

XHTML, una llamada al orden para el HTML

Ramón Montero

En la actualidad, todos sabemos algo del HTML (*HyperText Markup Language*), el “lenguaje de marcas” que se ha extendido por Internet y ha hecho posible la utilización de un “estándar” para la presentación de documentos. El propio éxito del HTML, que ha hecho proliferar las utilidades WYSIWYG para la creación y edición de páginas web y la “guerra” que han mantenido las principales empresas creadoras de los navegadores web (Microsoft y Netscape) por imponer sus “etiquetas” (*tags*), han dado lugar a un “lenguaje” muy fácil de manejar (sobre todo a través de los editores que generan el código de forma automática), pero que no sigue las mismas normas y estándares, por mucho que se empeñen “los chicos” del W3C (*World Wide Web Consortium*, Consorcio World Wide Web) en sacar especificaciones para su control.

Nota: Para comprobar los ejemplos que siguen a continuación, es imprescindible disponer de un navegador/visualizador de última generación.

Vamos a ver en este artículo la última solución adoptada para arreglar esta situación.

ORIGEN DEL XHTML

Los visualizadores (*browsers*) y navegadores han superado la mayoría de las diferencias comentadas en el

primer párrafo asumiendo la posibilidad de que en los códigos HTML existan errores, olvidos, fallos de organización, repeticiones innecesarias, etc., cuestiones lógicas en los códigos generados por los muchos usuarios no profesionales que trabajan documentos HTML y por los conseguidos a través de conversiones automatizadas desde otros formatos, ayudando así a la proliferación de páginas web con códigos no estandarizados y llenos de “basura”, pero que son perfectamente visibles en las últimas versiones de los navegadores.

Cualquier observador de la Web puede comprobar fácilmente que la mayoría de las páginas web existentes en Internet presentan código mezcla del estándar HTML y de las especificaciones particulares de los editores-navegadores utilizados en cada caso, siendo en algunos casos verdaderos ejemplos de mala programación y de dejadez, aunque sean visualmente aptos.

No es muy difícil deducir que la existencia y utilización de etiquetas no especificadas por las normas (en la actualidad, la última versión de la normativa que regula el código HTML es la 4) y el consentimiento de “faltas de gramática HTML” por los navegadores, lleva a un punto difícil de controlar, por lo que, aprovechando la inercia que ha generado (y que generará) la publicación del estándar XML (*Extensible Markup Language*), mucho más estricto con las reglas del código, los perse-

verantes e indismallables gestores del W3C están trabajando en unas reglas que terminen con parte de este desajuste actual.

Esta nueva normativa se denomina XHTML (*Extensible HyperText Markup Language*), y describe las especificaciones que deben tenerse en cuenta para generar un código estricto que no se salga de las reglas gramaticales que debe contener una página web HTML bien realizada.

Por supuesto que esta normativa no resuelve todos los problemas del HTML, como la existencia de etiquetas “propietarias” o el diferente soporte de CSS o JavaScript, pero sí ayudará a eliminar los errores gramaticales, unificando la descripción del código y facilitando la portabilidad de los documentos. Todo navegador que se precie y todo editor HTML que desee mantener un lugar de prestigio, deberá ajustarse a sus normas, que por otro lado son muy sencillas de seguir, como veremos más adelante.

En realidad, el usuario no notará nada en especial si decide generar código XHTML en vez de HTML, ya que las etiquetas no cambian. Si realiza su diseño “a mano”, o sea, con un editor ASCII, solo tendrá que tener cuidado en seguir las reglas de la especificación. Si utiliza un editor WYSIWYG para crear sus páginas web, será el propio editor el encargado de generar el código adecuado (siempre que tenga esa opción), tal como ocurre en los editores actuales.

En cuanto a los navegadores, cuando lean la línea de código que especifica la adecuación a las normas del XHTML, aplicarán el DTD (*Document Type Definition*) correspondiente y el *parser* (analizador) adecuado, menos permisivo que el que aplican en la actualidad, pero de riguroso estándar, aunque siempre con total transparencia para el usuario.

RELACIÓN CON HTML Y XML

Se puede decir sin lugar a dudas que el XHTML está perfectamente interrelacionado con el XML y HTML, cogiendo lo mejor de cada uno, o sea, las conocidas y extendidas etiquetas del HTML (versión 4) y la estricta normativa del XML (versión 1).

Matemáticamente, se podría decir que:

XML + HTML = XHTML (*más o menos*)
Expresión poco técnica, pero aclaratoria.

Por si existe alguna duda en cuanto a la paternidad y origen de los estándares XML, HTML y XHTML, se pueden resumir en:

- XML es una simplificación del SGML (*Standard Generalized Markup Language*), eliminando todo lo que no es necesario para su utilización en Internet, pero manteniendo sus características más potentes e importantes. Es un metalenguaje, esto es, un lenguaje capaz de generar otros lenguajes.
- HTML es un lenguaje de marcas, subconjunto del SGML, diseñado para publicar documentos en la Web con la máxima sencillez.
- XHTML es una reformulación de HTML 4 para adaptarse a las normas del XML.

Aunque los orígenes son los comentados, ya he expuesto que la situación actual, en lo que respecta al HTML, no coincide con la idea original. Por este motivo, el W3C ha sido el responsable de tomar la decisión de reformular el HTML 4 para adaptarse al XML (solución muy fácil), en vez de crear un nuevo HTML que volviese al redil del SGML (solución mucho más difícil de imponer) o aconsejar que se utilice el ya existente SGML (realmente mucho más complejo y difícil de utilizar).

Las razones esgrimidas por el W3C para aconsejar el uso del XHTML son dos, principalmente:

1. XHTML, ya que es una aplicación XML, ha sido diseñado para ser ampliable (de ahí el añadido de la palabra *Extensible*). Esto significa que se pueden añadir nuevas etiquetas o elementos sin alterar la DTD en la que está basado el análisis del documento.
2. XHTML ha sido diseñado pensando en la portabilidad. Aunque hoy en día la unión de la potencia de los ordenadores y de los navegadores es suficientemente para asumir las posibles diferencias y pequeños errores del código HTML, se espera que para los próximos años se produzca un aumento considerable de los aparatos que sean capaces de tratar información en código HTML, no disponiendo estos de dicha potencia. Televisores, teléfonos móviles, ordenadores de bolsillo, calculadoras, hornos, tostadoras, etc., soportarán código HTML, siempre que esté realmente unificado y se ajuste a normas estrictas para no dar problemas que exijan soluciones complejas.

Después de lo dicho, creo que ya es el momento de ver las diferencias entre el XHTML y el HTML.

DIFERENCIAS CON HTML

Las normas que regulan el código XHTML son suficientemente sencillas como para no asustar a nadie, sobre todo si se es un profesional del diseño web y/o de la programación.

Las diferencias principales entre el clásico HTML y el nuevo XHTML son:

Toda la descripción del código debe estar en minúsculas.

Mientras el XML es sensible a utilización de las mayúsculas y de las minúsculas (las etiquetas <COCHE>, <Coche> y <coche> son diferentes) y el HTML es indiferente a la utilización de ambos tipos de letras (las etiquetas del ejemplo del coche serían iguales), las etiquetas del código XHTML deben estar siempre en minúsculas.

Esto no es problema alguno para los diseñadores que trabajan directamente con editores sencillos, y si crean sus documentos con editores automatizados, la mayoría de ellos soportan opciones capaces de pasar elementos definidos en mayúsculas a minúsculas de forma automática, realizando la conversión de un documento antiguo al nuevo estándar sin intervención del usuario. En la figura 1 se pueden ver algunas posibilidades útiles de Office 2000.

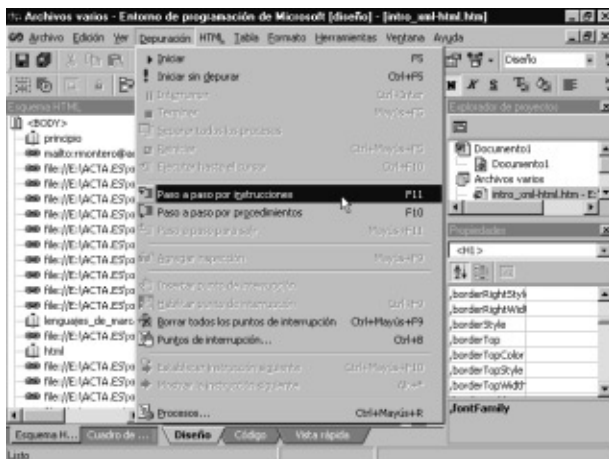


Figura 1. Al igual que ocurre con otros editores web, Office 2000 dispone de herramientas de depuración de código muy útiles para conseguir documentos XHTML.

En el cuadro siguiente se puede ver un ejemplo práctico sobre este punto.

HTML	XHTML
<BODY BgColor= "#FFFFFF">	<body bgcolor= "#ffffff">

En este punto existe una excepción, ya que los valores de los atributos definidos por el usuario pueden estar igualmente en mayúsculas como en minúsculas. En el ejemplo anterior, el código XHTML podría ser <body bgcolor= "#FFFFFF"> sin mayor problema.

Todos los valores de los atributos deben ir entrecomillados.

Ya no se permiten ambigüedades ni olvidos con respecto a la descripción de los valores de los atributos. Aunque sean numéricos, deben ir entre comillas, dobles (") o sencillas (').

En el cuadro siguiente se puede ver un ejemplo práctico sobre este punto.

HTML	XHTML
<TABLE BORDER= 2> ...	<table border= "2"> ...

Todos los elementos "no vacíos" deben ir entre la etiqueta de principio y la etiqueta de final.

Todos los diseñadores acostumbrados a poner una única etiqueta <P> para terminar un párrafo deben olvidarse de esa costumbre, ya que en XHTML es obligatorio utilizar la etiqueta de principio <P> y la de final </P>. Esto es aplicable a todos los casos, incluidos los , <DT> y <DD>, que ahora deben definirse como ... , <dt> ... </dt> y <dd> ... </dd>.

En el cuadro siguiente se puede ver un ejemplo práctico sobre este punto.

HTML	XHTML
Texto 1<P>	<p>Texto 1</p>
Texto 2<P>	<p>Texto 2</p>
...	...

Los elementos "vacíos" deben llevar terminación.

Un elemento vacío, como su propio nombre indica es el que no tiene contenido. Lo normal es que los elementos sí tengan contenido entre las etiquetas de principio y de final, y así, las etiquetas <p> y </p> contienen un párrafo, las etiquetas <i> y </i> contienen un

XHTML, una llamada al orden para el HTML

texto en cursiva, las etiquetas `<table>` y `</table>` contienen una tabla, etc.

No obstante, en HTML también existen algunos elementos que no contienen nada, como `
`, `<hr>` e ``, por lo que solo existen como etiquetas únicas, que hacen las veces de principio y de final. Algunas de estas etiquetas vacías admiten atributos, como "src" en ``, pero eso es otra cuestión distinta.

Pues bien, en XHTML no se permite la existencia de elementos sin terminación, por lo que los elementos vacíos incluyen su propia terminación en la misma etiqueta. El problema se resuelve añadiendo un "espacio" y una "barra" (/) justo antes del signo "mayor" (>).

Según lo dicho, los ejemplos anteriores quedan en XHTML como se muestra en el cuadro siguiente.

HTML	XHTML
<code>
</code>	<code>< br /></code>
<code><HR></code>	<code>< hr /></code>
<code></code>	<code></code>

Todos los elementos deben estar anidados ordenadamente.

En HTML no hace falta tener especial cuidado en ordenar los anidamientos de las etiquetas (etiquetas dentro de otras etiquetas), siendo posible que existan solapamientos. Al igual que sucede con XML, en XHTML no se permiten tales libertades, debiendo tener especial cuidado en el orden en el que se realizan los anidamientos, y si una etiqueta de principio tiene el primer orden, otra el segundo y otra el tercero, por ejemplo, se deben situar las etiquetas de final de tal manera que primero se defina la del tercer orden después la del segundo y finalmente la del primero.

En el siguiente cuadro se puede ver un ejemplo práctico sobre este punto.

HTML	XHTML
Este texto está en <code><I></code> cursiva	<code><p></code> Este texto está en <code><i></code> cursiva
<code></code> cursiva-negrita <code></I></code> negrita	<code></code> cursiva-negrita
<code></code> y normal. <code><P></code>	<code></i></code> negrita <code></code> y normal. <code></p></code>

Los valores de atributos iguales sin variantes no pueden ser simplificados.

Algunos atributos de HTML solo pueden tener un único valor, por lo que se permite "minimizarlos", o sea, dejar solo el atributo (o el valor, ya que son iguales).

Esto es corriente con los elementos `<option>`, `<input>` y `<dl>`, y así, es muy usual encontrarnos con descripciones como `<option value="valor" selected>`, `<input type="tipo" checked>` o `<dl compact>`, cuando se tendrían que describir como `<option value="valor" selected="selected">`, `<input type="tipo" checked="checked">` o `<dl compact="compact">`.

Según lo dicho, los ejemplos anteriores quedan en XHTML como se muestra en el cuadro siguiente.

HTML	XHTML
<code><OPTION VALUE="valor" SELECTED><INPUT TYPE="tipo" CHECKED><DL COMPACT></code>	<code><option value="valor" selected="selected"><input type="tipo" checked="checked"><dl compact="compact"></code>

Existen elementos obligatorios.

A alguno le puede parecer un tanto quisquilloso este punto, pero en XHTML no se permite la ausencia de cualquiera de los elementos `<head>` y `<body>`.

También está regulado que `<title>` debe ser el primer elemento de la sección `<head> ... </head>`.

En el cuadro siguiente se puede ver un ejemplo práctico sobre este punto.

HTML	XHTML
<code><HTML></code>	<code><html></code>
<code><HEAD></code>	<code><head></code>
<code><STYLE>...</STYLE></code>	<code><title>...</title></code>
<code><TITLE>...</TITLE></code>	<code><style>...</style></code>
<code></HEAD></code>	<code></head></code>
<code></HTML></code>	<code><body></code>
	...
	<code></body></code>
	<code></html></code>

NOVEDADES

Las cuestiones comentadas en el anterior apartado son diferencias que hay que tener en cuenta, pero no implican nada nuevo. En este apartado vamos a ver las novedades que se encuentran en la normativa de XHTML con respecto al HTML.

Los documentos XHTML deben incluir una declaración de "tipo de documento".

Aunque esta norma ya existe en los documentos HTML, la verdad es que se utiliza en muy pocas ocasiones, siendo una novedad para muchos diseñadores web.

El motivo de la necesidad de esta declaración es dejar bien claro que nuestro documento se ajusta a una determinada DTD, definida por el W3C como "una colección de declaraciones XML que define la estructura, los elementos y los atributos que es posible utilizar en un determinado documento". En otras palabras, una DTD es una descripción de las normas que nos indica qué cosas pueden hacerse en nuestro documento y qué cosas no pueden hacerse.

La declaración de "tipo de documento" debe ser la primera línea de una página XHTML, delante incluso del elemento <html>.

Los documentos XHTML deben hacer referencia a una de las tres siguientes DTDs: *Strict*, *Transitional* o *Frameset*, siendo todas ellas unas aproximaciones, más o menos completas, a la especificación HTML 4. Sus formatos y características más importantes son:

Strict:

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

Se utiliza cuando se da formato a los textos a través de CSS (*Cascading Style Sheets*), o sea, cuando no se recurre a las etiquetas , <table> y similares para controlar la forma en la que los navegadores muestran el contenido del documento.

Transitional:

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

Se utiliza cuando no se describe la presentación de los documentos por medio de hojas de estilo en cascada, prefiriendo la descripción a base de etiquetas. Es el sistema adecuado para cuando se desea facilitar el acceso a usuarios con navegadores sin posibilidades de tratamiento de CSS.

Frameset:

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

Se utiliza cuando los documentos incorporan cuadros (*frames*).

En la figura 2, figura 3 y figura 4 se pueden apreciar las primeras líneas de los DTDs *Strict*, *Transitional* y *Frameset*, respectivamente.

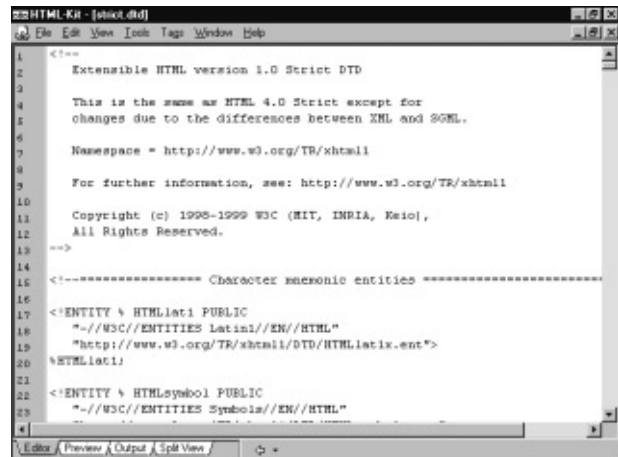


Figura 2. El archivo *strict.dtd* está disponible en la sede del W3C para poder ser analizado o utilizado por cualquier usuario.

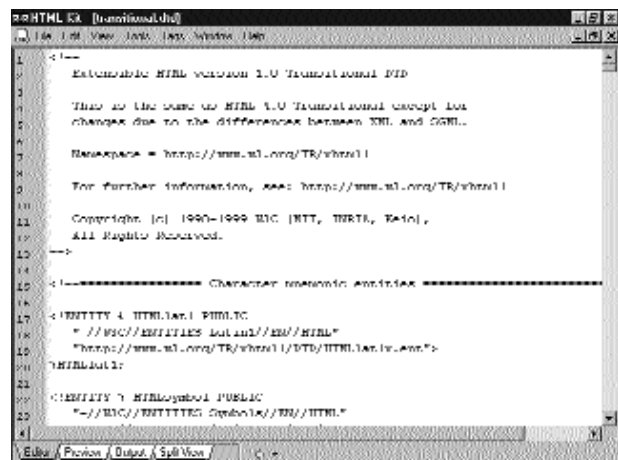


Figura 3. El archivo *transitional.dtd* está disponible en la sede del W3C para poder ser analizado o utilizado por cualquier usuario.

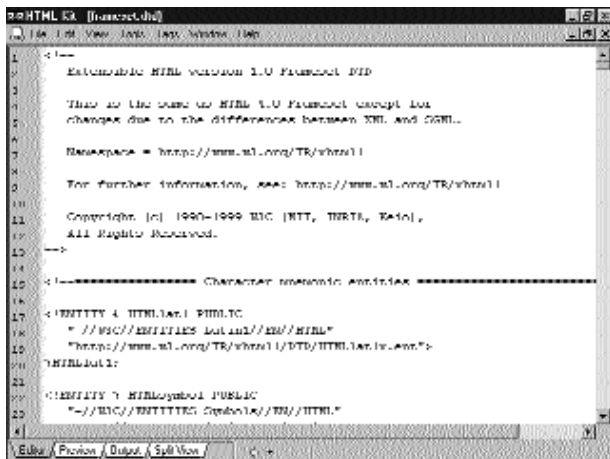


Figura 4. El archivo frameset.dtd está disponible en la sede del W3C para poder ser analizado o utilizado por cualquier usuario.

El elemento raíz

El elemento raíz de un documento XHTML debe ser siempre `<html>`. No puede existir nada antes de la etiqueta de principio `<html>` (salvo la declaración del tipo de documento). Tampoco se puede añadir nada después de la etiqueta de final `</html>`.

La etiqueta de principio `<html>` de un documento XHTML debe incluir un atributo que especifique el "espacio de nombre" (*namespace*) que utiliza el documento. El atributo es el mismo que se utiliza en XML, esto es: `xmlns`, siendo el valor de dicho atributo la palabra `xhtml` seguida del número 1 (uno).

Según el W3C, "un nombre de espacio XML es una colección de nombres, identificados por una referencia URI, que es utilizada en los documentos XML como tipos de documentos y nombres de atributos". Dicho más claro, el nombre de espacio XHTML es una lista con las etiquetas válidas que pueden ser utilizadas en un documento XHTML.

De acuerdo con lo dicho anteriormente, la etiqueta del elemento raíz será:

```
<html xmlns="http://www.w3.org/TR/xhtml1">
```

Los elementos `<script>` y `<style>`

Si dentro del código HTML se describen elementos que incluyen listados en lenguajes diferentes del HTML, como ocurre con los elementos `<script>` o `<style>`, XHTML exige que se acoten los guiones en una sección

CDATA. Las secciones CDATA ignoran el significado de los caracteres que incluyen, evitando problemas con entidades que puedan confundirse con las etiquetas del HTML, como ocurre con los delimitadores "`<`" y "`>`", por ejemplo.

El único delimitador que no puede ser utilizado dentro de los guiones es "`]]>`", ya que es que utiliza la propia sección CDATA para saber dónde finaliza su función.

En el cuadro siguiente se puede ver un ejemplo práctico sobre este punto.

HTML	XHTML
<pre><SCRIPT LANGUAGE="JavaS- cript"><!-- document.write("<P> Texto de prueba</P>");//-- ></SCRIPT></pre>	<pre><script language="JavaS- cript"><!--<![CDATA[docu- ment.write("<p> Texto de prue- ba</p>");//--></script></pre>

En muchos casos se puede evitar la utilización de las secciones CDATA, incluyendo los códigos en archivos externos y aplicando variantes similares a:

```
<script language="JavaScript"
src="codigo.js"></script>
<link href="hoja_de_estilo.css" />
```

CONSEJOS FINALES

Espero que con lo comentado en los apartados anteriores habrá quedado clara la forma de conseguir que nuestros nuevos proyectos HTML se ajusten a las normas del XHTML, e incluso que es relativamente fácil convertir nuestros antiguos documentos HTML en renovadas páginas web que siguen la normativa más reciente.

No obstante, existen documentos y herramientas que nos pueden facilitar enormemente nuestra labor, sobre todo al principio, cuando surgen las primeras dudas.

La utilidad más importante para los interesados en el XHTML es la propia especificación *XHTML 1.0 Specification*, que se puede encontrar en el W3C (<http://www.w3.org>), y aunque todavía es un "borrador de trabajo" y se modificará en algunos detalles, el documento actual es suficientemente estable como para ser utilizado sin mayores problemas.

En el *website* del W3C se encuentran disponibles las DTDs: *Strict*, *Transitional* y *Frameset*, que si se pueden referenciar desde cada documento XHTML, también es práctico bajárselas al propio equipo, tanto para estudiar su contenido, como para ser utilizadas sin necesidad de conexión. Eso si, hay que asegurarse de que las DTDs que tenemos en nuestro equipo están actualizadas.

Si se desea utilizar un analizador de nuestro código, la forma más cómoda es aprovechar el que se encuentra disponible en W3C (¡cómo no!). Es suficiente con incluir un enlace en nuestra página web como:

<http://validator.w3.org/check/referer>

para que al ser activado realice la validación y análisis de forma automática.

Si se desea realizar la conversión de documentos HTML en XHTML en el propio ordenador y de forma totalmente automática, lo mejor es bajarse la utilidad Tidy, de Dave Raggett, que podemos encontrar gratis en la sede del W3C (¡qué sorpresa!, ¿no?).

Ya solo me queda terminar comentando que la adaptación de este artículo para la web que se encuen-

tra en <http://www.acta.es/manualesf.htm> es un ejemplo real de documento XHTML, aunque también puede verse en código HTML, lo que permite analizar sus diferencias en tiempo real.

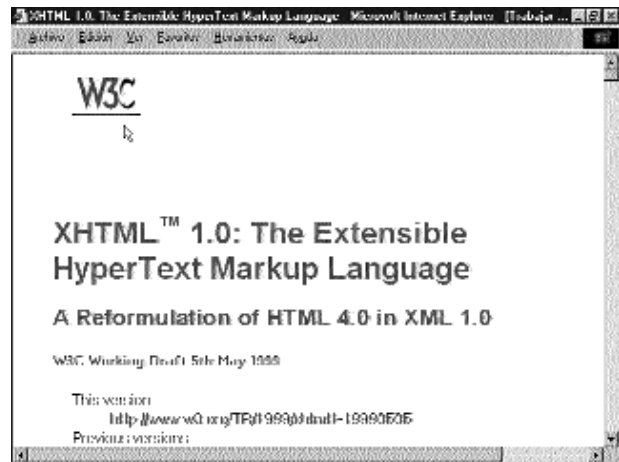


Figura 5. Las especificaciones de XHTML están accesibles para todo el mundo en www.w3c.org.