CAPITULO IV

World Wide Web (WWW)

4.1. INTRODUCCION

En los primeros años, la interfaz de los usuarios con Internet era mediante comandos de texto Unix, desde máquinas Unix, lo cual daba como resultado que Internet solo fuera utilizado por un reducido número de usuarios, debido a que la interfaz era un poco complicada. Y es por esta razón que Timothy Berners-Lee, un físico del Laboratorio Europeo para la Física de Partículas (CERN) en Ginebra, Suiza, planeó el Web en 1990 y en la actualidad está dirigido por The World Wide Web Consortium (http://www.3w.org), también conocido como la **Iniciativa World Wide Web**.

Berners-Lee propuso un sistema de hiperenlaces, una red de enlaces que permitiera a los usuarios de computadora moverse fácilmente de una computadora host a otra, en Internet, en busca de información relacionada. Los documentos se escribirían en un lenguaje abierto (HTML) que pudiera interpretar cualquier tipo de computadora, independientemente de su sistema operativo, y lo más importante, los documentos incluirían enlaces integrados hacia otros documentos. Inicialmente, las páginas web incluían solamente texto. La verdadera revolución en el Web ocurrió cuando Marc Andreeson desarrolló un visualizador (Mosaic) que se ejecutaba en un PC y permitía leer páginas en HTML mediante una interfaz gráfica de usuario (GUI), y a partir de este momento se fueron creando y perfeccionando los servidores y visualizadores web.

En la actualidad el Web permite saltar mediante un hipervínculo[1] de una página a otra. Las páginas pueden contener imágenes, películas, sonidos, gráficos en tres dimensiones, secuencias de video, entre otros. Estas páginas pueden estar situadas en sistemas en cualquier lugar del mundo y tiene igual acceso a información desde cualquier lugar del planeta; sin restricciones o costos de larga distancia, ver la Figura 4.1.

[1] Es una palabra resaltada, que cuando se selecciona con el ratón transfiere al usuario a otra página Web en la misma máquina o en cualquier otra en Internet.

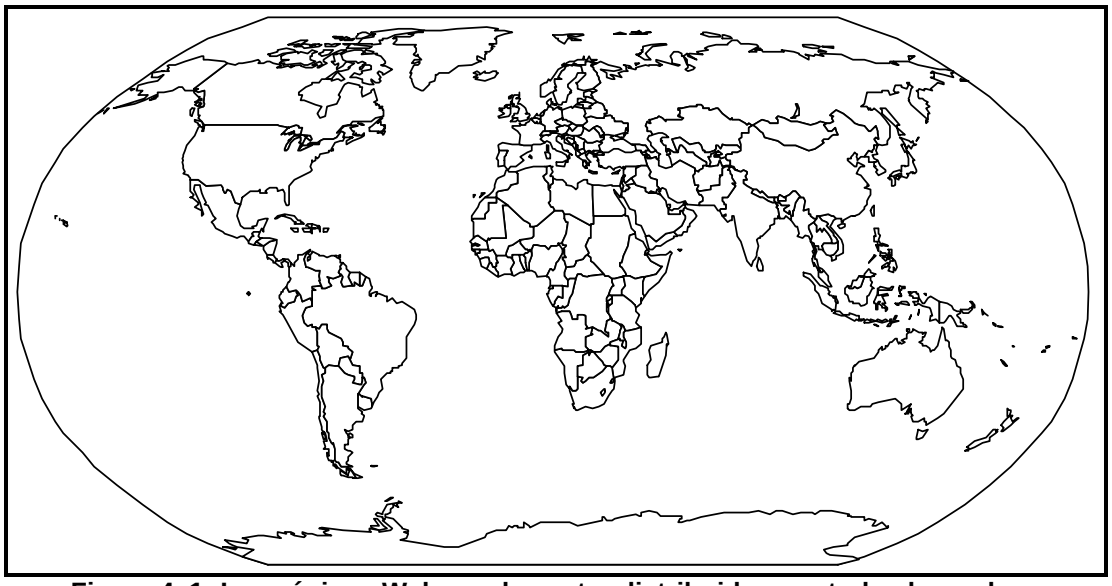

Figura 4. 1: Las páginas Web pueden estar distribuidas por todo el mundo y enlazadas

El World Wide Web puede estar perfectamente incluido dentro de las siete maravillas del mundo. Tras haber pasado algo de tiempo navegando por el Web, uno empieza a sentir que no hay límite en la cantidad de información que es posible encontrar. Se pueden pasar horas entretenido por la variedad de cosas que la gente publica en sus páginas Web. La pregunta es simple: ¿qué es entonces el Web? Se puede imaginar el Web como un sistema de documentos de todas partes del planeta e interrelacionados.

Generalmente se asocia a Internet únicamente con World Wide Web, la sección más importante de la Red. Pero esa es solo una parte, ya que también existen otros servicios tales como FTP, Correo, Noticias, Conversaciones, etc.

La pagina de entrada a un sitio de WWW se llama Página Principal. En ella se encuentra distribuida la información de tal manera que la puede enlazar con capítulos del mismo sitio o con otras paginas, quizás ubicadas al otro extremo del mundo. Dichos enlaces están representados por palabras, frases, fotos o gráficos, resaltados en su mayoría con color azul.

Para 'navegar' por las páginas de WWW se usa un programa de computador llamado browser o navegador. Algunos de los más populares son Internet

Explorer y Netscape. Para llegar a un sitio de World Wide Web es indispensable escribir su dirección o URL; en Internet usualmente comienza con http://...

En sus paginas hay información de todo tipo. Empresas, medios de comunicación, entidades gubernamentales, universidades, museos, partidos políticos, grupos religiosos, centros de investigación, centros comerciales virtuales e incluso personas naturales tienen paginas de presentación para promocionar sus productos, servicios y opiniones. Unos pocos segundos es toda la distancia que hay entre el sitio de la NASA y el de la Universidad Nacional de Tokio. En razón de que el lenguaje común en WWW y en todo Internet es él ingles, el usuario que la consulte deberá entenderlo medianamente para aprovechar el mar de información de la Red.

Sin embargo, encontrar sitios puede ser una tarea ingrata si no se manejan las herramientas adecuadas y disponibles en la Red para hacerlo. Hay dos herramientas básicas para hallar información en WWW: los motores de búsqueda (robot) y los directorios.

Los Motores de búsqueda como AltaVista, están diseñados para buscar en las computadoras que almacenan la información de la Red (conocidos como servidores) y crear índices de las páginas que hay en ellos. El usuario escribe una serie de palabras y el motor de búsqueda arroja una lista de páginas que las contienen, con una descripción de su contenido.

Los directorios como Yahoo, organizan las páginas web en diferentes categorías. También es posible realizar búsquedas en ellos, pero son más restringidas. Mientras que los motores de búsqueda pueden llegar a listar 50 millones de páginas, los directorios listan sólo una parte de ellas.

Otra opción para encontrar información en WWW son los motores de búsqueda múltiples, como Dogpile y Search. Estos envían la búsqueda de manera simultanea a los buscadores más usados de la red, acopia los resultados y los presenta en una sola lista. Algunos de ellos, como

HuskySearch, desarrollado por la Universidad de Washington, permite especificar si quiere una búsqueda rápida (menos de 5 segundos), una media (hasta 30 segundos) o una búsqueda intensiva (hasta 3 minutos). Como es de suponer, a mayor tiempo que se le dé al buscador, mayor número de documentos encontrará.

Buscopio es un buscador de buscadores. Tiene en su directorio una lista de 3069 (5/12/2000) motores clasificados por temas, por idiomas y por tipo de búsqueda. También permite buscar los motores apropiados para un tema específico utilizando una opción de búsqueda. Una vez efectuada la consulta, el sitio provee la facilidad de consultar en el motor seleccionado.

Es por esta razón que el World Wide Web está cambiando la forma de comunicarse de las personas en todo el mundo. Este nuevo medio global está siendo aceptado más rápidamente que ningún otro medio de comunicación en la historia. En los dos últimos años, ha crecido hasta incluir una vasta gama de información: cualquier cosa, desde cotizaciones bursátiles hasta ofertas de trabajo, boletines de noticias, pre_estrenos de películas, revistas literarias y juegos. La gama de información oscila desde los temas más desconocidos, hasta los de importancia mundial. La gente suele hablar de "explorar" el Web y visitar nuevos sitios.

"Explorar" como anteriormente se mencionó significa seguir los hipervínculos entre páginas y temas sobre los que es posible que nunca se haya oído hablar, conocer a gente, visitar nuevos lugares y aprender acerca de cosas de todo el mundo.

Ahora que ya se tiene una idea de lo que es el World Wide Web se puede explicar en que consiste un servidor WEB.

4.2. SERVIDOR WEB

Los servidores Web son PCs en los cuales se ha instalado software capaz de manejar las solicitudes de los visualizadores, es necesario señalar que en

estos equipos debe encontrarse el protocolo de transferencia de hipertexto (HTTP). El mismo que ha sido definido como un conjunto de comandos basados en ASCII para su lenguaje de comandos. Los visualizadores son programas que utilizan estos comandos HTTP para solicitar servicios a un servidor Web o HTTP.

Una transacción HTTP se compone de cuatro partes: una conexión, una solicitud, una respuesta y una conclusión. Aparte de realizar tareas de recuperación de archivos un servidor Web puede correr programas de aplicaciones, los cuales realizan tareas de búsquedas de una base de datos, procesamiento de un formato con entradas de usuario, tareas que las programa el cliente.

En la actualidad se dispone de gran cantidad de servidores Web potentes tanto para Windows NT, UNIX y Linux. Por lo que para su implantación se debe considerar aparte del costo, soporte para los desarrolladores Web, ya que de la selección depende la facilidad de ampliación.

4.3. DIRECCIONES DE WEB

Podemos considerar el World Wide Web como un sistema de documentos multimedia juntos en el mismo entramado por todo el globo, y el hipertexto o los enlaces como el pegamento que los une. Los URL son las direcciones de los recursos en el World Wide Web. Son lo que el Web usa para representar enlaces hipertexto a otros archivos o servicios de red (por ejemplo, funciones de búsqueda y transferencia de archivos) contenidos dentro de documentos HTML. La información presente en un URL proporciona la facilidad de saltar de un lugar del Web a otro simplemente con una pulsación en nuestro ratón.

La mayoría de los visualizadores de Web permiten entrar en un URL y conectar a ese documento o servicio. Cuando se pulsa sobre un enlace hipertexto en un documento HTML, lo que estamos haciendo es mandar una

solicitud de entrada a un URL. De esta manera, los hiperenlaces no sólo conectan con otros textos y medios, sino también a otros servicios de red.

¿Cómo se diferencia un URL típico? Aquí hay varios ejemplos:

http://www.utn.edu.ec

Esta es la página del servidor de la Universidad Técnica del Norte.

ftp://srvfica/

Esto es un directorio de archivos disponibles para ser copiados desde el Servidor de la FICA.

La primera parte de un URL (antes de las dos barras inclinadas) dice el tipo de recurso (o método de acceso) a esa dirección. Por ejemplo:

- **http** - un documento o directorio hipertexto
- **gopher** - un documento o menú de gopher
- **ftp** - un archivo disponible para copiar o un directorio de tales archivos
- **news** - un grupo de noticias
- **telnet** - un sistema en una computadora al que podemos acceder desde Internet
- **WAIS** - una base de datos o un documento en una base de datos WAIS (Wide Area Information Search o Búsqueda de información en áreas amplias)
- **file** - un archivo en un disco local (por ejemplo, nuestro disco duro)

La segunda parte de un URL es por lo general la dirección del computador donde los datos o el servicio se encuentran. Otras partes adicionales pueden especificar los nombres de los archivos, el puerto de conexión o el texto a buscar en la base de datos.

Se puede acceder al URL de un servidor pulsando sobre el botón **Ir** a en la barra de herramientas del navegador o desplazándose al menú **Abrir archivo**. De cualquier manera, se abrirá una ventana para introducir un URL. La mayoría de los visualizadores pueden recordar los URL que se desee usar de nuevo. Esto se consigue añadiéndolos a un menú especial en nuestro navegante llamado 'lista caliente' (algunos navegantes lo llaman 'marcador de libro' o “favoritos”). Una vez que se ha añadido un URL a la lista se puede volver a esa página pulsando sobre el enlace en la lista en vez de escribir de nuevo todo el URL.

La mayoría de los URL que se usa comienzan con ***http*** que significa **Protocolo de transferencia de hipertexto*. Http*** es el método por el que los archivos HTML son copiados a través del Web. Estas son unas pocas cosas importantes que se debe recordar sobre los URL:

1. Un URL no tiene espacios.
2. Un URL siempre usa barras inclinadas hacia la derecha (/).
3. Si se escribe incorrectamente un URL, el visualizador no será capaz de localizar el servidor o recurso que se desea.

Se puede encontrar el nombre del URL detrás de cualquier enlace pasando el puntero sobre el enlace. El puntero tomará la forma de una mano y el URL aparecerá en la barra de estado del navegante.

4.4. Estructura de una pagina de web

Navegando por el World Wide Web, se puede encontrar el término 'home page' o 'página principal' bastante a menudo. Se debe imaginar la página principal como el punto de partida de un servidor de Web. Exactamente como si se tratase del índice de un libro o revista, la página principal ofrece, en la mayoría de los casos, un resumen de lo que se encontrará en el servidor. Un servidor de Web puede contener una página, muchas páginas o unas pocas muy largas, dependiendo de cómo esté diseñado. Si no hay mucha información, la página principal puede ser la única. Sin embargo, por lo general se encontrará al menos unas pocas páginas. En la Figura 4.2, se muestra la página principal de la FICA.

Figura 4. 2: Página Principal de la FICA

Las páginas de Web tienen una infinita variedad de diseño y contenido, pero la mayoría usan el formato tradicional de las revistas. Al principio de la

página hay un encabezamiento o un gráfico. Debajo suele aparecer una lista de apartados con una breve descripción. Las descripciones contienen enlaces hipertexto a otro material en el mismo o en otro servidor. A veces estos enlaces son palabras remarcadas dentro del cuerpo del propio texto o bien ordenadas en una lista en forma de índice. La mayor parte de las veces, los enlaces son una combinación de ambos tipos.

¿Cómo se puede diferenciar un enlace? Los enlaces de texto aparecen en un color diferente al del resto del texto -en azul y subrayados, por lo general- Cuando se mueve el cursor sobre un enlace de texto o sobre un enlace gráfico, el cursor cambia su forma de flecha por la de una mano. Hay que notar que las palabras que sirven de enlace por lo general dan una idea de a dónde lleva ese enlace.

Cuando se vuelve a una página que contiene enlaces ya visitados, dichos enlaces hipertexto aparecerán en diferente color -rojo o rosa, por lo general- de tal manera, siempre se conoce que ya se ha estado allí, aunque se puede volver. Sin embargo, no se debe sorprender si la siguiente vez que se visite un servidor la página es diferente y la información ha cambiado. El Web es un mundo muy dinámico. Y a mucha de la gente que crea páginas de Web le gusta cambiarlas a menudo. ¡Esa es la razón por la que mostrar información en el Web es tan excitante!

4.5. Navegadores

Los navegantes, navegadores o visualizadores son programas que permiten acceder al World Wide Web, es la parte gráfica o visual de Internet. El primer navegante, llamado **NCSA Mosaic**, fue desarrollado en el **National Center for Supercomputing Applications** hace tan sólo unos pocos años. El interfaz gráfico muy sencillo de usar a través de punteros popularizó el Web, aunque sólo unos pocos podían imaginar el crecimiento tan explosivo que ocurriría, pues en muy pocos años el Web se ha convertido en una herramienta del convivir diario de las instituciones educativas, de negocios e industriales y muy pronto en todos los hogares del mundo.

Aunque están disponibles una gran cantidad de visualizadores diferentes, Microsoft Explorer y Netscape Navigator son los más utilizados y por tanto se llevan todos los honores. A la hora de tener que elegir son los dos visualizadores que debería usted considerar para la navegación general de la Red. Netscape y Microsoft han jugado tanto dinero en sus respectivos navegadores o visualizadores que la competencia no puede mantenerse a su ritmo. La encarnizada lucha entre las dos compañías para dominar el mercado ha conducido a mejoras continuas en los programas. Las últimas versiones de ambos navegadores son elecciones excelentes. Es necesario mencionar que ambos visualizadores están basados en **NCSA Mosaic**. Usted puede recibir ***Explorer*** gratuitamente a través del sitio de Microsoft o ***Navigator*** desde el sitio de Netscape para su evaluación y posterior elección. En las versiones de Windows y Mac existen ligeras diferencias, lo que hace que sea casi imperceptible.

Los navegadores poseen todo tipo de opciones. Explorer y Navigator tienen más similitudes que diferencias.

4.6. El Web en la Intranet

El Web en la intranet es un servicio que permitirá realizar transacciones, las mismas que se las efectuará en menor tiempo. El Web en una organización agilitará la comunicación interna de la empresa, es decir que con la instalación de este servicio mejorará las relaciones de las diferentes dependencias.

El WWW en una organización genera que posibilita realizar transacciones comerciales desde cualquier lugar sin tener que asistir a las dependencias de la empresa, servicio que mejorará las actividades económicas de la misma.

Estas son algunas de las ventajas del Web en una Intranet, entonces se demuestra que colocado este servicio en una organización, así como la buena utilización del mismo garantizará la mejora de la eficiencia y eficacia de la empresa. En la Figura 4.3 se muestra un ejemplo de un servidor Web en una Intranet.

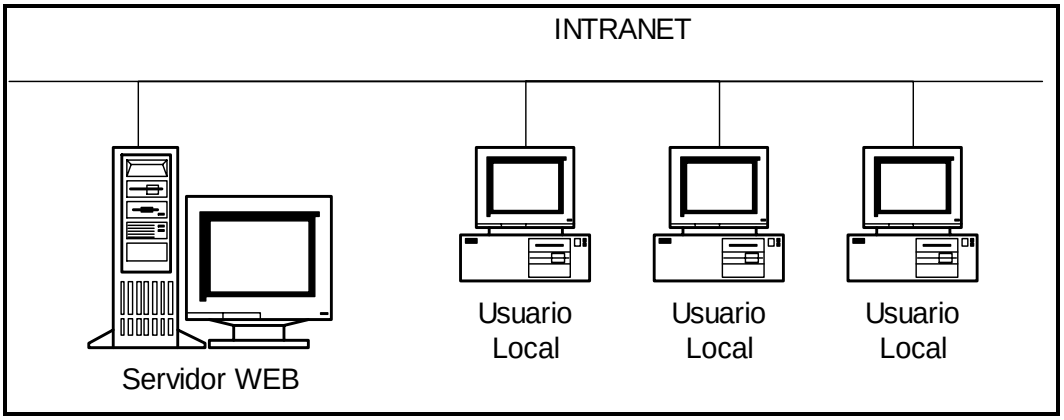

Figura 4. 3: Un servidor web en la Intranet

4.7. Herramientas para el Desarrollo de Hipertexto

4.7.1. Lenguajes de marcas

En los años 60, IBM intentó resolver sus problemas asociados al tratamiento de documentos por diferentes plataformas a través de lo que denominó GML (Generalized Markup Language, lenguaje de marcas generalizado) y más tarde, GML cayó en manos de ISO, que lo convirtió en un estándar oficial en los 80 (ISO 8879), denominándose SGML (Standard Generalized Markup Language, lenguaje de marcas generalizado estándar). Esta norma de carácter general se aplica desde entonces para diseñar lenguajes de marcas específicos, cuyos ejemplos más conocidos son el HTML y el RTF (Rich Text Format, formato de texto enriquecido).

Fue a finales de los 80 cuando Tim Berners-Lee aplicó las normas del SGML para diseñar el HTML como solución para publicar las investigaciones de muy diversas fuentes y autores que se producían en el CERN (Laboratorio Europeo de la Física de Partículas).

Los lenguajes de marcas no son equivalentes a los lenguajes de programación, aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que si se ajustan a SGML, se pueden controlar desde cualquier editor ASCII. Las marcas más utilizadas suelen describirse por textos descriptivos encerrados entre signos de "menor" (<) y "mayor" (>), siendo lo más usual que exista una marca de principio y otra de final, y así por ejemplo, la siguiente marcación: <NOMBRE>ASISOR</NOMBRE>, corresponde a un nombre, y de paso, indica con claridad el principio <NOMBRE> y el final </NOMBRE> de la marcación.

Se puede decir que existen tres utilizaciones básicas de los lenguajes de marcas: los que sirven principalmente para describir su contenido, los que sirven más que nada para definir su formato y los que realizan las dos

funciones indistintamente. Las aplicaciones de bases de datos son buenas referencias del primer sistema, los programas de tratamiento de textos son ejemplos típicos del segundo tipo, y aunque no lo parezca, el HTML es la muestra más conocida del tercer modelo.

4.7.2. HTML

El lenguaje HTML es originariamente un subconjunto del más completo SGML, especializado en la descripción de documentos en pantalla a través de marcas (tags, etiquetas). El proyecto inicial se basaba en una colección de etiquetas que permitían describir documentos de texto y vínculos de hipertexto que permitían desplazarse entre diferentes documentos, siempre con independencia de la máquina. Conociendo las normas de actuación de estas etiquetas y disponiendo de un sencillo editor ASCII de textos, se pueden confeccionar fácilmente documentos HTML.

La facilidad de uso y la particularidad de que no es propiedad de nadie, hizo al HTML el sistema idóneo para compartir información en Internet. La expansión de Internet le ha dado una posición de privilegio y ha hecho que la idea inicial se modifique considerablemente.

En principio, la intención de HTML era que las etiquetas fueran capaces de marcar la información de acuerdo con su significado, sin importar cómo se mostraban en la pantalla. Lo importante era el contenido y no la forma, es decir, era un lenguaje de marcas orientado a describir los contenidos. En otras palabras: el título del documento, los títulos de los apartados, el autor del documento, los textos resaltados,..., eran marcados por las etiquetas TITLE, Hx, ADDRESS, STRONG, etc., dejando a cada visualizador (browser) la tarea de formatear el documento según su criterio.

Esto producía presentaciones diferentes, pero permitía controlar fácilmente su contenido. Si una persona o un motor de búsqueda quería conocer el título del documento, el autor de la página o las cabeceras de los capítulos, siempre buscaba en el código las etiquetas TITLE, ADDRESS o Hx.

Además, si a alguien no le gustaba la idea de dejar a cada aplicación la decisión de cómo mostrar el contenido de las etiquetas, siempre le quedaba la posibilidad de controlar el formato del documento con descripciones particulares, como es el caso de las hojas de estilo en cascada (CSS).

Por diversos motivos, los creadores de los navegadores fueron añadiendo más etiquetas HTML dirigidas a controlar la presentación, como FONT, I, CENTER, xCOLOR, etc., y los usuarios las utilizaron para que sus documentos estuviesen perfectamente formateados, sin permitir diferencias importantes entre visualizadores distintos, por lo que HTML pasó a ser un lenguaje de marcas más dirigido al control de la presentación. Ahora es más difícil encontrar al autor o las cabeceras de los capítulos de un documento, pues todos los textos se describen con P y FONT, sobre todo si se utilizan los editores WYSIWYG[2] (Microsoft FrontPage, Netscape Composer) que proliferan por doquier.

Si a esto se le añade para facilitar la vida a los usuarios, los analizadores sintácticos de las marcas que incluyen los navegadores permitieron saltarse algunas normas sin que el propio usuario lo notase (por ejemplo, permiten trabajar solo con la etiqueta <P>, cuando lo correcto es que se necesite las etiquetas de principio y de final: <P> y </P>), dando como resultado que HTML ya no es un lenguaje que sigue las normas estrictas del SGML.

En la Figura 4.4 se puede ver la transformación del HTML, que en un principio era un subconjunto de SGML, pero que ahora es una entidad con mucha autonomía propia, más difícil de controlar.

Llegados a un punto en el que HTML dejó de servir para su función inicial, no le ha quedado más remedio al Consorcio World Wide Web (W3C) la descripción de un nuevo subconjunto del SGML que sirva para describir contenidos de documentos, al que ha denominado XML, publicando las especificaciones de la versión 1.0 en 1998.

[2] Lo que se ve es lo que se obtiene

Bien ya se puede empezar a hablar de XML con un poco de visión retrospectiva de los motivos de su aparición.

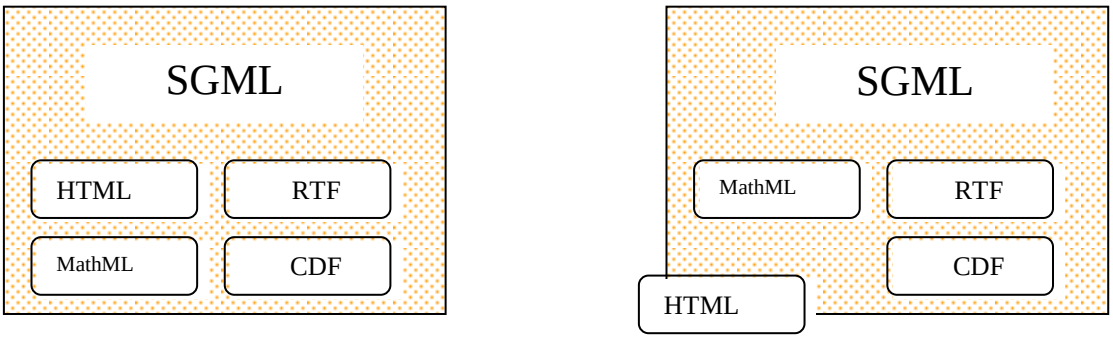

Figura 4. 4: Transformación del HTML

4.7.3. XML, el lenguaje universal

El mayor error consiste en considerar a XML una versión extendida de HTML. Sus iniciales provienen de *Extensible Markup Language* (Lenguaje de marcas ampliable), pero como para la mayoría de la gente, el único lenguaje de marcas que conoce es el HTML (*HyperText Markup Language*), da por hecho que XML es una ampliación de HTML. Es por tanto imprescindible empezar hablando de los lenguajes de marcas.

En primer lugar hay que olvidarse un poco de HTML para entender mejor XML. En la situación actual, en teoría, HTML es un subconjunto de XML especializado en presentación de documentos para la Web, mientras que

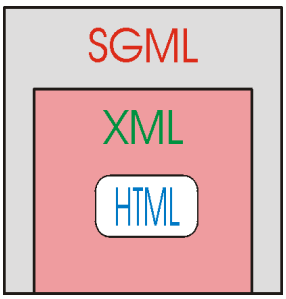

Fig: 4.5.a

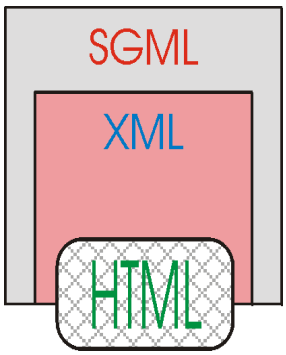

Fig. 4.5.b

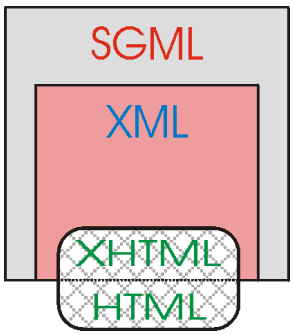

Fig. 4.5.c

Figura 4. 5: Relación del XML con el HTTP

XML es un subconjunto de SGML especializado en gestión de información para la Web ver Figura 4.5.a. En la práctica, HTML está un poco dentro de XML (que a su vez es parte de SGML) y otro poco fuera de SGML ver Figura 4.5.b. Para redirigir esta situación, el grupo W3C ha dictado reglas expresas para distinguir el HTML que sigue a rajatabla las normas de XML, denominándolo XHTML (eXtendible HyperText Markup Language), que no es más que una reformulación de HTML 4 dentro de las normas de XML. Ver Figura 4.5.c.

La particularidad más importante del XML es que no posee etiquetas prefijadas con anterioridad, ya que es el propio diseñador el que las crea a su antojo, dependiendo del contenido del documento.

Otra cuestión que hay que aclarar es que XML es más un sistema complejo de tratamiento de información que un simple lenguaje de descripción. XML es más una familia de lenguajes, y al igual que podemos decir que HTML es un "lenguaje", para ser exactos, hay que definir al XML como un "metalenguaje", o sea, un lenguaje capaz de definir otros lenguajes. Veamos esto con más detalle.

4.7.3.1. DTD/Esquema

Cualquier diseñador de HTML está totalmente despreocupado de las interioridades del sistema que hacen posible la creación de los archivos HTML (o HTM) cuando utiliza algún editor de los que genera el código de forma automática. Es suficiente con componer el documento de forma similar a como se hace con un programa de tratamiento de textos, activar la opción de "Guardar como HTML" y ya está. A partir de entonces, cualquiera que abra el archivo desde un navegador, verá dicho documento en su pantalla.

Esto que parece tan sencillo (gracias al esfuerzo realizado por las empresas creadoras de los navegadores), exige que se estén manejando varias herramientas internas al sistema.

Cuando se genera el documento en un editor HTML, el código HTML que se crea está basado en un DTD (Document Type Definition, definición de tipo de documento) interno, que es una definición de las normas que regulan la formación de las etiquetas de un lenguaje de marcas determinado, en este caso el HTML. Por lo tanto, en el DTD que integra cada editor de HTML se especifican todas las etiquetas existentes, sus atributos, sus valores, entre otras, haciendo posible que se vayan integrando en el código del documento únicamente de acuerdo con dichas normas. "*Se puede decir que un DTD es una definición exacta de la gramática de un documento, con la misión de que se genere el código adecuado sin errores".*

El fichero HTML creado con el editor, al cargarse en un navegador, es vuelto a analizar por su DTD interno para descubrir las etiquetas correctas y las equivocadas, siempre de acuerdo a sus normas internas. Por este motivo, si se crea una página web con el editor de una empresa y se visualiza con el navegador de otra, puede ocurrir que algunas entidades no se vean bien (o incluso puede pasar que no se puedan controlar), bien porque no existan dichas etiquetas (casos de BLINK o MARQUEE), o bien porque no se gestionen con las mismas normas (caso de TABLE).

En XML no existen DTDs predefinidas, por lo que es labor del diseñador especificar su propia DTD para cada tipo de documento XML. En la especificación de XML se describe la forma de definir DTDs particularizadas para documentos XML, que pueden ser internas (cuando van incluidas junto al código XML) o externas (si se encuentran en un archivo propio).

Que cada usuario pueda crear su propia DTD es una gran ventaja, ya que proporciona total libertad de adecuación a cada documento, pero también puede suponer un grave inconveniente, ya que es muy fácil que para documentos de un mismo sector (arquitectura, edición, educación, etc.), existan muy variadas DTDs, haciendo muy difícil su manejo por usuarios distintos a los que hayan diseñado la información.

Por este motivo, en la actualidad se están definiendo DTDs por grupos sectoriales con similares intereses, de forma que existirán DTDs estándares avalados por asociaciones de empresas y organismos que garanticen que cualquier usuario que las adopte como suyas, trabaje con las mismas etiquetas e idénticas normativas (de forma similar al actual HTML). Como ejemplos de estas DTDs estándares tenemos: CDF - *Channel Definition Format* (define canales para enviar información periódica a los clientes), CML - *Chemical Markup Language* (define información del sector químico), MathML - *Mathematical Markup Language* (define datos matemáticos), SMIL *Synchronized Multimedia Integration Language* (define presentaciones de recursos multimedia), y así podríamos continuar enumerando DTDs.

Últimamente se está imponiendo otra forma más eficaz de definición de elementos, conocida como "esquema" (*schema*), que de forma sencilla, se puede definir como un DTD que permite su ampliación mediante un lenguaje de definición de esquemas. Se pueden ver ejemplos de esquemas en el código XML que añade Office 2000 al principio de sus documentos.

El navegador, después de chequear sintácticamente el código del archivo, debe presentar la información del documento con un formato determinado. Los documentos HTML utilizan las descripciones de formatos internas del propio navegador, o si existen descripciones CSS (que son opcionales), utilizan la información de la hoja de estilo para ajustar la presentación en la pantalla. Los documentos XML siempre necesitan normas que describan su presentación, por lo que el paso siguiente obliga a que hablemos de este tema.

4.7.3.2. CSS/XSL

Para describir cómo se deben presentar los documentos XML podemos optar por dos soluciones: las mismas descripciones CSS que se utilizan con HTML y/o las descripciones que se basan en XSL (*Extensible Stylesheet Language*, lenguaje de hojas de estilo extensible).

CSS (*Cascading Style Sheets*, hojas de estilo en cascada) es sobradamente conocido por todos los diseñadores profesionales de HTML. Si el lector no conoce las posibilidades de las especificaciones CSS (actualmente en su versión 2), debe entenderlas como una descripción del formato en el que se desea que aparezcan las entidades definidas en un documento.

Si ya existe una forma de definir las presentaciones de los documentos web, ¿por qué se ha sacado a la luz otra (XSL), específica para XML?

La respuesta es que CSS es eficaz para describir formatos y presentaciones, pero no sirve para decidir qué tipos de datos deben ser mostrados y cuáles no deben salir en la pantalla. Esto es, CSS se utiliza con documentos XML en los casos en los que todo su contenido debe mostrarse sin mayor problema.

XSL no solo permite especificar cómo queremos presentar los datos de un documento XML, sino que también sirve para filtrar los datos de acuerdo a ciertas condiciones. Se parece un poco más a un lenguaje de programación.

No es el objetivo de la tesis ver las posibilidades del XSL, siendo suficiente con entender que además de permitir la descripción de la presentación física, también posibilita la ejecución de bucles, sentencias del tipo IF...THEN, selecciones por comparación, operaciones lógicas, ordenaciones de datos, utilización de plantillas,..., y otras cuestiones similares.

Nos queda por comentar que un estándar basado en SGML que regula las normas de presentación de documentos de marcas para la Web se denomina DSSSL (*Document Style Semantics & Specification Language*, lenguaje de especificación y semántica de estilo de documentos), del que se puede decir que XSL es un subconjunto, siempre cumpliendo las normas del XML.

Una de las características de todo documento web es la inclusión de enlaces de todo tipo: a imágenes, a sonidos, a vídeos, a otros párrafos, a otros documentos, etc. En HTML la cuestión está resuelta, ya que existen

etiquetas para cada caso, pero ya sabemos que en XML todo está por hacer, así que vamos a ver cómo se diseñan los enlaces de forma particular.

4.7.3.3. XLink/XPointer

La cuestión de los enlaces e hipervínculos es tan importante para los documentos XML que el W3C ha sacado las especificaciones que las controlan fuera de las descripciones del DTD, no conformándose con crear una sola norma, ya que existen dos: XLink y XPointer.

XLink (anteriormente conocido como XLL, *Extensible Linking Language*) define la forma en la que los documentos XML deben conectarse entre sí. XPointer describe cómo se puede apuntar a un lugar específico de un determinado documento XML. Resumiendo, XLink determina el documento al que se desea acceder y XPointer marca el lugar exacto de dicho documento.

A los diseñadores de HTML les puede parecer que esto es una complicación sin sentido, ya que están acostumbrados a las pocas posibilidades que brinda la etiqueta **<A>**, pero si pensamos en las muchas variaciones que pueden tener los vínculos, se comprende la solución adoptada.

Las especificaciones de los hipervínculos para XML permiten cosas como: adherirse a cualquier etiqueta, hacer referencia a un lugar concreto de un documento determinado a través de su nombre o localización, ser descritos en documentos externos, ser procesados de formas distintas (automáticamente, activándolo,...), ser multifuncionales (permitir varios saltos), etc.

Al contrario de lo que ocurre con HTML, en XML existen dos tipos básicos de hipervínculos: simples y extendidos.

Aunque no se han explicado todas las posibilidades de los hipervínculos XML, sí debe quedar claro que los enlaces XML son más variados que los que proporciona la sencilla, útil y conocida etiqueta **<A>** del HTML.

4.7.3.4. Parser/DOM

Siguiendo con el proceso que se desarrolla en el interior del navegador, después de recoger la información de todos los documentos que definen la información XML, se genera internamente una estructura que organiza a los elementos que describen las etiquetas en forma de árbol jerárquico, lo que facilita el control de dichos elementos.

En caso que se detecte algún error incompatible con las estrictas normas XML, el navegador interrumpe el proceso y muestra dicho error en la pantalla.

Todo este proceso se puede realizar gracias al analizador (*parser*) interno que incluye cada navegador, que en la mayoría de los casos se relaciona directamente con el estándar DOM (*Document Object Model*, modelo de objeto de documento), que entre otras cosas permite acceder a cada nodo del árbol a través de *scripts*.

Las reglas mínimas que hay que cumplir para no ser rechazados por un analizador XML son:

- Solo se permite un elemento raíz. Es imprescindible cumplir esta norma para que el *parser* pueda saber que el documento está completo. Es el equivalente a la etiqueta **<HTML>** del HTML.
- Incluir etiquetas de inicio (sin la "barra") y final (con la "barra") para todos los elementos. Siempre debe existir una marca **<ETIQUETA>** y otra **</ETIQUETA>** antes y después de cada elemento. No se permiten casos de etiquetas "sueltas" como las **<P>** o **<LI>** del HTML.

4.7.4. 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, entre otras, 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 perseverantes 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 se verá 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, 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, menos permisivo que el que aplican en la actualidad, pero de riguroso estándar.

4.7.4.1. Relación con HTML y XML

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

Matemáticamente, se podría decir que: **XML + HTML = XHTML** (*más o menos*) <-- expresión poco técnica, pero efectiva.

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 se ha 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:

- 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.

- 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 éstos de dicha potencia. Televisores, teléfonos móviles, ordenadores de bolsillo, calculadoras, hornos, tostadoras, y otras, soportarán código HTML, siempre que esté realmente unificado y se ajuste a normas estrictas para no dar problemas que exijan soluciones complejas.

4.7.4.2. Diferencias con HTML

Las normas que regulan el código XHTML son suficientemente sencillas como para no asustar a nadie, sobre todo si se trata de 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.

- **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 (').

- **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 <LI>, <DT> y <DD>, que ahora deben definirse como <li> ... , <dt> ... </dt> y <dd> ... </dd>.

- **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 si tengan contenido entre las etiquetas de principio y de final, y así, las etiquetas y contienen un párrafo, las etiquetas y contienen un texto en cursiva, etc.

No obstante, en HTML también existen algunos elementos que no contienen nada, como
, <hr> e <img>, por lo que solo existen como etiquetas únicas, que hacen las veces de principio y de final.

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" (>), ejemplo:

- **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.

- **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 corriente encontrarse 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">**.

- **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>**.

4.7.4.3. Novedades

Las cuestiones comentadas en el apartado anterior 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 indica qué cosas pueden hacerse en el 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 <font> y <table> 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*).

- **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 URL, 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.

4.7.5. Consejos finales

Lo comentado en los apartados anteriores habrá quedado clara la forma de conseguir que nuevos proyectos HTML se ajusten a las normas del XHTML, e incluso que es relativamente fácil convertir antiguos documentos HTML en renovadas páginas web que siguen la normativa más reciente. No obstante, existen documentos y herramientas que pueden facilitar enormemente una 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 se tiene en el equipo estén actualizadas.

Si se desea utilizar un analizador de 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 activa el enlace desde esta página se puede ver cómo actúa en tiempo real el validador).

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 se puede encontrar gratis en la sede del W3C (¡qué sorpresa!, ¿No?).

Sólo se puede comentar que algunos de los tutoriales que se incluyen en el web son ejemplos reales de documentos XHTML.

RECOMENDACIONES:

Investigar la ultimas herramientas para el desarrollo de páginas web, así como también la tecnología Java, ya que estas tecnologías están desarrollándose a un ritmo vertiginoso.

Visitar las páginas del W3 Consorcio, para ver las últimas novedades en cuando a estándares para realizar páginas web.

BIBLIOGRAFIA:

Ambegaonkar Prakash.	Kit de Recursos de Intranet Editorial Osborne / McGraw-Hill primera edición
Greer Tyson.-	Así son las Intranets Editorial Microsoft Press primera edición 1997
Kris Jamsa / Ken Cope.	Programación en Internet (Curso sobre TCP/IP) Editorial McGraw-Hill primera edición 1996
http://www.w3c.org	W3 Consortium Organization.
http://www.isoc.org	Sociedad de Internet
http://www.learnthenet.com	Comprenda la Red (Internet)