



Guía para construir sitios web accesibles Aplicación a Proyectos EQUAL

Guía para construir sitios web accesibles – Aplicación a Proyectos EQUAL



Edita Unidad Administradora del Fondo Social Europeo
Ministerio de Trabajo y Asuntos Sociales

Realiza Fundación CTIC • www.fundacionctic.org

Esta guía técnica está basada en el trabajo del WAI-W3C, y más concretamente en el currículo y pautas WCAG 1.0 (pautas de accesibilidad para contenido web 1.0).

Todos los nombres propios de programas, sistemas operativos y equipos de hardware que aparecen en este guía, son marcas registradas de las respectivas compañías u organizaciones.

Diseño Serrador & Asociados

Impresión Egraf, S. A.

Edición Junio 2006

NIPO 201-06-139-8

Depósito Legal M. 29943-2006

Prólogo

La Iniciativa Comunitaria EQUAL se dirige a aquellas personas cuya situación de discriminación y desigualdad les impide su incorporación a procesos de integración en el mercado de trabajo y el acceso a las políticas de fomento del empleo. De esta forma, EQUAL se erige en parte esencial de la estrategia de la Unión Europea para la promoción de la inclusión social.

Por otro lado, las Tecnologías de la Información y la Comunicación se han convertido en un factor fundamental para el crecimiento económico, fomentando la competitividad empresarial y mejorando el funcionamiento de todos los sectores económicos. Su aplicación en todos los procesos y ámbitos de nuestra sociedad -económico, social, cultural y político-, es lo que se denomina la Sociedad de la Información, y la accesibilidad web es la capacidad de que todas y todos los usuarios de Internet puedan acceder a estos recursos, es la posibilidad de que un sitio o servicio web sea visitado y utilizado de forma satisfactoria por el mayor número de personas, independientemente de las limitaciones que éstas tengan o las derivadas de su entorno.

Asimismo, la Web (World Wide Web) creada como una red universal de conocimiento, ha supuesto un enorme salto cualitativo y cuantitativo en cuanto a la adquisición y tratamiento de información se refiere. Sin embargo, y debido a diferentes motivos, se ha convertido también en una barrera para muchos ciudadanos y ciudadanas y sobre todo, para las personas con discapacidad.

Por eso, la accesibilidad es un elemento esencial que favorece la igualdad de oportunidades de las personas, en especial, de aquellas con discapacidad, permitiendo el acceso al trabajo, la formación, la cultura, el ocio y el tiempo libre.

Ya en el año 2000, la Comunicación de la Comisión titulada «Estrategias para la creación de empleo en la sociedad de la información» [COM(2000) 48], aprobada el 4 de febrero de 2000, propugnaba que tanto la Estrategia Europea para el Empleo como la iniciativa e-Europe, abordaran específicamente la promoción del empleo de las personas con discapacidad y recomendaba a las industrias de la Sociedad de la Información que velaran por que a partir de 2000 se aplicasen normas que propugnaran el uso de equipos de fácil utilización.

Por lo que se refiere a EQUAL, la Sociedad de la Información constituye un principio transversal de la misma. A través del fomento y la utilización de las Tecnologías de la Información y la Comunicación se intenta facilitar la incorporación y el mantenimiento de las personas más vulnerables en el mercado laboral. La accesibilidad supone una herramienta importante para la consecución del objetivo final de EQUAL, que es, en definitiva, la inclusión de todos y todas en el mercado laboral.

El objetivo de esta guía es lograr que, en los proyectos que se están desarrollando en la Iniciativa Comunitaria EQUAL, se tenga un conocimiento tanto por parte de los responsables de proyecto y gestores, como por parte del personal técnico sobre qué es la accesibilidad web, qué normas se deben seguir en la elaboración de los productos, mate-



riales, páginas web, herramientas, etc., que estén realizadas o soportadas en las Tecnologías de la Información y la Comunicación, y lograr que todas las personas, en especial aquellas con discapacidad, participen con pleno derecho y en igualdad de condiciones del acceso a la información, a la formación y a los servicios que las Tecnologías de la Información y la Comunicación les ofrecen.

Es importante que desde los proyectos Equal, se sigan las pautas, normas y recomendaciones necesarias para lograr una mejor implantación de servicios y una mejora de condiciones de los contenidos de los sitios web. Ello facilitará la integración y la participación de todas las personas, en especial aquellas con discapacidad. Desde esta guía se describen y justifican recomendaciones de codificación y se ofrecen alternativas posibles en cada caso. Con este formato pretendemos que el usuario/a tenga una visión lo más real posible de los errores que se cometen de forma habitual y de las mejores prácticas para corregirlos.

Estas condiciones incluyen, entre otros, el nivel de adecuación que se deberá alcanzar en relación a las Pautas de Accesibilidad para el Contenido Web (WAI-W3C1), que constituyen recomendaciones universalmente aceptadas, y también una aproximación a la terminología, tecnologías empleadas, entidades relacionadas y legislación.

Subdirección General de la UAFSE

1.-WAI-Iniciativa para la Accesibilidad Web desarrollada por el W3C. Ver capítulo "organismos".

2.-W3C (Consortio para la web) o consorcio de organizaciones unidas para desarrollar tecnologías estándares con el fin de guiar la web hacia su máximo potencial. Ver capítulo "organismos".

Índice

Introducción a la accesibilidad web

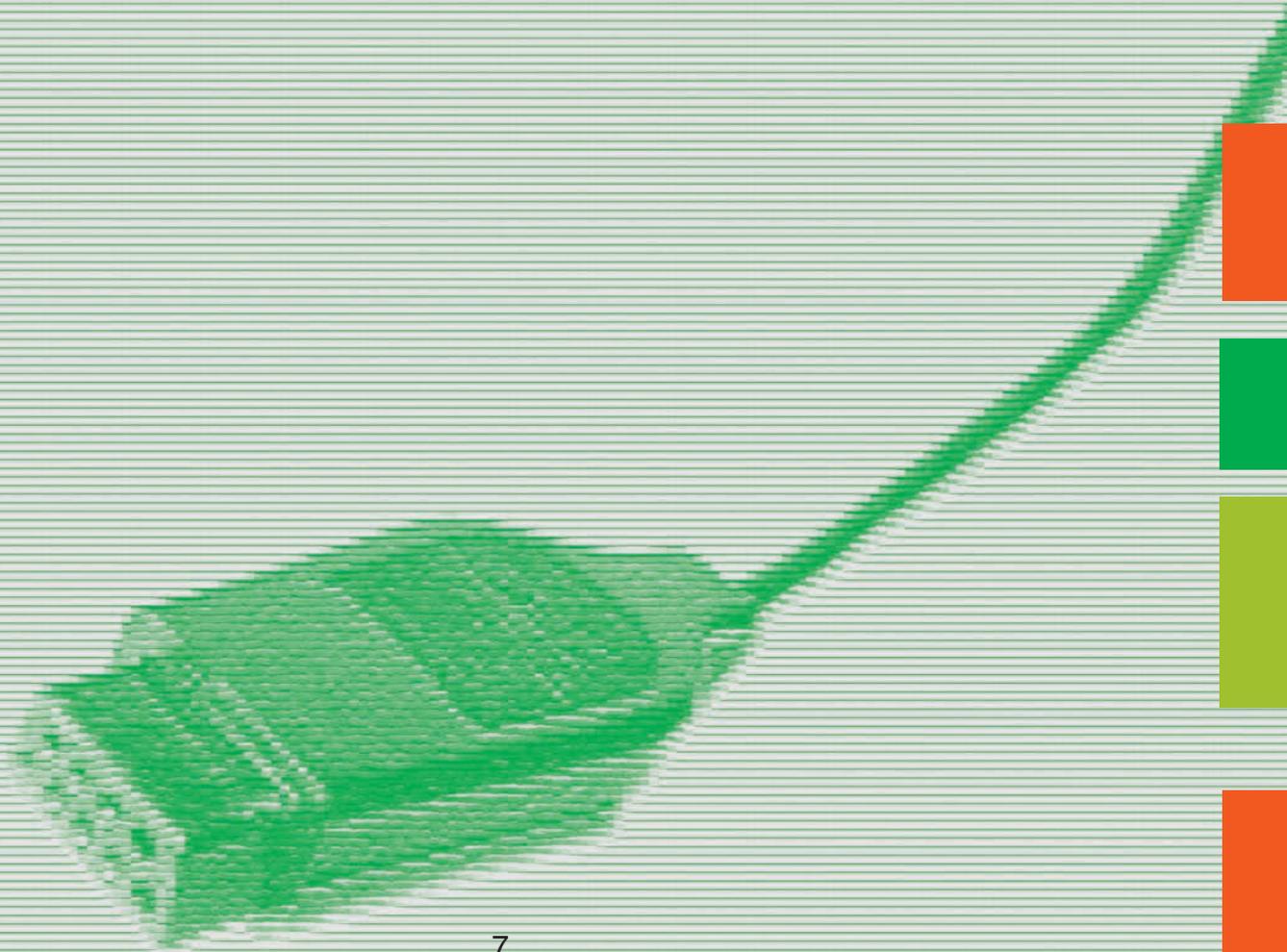
Consideraciones técnicas	pag 9
Diseño para todos y todas	pag 11
Organismos	pag 12
• W3C	
• WAI	
Ayudas técnicas	pag 14
Ventajas de la accesibilidad web	pag 16
Referencia normativa	pag 17

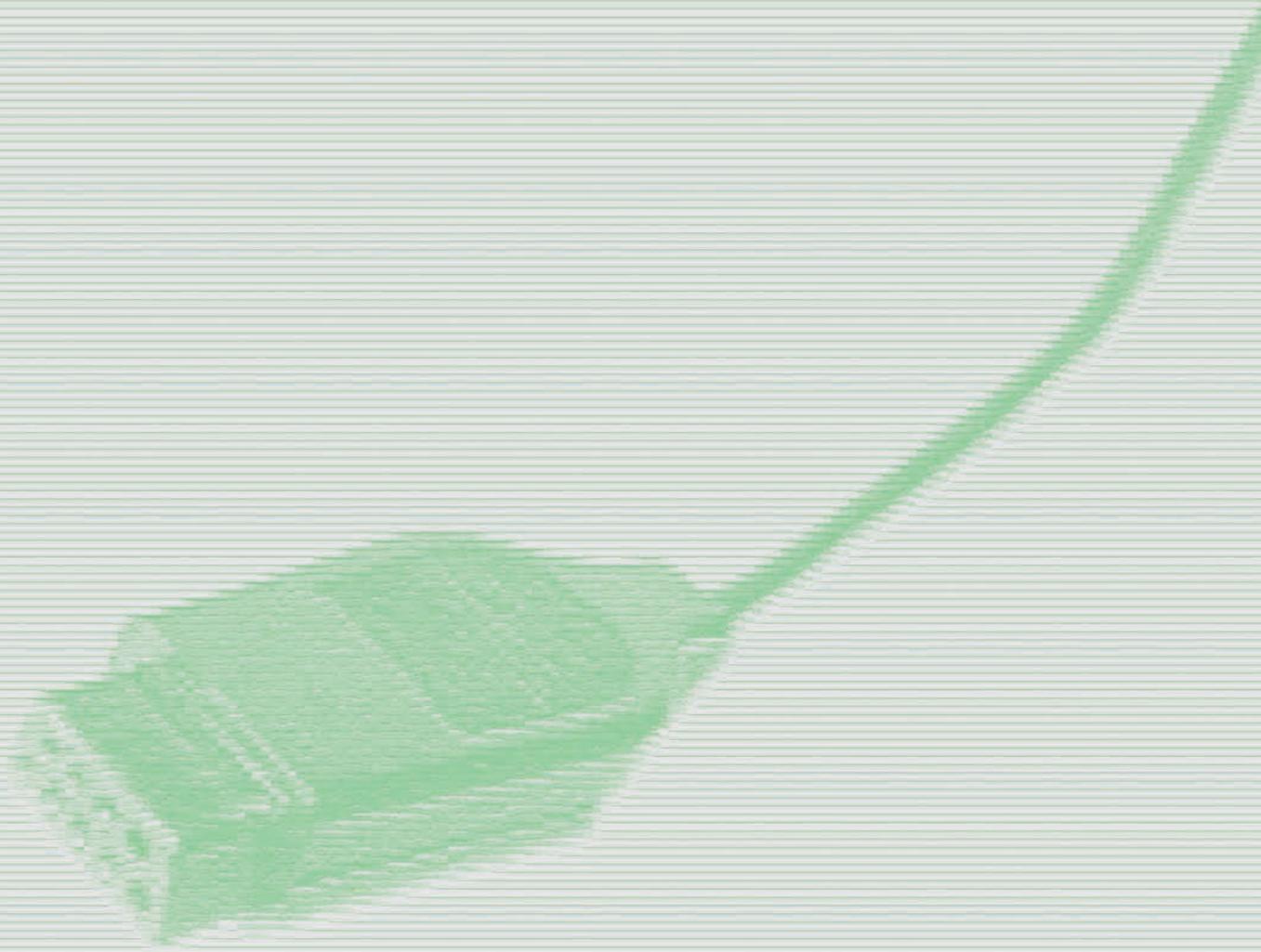
Accesibilidad web

Organización de la guía	pag 22
Imágenes y Multimedia	pag 24
• Imágenes	
• Mapas de imagen	
• Multimedia	
Objetos Programados	pag 31
• Equivalentes accesibles	
• Contenido dinámico	
• Independencia de dispositivo	
Marcos	pag 38
• Equivalentes textuales	
Tablas	pag 41
• Tablas de datos	
• Tablas de maquetación	
Formularios	pag 47
• JavaScript y formularios	
• Asociación entre etiquetas y controles	
• Independencia de dispositivo	
• Otros aspectos sobre accesibilidad	

Estándares <ul style="list-style-type: none"> • Gramáticas formales • Tecnologías W3C • Elementos desaconsejados • Página alternativa accesible 	pag 54
Estructura y Presentación <ul style="list-style-type: none"> • Hojas de estilo • Color y contraste • Encabezamientos • Listas • Citas • Dividir la información • Marcadores mejor que imágenes 	pag 59
Navegación, lenguaje y comprensión <ul style="list-style-type: none"> • Lenguaje y comprensión • Actualizaciones automáticas y redireccionamiento • Enlaces • Mecanismos de navegación • Metadatos • Independencia de dispositivo 	pag 79
Evaluación de accesibilidad	pag 101
Apéndice <ul style="list-style-type: none"> • Pautas de Accesibilidad al Contenido en la Web 1.0 	pag 107
Bibliografía	pag 126

Introducción a la accesibilidad web





Consideraciones técnicas

Algunas personas con discapacidad interactúan con la web mediante el uso de dispositivos físicos (por ejemplo, cinta braille) o programas (ejemplo: lector de pantalla), denominados **ayudas técnicas**, que convierten el texto en algo asequible para ellos (puntos braille, voz).

En un nivel más concreto, cuando hablamos de accesibilidad web, nos referimos a las condiciones que tiene una web para que las personas con discapacidad interactúen de forma completa, es decir, sin pérdida de información ni funcionalidad. Para ello, y para que las páginas web sean compatibles con las ayudas técnicas, el sitio web debe ser programado y diseñado sobre la base de unas pautas.

En este sentido, las normas que en la actualidad gozan de una mayor difusión y aceptación son las **Pautas de Accesibilidad para el Contenido Web 1.0 (WCAG 1.0)**. Estas recomendaciones fueron publicadas en mayo de 1999 por el WAI, iniciativa emprendida por el W3C (World Wide Web Consortium, <http://www.w3c.org>). **Dichas normas son reconocidas internacionalmente como un estándar en accesibilidad.**

Cada una de las 14 pautas en que se dividen las WCAG 1.0, se subdividen a su vez en puntos de verificación, que tienen una prioridad mayor o menor según su impacto en la accesibilidad final.

Así, tendremos que los puntos de verificación con **prioridad 1** son los de mayor importancia en cuanto a la accesibilidad final, los de **prioridad 2** deben ser observados si queremos eliminar importantes barreras de acceso y los de **prioridad 3** confieren a la web un mejor nivel de accesibilidad.

Existe también una clasificación del grado de adecuación de una web en relación con las normas y prioridades del W3C-WAI (que será el utilizado en esta guía):

- Nivel "A" (A). Incluye los puntos de verificación de nivel 1.
- Nivel "AA" (AA). Incluye los puntos de verificación del nivel 1 y 2.
- Nivel "AAA" (AAA). Se deben cumplir las tres prioridades.

A continuación, detallamos la estructura y documentos que conforman las Pautas de Accesibilidad para el contenido web y que servirán, en su caso, de referencia técnica:

Pautas de Accesibilidad al Contenido en la Web 1.0.

Este documento contiene las 14 pautas en que se dividen las WCAG 1.0, así como los respectivos puntos de verificación de cada una de ellas y accesos directos a los diferentes documentos de técnicas.

Técnicas para las Pautas de Accesibilidad al Contenido en la Web 1.0.

Se trata de un conjunto de documentos complementarios al apartado anterior, en los que se describen los procedimientos para aplicar dichas pautas. Presenta mediante descripciones y ejemplos prácticos cómo debemos diseñar nuestras páginas web para que éstas cumplan los requisitos de accesibilidad.

Temas generales de las técnicas de accesibilidad.

- Técnicas fundamentales.
- Técnicas HTML.
- Técnicas CSS.

Tabla de Puntos de Verificación para la Pautas de Accesibilidad al Contenido en la Web 1.0

Este último documento contiene la tabla con todos los puntos de verificación de las pautas, agrupados por prioridad y por tipología de elemento (imágenes, tablas, JavaScript). Permite al desarrollador y al validador repasar su trabajo y comprobar el grado de accesibilidad alcanzado en su diseño. También se utiliza como referencia rápida a los puntos de verificación, según su prioridad y tipo.

Diseño para todos y todas

Otro de los cánones asociados con la accesibilidad son los principios del **diseño para todos o diseño universal**. Los principios del denominado diseño para todos 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 forma especial.

El diseño universal, así pues, beneficia a todas las personas, sean cuales sean su edad y/o habilidades.

Estos principios son:

1. **Igualdad de uso:** el diseño debe ser fácil de usar y adecuado para todas las personas independientemente de sus capacidades y habilidades. Debe proporcionarla misma forma de uso a todos los usuarios: idénticas, cuando sea posible; equivalentes, cuando no lo sea.
2. **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 a su ritmo de uso.
3. **Simple e intuitivo:** el diseño debe ser fácil de entender independientemente de la experiencia, los conocimientos, las habilidades o el nivel de concentración del usuario. Esto elimina la complejidad innecesaria y prioriza la entrega de información acorde a su importancia.
4. **Información fácil de percibir:** el diseño debe ser capaz de intercambiar información con el usuario, independientemente de sus condiciones ambientales o sus capacidades sensoriales.

Algunos ejemplos de ello son la presentación por medios redundantes (texto, voz), la mejora de la legibilidad de la información esencial, la compatibilidad con las ayudas técnicas, etc.

5. **Tolerante a errores:** el diseño debe minimizar las acciones accidentales o fortuitas que puedan tener consecuencias fatales o no deseadas. Es principio del diseño universal la proactividad para evitar los posibles errores que el usuario pueda cometer en su interacción con la web.
6. **Escaso esfuerzo físico:** el diseño debe poder ser usado eficazmente y con el mínimo esfuerzo posible. Por ejemplo, evitando las acciones repetitivas.
7. **Dimensiones apropiadas:** los tamaños y espacios deben ser apropiados para su manipulación y uso por parte del usuario, independientemente de su tamaño, posición, y movilidad.

Organismos

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

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

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 para trabajar en conjunción con otros sistemas o productos, sin que sea necesario 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.

Los principales objetivos del W3C son:

Acceso universal

Hacer la web accesible a todos, promocionando tecnologías que tengan en cuenta las enormes diferencias de 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 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.

En octubre de 2003, se inauguró la oficina española del W3C (<http://www.w3c.es>), 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. Está hospedada por la Fundación CTIC (Centro Tecnológico para la Información y la Comunicación, <http://www.fundacionctic.org>), cuya misión es la difusión y desarrollo de la Sociedad de la Información y las Tecnologías de la Información y la Comunicación. Gracias a la creación de la oficina española, ha aumentado la representación de la comunidad española en el W3C.

WAI

El compromiso del W3C de guiar la web hasta su máximo potencial, incluye alcanzar un alto grado de funcionalidad para personas con discapacidades. De ahí que desarrollase la Iniciativa de Accesibilidad Web (WAI).

Esta iniciativa publicó en mayo de 1999 las Directrices de Accesibilidad para el Contenido Web (WCAG 1.0), que explican cómo deben crearse páginas y sitios web accesibles y que son el objeto de estudio de la presente guía.

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 la industria, colectivos de discapacitados, gobiernos, escuelas, universidades, etc. de todo el mundo.



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 desarrollar fácilmente. Así, una silla de ruedas, una máquina lectora o un audífono son ayudas técnicas.

Dentro del ámbito 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.

Líneas braille

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.

Lectores de pantalla

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).

JAWS® for Windows® (<http://www.freedomscientific.com>)

Window-Eyes® (<http://www.gwmicro.com/>)

Navegadores de voz

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

Home Page Reader® (http://www-306.ibm.com/able/solution_offerings/hpr.html)

Navegadores de texto

Son aplicaciones utilizadas por algunos usuarios/as ciegos en conjunción con algún sintetizador de voz. También los pueden usar personas con conexiones lentas o que utilizan dispositivos de pantalla con escasa capacidad gráfica.

LYNX (<http://lynx.browser.org/>)

BrailleSurf (<http://www.snv.jussieu.fr/inova/bs4/uk/>)

Amplificadores de pantalla

Los **amplificadores 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.

Web The Screen Magnification Home Page (<http://www.magnifiers.org/>).

Reconocimiento de voz

Estos mecanismos permiten a los usuarios con alguna discapacidad física, lesiones en manos o antebrazos (crónicas o temporales) manejar el ordenador mediante ordenes de voz.

Dragon Naturally Speaking® (<http://www.astvtech.com/>)

IBM® ViaVoice® (<http://www-306.ibm.com/software/info1/voice/viavoice>)

Teclados alternativos y conmutadores

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

Software de rastreo

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

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 y a los controles de formulario de forma **secuencial**.

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.

Ventajas de la accesibilidad web

La accesibilidad web no sólo presenta ventajas para los usuarios con discapacidades físicas, psíquicas o sensoriales, sino que beneficia 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, lo que genera dificultades para, por ejemplo, entender ciertos contenidos o 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 posibilitan 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 algunos ejemplos.

Mejora del acceso en general.

Como hemos comentado, las mejoras de usabilidad, de navegación, de estructuración, etc. asociadas a la accesibilidad constituyen valores que benefician en sí mismos a todos los usuarios 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, la accesibilidad no obliga al usuario 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.



Referencia normativa

Una de las razones para la creciente difusión que está teniendo la accesibilidad web es la incorporación, en los últimos años, de este concepto en la redacción de directivas europeas y, en particular, en la legislación de los países de la Unión Europea.

Referencia europea

En el orden europeo, son varias las referencias a la accesibilidad web como elemento potenciador para la participación de las personas con discapacidad en la Sociedad de la Información. La sensibilización en estos temas ya se significó en la Comunicación e-Europe.

Algunas referencias importantes son éstas:

9 de diciembre de 1999. Comunicación e-Europe: Una sociedad de la información para todos.

Dentro de las diez áreas prioritarias de la Comunicación e-Europe, destaca la participación de las personas con discapacidad en la cultura electrónica. Para alcanzar ese objetivo, la UE instaba a los Estados miembros a la adopción de medidas relacionadas con la accesibilidad en unos plazos, y planteaba:

- La adopción de las normas WAI del W3C para sitios web públicos.
- Publicación de una norma de diseño para todos.
- Crear y conectar en red centros nacionales de excelencia y formular recomendaciones para un currículo europeo sobre accesibilidad.

Informes posteriores en relación con esta iniciativa concretaron:

- El establecimiento de un grupo interinstitucional para fomentar y garantizar la adopción, puesta en práctica y actualización regular -en función de la evolución de la Iniciativa de Accesibilidad a la Web del World Wide Web Consortium- de estas pautas en las instituciones europeas.
- Que tanto los estados miembros como la Comisión Europea exijan que los proyectos, actividades y organizaciones que reciban financiación pública en relación con el diseño de sitios web observen las directrices.
- Adopción de la recomendación del nivel de adecuación de accesibilidad WAI doble A para los portales de la Administración.

Listado de comunicaciones sobre e-Europe:

25 de septiembre de 2001. Comunicación de la Comisión al Consejo, al Parlamento Europeo, al Comité Económico y Social y al Comité de las Regiones. eEurope 2002: *Accesibilidad de los sitios Web públicos y de su contenido.*

24 de abril de 2002. 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.*

14 de enero de 2003. Resolución del Consejo sobre Accesibilidad electrónica – Mejorar el acceso de las personas con discapacidad a la sociedad el conocimiento.

Referencia española

En cuanto a legislación específica sobre accesibilidad en España, contamos con:

Esta normativa, en su punto séptimo, añade:

Orden PRE/1551/2003, de 10 de junio, por la que se desarrolla la disposición final primera del Real Decreto 209/2003, de 21 de febrero, por el que se regulan los registros y las notificaciones telemáticas, así como la utilización de medios telemáticos para la sustitución de la aportación de certificados por los ciudadanos.

“El registro telemático y el servicio de notificación telemática deberán cumplir los requerimientos en materia de accesibilidad establecidos por la Iniciativa para una Web Accesible (WAI) del Consorcio World Wide Web y en particular las especificaciones de la Recomendación de 5 de mayo de 1999 sobre Pautas de Accesibilidad del Contenido en la Web, versión 1.0, en su nivel AA”.

Sobre accesibilidad, la ley dice, en sus disposiciones adicionales:

LEY 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSICE).

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.”

LEY 51/2003, de 2 de diciembre, de Igualdad de Oportunidades, no Discriminación y Accesibilidad Universal de las Personas con Discapacidad (LIONDAU)

Contempla 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 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.”

Para la puesta en marcha de la LIONDAU, se desarrollaron dos planes: Plan Nacional de Accesibilidad 2004-2012 y el II Plan de Acción para las personas con discapacidad 2003-2007.

En el marco de estos planes, en el año 2003, a propuesta del Ministerio de Ciencia y Tecnología, el Consejo de Ministros creó:

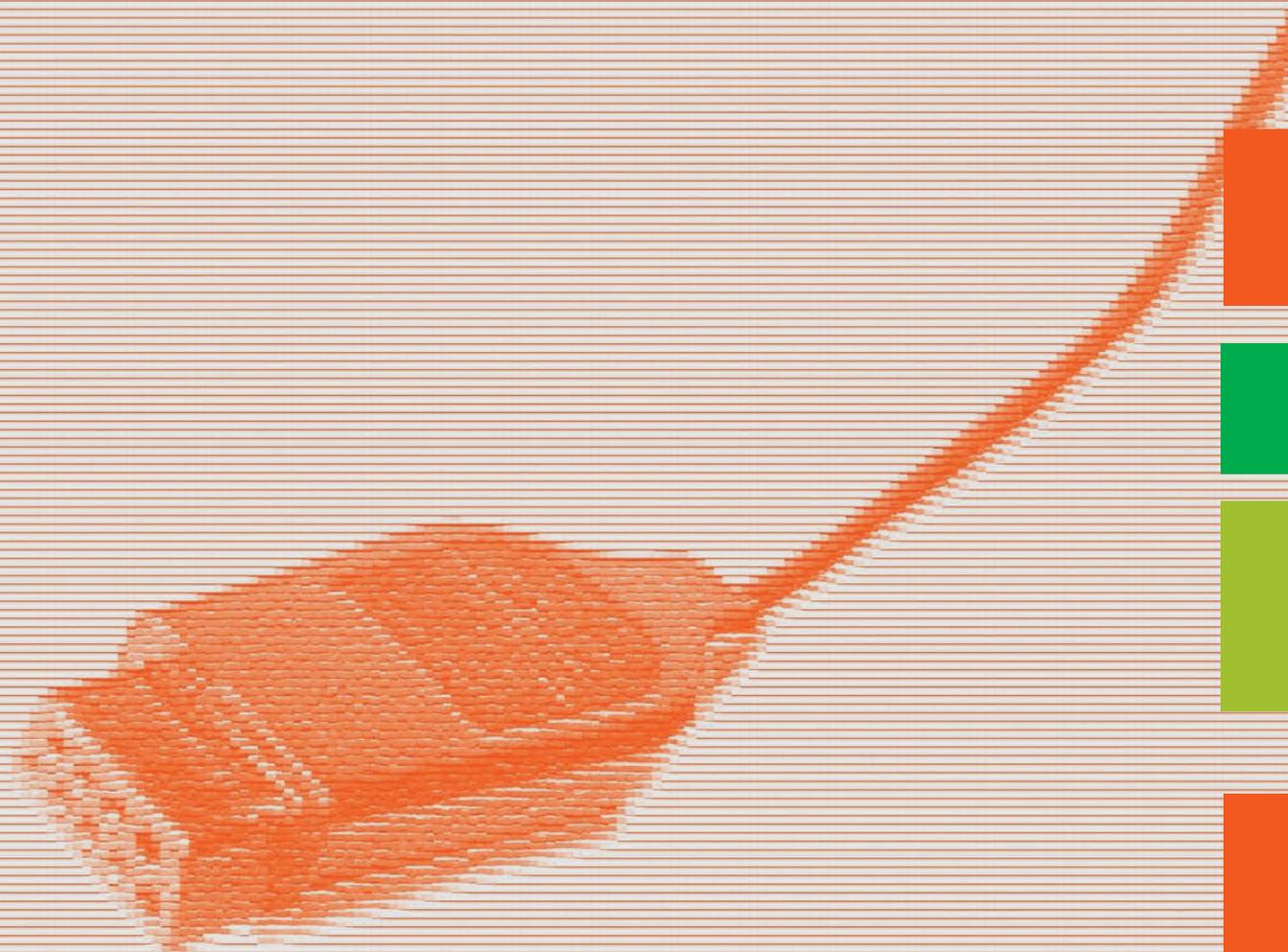
La Comisión Especial de Estudio para el Desarrollo de la Sociedad de la Información en España (CDSI).

La CDSI está formada por representantes del ámbito profesional, tecnológico, empresarial y académico, y recibió el mandato de analizar la implantación de la Sociedad de la Información, y proponer medidas para su desarrollo.

En abril de 2003, presentaron un informe denominado ***Aprovechar la Oportunidad de la Sociedad de la Información en España: Recomendaciones de la Comisión Especial de Estudio para el Desarrollo de la Sociedad de la Información***, en el que se definían las líneas a seguir en política de accesibilidad web en España.

Sus principales conclusiones en relación con la accesibilidad web son fomentar la accesibilidad de los discapacitados a sitios web, diseñando la estructura y contenidos de las páginas teniendo en cuenta las necesidades específicas de los distintos colectivos (personas con dificultades visuales, motrices, etc.).

Accesibilidad web



Organización de la guía

Esta guía técnica se basa en el trabajo del WAI-W3C, principalmente en el Currículo de WCAG 1.0.

La presente guía esta basada en las WCAG 1.0 (Pautas de Accesibilidad al Contenido en la Web). Reorganiza sus puntos de verificación, lo que proporciona una orientación más didáctica y clara.

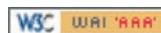
Las diferentes categorías en las que se han dividido los puntos de verificación de la guía son las siguientes:

- 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 el tema 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 los dos temas. Se pretende que no haya que consultar otro tema en busca de puntos de verificación aplicables a los conceptos explicados en el tema actual.

Se han ordenado los puntos de verificación por **conceptos** y, dentro de cada concepto, por **prioridades**. Por ejemplo, en el tema de formularios están listados, ordenados por prioridad, todos los puntos de verificación aplicables, directa o indirectamente, a los formularios. Estas prioridades están asociadas al nivel de conformidad que se obtiene al cumplirlas:

- Nivel de conformidad 'A'. 
- Nivel de conformidad doble 'A'. 
- Nivel de conformidad triple 'A'. 

Con el enunciado de cada punto a tratar incluimos un grupo de dígitos entre corchetes (**Equivalente textual [1.1]**). Estos números representan el punto de verificación de las pautas de accesibilidad al contenido web del W3C (en el apéndice se pueden ver todas).

A lo largo de la guía, aparecen resaltadas informaciones con buenas prácticas según los tipos:

Importante

Información relevante que se tiene que tener siempre en cuenta.

Recordatorios

Información que debe recordarse para conseguir sitios web accesibles.

Ejemplos

Códigos de ejemplos con la formulación correcta.

Ejemplos desaconsejados

Ejemplos de malas prácticas que se suelen ver de forma habitual.

Para cualquier puntualización, han de tenerse en cuenta las pautas de accesibilidad web que figuran en el apéndice final de esta guía.

Imágenes y Multimedia

Imágenes

Equivalente textual [1.1]

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

Importante: El **texto alternativo** indica a los dispositivos de usuario 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.

A continuación te mostramos diferentes casos que te puedes encontrar a la hora de incluir imágenes en la web.

Imágenes sencillas

Importante: Muchas de las imágenes que se incluyen en las páginas web son imágenes sencillas que no requieren de una descripción demasiado extensa. Para este tipo de imágenes proporciona una **breve descripción** mediante el atributo "alt".

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, etc.). Aunque no sea necesario un texto alternativo, como el uso del atributo "alt" es obligatorio, entonces se debe poner un texto alternativo vacío.

```

```

De cualquier manera la mejor forma de incluir imágenes decorativas es desde las hojas de estilo. En posteriores capítulos se amplía información a este respecto.

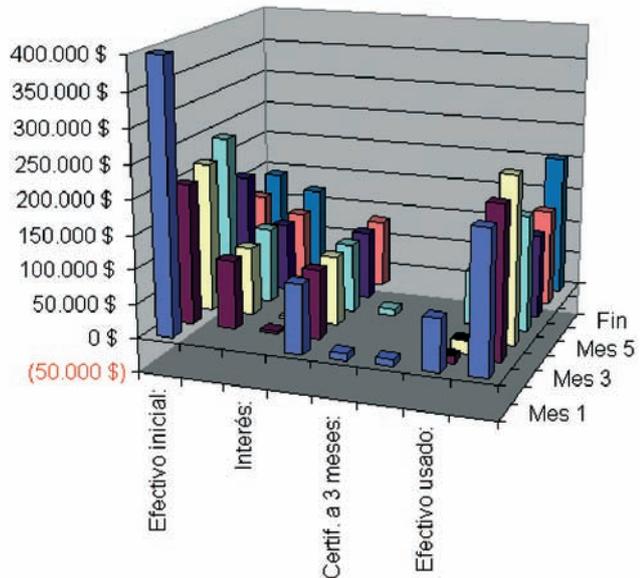
Imágenes complejas

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

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

```

```



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").

Importante: 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.

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.

```

```



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.

Imágenes con animaciones

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

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".

```
<a href="cap2.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).

```
<a href="cap2.html">  
Anterior</a>
```



Anterior



ASCII Art [13.10]

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 de lectores de pantalla, ya que los leerán sin ningún sentido. Por tanto, conviene que proporcionen un texto descriptivo y una manera de saltar los ASCII art.

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.

Mapas de imagen



Equivalentes textuales en mapas de imagen de cliente [1.1]

De ese modo, un lector de pantalla puede acceder a la información sin ninguna dificultad.

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

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**.

```

  <map name="map1">
    <area shape="rect" coords="0,0,120,75" href="#"
alt="Orientate">
    <area shape="rect" coords="121,0,240,75" href="#"
alt="UAW">
    <area shape="rect" coords="0,76,120,150" href="#"
alt="Formación">
    <area shape="rect" coords="121,76,240,150" href="#"
alt="W3C">
  </map>
```





Vínculos redundantes en mapas de imagen de servidor [1.2]

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

Como todo el proceso se realiza del lado del servidor, el navegador no aporta información al usuario 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.

Importante: 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**.

```
<a href="../../../cgi-bin/imagemap/mapa"> </a>
<p>
  [ <a href="#">Orientate</a>]
  [ <a href="#">UAW</a>]
  [ <a href="#">Formación</a>]
  [ <a href="#">W3C</a>]
</p>
```



[Orientate] [UAW] [Formación] [W3C]

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".



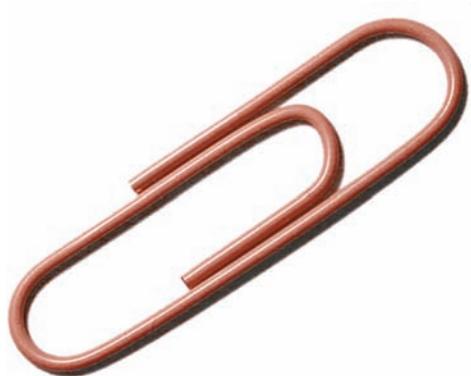
Mapas de imagen de cliente mejor que de servidor [9.1]

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.



Vínculos redundantes en mapas de imagen de cliente [1.5]

Para llegar a este nivel de adecuación es preciso que los mapas de imágenes tengan los vínculos redundantes, es decir, que no sólo se pueda acceder a estos desde la imagen.



Multimedia



Sincronización de alternativas accesibles (subtítulos) [1.4]

Para las personas con discapacidad auditiva, que pueden ver la información visual, es necesario proporcionar subtítulos a los vídeos. Los subtítulos son transcripciones del audio de la película e incluyen diálogos y sonidos más importantes, de forma sincronizada con el vídeo. Los subtítulos pueden estar incrustados en el propio vídeo, es decir, formar parte de la propia imagen, o bien estar en un archivo aparte. En el primer caso, se pueden proporcionar dos versiones del vídeo: una con subtítulos y otra, sin ellos. En el segundo caso se pueden emplear diferentes tecnologías (como SMIL o SAMI) para sincronizar distintos archivos en una presentación multimedia, por ejemplo, una pista de vídeo con uno o varios archivos de texto aparte con los subtítulos.

- *SAMI (Microsoft Synchronized Accessible Media Interchange).*
- *SMIL (Synchronized Multimedia Integration Language).*



Descripciones en audio para personas con discapacidades visuales [1.3]

Es necesario incluir descripciones auditivas de la parte visual, es decir, del contenido de la imagen. Pero hay que prestar especial atención a que esa información en audio no interfiera con el audio propio del vídeo (diálogos, sonidos más importantes, etc.). La descripción auditiva debería incluir los elementos visuales clave, las acciones, los escenarios, lenguaje corporal, gráficos y texto mostrado. Es similar a la voz de un narrador que va explicando de palabra lo que ocurre en la imagen, sin interferir con el audio original, aprovechando los silencios.



Transcripciones textuales completas [1.1]

Por si no se tiene acceso al archivo multimedia (audio y vídeo) por las circunstancias que sean (navegador de texto, ausencia de reproductor o plugin adecuado, etc.), también es necesario incluir una transcripción textual completa del archivo multimedia. Es decir, una transcripción en formato de texto accesible en la que se incluyan tanto las descripciones del vídeo como del audio. Estas transcripciones permiten el acceso a todos los usuarios, a la vez que permiten la indexación de los contenidos y de la información presente en la presentación multimedia.

Objetos programados

Equivalentes accesibles



Desactivación de objetos programados [1.1; 6.3]

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

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

Importante: Como el texto se considera accesible para casi todos los usuarios, se debe proporcionar un texto equivalente para todos los elementos no 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: 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:

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:

```
<object classid="java:nombreadaplet.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:

```
<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 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.

```
<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>
```

Actualización de los equivalentes del contenido dinámico [6.2]

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

Importante: 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.

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

```
<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.



Contenido dinámico

No provocar destellos en la pantalla [7.1]

Los desarrolladores deben **evitar los destellos** en la pantalla hasta que las aplicaciones de usuario 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).

Contenido dinámico accesible [6.5]

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 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>
```

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:

```
<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.mtas.es/uafse"
onclick="return abrirURL(this.href);" title="El enlace se abre en
nueva ventana: Página de la UAFSE "> UAFSE</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".

Evitar el parpadeo del contenido [7.2]

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

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 elimine el parpadeo desactivando la CSS.

Movimiento del contenido [7.3]

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 desactivándolo en el navegador.

Independencia de dispositivo

Independencia de dispositivo de entrada [6.4]

La **independencia de dispositivo** garantiza que los usuarios 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.

Importante: 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]

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

Objetos programados directamente accesibles [8.1]

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

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.

Manejadores de eventos lógicos en los scripts [9.3]

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

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"**.

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.

Marcos

Equivalentes textuales

Contenido alternativo para marcos [1.1]

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.

En el siguiente código, puedes ver un 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">
  <noframes>
<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.



Títulos para los marcos [1.1 y 12.1]

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

Un ejemplo de una estructura con marcos podría ser éste:

```
<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]

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.



Descripción de la relación entre los marcos [12.2]

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.

```
<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>
```

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.



Tablas

Tablas de datos

Identificar encabezamientos [5.1]

Cuando presentes datos en forma tabular identifica los encabezamientos de fila y columna mediante el elemento TH.

Tablas de datos complejas [5.2]

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:

- 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.
- **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.

A continuación, profundizaremos en los elementos enumerados.

Atributos "id", "headers" y "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 proporciona 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.

Por otra parte, el atributo "**scope**", definido para encabezamientos, se puede utilizar 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.

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.

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.

A continuación te proponemos un ejemplo para mostrar cómo se relacionan las celdas de datos y de encabezamiento mediante "id" y "headers", así como el empleo de algunos elementos de tablas explicados.

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			

```

<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>

```

Si quieres profundizar en el uso de las tablas, sus elementos y atributos puedes revisar la sección sobre tablas del W3C (<http://www.w3.org/TR/REC-html40/struct/tables.html>).

Proporcionar resúmenes de las tablas [5.5]

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.

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:

```
<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]

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:

```
<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>
```

Tablas de maquetación

Evita la utilización de tablas para maquetar [5.3]

Cuando las aplicaciones de usuario 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.

Importante: 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.

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.

No usar marcadores estructurales para dar formato [5.4]

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

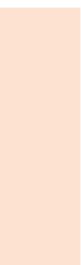
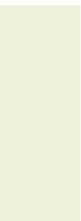
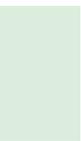
Por ejemplo, evita recurrir a la utilización del elemento TH para conseguir que el texto quede centrado y en negrita cuando en realidad no se trata de un encabezado de fila o columna de datos. En su lugar, emplea las hojas de estilo como vemos en el siguiente código:

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



Texto lineal alternativo para las tablas [10.3]

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.



Formularios

JavaScript y formularios



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.).

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 JavaScript es una funcionalidad segura.

Importante: 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.



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.

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.



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 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.



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.

Asociación entre etiquetas y controles



Asociación explícita [12.4]

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.

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.

Importante: 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

Elevalunas eléctrico

Techo Solar

```
<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">Elevalunas eléctrico</label><br>
    <input id="opt3" type="checkbox" value="ts">
    <label for="opt3">Techo Solar</label>
  </fieldset>
</div>
</form>
```

```
</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]

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.

Importante: 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.

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:

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



Independencia de dispositivo

Tabindex [9.4]

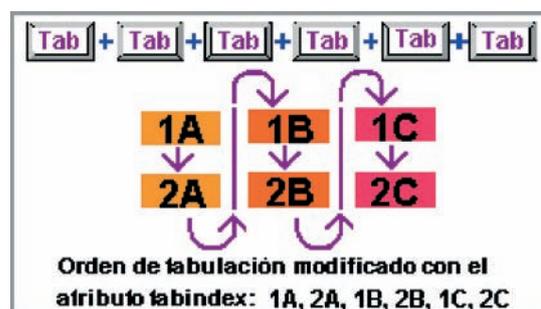
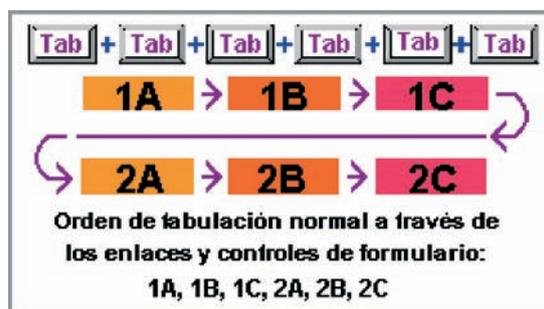
Uno de los principios básicos de la accesibilidad es el diseño independiente de dispositivo, por ejemplo, permitiendo la navegación por teclado y no sólo con el ratón.

Importante: El acceso mediante teclado es el principal medio de interacción de las páginas accesibles, ya que una gran parte de las ayudas de usuario se basa en el uso del teclado o emula 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.

Accesskey [9.5]

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.

Para resaltar cuál es la letra que funciona como atajo de teclado puedes utilizar un subrayado o, si se trata del texto de 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.

Otros aspectos sobre accesibilidad



Agrupar la información de los formularios [12.3]

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:

```
<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>
    </optgroup>
  </select>
</div>
```

```

        <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:

```

<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>
<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>

```



Caracteres por defecto en controles vacíos [10.4]

Algunos navegadores antiguos no permiten al usuario 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.



Estándares

Gramáticas formales



Validar los documentos contra gramáticas formales [3.2]

Procura escribir el código siguiendo una gramática formal, identificar el tipo de documento mediante **!DOCTYPE** y validarlo con alguna de las herramientas disponibles proporcionadas por el W3C.

Una de estas herramientas de validación gramatical es el **W3C HTML Validation Service**. El validador HTML del W3C (<http://validator.w3.org/>) 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).

Además, existe un validador de CSS, **W3C CSS Validator**. Herramienta gratuita para validar las hojas de estilo CSS (<http://jigsaw.w3.org/css-validator/>) solas o presentes en documentos (X)HTML .

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**.

Importante: 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.

```
<!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.

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

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

```
<!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.

```
<!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.

```
<!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.

```
<!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, **te recomendamos** que uses gramáticas formales modernas como **XHTML 1.0** y **HTML 4.01**, 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".



Tecnologías W3C

Utilizar tecnologías W3C [11.1]

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

Importante: 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, 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.**
- Aseguran 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.

Importante: 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".

Elementos desaconsejados



No usar características desaconsejadas [11.2]

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.

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 (<http://www.w3.org/TR/REC-html40/index/elements.html>) y la lista de atributos (<http://www.w3.org/TR/REC-html40/index/attributes.html>) de HTML para saber cuáles están desaconsejados.

Página alternativa accesible



Proporcionar alternativas accesibles [11.4]

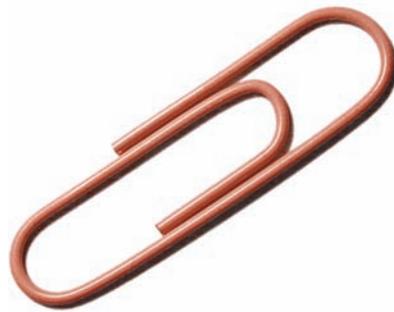
En alguna ocasión, es imposible lograr que toda la información de una página sea accesible. En estos casos, puedes usar una **página alternativa** cuyo contenido sí sea accesible.

Para solventar el problema, proporciona **enlaces** al inicio de la página original y de la alternativa, que permitan a los usuarios 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.

Importante: 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.



Estructura y Presentación

Hojas de estilo



Lectura sin hojas de estilo [6.1]

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**.

Importante: 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.

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

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.

Utilizar hojas de estilo [3.3]

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.

Importante: 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, etc. 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 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, por elección o necesidad, requieren cambiar la presentación de las páginas. Si usas las hojas de estilo facilitas a los usuarios 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, PDA, 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.

Con todo, 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. 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 desaprueban 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 el tema 5 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.

```
<FONTsize="+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.*

```
<style type="text/css">
  .para { font-size: 130%; color: Green;}
  .lineado { font-size: 70% ; font-style: italic ; color: Navy }
</style>
...
<p class="para">Esto está en verde y con letra grande, <span
class="lineado">
  y esto en azul y es más pequeño.</span>
</p>
```

Debes usar unidades relativas en vez de unidades absolutas tanto en las 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 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:

```
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:

```
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.

Importante: 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. 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 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 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).

Importante: 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 longitudes relativas y absolutas también existen unos **tipos especiales** o **palabras clave** para definir el tamaño de la fuente.

Estas palabras clave absolutas son (xx-small, x-small, small, médium, large, x-large y xx-large) y relativas (larger y smaller).

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.

```
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, etc.) y marcos (FRAMESET, IFRAME), utiliza unidades de medida relativas en lugar de absolutas.

Las medidas relativas que puedes utilizar en HTML son:

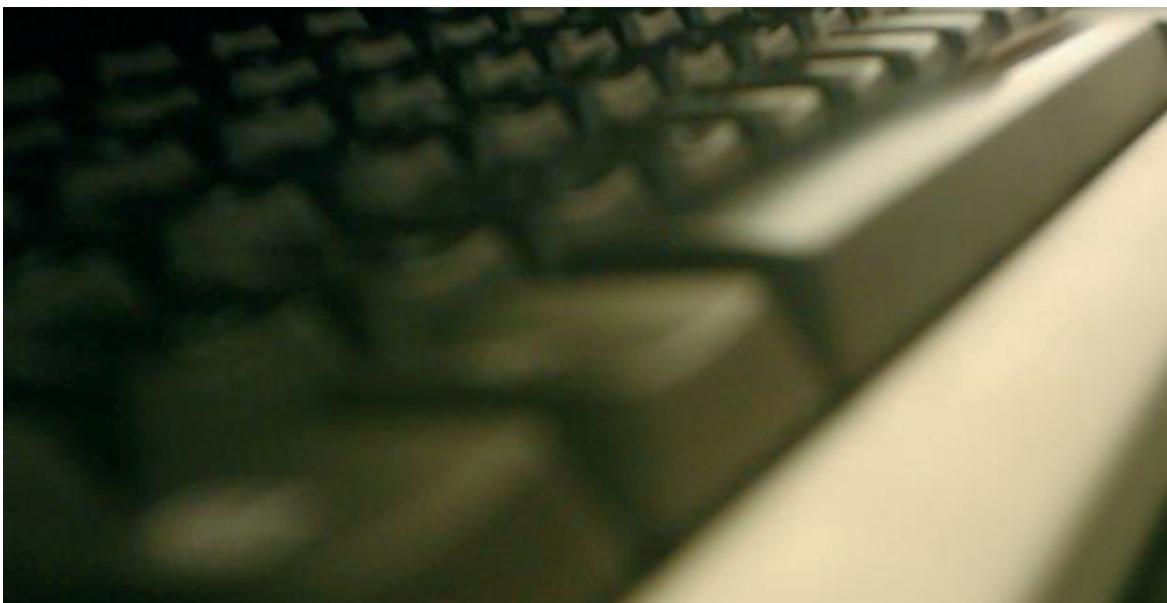
Porcentajes

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.



Color y contraste

No transmitir información solo por el color [2.1]

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 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.

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

Usar un contraste suficiente [2.2]

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 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).



Encabezamientos

Transmitir la estructura lógica [3.5]

Importante: Todas las páginas deben tener al menos un elemento de encabezado H1.

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.

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 u otro H1 o un H2, pero no un H3. Sin embargo, puedes bajar 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:

```
<h1>Música</h1>
  <h2>Tipos de música</h2>
    <h3>Rock</h3>
    <h3>Clásica</h3>
```

Listas

Imágenes como viñetas [1.1]

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:

```
ul {
  list-style-image: url(path_imagen);
}
```

En él, se especifica que la imagen indicada en el path se utiliza como viñeta.

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:

```
<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).

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...

```
<p>
  Noticias (imágenes usadas como viñetas):
</p>
<div>
  En la última
reunión...
</div>
<div>
  Según el informe...
</div>
<div>
  Inaugurado el...
</div>
```

Se puede conseguir el mismo efecto con CSS:

```
<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.



Marcar correctamente las listas [3.6]

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

1. 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.

La numeración compuesta mediante hojas de estilo sólo 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.

```
<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 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, entonces se mostrará el final de la lista.

Suministros de oficina:

- Papelería:
 - Sobres
 - Blocks 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*)

```

<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:

```
<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 en el ejemplo de “suministros de oficina”.

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 sólo 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.

Citas

Marcar correctamente las citas [3.7]

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**

```
<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>¡demonios!</q> </p>
```

...y al caérsele el piano encima del pie, salió de su boca un aplastante ¡demonios!

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:

```
div.nombre_clase{padding-left: 2em;padding-right: 2em;}
...
<div class="nombre_clase">Texto al que se quiere dar formato</div>
```

Dividir la información

Dividir bloques largos de información [12.3]

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.



Marcadores mejor que imágenes

Usar marcadores en vez de imágenes [3.1]

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 y se inserta como una imagen en la página web.

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.



Navegación, lenguaje y comprensión

Lenguaje y comprensión

Identificar cambios en el idioma [4.1]

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 que usan traductores a braille.

Para marcar un cambio de idioma en un bloque de texto (P, DIV, BLOCKQUOTE, etc.) utiliza el atributo "lang".

```
<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."

```
<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 o bien uno de los caracteres especiales reservados como "i" o "x".

A continuación te mostramos algunos ejemplos de códigos de nombres de idioma:

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 (<http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html>) 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-chokee).

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 (<http://www.w3.org/TR/REC-html40/references.html#ref-RFC1766>).



Usar un lenguaje claro y sencillo [14.1]

A la hora de redactar el contenido procura que sea comprensible con un lenguaje claro y sencillo.

Un lenguaje con esas características beneficia a los usuarios con discapacidades cognitivas, de lectura o aprendizaje, o a aquellos 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 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. Asimismo, evita el uso de verbos en su forma pasiva.

Identificar el idioma principal del documento [4.3]

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 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.

```
<html lang="es">
```

En **XHTML 1.0** utiliza "**xml:lang**" combinado con "**lang**".

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="es" xml:lang="es">
```

En **XHTML 1.1** utiliza "**xml:lang**"

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
```

Con encabezados **HTTP** emplea un elemento **META**.

```
<meta http-equiv="content-language" content="es" />
```

Especificar la expansión de abreviaturas y acrónimos [4.2]

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)
- Fdo. (firmado)
- S.S.M.M. (Sus Majestades)

Los **acrónimos** son vocablos formados por la unión de las letras de dos o más palabras. También se denomina acrónimo a las formas abreviadas que se pronuncian como una palabra.

Algunos ejemplos de acrónimos son:

- Módem: modulación demodulación.
- Sonar: Sound Navigation And Ranging.
- Ovni: Objeto Volante No Identificado.

Importante: 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 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.

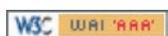
La identificación de las abreviaturas y acrónimos se realiza con **ABBR** y **ACRONYM** respectivamente y para especificar la expansión se usa el atributo **"title"**.

```
La Fundación <acronym title="Centro Tecnológico de la Información
y la 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.
```



Situar la información importante al inicio [13.8]

Si quieres ayudar a los usuarios 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"**.



Complementar el texto [14.2]

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.

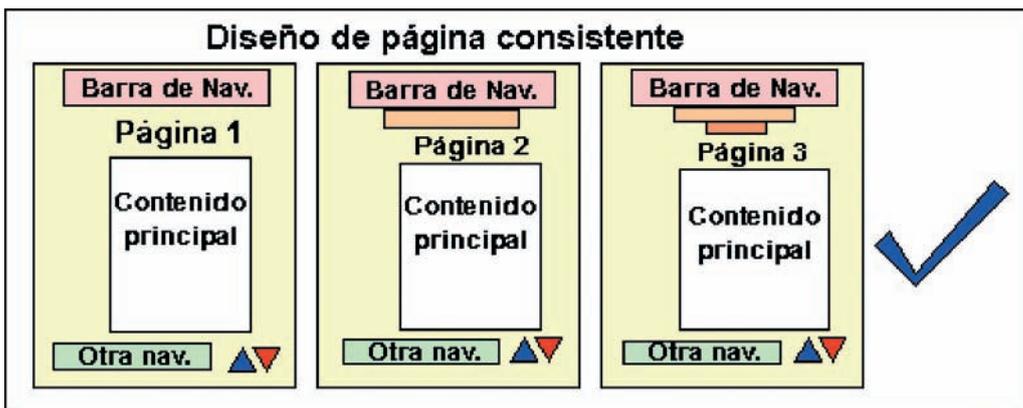
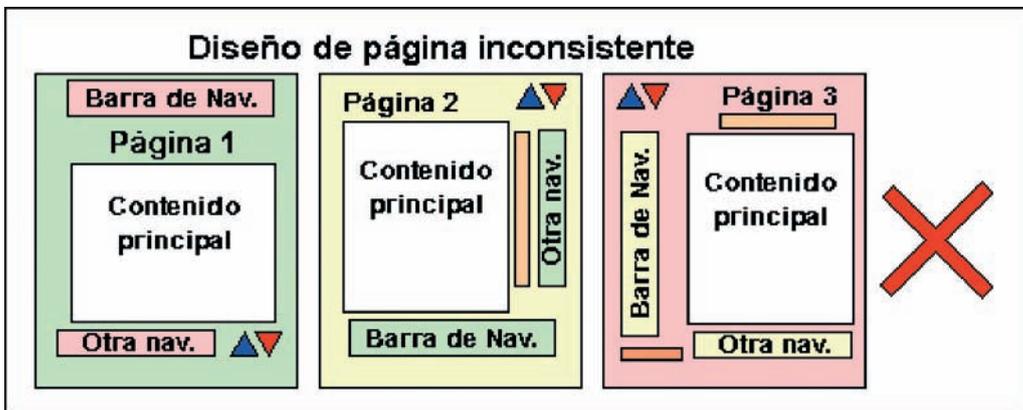
De todas formas, evita el uso indiscriminado de elementos multimedia. Utiliza estos recursos en su justa medida e inclúyelos sólo si mejoran y facilitan la comprensión de los contenidos. Si no aportan nada al texto, su uso puede ser perjudicial.

 **Usar un estilo de presentación coherente [14.3]**

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 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.



Actualizaciones automáticas y redireccionamiento

Evitar las actualizaciones automáticas [7.4]

Las actualizaciones automáticas de una página pueden **desorientar** a algunos usuarios. No podemos prever cuánto tiempo puede tardar un usuario en revisar el contenido de una página.

Si es necesario actualizar el contenido de la página cada cierto tiempo, informa al usuario para que lo haga por sí 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]

El usuario 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 y pueden corromper el historial de páginas visitadas del navegador.

Enlaces

Identificar claramente el destino de los enlaces [13.1]

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 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)

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.

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, diferencia unos enlaces de otros por medio del atributo "title".



Avisar de la apertura de nuevas ventanas [10.1]

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 (independientemente de sus características) pueden sentirse desorientados.
- 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, que el botón "atrás" se encuentre inactivo es la única pista que tienen los usuarios 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 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 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:

```
<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>
```

Incluir caracteres entre enlaces [10.5]

Los caracteres imprimibles entre los enlaces posibilitan que los lectores de pantalla hagan una pausa, permitiendo que los usuarios 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](#)]

```
[ <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 omites 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.



```
<a href="a.htm"></a>
<a href="b.htm"></a>
<a href="c.htm"></a>
<a href="d.htm"></a>
```

Es una práctica general situar los enlaces de navegación al inicio de la página. Aquellos usuarios que lean el contenido de forma secuencial, como los usuarios 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 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:

```
<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 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 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 de páginas web los que deben proporcionar esos mecanismos como se ha explicado antes.

Mecanismos de navegación



Proporcionar información general del sitio (mapa del sitio) [13.3]

El uso de **mecanismos de navegación** claros e intuitivos mejora la accesibilidad y la usabilidad.

Estos recursos benefician a usuarios 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, etc).

Los usuarios 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 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.

- [El F.S.E en Español](#)
 - [La UAFSE](#)
 - [¿Qué es la UAFSE?](#)
 - [Funciones de la UAFSE](#)
- [Planes](#)
- [Iniciativas Comunitarias](#)

```

<ul>
  <li>
    <a href="http://www.mtas.es/uafse" title="Nivel 1">El F.S.E en
    España</a>
    <ul>
      <li>
        <a href="http://www.mtas.es/uafse/uafse" title="Nivel 2">La UAF
        SE</a>
        <ul>
          <li>
            <a href="http://www.mtas.es/uafse/uafse/quees"
            title="Nivel 3">
              ¿Qué es la UAFSE?</a>
            </li>
            <li>
              <a href=" http://www.mtas.es/uafse/uafse/funciones"
              title="Nivel 3">
                Funciones de la UAFSE</a>
              </li>
            </ul>
          </li>
          <li>
            <a href="http://www.mtas.es/uafse/planes" title="Nivel 2">
            Planes</a>
          </li>
          <li>
            <a href=" http://www.mtas.es/uafse/iniciativas" title="Nivel 2">
            Iniciativas Comunitarias</a>
          </li>
        </ul>
      </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]

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 con problemas de lectura y aprendizaje o que usen lectores de pantalla al facilitar la localización de la información en los sitios web.

Proporcionar barras de navegación [13.5]

Las barras de navegación facilitan la **localización de la información** por parte del usuario. Cuando éste se ha familiarizado con el estilo y el funcionamiento de la barra de navegación, navegará 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).

En la siguiente imagen se puede ver la barra de navegación empleada en Amazon.



Proporcionar funciones de búsqueda de varios niveles [13.7]

La mayoría de los buscadores solicitan al usuario que introduzca palabras clave para buscar recursos. Los usuarios 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 seleccionando opciones para permitir 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**.*

Metadatos

Utilizar metadatos para añadir información [13.2]

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, robots, indexadores, etc.).

```
<head>
  <title>UAFSE. 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="Unidad Administradora Fondo Social Europeo. ">
  <meta name="Keywords" content="UAFSE, Eual, Eropa, DGES, TA y FSE, ...">
  <meta name="Author" content="UAFSE. Desarrollo">
  <meta name="Copyright" content="UAFSE. 2004">

  <link rel="alternate" type="application/rss+xml" title="RSS" href=" http://www.mtas.es/uafse/uafse.rss">
<link rel="meta" href="uafse_about.rdf">
<link rel="stylesheet" type="text/css" href=" css/estilo_uase.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.

```
<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 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 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 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**.

Importante: 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.

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 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.

Elemento META (metadatos)

El elemento **META** identifica propiedades de un documento (por ejemplo, el autor, copyright, una lista de palabras clave, etc.) 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, etc. Las propiedades se indican con los caracteres DC seguidos de un punto y el nombre de la propiedad.

A continuación, te mostramos algunos ejemplos:

```
<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 (<http://dublincore.org/>) 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 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).



Proporcionar documentos alternativos según preferencias [11.3]

La forma más sencilla para que los usuarios 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. 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).

```
<a href="http://www.w3.org/WAI/" type="text/html" hreflang="en">Página del WAI</a>
```

Si deseas más información puedes leer el FAQ sobre negociación de lenguaje del W3C (<http://www.w3.org/International/questions/qa-when-lang-neg>).

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.

Se puede indicar a los navegadores y buscadores versiones alternativas del documento original. Para ello, define el tipo de vínculo (atributo "rel" del elemento LINK) como

"**alternate**" para indicar que se trata 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.

```
<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 está 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.

```
<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 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"**.

```
<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**.

De igual forma, también se puede usar sobre el elemento **STYLE** en las hojas de estilo internas.

```
@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}}
```

```
<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.

Independencia de dispositivo



Proporcionar un orden lógico de tabulación [9.4]

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

El orden de tabulación establecido por defecto empieza en el primer enlace del formulario del documento y sigue por los siguientes enlaces 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.

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, etc).

```
<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.

Importante: 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]

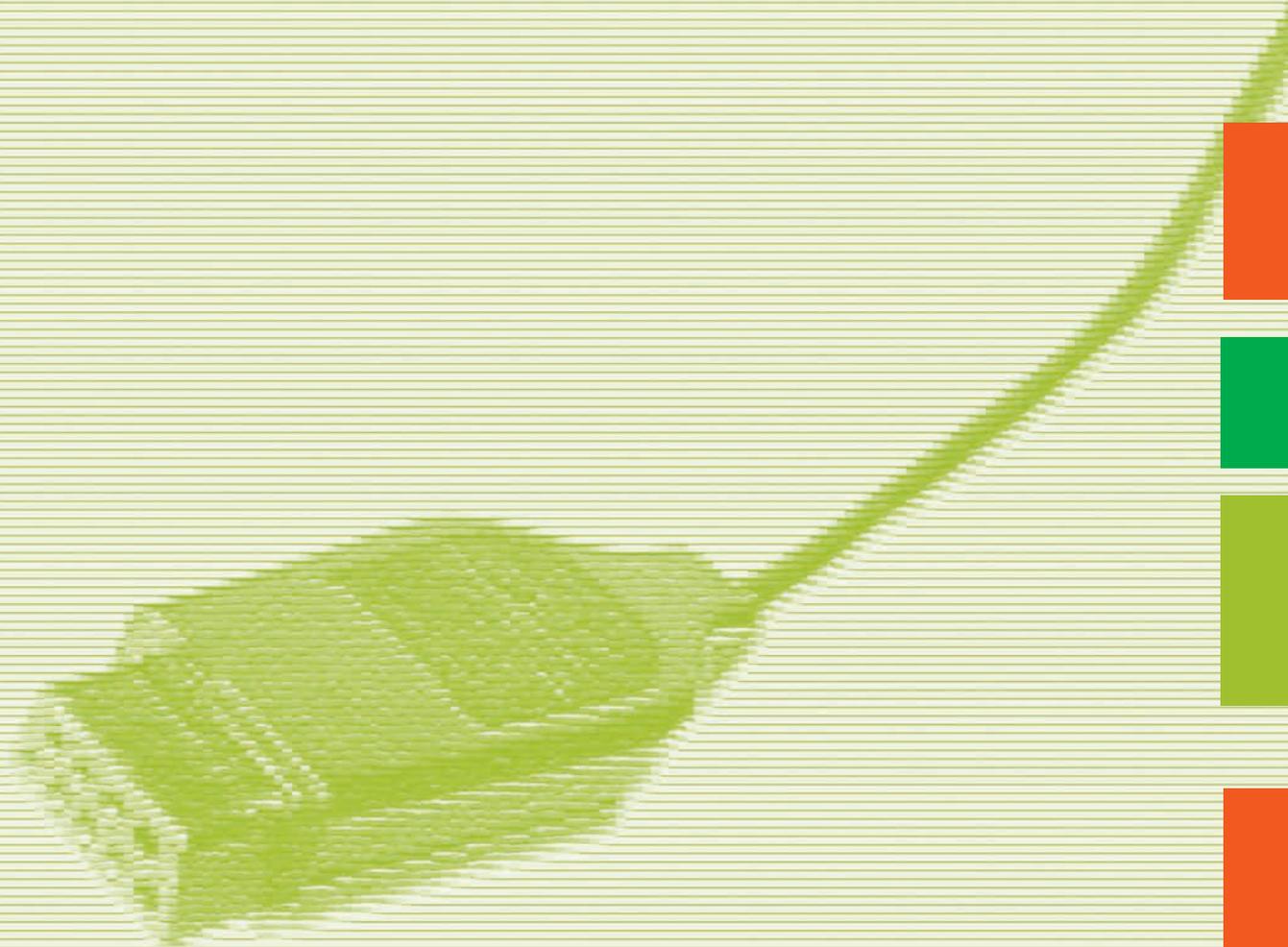
HTML 4.01 permite a los desarrolladores de contenidos especificar **atajos de teclado** en sus documentos para los enlaces y controles de formulario mediante el atributo "**accesskey**" (que funciona con los elementos A, AREA, BUTTON, INPUT, LABEL, LEGEND, TEXTAREA).

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 los actuales sí.



Evaluación de accesibilidad





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, desarrolladas por la Iniciativa de Accesibilidad Web del W3C (vistas en la presente guía y de aplicación en el contexto europeo).

Evaluación de la accesibilidad

Importante: 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 a la hora de la validación, haciendo imprescindible la participación de un experto.**

El WAI-W3C mantiene una lista de herramientas de Evaluación, Reparación y Transformación para la Accesibilidad del Contenido Web. (<http://www.w3.org/WAI/ER/existing-tools.html>). Por la importancia de la herramienta TAW, que es la universalmente aceptada en España para la validación de sitios web, se detalla el funcionamiento de ésta.

TAW: Test de Accesibilidad Web (<http://www.tawdis.net>)

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.

t.a.w³
test accesibilidad web
TAW 3.0 (3/7/06 9:28 AM) Validación utilizando normas WAI 5 de mayo de 1999
Resultado de la verificación: <http://www.telefonica.net>

Resultados del análisis

	Automático	Manual
Prioridad 1	95	158
Prioridad 2	142	203
Prioridad 3	22	56

Telefónica Net[®] España, martes 7 de marzo de 2006

Acceso telefónico a Internet y Línea ADSL | Soluciones ADSL

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 resumen:** 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.

-  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 (por ejemplo, la necesidad del uso de "longdesc" en las imágenes).

En el **resumen**, el TAW muestra:

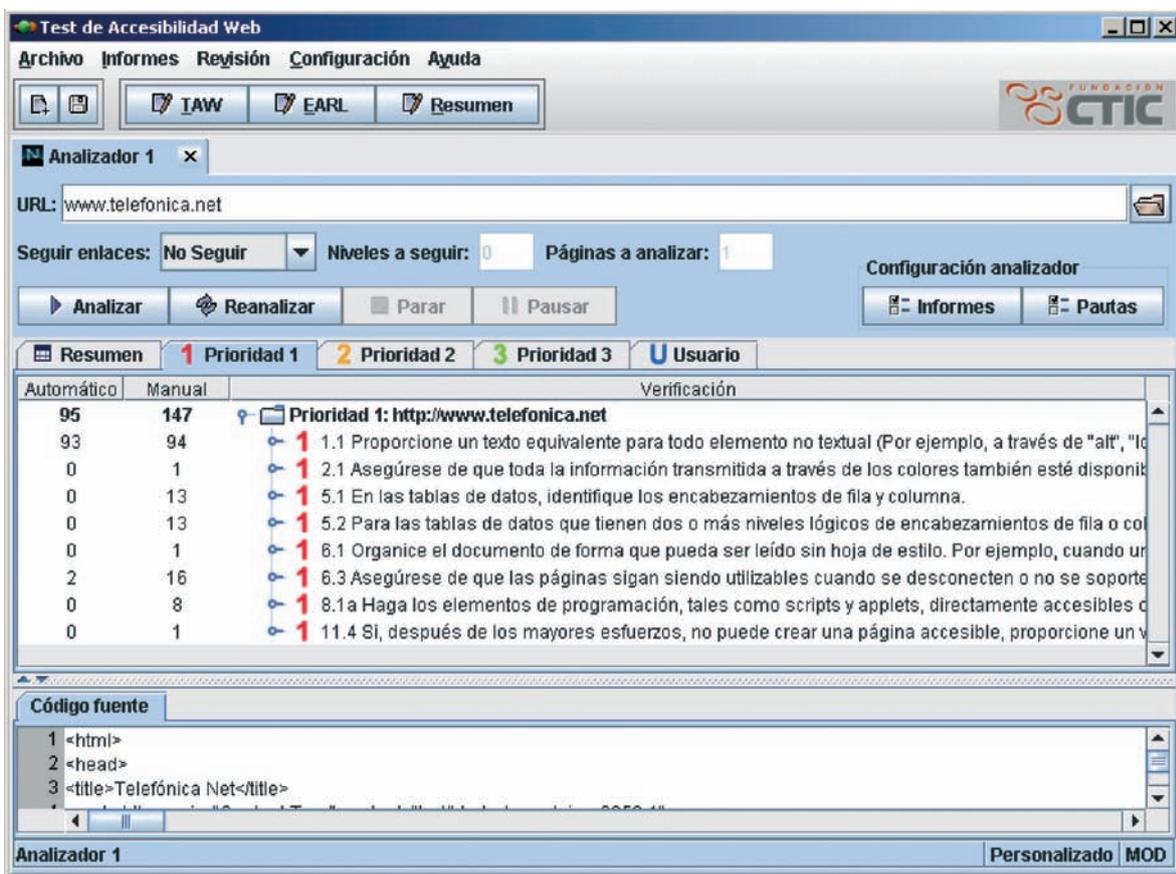
- Un listado con los **problemas de accesibilidad** encontrados (ordenados por prioridad) y su descripción.
- El **número de línea** de la página analizada.
- **La etiqueta HTML** que genera el problema de accesibilidad.
- El criterio de la versión online de TAW para considerar que una página es **TAW-válida** 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 el 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.

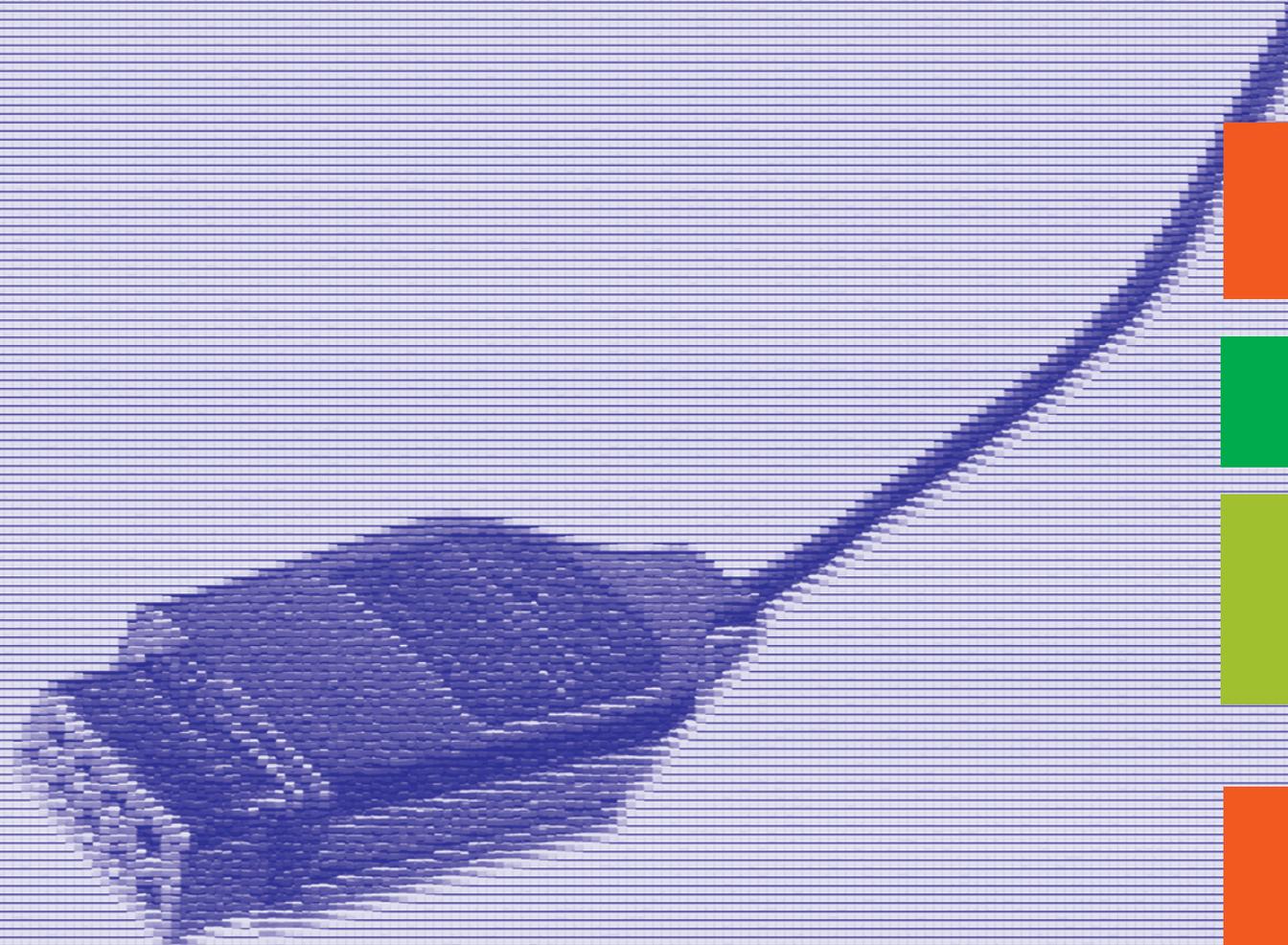
Es multiplataforma y dispone de instaladores para varios sistemas operativos: Windows®, Mac OS®, Unix y familiares (Linux, Solaris®, etc.)

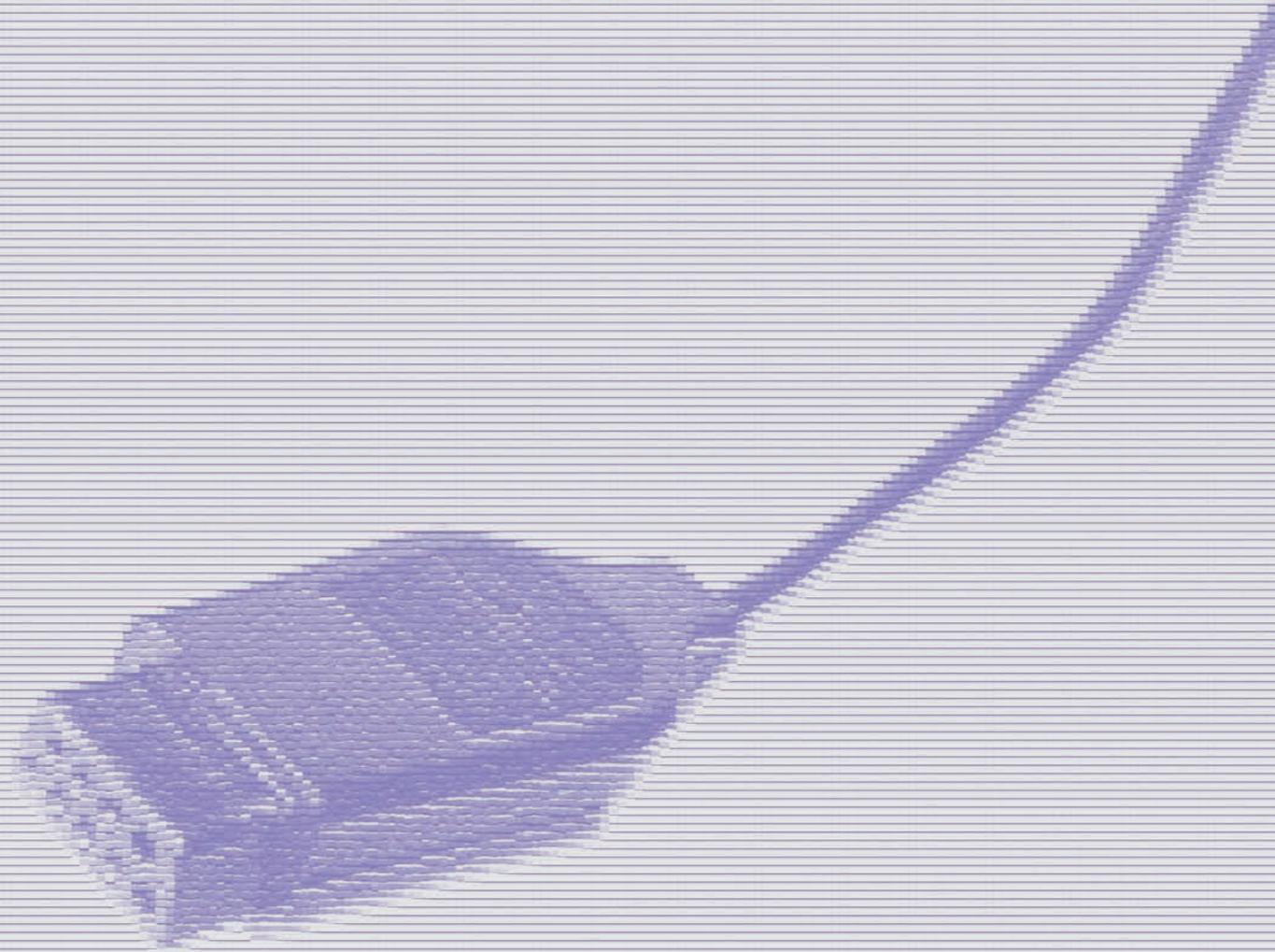
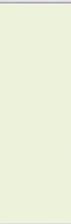
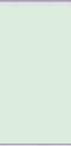


Existe un icono que se puede añadir a las páginas web para indicar que ha superado el test de accesibilidad web del TAW, similar a los iconos del W3C de conformidad con el código de las gramáticas formales.



Apéndice





Pautas de Accesibilidad al Contenido en la Web 1.0

Recomendación W3C de 5 de mayo de 1999

La única versión normativa de este documento es la versión original en inglés que se encuentra en el sitio web del W3C <http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505>.

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

Esta versión puede contener errores debidos a la traducción. Para comunicar errores de esta traducción o proponer comentarios a la misma, envíe un mensaje a los traductores.

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

Traductores: Carlos Egea, Alicia Sarabia, Alan Chuter

Pauta 1 - "Proporcione alternativas equivalentes para el contenido visual y auditivo".

Proporcione un contenido que, presentado al usuario, cumpla esencialmente la misma función o propósito que el contenido visual o auditivo.

Si bien algunas personas no pueden utilizar imágenes, películas, sonidos, applets, etc, directamente, sí pueden utilizar páginas que incluyen información equivalente a los contenidos visuales o auditivos. La información equivalente debe cumplir la misma finalidad que los contenidos visuales o auditivos. Así, un texto equivalente para la imagen de una flecha ascendente que vincule con una tabla de contenidos, podría ser "Ir a tabla de contenidos". En algunos casos, un equivalente debería describir la apariencia del contenido visual (por ejemplo, para tablas complejas, carteles o diagramas) o el sonido del contenido auditivo (por ejemplo, para los ejemplos sonoros usados en educación).

Esta pauta enfatiza la importancia de aportar equivalentes textuales para los contenidos no textuales (por ejemplo, imágenes, sonido pregrabado, vídeo, etc). La importancia del texto equivalente radica en su capacidad para ser interpretado por vías que son accesibles para personas pertenecientes a diversos grupos de discapacidad usando diversa tecnología. El texto puede ser interpretado por sintetizadores de voz o dispositivos braille y puede ser presentado visualmente (en varios tamaños) en visualizadores de ordenador y papel. El sintetizador de voz es esencial para personas ciegas y para las que tienen dificultades de lectura que, a menudo, acompañan a discapacidades cognitivas, de aprendizaje o sordera. El braille es esencial para personas con discapacidad auditiva y visual, tanto como para los individuos que solamente son ciegos. La salida visual de texto beneficia tanto a los usuarios sordos como a la mayoría de usuarios de la web.

Proporcionar equivalentes no textuales (dibujos, vídeos, sonido) del texto es también beneficioso para algunos usuarios, especialmente las personas analfabetas o con dificultad para la lectura. En las películas o presentaciones visuales, la acción representada, tal como el lenguaje corporal u otras pistas visuales, podrían no estar acompañadas de suficiente información auditiva como para transmitir la misma información. A menos que se proporcionen descripciones verbales de las acciones representadas, las personas que no puedan ver (o visualizar) el contenido visual, no podrán percibirlo.

► Puntos de verificación:

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

Por ejemplo, en HTML:

- Utilice "alt" para los elementos IMG, INPUT y APPLET o proporcione texto equivalente en el contenido de los elementos OBJECT Y APPLET.
- Para contenidos complejos (por ejemplo, las gráficas) en los que el texto del atributo "alt" no es suficiente, proporcione una descripción adicional usando, por ejemplo longdesc" con IMG o FRAME, un enlace dentro de un elemento OBJECT o un enlace descriptivo en el documento.
- Para mapas de imagen, use el atributo "alt" con AREA o el elemento MAP con elementos A (y otro texto) como contenido.

Consultar también punto de verificación 9.1 y punto de verificación 13.10.

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

Consultar también punto de verificación 1.5 y punto de verificación 9.1.

1.3 Hasta que las aplicaciones de usuario puedan leer automáticamente el texto equivalente de la banda visual, proporcione una descripción auditiva de la información importante de la pista visual de una presentación multimedia. [Prioridad 1]

Sincronice la descripción auditiva con la banda sonora como en el punto de verificación 1.4.

Consultar también punto de verificación 1.1 para información sobre textos equivalentes para el contenido visual.

1.4 Para toda presentación multimedia tempodependiente (por ejemplo, una película o animación) sincronice alternativas equivalentes (por ejemplo, subtítulos o descripciones de la banda visual) con la presentación. [Prioridad 1]

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. [Prioridad 3]

Consultar también punto de verificación 1.2 y punto de verificación 9.1.

Pauta 2: No se base sólo en el color.

Asegúrese de que los textos y gráficos son comprensibles cuando se vean sin color.

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.

► Puntos de verificación:

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. [Prioridad 1]

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

Pauta 3. Utilice marcadores y hojas de estilo y hágalo apropiadamente.

Marque los documentos con los elementos estructurales apropiados. Controle la presentación con hojas de estilo en vez de con elementos y atributos de 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 (consultar la descripción de diferencia entre contenido, estructura y presentación).

Los desarrolladores de contenidos pueden sentir la tentación de usar (o usar mal) construcciones que aseguren el formato deseado en los navegadores antiguos. Deben darse cuenta de que estas prácticas causan problemas de accesibilidad y deben considerar si el formato es tan importante como para hacer el documento inaccesible a algunos usuarios.

En el otro extremo, los desarrolladores de contenidos no deben sacrificar el marcador apropiado porque un determinado navegador o ayuda técnica no pueda procesarlo correctamente. Por ejemplo, es apropiado usar el elemento TABLE en HTML para marcar información tabular aunque algunos lectores de pantalla antiguos no manejen correctamente el texto contiguo (consultar el punto de verificación 10.3). Usando el elemento TABLE correctamente y creando tablas que se transformen adecuadamente (consultar la pauta 5) hace posible al software interpretar tablas de otra forma que como rejilla en dos dimensiones.

► Puntos de verificación:

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

Por ejemplo, utilice MathML para marcar ecuaciones matemáticas y hojas de estilo para el formato de texto y el control de la maquetación. Igualmente, evite la utilización de imágenes para representar textos. Utilice en su lugar texto y hojas de estilo.

Consultar también pauta 6 y pauta 11.

3.2 Cree documentos que estén validados por las gramáticas formales publicadas [Prioridad 2]

Por ejemplo, incluya una declaración del tipo de documento, al comienzo del mismo, que haga referencia a una DTD publicada (por ejemplo, la DTD HTML 4.0 estricto).

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

Por ejemplo, utilice la propiedad 'font' de CSS en lugar del elemento HTML FONT para controlar el estilo de las fuentes.

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. [Prioridad 2]

Por ejemplo, en CSS, utilice 'em' o medidas porcentuales, en vez de 'pt' (puntos) o 'cm' (centímetros), que son unidades absolutas. Si se usan unidades absolutas, valide que el contenido presentado es utilizable. [NdT]

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

Por ejemplo, en HTML, utilice H2 para indicar una subsección de H1. No utilice encabezados para hacer efectos de fuente.

3.6 Marque correctamente las listas y los ítems de las listas. [Prioridad 2]

Por ejemplo, en HTML, anide los elementos de listas OL, UL y DL adecuadamente.

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

Por ejemplo en HTML, utilice los elementos Q y BLOCKQUOTE para marcar citas cortas y largas, respectivamente.

Pauta 4. Identifique el idioma usado.

Use marcadores que faciliten la pronunciación o interpretación de texto abreviado o extranjero.

Cuando los desarrolladores de contenido especifican los cambios en el idioma de un documento, los sintetizadores de voz y los dispositivos braille pueden cambiar automáticamente al nuevo lenguaje, haciendo el documento más accesible a usuarios multilingües. Los desarrolladores de contenido deberían identificar el idioma predominante del contenido de un documento (a través de un marcador o en el encabezado HTTP). Deberían también proporcionar la expansión de las abreviaturas y los acrónimos.

Además de apoyar a las ayudas técnicas, la identificación del idioma usado permite a los motores de búsqueda localizar las palabras claves e identificar los documentos en el idioma deseado. Los marcadores de idioma mejoran también la legibilidad de la web para todo el mundo, incluso para aquellos con discapacidades de aprendizaje, cognitivas o sordera.

Cuando los cambios en las abreviaturas y el idioma no son identificados, pueden ser indescifrables para los lectores de pantalla y los dispositivos braille.

► Puntos de verificación:

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

Por ejemplo en HTML, utilice el atributo "lang". En XML, utilice "xml:lang".

4.2 Especifique la expansión de cada abreviatura o acrónimo cuando aparezcan por primera vez en el documento. [Prioridad 3]

Por ejemplo, en HTML, use el atributo "title" de los elementos "ABBR" y "ACRONYM". Proporcionar la expansión en el cuerpo principal del documento también ayuda a la usabilidad del documento.

4.3 Identifique el idioma principal de un documento. [Prioridad 3]

Por ejemplo, en HTML, coloque el atributo "lang" en el elemento HTML. En XML, utilice "xml:lang". Los operadores de servidores podrían configurar sus servidores para aprovechar los mecanismos de transferencia del contenido del protocolo HTTP ([RFC2068], sección 14.13), de forma que los clientes puedan recibir automáticamente los documentos en el idioma seleccionado.

Pauta 5. Cree tablas que se transformen correctamente.

Asegure que las tablas tienen los marcadores necesarios para transformarlas mediante navegadores accesibles y otras aplicaciones de usuario.

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 (consultar punto de verificación 10.3).

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 proporcionarán a la aplicación de usuario la información necesaria para ello (consultar también la pauta 3).

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.).

► Puntos de verificación:

5.1 En las tablas de datos, identifique los encabezamientos de fila y columna. [Prioridad 1]

Por ejemplo, en HTML, use TD para identificar las celdas de datos y TH para los encabezamientos.

5.2 Para las tablas de datos que tienen dos o más niveles lógicos de encabezamientos de fila o columna, utilice marcadores para asociar las celdas de encabezamiento y las celdas de datos. [Prioridad 1]

Por ejemplo, en HTML, utilice THEAD, TFOOT, y TBODY, para agrupar las filas, COL y COLGROUP para agrupar las columnas y los atributos "axis", "scope" y "headers" para describir relaciones más complejas entre los datos.

5.3 No utilice tablas para maquetar, a menos que la tabla tenga sentido cuando se alinee. Por otro lado, si la tabla no tiene sentido, proporcione una alternativa equivalente (la cual debe ser una versión alineada). [Prioridad 2]

Nota. Una vez que las aplicaciones de usuario soporten la colocación mediante hojas de estilo, las tablas no se deben utilizar para maquetar. Consultar también punto de verificación 3.3.

5.4 Si se utiliza una tabla para maquetar, no utilice marcadores estructurales para realizar un efecto visual de formato. [Prioridad 2]

Por ejemplo, en HTML no utilice elemento TH para hacer que el contenido de una celda (que no sea de encabezamiento de tabla) se visualice centrado y en negrita.

5.5 Proporcione resúmenes de las tablas. [Prioridad 3]

Por ejemplo, en HTML, use el atributo "summary" en el elemento TABLE.

5.6 Proporcione abreviaturas para las etiquetas de encabezamiento. [Prioridad 3]

Por ejemplo, en HTML, use el atributo "abbr" en el elemento TH.

Consultar también punto de verificación 10.3.

Pauta 6. Asegúrese de que las páginas que incorporan nuevas tecnologías se transformen correctamente.

Asegúrese de que las páginas son accesibles incluso cuando no se soportan las tecnologías más modernas o éstas estén desconectadas.

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 característica.

► Puntos de verificación:

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. [Prioridad 1]

Cuando el contenido está organizado lógicamente, es interpretado de forma que la organización continúa siendo clara incluso cuando se desconecten o no se soporten las hojas de estilo.

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

6.3 Asegúrese de que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, applets u otros objetos programados. Si esto no es posible, proporcione información equivalente en una página alternativa accesible. [Prioridad 1]

Por ejemplo, asegúrese de que los enlaces que lanzan scripts funcionan cuando éstos se desconecten o no se soporten (por ejemplo, no utilizar un "javascript" como objetivo de un enlace). Si no es posible hacer la página utilizable sin scripts, proporcione un texto equivalente con el elemento NOSCRIPT o utilice un script del servidor en lugar de un script de cliente o proporcione una página alternativa accesible como para el punto de verificación 11.4. Consultar también la pauta 1.

6.4 Para los scripts y applets, asegúrese de que los manejadores de evento sean independientes del dispositivo de entrada. [Prioridad 2]

Consultar la definición de independencia del dispositivo.

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

Por ejemplo en HTML, utilice NOFRAMES al final de cada 'frameset'. Para algunas aplicaciones, los scripts del servidor pueden ser más accesibles que los del cliente.

Consultar también punto de verificación 11.4.

Pauta 7. Asegure al usuario el control sobre los cambios de los contenidos tempo-dependientes.

Asegúrese de que los objetos o páginas que se mueven, parpadean, se desplazan o se actualizan automáticamente, puedan ser detenidos o parados.

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.

Nota: Todos los puntos de verificación que siguen, implican alguna responsabilidad por parte del desarrollador del contenido hasta que las aplicaciones de usuario proporcionen adecuados mecanismos de control de la característica.

► Puntos de verificación:

7.1 Hasta que las aplicaciones de usuario permitan controlarlo, evite provocar destellos en la pantalla. [Prioridad 1]

Nota: Los usuarios con epilepsia fotosensitiva pueden tener ataques desencadenados por parpadeos o destellos que oscilen entre los 4 y los 59 destellos por segundo (hertzios), con un nivel máximo a los 20 destellos por segundo, así como con los cambios rápidos de oscuridad a iluminación (como las luces estroboscópicas).

7.2 Hasta que las aplicaciones de usuario permitan controlarlo, evite el parpadeo del contenido (por ejemplo, cambio de presentación en periodos regulares, así como el encendido y apagado). [Prioridad 2]

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

Cuando una página incluye contenido móvil, proporcione un mecanismo dentro de un script o un applet que permita a los usuarios congelar el movimiento o actualización. El uso de las hojas de estilo con scripts que creen movimiento, permite a los usuarios desconectar u obviar el efecto más fácilmente. Consultar también la pauta 8.

7.4 Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener las actualizaciones, no cree páginas que se actualicen automáticamente de forma periódica. [Prioridad 2]

Por ejemplo, en HTML, no cree páginas que se actualicen automáticamente con "HTTP EQUIV=refresh" hasta que las aplicaciones de usuario permitan desconectar esta característica.

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. [Prioridad 2]

Nota. Los elementos BLINK y MARQUEE no están definidos en ninguna especificación W3C HTML, y no deberían ser utilizados. Consultar también la pauta 11.

Pauta 8. Asegure la accesibilidad directa de las interfaces de usuario incrustadas.

Asegure que la interfaz de usuario sigue los principios de un diseño accesible: funcionalidad de acceso independiente del dispositivo, teclado operable, voz automática, etc.

Cuando un objeto incrustado tiene su "propia interfaz", ésta (al igual que la interfaz de su navegador) debe ser accesible. Si la interfaz del objeto incrustado no puede hacerse accesible, debe proporcionarse una solución alternativa accesible.

Nota: Para información sobre interfaces accesibles, por favor consulte las Pautas de Accesibilidad a las Aplicaciones de Usuario [WAI-USERAGENT] y las Pautas de Accesibilidad para las Herramientas de Creación [WAI-AUTOOL].

► Puntos de verificación:

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]

Consultar también la pauta 6.

Pauta 9. Diseño para la independencia del dispositivo.

Utilice características que permitan la activación de los elementos de la página a través de diversos dispositivos de entrada.

El acceso independiente del dispositivo significa que el usuario puede interactuar con la aplicación de usuario o el documento con un dispositivo de entrada (o salida) preferido - ratón, teclado, voz, puntero de cabeza (licornio) u otro. Si, por ejemplo, un control de formulario sólo puede ser activado con un ratón u otro dispositivo de apuntamiento, alguien que use la página sin verla, con entrada de voz, con teclado o quien utilice otro dispositivo de entrada que no sea de apuntamiento, no será capaz de utilizar el formulario.

Nota: Proporcionando textos equivalentes para los mapas de imagen o las imágenes usadas como vínculos, se hace posible a los usuarios interactuar con ellos sin un dispositivo de apuntamiento. Consultar también la pauta 1.

Generalmente, las páginas que permiten la interacción a través del teclado son también accesibles a través de una entrada de voz o una serie de comandos.

► Puntos de verificación:

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

Consultar también punto de verificación 1.1, punto de verificación 1.2, y punto de verificación 1.5.

9.2 Asegúrese de que cualquier elemento que tiene su propia interfaz pueda manejarse de forma independiente del dispositivo. [Prioridad 2]

Consultar la definición de independencia del dispositivo.

Consultar también la pauta 8.

9.3 Para los "scripts", especifique manejadores de evento lógicos en vez de manejadores de evento dependientes de dispositivos. [Prioridad 2]

9.4 Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos. [Prioridad 3]

Por ejemplo, en HTML, especifique el orden de navegación con el tabulador a través del atributo "tabindex" o asegure un diseño de página lógico.

9.5 Proporcione atajos de teclado para los vínculos más importantes (incluidos los de los mapas de imagen de cliente), los controles de formulario y los grupos de controles de formulario. [Prioridad 3]

Por ejemplo, en HTML, especifique los atajos a través del atributo "accesskey".

Pauta 10. Utilice soluciones provisionales.

Utilice soluciones de accesibilidad provisionales de forma que las ayudas técnicas y los antiguos navegadores operen correctamente.

Por ejemplo, los navegadores antiguos no permiten al usuario navegar a cuadros de edición vacíos. Los antiguos lectores de pantalla leen las listas de vínculos consecutivos como un solo vínculo. Estos elementos activos son, por tanto, de difícil o imposible acceso. Igualmente, cambiar la ventana actual o hacer aparecer inesperadamente nuevas ventanas, puede ser muy desorientador para los usuarios que no pueden ver lo que está ocurriendo.

Nota: Los siguientes puntos de verificación se aplican hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) solucionen estos problemas. Estos puntos de verificación están clasificados como "provisionales" lo que significa que el Grupo de Trabajo de las Pautas de Contenido en la Web los considera válidos y necesarios para la accesibilidad de la web en el momento de la publicación de este documento. Sin embargo, el Grupo de Trabajo espera que estos puntos de verificación no sean necesarios en un futuro, una vez que las tecnologías de la web hayan incorporado las características y capacidades esperables.

► Puntos de verificación:

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

Por ejemplo, en HTML, evite usar un marco cuyo objetivo es una nueva ventana.

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

La etiqueta debe preceder inmediatamente a su control en la misma línea (se permite más de una etiqueta/control por línea) o estar en la línea que precede al control (con sólo una etiqueta y un control por línea) [NdT]. Consultar también punto de verificación 12.4.

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, columnas envoltorio de palabras. [Prioridad 3]

Nota: Por favor, consulte la definición de tabla alineada. Este punto de verificación beneficia a aquellos que tienen aplicaciones de usuario (como algunos lectores de pantalla) que son incapaces de manejar bloques de texto contiguo; el punto de verificación no debe desanimar a los desarrolladores de contenidos en el uso de tablas para presentar información tabular.

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. [Prioridad 3]

Por ejemplo, en HTML, haga esto con TEXTAREA e INPUT.

10.5 Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten claramente los vínculos contiguos, incluya caracteres imprimibles (rodeados de espacios), que no sirvan como vínculo, entre los vínculos contiguos. [Prioridad 3]

Pauta 11. Utilice las tecnologías y pautas W3C.

Utilice tecnologías W3C (de acuerdo con las especificaciones) y siga las pautas de accesibilidad. Donde no sea posible utilizar una tecnología W3C, o usándola se obtengan materiales que no se transforman correctamente, proporcione una versión alternativa del contenido que sea accesible.

Las actuales pautas recomiendan las tecnologías W3C (por ejemplo, HTML, CSS, etc.) por varias razones:

- Las tecnologías W3C incluyen características accesibles "incorporadas".
- Las especificaciones W3C pronto serán revisadas para asegurar que los temas de accesibilidad se toman en consideración en la fase de diseño.
- Las especificaciones W3C están desarrolladas en un proceso abierto de laboratorio consenso.

Muchos formatos no recomendados por W3C (por ejemplo, PDF, Shockwave, etc.) requieren ser vistos bien con plug-ins o con aplicaciones autónomas. A menudo, estos formatos no pueden ser visualizados o navegados con aplicaciones de usuario estándares (incluyendo ayudas técnicas). Evitar estos formatos y características no estándar (elementos, atributos, propiedades y extensiones patentados), tenderá a hacer más accesibles las páginas a más gente que utiliza una amplia variedad de hardware y software. Cuando deba utilizar tecnologías no accesibles (patentadas o no), debe proporcionar una página equivalente accesible.

Incluso cuando se utilicen tecnologías W3C, deben ser usadas de acuerdo con las pautas de accesibilidad. Cuando utilice nuevas tecnologías, asegúrese de que se transforman correctamente (consultar también la pauta 6).

Nota: Convertir los documentos (desde PDF, Postscript, RTF, etc.) a lenguajes de marcado W3C (HTML, XML) no siempre crea un documento accesible. Por tanto, valide cada página respecto a la accesibilidad y utilidad después del proceso de conversión. Si una página no se convierte de forma legible, revise la página hasta que su presentación original se convierta adecuadamente o bien proporcione una versión en HTML o en texto plano.

► Puntos de verificación:

11.1 Utilice tecnologías W3C cuando estén disponibles y sean apropiadas para la tarea y use las últimas versiones que sean soportadas. [Prioridad 2]

Consulte la lista de referencias para información sobre dónde encontrar las últimas especificaciones W3C y [WAI-UA-SUPPORT] para información sobre como las aplicaciones de usuario que soportan las tecnologías W3C.

11.2 Evite características desaconsejadas por las tecnologías W3C. [Prioridad 2]

Por ejemplo, en HTML, no utilice el elemento desaconsejado FONT; use en su lugar hojas de estilo (por ejemplo, la propiedad "font" en CSS).

11.3 Proporcione la información de modo que los usuarios puedan recibir los documentos según sus preferencias (por ejemplo, idioma, tipo de contenido, etc.). [Prioridad 3]

Nota: Use la negociación de contenidos donde sea posible.

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. [Prioridad 1]

Nota: Los desarrolladores de contenido sólo deben enviar a páginas alternativas cuando otras soluciones fallen, porque las páginas alternativas se actualizan con menor frecuencia que las páginas primarias. Una página no actualizada puede ser tan frustrante como una página inaccesible, puesto que en ambos casos, la información de la página original no está disponible. La generación automática de páginas alternativas puede conducir a actualizaciones más frecuentes, pero los desarrolladores de contenidos deben asegurar que las páginas generadas siempre tengan sentido y que los usuarios puedan navegar por el sitio siguiendo los vínculos de las páginas primarias, las páginas alternativas o ambas. Antes de enviar a una página alternativa, reconsidere el diseño de la página original; haciéndola accesible es probable que la mejore para todos los usuarios.

Pauta 12. Proporcione información de contexto y orientación.

Proporcione información de contexto y orientativa para ayudar a los usuarios a entender páginas o elementos complejos.

Agrupar los elementos y proporcionar información contextual sobre la relación entre elementos puede ser útil a todos los usuarios. Las relaciones complejas entre las partes de una página pueden resultar difíciles de interpretar a personas con discapacidades cognitivas o visuales.

► Puntos de verificación:

12.1 Titule cada marco para facilitar su identificación y navegación. [Prioridad 1]

Por ejemplo, en HTML, utilice el atributo "title" en los elementos FRAME.

12.2 Describa el propósito de los marcos y como éstos se relacionan entre sí, si no resulta obvio solamente con el título del marco. [Prioridad 2]

Por ejemplo, en HTML, utilice "longdesc" o un vínculo a una descripción.

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

Por ejemplo, en HTML, utilice OPTGROUP para agrupar los elementos OPTION dentro de un SELECT; agrupe controles de formulario con FIELDSET y LEGEND; utilice listados anidados cuando sea apropiado; utilice encabezamientos para estructurar documentos, etc.

Consultar también la pauta 3.

12.4 Asocie explícitamente las etiquetas con sus controles. [Prioridad 2]

Por ejemplo, en HTML, utilice LABEL y su atributo "for".

Pauta 13. Proporcione mecanismos claros de navegación.

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 ceguera y benefician a todos los usuarios.

► Puntos de verificación:

13.1 Identifique claramente el objetivo de cada vínculo. [Prioridad 2]

El texto del vínculo tiene que tener significado suficiente cuando sea leído fuera de contexto (por sí mismo o como parte de una secuencia de vínculos). También debe ser conciso.

Por ejemplo, en HTML, escriba "información sobre la versión 4.3" en lugar de "pincha aquí". Además de textos de vínculos claros, los desarrolladores de contenidos deben aclarar el objetivo de un vínculo con un título informativo del mismo (por ejemplo, en HTML, el atributo "title"),

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

Por ejemplo, use RDF ([RDF]) para indicar el autor de los documentos, el tipo de contenido, etc.

Nota: Algunas aplicaciones de usuario de HTML pueden construir herramientas de navegación a partir de las relaciones entre documentos descritas en el elemento HTML LINK y los atributos "rel" o "rev" (por ejemplo rel="siguiente"; rel="anterior"; rel="índice", etc.). Consultar también el punto de verificación 13.5.

13.3 Proporcione información sobre la maquetación general de un sitio (por ejemplo, mapa del sitio o tabla de contenidos). [Prioridad 2]

En la descripción de la maquetación del sitio, destaque y explique las características de accesibilidad disponibles.

13.4 Utilice los mecanismos de navegación de forma coherente. [Prioridad 2]

13.5 Proporcione barras de navegación para destacar y dar acceso al mecanismo de navegación. [Prioridad 3]

13.6 Agrupe los vínculos relacionados, identifique el grupo (para las aplicaciones de usuario) y, hasta que las aplicaciones de usuario lo hagan, proporcione una manera de evitar el grupo. [Prioridad 3]

13.7 Si proporciona funciones de búsqueda, permita diferentes tipos de búsquedas para diversos niveles de habilidad y preferencias. [Prioridad 3]

13.8 Localice al principio de los encabezamientos, párrafos, listas, etc., la información que los diferencie. [Prioridad 3]

Nota: Esto es comúnmente denominado "front-loading" (colocar al frente) y es especialmente útil para los que acceden a la información con dispositivos seriales como un sintetizador de voz.

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

Por ejemplo, en HTML, especifique las colecciones de documentos con el elemento LINK y los atributos "rel" y "rev". Otro modo de crear una colección es

construyendo un archivo (por ejemplo, con zip, tar and gzip, stuffit, etc.) de las páginas múltiples.

Nota: La mejora en la presentación ganada por un procesamiento fuera de línea (offline) puede hacer la navegación mucho menos costosa a las personas con discapacidad que puedan estar navegando lentamente.

13.10 Proporcione una manera de saltar sobre un ASCII art de varias líneas. [Prioridad 3]

Consultar punto de verificación 1.1 and ejemplo de ASCII art en el glosario.

Pauta 14. Asegúrese de que los documentos sean claros y simples.

Asegure que los documentos son claros y simples para que puedan ser más fácilmente comprendidos.

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 las personas con discapacidad visual, los de baja visión o para cualquier usuario que no puede o ha elegido no ver los gráficos. Consulte también la pauta 1).

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.

► Puntos de verificación:

14.1 Utilice el lenguaje apropiado más claro y simple para el contenido de un sitio. [Prioridad 1]

14.2 Complemente el texto con presentaciones gráficas o auditivas cuando ello facilite la comprensión de la página. [Prioridad 3]

Consultar también la pauta 1.

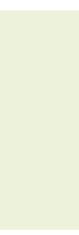
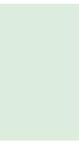
14.3 Cree un estilo de presentación que sea coherente para todas las páginas. [Prioridad 3]

Copyright © 1994-2006 W3C ® (MIT, ERCIM, Keio). Todos los derechos reservados. Son aplicables las reglas de responsabilidad, marcas registradas, utilización de documentos y licencias de software.

Bibliografía

- **How People with Disabilities Use the Web.**
(<http://www.w3.org/WAI/EO/Drafts/PWD-Use-Web/Overview.html>) W3C Working Draft, 8 Julio de 2004. Judy Brewer.
- **Web Content Accessibility Guidelines 1.0.**(<http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/>) Recomendación W3C, 5 de Mayo de 1999. Wendy Chisholm, Gregg Vanderheiden, Ian Jacobs.
- **Alternative Web Browsing.** (<http://www.w3.org/WAI/References/Browsing>) Peter Boshier y Judy Brewer (2001).
- **Web Accessibility Initiative (WAI).** (<http://www.w3.org/WAI/>) Iniciativa de accesibilidad Web.
- **W3C Oficina Española.** (<http://www.w3c.es/>) Oficina española del W3C.
- **What is Universal Design?**
(http://www.design.ncsu.edu:8120/cud/univ_design/princ_overview.htm) NC State University, The Center for Universal Design (1997).
- **LEY 34/2002, de 11 de julio**, de servicios de la sociedad de la información y de comercio electrónico (<http://www.boe.es/g/es/boe/dias/2002/07/12/>). 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, (<http://sid.usal.es/mostrarficha.asp?ID=5979&Fichero=3.1.2>). LIONDAU, publicada el BOE del 03/12/2003, nº 289.
- **Informe al Parlamento Europeo.**(http://europa.eu.int/information_society/policy/accessibility/web/wai_2002/a_documents/cec_com_web_wai_2001.html) sobre la Comunicación de la Comisión de Industria, Comercio Exterior, Investigación y Energía.
- **Discapnet, Discapacidades y Deficiencias.**
(<http://www.discalpnet.es/Discalpnet/Castellano/Salud/default.htm>) Portal Web sobre la salud y la discapacidad.
- **Clases de productos de tecnología de asistencia.**
(<http://www.microsoft.com/spain/accesibilidad/at/types.aspx>) Microsoft® Accessibility Technology for Everyone.
- **Tecnum, Catálogo de Ayudas Técnicas** (<http://www.tecnum.net/tecnum.htm>).

- **Evaluation, Repair, and Transformation Tools for Web Content Accessibility.** (<http://www.w3.org/WAI/ER/existingtools.html>) Shadi Abou-Zahra, Wendy Chisholm, Judy Brewer.
- **Evaluating Web Sites for Accessibility.** (<http://www.w3.org/WAI/eval/>) Judy Brewer, Chuck Letourneau.
- **Comprendiendo la Accesibilidad: Una Guía para Lograr la Conformidad en Sitios Web e Intranets.** (<http://www.hisoftware.com/uaes/uaccess-es.htm>) Robert B. Yonaitis (2002, HiSoftware Inc.).



Financia:



UNIÓN EUROPEA

Fondo Social Europeo



MINISTERIO
DE TRABAJO
Y ASUNTOS SOCIALES

Edita:



Realiza:



FUNDACIÓN
CTIC
Centro Tecnológico
de la Información y la Comunicación