

Prototipo Accesible (AA) para Normaweb+

IDENTIFICACIÓN

Proyecto	Normaweb+
Nombre del Documento	normaweb+-tec-20060629-01.odt
Autor	Inmaculada Bermejo Salar Juan Luis Serradilla Amarilla
Versión Actual	1.1
Fecha de la Versión	29 de junio de 2006

RESUMEN

Este documento explica cómo se ha hecho el nuevo prototipo de Normaweb+, de modo que siga las pautas de accesibilidad del W3C (adecuación AA). También da recomendaciones para mejorar la accesibilidad y la usabilidad de las páginas web.

VERSIONES

Versión	Fecha	Autor	Descripción
1.0	26/06/2006	Inmaculada Bermejo Salar	Versión inicial sin maquetar, con apartados Explicación, Accesibilidad, Usabilidad y Normaweb
1.1	29/06/2006	Juan Luis Serradilla Amarilla	Maquetación e inclusión de los apartados de Introducción, Objetivos, Herramientas y Pautas

Índice de contenido

1.INTRODUCCIÓN.....	3
2.OBJETIVOS.....	3
3.EXPLICACIÓN DEL PROTOTIPO.....	4
3.1. Esquema del documento HTML del prototipo.....	4
3.2. Separación de contenido y la presentación.....	5
3.3. Separación de contenido y comportamiento.....	5
3.4. Semántica.....	5
3.5. Cambios gráficos.....	5
4.ACCESIBILIDAD.....	6
5.USABILIDAD.....	8
6.NORMAWEB.....	9
7.HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES WEB ACCESIBLES.....	9
8.PAUTAS DE ACCESIBILIDAD A LOS CONTENIDOS WEB DEL W3C.....	10

1. INTRODUCCIÓN

Aquellos que no estén familiarizados con los problemas de accesibilidad relacionados con el diseño de páginas Web, deben tener en cuenta que muchos usuarios pueden estar operando en contextos muy diferentes al suyo propio:

- Pueden no ser capaces de ver, escuchar, moverse; o pueden no ser capaces de procesar algunos tipos de información.
- Pueden tener dificultad en la lectura o comprensión de un texto.
- No tienen por qué tener o ser capaces de usar un teclado o un ratón.
- Pueden tener una pantalla que sólo presenta texto, una pantalla pequeña o una conexión lenta a Internet.
- Pueden no hablar o comprender con fluidez el idioma en que esté redactado el documento.
- Pueden encontrarse en una situación en la que sus ojos, oídos o manos estén ocupados u obstaculizados (Por ejemplo, conduciendo un automóvil, trabajando en un entorno ruidoso,...)
- Pueden tener una versión anterior del navegador, un navegador completamente diferente, un navegador de voz o un sistema operativo distinto.

Además, desde el 1 de Enero del 2006, las administraciones públicas, deben cumplir las recomendaciones de accesibilidad de los contenidos web propuestas por el W3C (adecuación AA). En este sentido la LSSICE (LEY 34/2002, de 11 de Julio entrando en vigor el 12 de Octubre de 2002) (Ley de Servicios de la Información y Comercio Electrónico) dice:

Quinta. Accesibilidad para las personas con discapacidad y de edad avanzada a la información proporcionada por medios electrónicos.

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

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

Siguiendo las pautas de accesibilidad recomendadas por el W3C, además de cumplir con la legislación vigente, conseguiremos hacer que nuestras aplicaciones web sean más asequibles para todos los usuarios, cualquiera que sea el navegador que esté utilizando (navegador de sobremesa, de voz, teléfono móvil, PC de automóvil, etc.), o las limitaciones bajo las que opere (entornos ruidosos, habitaciones infra o supra iluminadas, entorno de manos libres, etc.). Seguir estas pautas ayudará también a que cualquier persona encuentre la información que busca más rápido.

2. OBJETIVOS

El objetivo del presente documento es doble:

- Explicar cómo se ha hecho el prototipo de aplicación web, de modo que sea accesible (adecuación AA).
- Mostrar una serie de buenas prácticas, con el fin de conseguir cumplir con las pautas de accesibilidad (adecuación AA).

3. EXPLICACIÓN DEL PROTOTIPO

Hemos construido un nuevo prototipo, siguiendo las pautas de accesibilidad del W3C (adecuación AA), que además ha sido debidamente validado. Estas pautas pueden consultarse, en su versión original en inglés, en "<http://www.w3.org/TR/WAI-WEBCONTENT/>". Además existe una traducción al castellano, que se puede leer en "http://www.discapnet.es/web_accesible/wcag10/WAI-WEBCONTENT-19990505_es.html".

Se puede ver al prototipo en la dirección "<http://www.um.es/atICA/documentos/prototipos/normaweb/>".

3.1. Esquema del documento HTML del prototipo.

Uno de los principales problemas del prototipo estribaba en que, debido al uso de tablas para maquetar, tanto los formularios como los menús y disposición de la página, el código HTML se hacía bastante complejo, más difícil de comprender y mantener, y de adaptar para las diferentes aplicaciones a las que tenía que servir.

Por eso la primera medida a tomar ha sido la organización y maquetación del prototipo de la forma más sencilla posible. Se han eliminado todos los elementos posibles y se ha organizado la página con los siguientes:

- **div#contenedor:** contiene al resto del código, se utiliza para restringir el ancho de la página de forma sencilla.
- **div#cabecera:** contiene un h1 con la imagen del escudo de la Universidad y el texto alternativo institucional "Universidad de Murcia", y un h2 con el nombre de la aplicación o la imagen de su logotipo.
- **div#menu:** en este caso utilizamos un elemento div para agrupar los diferentes niveles de menú. Para cada uno se utilizará una lista, teniendo por ahora dos tipos

- ul.uno: clase para los menús de primer nivel

- ul.dos: clase para los menús de segundo nivel

Además, disponemos de la clase li.activo para indicar la opción de menú en la que nos encontramos en ese momento.

- **div#contenido:** en esta capa pondremos el contenido propio de la página en cuestión. Dentro de ella se podrán utilizar todos los elementos HTML, crear nuevos menús, encabezados y así debemos hacerlo. Además se han añadido estilos para los formularios, párrafos y listas que se aplicarán automáticamente al elegir la hoja de estilo del prototipo. Hay algunas clases especiales para casos que se producen frecuentemente:

- div.info: mensajes informativos o para aclarar el funcionamiento de una página, como por ejemplo, indicación de que los campos requeridos de un formulario están marcados de determinada manera, o alguna otra aclaración.

- div.avisoatica: avisos sobre las aplicaciones (como por ejemplo, el mensaje de mantenimiento).

- div.exito: para cuando una acción se ha llevado a cabo con éxito.

- div.error: para cuando una acción ha producido un error.

- .obligatorio: para marcar los campos obligatorios de un formulario. Se aplica al elemento *label* del elemento que corresponda. También se dispone del elemento *em* para marcar el elemento distintivo de los campos obligatorios (en este caso, el asterisco).

- .bloque: esta clase se ha creado por comodidad, sirve para poder aislar un elemento cuando sea necesario, sustituyendo al elemento *br* que está en desuso.

- div.uno, div.dos, div.tres, div.cuatro: se han dispuesto estas clases para evitar maquetar el contenido con tablas (menos accesibles y menos flexibles). Se puede utilizar cuando se quiere disponer el contenido por columnas (cada clase hace referencia al número de columnas por línea) aunque no siempre es necesario.

Pueden verse ejemplos claros de su uso en los formularios de entrada construidos para el prototipo.

- **div#pie:** la información del pie de página se ha organizado en dos secciones:

- `ul#ayuda`: iconos de ayuda para la navegación, se trata como un menú y tiene su propio formato.
- `div#meta`: información sobre la propia página. Copyright e información de validación
 - `ul#validacion`: iconos de validación para las páginas que lo cumplan, con los enlaces a donde corresponde según la documentación Normaweb.

3.2. Separación de contenido y la presentación

De cara a la adecuación a los nuevos estándares, así como para tener mayor flexibilidad y facilitar la validación de accesibilidad, no debe usarse ningún elemento que proporcione formato en el código HTML.

Hay que distinguir entre formato y contenido, especialmente cuando se trata de imágenes, y proporcionar la descripción adecuada cuando se trate de información valiosa.

Para lograr separar el contenido y la presentación se utilizan las hojas de estilo, donde podemos especificar **cualquier** propiedad relativa al aspecto de un elemento.

Así mismo, se procurará reservar algunos formatos como la negrita y la cursiva para el propósito que tienen, es decir, enfatizar, al menos en la medida de lo posible, no abusar de ellos.

Es importante recordar que, en los controles de formularios, el atributo *size* puede considerarse contenido, ya que debe ser indicativo del tamaño aproximado que pueden tener los datos que se deban escribir, y en este sentido ayuda al usuario a hacerse una idea (por ejemplo, en controles para fechas). Así podemos especificar el tamaño del control sin tener que recurrir a propiedades CSS.

3.3. Separación de contenido y comportamiento

Especialmente problemático para la accesibilidad es el aspecto del comportamiento en el cliente (navegador) de las aplicaciones, controlado mediante javascript.

Además de para mejorar la accesibilidad, es muy conveniente separar la capa de comportamiento de la de contenido de la misma forma que hicimos con la presentación.

Para ello, podemos ver el ejemplo de la validación de campos obligatorios en los formularios de entrada y sugerencias del prototipo. No hay ninguna línea ni llamada a función javascript en el propio formulario, sino que está relacionada en el fichero javascript que contiene las propias funciones.

3.4. Semántica

- Uso de encabezados, párrafos y listas para el contenido. Es importante utilizar los niveles de los encabezados (h1, h2, h3) para marcar los niveles de profundidad del contenido asociado a ellos en el documento, en lugar de para dar formato.
- Evitar la maquetación con tablas.
- Uso de los elementos de agrupación para formularios:
 - Fieldset: agrupación de campos relacionados entre sí. El elemento *legend* servirá para etiquetar el *fieldset* como si fuera su título correspondiente.
 - Label: imprescindible para lograr que el formulario sea accesible, porque relaciona el texto escrito con el control del formulario y será leído junto con él al saltar por los diferentes campos del formulario con las teclas.

3.5. Cambios gráficos

Se pueden observar diversos cambios gráficos que responden principalmente a requerimientos de accesibilidad y usabilidad.

- El más notable de ellos es el **cambio en los colores utilizados**. Se ha tomado esta determinación con el fin de conseguir el contraste adecuado entre color de fondo y primer plano, según las fórmulas publicadas por el W3C.
- También se han creado algunos bloques, como la delimitación de los formularios aplicando un color de

fondo y borde, para centrar la atención del usuario en la zona en concreto en la que tiene que interactuar.

- Se han suprimido todas las imágenes que sustituían texto. Esto debe procurarse en la medida de lo posible, ya que así dejamos al usuario que controle el tamaño del texto, así como el contraste, y nos aseguramos que el texto será leído (previniendo, por ejemplo, olvidos en etiquetas alt para imágenes) y que economizamos al máximo en la descarga de la página.
- El color de los enlaces ha pasado a ser azul. En este cambio no sólo contamos con el contraste (anteriormente insuficiente) sino también juega un papel el estándar de facto en el que se ha convertido el color azul para los enlaces (color predeterminado). Aun así, esto no es determinante, mientras el color tenga el contraste adecuado. También se ha considerado que los enlaces visitados tengan un color diferente.
- Además del color, los enlaces han pasado a estar subrayados y a tener un tamaño de letra normal, reservando la negrita para los casos en los que queramos hacer énfasis, poder distinguirlo visualmente de un enlace normal.
- Leve disminución del tamaño de la fuente. Se conserva un tamaño razonable pero que permita optimizar el espacio disponible a 800x600. El tamaño se especifica con em, unidad escalable que permite al usuario modificarlo en cualquier navegador.

4. ACCESIBILIDAD

Consejos para conseguir que una página sea accesible.

- Uso adecuado de HTML:
 - Validar el código según los estándares de W3C.
 - Especificar siempre el tipo del documento, el idioma y la codificación del documento.
 - Asegurar que todos los **elementos** se usan para proporcionar estructura al documento (*h1...h6, p, ul, ol, em, strong, img...*). Evitar elementos como *b, font, i...* Ejemplo: usar *em* en lugar de *i* para enfatizar un texto. Usar *blockquote* o *q* junto con el elemento *cite* para citas textuales.
 - Asegurar que todos los atributos se usan para proporcionar estructura (*title, alt, src, size, lang...*). Evitar atributos como *background, width, padding, cellpadding...* todos pueden especificarse después con CSS.
 - Usar los elementos estructurales para la función que les corresponde:
 - Títulos para aportar una jerarquía al documento
 - Tablas para listar datos tabulares (2 columnas o más, en otro caso, usar listas). A las tablas de datos hay que darles la estructura adecuada, con los elementos y atributos que las hacen accesibles:
 - `summary`: resumen del contenido de la tabla. Ayuda para saltar contenido que no interese al usuario
 - `caption`: título de la tabla
 - `thead, tbody`: identifican diferentes secciones de la tabla, la cabecera y el cuerpo
 - `th`: celda de encabezado en la cabecera de la tabla
 - `td`: celda de datos del cuerpo de la tabla

Ejemplo de tabla accesible:

```
<table summary="Porcentaje de uso de navegadores en el sitio web de
la Universidad de Murcia">
<caption>Navegadores más usados</caption>
<thead>
<tr>
<th scope="col">País</th>
<th scope="col">Internet Explorer</th>
```

```
<th scope="col">Mozilla Firefox </th>
<th scope="col">Opera</th>
<th scope="col">Otros</th>
</tr>
</thead>
<tbody>
<tr>
<td>España</td>
<td>60%</td>
<td>25%</td>
<td>10%</td>
<td>5%</td>
</tr>
</tbody>
</table>
```

- Listas para listar elementos y para dar estructura a los menús
 - Aportar elementos y atributos que ayudan a la accesibilidad:
 - Elementos: *abbr, acronym, caption, thead, th, label, fieldset...*
 - Atributos: *title, summary, accesskey, tabindex, alt...*
- Separación de la presentación en hojas de estilo aparte:
 - En el documento no debe existir ningún elemento ni atributo que proporcione información de decoración.
 - Asegurarnos **siempre** de que el contraste entre fondo y primer plano es suficiente.
 - Evitar basarnos **sólo en el color** para distinguir elementos (ej. enlaces del resto del texto) o estados (*rollovers*). Hay que utilizar algún elemento distintivo más, por ejemplo el subrayado, la negrita o un tamaño de texto mayor.
 - Utilizar unidades adecuadas para el tamaño del texto: *em*, porcentajes (con un tamaño base de *em*)
 - Es importante distinguir entre imágenes decorativas e imágenes que proporcionan información. Las primeras siempre deberán ir en la hoja de estilo, y las segundas siempre deberán tener su alternativa textual adecuada.
 - Proporcionar varias hojas de estilo para distinguir los diferentes medios, si se prevee que la aplicación puede requerirlo: *screen, print, handheld, projector, all...*
 - Para usar efectos rollover en los menús, se usa CSS, nunca javascript.
- Identificar cambios en el lenguaje: la página tiene definido un idioma en su cabecera. Si a lo largo del texto se produce un cambio en este idioma, debe notificarse:

```
<p>Y entonces el extranjero le dijo:<q lang="en">How are you?</q></p>
```

- Proporcionar contenido alternativo para **todos los elementos no textuales**:
 - Imágenes: atributo *alt* suficientemente descriptivo. No usar por ejemplo, la descripción “Logotipo” o “Gráfico” y similares. Usar *longdesc* si *alt* no es suficiente.
 - Scripts, javascript: las páginas deben ser funcionales sin javascript. Usar el elemento *noscript* de forma eficiente y no como mera información, cuando sea posible. Realizar siempre las validaciones tanto a nivel de cliente como en servidor. Es **muy importante** reducir el uso de javascript para lo estrictamente necesario. Evitar a toda costa su uso cuando hay elementos HTML disponibles para realizar la misma interacción (por ejemplo un enlace).
 - Objetos embebidos (flash, java...): se debe usar el elemento *object* en lugar de *embed*. El **contenido alternativo** se pondrá dentro del objeto de la siguiente manera:

```
<object title="Universidad de Murcia" type="application/x-shockwave-
flash" data="http://www.um.es/imagenes/home/cabeceraum2.swf"
width="730" height="82">
<param name="movie"
```

```
value="http://www.um.es/imagenes/home/cabeceraum2.swf" />
<param name="quality" value="high" />
<h1>Universidad de Murcia</h1>
<h2>Formación, prestigio, investigación, tradición</h2>
</object>
```

- Marcos: evitar su uso en la medida de lo posible. Si es inevitable, utilizar su atributo *title* para indicar lo que contienen y facilitar la navegación entre ellos. Para proporcionar la alternativa se usará el elemento *noframes*, proporcionando desde ahí acceso a una versión sin marcos, o bien a cada una de las páginas que se accede a través del conjunto de marcos.

5. USABILIDAD

Algunas cuestiones relativas a la usabilidad:

- Legibilidad:
 - Tipo y tamaño de letra: para pantalla, sans-serif de un tamaño equivalente a unos 10 puntos, unos 0.9 *em* (cambiará según la familia de fuente escogida)
 - Bloques de texto: no crear líneas demasiado largas (entre 60 y 80 caracteres) y no crear párrafos demasiado grandes. En pantalla se leen más cómodamente los textos cortos.
 - Evitar el uso de controles de formulario para **mostrar** texto no editable, producen una sensación engañosa y son incómodos.
- Organización de la página: considerando que los usuarios leen de arriba a abajo y de izquierda a derecha, los estándares de facto más extendidos para la maquetación en entorno web son los siguientes:
 - Logotipo/identificación de la entidad: esquina superior izquierda. Con enlace de vuelta a la página principal.
 - Menús de navegación/opciones: horizontal o vertical en la esquina superior izquierda bajo el logotipo. Se puede duplicar de forma discreta en el pie de página.
 - Buscador y menús de ayuda: zona superior derecha.
- Proporcionar ayudas para la ubicación y navegación:
 - Rastro de migas
 - Mapa del web
 - Enlaces con texto suficientemente descriptivo. Si es necesario, usar el atributo *title* para describir lo que pasará al pulsar el enlace.
 - Enlaces fácilmente distinguibles del texto que los rodea. Intentar seguir el estándar de facto de color azul y subrayado, o al menos uno de los dos.
 - Proporcionar **atajos de teclado**:
 - En formularios, es útil indicar el orden del tabulador para cada campo (*tabindex*).
 - En menús se puede optar por teclas de acceso directo *accesskey* pero hay que tener cuidado con no sustituir a las propias del sistema operativo (alt+tecla). Se recomienda usar números.
- Al manipular el comportamiento del navegador a través de la página:
 - Evitar sustituir el botón atrás por un enlace con javascript: para ello está el botón. Si se quiere poner un enlace, usar un elemento HTML normal que lleve a la página que corresponda (por ejemplo, en un rastro de migas).
 - Evitar modificar el aspecto del navegador eliminando los menús y botones, o deshabilitando el botón derecho del ratón.
 - Evitar refrescar automáticamente la página sin intervención del usuario en la medida de lo posible (realizar los cambios con una interacción con el servidor).
 - Si se usa Ajax, facilitar la percepción del cambio que se produce utilizando elementos gráficos.
 - Evitar el uso del atributo *target* de los enlaces, y la creación de ventanas pop-ups en la medida de lo posible. Dejar al usuario la capacidad de decidir dónde quiere abrir el enlace.

- Proporcionar suficiente información al usuario:
 - Advertir de los **requisitos necesarios** (ej. marcos, javascript) para utilizar una aplicación al usuario **antes** de pedir que se identifique para entrar en ella.
 - Poner especial atención en los mensajes de respuesta del sistema. En el prototipo hemos proporcionado tres tipos de mensaje (éxito, error e información) que pueden ayudar para distinguir los mensajes de respuesta.
 - A la hora de poner ficheros disponibles para que el usuario descargue, avisar el tipo de fichero que es y su tamaño.
 - En formularios:
 - Advertir de cómo se señalizan los campos obligatorios **antes** de comenzar.
 - Señalar que un campo es obligatorio **antes de mostrar su etiqueta**, e incluirlo en el elemento *label* para poder ser leído al navegar el formulario por teclado.

6. NORMAWEB

Hemos actualizado Normaweb ("<http://www.um.es/atika/mnncs/normaweb>"), de modo que se sigan las pautas de accesibilidad del W3C (adecuación AA). Además, se han introducido (en Normaweb) las siguientes recomendaciones :

- A la hora de usar imágenes con degradados, asegurar la adecuada legibilidad del texto de la imagen.
- Evitar en la medida de lo posible la sustitución de texto por imágenes.
- Fuente y tamaño: ver apartado anterior referente a accesibilidad.
- Evitar la maquetación con tablas para los formularios. Deben utilizarse los elementos *fieldset* para la agrupación de campos relacionados, y *label* para relacionar un control con su etiqueta. El control debe ir **dentro** de la etiqueta label. Hay disponible una clase para señalar los campos obligatorios (ver prototipo).
- Marcar los menús con listas desordenadas. Siempre que hay más de una opción, hay que usarlas o no se pasará AA.
- Evitar, a ser posible, decir que una aplicación está **optimizada** para un navegador concreto. En caso de que sea necesario, indicar una lista de navegadores en los que ha sido probada. Si la página es accesible, funcionará en la gran mayoría.
- Utilizar el atributo *title* para los “tooltips” en imágenes o enlaces. El atributo *alt* sin embargo es un texto **sustitutivo** por si la imagen no saliera, que tiene que estar presente siempre.

7. HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES WEB ACCESIBLES

A continuación, suministramos las direcciones de algunas de las **herramientas** más utilizadas, en el desarrollo de aplicaciones web accesibles:

- **Barras de herramientas** de accesibilidad para el navegador:
 - [Internet Explorer](#)
 - [Firefox/Mozilla](#)
- [Colour Contrast Analyser](#)
 - [Extensión para Firefox](#)
- Herramientas de validación de las pautas de accesibilidad propuestas por el W3C:
 - [HERA: utilidad para revisar la accesibilidad de acuerdo con las recomendaciones del W3C.](#)
 - [Extensión para Firefox](#)
 - [TAW: Test de Accesibilidad Web](#)
- [GRAYBIT](#): para ver una página web en blanco y negro (escala de grises).

8. PAUTAS DE ACCESIBILIDAD A LOS CONTENIDOS WEB DEL W3C

La [Iniciativa de Accesibilidad en la Web \(WAI\)](http://www.w3.org/WAI) del W3C se puede encontrar en "http://www.w3.org/WAI". A través de esta iniciativa, [el W3C establece unas pautas de accesibilidad al contenido web](http://www.w3.org/TR/WAI-WEBCONTENT), que se pueden consultar en "http://www.w3.org/TR/WAI-WEBCONTENT".

Existe una excelente [traducción al castellano de las pautas de accesibilidad del W3C](http://www.discalpnet.es/web_accesible/wcag10/WAI-WEBCONTENT-19990505_es.html), en la dirección "http://www.discalpnet.es/web_accesible/wcag10/WAI-WEBCONTENT-19990505_es.html". Además, en la misma web se pueden encontrar una interesante serie de [documentos para el diseño accesible de contenidos en la Web](http://www.discalpnet.es/web_accesible/) ("http://www.discalpnet.es/web_accesible/"). Entre los documentos citados están los siguientes (que son traducciones al castellano de los originales y oficiales del W3C):

- [Preguntas más frecuentes sobre las Pautas de Accesibilidad al Contenido en la Web 1.0.](#)
- [Guía breve para crear sitios Web accesibles.](#)
- [Pautas de Accesibilidad al Contenido en la Web 1.0.](#)
- [Tabla de puntos de verificación para las Pautas de Accesibilidad al Contenido en la Web 1.0.](#)
- [Técnicas para las Pautas de Accesibilidad al Contenido en la Web 1.0.](#)
 - [Técnicas Fundamentales.](#)
 - [Técnicas HTML.](#)
 - [Técnicas CSS.](#)