

Guía para el Desarrollo de la Web del Estado

Guía Técnica de accesibilidad y estándares W3C para la Web del Estado Uruguayo

2013

Ciudadanía Digital

Licencia de uso de este documento	4
Introducción	5
Objetivo	5
Ámbito de aplicación	5
Referencia Técnica	6
WCAG 2.0.....	6
Tecnologías compatibles con la accesibilidad.....	7
Estándares Web.....	9
Introducción	9
(X)HTML y CSS	9
Metadatos	14
DOM para la manipulación del contenido	17
Accesibilidad de las herramientas de autor (ATAG).....	18
Agentes de usuario.....	19
Versiones alternativas accesibles	20
Imágenes.....	22
Imágenes decorativas.....	23
Imágenes funcionales.....	24
CAPTCHAS.....	25
Mapas de imágenes.....	26
Audio, vídeo y multimedia	27
Destellos	30
Movimiento y Parpadeo	31
Estructura y semántica	33
Encabezados	34
Listas	34
Datos tabulares.....	35
Énfasis estructural	35
Citas	36
Presentación	37
Significado mediante características sensoriales	37
Contraste	38
Redimensión del texto.....	39
Imágenes de texto	39
Marcos (Frames e Iframes).....	40
Tablas de maquetación.....	41
Foco visible	42
Legibilidad	43
Encabezados y etiquetas	43
Vocabulario neutro.....	43
Idiomas	44

Navegación en interacción	46
Enlaces y mecanismos de navegación.....	46
Título de páginas	47
Opciones de navegación.....	47
Información sobre accesibilidad.....	48
Evitar bloques de material repetido.....	49
Independencia de dispositivo	49
Ventanas emergentes	50
Datos de entrada	51
Contenido dinámico	55
Introducción	55
Scripts	55
Contenido con límite de tiempo.....	61
Recargas y redireccionamiento	62
Formatos diferentes de (X)HTML	64
Introducción	64
Flash.....	64
Documentos PDF	65
WAI-ARIA	67
Aplicaciones Ricas de Internet Accesibles	69

Licencia de uso de este documento

Este documento ha sido desarrollado por **AGESIC** (Agencia de Gobierno Electrónico y Sociedad de la Información).

Este documento ha sido distribuida bajo licencia Creative Commons, bajo las condiciones de **reconocimiento** y **uso no comercial**, que permite la utilización de la misma para todo el mundo, pudiéndose reutilizar total o parcialmente el contenido de dicho documento siempre que sea para uso no comercial y manteniendo el reconocimiento de autoría de la obra.



Reconocimiento - NoComercial (by-nc): Se permite la generación de obras derivadas siempre que no se haga un uso comercial. Tampoco se puede utilizar la obra original con finalidades comerciales.

Introducción

Objetivo

El objetivo de los requisitos presentes en esta guía es permitir la implantación y/o mejora sustancial de las condiciones de accesibilidad de los sitios web pertenecientes al Estado Uruguayo.

Para cumplir con este objetivo se establecen requisitos mínimos de Accesibilidad para el contenido web, basados en el nivel **AA** (doble "A") de las Pautas de Accesibilidad para el Contenido Web **WCAG 2.0**, además de algunos puntos de **AAA** (triple "A"), de las Buenas prácticas de la Web Móvil y de usabilidad.

Ámbito de aplicación

Los contenidos de este documento serán de aplicación a todos los portales e intranets del organismo, incluyendo las zonas restringidas de acceso mediante usuario y contraseña que estén disponibles para los ciudadanos, especialmente las aplicaciones web de trámites y servicios que permitan interacción con los mismos.

Estos requisitos alcanzan a desarrollos realizados por el organismo como a todos los de proveedores externos.

Estos requisitos habrán de tenerse en cuenta en el diseño gráfico, diseño de la arquitectura de la información y desarrollo de los portales.

Importante: las normas y recomendaciones descritas a continuación están plenamente interrelacionadas y son de aplicación global, por lo que no se deben tomar como apartados aislados, ya que un mismo elemento puede verse afectado por distintas recomendaciones de diferentes apartados de la norma.

Referencia Técnica

WCAG 2.0

En diciembre de 2008, la WAI publicó la nueva versión de las **Pautas de Accesibilidad para el Contenido Web (WCAG) 2.0** (www.w3.org/TR/WCAG20/). Construidas sobre la creciente experiencia y las aportaciones realizadas por la comunidad y público relacionados con la accesibilidad (personal encargado del desarrollo web, usuarios y usuarias, etc.), las WCAG 2.0 añaden una serie de mejoras sobre el estándar inicial creado por el W3C.

La base de las pautas son los **4 principios** fundamentales: **perceptible, operable, comprensible y robusto**. Bajo estos principios se establecen las pautas, que proporcionan objetivos básicos sobre los que se debe trabajar para ofrecer un contenido más accesible para personas con distintos tipos de discapacidad. Se trata de **12 pautas** que no son comprobables de forma directa. Para cada una de las pautas se articulan **criterios de conformidad** comprobables. Cualquier tipo de contenido web puede ser evaluado y comprobado frente a estos criterios de conformidad.

Cada criterio de conformidad tiene asignado un determinado nivel y en base al cumplimiento de estos criterios se establece el nivel de conformidad del sitio web:

- » **Nivel A:** la página Web satisface todos los criterios de conformidad de nivel A o se proporciona una versión alternativa que cumple este nivel.
- » **Nivel AA:** la página Web satisface todos los criterios de conformidad de nivel A y nivel AA o se proporciona una versión alternativa que cumple este nivel.
- » **Nivel AAA:** la página Web satisface todos los criterios de conformidad de nivel A, nivel AA y nivel AAA o se proporciona una versión alternativa que cumple este nivel.

Las recomendaciones de esta guía técnica de accesibilidad se basan en las **WCAG 2.0** para un nivel de conformidad **AA**.

Para facilitar su aplicación, esta guía técnica no se estructura según las WCAG 2.0 sino que se ha optado por una organización práctica en torno a temas y elementos comunes de los sitios web, así como por una abstracción de los requisitos de accesibilidad explicados respecto a las Pautas y Criterios de Conformidad de las WCAG 2.0. No obstante, y para cada uno de los apartados de la guía técnica, se incluye una tabla con las referencias a los criterios de conformidad de las WCAG 2.0 que están relacionados con el tema tratado.



Tecnologías compatibles con la accesibilidad

Muchos de los criterios de conformidad tratan sobre cómo crear contenido accesible basándose en los productos de apoyo o en características especiales de accesibilidad de los principales navegadores. Es decir, el criterio de conformidad requiere que se haga algo en el contenido web de forma que los productos de apoyo puedan mostrar correctamente la información de ese contenido a las personas.

Por ejemplo, para que la información que proporciona una imagen esté disponible para las personas con ceguera es necesario ofrecer una alternativa textual. Lo importante aquí es que esta alternativa textual se incluya de una forma **compatible con la accesibilidad**, es decir, los navegadores y productos de apoyo (p.ej. lectores de pantalla) sean capaces de encontrarla, reconocerla y mostrarla a las personas. Así, en HTML, para proporcionar alternativas textuales podemos usar el atributo **alt** del elemento ``.

Cuando surge una **nueva tecnología web** es necesario que se cumplan dos condiciones para que el contenido creado con dicha tecnología esté disponible para las personas que usan productos de apoyo:

1. La tecnología debe **diseñarse de forma que los navegadores y productos de apoyo puedan acceder a toda la información necesaria** para mostrar el contenido a las personas.
2. Puede ser necesario rediseñar o modificar los navegadores y **productos de apoyo** para que sean **capaces de trabajar con la nueva tecnología**.

Siguiendo con el ejemplo de HTML, este lenguaje permite proporcionar alternativas textuales para las imágenes a través del atributo **alt** del elemento ``. Los navegadores están diseñados para mostrar dicho texto si la imagen no se puede mostrar. De igual forma, los lectores de pantalla leen el contenido del atributo **alt**.

El concepto **compatible con la accesibilidad** significa que ambas condiciones se han cumplido. Es decir, la tecnología dispone de los mecanismos necesarios para proporcionar información de accesibilidad y los navegadores y productos de apoyo son capaces de comprender estos mecanismos y proporcionar dicha información a las personas que la requieran.

En las WCAG 2.0 no se limita el uso de tecnologías web únicamente a tecnologías propias del W3C, sino que permite **usar cualquier tecnología web que sea compatible con la accesibilidad** siempre que:

- » **La tecnología se use de forma accesible:** es decir, la forma en la que se usa ha sido probada y es compatible con los productos de apoyo empleados por las personas.
- » **Las personas que usan la web dispongan de navegadores y productos de apoyo que soporten dicha tecnología:** es decir, los navegadores deben tener soporte nativo para la tecnología, estar ampliamente difundidos o estar disponibles para su descarga o compra con la misma facilidad y precio para las personas con discapacidad como para una persona sin discapacidad. Lo mismo ocurre en el caso que sea necesario un *plug-in* para acceder a la tecnología.

Nota: este requisito permite usar tecnologías sin soporte de accesibilidad o usadas de forma no accesible siempre y cuando no se dependa de ellas. Es decir, la página web puede ser conforme si existen alternativas accesibles para el contenido que se incluye con tecnologías no accesibles o de forma no accesible.

Estándares Web

Introducción

Los Estándares Web del W3C, al igual que el resto de estándares existentes en cualquier área, tienen la finalidad principal de asegurar la calidad final de los productos así como la interoperabilidad presente y futura, haciendo posible la comunicación efectiva de todos los componentes de la Web.

Además, los Estándares Web del W3C se desarrollan teniendo siempre en cuenta la accesibilidad en todas las fases del proceso, por lo que la accesibilidad es una característica más, común en todos ellos.

En el desarrollo de portales web los estándares utilizados son el (X)HTML (*HyperText Markup Language* -lenguaje de marcado de hipertexto-) como idioma de marcado para la creación de páginas web, el CSS (*Cascading Style Sheets* -hojas de estilo en cascada-), como elemento para la presentación de los contenidos de las páginas y el DOM (*Document Object Model* -Modelo de Objetos de Documentos-), idioma para programar scripts que permitan la modificación dinámica de los contenidos. También se deberá tener en cuenta la representación de información estructurada, este punto es abordado en mayor detalle en el capítulo 6. Estructura y semántica.

Toda la referencia técnica está disponible en el sitio web de W3C (www.w3.org) o en sus versiones localizadas al idioma español en el sitio web de la oficina española de W3C (www.w3c.es).

(X)HTML y CSS

El principal objetivo de las especificaciones del W3C 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. Las tecnologías del W3C además han sido creadas, entre otras cosas, teniendo en cuenta la accesibilidad. Para incorporar a los documentos los mayores beneficios se recomienda emplear las últimas versiones soportadas.

Respecto al uso de código (X)HTML y CSS se deben seguir las siguientes pautas:

(X)HTML

- » **Declaración de documento válida (DOCTYPE)**

Siempre se debe especificar la gramática que se vaya a utilizar mediante la declaración `DOCTYPE` al inicio de todo documento.

- » **Código válido gramaticalmente**

Se debe comprobar que los documentos son válidos según la especificación de (X)HTML indicada en la declaración de tipo de documento. Si no se valida el código frente a una gramática formal se estará comprometiendo la compatibilidad del código tanto con los navegadores actuales como futuros.

Para ello se puede usar el servicio de validación proporcionado por el W3C (validator.w3.org).

- » **Ausencia de elementos propietarios**

No se deben usar elementos propietarios no estándar (como `<marquee>`, `<blink>` y `<embed>`) debido al limitado soporte que tienen entre todos los navegadores y a la falta de compatibilidad con las especificaciones, impidiendo la validación de los documentos.

- » **Ausencia de elementos desaconsejados ("*deprecated*")**

Un elemento o atributo desaconsejado es aquel que ha quedado anticuado. Los elementos desaconsejados quedarán obsoletos en futuras versiones de HTML y por tanto los desarrolladores deben evitar su uso. Generalmente se trata de elementos o atributos de presentación (``, `border`, `bgcolor`, `align`, `height`, `width`, `language`, etc...) que deben desaparecer en beneficio de las hojas de estilo.

- » **XHTML de tipo "*Transitional*"**

Para todo nuevo desarrollo se utilizará XHTML 1.0 con un DTD de tipo *Transitional*, cumpliendo lo siguiente:

- > Ausencia de elementos desaconsejados.
- > Ausencia de elementos y atributos de presentación, utilizando hojas de estilo CSS en su lugar.

Como mejora, y siempre que sea técnicamente posible, se recomienda el uso de una gramática XHTML 1.0 de tipo *Strict*.

En los sitios web ya desarrollados, o en aquellos que tengan limitaciones técnicas que impidan el uso de las gramáticas antes citadas, se permitirá el uso de HTML 4.01 con un DTD *Transitional* cumpliendo con los requisitos anteriores de ausencia de elementos y atributos desaconsejados y/o de presentación. En estos casos se debe planificar una transición hacia XHTML 1.0.

- » **Gramática estricta ("*Strict*")**

Siempre que sea posible, es recomendable usar gramáticas estrictas por las ventajas que aportan frente a otras menos restrictivas como *Transitional* o *Frameset*.

Las gramáticas de tipo *Strict* están ideadas para lograr la máxima separación entre contenido y presentación. Este tipo de gramáticas no permiten el uso de elementos y atributos desaconsejados, principalmente con carácter presentacional (``, `<center>`, `<u>`), estando orientadas principalmente para su uso junto con hojas de estilo (CSS).

» **Usar (X)HTML de acuerdo a la especificación**

Un marcado (X)HTML adecuado no sólo debe validar, sino que debe usar los elementos y atributos con la finalidad que les corresponde según la especificación, de forma que describan el tipo de información que contienen. Es decir, que aporten la información estructural y semántica adecuada para el contenido (encabezados, párrafos, listas, énfasis, citas, etc.)

CSS

» **Código (CSS) válido gramaticalmente**

Siempre que sea posible se deberá utilizar CSS estándar, en su versión 2.1, como formato por defecto para la presentación de la información, teniendo en cuenta especialmente que no se deben utilizar extensiones propietarias o experimentales de los distintos agentes de usuario.

Se puede comprobar la corrección de las hojas de estilo con el servicio de validación proporcionado por el W3C (jigsaw.w3.org/css-validator/).

» **Separación de contenido y presentación**

Se deben usar únicamente las hojas de estilo para controlar la presentación y la maquetación de los documentos, en lugar de elementos y atributos (X)HTML de presentación evitando, en la medida de lo posible, el uso de tablas de maquetación y de imágenes decorativas en el código (X)HTML en vez de en las hojas de estilo.

Asimismo, siempre que sea posible, se deben usar archivos externos para las hojas de estilo, en lugar de estilos en línea (atributo `style`) o incluidos en la cabecera de los documentos (elemento `<style>`), para aprovechar todas las ventajas en cuanto a separación de contenido y presentación.

Por otra parte, la información y el contenido debe incluirse en el código (X)HTML y no debe generarse desde las hojas de estilo. Por ejemplo, las imágenes incluidas desde CSS deben ser puramente decorativas y no transmitir información alguna. De esta forma se asegura que, aunque no se soporten estas tecnologías, todo el contenido está presente en el documento.

Usar archivos externos para las hojas de estilo

Para aprovechar todas las ventajas de las hojas de estilo en cuanto a separación de estructura y presentación debemos usar hojas de estilo externas en vez de estilos en línea (atributo `style`) o estilos incluidos en la cabecera de los documentos (elemento `STYLE`)

Los estilos incluidos en línea o en la cabecera de los documentos dificultan la mantenibilidad ya que no se pueden modificar simultáneamente para todo el sitio Web, teniendo que hacer las modificaciones de forma individual para cada documento. Además, los estilos incluidos en el documento aumentan el tamaño de los mismos y, por tanto, el ancho de banda consumido y el tiempo necesario para descargarlos.

Sólo deberíamos emplear estilos en línea para estilos que sean una excepción al estilo general del sitio y de la página.

Estructurar y jerarquizar el código

Usar hojas de estilo correctamente estructuradas, organizadas y comentadas, así como emplear propiedades y selectores de forma coherente aprovechando las propiedades de las hojas de estilo, facilita su legibilidad y el futuro mantenimiento. A continuación se dan algunas recomendaciones básicas para ayudarnos a estructurar y organizar el código.

- » Usar el número mínimo de contenedores necesarios en el código (X)HTML. En ocasiones se abusa del elemento contenedor `DIV` con carácter genérico para lograr determinadas maquetaciones, de forma similar al uso de tablas de maquetación. Se deben aprovechar al máximo los elementos estructurales ya presentes en el código para conseguir el efecto deseado, sin emplear contenedores extra de forma innecesaria. Esto nos facilitará el tener luego unas hojas de estilo limpias y bien estructuradas.
- » Organizar los estilos según sus características:
 - En varias hojas de estilo para separar los estilos genéricos aplicables a todo el sitio web de aquellos específicos para plantillas o secciones.
 - Dentro de la misma hoja de estilos, en diferentes secciones. Por ejemplo, según la maquetación (cabecera, contenido, pie), el tipo de contenido (listas, formularios, texto, tablas), etc., e incluso una combinación de agrupaciones.
- » Agrupar selectores con estilos similares y aprovechar el poder de la cascada en CSS (<http://www.w3.org/TR/CSS21/cascade.html>). Cuando tengamos varios selectores que compartan las mismas propiedades debemos agruparlos para no repetirlas varias veces en las hojas de estilo. Si alguno de los selectores necesita un estilo específico

adicional se lo especificaremos aparte.

- » Aprovechar el poder de la herencia de las propiedades de CSS, definiendo estilos básicos en algunos elementos que se reaprovecharán en aquellos elementos que hereden sus propiedades. Por ejemplo, al igual que el tipo de fuente se puede definir para `BODY` y será aplicable a todo el documento (redefiniéndose sólo cuando sea necesario) también podemos aprovechar la herencia de otras propiedades.
- » Usar selectores contextuales (descendientes) siempre que sea posible en vez de abusar de identificadores (`id`) y nombres de clase. El uso de selectores contextuales (incluso los más largos) facilita la lectura de las hojas de estilo y permite reducir el uso de clases, identificadores, etc. Por ejemplo, para aplicar estilos a partir del segundo nivel de elementos de una lista anidada, en vez de usar clases o identificadores se puede usar el selector contextual: `li li`.
- » Usar notaciones abreviadas para definir grupos de propiedades relacionadas (márgenes, paddings, fuente, etc) Así, por ejemplo, en vez de usar (`margin-top: 1em; margin-right: 0; margin-bottom: 2em; margin-left: 0.5em;`) se puede usar (`margin: 1em 0 2em 0.5em;`).
- » Usar comentarios en las hojas de estilo que ayuden a identificar y separar las diferentes secciones empleadas, así como para comentar técnicas o *hacks* empleados.

HTML5

La especificación **HTML5** introduce nuevos elementos que permiten **mejorar la estructura y semántica** de los documentos. Asimismo, entre las novedades que introduce, caben destacar los nuevos tipos de campos de formulario. Este enriquecimiento de los componentes de interacción permite **mejorar la experiencia de los usuarios** así como **simplificar el desarrollo** al incorporar de forma nativa características que antes necesitaban ser programadas.

Sin embargo, y mientras HTML5 no esté ampliamente soportado, se deben usar las nuevas características de HTML5 de forma que se mantenga la **compatibilidad con los navegadores antiguos**, por lo que mientras no esté estandarizado y todos los agentes de usuario sean capaces de interpretar y renderizar de forma común el código es preciso desarrollar tanto para la versión (X)HTML como la versión HTML5, con todas las variantes existentes para cada navegador. Por otra parte, las **recomendaciones generales de accesibilidad** explicadas en esta guía son aplicables también a los desarrollos en HTML5.

REFERENCIA TÉCNICA			
Requisito / Criterio de Conformidad	Nivel	Principio	WCAG 2
Procesamiento	A	Robusto	4.1.1
Requisito de Conformidad 1: Nivel de adecuación	-	-	-
Requisito de Conformidad 4: Uso exclusivo de tecnologías de modo compatible con la accesibilidad	-	-	-
Requisito de Conformidad 5: No Interferencia	-	-	-
El cumplimiento de esta recomendación contribuye a mejorar la Independencia de dispositivo			

CSS3

La especificación **CSS3** (que como en el caso de HTML5 está aun en desarrollo) incluye nuevas características sobre las versiones de CSS 2/2.1, principalmente en la representación multimodal de las estructuras, permite desarrollaos mucho más rápidos (por la utilización de expresiones para agrupar atributos, nuevos pseudo-elementos, reglas y selectores), el código tiene un tamaño mucho menor, siendo más eficaz y rápido en interpretarse por los agentes de usuario y siendo necesario un menor número de peticiones al servidor, con el consiguiente ahorro de recursos y ancho de banda.

Metadatos

Cuando hablamos de Metadatos en este documento nos referimos solamente a aquellos elementos que se deben incluir en las páginas web (elemento “meta” de HTML) para facilitar la adición de valores semánticos básicos en las páginas, que facilitan la indexación y el posicionamiento en los buscadores. (No confundir estos metadatos a la norma técnica publicada por AGESIC para el intercambio de información¹).

Existen diferentes etiquetas² que se podrán incluir en las páginas, algunas de las cuales deberán ser específicas para cada una de ellas (siempre el título y en función del contenido las “palabras clave”) y otras genéricas para todo el portal.

Título de la página
`<title>TITULO DE LA PÁGINA </title>`

¹ Especificaciones técnicas sobre Metadatos, accesible desde la dirección de Internet de AGESIC http://www.agesic.gub.uy/innovaportal/v/2237/1/agesic/especificacion_tecnica_sobre_metadatos.html

² Mas información sobre las etiquetas meta de HTML en la dirección de internet del consorcio W3C <http://www.w3.org/wiki/HTML/Elements/meta>

Codificación de la página

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

Descripción del organismo

```
<meta name="description" content="Breve descripción del organismo"/>
```

Palabras clave

```
<meta name="keywords" content="Palabras claves de la página"/>
```

Autor del contenido de la página (no del desarrollador del portal)

```
<meta name="author" content="autor de los contenidos o nombre del organismo propietario del portal">
```

Si el portal posee algún sistema de sindicación de contenidos

```
<link rel="alternate" type="application/rss+xml" title="Noticias del ORGANISMO (formato RSS 1.0)" href="URL_DEL_RECURSO"/>
```

```
<link rel="alternate" type="application/atom+xml" title="Noticias del ORGANISMO (formato Atom)" href="URL_DEL_RECURSO"/>
```

Si el portal tiene una página de ayuda

```
<link rel="Help" href="/Help/">
```

Si el portal utiliza favicon

```
<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon"/>
```

Codificación de caracteres

La codificación de caracteres indica a los navegadores cuál es el código, el patrón de bits, usado para codificar los caracteres usados en una página web. Si no se especifica una codificación de caracteres los navegadores intentarán identificarla o usarán una por defecto que en caso de que no coincida con la codificación usada dará lugar a una representación incorrecta de los caracteres, dificultando o impidiendo la lectura de los documentos. Por este motivo es de especial importancia declarar correctamente la codificación de caracteres usada.

La codificación de caracteres recomendada y predominante para páginas web es UTF-8. Esta codificación surgió para poder representar todos los caracteres Unicode de forma que sirva para todos los caracteres especiales de cualquier idioma siendo además retrocompatible con la codificación ASCII.

En HTML 4 y XHTML esa codificación se declara de la siguiente manera:

```
<meta http-equiv='Content-Type' content='Type=text/html; charset=utf-8'>
```

En HTML5 se puede declarar mediante la forma abreviada:

```
<meta charset='utf-8'>
```

Es importante señalar que la codificación de caracteres declarada en los documentos debe coincidir con la usada para generar las páginas web, bien desde editores o desde gestores de contenido, así como con la declarada en la configuración del servidor para servir las páginas web.

Dublin Core

Es muy habitual publicar infinidad de documentos, Leyes, estadísticas, guías, normas, etc. Para facilitar la encontrabilidad de este tipo de material y en general la interoperabilidad entre modelos descriptivos dispares se ha creado el estándar **Dublin Core**³.

Con este estándar se pretende facilitar la localización de obras o parte de estas por medio de los elementos comunes dentro de la publicación de libros, artículos, leyes etc. como son el título, autor, descripción, tema, editor, tipo de recurso, fecha de publicación, idioma, formato, documentos relacionados o los derechos de autor.

```
<meta name="DC.Title" content="Título del documento">
<meta name="DC.Creator" content="Autor/es del documento">
<meta name="DC.Subject" content="Contenido del documento">
<meta name="DC.Description" content="Descripción del contenido del documento">
<meta name="DC.Publisher" content="URL_DEL_ORIGEN_DEL_RECURSO">
<meta name="DC.Date" content="AAAA-MM-DD">
<meta name="DC.Type" scheme="DCMIType" content="Text">
<meta name="DC.Format.Medium" content="text/html">
<meta name="DC.Relation.isPartOf" content="
URL_DEL_CONTENDOR_DEL_RECURSO">
<meta name="DC.Identifier" content="URL_DEL_IDENTIFICADOR_DEL_RECURSO">
<meta name="DC.Language" content="es">
<meta name="DC.Rights" content="Derechos de autor y copyright del recurso">
```

³ <http://dublincore.org/>

DOM para la manipulación del contenido

El DOM o *Modelo de Objetos de Documento* (www.w3.org/DOM/) es un interfaz (API) en un lenguaje neutro definido por el W3C para permitir que programas y *scripts* puedan acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos. En el DOM se representan los documentos como una estructura de árbol. Cada parte del documento (texto, elementos, atributos) está representada en los nodos del árbol.

Existen métodos que permiten obtener cualquier elemento y sus atributos, modificarlos, eliminarlos, navegar por el árbol desde ellos y crear nuevos nodos con nuevos elementos. Es decir, usando DOM se puede manipular de forma dinámica contenido estructurado en documentos (por ejemplo, usando lenguajes como ECMAScript) y es el medio fundamental para la creación de *scripts no intrusivos*.

Por tanto, los scripts usados para generar y/o modificar el contenido de las páginas deben basarse en el DOM o Modelo de Objetos de Documento (*DOM Scripting*) sin usar métodos propietarios de navegadores específicos y basándose, siempre que sea posible, en la especificación ECMAScript y en técnicas de *scripting no intrusivo* para separar la capa de contenido de la **capa de comportamiento**.

Actualmente, la mayoría de navegadores incluyen una implementación de ECMAScript y del Modelo de Objetos de Documento. Aunque cada navegador tenga su propia versión de lenguaje de Script (JavaScript, JScript, etc.) o añada extensiones propias a ECMAScript, todo el código realizado en el estándar ECMAScript debería funcionar correctamente en los navegadores que lo soporten.

Scripting no intrusivo

De igual forma que se separan estructura y presentación eliminando todos los elementos y atributos de presentación del código (X)HTML y sustituyéndolos por sus propiedades equivalentes en las hojas de estilo, se recomienda usar scripting no intrusivo separando la capa de comportamiento al trasladar todo el código de scripting a archivos externos.

Para lograr esta separación entre estructura (HTML), presentación (CSS) y comportamiento (scripts) se deben evitar también las llamadas a funciones de script desde eventos directamente en el código (X)HTML (*onload*, *onclick*, *onmouseover*, etc.). Si se mantuviesen estos eventos en el código (X)HTML se necesitarían cambiar todos los documentos donde se llamase a una función que se quiera renombrar. Usando el Modelo de Objetos de Documento (DOM) se puede manipular el código (X)HTML para seleccionar el elemento adecuado y generar dinámicamente el evento con la llamada a la función correspondiente.

Mantener todos los scripts totalmente independientes del código (X)HTML permite aplicarlos a cualquier página del sitio Web sin tener que modificar el código de las mismas.

Es importante señalar que los scripts deben usarse como *mejora progresiva*, es decir, que no

sean un requisito para el correcto funcionamiento de la página.

Scripting como mejora progresiva

Todo el contenido y funcionalidad básica de los documentos debe estar disponible cuando no se soporten los scripts, aunque se presenten de forma diferente a la original. Estas tecnologías deben considerarse como una mejora, no como un requisito, de forma que los documentos se muestren y funcionen correctamente cuando no estén soportadas.

Se empieza con el documento en (X)HTML plano en el que se incluye todo el contenido de forma que tenga sentido y funcione correctamente sin scripting. Sobre ese contenido se irán añadiendo progresivamente scripting, preferiblemente de forma no intrusiva, para añadir funcionalidades. Como inicialmente el contenido y la estructura es ya la correcta, si se eliminan los scripts solo se pierde la capa de comportamiento que mejoraba la usabilidad, pero el contenido original permanecerá intacto.

REFERENCIA TÉCNICA			
Requisito / Criterio de Conformidad	Nivel	Principio	WCAG 2
Nombre, función, valor	A	Robusto	4.1.2
Requisito de Conformidad 1: Nivel de adecuación	-	-	-
Requisito de Conformidad 4: Uso exclusivo de tecnologías de modo compatible con la accesibilidad	-	-	-
Requisito de Conformidad 5: No Interferencia	-	-	-

Accesibilidad de las herramientas de autor (ATAG)

Las WCAG hacen referencia a la accesibilidad del contenido de un sitio o portal web (texto, imágenes, formularios, aplicaciones, multimedia, etc.), permitiendo que las personas con discapacidad interactúen adecuadamente con la Web.

Sin embargo, en ocasiones se pone a disposición de los usuarios herramientas de edición para que también puedan aportar contenido a la Web mediante editores (principalmente WYSIWYG) en blogs, foros, redes sociales y herramientas de la Web 2.0. Estas herramientas, al igual que los editores HTML o gestores de contenido (CMS) usados por los desarrolladores para la creación de páginas web, se conocen como *herramientas de autor*.

Relativo a esta área el W3C ha desarrollado las *Pautas de Accesibilidad para Herramientas de Autor*, **ATAG**, cuya versión vigente actualmente es la 1.0 estando la versión 2.0 en fase de borrador de trabajo. Se tienen en cuenta los siguientes aspectos:

- » Que las herramientas de autor puedan ser utilizadas por personas con discapacidad. Para ello el interfaz de usuario de la herramienta de autor debe ser accesible.
- » Que la herramienta de autor permita la producción de contenido accesible. Por ejemplo, permitiendo proporcionar alternativas textuales para el contenido no textual, usando los elementos estructurales y semánticos adecuados para marcar el contenido, etc.
- » Que estas herramientas den apoyo a los autores al crear contenido accesible. Por ejemplo, mediante la integración en la herramienta de validadores de accesibilidad que comprueben el código y asistan a los usuarios en su corrección.
- » Cuando en los portales web se empleen este tipo de herramientas se deben seguir, en la medida de lo posible, las *Pautas de Accesibilidad para Herramientas de Autor* (ATAG). Haciendo que su interfaz sea accesible se permite que cualquier persona pueda usar estas herramientas. Permitiendo la producción y dando apoyo en la creación de contenido accesible se reduce el esfuerzo empleado en las tareas de monitorización y corrección del contenido contribuido, necesarias para poder hacer una declaración de conformidad.

REFERENCIA TÉCNICA			
Requisito / Criterio de Conformidad	Nivel	Principio	WCAG 2
Nombre, función, valor	A	Robusto	4.1.2
Requisito de Conformidad 1: Nivel de adecuación	-	-	-
Requisito de Conformidad 2: Páginas completas	-	-	-
Requisito de Conformidad 3: Procesos completos	-	-	-
Requisito de Conformidad 4: Uso exclusivo de tecnologías de modo compatible con la accesibilidad	-	-	-
Requisito de Conformidad 5: No Interferencia	-	-	-

Agentes de usuario

Hoy en día existen gran variedad de agentes de usuario que acceden a Internet con distintas funciones: navegadores, agregadores de contenidos, rastreadores, etc. Estos agentes de usuario tienen características diferentes, incluso aquellos que sirven para funciones similares, y el único nexo común que comparten es la Web.

Este es el principal motivo por el que, en vistas a mejorar la interoperabilidad, accesibilidad y usabilidad general de nuestros portales web, siempre se deben utilizar desarrollos que se basen en ese nexo común: los Estándares Web.

Sin embargo existen diferencias en el soporte e interpretación de los estándares por parte de los distintos agentes de usuario, tanto entre diferentes compañías como entre distintas versiones de la misma compañía.

Es responsabilidad de los desarrolladores asegurarse que la información, y en la medida de lo posible el estilo de presentación, sea consistente a través de los diferentes productos, versiones y plataformas, teniendo especial cuidado en evitar las diferencias de interpretación que puedan ocasionar una pérdida de la información o funcionalidad contenida en el sitio.

Es por tanto necesario que se realicen pruebas con distintos tipos de navegadores (Internet Explorer, Mozilla/Firefox, Chrome, Opera, Safari, etc.), distintas versiones de cada uno de los navegadores (no sólo la última versión del mercado) y distintas plataformas (Windows, Macintosh, Linux, Unix, etc.), probando no sólo las versiones finales de los portales, sino durante las distintas fases del desarrollo.

También se recomiendan las pruebas con otros tipos de agentes de usuario como navegadores modo texto (por ejemplo LYNX) y navegadores de voz o lectores de pantalla (como JAWS, Windows-Eyes o Home Page Reader).

En general se debe seguir una serie de buenas prácticas y consejos durante el desarrollo de los proyectos que ayudarán a conseguir resultados finales consistentes y fiables:

- » Utilizar siempre código (X)HTML y CSS válido.
- » No utilizar extensiones propietarias de navegadores concretos.
- » Realizar las pruebas de los desarrollos primero en navegadores que tengan un buen soporte de los estándares.
- » Conocer, identificar y corregir potenciales errores de interpretación por parte de los navegadores que fallan en el soporte de algunas características de los estándares.

REFERENCIA TÉCNICA			
Requisito / Criterio de Conformidad	Nivel	Principio	WCAG 2
Procesamiento	A	Robusto	4.1.1
El cumplimiento de esta recomendación contribuye a mejorar la Independencia de dispositivo			

Versiones alternativas accesibles

Es posible que algunos contenidos enriquecidos (multimedia) no faciliten la creación de estos contenidos en formato estándar. En estos casos debe proporcionarse un contenido alternativo accesible. Cuando se utilice esta solución se deben tener en cuenta las recomendaciones expuestas a continuación.

Siempre que no sea posible crear un contenido que cumpla con todos los criterios de accesibilidad exigidos para el nivel declarado, se deberá proporcionar un contenido alternativo que cumpla estos requerimientos, mantenga la misma información y funcionalidad y se mantenga actualizada de forma sincronizada con el contenido que no cumple los requisitos de accesibilidad. Además deberá cumplir, al menos, uno de los siguientes requisitos:

- » se puede acceder a la versión accesible del contenido desde la página que no cumple los requisitos necesarios mediante un mecanismo accesible,
- » únicamente se puede llegar a la versión no accesible del contenido desde la versión accesible de la página,
- » desde cualquier página que cumple los requisitos de accesibilidad se proporcionan mecanismos para acceder a las dos versiones de los contenidos.

Nunca se debe abusar de este recurso, siendo preferible la creación de contenidos cuyo contenido sea completamente accesible.

Debe tenerse especialmente en cuenta que una versión alternativa accesible no significa una versión únicamente textual y por lo tanto no se debe prescindir de aquellos recursos gráficos y/o multimedia presentes en la versión original.

REFERENCIA TÉCNICA			
Requisito / Criterio de Conformidad	Nivel	Principio	WCAG 2
Requisito de Conformidad 1: Nivel de Adecuación	-	-	-

Imágenes

Como norma general, todo contenido no textual debe contar con una alternativa textual equivalente que cumpla la misma función y tenga un significado equivalente al contenido original en su contexto específico.

REQUISITOS

- » Todas las imágenes incluidas mediante el elemento `` deben tener un atributo `alt`.
- » Se debe proporcionar un texto alternativo breve mediante el atributo `alt` del elemento `` para todas las imágenes, que no tengan carácter decorativo, en el que se transmita la misma información de la imagen.
- » En las imágenes con contenido textual se debe proporcionar el texto completo mostrado en la imagen como parte del texto alternativo, en el atributo `alt` del elemento ``.
- » En el caso de imágenes (``) complejas (gráficas estadísticas, planos, diagramas, etc.) en las que no sea suficiente el atributo `alt` para proporcionar la misma información, se debe proporcionar una descripción breve en el texto alternativo y una URI mediante el atributo `longdesc` o mediante un enlace en la que se encuentre la descripción detallada de la imagen.

Las descripciones largas podrán estar en la misma página en la que se encuentra la imagen o bien en una página aparte. En todo caso, si la descripción de la imagen no es el único contenido de la página, el atributo `longdesc` debe enlazar justo a la descripción correspondiente (enlace interno o ancla). También se debe indicar el final de la descripción (por ejemplo, con un texto del estilo *Fin de la descripción de la imagen ...*).

- » En el caso de imágenes en las que, por su naturaleza, no sea posible proporcionar en el texto alternativo la misma información que proporciona la imagen se debe al menos identificar su contenido. Es el caso por ejemplo de imágenes de cuadros y expresiones artísticas en general que pierden su esencia si no se perciben por el sentido para el que fueron creadas. Para identificar su contenido se puede usar el nombre que le dio el autor de la obra, el nombre comúnmente aceptado o bien, si no tiene un nombre conocido, una breve descripción que identifique su contenido.
- » Para los grupos de imágenes contiguas las cuales transmiten una información de forma conjunta en vez de individualmente se debe usar el texto alternativo de una de las imágenes del grupo para describir todo el grupo (p. ej. el texto alternativo de la primera imagen), dejando vacío el texto alternativo del resto de las imágenes. De esta forma se evitan repeticiones innecesarias en los textos alternativos del resto de imágenes del grupo.

- » No es recomendable el uso de imágenes generadas mediante caracteres ASCII (*ASCII art*, *smilies*, etc.), procurando usar imágenes convencionales en su lugar, con su correspondiente texto alternativo.
- » En caso de usar ASCII art es necesario proporcionar una descripción textual, preferiblemente antes del mismo. Si además el ASCII art es largo es recomendable proporcionar en enlace de salto antes del mismo (ancla al siguiente contenido de la página) para que las personas que usan lectores de pantalla los puedan evitar.

REFERENCIA TÉCNICA			
Requisito / Criterio de conformidad	Relativo a		Código
Contenido no textual	Principio Perceptible	1:	1.1.1

Imágenes decorativas

Las imágenes decorativas son aquellas que no aportan información y su finalidad es meramente de presentación o de maquetación.

REQUISITOS

- » Las imágenes que no aporten información cuya finalidad sea puramente decorativa deben incluirse de forma que resulten transparentes para las personas que usan lectores de pantalla.
 - Como fondo mediante las hojas de estilo
 - Como parte del contenido con el elemento `` con atributo `alt` vacío (`alt=""`) o con un espacio en blanco (`alt=" "`) y sin atributo `title` o con atributo `title` vacío (`title=""`).
- » Las imágenes cuya única finalidad es la realización de efectos de maquetación o posicionamiento de elementos, deben ser sustituidas por las propiedades de estilo correspondientes.
- » Sólo deben incluirse desde las hojas de estilo aquellas imágenes que sean puramente decorativas. Las imágenes que aportan información deben estar en el contenido de los documentos con su correspondiente texto alternativo. En caso de incluir alguna imagen que transmite información desde las hojas de estilo, esta información también ha de estar disponible en el contenido de la página en forma de texto.

Requisito / Criterio de conformidad	Relativo a	Código
Contenido no textual	Principio Perceptible	1: 1.1.1

Imágenes funcionales

Se consideran imágenes funcionales aquellas que no sólo aportan información sino que además tienen una función asociada, como la de hacer la función de un enlace o formar parte de un enlace.

REQUISITOS

- » El texto alternativo de las imágenes que son el único elemento de un enlace debe indicar el destino o función de dicho enlace.

```
<a href="url"></a>
```

- » El texto alternativo de las imágenes que acompañen a un enlace textual debe combinarse con el texto del enlace de forma que entre ambos describan el destino del enlace, evitando repeticiones innecesarias.
 - Si la imagen que acompaña al texto del enlace no aporta información adicional entonces debe tratarse como una imagen decorativa.

```
<a href="url">Texto enlace </a>
```

- Si la imagen complementa la información del texto del enlace entonces se debe proporcionar dicha información en el texto alternativo. Por ejemplo, si la imagen es un icono que indica el formato del documento (PDF, Word, etc).

```
<a href="url">Texto enlace </a>
```


- » Solamente se podrá enlazar una imagen directamente con su correspondiente ampliación (es decir, directamente con un archivo de imagen) si se avisa explícitamente en el texto alternativo que el enlace lleva a una ampliación de la misma imagen.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Contenido no textual	Principio Perceptible	1: 1.1.1
Propósito de los enlaces (en contexto)	Principio Operable	2: 2.4.4
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

CAPTCHAS

Se denomina Captcha a los test usados en computación para diferenciar entre personas y ordenadores. Captcha es el acrónimo de *Completely Automated Public Turing test to tell Computers and Humans Apart* (Prueba de Turing pública y automática para diferenciar máquinas y humanos). Por ejemplo, incluir en una imagen caracteres o palabras distorsionadas para que una persona las reconozca e introduzca con el teclado.

El problema de este tipo de test es que también impiden el acceso a las personas que no puedan ver la imagen. No es posible proporcionar una alternativa textual que proporcione la misma información que la imagen (la palabra o caracteres distorsionados) porque de esa forma estamos rompiendo el propósito del captcha. Al igual que un lector de pantalla puede leer la alternativa textual, también lo podrá hacer cualquier otro software malintencionado.

REQUISITOS

- » En las imágenes usadas como captchas es necesario proporcionar una alternativa textual que describa su propósito (p. ej. *"Introduzca las letras de la imagen"*) y, además, proporcionar a su vez otro captcha en una modalidad diferente. Por ejemplo, un captcha auditivo en el que se proporciona un audio donde se pronuncia una palabra que hay que reconocer sobre un ruido de fondo o un tests con preguntas lógicas.

Todos los captchas tienen algún problema de accesibilidad para algún grupo de personas (con discapacidades visuales, auditivas, cognitivas, etc.). Aunque es complejo cubrir todas las situaciones, se considera suficiente usar al menos dos modalidades diferentes de captchas.

REFERENCIA TÉCNICA			
Requisito / Criterio de conformidad	Relativo a		Código
Contenido no textual	Principio Perceptible	1:	1.1.1

Mapas de imágenes

Los mapas de imágenes se usan como elemento de navegación e interacción y, por tanto, es necesario proporcionar información sobre su función y permitir que todas las personas los puedan usar

REQUISITOS

- » Cada región o zona activa de los mapas de imagen de cliente tendrán una alternativa propia mediante el atributo **alt** del elemento `<area>` que describa el área concreta y su funcionalidad específica.
- » En todos los mapas de imágenes, la imagen sobre la que se crea el mapa debe llevar una alternativa textual, mediante el atributo **alt**, que describa el conjunto del mapa y su funcionalidad general.
- » Se deben usar siempre mapas de imagen de cliente y no mapas de imagen de servidor, excepto si es necesario que las zonas activas de los mapas tengan una forma de tal precisión que no fuera posible expresarla con mapas de cliente sin ocasionar un aumento considerable del tamaño del documento.
- » En caso de usar mapas de imagen de servidor, es recomendable replicar la funcionalidad de cada una de las zonas activas del mapa mediante enlaces redundantes duplicados en el mismo documento.

REFERENCIA TÉCNICA			
Requisito / Criterio de conformidad	Relativo a		Código
Contenido no textual	Principio Perceptible	1:	1.1.1
Teclado	Principio Operable	2:	2.1.1
Propósito de los enlaces (en contexto)	Principio Operable	2:	2.4.4
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página			

Audio, vídeo y multimedia

REQUISITOS

- » Para cualquier tipo de contenido multimedia, independientemente de que se trate de solo audio, solo vídeo o la combinación de ambos, grabado o en directo, se debe proporcionar al menos un texto descriptivo que identifique el contenido.
- » Cuando este tipo de contenidos se incorpora mediante el elemento `<object>`, las alternativas textuales para los mismos (imágenes, vídeos, aplicaciones, etc.) se proporcionarán dentro del cuerpo de dicho elemento, entre sus etiquetas de apertura y cierre. En el cuerpo del elemento `<object>` podemos incluir la alternativa textual o bien otro contenido alternativo no textual (imágenes, otros elementos `<object>`, etc.) que a su vez disponga de su correspondiente alternativa textual.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Contenido no textual	Principio Perceptible	1: 1.1.1

Sólo Audio

Se entiende por *sólo audio* aquel contenido tempodependiente que sólo disponen de información sonora (sin vídeo), bien como archivo de audio para reproducirlo en la web o como archivo dispuesto para su descarga (por ejemplo, archivos en formato mp3, wma, ogg, flac, ape...

REQUISITOS

- » Cuando el contenido se trate de audio pregrabado, se debe proporcionar una transcripción textual en la que se incluyan los diálogos y sonidos más significativos. En caso de que existan diálogos, debe identificarse a cada uno de los hablantes.
- » Independientemente de las alternativas equivalentes que se deben proporcionar para este tipo de contenidos, el sonido no debería comenzar sin una petición por parte de las personas que usan la web. En caso de que el sonido comience a reproducirse inmediatamente al cargar una página, debe existir algún mecanismo que permita detenerlo. Este mecanismo se proporcionará al comienzo de los documentos y será

fácilmente identificable para que sea posible detener la reproducción.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Contenido no textual	Principio Perceptible	1: 1.1.1
Sólo audio y sólo vídeo (grabado)	Principio Perceptible	1: 1.2.1
Control del audio	Principio Perceptible	1: 1.4.2
Requisito de Conformidad 5: No Interferencia	Conformidad	RC5

Sólo Vídeo

Se entiende por *sólo vídeo* a las presentaciones tempodependientes que sólo disponen de información visual (sin audio).

REQUISITOS

- » Para contenidos de vídeo pregrabado, se debe proporcionar una transcripción textual completa o una alternativa en audio que incluya las descripciones del escenario, acciones, expresiones, lenguaje corporal, etc. En general, descripciones de todo el contenido visual importante.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Contenido no textual	Principio Perceptible	1: 1.1.1
Audiodescripción o Medio alternativo (grabado)	Principio Perceptible	1: 1.2.3
Audiodescripción (grabado)	Principio Perceptible	1: 1.2.5

Multimedia – Audio y Video

Se entiende por *multimedia* toda presentación tempodependiente cuyo contenido esté

formado por información sonora y visual, independientemente de su formato y del mecanismo empleado para proporcionarlo en la página web (uso de un reproductor incrustado o mediante un enlace para su descarga).

REQUISITOS

- » El contenido multimedia pregrabado debe ir acompañado del correspondiente subtítulo para sordos, que incluye el texto sincronizado con todos los sonidos (hablados y no hablados) necesarios para comprender la presentación (diálogos, identificación de hablantes, música, risas, aplausos, sonidos significativos en general).
- » Para todo contenido multimedia pregrabado es necesario proporcionar una transcripción textual que incluya descripciones de todo el contenido visual y auditivo que sea relevante: diálogos, sonidos significativos y descripciones del contenido visual relevante para su comprensión (acciones, escenarios, lenguaje corporal, gráficos, texto mostrado, etc.).
- » Cuando sea necesario, es decir, si existe información visual relevante, se debe incluir una audio descripción o narración con información sobre los elementos visuales clave (acciones, escenarios, lenguaje corporal, gráficos, texto mostrado, etc.) sin interferir con el audio original.
- » Como caso excepcional, cuando se proporciona contenido multimedia en directo, además de la etiqueta descriptiva que identifique el contenido deben proporcionarse subtítulos del contenido de modo que ayuden a comprender la presentación.
- » Al igual que sucede con los contenidos de solo audio, el sonido no debería comenzar sin una petición expresa. En caso de que el sonido comience a reproducirse inmediatamente al cargar una página, debe existir algún mecanismo que permita detenerlo. Este mecanismo se proporcionará al comienzo de los documentos y será fácilmente identificable para que sea posible detener la reproducción.

REFERENCIA TÉCNICA			
Requisito / Criterio de conformidad	Relativo a		Código
Contenido no textual	Principio Perceptible	1:	1.1.1
Subtítulos (grabado)	Principio Perceptible	1:	1.2.2
Audiodescripción o Medio alternativo (grabado)	Principio Perceptible	1:	1.2.3
Subtítulos (directo)	Principio Perceptible	1:	1.2.4

Audiodescripción (grabado)	Principio Perceptible	1:	1.2.5
Control del audio	Principio Perceptible	1:	1.4.2
Requisito de Conformidad 5: No Interferencia	Conformidad		RC5

Destellos

Los destellos son cambios en la luminosidad relativa que, en una frecuencia elevada (3Hz o cuando se producen más de 3 veces en un segundo) pueden provocar ataques epilépticos (ataques fotosensitivos).

REQUISITOS

- » Como norma general, debe evitarse la presencia de destellos en pantalla.
- » En caso de existir destellos en el contenido de una página web, estos no serán más de 3 en un segundo. Si superan este número, lo harán en un área de tamaño restringido de la pantalla. Esta área segura corresponde a un 25% para un campo de visión de 10º a una distancia habitual de visionado.

Para facilitar el cumplimiento de este requisito se puede tomar como referencia lo que se conoce como área segura para contenido web. Esta área se ha calculado en base a los criterios anteriores pero considerados para una resolución estándar de 1024x768 píxeles y a una distancia de visión de entre 11 y 26 pulgadas. Según esto, el área segura para contenido web es cualquier forma con un área inferior a 21824 píxeles cuadrados. Esta área, traducida a un cuadrado, se corresponde aproximadamente con un cuadrado de 148 píxeles de lado. El área con destellos debe ser inferior a esta área segura para contenido web.

Importante: no se permite la posibilidad de ofrecer un control que detenga los destellos, dado que los ataques fotosensitivos pueden suceder antes de que sea posible detenerlos.

REFERENCIA TÉCNICA

Requisito / Criterio de conformidad	Relativo a	Código
-------------------------------------	------------	--------

Umbral de tres destellos o menos	Principio Operable	2:	2.3.1
Requisito de Conformidad 5: No Interferencia	Conformidad		RC5

Movimiento y Parpadeo

Los contenidos que se presentan en movimiento suponen una distracción para muchas personas en su interacción habitual con las páginas web. Asimismo, cuando este contenido en movimiento o parpadeo transmite información está imponiendo un límite de tiempo en su lectura.

REQUISITOS

- » No se deben incluir elementos parpadeantes a menos que el parpadeo se detenga por sí mismo o se proporcione un método de detención que asegure la visibilidad de la información.
- » Se deben evitar los movimientos innecesarios en las páginas web, especialmente los movimientos cíclicos que se repiten indefinidamente. No se debe incluir ningún elemento que tenga movimiento (barras de noticias *tickers*, anuncios publicitarios *banners*, animaciones...) salvo que se cumpla alguna de las siguientes condiciones:
 - Que las personas que usan la web puedan decidir si lo quiere poner en marcha o no
 - Que el movimiento se detenga por si mismo antes de 5 segundos.
 - Que se proporcione un mecanismo de detención
- » Si el contenido en movimiento proporciona información importante (noticias que se actualizan constantemente, *scroll ticker*), al detenerlo se tiene que cumplir alguna de las dos condiciones siguientes:
 - Se puede reanudar el movimiento desde el punto en el que se detuvo para permitir el acceso al resto del contenido.
 - Al detenerse, se muestra todo el contenido de forma visible permitiendo así su lectura.
- » Si se usan gifs animados de carácter decorativo, deben configurarse de modo que se detengan automáticamente antes de que pasen 5 segundos. Si estos elementos contienen información textual u otro contenido importante, deben detenerse de forma que muestren toda la información importante.
- » No se usarán los elementos `<blink>` ni `<marquee>` para efectos de parpadeo o movimiento debido a que no son elementos recogidos en la especificación de HTML.
- » En el caso de contenido de audio, vídeo y multimedia, no debe comenzar su reproducción sin una petición expresa para evitar los problemas de interferencias producidos entre este contenido y los lectores de pantalla. Si comienza de forma

automática tiene que ser posible detenerlo y volver a reiniciarlo desde donde fue detenido.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Poner en pausa, detener, ocultar	Principio Operable 2:	2.2.2
Requisito de Conformidad 5: No Interferencia	Conformidad	RC5
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

Estructura y semántica

La función principal de (X)HTML es la representación de información estructurada. Cada uno de los elementos de (X)HTML cumple una función estructural importante y tiene un valor semántico que se debe respetar y aprovechar.

REQUISITOS DE CARÁCTER GENERAL

Usar preferiblemente XHTML 1.0 en su versión *Strict* para crear las estructuras básicas de información de manera acorde a la especificación, respetando y usando la semántica de todos los elementos disponibles.

- » Asegurar que el orden de los elementos en el código sea correcto; de este modo se consigue siempre un orden lógico y semántico que tenga sentido sin depender de las características de presentación, el posicionamiento o el comportamiento de los elementos.
- » Usar los elementos que proporciona la tecnología empleada para dividir el texto en párrafos, `<p>` en (X)HTML, que se correspondan con cada una de las ideas básicas que integran el documento.
- » Emplear la capacidad de expresión semántica que ofrezcan las tecnologías de contenido web al crear documentos para la Web, prestando especial atención al usar los elementos basándose en su significado y no en su función visual o de formato.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible	1: 1.3.1
Secuencia significativa	Principio Perceptible	1: 1.3.2
Encabezados y etiquetas	Principio Operable	2: 2.4.6
Procesamiento	Principio Robusto	4: 4.1.1
Requisito de Conformidad 5: No Interferencia	Conformidad	RC5
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo		

Encabezados

REQUISITOS

- » Usar textos claros e informativos en los encabezados para definir las distintas secciones de los documentos, sin emplear encabezados para definir secciones sin contenido o con un contenido mínimo, sin entidad suficiente para formar una nueva sección del documento.
- » Usar los elementos de encabezado `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` hasta el nivel de profundidad adecuado para proporcionar una representación fiel de la estructura de los documentos y su jerarquía.
- » Respetar el anidamiento de los encabezados, evitando incluir un encabezado sin que estén presentes los correspondientes encabezados de niveles superiores (por ejemplo no poner un `<h4>` sin que haya antes `<h1>`, `<h2>` y `<h3>`).
- » Todos los documentos deberán tener siempre, al menos, un elemento de encabezado principal `<h1>` puesto que todo documento debe tener una estructura mínima.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible	1: 1.3.1
El cumplimiento de esta recomendación contribuye a mejorar la independenciade dispositivo		

Listas

REQUISITOS

- » Usar los elementos adecuados de HTML para identificar las series o enumeraciones de elementos relacionados entre sí: listas ordenadas ``, listas desordenadas `` o listas de definición `<dl>`, según corresponda.
- » En las listas siempre debe haber al menos un elemento (``, `<dt>`, `<dd>`), es decir, no están permitidas construcciones contenedor del tipo: `Texto`.
- » Cuando se usen imágenes como viñetas o marcadores de listas, estas deben incluirse mediante las hojas de estilo, ya sea como imágenes de lista (`list-style-image`) o fondos de elementos de lista (`background-image`). No se deben simular listas usando elementos distintos a ``, ``, `<dl>`, como por ejemplo, guiones, asteriscos, números y letras (-, *, 1), a), IV).

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible	1: 1.3.1
El cumplimiento de esta recomendación contribuye a mejorar la Independencia de dispositivo		

Datos tabulares

REQUISITOS

- » Los títulos de las tablas deben marcarse mediante el elemento `<caption>` y no con un elemento de encabezado (`<th>`) que englobe toda la tabla.
- » Identificar todos los encabezados de fila y de columna mediante el elemento `<th>`, y todos los datos de la tabla (`<td>`) deben tener la posibilidad de encontrarse asociados a un encabezado que lo defina (no se usarán encabezados vacíos).
- » Expresar las relaciones entre los datos (`<td>`) y sus encabezados (`<th>`) mediante los atributos `id` y `headers`, ya que tienen una mejor implementación y soporte en los productos de apoyo, pudiendo, en el caso de las tablas con encabezados simples, usar adicionalmente el atributo `scope` con la misma finalidad.
- » En las tablas de datos complejas, proporcionar una descripción del tipo de contenido y las relaciones entre los distintos datos de la tabla mediante el atributo `summary` del elemento `<table>`.
- » En caso de proporcionar un resumen debe ser para dar información detallada y no es válido repetir el contenido del elemento `<caption>` en el atributo `summary`.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible	1: 1.3.1
El cumplimiento de esta recomendación contribuye a mejorar la Independencia de dispositivo		

Énfasis estructural

REQUISITOS

- » Identificar el énfasis mediante los elementos semánticos `` (énfasis normal) y

**** (énfasis acentuado) para enfatizar la información.

- » Evitar el uso de propiedades físicas, como ****, **<u>** e **<i>** para dar resaltar la información ya que, si bien proporcionan el mismo aspecto visual no tienen el mismo carácter semántico.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible	1: 1.3.1

Citas

REQUISITOS

- » Identificar las citas formadas por bloques de texto mediante el elemento **<blockquote>**.
- » Identificar las citas en línea, dentro de bloques de texto, mediante el elemento **<q>**.
- » Usar, siempre que sea posible, el atributo **cite** de los elementos **<blockquote>** y **<q>** para ofrecer una referencia útil (URI) sobre dichas citas.
- » Identificar las referencias a fuentes externas mediante el elemento **<cite>**.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible	1: 1.3.1

Presentación

REQUISITOS GENERALES

- » Usar siempre CSS como medio para añadir las características de presentación y maquetación necesarias.
- » No se deben introducir elementos ni atributos de presentación en el lenguaje de marcado de los sitios web (tales como ``, `<center>`, `color`, etc.) usando siempre las hojas de estilo en cascada (CSS) para definir las características de presentación.
- » No se deben usar los elementos de HTML únicamente para proporcionar características de presentación al texto, prescindiendo de la semántica de dichos elementos.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible	1: 1.3.1

Significado mediante características sensoriales

REQUISITOS

- » No se debe añadir significado a los elementos usando como único medio de distinción propiedades físicas de presentación, tales como pueden ser el color, el tamaño de letra o la disposición de un elemento en el documento. No es necesario prescindir de esas características de presentación, que pueden ser un buen refuerzo de la información, sino que se debe añadir una forma adicional de diferenciación que no dependa de características físicas. Por ejemplo, campos obligatorios de un formulario marcados en rojo y con la palabra "*obligatorio*" entre paréntesis.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Características sensoriales	Principio Perceptible	1: 1.3.3
Uso del color	Principio	1: 1.4.1

	Perceptible	
--	-------------	--

Contraste

REQUISITOS

- » El contraste entre el color del texto (que esté destinado a leerse) y el color de fondo de los diferentes elementos (texto, texto en imágenes) debe ser el suficiente para garantizar su correcta percepción. Este nivel de contraste varía dependiendo del tamaño del texto:
 - **Texto normal** (menor de 18pt o 14pt y negrita): contraste de al menos 4.5:1
 - **Texto grande** (menor de 18pt o 14pt y negrita): contraste de al menos 3:1

Para evaluar el contraste se usará el nuevo algoritmo de luminosidad recomendado por el W3C. En este algoritmo el contraste es el ratio entre la luminosidad del color de primer plano (L1) y el fondo (L2): L1/L2.

- » Como caso excepcional, están excluidos de cumplir los requisitos de contraste los textos con carácter decorativo, los textos que se muestran de forma accidental en imágenes y que no estén destinados a leerse y los que formen parte de logotipos o imagen de marca (las guías de estilo corporativas no están exentas de cumplir los niveles de contraste).
- » Las imágenes incluidas como fondo de los contenidos no deben impedir la correcta legibilidad de los mismos, usando preferiblemente fondos planos o fondos con tramas muy simples y altamente difuminados.
- » Para evitar efectos de combinación de colores no deseados que pueden dar lugar a textos no legibles, debido a las propiedades de las hojas de estilo y su funcionamiento de herencia y cascada, es recomendable que, cuando se defina un color de fondo o de primer plano (`background-color` y `color`, para CSS) siempre se haga en pareja. Es decir, que siempre se definan ambas propiedades de color para el elemento en cuestión, evitando los valores "inherit" y "transparent" que por sus características darían lugar a los mismos problemas.
- » Si el contraste entre el texto y el color de fondo no es el suficiente en la presentación por defecto se podrá usar una presentación alternativa (hoja de estilo alternativa) que cumpla los requisitos de contraste. El control usado en la presentación original para cambiar a la versión alternativa deberá ser accesible y tener un contraste

suficiente.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Contraste (mínimo)	Principio Perceptible	1: 1.4.3

Redimensión del texto

REQUISITOS

- » El tamaño de los textos (*font-size*) no podrá expresarse en valores absolutos, usando siempre en su lugar unidades relativas para facilitar su redimensionamiento (*%*, *em* ó *ex*). Esta regla también se aplicará a todas aquellas características que afecten directamente a los textos, tales como la altura de línea (*line-height*), el espaciado entre letras (*letter-spacing*) y la indentación (*text-indent*) para que mantengan la coherencia cuando se produzca el redimensionamiento.
- » La maquetación del sitio y los contenedores de texto debe adaptarse adecuadamente a los diferentes tamaños del texto, de forma que este se pueda aumentar al menos hasta un 200% sin que se produzcan desbordamientos o solapamientos del contenido que dificulten o impidan su percepción.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Cambio de tamaño de texto	Principio Perceptible	1: 1.4.4
El cumplimiento de esta recomendación contribuye a mejorar la Independencia de dispositivo		

Imágenes de texto

REQUISITOS

- » La información textual debe presentarse en forma de texto (XHTML+CSS), pudiendo introducir texto en imágenes únicamente bajo determinadas circunstancias aisladas y sin perjuicio de tomar las medidas necesarias para proveer alternativas adecuadas que garanticen la accesibilidad final. Algunas de estas excepciones pueden ser:

- La presentación visual deseada para el texto no se puede lograr mediante las tecnologías empleadas (CSS). Por ejemplo, encabezados de sección que necesiten mantener una tipografía corporativa o acorde al *look and feel* general del sitio y no reproducible mediante (X)HTML+CSS.
 - Imágenes que contengan texto como parte integral de la imagen. Por ejemplo, las imágenes publicitarias.
 - Imágenes en las que la presentación del texto es esencial, como el texto o eslóganes en los logotipos o nombres de marca. También en imágenes de textos impresos o manuscritos cuya presentación original forma parte de su valor documental, en imágenes de ejemplo de diferentes tipografías, etc.
- » En el caso de usar imágenes de texto que no entran dentro de las excepciones permitidas entonces, en vez de incluir las imágenes directamente en el código (X)HTML, es necesario usar alguna técnica de reemplazo de texto por imágenes de texto y permitir que las personas que usan la web puedan cambiar, mediante un control, entre ambas presentaciones según sus necesidades. Es necesario incluir este control para cambiar entre ambas presentaciones por la dificultad de que estas técnicas de reemplazo sean completamente accesibles (compatibles con productos de apoyo sin que se pierda información si los scripts, hojas de estilo o imágenes, o una combinación de las anteriores, se encuentran desactivadas).

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Imágenes de texto	Principio Perceptible 1:	1.4.5

Marcos (Frames e Iframes)

REQUISITOS

- » Debe evitarse el uso de todo tipo de marcos, ya que su uso presenta varios problemas, tanto de accesibilidad como en otros campos, y su funcionalidad es fácilmente replicable mediante otras técnicas hoy en día.
- » Debe usarse el atributo **title** para indicar claramente la función de cada marco de manera que sean fácilmente identificables y navegables (no serán admisibles títulos como "*marcoSuperior*", "*marcoInferior*", "*marcoPrincipal*", etc.)
- » Es recomendable usar también un nombre significativo para el atributo **name**, siempre que la gramática que utilizemos nos lo permita, por compatibilidad con los agentes de usuario/a que utilicen este atributo en lugar de **title**.

- » En el caso de los marcos creados mediante `<frameset>` se debe añadir un elemento `<noframe>` donde se proporcione una forma alternativa de acceder a todo el contenido que proporcionan los marcos de origen (por ejemplo un índice de enlaces desde el cual se pueda acceder a todo su contenido).
- » En los marcos en línea o marcos incrustados en las páginas mediante los elementos `<iframe>` también debe proporcionarse el mismo tipo de alternativa, pero en este caso se hará como contenido del propio elemento (entre las etiquetas de apertura y cierre).
- » No deben usarse marcos con el único propósito de realizar técnicas de redirección o sobreescritura de URL (mantener siempre visible la misma URL para todo el sitio), ya que esta técnica ocasiona varios problemas de accesibilidad: desorientación, alteración del historial de navegación y uso de marcadores (*favoritos*).

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Contenido no textual	Principio Perceptible	1: 1.1.1
Evitar bloques	Principio Operable	2: 2.4.1
Titulado de páginas	Principio Operable	2: 2.4.2
Nombre, función, valor	Principio Robusto	4: 4.1.2
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

Tablas de maquetación

REQUISITOS

- » Debe evitarse el uso de tablas de maquetación empleando en su lugar una maquetación mediante hojas de estilo. La finalidad de las tablas es la de proporcionar información tabular y no la de maquetar los contenidos, práctica que además genera problemas de accesibilidad.
- » En caso de usar tablas para maquetar estas deben crearse de forma que tengan sentido cuando se acceda a ellas de forma lineal (orden del código). Por ejemplo, deben tener sentido cuando se lea su contenido con un lector de pantalla. Adicionalmente, nunca debe abusarse de este recurso teniendo especial cuidado en evitar el anidamiento continuo de tablas para conseguir efectos de maquetación sin introducir un número elevado de tablas con el mismo fin.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible	1: 1.3.1
Secuencia significativa	Principio Operable	2: 1.3.2

Foco visible

REQUISITOS

- » Se debe asegurar que el indicador del foco es visible en cualquier interfaz de usuario/a que sea operable mediante teclado. Esto facilita el uso de las páginas web mediante el teclado al permitir reconocer visualmente y en todo momento cuál es el componente con el cual se está interactuando. Por tanto, no se debe modificar el indicador de foco por defecto (usado por los navegadores) y, en caso de que tengamos que modificarlo, debemos hacerlo únicamente para mejorar su visibilidad.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Foco visible	Principio Operable	2: 2.4.7
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

Legibilidad

Encabezados y etiquetas

REQUISITOS

- » Los encabezados para ser adecuados deben identificar la sección de contenido que encabezan y ser claros y breves.
- » También es necesario proporcionar etiquetas descriptivas para los elementos de interacción de la página, como los enlaces, botones, controles de formulario, componentes personalizados (scripts, flash), etc. Las etiquetas deben identificar el propósito de los elementos de interacción de la página de forma clara.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Encabezados y etiquetas	Principio Operable 2:	2.4.6
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

Vocabulario neutro

REQUISITOS

- » No se emplearán términos ni se hará referencia a elementos de la página web únicamente en función de su forma, tamaño, posición, orientación o sonido, sino también de forma que la información proporcionada no dependa únicamente de características sensoriales. Esto es necesario porque para determinadas discapacidades no tienen sentido mensajes basados en características sensoriales. Por ejemplo, al hacer referencia a un botón de formulario se puede mencionar su función, o la etiqueta que lo describe, en vez de dar únicamente información sensorial como su forma o color.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Características sensoriales	Principio 1:	1.3.3

	Perceptible	
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

Idiomas

Los lectores de pantalla y sintetizadores que reproducen varios idiomas tienen la capacidad de usar el acento y la pronunciación adecuados para cada idioma. De igual forma, los agentes de usuario/a y navegadores gráficos pueden mostrar los caracteres y sistema de escritura (alfabeto, signos de puntuación, etc.) de forma correcta según el idioma.

REQUISITOS

- » Se debe declarar el idioma principal o idioma por defecto de cada página web en el elemento `<html>` y no en el elemento `<body>` ni mediante **Content-Language**.

Cuando un documento ofrezca contenidos en más de un idioma, se han de seguir los siguientes criterios para establecer cuál es el idioma principal:

- Si todos los idiomas están en la misma proporción, se identificará como idioma principal el que primero se encuentre en el documento.
- Si no están en la misma proporción, se identificará como principal el del grueso del contenido o el que más frecuencia presente.
- » Usar siempre los atributos **lang** y/o **xml:lang** para indicar el idioma principal del documento y cualquier cambio respecto al mismo (incluyendo el texto en imágenes, las alternativas textuales, etc.) según las siguientes normas:
- » En HTML usar el atributo **lang** exclusivamente.

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

- » En XHTML 1.0 servido como **text/html** usar los atributos **lang** y **xml:lang** simultáneamente.

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

- » En XHTML 1.0 y 1.1 servido como XML usar el atributo **xml:lang** exclusivamente.

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



Una posible excepción a esta regla es el caso de los idiomas en nombres propios, direcciones, términos técnicos, etc., en los que en ocasiones puede ser más adecuado dejar el cambio idiomático sin especificar.

Para la identificación de los idiomas en los documentos (X)HTML se usarán los códigos de idioma del Registro de etiquetas de idioma del IANA: <http://www.iana.org/assignments/language-subtag-registry>

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Idioma de la página	Principio 3: Comprensible	3.1.1
Idioma de las partes	Principio 3: Comprensible	3.1.2
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

Navegación en interacción

Enlaces y mecanismos de navegación

REQUISITOS

- » Deben usarse enlaces y controles de formulario estándar ya que permiten la navegación mediante el teclado y productos de apoyo.
- » El texto de los enlaces tiene que ser breve y conciso, identificando con claridad cuál es su destino. Debe tener significado por sí mismo o, de no ser así, debe tener significado gracias a su contexto inmediato o a un título descriptivo (atributo **title**).
- » Si hay dos o más vínculos en una página que comparten el mismo texto, todos estos vínculos deben remitir al mismo recurso. En caso contrario, deben diferenciarse bien en el propio texto de los enlaces ocultando parte de los mismos mediante técnicas accesibles de ocultación en CSS, bien mediante el atributo **title** para aclarar el recurso al cual hacen referencia, o bien mediante su contexto inmediato.
- » En general, cualquier elemento de interacción que tenga la misma funcionalidad dentro de un conjunto de páginas web se debe identificar de manera consistente. Así, se deben usar las mismas etiquetas para los mismos controles de formulario, el mismo texto para enlaces al mismo destino, textos consistentes para enlaces a destinos similares o las mismas alternativas textuales para los iconos que tengan la misma funcionalidad.
- » Cualquier texto que se utilice como enlace de navegación principal a una sección permanente del portal web deberá mantenerse igual a lo largo de todo el portal.
- » Los mecanismos de navegación que se repiten en múltiples páginas web dentro de un conjunto de páginas web, deben aparecer siempre en el mismo orden relativo cada vez que se repiten.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Propósito de los enlaces (en contexto)	Principio Operable 2:	2.4.4
Navegación coherente	Principio Comprensible 3:	3.2.3

Identificación coherente	Principio Comprensible	3:	3.2.4
Nombre, función, valor	Principio Robusto	4:	4.1.2
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página e independencia de dispositivo			

Título de páginas

REQUISITOS

- » Debemos proporcionar títulos descriptivos para las páginas web mediante el elemento `<title>`. Los títulos deben ser breves e identificar el contenido de la página web. Los títulos no pueden ser genéricos para todo el portal o sitio web.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Titulado de páginas	Principio Operable	2: 2.4.2
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo		

Opciones de navegación

REQUISITOS

- » Debemos permitir que los usuarios puedan navegar por el sitio web de diferentes maneras según sus necesidades y preferencias. Para algunas personas, como los usuarios de lectores o magnificadores de pantalla, es más fácil usar funciones de búsqueda en vez de recorrer largos menús de navegación. De igual forma, otras preferirán usar tablas de contenidos o el mapa del sitio en lugar de usar los menús de navegación.
- » A nivel de sitio web, además de la navegación principal por el sitio, es necesario proporcionar un mapa web y/o un buscador global.

Las condiciones que debe cumplir el mapa web para ser correcto son las siguientes:

- Incluir enlaces a las diferentes secciones del sitio web.

- Debe estar enlazado, al menos, desde cada una de las páginas que se listan en el mapa web.
- Todas las páginas del sitio web deben ser accesibles, directa o indirectamente, desde el mapa web.
- La presentación y estructura del mapa web debe reflejar la organización del sitio.

La función de búsqueda, útil en sitios web de gran tamaño, permite localizar los contenidos sin necesidad de comprender o navegar por su estructura. Si es posible, se ofrecerá la posibilidad de una búsqueda avanzada (p. ej. por secciones, por tipologías de contenidos, por sinónimos y similitud, con corrección ortográfica, etc.)

- » En los documentos extensos, bien en una misma página o divididos en varias páginas relacionadas entre sí, es necesario incluir una tabla de contenidos con enlaces que permitan acceder a las distintas secciones del mismo.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Múltiples vías	Principio Operable 2:	2.4.5
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

Información sobre accesibilidad

Debemos incluir una sección de accesibilidad (normalmente en una página web dedicada exclusivamente a ello) en la que se proporcionará la información indicada en el apartado "**Declaración de conformidad**". También se podrá informar sobre las funcionalidades adicionales de accesibilidad si los hay: atajos de teclado disponibles, estilos de alto contraste, estilos de texto ampliados, etc.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Múltiples vías	Principio Operable 2:	2.4.5

Evitar bloques de material repetido

REQUISITOS

- » Agruparemos e identificaremos los conjuntos de enlaces relacionados (por ejemplo, mediante el uso de listas ``, `` o `<dl>`). Esto permite que los lectores de pantalla reconozcan los bloques de enlaces (una lista) y ofrezcan la posibilidad de evitarlos para acceder directamente al siguiente contenido, siendo especialmente útil en los menús de navegación que se repiten en múltiples páginas del sitio web.
- » De igual forma, y por el mismo motivo, debemos identificar las diferentes secciones de las páginas mediante encabezados (`<h1>`, `<h2>`, `<h3>`, ..., `<h6>`) ya que los lectores de pantalla puede saltar de encabezado a encabezado sin necesidad de leer todo el contenido de la página.
- » Adicionalmente, y con la misma finalidad, podemos proporcionar enlaces de salto entre los distintos bloques de información (por ejemplo, entre las distintas secciones indicadas por los encabezados) mediante enlaces internos dentro del documento ("*Saltar navegación*", "*Ir al contenido principal*", etc.).

REFERENCIA TÉCNICA			
Requisito / Criterio de conformidad	Relativo a		Código
Información y relaciones	Principio Perceptible	1:	1.3.1
Evitar bloques	Principio Operable	2:	2.4.1
Encabezados y etiquetas	Principio Operable	2:	2.4.6
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página			

Independencia de dispositivo

REQUISITOS

- » Debemos asegurarnos de que todos los elementos de navegación pueden ser seleccionados y activados a través del teclado o mediante las ayudas técnicas.
- » Debemos evitar que los contenidos que se encuentran presentes en las páginas web (multimedia, ventanas modales, etc.) bloqueen el teclado impidiendo que se pueda salir. Es decir, todo elemento que obtenga el foco del teclado debe permitir continuar la navegación con teclado por el resto de la página.

- » Es necesario asegurar que la secuencia lógica de tabulación por defecto es adecuada y coherente. Si no es así, deberemos establecer una nueva secuencia de tabulación mediante el uso del atributo **tabindex**, asegurándonos de que la nueva secuencia definida es completa y no deja ningún elemento tabulable de la página fuera de la secuencia.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Teclado	Principio Operable 2:	2.1.1
Sin trampas para el foco del teclado	Principio Operable 2:	2.1.2
Orden del foco	Principio Operable 2:	2.4.3
Requisito de Conformidad 5: No Interferencia	Conformidad	RC5
El cumplimiento de esta recomendación contribuye a mejorar la Independencia de dispositivo		

Ventanas emergentes

REQUISITOS

- » Debemos evitar la navegación mediante la apertura de ventanas emergentes (o diálogos modales), teniendo siempre en cuenta las siguientes consideraciones:
 - Nunca se abrirá una ventana emergente de forma automática cuando algún elemento de interacción recibe el foco o cuando se cambia su valor o estado. Por ejemplo, no se abrirán ventanas cuando se seleccione una casilla de verificación o un botón de radio, cuando se rellene un campo de edición o se escoja una opción de un menú de selección.
 - La apertura de nuevas ventanas únicamente debe producirse como respuesta a una solicitud de acción por parte del usuario, como seleccionar un enlace o activar un botón de envío de un formulario.
 - Avisar siempre, bien de manera explícita (indicándolo textualmente en el enlace o botón o antes del mismo) o bien de manera implícita (mediante el atributo **title** del enlace o botón que provoca la apertura).

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código

Teclado	Principio Operable	2:	2.1.1
Propósito de los enlaces (en contexto)	Principio Operable	2:	2.4.4
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo			

Datos de entrada

Etiquetado de los controles

- » Todos los controles de formulario, a excepción de los de tipo "button", "hidden", "image", "reset" y "submit", utilizarán una etiqueta `<label>` asociada explícitamente con su correspondiente control (atributos `id` y `for`).

```
<label for="ident">Etiqueta: </label>
<input id="ident" ... />
```

- » Como norma general en el diseño de interfaces de usuario/a, las etiquetas de los campos siempre irán antes que el campo al que hacen referencia, a excepción de los "checkbox" y "radio", en los que las etiquetas irán a continuación.
- » Los botones de envío de la información deberán ser siempre de tipo "submit" y los de borrado, si existen, de tipo "reset". Si se usa un botón de envío de tipo "image" siempre deberá contar con su respectiva alternativa textual a través del atributo `alt`.
- » En aquellos casos en los que el uso de una etiqueta pueda causar confusión o no se pueda emplear un texto para el elemento `<label>` podremos usar el atributo `title` para identificar su propósito. Si no hay un elemento `<label>` los lectores de pantalla, y los agentes de usuario/a en general, pueden leer o mostrar el valor del atributo `title`. Por ejemplo, campo de búsqueda que carece de una etiqueta textual visible.

```
<input type="text" id="search" title="Buscar" />
```

Requisito / Criterio de conformidad	Relativo a	Código
Alternativas textuales	Principio Perceptible	1: 1.1.1
Información y relaciones	Principio Perceptible	1: 1.3.1
Encabezados y etiquetas	Principio Operable	2: 2.4.6
Etiquetas o instrucciones	Principio Comprensible	3: 3.3.2
Nombre, función, valor	Principio Robusto	4: 4.1.2
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo		

Agrupación de elementos

- » Si hay dos o más categorías de datos distintas (por ejemplo datos personales y datos laborales), debemos agrupar los datos pertenecientes a cada categoría dentro de un elemento `<fieldset>` e identificaremos cada uno de los grupos mediante el elemento `<legend>`.
- » En los listados de opciones desplegadas que se presentan mediante los elementos `<select>` de los formularios, siempre que necesitemos categorizar las opciones en distintos grupos usaremos el elemento `<optgroup>` para indicar cada una de las categorías.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible	1: 1.3.1
Encabezados y etiquetas	Principio Operable	2: 2.4.6
Etiquetas o instrucciones	Principio Comprensible	3: 3.3.2
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo		

Información sobre errores, ayuda y pasos

- » Toda la información relativa al uso de los distintos campos de los formularios (campos obligatorios, instrucciones para rellenar los datos, errores al rellenar algún campo, etc.) debemos procurar que esté contenida en el documento de forma textual antes que el campo en cuestión.
- » Se debe informar de forma textual, y al comienzo del documento, de los errores que se han producido al rellenar un formulario. Si es posible, propia descripción del error debería ofrecer acceso, mediante un enlace, al campo en el que se ha detectado el fallo. De esta forma, las personas que usan lectores de pantalla, podrán identificar y localizar fácilmente los errores.

De forma adicional a la identificación textual de los errores, ésta se podrá complementar mediante el uso de imágenes, colores y validaciones en el lado del cliente, de modo que se proporcionen, por ejemplo, alertas acerca de los campos erróneos.

- » Si el error fuera provocado por el tipo de datos o formato requerido (fecha en formato `dd-mm-aa`, `dd/mm/aaaa`), deberá proporcionarse, además de información sobre la localización del error, información suficiente para completar el campo con la información adecuada. Por ejemplo, indicar que un campo *teléfono* debe introducirse sin espacios entre los dígitos (`999123456` en lugar de `999 123 456`).
- » Siempre que sea posible, se proporcionarán sugerencias que permitan corregir los errores producidos cuando estos se detectan de forma automática. Por ejemplo, si en un campo *País* el nombre introducido no se reconoce, sugerir el nombre de país más parecido al introducido.
- » En los formularios complejos que soliciten una cantidad importante de información es recomendable dividir el proceso en varios pasos que nos faciliten la tarea (preferiblemente no más de 4 o 5). En estos casos proporcionaremos siempre información sobre el paso en el que nos encontramos y daremos la posibilidad de volver a los pasos anteriores.
- » Se debe mantener la información ya proporcionada en los formularios de varios pasos cuando se retrocedo o avanza por los diferentes pasos. De igual forma, se mantendrá la información ya proporcionada cuando se detecten errores de validación y se muestre el formulario para su corrección.
- » Debemos prestar especial atención cuando los formularios hagan uso de información sensible para las personas, como pueden ser transacciones económicas (compras online, contratación de servicios, impuestos, multas, pago de tasas, etc), que conlleven la transmisión de datos (creación, edición, borrado de cuentas personales) o tengan implicaciones legales. Si este tipo de transacciones tienen lugar inmediatamente y no se pueden deshacer, un error en la introducción de datos puede tener importantes y costosas consecuencias.

Por tanto, este tipo de transacciones deberían ser reversibles ofreciendo un periodo

de tiempo durante el cual se pueda cancelar o cambiar la acción realizada, informando sobre el tiempo disponible para ello y describiendo el proceso necesario. En caso de que por la naturaleza de la transacción ésta no pueda ser cancelada, entonces debemos al menos dar la posibilidad de revisar la información antes de enviarla para la corrección y/o confirmación de los posibles errores detectados.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Identificación de errores	Principio Comprensible	3: 3.3.1
Etiquetas o instrucciones	Principio Comprensible	3: 3.3.2
Sugerencias para errores	Principio Comprensible	3: 3.3.3
Prevención ante errores (legales, financieros, datos)	Principio Comprensible	3: 3.3.4
Nombre, función, valor	Principio Robusto	4: 4.1.2
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo		

Contenido dinámico

Introducción

Se puede considerar contenido dinámico a todo aquel que cambia o se modifica automáticamente a través del tiempo o en respuesta a determinadas acciones o eventos provocados por las personas que usan la web.

REQUISITOS

- » Como norma general, todas las tecnologías usadas para generar contenido dinámico (scripts, applets, Flash, etc.) han de emplearse de forma compatible con la accesibilidad. Es decir, deben generar contenido accesible e interactuar correctamente con las ayudas técnicas o productos de apoyo.

En caso contrario, si las tecnologías no son compatibles con la accesibilidad, o no se emplean de forma compatible con la accesibilidad, la página no puede depender de ellas para su correcto funcionamiento y es necesario proporcionar una alternativa accesible.

Todas las alternativas accesibles deben estar sincronizadas correctamente con los contenidos correspondientes. Es decir, las alternativas accesibles deben actualizarse a la vez que los contenidos dinámicos.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Requisito de Conformidad 4: Uso exclusivo de tecnologías de modo compatible con la accesibilidad	Conformidad	RC4
El cumplimiento de esta recomendación contribuye a mejorar la Independencia de dispositivo		

Scripts

La mayor parte de los agentes de usuario/a disponibles en el mercado, así como los productos de apoyo que usan las personas con algún tipo de discapacidad, tienen un adecuado soporte para las tecnologías de script. Por este motivo, cuando estas tecnologías se usan en el sitio web de modo compatible con la accesibilidad no es necesario ofrecer una alternativa equivalente y accesible a su contenido y/o funcionalidad.

A pesar de todo, y dado que es imposible que se presente una homogeneidad tecnológica para todas las personas (equipamientos obsoletos, conexiones reducidas, etc), se recomienda que esta funcionalidad y/o contenido no se pierda en caso de que no exista soporte para las tecnologías de script. Es decir, se recomienda proporcionar una alternativa

accesible para el caso en que no se soporten los scripts. Estas tecnologías deberían considerarse como una mejora, no como un requisito, de forma que los documentos se muestren y funcionen correctamente cuando no estén soportadas.

REQUISITOS

- » Si los scripts se usan de forma no compatible con la accesibilidad, no generan contenido accesible o no es posible la interacción con ayudas técnicas o productos de apoyo, entonces estamos obligados a ofrecer una alternativa equivalente y accesible a su contenido y/o funcionalidad.

En cuanto a las alternativas, existen problemas de compatibilidad del elemento `<noscript>` con algunos productos de apoyo. Algunos lectores de pantalla no leen el contenido de este elemento, incluso cuando no existe soporte de javascript. Por tanto, debemos evitar usarlo para asegurar el correcto funcionamiento de los documentos empleando *scripting no intrusivo* y *mejora progresiva* en su lugar.

- » En cualquier caso, tanto si los scripts se usan de forma compatible con la accesibilidad como si no, debemos tener especial cuidado con que los scripts usados funcionen de manera consistente a través de los distintos navegadores, plataformas, etc. usando únicamente el *Modelo de Objetos de Documento (DOM)*, evitando funciones propietarias de un determinado navegador y comprobando que no se presenten inconsistencias para las funciones implementadas.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Requisito de Conformidad 1: Nivel de adecuación	Conformidad	RC1
Requisito de Conformidad 4: Uso exclusivo de tecnologías de modo compatible con la accesibilidad	Conformidad	RC4
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo		

Independencia de dispositivo

REQUISITOS

- » Los eventos usados en las funciones de scripting no deben estar ligados a dispositivos concretos y siempre deben poder lanzarse desde teclado. Por ejemplo, debemos usar siempre que podamos eventos lógicos (no ligados a dispositivos) como `onsubmit`, `onreset`, `onfocus`, `onblur` y `onload`.

Se consideran como excepciones aquellas acciones que no se pueden controlar mediante teclado de forma razonable (por ejemplo, un programa de dibujo a mano alzada).

Estas excepciones tienen que ver con la función subyacente y no con la técnica usada para introducir datos. Puede haber aplicaciones que su forma principal de interacción sea mediante ratón, pero eso no impide que también se puedan implementar para ser operadas mediante teclado. Por ejemplo, la función subyacente de un programa de reconocimiento de escritura a mano alzada es la entrada de texto y se debe poder realizar de forma alternativa mediante teclado (real o en pantalla).

- » En caso de usar eventos dependientes de dispositivo es necesario replicar dichos eventos con eventos equivalentes para otros dispositivos. Así, por ejemplo usaremos siempre en conjunto:
 - **onmouseover** y **onfocus**
 - **onmouseout** y **onblur**
 - **onmousedown** y **onkeydown**
 - **onmouseup** y **onkeyup**

Aunque **onclick** sea, en principio, un manejador de evento de ratón la mayoría de los navegadores (X)HTML lo interpretan como el manejador de evento de acción por defecto para enlaces y botones y se activa tanto con el ratón como con el teclado, siendo así independiente de dispositivo.

Por tanto, evitaremos replicar el evento **onclick** con **onkeypress** ya que, además de no ser necesario, esta práctica puede tener efectos no deseados en la tabulación con teclado.

- » No debemos usar los eventos **ondblclick** y **onmousemove** para ningún uso que afecte al contenido o la funcionalidad, ya que son eventos totalmente dependientes de dispositivos de apuntamiento (ratón, *trackball*, etc.) y no tienen equivalentes para otros dispositivos, como el teclado, por ejemplo.
- » Finalmente, si no es posible lograr la misma forma de interacción mediante eventos de teclado, debemos proporcionar un mecanismo redundante basado en el teclado para lograr la misma funcionalidad. Por ejemplo, en una aplicación que use la técnica *arrastrar y soltar* podemos usar, como mecanismo redundante de teclado, la técnica *cortar y pegar*. Asimismo, si *arrastrar y soltar* se usa, por ejemplo, para relacionar elementos entre sí, podemos replicar la misma funcionalidad mediante enlaces, botones u otros controles estándar operables con teclado, como menús de selección para escoger el elemento adecuado.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Teclado	Principio Operable	2: 2.1.1
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la		

Scripts en formularios

REQUISITOS

- » Cuando se empleen eventos asociados a controles de formulario o, de forma general, a cualquier componente de interfaz de usuario/a, que conlleven la ejecución de cualquier acción, evitaremos que provoquen un cambio de contexto cuando un control o componente de la interfaz reciba el foco.

Se entiende por cambio de contexto un cambio importante en el contenido de la página que, si se realiza sin avisar, puede desorientar a las personas que no pueden ver todo el contenido de la página de forma simultánea. Esto incluye cambios en el agente de usuario/a (navegador, cliente de correo, visor de documentos, etc.), en la vista o área de visualización (ventanas y pestañas del navegador, marcos), en el foco y, de forma general, cambios en el contenido de forma que alteren el significado de la página (página nueva o cambios de contenido tales que parezca que se ha ido a una página nueva).

Por tanto, no usaremos técnicas de scripting para alterar el foco natural de los formularios (por ejemplo cambiando de control automáticamente al llegar a un límite de caracteres) o enviando el formulario al salir del último campo editable.

- » De igual forma, tampoco provocaremos un cambio automático de contexto cuando se cambie el estado o valor de un componente de interfaz de usuario/a, a menos que se haya advertido de ese comportamiento antes de usar el componente.
- » Debemos usar con precaución el evento **onchange**. No usaremos este manejador de evento para implementar un menú de navegación en un control de tipo `<select>` ya que esto provoca un cambio de contexto (página nueva) al cambiar la opción escogida. Si deseamos implementar un menú de este tipo usaremos un botón de envío junto al `<select>` para realizar la acción o, si finalmente usamos el manejador de evento **onchange**, avisaremos previamente de este comportamiento.

De forma general, procuraremos emplear el manejador de evento **onchange** de forma que no provoque un cambio de contexto. Por ejemplo, cuando mostremos una lista de opciones en un menú desplegable de un formulario (`<select>`), se puede usar el evento **onchange** para recargar otro desplegable (p. ej. un desplegable de provincias activa el desplegable de localidades en función de la selección realizada). Este cambio de contenido no se considera un cambio de contexto al no cambiar el significado de la página.

- » Los formularios deben tener un botón de envío estándar y no depender de técnicas de scripting para su envío. Asimismo, si se desea implementar una función de

validación, deberá asociarse al evento `onsubmit` del elemento `<form>`.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Teclado	Principio Operable 2:	2.1.1
Al recibir el foco	Principio Comprensible 3:	3.2.1
Al recibir entradas	Principio Comprensible 3:	3.2.2
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

Scripts en enlaces y cambios de contexto

REQUISITOS

- » El uso de tecnologías de script en los enlaces no debe provocar cambios de contexto imprevistos, como puede ser la apertura de una nueva ventana cuando el enlace recibe el foco del teclado.

Para evitar comportamientos impredecibles, se recomienda emplear técnicas de *mejora progresiva (progressive enhancement)* a la hora de provocar la apertura de nuevas ventanas cuando se usan tecnologías de script.

- » Debemos recordar que no existe un protocolo `javascript:`, por tanto no se usarán enlaces directos a funciones de script en las referencias de los enlaces, por ejemplo:

```
<a href="javascript:abrePagina1();">
```

En su lugar asociaremos la función a un evento y daremos una opción alternativa en la referencia del enlace, por ejemplo:

```
<a href="pagina1.html" onclick="javascript:abrePagina1();">
```

Del mismo modo evitaremos enlaces vacíos (`href="#"`) funcionales únicamente mediante scripts. En su lugar usaremos la misma técnica anterior, dando nuevamente una alternativa en la referencia.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Al recibir el foco	Principio 3: Comprensible	3.2.1
Nombre, función, valor	Principio 4: Robusto	4.1.2
Requisito de Conformidad 4: Uso exclusivo de tecnologías de modo compatible con la accesibilidad	Conformidad	RC4
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo		

Generación de contenido y funcionalidad mediante scripts

REQUISITOS

- » Cuando se usan tecnologías de script para la generación de contenidos o inclusión de funcionalidades, estas deben usarse de modo compatible con la accesibilidad. En caso contrario, no se puede depender de su soporte y se proporcionará una forma alternativa y accesible de acceder al mismo contenido y/o funcionalidad.

- » En caso de usar scripts para generar o manipular contenido de las páginas debe hacerse usando el DOM o Modelo de Objetos de Documento (www.w3.org/DOM/).

No se empleará la función `document.write`, debido a que no pertenece al DOM ni funciona cuando se usa XHTML servido como `application/xhtml+xml`. Tampoco es adecuado emplear la función `object.innerHTML` ya que no forma parte de la recomendación DOM y presenta problemas entre los diferentes agentes de usuario/a, provocando incompatibilidades.

- » En cualquier caso, debemos asegurarnos que las relaciones entre los contenidos existentes y los creados tienen sentido cuando se accede a ellos mediante el uso de agentes de usuario/a y productos de apoyo. Así, cuando insertemos contenido de forma dinámica, debemos ubicar dicho contenido en el DOM (Modelo de Objetos del Documento) de forma que el orden de tabulación y de lectura sean correctos.

Tanto el orden de lectura como el orden del foco por defecto usado por un lector de pantalla se basa en el orden de los elementos (X)HTML en el DOM. Por tanto, para asegurar que este orden sigue siendo correcto, debemos insertar el contenido dinámico en el DOM justo a continuación del elemento que lo lanza.

Por ejemplo, un calendario creado usando DOMScripting debe encontrarse en el código ((X)HTML) en la ubicación adecuada (a continuación del enlace que provoca su aparición) para mantener el orden del foco y de lectura correcto. En caso contrario, aunque visualmente el calendario se muestre en su lugar, si en el código se encuentre

al final del documento algunas personas tendrán problemas para saber de su existencia.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Información y relaciones	Principio Perceptible 1:	1.3.1
Orden del foco	Principio Operable 2:	2.4.3
Procesamiento	Principio Robusto 4:	4.1.1
Nombre, función, valor	Principio Robusto 4:	4.1.2
Requisito de Conformidad 1: Nivel de adecuación	Conformidad	RC1
Requisito de Conformidad 4: Uso exclusivo de tecnologías de modo compatible con la accesibilidad	Conformidad	RC4
Requisito de Conformidad 5: No Interferencia	Conformidad	RC5
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo		

Contenido con límite de tiempo

Se considera contenido con límite de tiempo aquel que limita el tiempo disponible para realizar ciertas acciones (rellenar un formulario, leer un texto, etc.). Debemos asegurar que en todo momento las personas cuentan con tiempo suficiente para interactuar con la Web, siempre que sea posible.

REQUISITOS

- » Debemos evitar el uso de ventanas, sesiones o páginas que se cierran automáticamente al cabo de un cierto tiempo, siendo la persona que usa la web la que decide cuándo quiere cerrarlas. Si se usan estos mecanismos, es necesario permitir que los usuarios y usuarias puedan desactivar el límite de tiempo o configurar el tiempo de vida de las ventanas, sesiones o páginas y mostrar un aviso antes del cierre que permita cancelar la acción.
- » Si el sitio web o una determinada funcionalidad necesitan tener un límite de tiempo para realizar alguna tarea, entonces debemos proporcionar opciones para desactivar el límite de tiempo, ajustar su duración o extender el límite de tiempo para permitir que las personas puedan completar la tarea con éxito.

Existen excepciones a las que no habría que aplicar estas soluciones, como pueden ser las actividades en las que el límite de tiempo es esencial (subastas o exámenes con una duración determinada, etc.) en las que todas las personas deben tener las mismas condiciones para no favorecer a unas frente a otras.

- » Un caso especial es el contenido en movimiento, con desplazamiento o cambio de presentación en periodos regulares ya que introduce un límite de tiempo de lectura del contenido. Asimismo, es importante recordar que este tipo de contenido puede provocar desorientación a algunas personas con ciertas discapacidades. Por tanto, se debe dar la posibilidad de pausar y reanudar el movimiento, parpadeo, desplazamiento o cambio de presentación del contenido.

REFERENCIA TÉCNICA		
Requisito / Criterio de conformidad	Relativo a	Código
Tiempo ajustable	Principio Operable 2:	2.2.1
Al recibir el foco	Principio Comprensible 3:	3.2.1
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página		

Recargas y redireccionamiento

REQUISITOS

- » No se debe usar actualizaciones automáticas de contenido en una página. En caso de ser imprescindible debe darse la posibilidad de omitir la actualización, configurar su frecuencia o realizarlo bajo demanda.
- » Cuando sea necesario usar redireccionamientos automáticos, la mejor opción será usar redirecciones en el lado servidor (y los códigos de redirección adecuados) para que sean transparentes a las personas que usan la web.

Si no fuera posible configurar el servidor para realizar adecuadamente las redirecciones podemos usar mecanismos de script con el mismo propósito pero siempre de forma transparente, es decir, que el intervalo de tiempo pase desapercibido a las personas (tiempo 0).

Del mismo modo, si se usan elementos `<meta>` con el atributo `http-equiv="refresh"` para provocar redirecciones, su valor será cero (tiempo 0).

REFERENCIA TÉCNICA

Requisito / Criterio de conformidad	Relativo a	Código
Tiempo ajustable	Principio 2: Operable	2.2.1
Poner en pausa, detener, ocultar	Principio 2: Operable	2.2.2
Interrupciones	Principio 2: Operable	2.2.4
Requisito de Conformidad 5: No Interferencia	Conformidad	RC5
El cumplimiento de esta recomendación contribuye a mejorar la Usabilidad de la página y a la Independencia de dispositivo		

Formatos diferentes de (X)HTML

Introducción

El principal objetivo será asegurar que los documentos sigan siendo utilizables en caso de que los objetos programados que están incrustados en ellos no estén soportados.

REQUISITOS

Procuraremos que en cualquier elemento de este tipo que tenga una interfaz propia para su uso, sea posible manejarlo independientemente del tipo de dispositivo que utilicemos, por ejemplo, con ratón y con teclado.

Del mismo modo, debemos asegurar que ninguna de las tecnologías empleadas para la generación de contenido de la página web impidan el acceso al resto de contenidos de la página.

Siempre que se pueda se evitará la introducción de funcionalidades y/o contenidos importantes (menús, formularios, galerías de imágenes, textos, etc.) mediante estas tecnologías, apostando por el uso de técnicas estándar XHTML + CSS.

Los contenidos y funcionalidades principales de los portales web deben estar desarrollados mediante XHTML + CSS como base, pudiendo usar otros formatos adicionales para ampliar determinados contenidos o funcionalidades.

A pesar del importante soporte y la amplia difusión de los plugins para estas tecnologías, no se debería confiar la disponibilidad completa de la información o la funcionalidad a su presencia de forma exclusiva. Es preferible ofrecer una alternativa final equivalente en formato XHTML + CSS.

En aquellas ocasiones que hagamos uso de estos complementos o plugins, deberemos indicar claramente que estamos usando un formato distinto, cuál es el plugin requerido y proporcionar una forma de obtenerlo.

Importante: se deberán usar todas las características de accesibilidad directa añadidas que proporcionen cualquiera de las tecnologías que utilicemos.

Flash

El soporte de accesibilidad en Flash se introdujo a partir de la versión 6.0. Sin embargo,

aunque en las últimas versiones se han producido avances significativos, el soporte de accesibilidad es aún limitado y está bastante alejado del nivel de accesibilidad que se puede lograr usando tecnologías web estándar, como los lenguajes estructurados HTML o XHTML combinados con hojas de estilos CSS.

Por otra parte, el funcionamiento de una aplicación Flash depende de la existencia de un plugin o complemento, Flash Player, para poder ejecutarse en los navegadores web.

Nota: en general, aunque el uso de esta tecnología no es incompatible con la creación de sitios web accesibles, se recomienda que no empleemos Flash para contenido que podría presentarse adecuadamente mediante (X)HTML y CSS. En particular, no debemos emplear Flash para realizar sitios íntegramente en Flash, para contenido fundamentalmente estático o para implementar los mecanismos de navegación e interacción del sitio web.

REQUISITOS

En caso de usar Flash con fines que no sean únicamente decorativos (contenidos, menús, formularios, etc.), se debe dar al menos una alternativa final equivalente en formato XHTML+CSS, pudiendo usar el mecanismo de degradación del elemento `<object>` para ofrecer otras alternativas intermedias.

Si la funcionalidad de la animación es únicamente decorativa, no es necesaria una alternativa. Si de cualquier modo se desea dar alternativa, se puede usar también el mecanismo de degradación del elemento `<object>` para ofrecer una serie de alternativas en cascada. Por ejemplo, una primera alternativa en formato video, una segunda alternativa en formato imagen y una tercera alternativa en formato textual si esto resultase útil.

Nota: todos los requisitos de accesibilidad y recomendaciones establecidos a lo largo de la presente guía son aplicables a los contenidos desarrollados mediante tecnología Flash y se llevarán a cabo teniendo en cuenta las características de accesibilidad que proporciona la herramienta (<http://www.adobe.com/accessibility/products/flash>).

Documentos PDF

PDF es un formato de representación de documentos ampliamente difundido y usado.

Los documentos en formato PDF necesitan visualizarse con programas externos diferentes a los navegadores web. Por tanto, si tenemos que incluir documentos PDF en un sitio web accesible, debemos asegurarnos que este tipo de documentos que tienen su propia interfaz también sean accesibles. Por ejemplo, tiene que ser posible manejarlos de forma independiente del tipo de dispositivo (ratón, teclado, etc.) y deben ser compatibles con los productos de apoyo o ayudas técnicas como los lectores de pantalla.

Los PDF no se deben usar como excusa para no generar contenidos mediante XHTML + CSS. En general no se debe abusar de los documentos PDF para proporcionar información, utilizándolos únicamente en determinados casos, por ejemplo:

- » Folletos, documentos legales o similares, destinados principalmente a ser impresos y que deben mantener un formato predeterminado.
- » Información que por naturaleza es muy extensa (por ejemplo un boletín oficial, unas actas, etc.), ya que es conveniente proporcionar una versión descargable e imprimible que facilite la lectura fuera de pantalla.

Aún en estos casos, es necesario que incluyamos en nuestros documentos, y con formato XHTML+CSS, un esquema o resumen de la información que esté contenida en los documentos PDF. Si no es muy extenso, este esquema o resumen se puede incluir en la misma página desde la que se enlaza al documento PDF, a modo de descripción del mismo. En caso de que el resumen sea más extenso, o se proporcione una versión alternativa completa en formato (X)HTML, se puede incluir en otra página y proporcionar un enlace a la misma junto al enlace al documento en PDF.

Como medida adicional, los enlaces que proporcionen acceso a este tipo de documentos deberán describir en el texto de los mismos el formato de los mismos así como su peso aproximado.

Nota: todos los requisitos de accesibilidad y recomendaciones establecidos a lo largo de la presente guía son aplicables a los contenidos en formato PDF y se llevarán a cabo teniendo en cuenta las características de accesibilidad que proporciona la herramienta:

<http://www.adobe.com/accessibility/products/acrobat/>

WAI-ARIA

Para que los **componentes de interacción** de una página web sean accesibles es necesario que las aplicaciones de usuario, y en particular los productos de apoyo, sean capaces de **comprender e interactuar con los mismos**. Si los productos de apoyo no *comprenden* ni pueden interactuar con los componentes de interacción, tampoco lo podrán hacer los usuarios de estos productos de apoyo.

Con este objetivo, para que sea posible usar los componentes de interacción a través de los productos de apoyo es necesario que estos sean capaces de **obtener la siguiente información**:

- » **Función (rol)**: la funcionalidad del componente de interacción.

Los productos de apoyo tienen que poder **reconocer cuál es la función** del componente de interacción para transmitirla a los usuarios. En (X)HTML, cuando un lector de pantalla encuentra un enlace avisa a los usuarios diciendo la palabra "enlace" y si encuentra un `<input type="text">` avisa diciendo "campo de edición de texto" o un mensaje similar.

Si el lector de pantalla no es capaz de determinar la función del componente de interacción, entonces tampoco será capaz transmitirla a los usuarios y dicho elemento será inaccesible.

- » **Nombre**: texto a través del cual se identifica un componente de interacción.

Los productos de apoyo tienen que poder **reconocer cuál es el nombre** del componente para transmitirlo a los usuarios. En (X)HTML, el nombre de un enlace será su texto, en un campo de formulario será el contenido de su etiqueta `<label>` o del atributo `title`, etc. Por este motivo algunos requisitos de accesibilidad obligan a proporcionar nombres significativos (textos de los enlaces, etiquetas, etc.) para que los usuarios puedan reconocerlos.

- » **Valor**: propiedades y estados de los componentes de interacción.

Los productos de apoyo tienen que poder reconocer cuál es el valor (si lo tienen) de los componentes de interacción para transmitírselo a los usuarios y operar con él.

Por ejemplo, en (X)HTML el valor de un enlace es su atributo `href` y el valor de un `<textarea>` será el texto que contiene.

Algunos elementos de interacción pueden tener varios estados. Por ejemplo, los botones de radio pueden estar seleccionados o no estarlo. Lo mismo ocurre con las casillas de verificación. Los productos de apoyo tienen que poder **reconocer tanto los valores como los estados** de los elementos de interacción para transmitírselos a los usuarios.

De igual forma, las propiedades, valores y estados que pueden ser asignados por los usuarios tienen que poder **ser especificados por software**. Es decir, si el usuario tiene la posibilidad de cambiar el valor o el estado de los componentes de interacción,

tiene que poder hacerlo también por medio de un producto de apoyo. Así, por ejemplo, un usuario podrá escribir en un campo de edición o cambiar el estado de un botón de radio o una casilla de verificación interactuando "a través" de un lector de pantalla.

» **Cambios en los valores o estados**

Finalmente, cuando se producen cambios en los elementos de interacción, como cambios en sus valores o estados, **los avisos sobre estos cambios deben estar disponibles** para los agentes de usuario, incluyendo los productos de apoyo. Por ejemplo, uno de los estados más importantes de los elementos de interacción es el foco, si lo tiene o no. El estado del foco de un componente de interacción tiene que poder determinarse por software, y los avisos sobre los cambios del foco se tienen que enviar a los agentes de usuario y productos de apoyo.

Para poder llevar a cabo todo este proceso de comunicación es necesario que los componentes de interacción se creen con una tecnología compatible con la accesibilidad. Así, **si usamos tecnologías accesibles, componentes de interacción estándar** y siguiendo la especificación (por ejemplo, usando XHTML estándar) **este proceso es directo**.

Todos los componentes de interacción estándar de (X)HTML tienen definido cuál es su rol, y a partir de qué elementos o atributos se obtiene su nombre, valor y estado.

Cómo se determina la función, nombre, valor y estado a partir de enlaces y controles de formulario (X)HTML

ELEMENTO	ROL	NOMBRE	VALOR	ESTADO
<code><a></code>	Enlace	Texto del elemento <code><a></code> , atributo alt si se trata de una imagen en un enlace, o atributo title . (Si en el enlace hay tanto un texto como una imagen con atributo alt , se usan ambos conjuntamente.)	Atributo href	
<code><button></code>	Botón de envío	Texto dentro del elemento <code><button></code> o atributo title		
<code><fieldset></code>	Grupo	Elemento <code><legend></code>		
<code><input type="button", "submit" o "reset"></code>	Botón de envío	Atributo value		

<code><input pe="image"></code>	ty-	Botón de envío	Atributo alt o atributo title		
<code><input pe="text"></code>	ty-	Texto editable	Elemento <label> asociado o atributo title	Atributo value	
<code><input pe="password"></code>	ty-	Texto editable	Elemento <label> asociado o atributo title		
<code><input pe="checkbox"></code>	ty-	Casilla de verificación	Elemento <label> asociado o atributo title		Atributo checked
<code><input pe="radio"></code>	ty-	Botón de radio	Elemento <label> asociado o atributo title		Atributo checked
<code><select></code>		Menú de selección, lista o lista desplegable	Elemento <label> asociado o atributo title	Elemento <option> con atributo selected en el estado selected	
<code><textarea></code>		Texto editable	Elemento <label> asociado o atributo title	Texto dentro del elemento <textarea>	

Al usar estos controles (X)HTML estándar aseguramos la correcta interacción entre estos y los productos de apoyo. Nuestra responsabilidad como desarrolladores es proporcionar los valores de aquellas propiedades de accesibilidad que sea necesario establecer y no se proporcionan de forma automática, como el nombre. Por ejemplo, un texto significativo para los enlaces o una etiqueta identificativa (**<label>** o **title**) para los controles de formulario.

Una vez hecho esto, los productos de apoyo serán capaces de obtener la información necesaria, interactuar y estar al corriente del estado de los controles usados sin que tengamos que adoptar medidas adicionales.

Sin embargo, si usamos **controles personalizados** mediante scripts u otras tecnologías, ya sea modificando los controles estándar existentes o creando controles propios, entonces **debemos adoptar medidas adicionales para asegurar la compatibilidad con los productos de apoyo.**

Aplicaciones Ricas de Internet Accesibles

Con la aparición y evolución de las aplicaciones web, el grupo de componentes de interacción estándar de (X)HTML resulta limitado para lograr interfaces de usuario complejas similares a las aplicaciones de escritorio (vistas en árbol, sliders o controles deslizantes, etc.). Esto es debido a que HTML no se diseñó para crear aplicaciones web. Para sortear esta

limitación, los desarrolladores han ido creando nuevos **componentes personalizados** que cubriesen sus necesidades.

Estos componentes personalizados se suelen crear a partir de los controles estándar de (X)HTML (enlaces y controles de formulario) junto con otros elementos estáticos (como gráficos e imágenes, `<div>`, ``, `<td>`, etc.) a los que se les añade una **nueva capa de comportamiento mediante tecnologías como JavaScript y Ajax** para crear aplicaciones de Internet ricas, emulando la funcionalidad de los componentes de interacción de escritorio u otros componentes complejos.

Problemas de accesibilidad

Para que todos estos nuevos componentes de interacción sean accesibles para las personas con discapacidad, **los productos de apoyo tienen que poder interactuar con los mismos**. Sin embargo, la información de accesibilidad que necesitan los productos de apoyo no está disponible si estos controles se implementan únicamente con las tecnologías web habituales como (X)HTML, JavaScript y/o Ajax, sin aplicar medidas adicionales.

» Los productos de apoyo no saben cuál es la **FUNCIÓN** del componente

Por ejemplo, imaginemos un control deslizante (*slider*) que permita especificar múltiples valores. Este control no existe en (X)HTML y por lo tanto es necesario programarlo. Para ello, el nuevo control se puede basar en un control existente al que se le añade una nueva capa de comportamiento mediante JavaScript. Una posibilidad es usar un botón estándar que al pulsar sobre él se detecte el desplazamiento del ratón y en función de este desplazamiento se asigne un valor u otro.

Volumen: 56%

Control deslizante personalizado para manejar el volumen.

La nueva presentación visual de este componente se lograría también mediante una combinación de JavaScript y hojas de estilo. Así, en el código (X)HTML sólo tendríamos una etiqueta y un botón.

Volumen:

Botón de envío en el que se basa el control deslizante para manejar el volumen.

Los usuarios de lectores de pantalla únicamente sabrán que se trata de un "*botón de envío*", ya que esa es la función del control estándar en el que se basa el nuevo componente de interacción. Si está bien etiquetado ("*Volumen*") sabrán que se trata de un botón para controlar el volumen, pero no podrán saber cuál es realmente la función y forma de uso del nuevo control ("*slider*" o "*control deslizante*")

» Los productos de apoyo no saben cuál es el **ESTADO** actual del componente

Si el componente tiene determinadas propiedades que varían de forma dinámica, según se interactúe con los mismos, los productos de apoyo y los usuarios no sabrán cuál es el valor actual de estas propiedades.

Siguiendo con el ejemplo del control deslizante, los usuarios de lectores de pantalla no pueden saber cuál es el valor actual del control al manejarlo, ya que el botón en el que se basa no tiene estados en HTML. ¿El volumen está al 20% o al 80%?

» **Los productos de apoyo no sabrán el VALOR de otras propiedades importantes**

El valor de otras propiedades tampoco está disponible para los productos de apoyo. Por ejemplo, en el control de volumen no saben cuáles son los valores mínimos y máximos que se pueden asignar con el control, ya que el elemento `<button>` en el que se basa no dispone de estas propiedades.

» **Los productos de apoyo no serán conscientes de las ACTUALIZACIONES del contenido**

Cuando el uso de un control implica un cambio en el contenido de la página, estas actualizaciones no suelen ser percibidas por los usuarios de productos de apoyo. Mediante AJAX se puede actualizar el contenido de la página, o parte del mismo, mediante consultas al servidor en segundo plano de forma asíncrona. Estos cambios en el contenido no suelen ser percibidos por los productos de apoyo y, en el caso de que sí lo fuesen, los usuarios podrían no darse cuenta de que se ha actualizado el contenido o cómo localizar el nuevo contenido.

Un control deslizante como el del ejemplo puede usarse también para desplazarse por una galería de imágenes. Según desplazemos el control cambiará la imagen visualizada en primer plano en ese momento. Es decir, se está produciendo un cambio en el contenido de la página a consecuencia del desplazamiento del control. Tanto los productos de apoyo (p. ej. lectores de pantalla) como los usuarios no serán conscientes de dicha actualización ya que lo que ellos perciben que están manejando es un simple *“botón de envío”*.

» **Navegación mediante el TECLADO limitada**

La navegación mediante teclado se puede ver limitada o impedida por varios motivos:

- Los componentes se puede implementar usando elementos estáticos de (X)HTML. Por ejemplo, se puede añadir funcionalidad a gráficos o imágenes (``), encabezados o celdas de tablas (`<th>` y `<td>`), o incluso a contenedores como `<div>`, entre otros. Todos estos elementos estáticos de (X)HTML no pueden recibir el foco y, por tanto, son inaccesibles mediante teclado.
- Si los componentes se implementan usando controles estándar de (X)HTML sí pueden recibir el foco. Sin embargo, si se modifica la forma natural de interacción con los mismos es posible que tampoco se puedan operar mediante teclado. Por ejemplo, un control deslizante implementado con un botón (`<button>`) se maneja pulsando sobre él y *“desplazándolo”* con el ratón.

A pesar de ser accesible con teclado, la forma de interacción normal del control estándar es simplemente pulsando sobre él y no desplazándolo. Los usuarios de teclado no podrán manejarlo correctamente mediante el tabulador. Incluso en el caso de que se implemente con scripts un mecanismo de uso mediante el teclado, como las teclas de dirección, los usuarios podrían no ser conscientes de esta posibilidad ya que perciben el componente como un “botón de envío” (su función por defecto) y no esperan un comportamiento diferente.

WAI-ARIA

La especificación **WAI-ARIA** (*WAI - Accessible Rich Internet Applications*, Aplicaciones Ricas de Internet Accesibles de la Iniciativa de Accesibilidad Web) trata de solucionar los problemas de accesibilidad relacionados con el uso de contenidos dinámicos e interfaces de usuario avanzadas realizadas en Ajax, Javascript, HTML y tecnologías relacionadas.

El objetivo de WAI-ARIA es **solucionar estos problemas de accesibilidad sin renunciar a las ventajas y funcionalidades de este tipo de aplicaciones**. Es decir, no se trata de impedir o prohibir su uso, sino de aportar soluciones que permitan crear aplicaciones ricas de Internet accesibles.

Para ello, permite establecer atributos a los elementos para identificar la **función**, (*botón, deslizador, desplegable, etc.*), **propiedades y valores**, (*valor máximo, mínimo, actual, etc.*) y **estado** (*activado, desactivado, etc.*) de cada elemento de modo que estas se transmitan a los productos de apoyo, así como las notificaciones sobre los cambios producidos en los mismos. Además, permite identificar aquellas **regiones activas** o susceptibles de sufrir cambios sin que la persona que usa la web pierda el foco de su actividad.

Asimismo, permite extender el uso del atributo **tabindex**, de modo que todos los elementos visibles del documento puedan recibir el foco y se facilite la navegación mediante el teclado.

En caso de usar componentes de interacción no estándar, debemos asegurarnos que sean directamente accesibles y que aplican las prácticas definidas por WAI-ARIA cuando sea necesario.

Para ampliar información sobre WAI-ARIA se puede consultar la documentación del W3C:

- » WAI-ARIA, especificación técnica [<http://www.w3.org/TR/wai-aria/>]: Recomendación W3C de Estándar Web, en estado de recomendación candidata. Destinada a desarrolladores de navegadores web, productos de apoyo y otros agentes de usuario; desarrolladores de tecnologías web; y desarrolladores de herramientas de evaluación de accesibilidad.
- » WAI-ARIA Primer [<http://www.w3.org/TR/wai-aria-primer/>]: Nota del Grupo de

Trabajo del W3C. Se trata de una introducción a WAI-ARIA explicando los problemas de accesibilidad que trata de resolver y los conceptos fundamentales.

- » WAI-ARIA Authoring Practices [<http://www.w3.org/TR/wai-aria-practices/>]: Nota del Grupo de Trabajo del W3C. Destinada a los desarrolladores de contenido web, describe cómo desarrollar aplicaciones enriquecidas de Internet usando WAI-ARIA.
- » WAI-ARIA User Agent Implementation Guide [<http://www.w3.org/TR/wai-aria-implementation/>]: Nota del Grupo de Trabajo del W3C. Describe cómo deben dar soporte a WAI-ARIA los navegadores y otros agentes de usuario. De forma específica, cómo deben transmitir las características de WAI-ARIA a las API de accesibilidad de la plataforma.
- » WAI-ARIA Roadmap [<http://www.w3.org/TR/wai-aria-roadmap/>]: Nota del Grupo de Trabajo del W3C. Describe el camino a seguir para hacer accesibles las aplicaciones ricas de Internet, indicando los pasos ya realizados, los pasos pendientes y la temporalización de los mismos.