*

(Fin de suministros de oficina)

Ejemplo.

```

<style type="text/css">
  .findelista { display: none }
</style>
...
<p>Suministros de oficina:</p>

```

```

<ul>
  <li>Papelería:
    <ul>
      <li>Sobres</li>
      <li>Blocs de notas</li>
      <li>Papel con membrete</li>
      <li>Papel carteles
        <span class="findelista">
          (Fin de papelería)
        </span>
      </li>
    </ul>
  </li>
  <li>Lápices:
    <ul>
      <li>Bolígrafos azules</li>
      <li>Rotuladores pizarra blanca
        <span class="findelista">
          (Final de lápices)
        </span>
      </li>
    </ul>
  </li>
  <li>Sujetapapeles:
    <ul>
      <li>clips</li>
      <li>grapas</li>
      <li>cuerda.
        <span class="findelista">
          (Fin de sujetapapeles)
        </span>
      </li>
    </ul>
    <span class="findelista">
      (Fin de suministros de oficina)
    </span>
  </li>
</ul>

```

Listas de definición

Con las listas de definición (**DL**) puedes proporcionar definiciones para diferentes palabras y términos. A diferencia de los otros tipos de listas, constan de dos partes:

- **Término:** se declara con DT.
- **Definición:** se declara con DD y puede contener elementos de bloque en su interior (párrafos, listas, etc.)

Las listas de definición pueden tener varios términos para una definición así como varias definiciones para un mismo término. Revisa el ejemplo siguiente:

Ejemplo.

```

<dl>
  <dt>Accesibilidad<dt>
  <dd>

```

```
<p>Posibilidad de que un sitio o
servicio Web pueda ser visitado y
utilizado de forma satisfactoria
por el mayor número posible de personas,
independientemente de las limitaciones
que tenga la persona o de las derivadas
de su entorno.</p>
</dd>
<dt>Aplicación</dt>
<dd>
  <p>1. Programa preparado para una
  utilización específica, como el pago de
  nóminas, formación de un banco de
  términos léxicos, etc.</p>
</dd>
<dd>
  <p>2. Operación por la que se hace
  corresponder a todo elemento de un
  conjunto un solo elemento de otro
  conjunto.</p>
</dd>
<dt>Ordenador</dt>
<dt>Computador</dt>
<dd>
  <p>Máquina electrónica dotada de una
  memoria de gran capacidad y de métodos
  de tratamiento de la información,
  capaz de resolver problemas aritméticos
  y lógicos gracias a la utilización
  automática de programas registrados
  en ella.</p>
</dd>
</dl>
```

Una característica de las listas de definición es que los navegadores suelen mostrarlas indentando el texto de la definición, como puedes ver a continuación.

Accesibilidad

Posibilidad de que un sitio o servicio Web pueda ser visitado y utilizado de forma satisfactoria por el mayor número posible de personas, independientemente de las limitaciones que tenga la persona o de las derivadas de su entorno.

Aplicación

1. Programa preparado para una utilización específica, como el pago de nóminas, formación de un banco de términos léxicos, etc.
2. Operación por la que se hace corresponder a todo elemento de un conjunto un solo elemento de otro conjunto.

Ordenador Computador

Máquina electrónica dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas aritméticos y lógicos gracias a la utilización automática de programas registrados en ella.

En este ejemplo sólo hemos usado estilos para la fuente (tamaño, justificación, negrita), pero la indentación la crea automáticamente el navegador.

Puedes usar también las listas de definición para marcar texto cuya estructura corresponda con la proporcionada por la lista, sin necesidad de que se trate estrictamente de términos y definiciones.

Por ejemplo, puedes usarlas para marcar un diálogo, usando DT para indicar los personajes y DD para el texto de los diálogos.

Juan

Hola. ¿Qué tal estás?

Pedro

Bien, gracias. ¿Y tú?

Ejemplo.

```
<dl>
  <dt>Juan</dt>
  <dd>Hola. ¿Qué tal estás?</dd>
  <dt>Pedro</dt>
  <dd>Bien, gracias. ¿Y tú?</dd>
</dl>
```

5 Citas

Prioridad 2

3.7 Marque las citas. No utilice el marcador de citas para efectos de formato tales como sangrías.

Marcar correctamente las citas [3.7 - Prioridad 2]

La finalidad de los elementos **Q** y **BLOCKQUOTE** es marcar las citas y no debes usarlo para dar formato (indentar texto o crear sangrías).

Además, estos elementos de marcado de citas tienen un carácter estructural y semántico. Cada navegador puede aplicar su propio estilo visual a estos marcadores (por ejemplo, para **BLOCKQUOTE** un bloque de texto con sangría), o incluso, puedes modificar su estilo visual mediante CSS.

Recuerda que ambos elementos (**Q**, **BLOCKQUOTE**), aunque se utilicen erróneamente para dar formato, semánticamente siempre marcarán una cita.

El elemento **BLOCKQUOTE** marca las **citas largas** en un bloque de texto, por ejemplo párrafos, mientras que el elemento **Q** señala **citas cortas** "en línea", por ejemplo una frase dentro de un párrafo.

Ejemplo **BLOCKQUOTE**

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor...

Ejemplo **Q**

...y al caérsele el piano encima del pie, salió de su boca un aplastante "imecachis en la mar!"

Ejemplo.

```
<blockquote  
cite="http://www.elquijote.com/cap1">  
<p>  
  En un lugar de la Mancha, de cuyo  
  nombre no quiero acordarme, no ha  
  mucho tiempo que vivía un hidalgo  
  de los de lanza en astillero,  
  adarga antigua, rocín flaco y galgo  
  corredor...  
</p>  
</blockquote>  
<p>
```

```
...y al caérsele el piano encima del  
pie, salió de su boca un aplastante  
<q>mecachis en la mar!</q>  
</p>
```

El atributo opcional "**cite**", definido tanto para Q como para BLOCKQUOTE, indica la URL de origen de la cita.

En vez de usar Q o BLOCKQUOTE para lograr **efectos de formato** emplea hojas de estilo para lograr tales efectos. Por ejemplo, para indentar el texto usa la propiedad "**text-indent**" o bien juega con las propiedades "**margin**" y "**padding**". El siguiente código ejemplifica esto:

Ejemplo.

```
div.nombre_clase{  
  padding-left: 2em;  
  padding-right: 2em;  
}  
...  
<div class="nombre_clase">  
  Texto al que se quiere dar formato  
</div>
```

6 Dividir la información

Prioridad 2

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

Dividir bloques largos de información [12.3 - Prioridad 2]

A continuación te detallamos algunos elementos de HTML 4.01 para **agrupar el contenido** y facilitar su comprensión:

- Emplea el elemento **FIELDSET** para agrupar controles de formulario y describe el grupo con **LEGEND**.
- Utiliza **OPTGROUP** para organizar las listas largas.
- Usa **tablas para datos tabulares** y describe la tabla con **CAPTION**.
- Agrupa las filas y columnas de las tablas con **THEAD, TBODY, TFOOT** y **COLGROUP**.
- Anida las listas con **UL, OL** y **DL**.
- Emplea los encabezamientos de sección (**H1 - H6**) para crear documentos estructurados y separar tramos largos de texto.
- Separa las líneas de texto en párrafos (con el elemento **P**).
- **Agrupar los vínculos** relacionados entre sí.

De todas formas, usa sólo estos mecanismos de agrupación cuando sean apropiados (cuando la información se ofrezca en grupos lógicos) y evita crear grupos de forma aleatoria, puesto que pueden confundir a los usuarios-as.

7 Marcadores mejor que imágenes

Prioridad 2

3.1 Cuando exista un marcador apropiado, use marcadores en vez de imágenes para transmitir la información.

Usar marcadores en vez de imágenes [3.1 - Prioridad 2]

En el texto de la WCAG puedes encontrar la siguiente cita:

"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."

En ocasiones se utilizan imágenes para **representar datos tabulares** en lugar de los marcadores de tablas.

A veces, para evitar codificar una tabla en HTML y darle estilos mediante CSS, se crea la tabla en otro programa e inserta como una imagen en la página Web.

En la imagen siguiente te mostramos una tabla de datos que podría haberse elaborado mediante los marcadores de tabla de HTML.

País	Oro	Plata	Bronce
EEUU	22	26	18
Rusia	18	24	20
China	15	19	19

A continuación puedes ver la misma tabla realizada mediante HTML y hojas de estilo:

País	Oro	Plata	Bronce
EEUU	22	26	18
Rusia	18	24	20
China	15	19	19

Con frecuencia, se utilizan imágenes para **simular encabezados**. A veces se recurre a programas de diseño para crear imágenes que sirvan como títulos en lugar de emplear los elementos de encabezado (H1, H2, etc.).

Generalmente, esta técnica se emplea con el objetivo de usar fuentes especiales y aplicarles un aspecto visual determinado. En su lugar, te recomendamos que emplees texto y le des estilo mediante CSS en lugar de construir imágenes. Si esto es imposible, emplea el atributo "alt" de la imagen para proporcionar el mismo texto y, si se trata de un elemento estructural (como un encabezado), usa los marcadores apropiados.

Por ejemplo:

Ejemplo.

```
<h1></h1>
```

A continuación, te presentamos un ejemplo de encabezamiento, tanto si se puede renderizar la imagen como si no.



Introducción

Un ejemplo menos frecuente es la utilización de imágenes para representar **ecuaciones matemáticas**, en vez de usar lenguajes de marcado como **MathML**. Observa la siguiente imagen:

$$\left(\frac{\sqrt[3]{\frac{5}{2}+3}}{7}\right)^{10}$$

El código utilizado para representar esta fórmula en MathML es el siguiente:

Ejemplo.

```
<math
xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mo>( </mo>
    <mfrac>
      <mrow>
        <mroot>
          <mfrac>
            <mn>5</mn>
            <mn>2</mn>
          </mfrac>
          <mn>3</mn>
        </mroot>
        <mo>+</mo>
        <mn>3</mn>
      </mrow>
    </mfrac>
    <mn>7</mn>
  </mrow>
```

```
<mn>7</mn>  
</mfrac>  
<mo>)</mo>  
</mrow>  
<mo>*</mo>  
<mn>10</mn>  
</math>
```

Navegación, lenguaje y comprensión

"Proporcione mecanismos de navegación claros y coherentes, (información orientativa, barras de navegación, un mapa del sitio, etc.) para incrementar la probabilidad de que una persona encuentre lo que está buscando en un sitio. Los mecanismos de navegación claros y coherentes son importantes para las personas con discapacidad cognitiva o ciega y benefician a todos los usuarios.

La maquetación coherente de páginas, los gráficos reconocibles y el lenguaje fácilmente comprensible benefician a todos los usuarios. En particular, ayudan a personas con discapacidades cognitivas o con dificultades en la lectura. (Por tanto, asegúrese de que las imágenes tienen textos equivalentes para los ciegos, los de baja visión o para cualquier usuario que no puede o ha elegido no ver los gráficos.

La utilización de un lenguaje claro y simple promueve una comunicación efectiva. El acceso a la información escrita puede ser difícil para personas con discapacidades cognitivas o de aprendizaje. La utilización de un lenguaje claro y simple también beneficia a las personas cuyo primer idioma es diferente al del autor, incluidos aquellos que se comunican principalmente mediante lengua de signos."

Pautas WCAG del WAI

1 Lenguaje y comprensión

Prioridad 1

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

[14.1](#) Utilice el lenguaje apropiado más claro y simple para el contenido de un sitio.

Identificar cambios en el idioma [4.1 - Prioridad 1]

Si utilizas varios idiomas en una página, debes identificar claramente cualquier cambio de idioma mediante el uso del atributo "**lang**" y un código de idioma.

Así, los lectores de pantalla y sintetizadores de voz que reproducen varios idiomas son capaces de usar el acento y la pronunciación adecuados para cada idioma y no intentarán pronunciar el texto en el idioma principal usado por el programa o el documento. De forma similar, sirve de ayuda para los usuarios-as que usan traductores a braille.

Para marcar un cambio de idioma en un bloque de texto (P, DIV, BLOCKQUOTE, etc.) utiliza el atributo "lang".

Ejemplo.

```
<blockquote lang="en">
  The power of the Web is in its universality.
  Access by everyone regardless of disability is
  an essential aspect
</blockquote>
```

Las palabras o frases sueltas escritas en otro idioma que se encuentran dentro de un bloque de texto debes marcarlas con el elemento SPAN y usar el atributo "lang" del elemento SPAN. Revisa el ejemplo que te presentamos a continuación:

"La versión 2 de las WCAG, está todavía en working draft."

Ejemplo.

```
<p>
  La versión 2 de las WCAG, está todavía en
  <span lang='en-us'>working draft</span>
</p>
```

El código de idioma está formado por un código principal seguido por una serie de códigos secundarios opcionales separados por guiones:

código_principal[-código_secundario]

El código principal contendrá dos letras de las especificadas en el estándar ISO 639 (si deseas más información, consulta los "[Códigos para la representación de nombres de idiomas](#)") o bien uno de los caracteres especiales reservados como "i" o "x".

A continuación te mostramos algunos ejemplos de códigos de nombres de idioma:

Ejemplo.

ENGLISH	EN	GERMANIC
FRENCH	FR	ROMANCE
GERMAN	DE	GERMANIC
SPANISH	ES	ROMANCE

En el primer subcódigo puedes utilizar una combinación de dos letras de las especificadas en el estándar [ISO 3166](#) para cada país y que sirva para indicar la zona en la que se habla dicho idioma (p.e. en-US).

También puedes introducir un código de entre 3 y 8 caracteres que previamente haya sido registrado y se emplee, por ejemplo, para indicar dialectos (p.e. en-cockney) o idiomas no definidos en el estándar ISO 639 usando el prefijo "i" (p.e. i-chokeee).

A continuación puedes ver algunos ejemplos:

- Español de argentina: es-ar
- Español de españa: es-es
- Inglés de EEUU: en-us

Para informarte sobre la definición y formación de los códigos de idioma puedes consultar el [RFC 1766](#).

Usar un lenguaje claro y sencillo [14.1 - Prioridad 1]

A la hora de redactar el contenido procura que sea comprensible y esté redactado con un lenguaje claro y sencillo.

Un lenguaje con esas características beneficia a los usuarios-as con discapacidades cognitivas, de lectura o aprendizaje, o a aquellos-as cuyo idioma es diferente del utilizado en el documento.

A continuación te mostramos algunos **consejos** que permiten hacer el contenido más comprensible:

- Organiza los documentos estructurando de una forma clara y precisa los **encabezamientos**. Los encabezamientos significativos facilitan a los usuarios-as la localización de la información.
- Redacta los párrafos de forma que la **idea principal** esté situada al **principio** para facilitar una lectura rápida o superficial del documento. Además limita, en la medida de lo posible, los párrafos a **una sola idea o concepto principal**.
- Utiliza un **vocabulario claro y sencillo**. Evita las jergas, argots, o tecnicismos si son innecesarios y, en caso de usarlos, proporciona un glosario con las

definiciones. Utiliza palabras de uso común y huye del lenguaje literario, con términos poco habituales o frases complejas, si no es estrictamente necesario. Evita el uso de verbos en su forma pasiva.

Prioridad 3

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

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

[13.8](#) Localice la información destacada al principio de los encabezamientos, párrafos, listas, etc.

[14.2](#) Complemente el texto con presentaciones gráficas o auditivas cuando ello facilite la comprensión de la página.

[14.3](#) Cree un estilo de presentación que sea coherente para todas las páginas.

Identificar el idioma principal del documento [4.3 - Prioridad 3]

Además de marcar los cambios de idioma en el texto (prioridad 1), cuando elabores un documento identifica su idioma principal, bien con un atributo en la etiqueta HTML (como se muestra abajo) o bien a través de encabezamientos HTTP. De esta forma, beneficias a:

- los usuarios-as que utilicen lectores de pantalla.
- los correctores ortográficos y gramaticales.
- los motores de búsqueda, para localizar las palabras clave e identificar los documentos en el idioma deseado.

En **HTML** debes utilizar el atributo "**lang**" y un código de idioma.

Ejemplo.

```
<html lang="es">
```

En **XHTML 1.0** utiliza "**xml:lang**" combinado con "**lang**".

Ejemplo.

```
<html xmlns="http://www.w3.org/1999/xhtml "
lang="es" xml:lang="es">
```

En **XHTML 1.1** utiliza "**xml:lang**"

Ejemplo.

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="es">
```

Con encabezados **HTTP** emplea un elemento **META**.

Ejemplo.

```
<meta http-equiv="content-language" content="es" />
```

Especificar la expansión de abreviaturas y acrónimos [4.2 - Prioridad 3]

Una **abreviatura** es la representación gráfica reducida de una palabra mediante la supresión de letras finales o centrales y que suele finalizar con un punto.

Ejemplos de abreviaturas son.

- Sr. (señor)
- Dcha. (derecha)
- Fdo. (firmado)
- S.S.M.M. (Sus majestades)

Los **acrónimos** son vocablos formados por la unión de las letras de dos o más palabras, generalmente por el inicio de la primera palabra y el final de la última, o mediante otro tipo de combinaciones similares. También se denomina acrónimo a las formas abreviadas que se pronuncian como una palabra.

Algunos ejemplos de acrónimos son:

- modem: modulación demodulación
- ofimática: oficina informática.
- sonar: Sound Navigation And Ranging
- ovni: Objeto Volante No Identificado

El académico Manuel Seco, en el "*Diccionario de dudas y dificultades de la lengua española*", explica claramente cómo distinguir abreviaturas y siglas (acrónimos):

"En esto se diferencian las siglas de las abreviaturas. Unas y otras sirven para abreviar lo que se escribe; pero la abreviatura se lee traduciendo lo escrito por lo que en ello se representa (así, Sr. se lee forzosamente "señor", mientras que la sigla no se traduce, sino que se lee tal como está escrita [bien como una palabra como en el caso de APA o BOE, bien deletreándola como en ISBN]. Las abreviaturas no son más que formas acortadas en la escritura; las siglas son verdaderas palabras usadas tanto en la escritura como en el habla".

Cuando utilices acrónimos o abreviaturas por primera vez en una página especifica su expansión. De esa forma proporcionarás una definición útil a todos los usuarios-as y a los lectores de pantalla que se puedan configurar para que las pronuncien de diferentes maneras (normal o deletreando).

Además, este punto de verificación es útil para los motores de búsqueda pues así "saben", por ejemplo, que aquellos documentos en los que aparece el acrónimo TIC se están refiriendo a "Tecnologías de la Información y la Comunicación" y podrán indexarlos mejor.

La identificación de las abreviaturas y acrónimos se realiza con **ABBR** y **ACRONYM** respectivamente y para especificar la expansión se usa el atributo **"title"**.

En el siguiente código podrás ver un ejemplo:

La Fundación CTIC alberga la oficina española del W3C en el Parque Tec. de Gijón.

Ejemplo.

```
La Fundación
<acronym title="Centro Tecnológico de la Información
y Comunicación">CTIC</acronym>
alberga la oficina española del
<acronym title="World Wide Web Consortium"
lang="en">W3C</acronym>
en el Parque
<abbr title="Tecnológico">Tec.</abbr>
de Gijón.
```

Los **lectores de pantalla** pueden configurarse para pronunciar las abreviaturas o acrónimos de diferentes maneras (normal o deletreando) tanto internamente como mediante propiedades CSS específicas de tipo **Aural** (ver la sección de Metadatos, punto de verificación 13.2):

Ejemplo.

```
abbr {speak:spell-out;} /* Deletreando */
acronym {speak:normal;} /* Normal */
```

Los **navegadores visuales** suelen mostrar de diferente forma las abreviaturas o acrónimos que disponen de una expansión, generalmente mediante un subrayado de puntos, y mediante un mensaje emergente (en inglés tooltip).

Si usas las abreviaturas o acrónimos sólo con carácter definitorio entonces sólo es necesario que indiques la expansión la primera vez que aparezca en el documento. De otra manera, el documento estaría lleno de molestos subrayados que dificultarían su lectura.

Sin embargo, para los lectores de pantalla es útil que señales la utilización de una abreviatura o acrónimo cada vez que aparezca en el documento.

Por tanto, la mejor opción (aunque no sea estrictamente necesario) es que identifiques todas las apariciones de las abreviaturas o acrónimos mediante ABBR o ACRONYM, pero sólo especificando su expansión la primera vez que aparezca en el documento.

Ejemplo.

```
<h1>
  ¿Qué es el
  <acronym title="World Wide Web Consortium">
    W3C
  </acronym>?
</h1>

<p>
  El <acronym>W3C</acronym> fue creado en
  octubre de 1994 para guiar la Web a su máximo potencial
  mediante el desarrollo de estándares y protocolos comunes
  que promoviesen su evolución y asegurasen su
  interoperabilidad...
</p>
```

Situar la información importante al inicio [13.8 - Prioridad 3]

Si quieres ayudar a los usuarios-as a encontrar la información que buscan sitúa la información importante al comienzo de los encabezamientos, párrafos, listas, etc. De esta manera pueden localizar la información buscada mediante una lectura superficial del documento, sin tener que profundizar en el contenido de los párrafos. Este uso se denomina "**colocación inicial**".

Así, entre las dos opciones siguientes es preferible la segunda.

- Si quieres encontrar más información sobre accesibilidad Web, visita la [página del W3C](#).
- Visita la [página del W3C](#), si quieres encontrar más información sobre accesibilidad web. (*Buena práctica*)

Complementar el texto [14.2 - Prioridad 3]

Los elementos gráficos o auditivos (vídeos, imágenes, sonidos, etc.) complementan el texto y facilitan su **comprensión** a las personas con dificultades de lectura.

Recuerda: los elementos gráficos o auditivos deben cumplir los requisitos de accesibilidad correspondientes.

Por ejemplo, las imágenes que uses deben tener un texto alternativo y los elementos multimedia deben llevar subtítulos o transcripciones textuales.

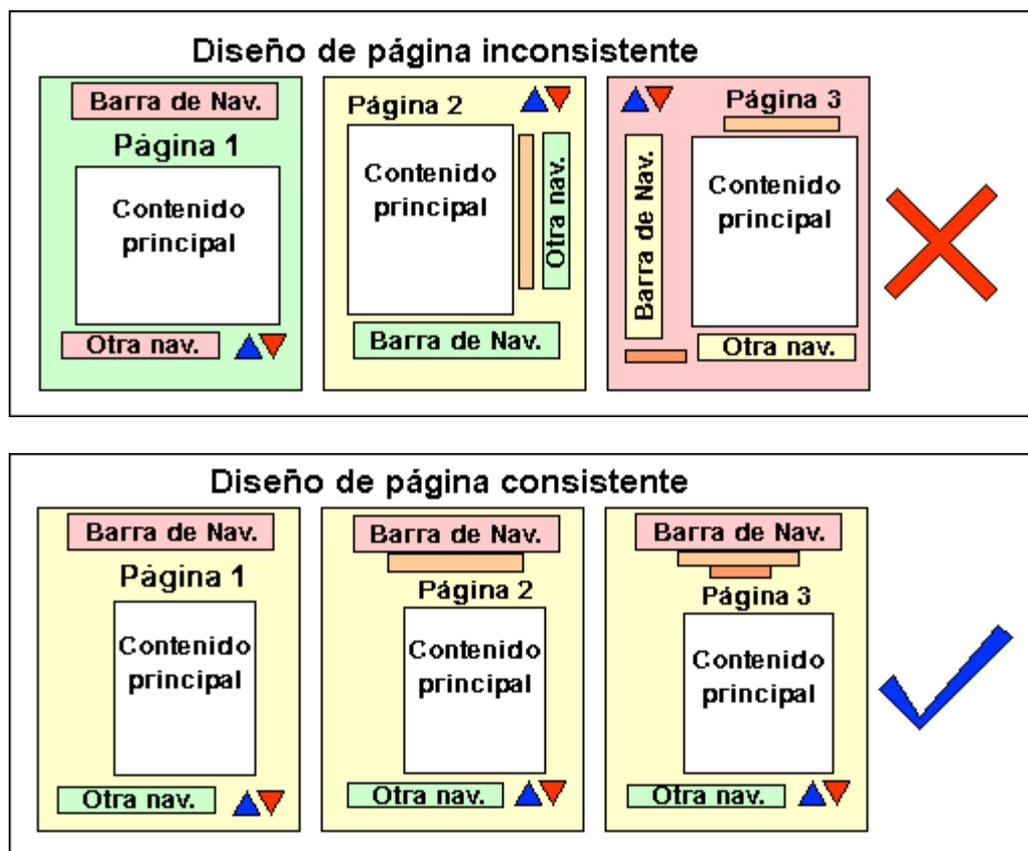
De todas formas evita el uso indiscriminado de elementos multimedia. Utiliza estos recursos en su justa medida e inclúyelos sólo si mejoran y facilitan la comprensión de los contenidos. Si no aportan nada al texto su uso puede ser perjudicial.

Usar un estilo de presentación coherente [14.3 - Prioridad 3]

Si empleas una misma ubicación y estilo en los enlaces principales de navegación de todas las páginas de un sitio Web, su uso acaba siendo familiar y los usuarios-as sabrán exactamente dónde localizarlos y cómo usarlos.

De igual forma, si utilizas una maquetación, estructura de contenido, un estilo de fuentes y colores similar en todas las páginas facilitarás la **identificación** y la **usabilidad** del sitio Web.

Los siguientes ejemplos muestran una buena práctica y otra mala.



2 Actualizaciones automáticas y redireccionamiento

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.

[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 que ejecute esta posibilidad.

Evitar las actualizaciones automáticas [7.4 - Prioridad 2]

Las actualizaciones automáticas de una página pueden **desorientar** a algunos usuarios. No podemos prever cuánto tiempo puede tardar un usuario-a en revisar el contenido de una página.

Si es necesario actualizar el contenido de la página cada cierto tiempo, informa al usuario-a para que lo haga por si mismo.

Por tanto, evita recargar constantemente el documento.

A continuación te mostramos un ejemplo de mala práctica:

Ejemplo desaconsejado.

```
<meta http-equiv="refresh" content="60;">
```

Evitar el redireccionamiento [7.5 - Prioridad 2]

El usuario-a puede desorientarse si utilizas redirecciones automáticas como la que aparece en el código:

Ejemplo desaconsejado.

```
<meta http-equiv="refresh" content="60;redireccion.html">
```

Las redirecciones automáticas no pertenecen a los estándares web, producen **desorientación** en los usuarios-as y pueden corromper el historial de páginas visitadas del navegador.

Si necesitas redirigir ten en cuenta los siguientes **consejos** para que funcione con todo tipo de contenido (no sólo con páginas HTML) y no desorientar al usuario-a:

- Configura el servidor para que utilice el **código de estado HTTP** apropiado (301). Si puedes utiliza **encabezamientos HTTP** porque reducen el tráfico de Internet y el tiempo de descarga (aplicable a documentos que no sean HTML y

útil para aplicaciones que sólo solicitan una petición de HEAD como por ejemplo, los revisores de vínculos).

Del mismo modo, los códigos de estado 30x te proporcionan información acerca del tipo de traslado ("permanente" o "temporal"), datos que no puedes conocer con una redirección automática.

- Sustituye la página redirigida por una **página estática** que contenga un **vínculo a la nueva página**.

3 Enlaces

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.

13.1 Identifique claramente el objetivo de cada vínculo.

Identificar claramente el destino de los enlaces [13.1 - Prioridad 2]

El **texto de los enlaces** debe ser corto y lo suficientemente **significativo** como para tener sentido cuando se lea **fuera de su contexto**, es decir, cuando se lea sólo el texto del enlace.

El texto de un enlace debe indicar claramente cuál es el tipo de contenido o información que encontrarás en el documento de destino. Los enlaces del tipo [pinche aquí](#) o [siga este enlace](#) no son significativos ya que ocultan hacia dónde dirigen.

Además, procura que los enlaces también sean significativos cuando se lean fuera de su contexto. Algunos usuarios-as de lectores de pantalla navegan con el tabulador saltando de enlace en enlace, con lo que un texto que no sea significativo, apenas aportará ninguna información.

En la siguiente lista puedes ver algunos ejemplos:

- Para ver los resultados de los partidos, [pulse aquí](#). (Mal)
- [Puede ver los resultados de los partidos](#). (Regular)
- Los [resultados de los partidos](#) de la jornada fueron sorprendentes. (Bien)

En la lista que aparece a continuación se muestra una secuencia de enlaces tal y como la percibiría un usuario-a que navegase con un lector de pantalla y recorriese los enlaces con el tabulador. Todos los enlaces dan una idea de dónde dirigen o del tipo de información disponible en el documento de destino, salvo el enlace *pulse aquí*, que no aporta ninguna información cuando se lee fuera de su contexto.

- Página de Inicio
- Mapa del sitio
- Noticias
- ***Pulse aquí***
- Publicaciones
- Contacto
- Enlaces de interés

El atributo "**title**" proporciona un título al enlace que **amplía la información sobre el destino** del mismo si el texto no es lo suficientemente significativo. En caso

contrario evita añadir un título. Los navegadores suelen representar los títulos mediante un mensaje emergente.

Además, los lectores de pantalla y los navegadores de texto no visuales pueden configurarse para que lean el título además del texto del enlace.

Por ejemplo, algunas veces tienes un enlace a un artículo, noticia, documento o libro, con un título poco apropiado para usarlo como texto del enlace. Observa el ejemplo siguiente:

En la red hay disponibles algunos artículos interesantes sobre la accesibilidad en Flash, como por ejemplo "[Desarrollo accesible en Flash - 5 pasos para hacer más accesible su página Flash](#)"

En lugar de ese texto, podrías redactar uno más apropiado:

En la red hay disponibles algunos artículos interesantes sobre la [accesibilidad en Flash](#).

Para este caso, en el atributo "title" del enlace añadimos información adicional sobre el destino, como por ejemplo:

Ejemplo.

```
<a href="#"
  title="Accesibilidad en la red - Desarrollo accesible en Flash
  - 5 pasos para hacer más accesible su página Flash">
  accesibilidad en Flash
</a>
```

Por otra parte, busca la **consistencia en los enlaces**, es decir, procura que aquellos que tengan el mismo texto hagan referencia a un mismo destino. En todo caso, si usas el mismo texto para hacer referencia a destinos distintos entonces diferencia unos enlaces de otros por medio del atributo "title".

Además, proporciona información introductoria en el primer enlace de un grupo de enlaces relacionados entre sí. Así los usuarios-as que naveguen leyendo una secuencia de enlaces mediante el tabulador dispondrán de información contextual sobre los mismos.

[El artículo está disponible en HTML](#), [PDF](#), y [texto](#)

Ejemplo.

```
<a href="articulo.html">
  El artículo está disponible en HTML
</a>
,
<a href="articulo.pdf" title=" El artículo en PDF">
  PDF
</a>
, y
```

```
<a href="articulo.txt" title=" El artículo en formato texto">
  texto
</a>
```

Avisar de la apertura de nuevas ventanas [10.1 - Prioridad 2]

Una de las prácticas que debes evitar es abrir enlaces en nuevas ventanas por varias razones:

- Al cambiar la ventana actual o aparecer inesperadamente nuevas ventanas, los usuarios-as (independientemente de sus características) pueden sentirse desorientados-as.
- Al abrirse un nuevo navegador se pierde el historial de las páginas visitadas y se entorpece la navegación.
- El botón "atrás" para volver a páginas visitadas anteriormente queda inactivo al no contar con un historial en el nuevo navegador abierto.
- En determinados navegadores, si se usan a pantalla completa, el que el botón "atrás" se encuentre inactivo es la única pista que tienen los usuarios-as de que se ha abierto una nueva ventana.

Por tanto, evita especificar una ventana nueva como objetivo ("target") de los enlaces (A o AREA) o de los marcos , así como la apertura de nuevas ventanas mediante scripts con código del tipo *window.open()*.

En algunos casos puedes utilizar ventanas emergentes. Por ejemplo, para abrir documentos no HTML en otro formato (PDF o Word) o para abrir imágenes de gran tamaño, de esta forma el usuario-a puede seguir navegando por el sitio Web mientras en la nueva ventana se descarga el documento seleccionado.

Si usas enlaces que se abran en nuevas ventanas informa al usuario-a previamente. Para avisar de la apertura en nueva ventana puedes usar el atributo "title" del elemento A o bien indicarlo en el texto del enlace o en el del documento antes del mismo. Por ejemplo:

Ejemplo.

```
<a href="prensa.htm" target="_blank">
  Notas de Prensa (Nueva ventana)
</a>

<a href="prensa.htm" target="_blank"
  title="Notas de prensa (se abre en nueva ventana)">
  Notas de Prensa
</a>
```

En el siguiente ejemplo te mostramos cómo crear un enlace en JavaScript para abrir una nueva ventana. En el atributo "title" del enlace se alerta de la apertura de una nueva ventana. De hecho el código Javascript se invoca desde el evento "**onclick**", no desde "href", de forma que si no se soporta JavaScript, se puede acceder a la página de destino en la misma ventana del navegador.

Ejemplo.

```
<a href="prensa.htm"
target="_blank"
onclick="window.open(this.href); return false;"
onkeypress="window.open(this.href); return false;"
title="Notas de prensa (se abre en nueva ventana)">
Notas de prensa
</a>
```

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.

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.

Incluir caracteres entre enlaces [10.5 - Prioridad 3]

Los caracteres imprimibles entre los enlaces posibilitan que los lectores de pantalla hagan una pausa, permitiendo que los usuarios-as diferencien dónde termina un enlace y comienza otro.

Separa los enlaces contiguos situados en la misma línea con un carácter imprimible rodeado de espacios en blanco. Por ejemplo:

[[Sección A](#) | [Sección B](#) | [Sección C](#) | [Sección D](#)]

Ejemplo.

```
[ <a href="a.htm">Sección A</a>
| <a href="b.htm">Sección B</a>
| <a href="c.htm">Sección C</a>
| <a href="d.htm">Sección D</a> ]
```

Si omite caracteres entre los enlaces es posible que algunos lectores de pantalla los interpreten como un único enlace. El resultado sería el siguiente:

[Sección A](#) [Sección B](#) [Sección C](#) [Sección D](#)

Ésto es innecesario para las imágenes que funcionan como enlaces, ya que el marcado que proporcionan los navegadores para identificar la presencia de una imagen ya sirve para delimitar el alcance del enlace.



Ejemplo.

```
<a href="a.htm">
  
</a>
<a href="b.htm">
  
</a>
<a href="c.htm">
  
</a>
<a href="d.htm">
  
</a>
```

Agrupar y saltar enlaces relacionados [13.6 - Prioridad 3]

Es una práctica general situar los enlaces de navegación al inicio de la página. Aquellos usuarios-as que leen el contenido de forma secuencial, como los usuarios-as de lectores de pantalla, tendrán que leer todos los enlaces antes de llegar al contenido de la página. De este modo se ven obligados a leer un elevado número de enlaces (generalmente los mismos) cada vez que accedan a otra página distinta del sitio Web.

Para evitar este problema, agrupa estos enlaces y proporciona mecanismos (enlaces) que permitan saltar a otras secciones del documento.

Un enlace utilizado para saltar secciones es un vínculo convencional a otra zona interna del documento que se ocultará a los usuarios-as visuales mediante hojas de estilo en cascada. El elemento MAP te permite agrupar el conjunto de enlaces para saltar. Por ejemplo, mira la siguiente barra de navegación:

Navegar por el sitio.

[[Sección A](#) | [Sección B](#) | [Sección C](#) | [Sección D](#)]

En este caso podemos agrupar los enlaces de la siguiente forma:

Ejemplo.

```
<map id="barra_nav" name="barra_nav"
title="Barra de navegación">
  <p>Navegar por el sitio.<br>
  <a href="#contenido" class="atajo"
  title="Saltar barra de navegación e ir al
  contenido">
  Ir al contenido
  </a>
  [ <a href="a.htm">Sección A</a> |
  <a href="b.htm">Sección B</a> |
  <a href="c.htm">Sección C</a> |
```

```
<a href="d.htm">Sección D</a> ]  
</p>  
</map>  
...  
<a name="contenido" class="atajo"></a>
```

En el ejemplo ocultamos el enlace "Ir al contenido" mediante CSS para que no se muestre en un navegador visual, pero los usuarios-as de lectores de pantalla podrán seguirlo para saltarse la barra de navegación. El enlace también se visualizará al desactivar CSS.

El objetivo de marcar e identificar los grupos de enlaces relacionados es que los navegadores los reconozcan y proporcionen a los usuarios-a la posibilidad de saltárselos. Sin embargo, para eso es necesario establecer un estándar para marcar los grupos de enlaces (MAP, DIV, FRAME).

En la actualidad la práctica más recomendada es el uso del elemento MAP. Mientras que los navegadores sean incapaces de proporcionar los mecanismos para saltar los grupos de enlaces, son los desarrolladores-as de páginas Web los que deben proporcionar esos mecanismos como se ha explicado antes.

4 Mecanismos de navegación

Prioridad 2

[13.3](#) Proporcione información sobre la maquetación general de un sitio (por ejemplo, mapa del sitio o tabla de contenidos).

[13.4](#) Utilice los mecanismos de navegación de forma coherente.

Proporcionar información general del sitio (mapa del sitio) [13.3 - Prioridad 2]

El uso de **mecanismos de navegación** claros e intuitivos mejora la accesibilidad y la usabilidad.

Estos recursos benefician a usuarios-as que no se pueden guiar por elementos visuales o a aquellos usuarios que sólo pueden acceder a una parte o sección de la página (pantallas pequeñas, dispositivos móviles...).

Los usuarios-as que usan lectores de pantalla y que sólo pueden acceder a la información textual necesitan de mecanismos de navegación adaptados a sus necesidades.

Algunos de los **mecanismos** que facilitan la navegación y permiten al usuario-a localizar la información que busca con mayor rapidez son:

- Barras de navegación.
- Mapa del sitio.
- Tabla de contenidos.
- Funciones de búsqueda a diferentes niveles.

Por ejemplo, puedes establecer el mapa Web mediante una lista anidada de enlaces que represente la estructura y organización del sitio Web.

- [Fundación CTIC](#)
 - [Actuaciones](#)
 - [Unidad Accesibilidad Web](#)
 - [Oficina Española del W3C](#)
 - [Contacto](#)
 - [Intranet](#)

Ejemplo.

```
<ul>
  <li>
    <a href="http://www.fundacionctic.org" title="Nivel 1">
```

```
Fundación CTIC</a>
<ul>
  <li>
    <a href="http://www.fundacionctic.org/lineas"
      title="Nivel 2">Actuaciones</a>
    <ul>
      <li>
        <a href="http://www.fundacionctic.org/uaw"
          title="Nivel 3">
          Unidad Accesibilidad Web</a>
        </li>
      <li>
        <a href="http://www.w3c-es.org/"
          title="Nivel 3">
          Oficina Española del W3C</a>
        </li>
      </ul>
    </li>
    <li>
      <a href="http://www.fundacionctic.org/contacto"
        title="Nivel 2">Contacto</a>
    </li>
    <li>
      <a href="http://www.fundacionctic.org/intranet"
        title="Nivel 2">Intranet</a>
    </li>
  </ul>
</li>
</ul>
```

Para el mapa Web también podrías emplear listas anidadas numeradas, proporcionando información sobre el nivel de anidamiento de la lista y, por tanto, el nivel de profundidad correspondiente en el sitio Web.

Lo más importante a tener en cuenta a la hora de hacer un mapa Web es que éste refleje semánticamente de forma fiel la **estructura del sitio Web**. Para ello, debes utilizar **elementos estructurales** como listas, encabezados, etc.

Usar mecanismos de navegación coherentes [13.4 - Prioridad 2]

Si los mecanismos de navegación son coherentes a lo largo de todo el sitio Web facilitarás su **identificación** y **localización** a la vez que aumentarás la **usabilidad** general del sitio.

De este modo, beneficiarás especialmente a los usuarios-as con problemas de lectura y aprendizaje o que usen lectores de pantalla al facilitar la localización de la información en los sitios Web.

Prioridad 3

[13.5](#) Proporcione barras de navegación para destacar y dar acceso al mecanismo de navegación.

[13.7](#) Si proporciona funciones de búsqueda, permita diferentes tipos de búsquedas para diversos niveles de habilidad y preferencias.

Proporcionar barras de navegación [13.5 - Prioridad 3]

Las barras de navegación facilitan la **localización de la información** por parte del usuario-a. Cuando éste-a se ha familiarizado con el estilo y el funcionamiento de la barra de navegación navegar por el sitio Web sin ninguna dificultad.

Otra ventaja de agrupar los enlaces principales es que se pueden proporcionar mecanismos para saltar estos grupos de enlaces (ver Punto de verificación 13.6).

Un ejemplo de barra de navegación con las principales secciones de un portal es:

[Quiénes somos](#) | [Convenios](#) | [Publicaciones](#) | [CTIC en Prensa](#) | [Patronato](#) 

En la siguiente imagen se puede ver la barra de navegación empleada en Amazon.



Proporcionar funciones de búsqueda de varios niveles [13.7 - Prioridad 3]

La mayoría de los buscadores solicitan al usuario-a que introduzca palabras clave para buscar recursos. Los usuarios-as con dificultades para deletrear o con un idioma diferente al del sitio, tendrán dificultades para encontrar lo que necesitan si la búsqueda requiere una ortografía perfecta.

Los mecanismos de búsqueda deberían incluir un revisor ortográfico, ofrecer alternativas de "la mejor opción", búsquedas mediante preguntas, búsquedas por similitud, etc.

Aunque no es obligatorio proporcionar funcionalidad de búsqueda, es conveniente que, en caso de que lo hagas, proporciones diferentes **niveles de búsqueda**:

- **Búsqueda simple** (por ejemplo, por un término).

Buscar en el sitio:

- **Búsqueda avanzada** (por ejemplo, booleana).

Mediante un formulario en el que se vayan seleccionado opciones para dar opciones de búsqueda más complejas, del tipo:

*Encontrar todos los documentos que contengan la **palabra 1** y la **palabra 2** pero no la **frase 3** en documentos anteriores a la **fecha**.*

5 Metadatos

Prioridad 2

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

Utilizar metadatos para añadir información [13.2 - Prioridad 2]

Los metadatos son básicamente "**datos sobre los datos**". Éstos se añaden a un documento Web y sirven para describir sus características. De esta forma aumenta la semántica de los documentos para que puedan ser leídos y "entendidos" por las aplicaciones (motores de búsqueda, etc.).

Ejemplo.

```
<head>
  <title>Fundación CTIC. Inicio</title>
  <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1">
  <meta http-equiv="Pragma"
    content="no-cache">

  <meta name="Description"
    content="Fundación CTIC. ">
  <meta name="Keywords"
    content="www, fundacion, ctic, tic, ...">
  <meta name="Author"
    content="Fundacion CTIC. Desarrollo">
  <meta name="Copyright"
    content="Fundacion CTIC. 2004">

  <link rel="alternate" type="application/rss+xml"
    title="RSS"
    href="http://www.fundacionctic.org/ctic.rss">

  <link rel="meta"
    href="ctic_about.rdf">

  <link rel="stylesheet" type="text/css"
    href="/css/global.css">
  <link rel="stylesheet" type="text/css"
    href="/css/home.css">
</head>
```

A continuación, puedes familiarizarte con los principales metadatos.

Información de contacto

El elemento **ADDRESS** se puede usar para proporcionar información de contacto o sobre la autoría del documento.

Ejemplo.

```
<address>
  Si tiene alguna duda póngase en contacto
  con el autor del documento
  <a href="../contacto/">Manuel Alonso</a>.
</address>
```

Título de las páginas

Todos los documentos HTML deben tener un elemento **TITLE** en la sección HEAD en el que se indique un título representativo para el documento.

La mala identificación de los documentos (dejándolos en blanco o usando el mismo título) entorpece su usabilidad, presentando varios **inconvenientes**.

- Si un usuario-a desea añadir una página a su colección de "bookmarks" o "favoritos" está obligado a escribir el título para identificarla.
- Si una persona desea desplegar el botón "Atrás" de los navegadores para volver a una página visitada anteriormente tendrá dificultades para distinguir unas de otras si todas tienen el mismo título.
- Los lectores de pantalla leerán siempre el mismo título para todas las páginas. Los usuarios-as que utilizan estas ayudas no pueden diferenciar una página u otra por su contenido visual y, al carecer de un título significativo, tendrían que leer el contenido para diferenciarlas.
- Un usuario-a que abra varias páginas de un mismo sitio Web en múltiples ventanas o pestañas de un navegador, difícilmente diferenciará unas de otras por su título.

También te encontrarás con estos inconvenientes si los títulos de las páginas son poco **significativos**.

La elección de un título apropiado es uno de los principales aspectos a tener en cuenta en cuanto a la usabilidad.

Además, permite optimizar el sitio Web para los motores de búsqueda e incrementar el número de visitas, ya que el título de las páginas es uno de los principales criterios usados al realizar búsquedas.

Si la página carece de título o se usa uno genérico para todas las páginas de un portal, posiblemente se perderá un elevado número de visitas.

Habitualmente los motores de búsqueda muestran el título en sus resultados y, por lo tanto, se convierte en la primera información que reciben los usuarios-as.

El lenguaje utilizado en los **títulos** ha de ser:

- **Significativo.** Cuando una persona lea el título debe obtener la suficiente información sobre el contenido de la página. Evita crear títulos ingeniosos y originales que apenas aporten información real sobre el contenido de la página.

- **Breve:** Si es demasiado largo, los buscadores pueden mostrar sólo una parte del mismo y los usuarios-as pueden pasar rápidamente sobre él sin prestarle mayor atención.
- **Claro:** El título debe informar sobre el contenido evitando la utilización de un estilo demasiado literario.

Un consejo útil es que sitúes el nombre del portal Web en último lugar, dejando el título de la página en primer lugar. Así la información más importante va al principio del título. Por ejemplo:

- "Unidad de accesibilidad Web - Fundacion CTIC"
- "Oficina española del W3C - Fundacion CTIC"

El inconveniente de utilizar títulos de esta forma es que los enlaces no quedan agrupados al añadirlos a los marcadores si éstos se ordenan alfabéticamente. Si deseas evitar esto, puedes situar el nombre del portal Web al inicio del título pero siempre teniendo en cuenta las recomendaciones anteriores. Procura que el nombre del sitio sea conciso para que la información importante, el título de la página, no se desplace demasiado hacia el final del título. Por ejemplo:

- "CTIC - Unidad de accesibilidad Web"
- "CTIC- Oficina española del W3C"

Elemento META (metadatos)

El elemento **META** identifica propiedades de un documento (por ejemplo, el autor-a, copyright, una lista de palabras clave...) y asigna valores a esas propiedades. Cada elemento META especifica una pareja propiedad/valor. El atributo "name" identifica la propiedad y el atributo "content" especifica el valor de la propiedad.

Recuerda: No debes usar el elemento META para redirigir a otras páginas o actualizar páginas automáticamente.

No existe ninguna especificación que defina las propiedades y sus posibles valores. Sin embargo, existen algunas propiedades que son de uso común. Por ejemplo:

- **Description.** Contiene una descripción del documento y lo usan algunos motores de búsqueda para mostrar los resultados.
- **Keywords.** Con esta propiedad puedes especificar una serie de palabras clave que sirvan a los motores de búsqueda para encontrar y clasificar los documentos. Sin embargo, algunos navegadores tienden a ignorar este campo.
- **Author.** Informa sobre la autoría del documento.
- **Copyright.** Esta propiedad señala el copyright del documento.
- **Robots.** Indica a los motores de búsqueda qué acción tomar sobre el documento. Los valores básicos que puede tomar son "index", "noindex", "follow" o "nofollow". Además puedes especificar varios valores a la vez. Cada uno de los valores permite o limita lo siguiente:
 - *index:* permite indexar el documento.
 - *noindex:* impide que se indexe el documento.
 - *follow:* permite recorrer el documento para buscar nuevos enlaces.

- *nofollow*: impide recorrer el documento en busca de nuevos enlaces.

Existen otros valores que puede tomar y que agrupan varios de los anteriores:

- *all*: equivale a "index" y "follow".
- *none*: equivale a "noindex" y "nofollow".

Algunos motores de búsqueda soportan otros valores específicos como "noarchive", que en Google indica que no se almacene el documento en su caché.

- **Dublin Core**. Estándar de metadatos que define un conjunto de propiedades para descripciones bibliográficas electrónicas. Algunas propiedades son: título, autor, descripción, fecha de publicación, idioma, documentos relacionados, copyright... Las propiedades se indican con los caracteres DC seguidos de un punto y el nombre de la propiedad.

A continuación te mostramos algunos ejemplos:

Ejemplo.

```
<meta name="description"
  content="Este documento contiene información sobre
  cómo hacer documentos Web accesibles." lang="es">
<meta name="keywords"
  content="accesibilidad, metadatos">
<meta name="robots"
  content="index, follow">
<meta name="robots"
  content="all">
<meta name="DC.Title"
  content="Pautas de accesibilidad">
<meta name="DC.Creator"
  content="WAI">
<meta name="DC.Description"
  content="Pautas de accesibilidad para el
  contenido en la Web">
<meta name="DC.Date"
  content="1999">
<meta name="DC.Language"
  content="es">
```

Puedes visitar la página de [Dublin Core](#) para obtener más información.

Gramáticas formales

Si validas contra una gramática formal y declaras esa validación al principio del documento con el elemento **!DOCTYPE** permites al usuario-a saber que la estructura del documento es fiable. De este modo las personas sabrán dónde buscar la semántica si fuera necesario (incluido en el apartado "Estándares - Gramáticas formales", punto de verificación 3.2).

Prioridad 3

[11.3](#) Proporcione la información de modo que los usuarios puedan recibir los documentos según sus preferencias (por ejemplo, idioma, tipo de contenido, etc.).

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

Proporcionar documentos alternativos según preferencias [11.3 - Prioridad 3]

La forma más sencilla para que los usuarios-as accedan a los documentos según sus preferencias es proporcionar varios enlaces a diferentes versiones del mismo documento. Por ejemplo:

Versión en español.

Versión en inglés

Versión en francés.

Siempre que puedas, utiliza la negociación de contenido para proporcionar al usuario diferentes opciones:

- Usa diferentes hojas de estilo según las preferencias del usuario-a. Utiliza metadatos con tipos de medio.
- Utiliza negociación de contenido en el servidor.
- Si no puedes usar la negociación de contenido, indica el tipo de contenido o el idioma a través de marcadores (por ejemplo, en HTML, utilizar "type" y "hreflang" en los elementos A o LINK para identificar el lenguaje de los documentos de destino).

Ejemplo.

```
<a href="http://www.w3.org/WAI/" type="text/html" hreflang="en">Página del WAI</a>
```

Puedes leer el [FAQ sobre negociación de lenguaje del W3C](#) si deseas más información.

El elemento **LINK** sirve para proporcionar información a los navegadores o a los motores de búsqueda sobre los documentos.

Por ejemplo, informa sobre los documentos alternativos o la relación existente entre los diferentes documentos de una colección. Éste elemento define un enlace o vínculo y sólo puede aparecer en la sección **HEAD** del documento, pudiendo utilizarlo las veces que necesites.

Los navegadores pueden cargar el documento original o alternativo de forma automática dependiendo de las preferencias, el idioma o el tipo de navegador del usuario-a. Para ello, define el tipo de vínculo (atributo "rel" del elemento LINK) como

"**alternate**" para indicar que se trate de una versión alternativa del documento en el que se encuentra el elemento LINK, y señala para qué casos se usará la versión alternativa.

Por ejemplo, si defines un tipo "alternate" y lo usas junto con el atributo "hreflang" estás indicando una versión del documento en un idioma diferente.

Ejemplo.

```
<link rel="alternate" type="text/html"
  title="Version en francés" lang="es"
  href="index_fr.htm" hreflang="fr">

<link rel="alternate" type="text/html"
  title="Version en inglés" lang="es"
  href="index_en.htm" hreflang="en">
```

El atributo "**hreflang**" del elemento LINK indica el idioma del documento de destino, mientras que el atributo "lang" indica el idioma en el que esté el texto del atributo "title".

Cuando se utiliza "Alternate" con el atributo "**media**" indica una versión alternativa diseñada para un navegador o medio diferente. A continuación te mostramos los descriptores de medios que puedes usar.

- **screen**: pantallas no paginadas de ordenador.
- **tty**: medios que utilicen una cuadrícula de caracteres de ancho fijo, como teletipos, terminales y dispositivos portátiles con posibilidades limitadas de representación.
- **tv**: dispositivos tipo televisión.
- **projection**: proyectores.
- **handheld**: dispositivos de mano, de pantalla pequeña con capacidades gráficas y ancho de banda limitado.
- **print**: documentos paginados y preparados para impresión.
- **braille**: dispositivos táctiles braille.
- **aural**: sintetizadores de voz.
- **all**: todos los dispositivos.

En el siguiente ejemplo te mostramos cómo puedes especificar documentos alternativos para usuarios con diferentes tipos de navegadores.

Ejemplo.

```
<head>
  <title>Titulo de la página</title>
  <link title="Pantalla pequeña"
    rel="alternate" href="smallscreen.html"
    media="handheld">
  <link title="Solo texto"
    rel="alternate" href="texto.html">
```

```
media="aural, braille">
</head>
```

El **control de medios** también se emplea para definir diferentes tipos de hojas de estilo, dependiendo del tipo de navegador del usuario-a.

A continuación, te mostramos cómo utilizar una hoja de estilo principal, en caso de que el documento se presente en una pantalla convencional, y varias hojas de estilo secundarias (para impresión y para dispositivos de pantalla pequeña). En el tipo de vínculo es necesario indicar que se trata de una hoja de estilo mediante **"stylesheet"**.

Ejemplo.

```
<head>
  <title>Título de la página</title>
  <link rel="stylesheet" type="text/css"
    href="estilo.css" media="screen">
  <link rel="stylesheet" type="text/css"
    href="print.css" media="print">
  <link rel="stylesheet" type="text/css"
    href="handheld.css" media="handheld">
</head>
```

El control de medios para hojas de estilo también se puede emplear directamente desde las hojas de estilo mediante las reglas **@media**.

Ejemplo.

```
@media screen{
  p {font-family: verdana, sans-serif;
    font-size: 0.8em}
}

@media print{
  p {font-family: times, serif;
    font-size: 10pt}
}

@media handheld{
  p {font-weight: bold}
}
```

De igual forma, también se puede usar sobre el elemento STYLE en las hojas de estilo internas.

Ejemplo.

```
<head>
  <style type="text/css" media="screen">
  ...
</style>
  <style type="text/css" media="print">
```

```
...  
</style>  
<style type="text/css" media="handheld">  
...  
</style>  
</head>
```

Sin embargo, te recomendamos que utilices la primera opción (con hojas de estilo externas), ya que los navegadores ahorran tiempo si descargan sólo la hoja de estilo correspondiente con el dispositivo usado.

Colecciones de documentos [13.9 - Prioridad 3]

Si deseas presentar una colección de documentos html usa el elemento **LINK** para relacionar los documentos entre sí y mejorar así la accesibilidad. Con este elemento y los diferentes tipos de vínculos puedes describir los mecanismos y organización de la navegación para la colección de documentos.

Algunas aplicaciones de usuario-a sintetizan herramientas de navegación o permiten la impresión ordenada de un grupo de documentos basados en esta etiqueta.

Por ejemplo, algunos navegadores pueden mostrar una barra de herramientas con los enlaces de uso común indicados con el elemento LINK ("next" para el documento siguiente, "prev" para el documento anterior, "index" para el índice...)

En la siguiente lista se muestran los diferentes tipos de vínculos del elemento LINK, relacionados con colecciones de documentos, y el tipo de documento al que hacen referencia:

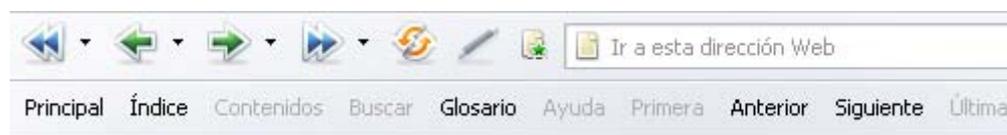
- **Start.** primer documento en una colección de documentos.
- **Next.** siguiente documento en una colección secuencial de documentos. Los agentes de usuario-a pueden precargar el siguiente documento para reducir los tiempos de carga.
- **Prev.** documento anterior en una colección.
- **Contents.** la tabla de contenidos.
- **Index.** índice del documento actual.
- **Glossary.** glosario de términos relacionados con el documento actual.
- **Chapter.** capítulo en una colección de documentos.
- **Section.** sección en una colección de documentos.
- **Subsection.** subsección de una colección de documentos.
- **Appendix.** apéndice de una colección de documentos.
- **Help.** documento con ayuda sobre la navegación, contacto, otros enlaces, etc.

En el siguiente ejemplo mostramos cómo podrías establecer la cabecera de un documento que contendría el capítulo 2 de una colección.

Ejemplo.

```
<head>
  <title>Capítulo 2</title>
  <link rel="Index" href="indice.html">
  <link rel="Start" href="introduccion.html">
  <link rel="Next" href="capitulo3.html">
  <link rel="Prev" href="capitulo1.html">
  <link rel="glossary" href="glosario.html">
</head>
```

En la siguiente imagen puedes ver cómo el navegador Opera crearía automáticamente, basándose en estos enlaces, una barra de navegación para moverse por la colección de documentos.



Otra forma de proporcionar colecciones de documentos es empaquetar los documentos relacionados en un solo archivo (por ejemplo con ZIP, TAR y GZIP, STUFFIT, etc.) que el usuario-a pueda descargar y consultarlos posteriormente sin necesidad de conexión.

6 Independencia de dispositivo

Prioridad 3

[9.4](#) Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos.

[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.

Proporcionar un orden lógico de tabulación [9.4 - Prioridad 3]

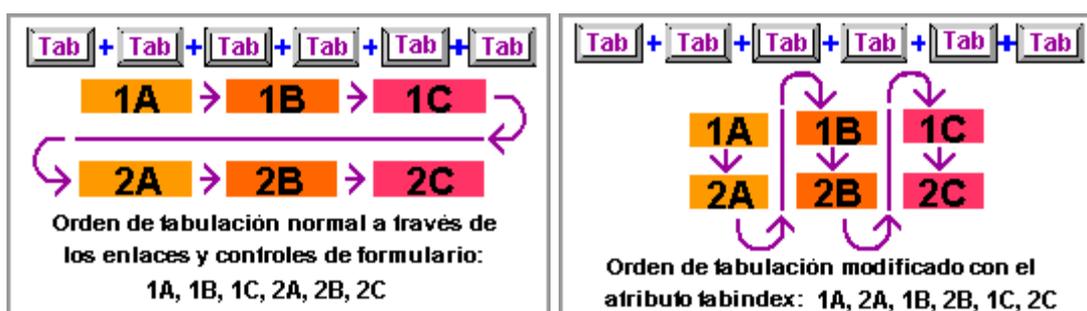
Uno de los principios básicos de la accesibilidad es el diseño **independiente del dispositivo** permitiendo la navegación por teclado u otros dispositivos y no solo con el ratón.

El acceso mediante teclado es el principal medio de interacción de las páginas accesibles ya que una gran parte de las ayudas de usuario-a se basan en el uso del teclado o emulan su funcionamiento.

El orden de tabulación establecido por defecto empieza en el primer enlace o control de formulario del documento y sigue por los siguientes enlaces o controles en el orden en el que aparecen en el código HTML.

En ocasiones este orden resulta inapropiado, con lo que es conveniente redefinirlo.

Para especificar un nuevo orden, emplea el atributo **"tabindex"** (que funciona con los elementos A, AREA, BUTTON, INPUT, SELECT y TEXTAREA) para especificar el orden de tabulación del elemento al que pertenece. Esto permite modificar el orden secuencial y adaptarlo a las necesidades del usuario-a.



Sin embargo, no conviene especificar un orden de tabulación si éste es innecesario (si puede servir el orden por defecto). Procura que el orden de tabulación por defecto sea el correcto mediante una buena disposición lógica del contenido.

El establecimiento de un orden de tabulación lógico, sin necesidad de recurrir al "tabindex", es importante atendiendo a la recomendación del punto de verificación 13.6, sobre la agrupación de enlaces y los mecanismos para saltar esos grupos de enlaces.

Por ejemplo, imagina que una página dispone de los enlaces principales agrupados al comienzo (Inicio, Mapa del sitio, Enlaces, Contacto, etc.) precedidos de un vínculo que permite saltarlos para ir a la sección del contenido principal y que el enlace de "atajo" (el que permite saltar el grupo de enlaces) tiene un orden de tabulación 1 y los del grupo tienen también asignado un "tabindex" (2, 3, 4, ...).

Ejemplo.

```
<map id="barra_navegacion" name="barra_navegacion"
title="Barra de navegación">
  <p>Navegar por el sitio.<br>
  <a href="#contenido" class="atajo" tabindex="1"
  title="Saltar barra de navegación e ir al contenido">
  Ir al contenido</a>
  [ <a href="a.htm" tabindex="2">Sección A</a> |
  <a href="b.htm" tabindex="3">Sección B</a> |
  <a href="c.htm" tabindex="4">Sección C</a> |
  <a href="d.htm" tabindex="5">Sección D</a> ]
  </p>
</map>
...
<a name="contenido" class="atajo"></a>
```

Si accedes con el tabulador al enlace "atajo" y lo accionas, te llevará a la sección del contenido principal. Pero, si después vuelves a usar el tabulador es posible que en algunos navegadores vuelva a dirigirte al enlace con un orden de tabulación 2 (el siguiente al enlace de atajo), en vez de seguir por los enlaces de la sección de contenido.

Por tanto, evita abusar del uso de "tabindex", dejándolo para cuando sea realmente necesario, y basa la tabulación en una disposición lógica y coherente del contenido del documento.

Proporcione atajos de teclado [9.5 - Prioridad 3]

El acceso a los elementos activos de una página es importante para muchos usuarios-as que no pueden manejar un dispositivo de apuntamiento.

Las aplicaciones de usuario-a incorporan características que permiten asignar combinaciones de teclas a ciertas acciones.

HTML 4.01 permite a los desarrolladores-as de contenidos especificar **atajos de teclado** en sus documentos para los enlaces y controles de formulario mediante el atributo "**accesskey**" (que funciona con los elementos A, AREA, BUTTON, INPUT, LABEL, LEGEND, TEXTAREA).

Las acciones que tienen lugar al usar las accesskey dependen del elemento con el que están asignadas. Si la tecla está asociada con un enlace se seguirá el enlace, si se trata de un campo de texto se situará el cursor sobre el mismo, si se trata de un checkbox se seleccionará, etc.

Los navegadores antiguos no reconocen las teclas de acceso rápido, pero sí los actuales.

La forma de usar las teclas de acceso rápido varía en función de la plataforma usada o del navegador. Por ejemplo, en Windows se utiliza Alt+accesskey, mientras que en Mac es Control+accesskey. En la siguiente tabla se muestra las combinaciones de teclas que se utilizan en diferentes plataformas y navegadores:

Combinaciones de teclas en diferentes plataformas

Plataforma	Navegador	Combinación de teclas
PC Windows	Internet Explorer 4+	ALT + AccessKey (enfoca) + ENTER activa
PC Windows	K-meleon	ALT + AccessKey
PC Windows	Mozilla	ALT + AccessKey
PC Windows	Netscape 6+	ALT + AccessKey
PC Windows	Opera 7	SHIFT + ESC, seguido por AccessKey
MAC OS X	I-Cab 2.8	CONTROL + AccessKey
MAC OS X	IE 5 5.1	CONTROL + AccessKey
MAC OS X	Mozilla	No soportado
MAC OS X	Netscape 6+	CONTROL + AccessKey
MAC OS X	OmniWeb 4	No soportado
MAC OS X	Opera	ACCESSKEY COMBINATION
MAC OS X	Safari	CONTROL + AccessKey
MAC OS 9	IE 4 and 4.5	No soportado
MAC OS 9	Mozilla	No soportado
MAC OS 9	Netscape 4	No soportado
MAC OS 9	Opera	No soportado

Sin embargo, el uso de las accesskey plantea una serie de **inconvenientes**:

- Los navegadores suelen usar también teclas de acceso rápido que pueden entrar en conflicto con las definidas en un sitio web.

Por ejemplo, la combinación Alt+A se usa por Internet Explorer para abrir el menú de Archivo, con lo que si se usa dicha letra como accesskey no funcionará en este navegador. Para evitar este problema el navegador Opera permite usar una combinación especial para las teclas de acceso rápido de las páginas web, Shift+Esc+Accesskey.

- Las teclas de acceso rápido usadas por los navegadores varían dependiendo del idioma. Mientras en español la tecla para abrir el menú archivo es la A, en inglés esa letra es la F.

Con lo cual, intentar usar accesskeys en un documento Web sin que interfiera con ninguna letra de ningún navegador (antiguo, actual o futuro) y en cualquier idioma, es muy complicado. Más aún si las teclas que pueden quedar "libres" no pertenecen a la palabra a la que hacen referencia. Es decir, si se usa la letra X para el enlace de "Página de inicio" no hay forma de resaltarla porque no aparece en el texto.

- Las teclas de acceso rápido varían normalmente de un sitio web a otro. El objetivo final es hacer la navegación más fácil a los usuarios-as, pero si tienen

Accesibilidad y Usabilidad Web

que aprenderse teclas de acceso rápido diferentes para cada sitio Web no se alcanza dicho objetivo.

En las aplicaciones de escritorio es normal que las personas sepan usar al menos las combinaciones de teclas básicas comunes para la mayoría de programas de un mismo idioma. Así, la combinación Control+G suele ser "Guardar", Control+C es "Copiar", Control+V es "Pegar", Control+X es "Cortar", Control+Z es "Deshacer", etc.

- Abusar de las teclas de acceso rápido puede provocar el efecto contrario al esperado. El usuario-a debe recordar las teclas de acceso y su uso., y si son muchas se complica.

Para eliminar estos problemas o, al menos minimizarlos, se emplean números del 0 al 9 en vez de letras y se asigna a cada número una funcionalidad común en la mayoría de sitios Web.

De hecho, existen muchos estándares posibles, pero las referencias más usadas son [Clagnut](#) y [DiveIntoAccessibility.org](#).

Las siguientes teclas de acceso rápido surgen al combinar ambas referencias:

- 1 - Página de Inicio
- 2 - Saltar enlaces de navegación.
- 3 - Tabla de contenidos
- 4 - Búsqueda
- 8 - Condiciones de uso
- 9 - Contacto
- 0 - Declaración de accesibilidad (página donde se indican las medidas de accesibilidad usadas en el sitio Web incluyendo las teclas de acceso rápido usadas).

El Web Handbook (Reino Unido) define en las pautas de diseño una lista de teclas de acceso rápido para todos los sitios Web de la administración.

- 1 - Página de inicio
- 2 - Novedades
- 3 - Mapa del sitio
- 4 - Búsqueda
- 5 - Preguntas frecuentes (FAQ)
- 6 - Ayuda
- 7 - Reclamaciones
- 8 - Términos y condiciones
- 9 - Contactar
- 0 - Definición de teclas rápidas

Accesibilidad y Usabilidad Web

Hasta que los navegadores informen de las teclas de acceso rápido, la única manera de conocer su existencia es que se documente en alguna página de la web, indicando su existencia y uso.

Generalmente se suele detallar su existencia en una página del sitio web donde se describen las características de accesibilidad del portal. Ésta página se suele llamar "**declaración de accesibilidad**".

Herramientas de Evaluación y Reparación

Debido a la importancia creciente de la accesibilidad Web, han aparecido una serie de herramientas software para la **evaluación automática** o reparación de los sitios Web. Estas herramientas se basan principalmente en las [Pautas de Accesibilidad al Contenido Web 1.0](#) (WCAG 1.0) desarrolladas por la Iniciativa de Accesibilidad Web del W3C (vistas en el presente curso y de aplicación en el contexto europeo) y la Sección 508 desarrollada para las agencias federales de los Estados Unidos en 1998.

Sin embargo, existen aspectos de accesibilidad que estas herramientas no pueden comprobar, siendo necesaria una **revisión manual** complementaria. La evaluación de la accesibilidad de un sitio Web se ha de realizar siguiendo una metodología en la que se emplearán diferentes técnicas. Las fases fundamentales del proceso de validación son:

- Análisis previo
- Evaluación automática
- Evaluación manual
- Pruebas de usuario-a

1 Análisis previo

En esta fase se identifica el problema para establecer las pautas a seguir durante el proceso de evaluación que se realizará a posteriori. Las principales tareas a realizar en esta fase son las siguientes:

- **Especificación del nivel de adecuación a obtener y el ámbito.** En esta etapa se define el nivel de adecuación que se pretende conseguir en el sitio Web (A, AA o AAA). Además se determina el alcance de la evaluación de accesibilidad sobre el sitio Web. Es decir, *qué* se va a analizar.
- **Análisis del diseño del sitio Web.** En esta fase se trata de identificar aspectos del sitio Web como su complejidad (número de páginas que lo componen) diseño, estructura, las diferentes tipologías de páginas usadas (con el fin de no repetir trabajo en el análisis).
- **Selección de una muestra del sitio Web.** Tomando como base el análisis anterior, se selecciona una muestra representativa del sitio Web para realizar la evaluación. Los validadores automáticos pueden hacer un análisis de un sitio Web completo pero en la evaluación manual, puede resultar imposible realizar esa tarea, con lo que es necesario reducir el problema.

Entre las páginas tomadas como muestra, se ha de incluir la página principal, las páginas de entrada a las diferentes secciones del portal (páginas de primer nivel), las más visitadas, aquellas que son de especial importancia para un sitio Web, así como alguna página de muestra de cada una de las diferentes plantillas usadas en el portal.

- **Elección de las herramientas de evaluación automática.** Se han de escoger al menos dos herramientas de evaluación automática de accesibilidad diferentes. La más usada a nivel mundial es **Bobby**, mientras que la más usada en los países de habla hispana es **TAW**.
- **Evaluación manual preliminar.** Se realiza una primera toma de contacto para obtener una aproximación preliminar e identificar el alcance de los problemas de accesibilidad del sitio Web. En esta fase no se analizan las páginas a nivel de código, sino que se usa una combinación de técnicas manuales y herramientas semi-automáticas, similares a las empleadas en la fase de evaluación manual. En este estadio no se pretende encontrar todos los problemas de accesibilidad del sitio Web.

A grandes rasgos los pasos que daremos son los siguientes:

- Usar un **navegador gráfico** y examinar la muestra de páginas seleccionadas configurando el navegador para comprobar que las imágenes y sonidos tienen equivalentes textuales. Conviene modificar el tamaño de la fuente, la resolución del navegador, desactivar las hojas de estilo y quitar el color para comprobar que la página sigue siendo legible y no se pierde información.
- Usar un **navegador de voz** (como Home Page Reader) o un navegador de texto (como Lynx) y comprobar que se proporciona la misma información que en un navegador gráfico y en el orden lógico correcto.
- Utilizar **herramientas de evaluación automática** y anotar todos los problemas encontrados por esas herramientas.

Accesibilidad y Usabilidad Web

- Realizar un **informe** con los problemas encontrados, indicando el método por el que fueron identificados, así como las buenas prácticas empleadas en el sitio Web. También conviene indicar que se trata de una evaluación preliminar.

2 Evaluación automática

En el proceso de evaluación automática hay que diferenciar entre dos tipos de evaluaciones:

- Sintaxis del código HTML y de las hojas de estilo.
- Accesibilidad.

El primer paso consiste en realizar una comprobación de la **sintaxis** de las páginas, tanto del código HTML como de las hojas de estilo, para verificar que están bien formadas y son válidas.

Recuerda: Si una página Web no cumple con ninguna gramática formal de las publicadas por el W3C, entonces ya no podrá ser accesible en un nivel de adecuación AA (punto de verificación 3.2).

En la evaluación de la **accesibilidad** se emplean herramientas automáticas que permiten identificar algunos de los problemas de accesibilidad del código HTML en un breve periodo de tiempo. Sin embargo, estas herramientas no detectan todos los problemas ya que una gran parte de ellos no se procesan automáticamente.

Por ejemplo, una herramienta de evaluación automática no puede saber si un texto alternativo es adecuado para una imagen. Determina si una imagen tiene o no texto alternativo pero, si lo tiene, no puede comprobar si es correcto, ya que no interpreta el contenido de la imagen.

Las herramientas automáticas han de entenderse como una **ayuda en el proceso de evaluación** y no como un análisis completo.

Si se usan incorrectamente, sin tener clara su función o utilidad, pueden provocar una relajación en los desarrolladores-as que les lleve a creer que están creando sitios Web accesibles cuando en realidad no es así. Por tanto, es necesaria la intervención de un experto-a para completar el proceso de evaluación con otras herramientas semi-automáticas y mediante un análisis manual.

2.1. Evaluación de la sintaxis

Para analizar la sintaxis se debería utilizar una herramienta de evaluación sobre todo el sitio Web o sobre la muestra de páginas seleccionadas.

Algunas de las herramientas de validación disponibles son:

- **W3C HTML Validation Service.** El [validador HTML del W3C](#) es un servicio online gratuito de validación de código HTML basado en un parser SGML. Comprueba la conformidad de los documentos HTML respecto a las gramáticas del W3C y otros estándares HTML (XHTML, XML, MathML).



- **WDG HTML Validator.** El [validador de WDG](#) ([Web Design Group](#)) usa el mismo motor que el validador HTML del W3C pero produce unos mensajes más sencillos y fáciles de entender y soporta un mayor abanico de codificaciones de caracteres.



- **W3C CSS Validator.** Herramienta gratuita para [validar las hojas de estilo CSS](#) solas o presentes en documentos (X)HTML para comprobar si cumplen las especificaciones del W3C. Existe una versión online y una versión descargable, programada en Java y por tanto multiplataforma.



- **W3C HTML Tidy.** El [Tidy](#) (disponible en [sourceforge](#)) es un validador, corrector y limpiador de código HTML que puede convertir los documentos HTML en XHTML. La versión ejecutable funciona en línea de comandos, pero si se desea se puede usar con el interfaz gráfico [TidyGui](#) o una [versión online](#) gratuita.



2.2. Evaluación de la accesibilidad

Se recomienda usar al menos dos herramientas de evaluación automática de la accesibilidad sobre la muestra de páginas seleccionadas y al menos una herramienta sobre todo el sitio Web.

A pesar de ser una inestimable **ayuda** en la evaluación de la accesibilidad de los sitios Web, hay que tener en cuenta que estas herramientas están lejos de ser infalibles y tienen ciertas **limitaciones**, pudiendo dar falsos positivos (considerar como error algo que no lo es) o no detectar algunos errores que el experto-a debe revisar manualmente.

Por tanto, conviene tener claro su funcionamiento para conocer todas sus funcionalidades, ventajas y desventajas. Con la experiencia se puede conocer en profundidad el comportamiento de cada herramienta ante determinados problemas de accesibilidad, de forma que es posible hacer una selección de aquellos puntos de verificación para los que se tiene la certeza que se evalúan correctamente.

A continuación presentamos algunas de los validadores automáticos más utilizado. Para revisar una lista más amplia se puede visitar la página Web del W3C de [Herramientas de Evaluación, Reparación y Transformación para la Accesibilidad del Contenido Web](#).

2.2.1. TAW: Test de Accesibilidad Web

[TAW](#), desarrollado por la [Fundación CTIC](#), es la herramienta de evaluación automática de accesibilidad de habla hispana más importante. Dispone de una versión online y de

otra descargable que permite trabajar sin conexión a Internet, siendo ambas versiones **gratuitas**.



Versión online

En la **versión online** se introduce una dirección URL, la aplicación analiza la página (basándose en las pautas de accesibilidad WCAG 1.0) y genera un informe HTML basado en la página original con información sobre el resultado del análisis.

En la siguiente imagen podemos ver un ejemplo del informe TAW.



El **informe** se divide en varias partes:

- **Cabecera:** muestra el logotipo, versión del TAW y versión de las normas WAI utilizadas para el análisis.
- **Página analizada:** se muestra la página con iconos insertados indicando los problemas de accesibilidad encontrados.
- **Informe resúmen:** se muestran los problemas de accesibilidad encontrados, organizados por prioridad.

El **TAW** genera varios iconos de diferentes colores para representar la prioridad de los puntos de verificación.

A continuación se describen los iconos utilizados por esta herramienta.

- **1** Prioridad 1 de tipo automático

-  Prioridad 1 de tipo manual
-  Prioridad 2 de tipo automático
-  Prioridad 2 de tipo manual
-  Prioridad 3 de tipo automático
-  Prioridad 3 de tipo manual

El **TAW** diferencia entre dos tipos de problemas:

1. **Automáticos:** Aquellos en los que la herramienta tiene la certeza de que incumplen las pautas (por ejemplo, una imagen sin texto alternativo).
2. **Manuales:** La aplicación no los comprueba automáticamente y necesitan una revisión manual por parte del desarrollador-a (por ejemplo, la necesidad del uso de "longdesc" en las imágenes).

En el **resumen** el TAW muestra:

- Un listado con los **problemas de accesibilidad** encontrados (ordenados por prioridad) y su descripción.
- El **número de línea** de la página analizada.
- La **etiqueta HTML** que genera el problemas de accesibilidad (solo en la revisión avanzada).

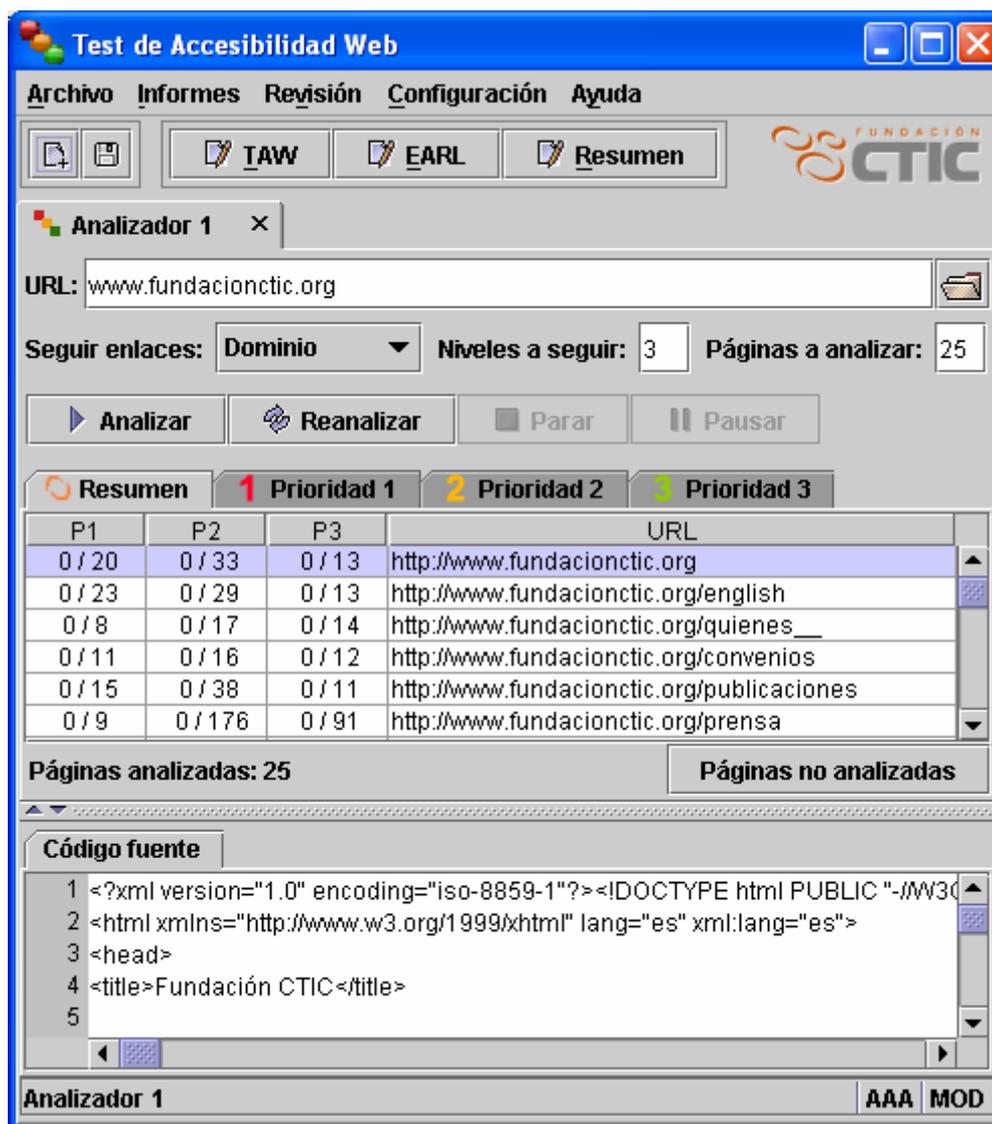
El criterio de la versión online de TAW para considerar que una página es **TAW-válida** es que se cumplan todos los puntos de verificación automáticos de prioridad 1.

Versión descargable

La versión descargable del TAW es una aplicación de escritorio para al análisis automático de sitios Web basándose en las Pautas de Accesibilidad al Contenido Web 1.0. TAW sigue los enlaces de los documentos HTML de forma que permite analizar desde una única página hasta sitios Web completos.

Para el análisis de las páginas, TAW permite la selección de los puntos de verificación a comprobar así como la creación de nuevas reglas personalizadas. Finalmente, genera diferentes tipos de informes sobre el resultado del análisis.

Es multiplataforma y dispone de instaladores para varios sistemas operativos: Windows, Mac OS, Unix y familiares (Linux, Solaris, etc.)



Existe un icono que se puede añadir las páginas Web para indicar que ha superado el test de accesibilidad web del TAW, similar a los iconos del W3C de conformidad con el código de las gramáticas formales.



En la versión descargable es necesario que la página cumpla todos los puntos de verificación de prioridad 1, tanto automáticos como manuales. Para ello, el usuario-a puede marcar todos los puntos de verificación que se cumplen a la vez que realiza la revisión manual.

2.2.2. Bobby

[Bobby](#) es el validador automático de accesibilidad más utilizado a nivel mundial y está desarrollado por [WatchFire](#). La comprobación de accesibilidad se basa tanto en las **pautas del WAI** como en las de la **sección 508** (Estados Unidos).

WatchFire proporcionar dos herramientas para el análisis de accesibilidad que pueden considerarse como dos versiones diferentes de Bobby:

- **WebXACT**: servicio online para probar las características de Bobby y que permite validar una página Web de cada vez.
- **Bobby** (aplicación de escritorio): versión descargable para ejecutarla sin conexión a la red que permite el análisis de archivos locales y de sitios Web completos.

WebXACT

[WebXACT](#) es un servicio online que sólo permite analizar una página cada vez y genera informes de accesibilidad similares a Bobby. Además de realizar un análisis de la accesibilidad también realiza otras comprobaciones sobre aspectos de privacidad (formularios, cookies, etc.) y de calidad de la página (enlaces rotos, compatibilidad con navegadores, etc.).



Bobby (descargable)

La versión descargable permite la creación de proyectos con numerosos parámetros de configuración y el análisis de sitios Web completos. Una vez completada la evaluación muestra un árbol con todas las páginas analizadas. Por cada página muestra:

- **Informe** de los problemas encontrados.
- Su **código HTML**.
- Una **previsualización** para identificar la página.

En la imagen que aparece a continuación podemos ver un ejemplo del tipo de informe que nos ofrece esta herramienta de accesibilidad.

Accessibility - W3C WCAG / AA Site last scanned 16/06/2005 12:48

Pages Report HTML Preview



W3C Web Content Accessibility Guidelines Compliance
For <http://www.google.es/>

✗ This page **does not comply** with all of the automatic and manual checkpoints of the W3C Web Content Accessibility Guidelines, and **requires repairs and manual verification.**

	Automatic Checkpoints			Manual Checkpoints		
	Status	Errors	Instances	Status	Warnings	Instances
Priority 1	✓	0	0	⚠	11	25
Priority 2	✗	4	8	⚠	18	38

[Show All](#) | [Hide All](#)

[Priority 1 Checkpoints](#) [Hide Section](#) ▲ | [Top of Page](#) ↑

✗ Errors
0 tests, 0 instances on page

⚠ Warnings
11 tests, 25 instances on page

Guideline	Instances	Line Numbers
1.1 If an image conveys important information beyond what is in its alternative text, provide an extended description.	4	13, 13, 13, 13
2.1 If you use color to convey information, make sure the information is also represented another way.	5	13, 13, 13, 13, 17

2.2.3. Cynthia Says

El portal Web de [Cynthia Says](#) es un proyecto educacional promovido por varias organizaciones y empresas creado para divulgar la accesibilidad Web entre los desarrolladores-as.

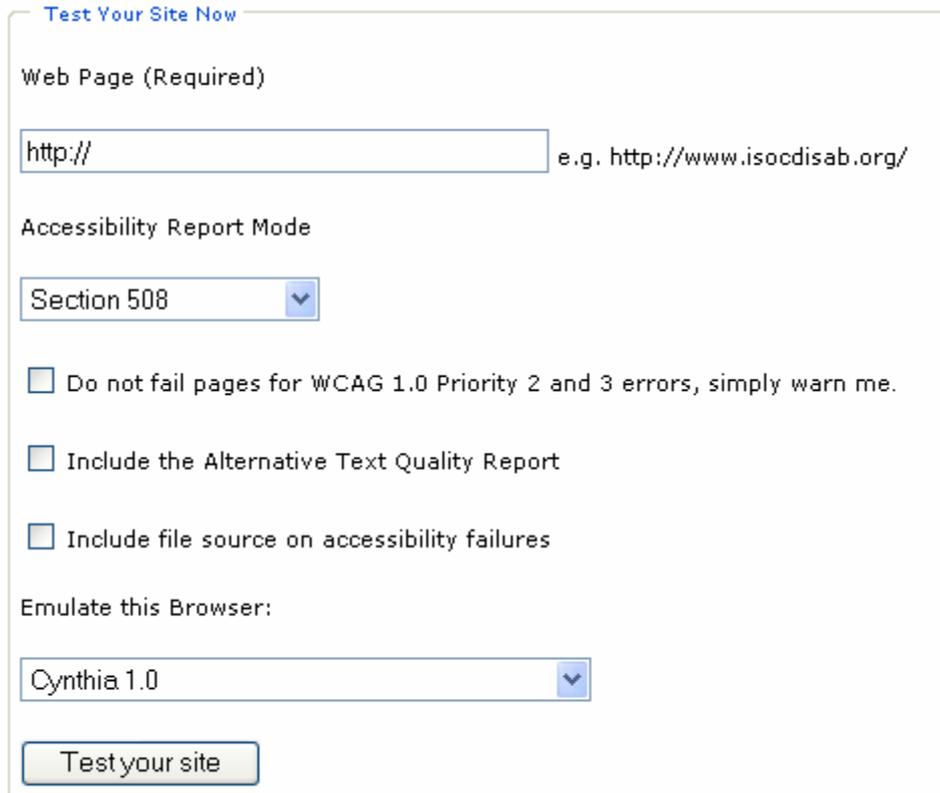
Al igual que Bobby, para hacer los análisis de accesibilidad se puede basar tanto en las pautas del WAI (WCAG) como en la sección 508.

También dispone de dos versiones diferentes.

Versión online

Versión gratuita que permite validar una sola página cada vez. Este validador permite escoger cuáles serán las pautas sobre las que se realizará la comprobación (**Sección 508** o **WCAG**) y permite emular un navegador a la hora de solicitar las páginas.

La siguiente imagen muestra el formulario de configuración de esta herramienta.



Test Your Site Now

Web Page (Required)

e.g. <http://www.isocdisab.org/>

Accessibility Report Mode

▼

Do not fail pages for WCAG 1.0 Priority 2 and 3 errors, simply warn me.

Include the Alternative Text Quality Report

Include file source on accessibility failures

Emulate this Browser:

▼

Versión descargable

La versión descargable es una edición especial de la herramienta [AccVerify](#) de [Hi-Software](#) (**AccVerify Cynthia Says Edition**) con algunas funcionalidades menos y de uso educacional. Es de pago, pero gratuita para escuelas de primaria.

Este proyecto también dispone de un icono que se puede colocar en las páginas que han pasado la validación de accesibilidad.



2.2.4. The Wave

[The Wave](#) es una herramienta de validación y de reparación que se basa en las normas del **WAI** y en la **Sección 508**. Este validador no realiza un análisis exhaustivo de las páginas, sino que pretende ayudar a los desarrolladores-as a realizar aquellas tareas que exigen una toma de decisiones de tipo personal.

En el informe que realiza muestra de forma visual sobre la propia página, entre otras cosas, el texto alternativo de las imágenes para comprobar si es el adecuado, los

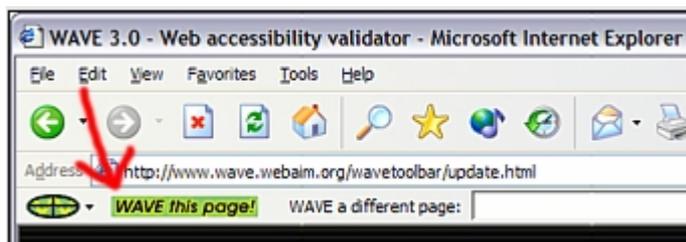
Accesibilidad y Usabilidad Web

títulos, los encabezados, las listas, los bloques de texto y demás elementos estructurales, así como el orden de lectura de la página (tablas, bloques de texto, etc.).

Permite establecer las [preferencias](#) del usuario-a para los análisis y también incluye un enlace a una página con la [explicación de los iconos](#).

The Wave puede utilizarse de diferentes maneras:

- Con el **servicio online** que permite validar tanto una Web pública como subir un fichero local para que lo analice.
- Instalando una **barra de herramientas** en el navegador.



- Añadiendo el **marcador** de The Wave al navegador. La herramienta analiza la accesibilidad de una página pulsando sobre su marcador.

2.2.5. Hera

[HERA](#) (Hojas de Estilos para Revisión de la Accesibilidad) es una herramienta online diseñada para facilitar a los desarrolladores-as la tarea de la revisión manual de accesibilidad de las páginas Web según las Pautas de Accesibilidad del Contenido Web 1.0 (**WCAG 1.0**). Esta herramienta está disponible en varios idiomas, como español, portugués, francés, inglés y danés.

Este validador, desarrollado por la [Fundación SIDAR](#) (Seminario Iberoamericano sobre Discapacidad y Accesibilidad a la Red) es gratuito y su principal objetivo es la realización de actividades que faciliten el desarrollo de la Sociedad de la Información de forma accesible en toda Iberoamérica.



HERA aplica sobre la página original diferentes **hojas de estilos** para identificar visualmente, mediante distintos recuadros y colores, los elementos de la página que presentan problemas de accesibilidad y que necesitan ser revisados.

En la siguiente imagen puedes ver una sección del informe que realiza HERA sobre el texto alternativo para las imágenes (punto de verificación 1.1).

HERA identifica y destaca las imágenes con y sin el atributo "alt", y muestra el código de (cuando el texto aparece subrayado indica que la imagen corresponde a un enlace). Verifique los textos alternativos de las imágenes.



<pre></pre> 	<pre></pre> 	<pre></pre> 	<pre>/i/eu/hp/yes1.gif" height=60 width=232 a</pre> <pre>(noscript)</pre> <pre>en la Web</pre> <pre>=http://eur.i1.yimg.com/eur.yimg.com/i/e</pre> <pre>ento web"></pre>
--	--	--	---

Y! Alojamiento web - Registro de dominios, alojamiento web profesional, serv

Correo gratis	Regístrame	Tu cuenta: Entrar	
Astrología	Corazón	Grupos	Música
Avatares	Correo	Guía Telf.	Móviles
Búsquedas	Deportes	Inmobiliaria	Noticias
Chat	Empleo	Juegos	Postales
Cine	Finanzas	Loterías	Viajes
Coches	Fotos	Messenger	Videos
Compras	Geocities	Mi Yahoo!	más servicios...

```

```



```

```

Yahoo! Cine

- **Batman Begins** : La historia detrás de la leyenda... Trailers, clip, fotos y más
- **La guerra de los mundos**: Gana entradas
- **El Zorro**: vuelve con Antonio Banderas. ¡Fotos!

Cartelera | Trailers | **más...**

3 Evaluación manual

Después del análisis automático es necesario realizar una **evaluación manual** complementaria. Básicamente, el proceso de evaluación manual consiste en probar las páginas HTML con diferentes herramientas, como navegadores o ayudas técnicas, así como la revisión de aquellos puntos de verificación que las herramientas de evaluación automática no pueden comprobar.

A continuación te mostramos algunos de los problemas de accesibilidad que los validadores automáticos no pueden comprobar y que requieren una revisión manual.

- El texto alternativo de las imágenes es el apropiado.
- Hay descripciones largas para las imágenes que lo requieren.
- Se han identificado correctamente las tablas de datos y maquetación.
- En las tablas de datos están bien definidos los encabezamientos.
- En las tablas de maquetación no se usan elementos estructurales para dar formato.
- Las tablas se alinean correctamente.
- El documento puede ser leído sin hojas de estilo.
- La página sigue siendo accesible y la información sigue estando disponible al desactivar los scripts y otros objetos programados.
- No se usan imágenes para transmitir información que se podría transmitir mediante marcadores.
- La información no se transmite sólo por el color.
- Hay suficiente contraste en las imágenes y entre el texto y el color de fondo.
- Se indican los cambios en el idioma empleado.
- Se utiliza un lenguaje claro y sencillo.
- Los elementos de navegación son coherentes en todo el sitio Web y que hay mapa del sitio Web.
- Los equivalentes de un contenido dinámico se actualizan cuando cambia el contenido.
- No hay elementos que provoquen parpadeo o movimiento que pueda afectar a usuarios con problemas fotosensitivos.
- Se indica la expansión de las abreviaturas y acrónimos cuando aparecen por primera vez.
- El acceso por tabulador y las teclas de acceso rápido se usan correctamente.

A continuación, te describimos una serie de pautas y técnicas que pueden ayudarte en el proceso de evaluación manual.

3.1. Leer el contenido de las páginas

Como desarrollador debes leer las páginas para comprobar que:

- El lenguaje usado es **claro y sencillo**.

- La **información importante** está al comienzo de los párrafos.
- El contenido está bien organizado y **estructurado**.
- El **texto de los enlaces** indican correctamente cual es el destino de los mismos.
- Se identifican los **cambios de idioma** y las **abreviaturas**.

3.2. Usar diferentes navegadores gráficos

Muchas páginas están diseñadas para funcionar perfectamente en un determinado navegador, normalmente Internet Explorer. Pero puede que al intentar acceder desde otro navegador, los resultados obtenidos sean desastrosos. Para evitar estos errores debes usar diferentes navegadores gráficos para comprobar cómo se visualizan las páginas en cada uno de ellos. La única forma de conocer si una página se visualizará "correctamente" y sin errores es probarla en los navegadores más comunes y en diferentes versiones.

De todas formas, es prácticamente imposible que la página se visualice de la misma forma en todos los navegadores. Pero esa visualización debe ser "**correcta**" en cada uno de ellos. Es decir, aunque la presentación pueda variar de un navegador a otro, lo importante es que no se dificulte o imposibilite la navegación y el acceso a la información. Un ejemplo típico es la utilización de scripts que dejan de funcionar al emplear otro navegador, desapareciendo o dejando de ser útiles esos elementos de navegación.

Al menos, deberían probarse en diferentes versiones de los **principales navegadores**:

- Internet Explorer
- Mozilla
- Opera
- Netscape
- Galeon
- Konqueror

Es imposible probar la página analizada en todos los navegadores y muy complicado tener instaladas varias versiones del mismo navegador en el mismo sistema operativo. Como mínimo debería probarse sobre una versión de cada navegador y, si es posible, en varios sistemas operativos. Para facilitar esta labor existen servicios, como [BrowserCam](#), que permiten comprobar cómo se visualizarían las páginas en diferentes navegadores y sistemas operativos.



La siguiente imagen muestra el **formulario** para la configuración del servicio de captura de pantalla.

Select Browsers For Each URL

Pages Selected:

<http://www.fundacionctic.org/>

It's easy to order simulations of every URL! Just select the browsers you want below:



Explorer 5.2 - Macintosh OSX 10.3
Mozilla 1.6 - Macintosh OSX 10.3
Netscape 7.0 - Macintosh OSX 10.3
Opera 6.0 - Macintosh OSX 10.3
Safari 1.2 - Macintosh OSX 10.3
Konqueror 3.0.5 - Red Hat Linux 8.0
Konqueror 3.0.5 (no Flash) - Red Hat Linux 8.0
Mozilla 1.6 - Red Hat Linux 8.0
Mozilla 1.6 (no Flash) - Red Hat Linux 8.0
Netscape 4.8 - Red Hat Linux 8.0
Netscape 4.8 (no Flash) - Red Hat Linux 8.0

Browser Window Size: 640x480 800x600 1024x768

Pause seconds after page loads

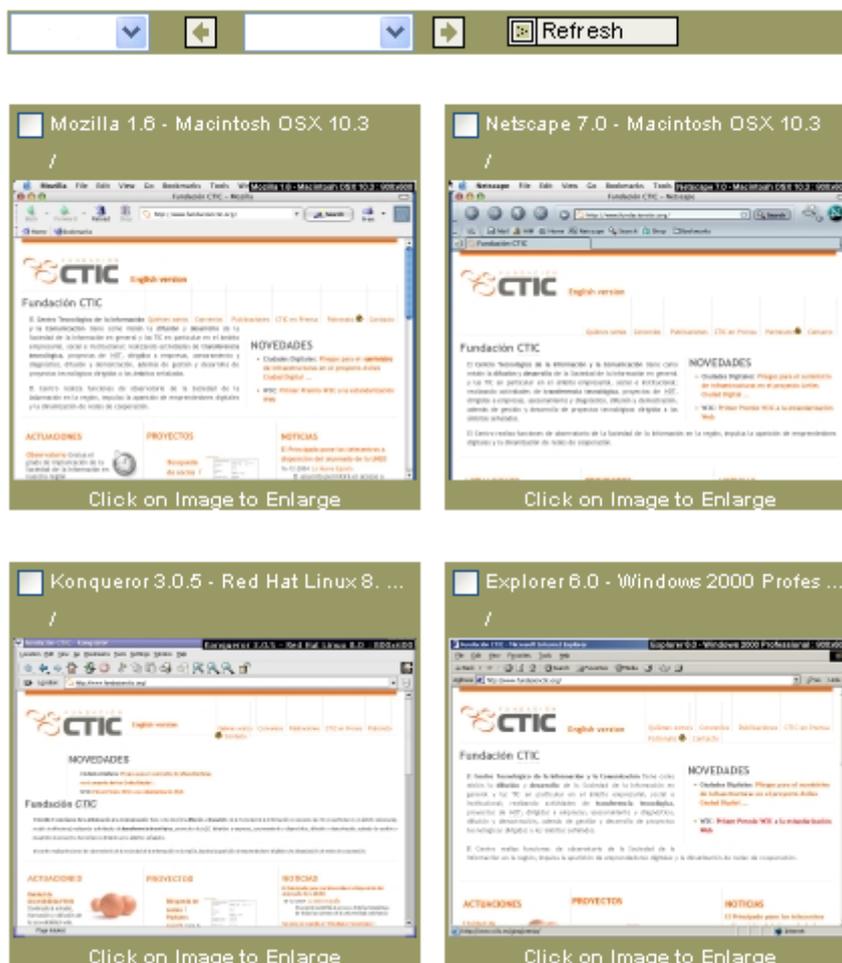
Press the "Page Down" key times after the page loads

Image Output Format:

Full Screen Including Desktop and Popups
 Above the Fold (Browser Window Only)

Make PSD

Una vez enviado el formulario, se generan las **capturas de pantalla** para los diferentes **navegadores** y **sistemas operativos** escogidos, mostrándose en miniaturas que se pueden ampliar pulsando sobre ellas.



3.3. Usar un navegador de texto

Con los **navegadores de texto** puedes observar si el contenido de la página está bien estructurado y en su correcto orden de lectura. De esta forma verás que toda la información ofrecida en un navegador visual también está disponible en uno de texto.

Si una página Web no es accesible con un navegador de texto es muy probable que tampoco lo sea con otros agentes de usuario-a o ayudas técnicas.

Los navegadores de texto más empleados son:

- **LYNX:** El navegador de texto más usado es [Lynx](#). Es gratuito y está disponible para diversos sistemas operativos como VMS, Windows, Linux o Unix. Este navegador puede instalarse en una máquina local o utilizar algún servicio online de emulación como [Delorie Lynx Viewer](#) o [Lynx-me](#).
- **Delorie Lynx Viewer:** Esta versión simula cómo se vería una página con Lynx de forma totalmente funcional. Los enlaces son funcionales y las páginas visitadas se visualizan también en modo texto a través de Lynx Viewer. Además los desarrolladores-as cuentan con otros servicios online gratuitos de Delorie Software, aparte de Lynx Viewer, que pueden ser de utilidad.

- **Lynx-me**: La desventaja de este navegador es que sólo muestra la página como texto, no como una página Web, con lo que no muestra el texto alternativo de las imágenes y no funcionan los enlaces.

3.4. Utilizar navegadores gráficos para realizar comprobaciones especiales

Al mismo tiempo es conveniente que pruebes navegadores gráficos en diferentes circunstancias para comprobar que las páginas siguen siendo accesibles y no se pierde información o funcionalidad.

Ten en cuenta las siguientes indicaciones:

- Desactiva las **imágenes** y comprueba que se muestra un texto alternativo adecuado.
- Desactiva los elementos **multimedia** de la página (audio, vídeos...) para comprobar que existe un texto equivalente apropiado. En los vídeos revisa si hay disponibles subtítulos y transcripciones de la banda sonora.
- Desactiva **JavaScript** y verifica que la página sigue siendo utilizable y se tiene acceso a toda la información. Desactiva también los applets y otros objetos programados como Flash, etc.
- Desactiva las **hojas de estilo** y comprueba que la página sigue siendo utilizable y legible y que el contenido se muestra en el orden correcto.
- Utiliza diferentes tamaños de ventana del navegador y **diferentes resoluciones**. Comprueba que la maquetación sigue siendo utilizable y que no aparecen barras de desplazamiento horizontales.
- Aumenta el **zoom** de la página o el tamaño de letra.
- Analiza la página en **blanco y negro**. Comprueba que presenta suficiente contraste y que no hay información que se base sólo en el color.

Además de estas comprobaciones realizadas con navegadores gráficos, también habría que visualizar la página en otro tipo de dispositivos menos habituales, como dispositivos de pantalla pequeña, monitores monocromos y teclados alternativos.

3.5. Usar teclado en lugar de ratón

Si las páginas son accesibles al navegar a través del teclado es muy probable que sean accesibles con cualquier otro dispositivo de entrada.

Comprueba el funcionamiento del tabulador y, si hay un orden establecido mediante "**tabindex**", mira si es apropiado. Si la página tiene definidas teclas de acceso rápido, comprueba que funcionan correctamente y no entran en conflicto con las teclas de acceso rápido de los navegadores o de los lectores de pantalla.

3.6. Utilizar lectores de pantalla y navegadores de voz

Los **lectores de pantalla** permiten a las personas con discapacidad visual la navegación y la interacción con el sistema operativo y sus aplicaciones, incluyendo los

navegadores Web. Los **navegadores de voz** sólo proporcionan acceso a Internet. Ambos transforman el texto en voz.

Estas herramientas son de utilidad para comprobar si toda la información disponible en una página también está disponible para los usuarios-as invidentes y con un orden lógico de lectura. También permiten comprobar el correcto funcionamiento de la página con estas ayudas técnicas, ya que hay veces que un pequeño error o problema puede interrumpir su ejecución .

A continuación encontrarás una breve descripción de algunas de estas herramientas:

- **JAWS for Windows:** JAWS for Windows, desarrollado por Freedom Scientific, es uno de los lectores de pantalla más conocidos. Esta herramienta permite a los usuarios-as invidentes o de escasa visión usar su ordenador, el sistema operativo y todas las aplicaciones, así como acceder a Internet. En cuanto a la accesibilidad Web, uno de los principales aspectos de JAWS es su integración con Internet Explorer.



Entre las principales desventajas de JAWS (si se va a usar para la evaluación de páginas Web) recordaremos que su uso es complicado y requiere de un periodo de aprendizaje para conocer todas sus funcionalidades. Otro de sus inconvenientes es su elevado precio.

- **Window-Eyes:** Window-Eyes, desarrollado por GW Micro, es posiblemente el segundo lector de pantalla más usado, después de JAWS. También es complicado de utilizar y la curva de aprendizaje es similar a la de JAWS.



- **Home Page Reader** de IBM: Esta herramienta es un *navegador de voz* para personas invidentes o con otras discapacidades visuales. La diferencia entre un navegador de voz y un lector de pantalla es que sólo proporciona acceso a Internet, mientras que los lectores de pantalla funcionan con todas las aplicaciones y con el sistema operativo.

Home Page Reader está desarrollado por IBM y usa el motor del Internet Explorer para la parte de navegador visual.



La versión de evaluación es completamente funcional y está disponible en varios idiomas. Entre otras cosas, permite a los usuarios-as con escasa visión

redimensionar las ventanas, las fuentes, cambiar los colores de fondo y primer plano, etc.

Su precio es bastante inferior al de lectores de pantalla como JAWS o Window-Eyes, aunque tiene menos funcionalidades y capacidad de configuración.

La evaluación de páginas con lectores de pantalla o navegadores de voz no es una tarea sencilla para personas sin discapacidad al no estar acostumbrados a recibir la información sólo de forma auditiva. Para facilitar esta tarea a los desarrolladores, Home Page Reader resalta en la página Web el texto que se está leyendo y, al mismo tiempo, proporciona otra vista con una versión sólo de texto del contenido de la página.

4 Herramientas que facilitan la evaluación manual

Algunas herramientas facilitan el proceso de evaluación manual, como extensiones para diferentes navegadores web que permiten, por ejemplo, cambiar su configuración rápidamente o que proporcionan enlaces a varios validadores gramaticales o de accesibilidad.

A continuación describiremos algunas de estas herramientas.

4.1. Web Accessibility Toolbar

La barra de herramientas [Web Accessibility Toolbar](#) de [AIS/NILS](#) es un plugin para Internet Explorer que ha sido desarrollado para facilitar la evaluación manual de la accesibilidad de las páginas Web.



Algunas de las **funciones** que ofrece son:

- Redimensionar el **tamaño del navegador** para comprobar cómo se ve la página Web en diferentes resoluciones sin tener que modificarla en el monitor.
- Activar y desactivar las **hojas de estilo**.
- Buscar elementos HTML **desaconsejados**.
- Analizar la página mediante enlaces a varios **validadores HTML o CSS**.
- Reconocer, listar o resaltar las **imágenes** presentes o sustituirlas por su texto alternativo.
- Hacer **pruebas de color**, como listar los colores usados o mostrar la página en escala de grises. Esta herramienta ofrece enlaces a direcciones Web para analizar el contraste, así como a las simulaciones de [Vischeck](#) de diferentes tipos de daltonismo.
- Identificar y resaltar los **elementos estructurales** o semánticos de la página Web (encabezados, listas, tablas de datos, frames...). En la siguiente imagen puedes ver que permite conocer algunas propiedades y la jerarquía estructural de un elemento situando el puntero del ratón sobre él.



- Proporciona enlaces a herramientas de evaluación de accesibilidad como **The Wave**, **Cynthia**, **Bobby**, **Webxact**, **TAW**, etc. También enlaza con **Lynx Viewer** para poder ver cómo se vería la página en un navegador de texto.
- **Simulaciones** de distintas **discapacidades visuales** como retinopatía diabética, cataratas, degeneración macular o glaucoma.
- **Información** detallada sobre la **página**, tamaño, velocidad de descarga o los metadatos.

En la página Web de **AIS** hay disponible una [descripción detallada de las funciones](#) de la barra de herramientas.

4.2. Web Developer Toolbar

[Web developer](#) es una extensión para [Mozilla Firefox](#) y [Mozilla](#) que añade una barra de herramientas con varias funciones de utilidad para los desarrolladores Web. Esta barra está enfocada hacia el desarrollador Web en general, aunque también incluye funciones útiles para la evaluación de la accesibilidad.

Web Developer Toolbar puede ejecutarse en cualquier sistema operativo sobre los que se puedan instalar los navegadores Mozilla y Mozilla Firefox, como por ejemplo en **Windows**, **Linux** o **Mac OS**.



Sus **funciones** más destacadas son:

- Deshabilitar **cookies**, **Java**, **JavaScript**, **imágenes**, **animaciones**, **colores** u **hojas de estilo**.

- Definir una **hoja de estilo de usuario-a** o ver el estilo de un elemento de la página con solo hacer click sobre el mismo. También permite **ver y editar** sobre la marcha las hojas de estilo, viendo los resultados al instante.
- Ocultar las **imágenes**, reemplazarlas por el texto alternativo, resaltar las imágenes sin texto alternativo o sin título, así como mostrar sobre la página el tamaño de las imágenes o la ruta donde se encuentran.
- Mostrar **información sobre la página**, los **encabezados HTTP**, el tamaño y tiempo de descarga. La barra muestra visualmente los valores de los atributos "id" y "class", las rutas de los enlaces, los **niveles de los bloques** (DIV), las **teclas de acceso rápido**, el orden de tabulación y los comentarios.
- Limpiar la **caché**, la **autenticación HTTP** y las **cookies** de sesión. Además permite marcar todos los enlaces como visitados, desmarcarlos, aumentar o disminuir la página. Dispone también de una consola de JavaScript donde se muestran los errores, avisos y mensajes en detalle.
- Enlaces a los **documentos del W3C** con las especificaciones de HTML 4.01, XHTML 1.0, XHTML 1.1, DOM 1 y 2, CSS 1 y 2, y WCAG 1.0.
- Marcar con un borde los **elementos estructurales**, **desaconsejados** o a nivel de bloque, los **frames**, los enlaces sin título, las **celdas de las tablas** y otros elementos personalizados.
- **Redimensionar el navegador** a cualquier resolución definida por el usuario-a.
- **Validar el código HTML**, las **hojas de estilo** y comprobar si hay enlaces rotos. Se pueden configurar los enlaces para usar los validadores que prefiera el usuario.
- Analizar la **accesibilidad de la página** con los validadores que trae por defecto en los enlaces o por los definidos por el usuario-a.
- Ver el **código fuente** de la página en un visor que tiene incorporado y que muestra el código con resalte de colores para hacer más fácil su lectura.

4.3.Opera: Opciones de accesibilidad y usabilidad

[Opera](#) es el navegador con más opciones de **accesibilidad y usabilidad** incorporadas por defecto. No tiene tantas funciones como las barras de herramientas mostradas anteriormente, pero pueden ser útiles para evaluar algunos aspectos de los documentos Web.

A continuación puedes ver algunas de las características de Opera relacionadas con la accesibilidad y usabilidad.

- Permite diferentes niveles de **zoom** en la página en los que redimensiona no sólo el texto, sino todo el contenido de la página, incluyendo imágenes y texto definido con unidades absolutas.
- Cambia el **color** de los enlaces, del texto y el **tamaño** de los botones.
- Desactiva las **animaciones**, los **vídeos**, **Java** y **JavaScript**.
- Permite usar una **hoja de estilos propia del usuario-a** en lugar de la hoja del documento. También pueden utilizarse a la vez varias hojas de estilo predefinidas por Opera. Algunas de las más útiles para evaluar la accesibilidad de la página son:

Accesibilidad y Usabilidad Web

- Imitar un navegador de texto
 - Diseño de accesibilidad
 - Alto contraste
 - Desactivar tablas
 - Ver elementos estructurales
 - Analizar con estructura
- Opera ofrece gran cantidad de accesos de teclado para manejar la propia aplicación y navegar por las páginas, el historial, cambiar los estilos, los tamaños, zoom, etc. Además, dispone de una **combinación de teclas especiales** (Shit+Esc+tecla) para usar teclas de acceso rápido que no entren en conflicto con las usadas por el navegador.
 - Muestra en otra pestaña una lista con todos los **enlaces** de la página en el mismo orden en el que aparecen en la misma.
 - La herramienta ofrece una vista en miniatura que permite comprobar cómo se ve la página en un dispositivo de pantalla pequeña. Para ello se sirve de la tecnología "**Small Screen Rendering**" (de Opera) que linealiza el contenido de la página y las tablas y redimensiona las imágenes y textos para que se muestren de la forma más correcta posible en una pantalla pequeña.

La mayor ventaja de la vista en miniatura es que al cambiar a esta opción el navegador modifica el tipo de medio de CSS de "screen" a "handheld". Así, si el sitio se ha diseñado para verse en pantallas pequeñas, se puede alternar entre la hoja de estilo normal y la hoja de estilo definida para medios "handheld" con solo activar o desactivar la vista en miniatura.

En la siguiente imagen se muestra como se vería la página de la fundación CTIC con la tecnología SSR.

Nota: Esta web está diseñada para navegadores que soporten los estándares web (XHTML, CSS, etc). No obstante, se visualiza en otros navegadores.



[English version](#)

[Contenido](#)

- [Quiénes somos](#)
- [Convenios](#)
- [Publicaciones](#)
- [CTIC en Prensa](#)
- [Patronato Acceso restringido](#)
- [Contacto](#)

NOVEDADES

- **Ciudades Digitales:** [Pliegos para el suministro de infraestructuras en el proyecto Aviles Ciudad Digital ...](#)
- **W3C:** [Primer Premio W3C a la estandarización Web](#)

Fundación CTIC

El **Centro Tecnológico de la Información y la Comunicación** tiene como misión la **difusión** y **desarrollo** de la Sociedad de la Información en general y las TIC en particular

5 Pruebas de usuario-a

Hasta ahora has visto cómo se realiza el análisis de las páginas Web a través de herramientas automáticas o por expertos-as que utilizan diferentes metodologías y herramientas para evaluar la accesibilidad de los sitios web.

Ese proceso constituye la parte teórica del proceso de evaluación. Se realizan pruebas automáticas, comprobaciones manuales, tests sintéticos o simulaciones para intentar reproducir situaciones de discapacidad y observar la información disponible. Sólo queda implicar al usuario-a final mediante las pruebas de usuario-a.

Las **pruebas de usuarios-as** complementan el proceso de validación anterior (evaluación automática y manual) al determinar el nivel de accesibilidad de un sitio Web a través de la **experiencia real de usuarios-as con discapacidades**.

A través de las pruebas de usuario-a se observa el nivel de éxito frente a tareas concretas, los problemas encontrados, las preguntas y dudas, rutas de navegación a través de las páginas, etc. Al mismo tiempo se detecta la forma de navegar y los problemas de accesibilidad encontrados por los usuarios-as.

La realización de pruebas de usuario-a implica la búsqueda de perfiles de usuario-a con diferentes discapacidades, niveles de experiencia técnica y niveles de familiaridad con la Web. Además conviene utilizar diversas ayudas técnicas y estrategias de adaptación.

La principal dificultad de estas pruebas es el reclutamiento de participantes y la **gran cantidad de tipologías de discapacidad**, por lo que resulta imposible cubrir todos los tipos y ayudas técnicas.

6 Monitorización continuada

La evaluación de accesibilidad sólo puede asegurar que un determinado sitio Web tiene cierto nivel de adecuación en la fecha en la que se realiza el análisis.

En los **sitios Web estáticos**, la certificación de un determinado nivel de accesibilidad puede tener carácter permanente, pero la práctica totalidad de los sitios Web actuales son dinámicos y, por tanto, están sujetos a numerosos cambios en sus contenidos. En un entorno tan cambiante, la evaluación de accesibilidad sólo tiene validez para la fecha de certificación.

Los **sitios Web dinámicos** usan diferentes plantillas para las páginas que comparten diseño, cambiando de una a otra página sólo su contenido, para el que se usan los gestores de contenido o CMS (Content Management System).

Para aumentar la probabilidad de que un sitio Web mantenga un nivel de adecuación en el tiempo, deben realizarse las siguientes acciones:

1. Identificar claramente el nivel de adecuación deseado y el ámbito o alcance del sitio Web al que se le aplicará.
2. Establecer las personas responsables del proceso de monitorización, y crear procedimientos que puedan usarse rápidamente para que las páginas que no sean accesibles lo vuelvan a ser.
3. Realizar especificaciones claras sobre la frecuencia, el método, y el alcance de las evaluaciones.
4. Evaluar todas las páginas y plantillas nuevas antes de que se añadan al sitio Web.
5. Disponer del software adecuado para facilitar la evaluación.
6. Añadir al sitio Web direcciones para aportar información (feedback) sobre la accesibilidad del mismo.

7 Herramientas de reparación

Las **herramientas de reparación** ayudan a los desarrolladore-as a corregir errores de accesibilidad presentes en las páginas Web. Debido a la propia dificultad de la reparación de accesibilidad, muchas de las herramientas de reparación se centran sólo en determinados aspectos específicos de accesibilidad.

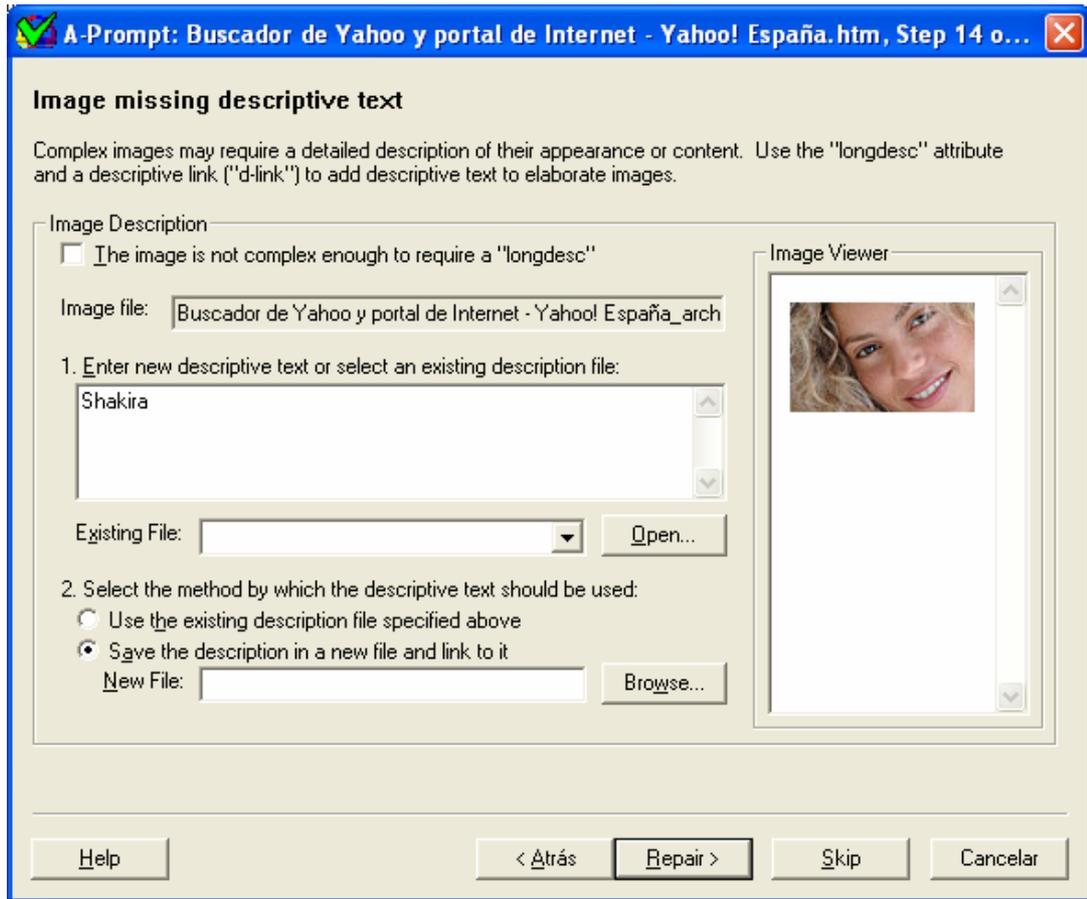
Por ejemplo, hay herramientas dedicadas exclusivamente al subtítulo de presentaciones multimedia. Otras herramientas incorporan características de evaluación, buscan los problemas de accesibilidad, y permiten corregirlos sobre la marcha mostrando ventanas o diálogos que facilitan la reparación a los desarrolladores-as.

Algunas de las herramientas de reparación más importantes son:

- **A-Prompt**

Desarrollado por la Universidad de Toronto, [A-Prompt](#) es una herramienta gratuita para identificar y reparar problemas de accesibilidad. Algunas de las reparaciones que se pueden realizar con A-Prompt son crear textos alternativos para los elementos no textuales, asignar títulos para todos aquellos elementos que lo requieran o crear encabezados para las tablas de datos, entre otras cosas.

En la siguiente imagen podemos ver un ejemplo del tipo de mensajes que ofrece esta herramienta.



- **InFocus**

[InFocus Suite](#), de [SSB Technologies](#), es una herramienta para la evaluación, monitorización y reparación de accesibilidad de sitios y aplicaciones Web. Dispone de dos versiones, InFocus Desktop e InFocus Enterprise. Está disponible para varios sistemas operativos como Windows, Linux, Unix y Mac OS.

InFocus Suite

- **Tidy**

Además de un validador de la sintaxis sirve para corregir y limpiar de código HTML, pudiendo convertir los documentos HTML en XHTML. Para más información puedes visitar la [página de su autor](#) o la [página de Tidy de Sourceforge](#).



- **AccRepair**

[AccRepair](#) está desarrollada por HiSoftware (AccVerify, AccMonitor, HiCaption). Esta herramienta sirve tanto para la evaluación como para la reparación de la

accesibilidad de los sitios Web, soportando las WCAG y la Sección 508. Incorpora todas las características de AccVerify añadiéndole capacidades de reparación.

ACCRepair puede utilizarse de forma individual o integrado con Microsoft FrontPage o Microsoft Office.



- **HiCaption**

[Hi-caption](#) es una herramienta desarrollada por HiSoftware que sirve para crear o editar subtítulos para presentaciones en SMIL o SAMI.



- **MAGpie**

Esta herramienta es similar a HiCaption y fue desarrollada por [NCAM](#) (National Center for Accessible Media). [MAGpie](#), Media Access Generator, se utiliza para crear subtítulos y descripciones de audio para archivos multimedia, Quicktime, Windows Media o SMIL.



8 Bibliografía

- [*Web Content Accessibility Guidelines 1.0*](#). Recomendación del W3C, 5 de Mayo de 1999. Wendy Chisholm, Gregg Vanderheiden, Ian Jacobs.
- [*Alternative Web Browsing*](#). Peter Boshier y Judy Brewer (2001)
- [*Evaluation, Repair, and Transformation Tools for Web Content Accessibility*](#). Shadi Abou-Zahra, Wendy Chisholm, Judy Brewer.
- [*Evaluating Web Sites for Accessibility*](#). Judy Brewer, Chuck Letourneau.
- [*Comprendiendo la Accesibilidad: Una Guía para Lograr la Conformidad en Sitios Web e Intranets*](#). Robert B. Yonaitis (2002, HiSoftware Inc.).