

ACCESIBILIDAD

INTRODUCCIÓN

Las pautas de accesibilidad son una recomendación de la W3C para hacer posible una experiencia más satisfactoria, no sólo a las personas con alguna discapacidad, sino que alcanza también a todos los usuarios que utilizan navegadores u ordenadores diferentes o basados en la voz. Estas pautas son principios generales para el diseño web accesible, y han sido desarrolladas por la WAI (Web Accessibility Initiative).

Este manual es una guía de ayuda para comprender estos documentos que a veces resultan un poco complicados, para diseñadores o desarrolladores web que deben revisar la accesibilidad de una interfase. Está organizado por temas, no por niveles de prioridad, para agrupar los conceptos y evitar la información fragmentada, con el objetivo de brindar conocimientos generales para lograr una web lo más accesible posible.

Te será de mucha utilidad tener conocimiento de como estructurar correctamente los documentos, para eso tenemos nuestro [manual de XHTML](#).

1. INFORMACIÓN ALTERNATIVA

Proporcione alternativas equivalentes de contenido visual y auditivo

Lo principal de este punto es el agregado del atributo alt en todos los tags img en caso de que la explicación sea un texto corto, para otros casos que haga falta una descripción más extensa, por ejemplo en gráficos, es recomendable la utilización adicional del atributo longdesc donde se especifica una url que apunta a un documento con la información completa.

Por ejemplo:

```

```

```

```

Incluir enlaces de texto redundantes para las zonas activas de los mapas de imágenes

Toda la información no textual (imágenes, sonidos, applets, presentaciones multimedia, etc.) deben contener información alternativa que transmita el contenido equivalente en formato textual (mediante alt y longdesc).

En caso de incluir en una página un mapa de imagen client-side, también deberemos proporcionar esa información en modo textual, incorporando links redundantes de texto para cada parte del mapa. Esto lo hacemos mediante el tag object que se mostraría cuando no fuera posible mostrar el mapa de imagen.

Por ejemplo:

```
<object data="imagen1.jpg" type="image/gif" usemap="#map1">
<map name="map1">
[<a href="index.html" shape="rect"
coords="0,10,16,25">Índice</a>]
[<a href="pagina1.html" shape="rect"
coords="10,20,26,35">Página 1 </a>]
[<a href="pagina2.html" shape="rect"
coords="24,30,50,60">Página 2 </a>]
</map>
</object>
```

Incluir presentaciones gráficas o auditivas cuando facilite la comprensión del contenido

Se ha comprobado que es conveniente acompañar cierta clase de información, como procesos o procedimientos, con gráficos e infografías, sonidos o animaciones que complementen el concepto porque sería mejor comprendida.

Por ejemplo:

- Un esquema con datos complejos como las cifras de los negocios del año fiscal.
- La traducción de un texto a una animación con lenguaje de señas. Esto sería útil para personas que pueden comunicarse con el lenguaje de señas americano, pero no saben leer el inglés.
- Sonidos pre-grabados de discursos o efectos sonoros pueden ayudar a los no videntes. Si bien se puede convertir el texto en discurso con el sintetizador, éste no reproduce los cambios de tono de voz que proporciona el discurso grabado, perdiendo parte de la información.

2. NO BASARSE SÓLO EN EL COLOR

Esta pauta se refiere a dos premisas: en primer lugar, cuando transmitimos información a través del color, debemos tener en cuenta que cuando el mismo está deshabilitado la información debe seguir estando disponible; la segunda premisa tiene que ver con generar una combinación de colores de texto y de fondo con suficiente contraste como para ser leído por personas con deficiencias de percepción de color o en pantallas en blanco y negro.

En este sitio ofrecen un validador de color: [AccessColor](#). Esta es una extensión para Firefox, que analiza el contraste de color: [Colour Contrast Analyser](#).

3. UTILIZAR MARCADORES Y HOJAS DE ESTILO

Usar marcadores en vez de imágenes para transmitir la información

No es recomendable el uso de imágenes para representar texto, debemos resolver el diseño con CSS. Para los casos en que existiera un lenguaje acorde para transmitir la información (como el caso de MathML para fórmulas matemáticas) se debe emplear éste para que no pierda su formato.

Documentos validados por las gramáticas formales publicadas.

El DOCTYPE del documento es la primera línea de texto del código e identifica y valida la estructura del archivo contra estándares prefijados. Es imprescindible que esté para que el documento sea válido. Si la declaración del DOCTYPE es correcta el navegador del usuario podrá conocer e interpretar la estructura del documento.

Por ejemplo:

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

En este caso estamos indicando que el contenido del documento está hecho conforme a los estándares de HTML 4.01 que se valida contra la DTD loose.dtd, la cual contiene la semántica de la especificación y se encuentra hosteada en el W3C.

Uso de las hojas de estilo (CSS) para controlar la maquetación y la presentación.

Es aconsejable diseñar los contenidos con CSS en vez de usar imágenes, tablas o definir el formato en el HTML, porque los contenidos estarán disponibles para más usuarios, especialmente para los que no utilizan dispositivos estándares para acceder a la web. Con CSS podemos manejar sangrías, espaciados entre palabras y letras, alineación del texto; podemos controlar la posición visual de los elementos en una página independientemente del lugar en que se encuentren dentro del documento, donde el orden debe responder al lugar que ocuparían sin hoja de estilos, con ella se puede hacer cualquier tipo de diseño visual.

Además, el uso de hojas de estilo permite a los usuarios definir la propia por intermedio de facilidades que ofrecen los navegadores de imponerlas por sobre las del autor, y así tener control total sobre contrastes o tamaños para navegar los sitios con comodidad de acuerdo a sus necesidades.

Un ejemplo de la diferencia de dar el formato en el HTML:

```
<div align="center"><h1>Titulo 1</h1></div>
```

Y con hojas de estilo:

```
h1 { text-align: center;}
```

Para mayor información sobre el uso de hojas de estilo visite nuestro [manual de CSS](#)

Usar unidades de medida relativas para los elementos de un documento

Otro ítem importante es el uso de medidas relativas (em o %) en lugar de absolutas (px o cm) para definir los elementos y posiciones, así el usuario pueda modificar el tamaño de la página sin que pierda su formato.

Por ejemplo, en vez de utilizar:

```
h1 { font-size: 11px; }
```

Usamos:

```
h1 { font-size: 0.69em; }
```

Aquí hay un conversor de píxeles a ems online: [Em Calculator](#).

Para usuarios de Mac, este es un widget muy útil, que, entre otras bondades útiles e interesantes, tiene un conversor de medidas para web: [Designers Toolbox](#).

Utilizar marcadores y hojas de estilo apropiadamente

Debemos utilizar los tags de manera que cumplan la función para la que fueron creados y no para dar formato al documento porque dificultaremos la comprensión del contenido a las personas que no puedan verlo. El formato del documento se lo daremos con las hojas de estilo.

Lo más común es utilizar los tag h1, h2, etc. para resaltar una palabra cuando deberían usarse para marcar título, subtítulos, etc. o marcar citas (q o blockquote) para generar sangrías.

Ejemplo erróneo:

```
<h4>El río más ancho del mundo es el Río de la Plata </h4>
```

Ejemplo correcto:

```
<strong>El río más ancho del mundo es el Río de la Plata</strong>
```

Específicamente los elementos de encabezado, deben guardar un orden lógico (primero h1, después h2, luego h3, y así sucesivamente), no debemos saltar ninguno de los niveles ni poner más de un h1 por página. Recuerda que no está bien utilizar los encabezados para crear efectos de diseño.

Un buen uso de las etiquetas para encabezamientos sería:

```
<h1>Título del libro</h1>
```

```
<h2>Título del capítulo</h2>
```

```
<h3>Título de la sección</h3>
```

Respecto a los tags para citas, q se utiliza para citas cortas y es un elemento *inline* y blockquote anida las citas en *bloque* y diferencia un párrafo común de uno que contiene citas, indentando los textos que contiene, efecto que puede controlarse con CSS.

Los elementos estructurales tales como títulos, subtítulos, párrafos y citas deben estar marcados correctamente para que las personas que utilizan lectores de voz puedan ir saltando de marcador en marcador a través de la página.

Listas e ítems de las listas marcados correctamente

Es importante estructurar correctamente los elementos de listado (dl, ul y ol) y obviamente, no utilizarlos para generar efectos de diseño, sino para crear listas.

Las personas ciegas podrían desorientarse con listas que no especifican el nivel de anidamiento de cada ítem (lo ideal es hacerlo con números compuestos: 1, 1.1, 1.1.1), también les es más fácil navegar por listas ordenadas que por aquellas no numeradas, especialmente si están anidadas, para estos casos los desarrolladores deberían implementar algún tipo de pista contextual.

Con hojas de estilo podemos modificar totalmente la manera en que se muestran, y controlar el estilo de números o viñetas.

4. IDENTIFICAR EL IDIOMA

Identificar el lenguaje natural utilizado en un documento y los cambios del idioma

Para que el lector de pantalla pueda acomodar la pronunciación de acuerdo al idioma de la página, es necesario incluir; en el encabezado, el lenguaje natural y aparte marcar cada cambio de idioma en el texto en el documento mediante tags. El lenguaje natural de un documento se pone con el atributo lang en el tag html.

Por ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Titulo de la página</title>
</head>
<body>
Este es el cuerpo del documento que contiene un cambio de idioma en las palabras <span
lang="en">World Wide Web Consortium</span>
</body>
</html>
```

Para documentos XML se utiliza xml:lang= y para los archivos XHTML se usan ambos así:
<html lang="es" xml:lang="es">

Cada vez que un acrónimo o abreviatura aparezca por primera vez en un documento, debe ser especificada

Este punto que recomienda especificar el significado completo de cada abreviatura o acrónimo, usando el atributo title para describir su significado en los elementos abbr y acronym.

Por ejemplo:

```
<acronym title="Hypertext Markup Language">html</acronym>
```

Según Wikipedia, un acrónimo es un conjunto de letras formado por las iniciales de otras palabras, por ejemplo: URL (Uniform Resource Locator), y una abreviatura es la representación escrita de una o varias palabras mediante una o varias de sus letras, a fin de que las palabras o las palabras en cuestión resulten más cortas en el texto. Como por ejemplo Mb (Mega Byte).

Una regla fácil (que encontré en el sitio de SIDAR) para diferenciar uno de otro es; que la abreviatura no se lee la palabra que se forma sino que se traduce su significado, por ejemplo “etc.” leemos “etcétera”, no “e-t-ce”, en cambio el acrónimo se lee la palabra que se crea con la forma abreviada o las siglas que la componen, como está escrita, por ejemplo, “FBI”, que no es pronunciable, la leemos como “Efe-be-i” y “ONU”, sí es pronunciable, la leemos como “onu”.

Pero estos criterios no están unánimemente decididos, su definición en W3C está medio difusa, encima IE decidió no soportar el tag <abbr>, así que muchas veces se privilegia el uso de <acronym> para cualquiera de los dos casos...

5. TABLAS

Utilizar tablas que se transformen correctamente

La W3C recomienda no utilizar tablas para maquetar, es decir para dar formato. Usaremos las tablas sólo cuando tengamos datos tabulares, en este caso debemos tener en cuenta una serie de cosas como: proporcionar un título (caption) para describir la tabla, identificar los headers (th), utilizar tags para asociar las celdas de datos (thead, tfoot, tbody, col, colgroup) y las celdas de encabezamientos (axis, scope, headers) en aquellas tablas con 2 o más niveles lógicos y proporcionar un resumen de las relaciones entre celdas (summary) y abreviaturas (abbr) para las etiquetas de los encabezamientos. También se puede utilizar el atributo title del tag table para una descripción de pocas palabras.

Ejemplo:

```
<table summary="tabla que presenta las siglas de un acrónimo y su significado" >
```

```
<caption>Acrónimos de uso común en tecnología</caption>
```

```
<thead>
```

```
<tr>
```

```
<th>Siglas</th>
```

```
<th>Significado</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>P2P</td>
```

```
<td>Peer to peer</td>
```

```
</tr>
```

```
<tr>
```

```
<td>WAP</td>
```

```
<td>Wireless Application Protocol</td>
```

```
</tr>
```

```
<tr>
```

```
<td>LDAP</td>
```

```
<td>Lightweight Directory Access Protocol</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

Proporcionar texto lineal alternativo para todas las tablas que maquetan texto en paralelo, en columnas de palabras

Cuando se maqueta la información con tablas, es muy dificultosa la comprensión de las mismas con lectores de voz viejos, porque hacen una lectura lineal en lugar de hacerlo columna por columna, esto produce una interpretación errónea del contenido. Esto también puede suceder con tablas que no son de maquetación, la información se desordena y se torna incomprensible.

Para establecer una orden de recorrido de las columnas se utiliza el atributo dir, que especifica una secuencia de recorrido. Sus valores pueden ser rtl (right to left) o ltr (left to right), esto puede cambiar con el idioma de la página que especifica un sentido de lectura.

6. NUEVAS TECNOLOGÍAS

Las páginas que incorporan nuevas tecnologías deben transformarse correctamente

Se recomienda diseñar la página para que pueda ser interpretada y cumplir su función independientemente del dispositivo utilizado, por ejemplo, que se pueda leer sin hojas de estilo o que funcione sin javascript. Si incorporamos contenido dinámico, scripts u otro tipo de programación, tenemos que proporcionar la información alternativa accesible.

Concretamente, si nuestra barra de navegación se basa en un applet java que realice un roll-over y el navegador no soporta java o lo tiene desactivado, directamente no podrá navegar nuestro sitio, si no podemos evitar usarlo debemos proporcionar un equivalente con el tag <noscript>, por ejemplo. También hay que tener en cuenta de actualizar el contenido alternativo al dinámico con la misma frecuencia que el original.

Específicamente para cuando necesitamos incorporar elementos de Flash, existe una técnica accesible que, mediante Javascript, detecta el plug-in, embebe el objeto de Flash, respeta los estándares de W3C y resuelve el problema de Eolas de IE. Además es gratuito y se puede descargar desde [aquí](#).

Proporcionar información de contexto y orientación

Titular cada frame para su identificación y navegación.

Ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 FRAMESET//EN">
<html>
<head>
<title>ejemplo de documento html con frames</title>
</head>
<frameset cols="15%,85%" title="página principal de interacciones">
<frame src="nav.html" title="barra de navegación">
<frame src="doc.html" title="contenido">
<noframes>
<a href="doc1.html" title="página sin frames de interacciones"> ir a Interacciones</a>
</noframes>
</frameset>
```


7. CONTROL SOBRE LOS CONTENIDOS TEMPO-DEPENDIENTES

Asegurar el control sobre los cambios de los contenidos tempo-dependientes

Esto significa evitar el uso de los tags <blink> y <marquee>, que, aparte de ser muy molesto, hacen que la pantalla parpadee o se desplace y puede provocar epilepsia en algunas personas.

Además debemos evitar crear páginas que se redireccionen o actualicen automáticamente (se recomienda que se ejecuten desde el servidor y no desde la página) y proporcionar la posibilidad de detener el contenido de scripts que generen movimiento.

8. ASEGURAR LA ACCESIBILIDAD DE LAS INTERFASES DE USUARIO INCRUSTADAS

Se trata de lograr que los elementos de programación, como los applets y scripts, sean directamente accesibles, es decir que puedan controlarse independientemente del dispositivo, que se lo pueda manejar con el teclado, que tenga voz automática, etc. Si no logramos que la interfaz de estos objetos incrustados sea accesible, tendremos que ofrecer una alternativa que sí lo sea. (La pauta 6 está relacionada con esto).

9. DISEÑAR PARA LA INDEPENDENCIA DEL DISPOSITIVO

Generar un orden lógico de navegación, para enlaces, controles de formularios y otros objetos

También debemos estructurar el documento de forma lógica para permitir la navegación a través de su contenido (enlaces, formularios, etc.) a personas que no utilizan el mouse o usan dispositivos manejados con la voz. Esto implica fijar, mediante el código HTML, estructuras alternativas de navegación de los elementos activos.

Con la incorporación del atributo tabindex daremos un orden de navegación con el tabulador, y permitiremos el salto de enlace en enlace o de campo en campo en el caso de formularios. Este atributo se puede usar en los tags: a, area, button, input, object, select, y textarea.

El procedimiento consiste en asignar el atributo tabindex a cada elemento activo de la página con un valor positivo. De esta manera el usuario saltará con la tecla tab desde el elemento más bajo al más alto, luego los que no tengan el atributo y no tomará en cuenta los que tengan el atributo disabled.

Por ejemplo:

```
<form action="" method="">
<input tabindex="2" type="text" name="user">
<input tabindex="1" type="password" name="password">
<input tabindex="3" type="submit" name="ingresar">
</form>
```


Brindar atajos de teclado para los links importantes y controles de formulario

El atributo `accesskey` permite otra manera de acceso fácil a los elementos activos de la página mediante atajos de teclado. Se utiliza en los tags: `a`, `area`, `button`, `input`, `label`, `legend` y `textarea`.

Por ejemplo:

```
<a href="../la-empresa.html" accesskey="e">LA EMPRESA</a>
<a href="../servicios.html" accesskey="s">SERVICIOS</a>
<a href="../trabajos.html" accesskey="t">TRABAJO</a>
```

En el ejemplo, activando la letra E, el usuario irá al destino del enlace.

Para asignar los atajos de teclado, se recomienda tener en cuenta los que, por defecto, usan los navegadores para no videntes. [Aquí hay una guía](#).

10. UTILIZACIÓN DE SOLUCIONES PROVISORIAS

Estos puntos son provisionales a la espera de que en el futuro las tecnologías web hayan incorporado nuevas capacidades y no sean necesarios, es decir que sean considerados válidos.

Uno de los puntos es evitar que se abran nuevas ventanas sin avisar al usuario (que puede desorientar a quienes no pueden verlo), entonces hay que evitar el uso del `target="blank"`, y si no podemos evitarlo avisarle al usuario que este enlace abrirá una nueva ventana.

En los formularios debemos agregar texto `alt` al botón de "Enviar" y escribir textos por defecto en las áreas de texto `input` y `textarea`.

Por ejemplo:

```
<form action="">
<textarea name="comentario" rows="20" cols="80">Ingrese aquí su comentario</textarea>
<input type="submit" value="Enviar" / >
</form>
```

Incluir caracteres imprimibles, que no sirvan como link, entre los links contiguos.

En el ejemplo anterior se han agregado caracteres imprimibles, que además de proporcionar una distinción visual que ayuda a los usuarios a diferenciar los enlaces a simple vista, evita el problema que tienen los lectores de pantalla viejos, que leen los enlaces adyacentes como un único enlace.

11. UTILIZAR LAS TECNOLOGÍAS Y PAUTAS W3C

Utilizar las tecnologías y pautas W3C

Si, luego de tratar de implementar los puntos anteriores, no se puede lograr una página accesible, debemos construir una página alternativa que use las tecnologías y recomendaciones del W3C, la cual debe ser:

- accesible
- equivalente a la original
- actualizada con la misma dinámica de la original

Si se usan tablas: No utilizar marcadores estructurales para generar un efecto visual. Por ejemplo, `th` es un tag que resalta y señala la celda que contienen el encabezamiento de una tabla y no debe ser usado para otra cosa que no sea eso.

Si se usan frames: hay que describir la función de cada uno de los marcos y cómo éstos se relacionan entre sí si no resulta obvio solamente con el título del marco.

Si se usan formularios: las etiquetas deben estar colocadas adecuadamente y asociadas explícitamente con sus controles.

Si se usan applets o scripts: el usuario debe poder manipular los eventos independiente mente del dispositivo que utilice. Por ejemplo: un script que sólo dependa del uso del mouse mostrará serias complicaciones para los usuarios que no manejen este elemento, como quienes tienen problemas motores.

Evitar características desaconsejadas por las tecnologías W3C

Hay que tener cuidado en relación a utilizar atributos que pueden estar obsoletos para la versión HTML 4.0, como por ejemplo el tag `font` o `bgcolor`. En cambio su función se realiza desde la hoja de estilos.

Preparar la información para que los usuarios puedan recibirla de acuerdo a sus preferencias de idioma, formato o tipo de contenido

Este punto tiene 2 apartados, por un lado propone mejoras como:

- Hacer diferentes versiones del contenido que se ofrece, como puede ser proveer diversos idiomas.
- Ofrecer una negociación del contenido que se provee al usuario que lo solicita, por ejemplo de acuerdo a idioma que lo representa.
- Indicar el idioma y el tipo de contenido de la información. Para esto se utilizan los tags `type` y `hreflang`. `hreflang` es un atributo del tag `a` con el cual indicamos el idioma con el que se va a encontrar el usuario si sigue el enlace. Su valor especifica los identificadores para cada lenguaje. Por ejemplo: `en` para inglés, `fr` para francés, `es-ar` para español argentino, etc.

Las otras premisas tienen que ver con aprovechar las facilidades técnicas del CSS2 para brindarle al usuario información de una manera que pueda comprender:

- Así como el tipo de letra proporciona información visual, se utilizan las propiedades auditivas para entregar información a los usuarios de lectores de voz. Es decir, distinguir títulos, énfasis, párrafos, etc. utilizando las voces.
- La regla `@media` permite diseñar hojas de estilo especiales para distintos dispositivos. Existen muchos tipos de media configurable, por ejemplo `aural` para sintetizadores de voz, `braille` para ese tipo de dispositivos táctiles, `print` para estilo de impresión, `handheld` para dispositivos portátiles, `tv` para televisión, etc.

Ejemplo:

```
@media print {h1 {color: #FF0000;}}
```

```
@media screen {h1 {color: #000000;}}
```

12. PROPORCIONAR INFORMACIÓN DE CONTEXTO Y ORIENTACIÓN

Dividir los bloques largos de información en grupos más manejables

En los formularios debemos agrupar unidades semánticas con fieldset y describirlas con el tag legend, que asocia las etiquetas con los controles de formulario. También se recomienda el uso de label (con el atributo for, por ejemplo, que debe tener un valor idéntico al del atributo id para asociarlos) y tabindex para completar la accesibilidad del formulario.

Por ejemplo:

```
<form action="" method="post">
<fieldset>
<legend>datos personales</legend>
<label for="nombre">nombre: <input type="text" id="nombre" tabindex="1"></label>
<label for="apellido">apellido: <input type="text" id="apellido" tabindex="2"> </label>
</fieldset>
<fieldset>
<legend>datos laborales</legend>
<label for="empresa">empresa: <input type="text" id="empresa" tabindex="3"> </label>
</fieldset>
</form>
```

Dentro de este concepto entra también el uso del tag optgroup para organizar listas de opciones de menú en los formularios. Además de otros ítems ya analizados como usar tablas para datos tabulares, describirlas con caption, agrupar filas y columnas con thead, tbody, tfoot y colgroup, anidar listas con ul, ol y dl y usar h1 - h6 para estructurar documentos y p para separar tramos largos de texto.

13. PROPORCIONAR MECANISMOS CLAROS DE NAVEGACIÓN

Identificar el objetivo de cada enlace

Esta pauta se refiere a que los enlaces deben ser claros, significativos y cortos para cuando se los lee fuera de contexto (por ejemplo, escribir “portfolio de trabajos 2007” en lugar de “haga clic aquí”). Además, los tag a deben llevar el atributo title que permite agregar más datos sobre el contexto al cual nos lleva, y funciona como el texto alternativo que aparece cuando el usuario posa el mouse sobre el enlace.

Por ejemplo:

```
<a href="aquivalaurl.html" title="descripción extendida">Aquí va el texto del enlace</a>
```

Cuando tenemos varios documentos relacionados debemos relacionarlos también en el elemento link con el atributo rel o rev (por ejemplo rel=siguiente; rev=anterior).

Utilizar los metadatos para agregar información de tipo semántica al sitio

Los metadatos describen la información sobre un recurso para que sea localizable más fácilmente. Los metadatos que provee el lenguaje HTML son:

title es el título de la página

adress se utiliza para dar información de contacto

meta en este tag se especifican las palabras clave, descripción, fecha, datos del autor, de copyright, etc. También se pueden incluir otros estándares que amplían estos metadatos, como por ejemplo [Dublin Core](#).

link ayuda a describir mecanismos y organización del documento y adjuntar documentos, como por ejemplo las hojas de estilo, que pueden ser varias dependiendo del contexto en que se visualice la página.

Brindar información adicional sobre la estructura del sitio

Es importante brindar un mapa del sitio que permita tener de un solo vistazo la estructura completa de la web.

También es recomendable agregar comentarios dentro del código de la página, indicando el comienzo y final de cada bloque de información que forme parte de la estructura de la página, como encabezado, menú principal, menú secundario, cuerpo principal, pie, etc.

Por ejemplo:

```
<body>
<!--Comienza el encabezado-->
<div id="cabecera"><?php include('includes/cabecera.php');?></div>
<!--Termina el encabezado-->
<!--Comienza la barra de navegaci&oacute;n principal-->
<div id="menuppl"><?php include('includes/menuppl.php');?></div>
<!--Termina la barra de navegaci&oacute;n principal-->
<div id="contenedor-principal">
<!--Comienza el contenido de la p&aacute;gina-->
```

Usar los mecanismos de navegación de forma coherente

Es importante que las páginas cuenten con una estructura simple y ordenada que se repita a lo largo de todo el sitio; mantener un estilo y ser consecuente con el mismo, las barras o elementos de navegación deben estar al alcance del usuario y siempre en los mismos lugares, lo mismo que los contenidos, la publicidad, las opciones de búsqueda, etc.

Incluir barras de navegación en el sitio como mecanismo de navegación

Las barras de navegación son grupos de enlaces por los cuales el usuario accede a los contenidos de un sitio web, por lo tanto es conveniente que sean fácilmente reconocibles, que la información que proveen esté ordenada de forma clara y que, preferiblemente, se encuentren en el mismo sitio y con la misma estética a lo largo de todo el sitio para que el usuario pueda reconocerlo fácilmente.

Por otra parte, se recomienda que las categorías o secciones que contenga la barra sean pocas y concisas, si el sitio es muy grande, conviene poner sólo las categorías principales y que éstas contengan subcategorías.

Identificar en conjuntos lógicos los enlaces relacionados

Se refiere a las barras de navegación principales, que están presentes en todas las páginas de un sitio. El usuario que usa lector de pantalla para interpretar una página, cada vez que ingrese a una nueva página deberá releer cada una de las opciones de la botonera. Por eso debemos agrupar los enlaces en un conjunto lógico (como la barra de navegación) e identificarlo como una unidad, y brindar al usuario la posibilidad de saltarlo en las páginas internas.

Tenemos diferentes alternativas para poner a disposición del usuario la posibilidad de saltarse grupos de enlaces:

- Incluir un enlace al comienzo de la barra que diga “Saltar navegación principal” que apunte a un *anchor* o ancla ubicado donde empieza el contenido dentro de la misma página.
- Proporcionar una hoja de estilos para que los usuarios puedan ocultar el grupo de enlaces.
- el uso tag map, identificado con el atributo title, permite agrupar links

Este último punto está ejemplificado a continuación:

```
<map title="Barra de navegación">
[ <a href="#salto">Saltar Barra de navegación</a> ]
[ <a href="prueba1.html">Sección 1</a> ]
[ <a href="prueba2.html">Sección 2</a> ]
[ <a href="prueba3.html">Sección 3</a> ]
[ <a href="prueba4.html">Sección 4</a> ]
[ <a href="prueba5.html">Sección 5</a> ]
</map>
<h1><a name="salto">Título del texto del sitio </a></h1>
```

El tag map agrupa los enlaces que forman la barra de navegación principal, que aparece al comienzo de cada página. El atributo title lo identifica como tal. El primer enlace permite saltar la información, llevando a un marcador ubicado al terminar el grupo.

Las opciones de búsqueda deben permitir diferentes tipos de búsquedas para diversos niveles de habilidad y preferencias

Si incluimos una búsqueda en el sitio, debemos brindar una interfaz que le sea útil al usuario, para lograrlo tenemos que hacer lo siguiente:

- Ofrecer interfases de búsqueda en todas las páginas del sitio. Es recomendable ubicarlo en el mismo sitio y con el mismo estilo para que el usuario lo encuentre en todas las páginas.
- Brindar mecanismos que faciliten las acciones de los usuarios. Por ejemplo, si una búsqueda no diera ningún resultado, ofrecer otras alternativas, información sobre cómo ampliar las posibilidades de búsqueda, etc.
- Debido a que muchos usuarios no miran la pantalla cuando escriben con el teclado y lo hacen recién al terminar, es necesario proveer un campo de texto lo suficientemente ancho que permita ingresar varias palabras o palabras largas. Así el usuario podrá ver la sentencia completa y corregirla si así lo desea.

- Deberíamos incluir una página de búsquedas avanzadas para aquellos usuarios que necesitan ampliar o refinar la búsqueda, más allá de las opciones que ofrece la búsqueda simple. Es mejor hacerlo en una página aparte, que sea accesible desde el cuadro de búsqueda que está presente en todas las páginas.

Ubicar la información destacada al principio de los encabezamientos, párrafos, listas, etc.

Los encabezados deben describir en pocas palabras la información que contienen, para orientar al usuario y pueda decidir si le sirve o no continuar leyendo.

Cuando se escribe para internet hay que tener en cuenta que, aparte de un título claro, se debe comenzar con un resumen corto e ir agregando gradualmente la información para que los usuarios puedan tener una idea global del contenido, aunque no terminen de leer por completo.

Para ayudar a una mejor navegación se recomienda desarrollar un concepto principal por párrafo y evitar utilizar palabras poco comunes, como la jerga, optando por un lenguaje más coloquial, con el uso de palabras más comunes y entendibles para los usuarios, además de evitar también frases o estructuras complicadas.

Permitir saltar sobre un ASCII art de varias líneas

Cuando se usa ASCII art (dibujos con caracteres) se tiene que proveer de un ancla para poder saltarlo, y en caso que el dibujo proporcione información que enriquezca el contenido se puede agregar el atributo longdesc para ampliar la descripción del gráfico.

Por ejemplo:

```
<a href="#salto">Saltar ASCII art</a>
```

```
  ,,  
____( )____  
/         \  
^\\|\\|\\^\\  
  ^^
```

```
<a name="salto">Continúa texto después del dibujo.</a>
```

Siempre es preferible utilizar imágenes antes que el ASCII, ya que a éstas se les puede agregar información por medio del tag alt.

Diferenciar y transmitir información acerca de la composición de los documentos

Cuando un documento se extiende por varias páginas, como en el caso de un libro dividido en capítulos o un libro como parte de una colección, no podemos prever en qué parte ingresarán los usuarios, entonces tenemos que agregar información a nuestro contenido que nos permita presentarlo como un paquete coherente.

Para agregar este tipo de información en las páginas, podemos usar el metadato link, que nos ayudará a describir la estructura de la navegación. Por ejemplo, si por intermedio de un buscador, el usuario cayó en el capítulo 7 del libro, la página podría tener la siguiente información:

```
<head>
<title>Chapter 7</title>
<link rel="Index" href="index.html">
<link rel="Next" href="chapter8.html">
<link rel="Prev" href="chapter6.html">
</head>
```

Otra alternativa es ofrecer el paquete comprimido, en formatos zip, tar, rar, etc., para que pueda ser descargado y leído sin conexión.

14. ASEGURARSE DE QUE LOS DOCUMENTOS SEAN CLAROS Y SIMPLES

Crear un estilo de presentación que sea coherente en todas las páginas

El sitio debe verse y sentirse igual a través de todas sus páginas, así el usuario sabrá que todavía se encuentra en el mismo sitio, las estructuras deben mantenerse para familiarización, comodidad y rápido movimiento del usuario entre las páginas.

Las hojas de estilo tienen gran potencialidad pero también naturaleza diversificada, y esto puede resultar peligroso, por eso es necesario tenerlas bajo control.

Se recomienda:

- Utilizar pocas hojas de estilo, documentarlas y mantener la estructura actualizada.
- Evitar las hojas de estilos incrustadas en las páginas si los estilos se van a repetir en otras páginas.

La estructura de clases, ids y elementos debe ser coherente y repetible a través de las hojas de estilo utilizadas.

Consultoría y Desarrollo

Serviweb S.L.

Calle Tierno Galván 11, 11, izq.
C.P. 30203 - Cartagena (Murcia)
Tels.: 902 929 442 / 968 086 993

Fax: 968 086 934

info@serviweb.es

www.serviweb.es