



## CENTRO DE ESTUDIOS FINANCIEROS

VIRIATO, 52	28010 MADRID	914 44 49 20
PONZANO, 15	28010 MADRID	914 44 49 20
G. DE GRÀCIA, 171	08012 BARCELONA	934 15 09 88
ALBORAYA, 23	46010 VALENCIA	963 61 41 99

www.cef.es

info@cef.es

## Índice Tema 10

1. HTML.
  - 1.1. Generalidades.
  - 1.2. HTML y los conjuntos de caracteres.
  - 1.3. SGML y HTML.
  - 1.4. La evolución de HTML.
  - 1.5. Etiquetas más importantes.
2. XML.
  - 2.1. Generalidades.
  - 2.2. Características del lenguaje XML.
  - 2.3. Estructura del XML.
  - 2.4. Desarrollo de aplicaciones con XML.
  - 2.5. Requisitos para desarrollar documentos XML.
  - 2.6. Componentes de un documento XML.
    - 2.6.1. Elementos.
    - 2.6.2. Atributos.
    - 2.6.3. Prólogo.
    - 2.6.4. Comentarios.
    - 2.6.5. CDATA.
    - 2.6.6. Entidades predefinidas.
  - 2.7. Documentos XML bien formados.
    - 2.7.1. La regla «document».
3. HTML, XML, *versus* SGML.
4. Cuadro-resumen con algunas de las diferencias significativas.





## CENTRO DE ESTUDIOS FINANCIEROS

VIRIATO, 52	28010 MADRID	914 44 49 20
PONZANO, 15	28010 MADRID	914 44 49 20
G. DE GRÀCIA, 171	08012 BARCELONA	934 15 09 88
ALBORAYA, 23	46010 VALENCIA	963 61 41 99

www.cef.es

info@cef.es

### TEMA 10

**Lenguajes de marca o etiqueta. Características y funcionalidades. SGML, HTML, XML y sus derivaciones.**

#### 1. HTML.

##### 1.1. GENERALIDADES.

HTML (HyperText Markup Language) es el lenguaje utilizado para crear documentos de hipertexto. Los ficheros HTML describen la distribución y estilo de cada uno de los elementos de una página de Web, a base de combinar el texto de la página con los diferentes comandos del lenguaje. El resultado de la presentación de un documento HTML es muy similar a lo que se puede conseguir con un procesador de textos; sin embargo, el código HTML puede ser interpretado y visualizado en una gran variedad de entornos.

Los ficheros HTML contienen texto plano (sin ningún carácter especial o de control, ni procesamiento por compiladores o filtros), y pueden ser editados con cualquier aplicación sencilla que exporte texto sin formato (edit en MSDOS, notepad en Windows o vi en UNIX). Sin embargo, existen editores especializados que simplifican esta tarea, automatizando determinadas labores de la creación de documentos de hipertexto.

HTML está construido a partir de «etiquetas» (tags), que marcan el formato de cada uno de los elementos de la página de hipertexto. Todas las etiquetas son palabras o abreviaturas inglesas rodeadas de los símbolos < >. Las etiquetas pueden contener parámetros u opciones que modifican su comportamiento por defecto.

<HTML>

<HEAD><TITLE>Home Page de Luis</TITLE></HEAD>

<BODY>

<IMG SRC="foto.gif" ALIGN=middle>Bienvenido a mi

<I>peque&ntilde;a</I> p&aacute;gina personal

</BODY></HTML>

Las etiquetas pueden ser de dos tipos: pareadas, cuando cambian el formato de un bloque (por lo que hay una etiqueta de principio y otra de final) y sin parear, cuando insertan un elemento (por ejemplo, una imagen) o cambian el formato desde su punto de inserción en adelante; en este caso, sólo se emplea una sola etiqueta.

Para crear un encabezado grande <H1>Esto es un título muy grande</H1>.

Para insertar una imagen <IMG SRC="foto.gif">.

El lenguaje HTML ha sido diseñado teniendo presentes dos normas básicas:

1. Define la estructura y componentes de un documento, no la forma concreta en que estos componentes se presentan cuando un cliente los visualiza.
2. No está ligado a ningún entorno particular. El contenido de un documento HTML puede ser interpretado y visualizado en ordenadores con características muy diferentes.

Estas dos condiciones son muy importantes, ya que permiten que una determinada información pueda ser vista por usuarios diferentes, con independencia de las capacidades de su entorno de trabajo. Un documento HTML sólo contiene la especificación de los elementos que lo componen (títulos, párrafos, imágenes, etc.), y es responsabilidad de cada cliente el mostrar esta información de la manera más adecuada, en función de sus capacidades y las características del entorno.

Se puede decir que HTML es un lenguaje interpretado, ya que son los browsers los encargados de procesar y representar su contenido. Un browser tiene mucha libertad para ajustar la presentación de un documento HTML en función de los recursos disponibles.

Por lo tanto, se debe ser consciente que, dependiendo del entorno de trabajo de cada cliente (sistema operativo, tipo y versión del browser, etc.), un documento HTML puede ser visualizado de manera diferente.

Uniando los servidores HTTP y los documentos HTML, se dispone de un sistema distribuido de acceso a información, que combina un formato de presentación muy atractivo con un sencillo mecanismo de navegación por la información, basado en la selección de elementos activos.

En general, un browser no respeta nada del posible formato que contenga el documento HTML que interpreta, por lo que se ignoran los saltos de línea o los espacios múltiples entre palabras. Si se desea incluir un salto de línea, se debe indicar explícitamente a través de su correspondiente etiqueta; por tanto, la inclusión de espacios adicionales entre líneas de código HTML ayuda a mejorar la legibilidad del documento, pero no afecta a la forma en que éste se representa. Todos estos párrafos tendrán la misma presentación:

<P>Internet es un enorme almacén de información

<P>Internet es un enorme almacén de información

<P>Internet es un enorme almacén de información

Un documento HTML está clasificado a partir de su descripción MIME como text/html (texto plano cuyo contenido es código HTML).

## 1.2. HTML Y LOS CONJUNTOS DE CARACTERES.

Se denomina conjunto de códigos de caracteres (coded character set) a cada una de las posibles asociaciones entre números enteros (no necesariamente bytes) y caracteres que se emplean para representar información, entendiendo por carácter cualquier unidad de información (letra, dígito, signo gráfico o de puntuación, etc.). La definición de un conjunto de códigos de caracteres no implica, por tanto, ningún formato de representación electrónica de la información.

Para intercambiar información en una red como Internet, se debe disponer de un formato de representación electrónica de estos caracteres, denominado tabla de codificación de caracteres (character encoding scheme), que construye una equivalencia entre bytes (o series de bytes) y caracteres, para un determinado conjunto de códigos de caracteres. Para juegos de caracteres con menos de 256 elementos, la asociación es evidente, pero en otros casos se deben definir asociaciones que impliquen a más de un byte.

El conjunto de códigos de caracteres que se pueden incluir dentro de un documento HTML está definido en el conjunto denominado ISO-LATIN-1 (iso-8859-1), un juego de caracteres de 256 elementos que incluye la mayoría de los caracteres empleados en los alfabetos de Europa Occidental, junto con algunos caracteres gráficos especiales.

El manejo de un juego de caracteres único en los documentos HTML es imprescindible para garantizar su portabilidad entre diferentes entornos.

Para complicar más la situación, cada sistema operativo, en función de su tipo y el país en que se ejecuta, dispone de unas tablas de códigos (o mapas de caracteres) que asocian códigos numéricos a las representaciones gráficas de los caracteres. Los 128 primeros elementos de estas tablas son prácticamente equivalentes en todos los sistemas operativos e idiomas, y se corresponden con la conocida tabla ASCII. Sin embargo, los 128 elementos superiores son totalmente dependientes del sistema operativo y del lenguaje utilizado, y contienen elementos alfabéticos y gráficos particulares de cada país. Es decir, el carácter de código numérico 213 se verá de forma diferente entre un PC en España y uno en Francia, o entre un PC y un Macintosh.

Por tanto, un browser que recibe un documento con un carácter de índice decimal 225 sabe que debe mostrar en la pantalla su código ISO-LATIN-1 asociado, la á, en lugar del elemento correspondiente en su propio mapa de caracteres.

Sin embargo, ésta no es la situación real. Algunos browsers muestran los caracteres que reciben según mapa de caracteres local, en lugar de hacer transformación a ISO-LATIN-1.

Para evitar este posible problema, HTML permite incluir caracteres ISO-LATIN-1 a través de su código numérico correspondiente &#225, o de un nombre mnemotécnico &acute;, denominado refe-

rencia a carácter, que al utilizar exclusivamente caracteres ASCII son correctamente interpretados en todos los entornos.

Este comportamiento obedece a una limitación de la especificación original de HTML, que no tuvo en cuenta la necesidad de presentar información en alfabetos diferentes del Europeo Occidental. Los clientes Web modernos ofrecen la posibilidad de alterar el mapa de caracteres con que se representa un documento en pantalla; para hacer este cambio, es preciso tener instalado un fichero de fuentes adecuado, que contenga las correspondencias entre los caracteres propios del idioma y el contenido del documento HTML.

En la actualidad se está realizando un importante esfuerzo para definir un modelo único de intercambio de información, a través del empleo de UNICODES (alfabetos de caracteres que ocupan más de un byte), o de la especificación del juego de caracteres incluido en un documento al proporcionar su descripción MIME.

Como resumen de todo lo anterior, los clientes Web necesitan conocer el juego de caracteres con el que representar el contenido de un documento HTML. En principio, estos documentos sólo pueden utilizar un juego de caracteres de 255 elementos, adecuado para los usuarios de Estados Unidos y Europa Occidental. Para insertar símbolos de otros idiomas o caracteres especiales se pueden utilizar las referencias a carácter, que asocian nombres mnemotécnicos a determinados símbolos.

### 1.3. SGML Y HTML.

HTML es un caso particular de los llamados SGMLs (Standard Generalized Markup Languages), un complejo sistema de definición de documentos estructurado, normalizado y aceptado internacionalmente. Surgió en 1969, como propuesta para disponer de un sistema único de representación e intercambio de documentos. Su estandarización definitiva se produce en 1986, al ser aceptado por ANSI e ISO.

El lenguaje HTML está definido a partir de unas reglas SGML, recogidas en un documento denominado HTML DTD (Document Type Definition). En él se especifica la estructura de un documento HTML, el conjunto de etiquetas disponibles, sus opciones, y la forma en que se pueden combinar. Así, un programa de ordenador puede chequear fácilmente la sintaxis de un documento HTML.

Para que un documento HTML sea considerado como tal desde el punto de vista de SGML debe comenzar por una línea `<!DOCTYPE>`, en la que se indica que contiene código HTML, la versión del mismo que utiliza y otros datos de interés para el browser que procesa esta información. Todo documento HTML debería incluir una de estas definiciones:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML Level 1//EN">
```

El contenido es código HTML, versión 1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 2.0//EN">
```

El contenido es código HTML, versión 2.0

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML X.0//EN">
```

El contenido es código HTML, versión X.0

Además, los clientes Web pueden conocer el tipo de datos que reciben por la información MIME que el servidor HTTP adjunta. Así, para un documento con extensión .html, el servidor HTTP avisará que se corresponde con text/html. El administrador de un servidor HTTP tiene capacidad para alterar la forma en que se resuelven las asociaciones MIME.

#### 1.4. LA EVOLUCIÓN DE HTML.

Desde su creación, el lenguaje HTML ha evolucionado rápidamente, impulsado por la necesidad de ampliar sus capacidades y adaptarse a los nuevos requerimientos de sus usuarios. Este cambio ha sido impulsado por organismos internacionales, como el World Wide Web Consortium (<http://www.w3.org>) o por empresas como Netscape y Microsoft.

La versión inicial se denominó HTML/1.0, y supuso un gran avance como sistema mundialmente aceptado de presentación de texto con formato. Sin embargo, pronto se descubrieron sus limitaciones. La introducción de los entornos gráficos de acceso al Web reclamaba mayores capacidades de formato que no fueron contempladas en esta especificación inicial. En la actualidad el estándar disponible se encuentra en la versión 4.0. Con el fin de dotar de mayores funcionalidades a este lenguaje se le han ido incorporando una serie de capacidades adicionales que permitan construir un contenido con apariencia dinámica y extensible a todo un sitio web. De esta forma aparece el denominado DHTML (Dynamic HTML). Se trata de un HTML al que se le ha añadido un lenguaje de scripting (p. ejem. JavaScript) y las denominadas hojas de estilo (CSS).

#### 1.5. ETIQUETAS MÁS IMPORTANTES.

##### • Estructura del documento html.

```
<html>
<head>
  Cabecera de la página. Metainformación
  <title> Título que aparecerá en la ventana del navegador </title>
</head>
<body>
  Cuerpo de la página. Información visible
</body>
```

##### • Etiquetas.

Etiqueta html	Descripción
 	Salto de línea
<hr>	Línea
<b></b>	Negrita
<h1></h1>	Formato de título. Hay cabeceras predefinidas desde h1 a h7
<font size=10></font>	Definición del formato de la fuente
<table></table>	Tablas
<a href=una_url></a>	Hipervínculos
<img src=mi_imagen>	Imagen

## • Tablas.

Etiqueta html	Descripción
<table></table>	Tabla
<tr></tr>	Fila
<td></td>	Columna

## • Frames.

Etiqueta html	Descripción
<frameset></frameset>	Conjunto de frames. Definición de una página html.
<frame>	Frame de una página

## 2. XML.

### 2.1. GENERALIDADES.

Estas razones han obligado a los miembros del W3 Consortium a, en lugar de desarrollar nuevas versiones de HTML desarrollar un nuevo lenguaje (mejor dicho metalenguaje) que han denominado XML (Extensible Markup Language) que aproveche las innegables ventajas del HTML pero que a su vez permita realizar muchas cosas más. Esto no significa, al menos por el momento, el fin del HTML. Existen demasiadas páginas en HTML y resulta muy sencillo crearlas. Además los navegadores no soportarán todavía en toda su potencia el XML y tecnologías asociadas, pero es evidente una reformulación del HTML como una aplicación XML(XHTML)21212 y un cambio radical en la forma de elaborar las páginas WEB.

La idea que subyace bajo el XML es la de crear un lenguaje muy general que sirva para muchas cosas. El HTML está diseñado para presentar información directamente a los humanos, y esto sin duda es algo bueno, pero es un lenguaje complicado de procesar para los programas informáticos. El HTML no es bueno porque no indica lo que está representando, se preocupa principalmente de que eso tiene que ir en azul, o con un tipo de letra determinada, pero no te dice que lo que está mostrando es el título de un libro o el precio de un artículo. El XML hace precisamente esto: describe el contenido de lo que etiqueta.

### 2.2. CARACTERÍSTICAS DEL LENGUAJE XML.

- Es una arquitectura más abierta y extensible. No se necesita versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet/intranet por medio de un validador de documentos (parser).
- Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML (los RDF), en comparación a los atributos de la etiqueta <META> del HTML.
- Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.



- Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje nos permitirá agrupar una variedad amplia de aplicaciones, desde páginas Web hasta bases de datos.
- Gestión y manipulación de los datos desde el propio cliente web.
- Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido Web en XML consigue que la estructura de la información resulte más accesible.
- Se desarrollarán de manera extensible las búsquedas personalizables y subjetivas para robots y agentes inteligentes. También conllevará que los clientes web puedan ser más autónomos para desarrollar tareas que actualmente se ejecutan en el servidor.
- Se permitirá un comportamiento más estable y actualizable de las aplicaciones Web, incluyendo enlaces bidireccionales y almacenados de forma externa (El famoso epígrafe «404 file not found» desaparecerá).
- El concepto de «hipertexto» se desarrollará ampliamente (permitirá denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc.). Creado a través del Lenguaje de enlaces extensible (XLL).
- Exportabilidad a otros formatos de publicación (papel, web, cd-rom, etc.). El documento maestro de la edición electrónica podría ser un documento XML que se integraría en el formato deseado de manera directa.

### 2.3. ESTRUCTURA DEL XML.

El metalenguaje XML consta de cuatro especificaciones (el propio XML sienta las bases sintácticas y el alcance de su implementación):

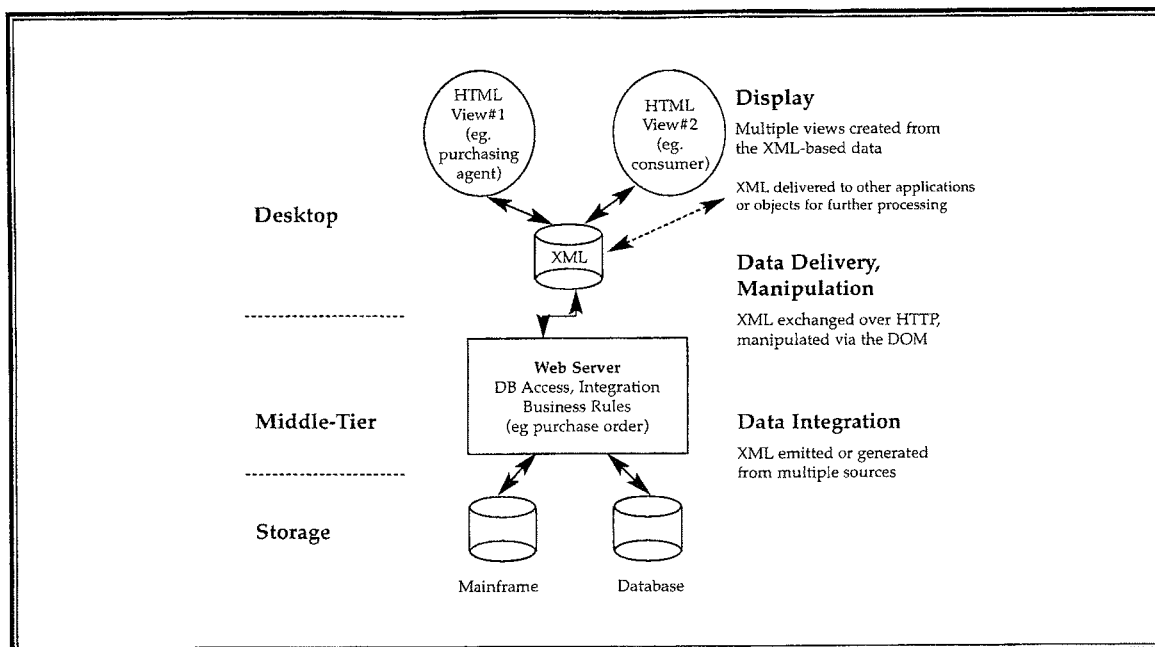
- DTD (Document Type Definition): definición del tipo de documento. Es, en general, un archivo/s que encierra una definición formal de un tipo de documento y, a la vez, especifica la estructura lógica de cada documento. Define tanto los elementos de una página como sus atributos. El DTD del XML es opcional. En tareas sencillas no es necesario construir una DTD, entonces se trataría de un documento «bien formado» (well-formed) y si lleva DTD será un documento «validado» (valid).
- XSL (eXtensible Stylesheet Language): define o implementa el lenguaje de estilo de los documentos escritos para XML. Desde el verano de 1997 varias empresas informáticas como ArborText, Microsoft e Inso vienen trabajando en una propuesta de XSL (antes llamado «xml-style») que presentaron a W3C. Permite modificar el aspecto de un documento. Se puede lograr múltiple columnas, texto girado, orden de visualización de los datos de una tabla, múltiples tipos de letra con amplia variedad en los tamaños. Este estándar está basado en el lenguaje de semántica y especificación de estilo de documento (DSSSL, Document Style Semantics and Specification Language, ISO/IEC 10179) y, por otro lado, se considera más potente que las hojas de estilo en cascada (CSS, Cascading Style Sheets), usado en un principio con el lenguaje DHTML.

- XLL (eXtensible Linking Language): define el modo de crear enlaces mucho más complejos que con la etiqueta <a href> de HTML, los denominados enlaces extendidos. Características:
  - Denominación independiente de la ubicación.
  - Enlaces que pueden ser también bidireccionales.
  - Enlaces que pueden especificarse y gestionarse desde fuera del documento a los que se apliquen. (Esto permitirá crear en un entorno intranet/extranet un banco de datos de enlaces en los que se puede gestionar y actualizar automáticamente. No habrá más errores del tipo «404 Not Found»).
  - Hiperenlaces múltiples (anillos, múltiples ventanas, etc.).
  - Enlaces agrupados (múltiples orígenes).
  - Transclusión (el documento destino al que apunta el enlace aparece como parte integrante del documento origen del enlace).
  - Se pueden aplicar atributos a los enlaces (tipos de enlaces).
- XUA (XML User Agent): estandarización de navegadores XML. Todavía está en proceso de creación de borradores de trabajo. Se aplicará a los navegadores para que compartan todos las especificaciones XML.

## 2.4. DESARROLLO DE APLICACIONES CON XML.

Según Jon Bosak el tipo principal de aplicaciones que surgirán como consecuencia de la implantación del XML serán:

«Aplicaciones que exijan que el cliente web medie entre dos o más bases de datos». Se hará posible la integración de bases de datos distribuidas en los navegadores que admitan XML, pudiéndose modificar el contenido y la estructura de ésta. Actualmente implantado en amplias redes nacionales, sin embargo, se limitan las posibilidades al establecerse una intranet/extranet y con amplias bases de datos que sólo permiten la visualización de los datos en el navegador. XML establecerá una arquitectura de 3 capas (three-tier) que está representada de la siguiente manera:



Otras aplicaciones que se desarrollarán son las operaciones para comercio electrónico con la normativa EDI. En este sentido son especialmente importantes los proyectos ebXML y UBL (Universal Business Language) que pretenden crear un estándar universal para el intercambio electrónico de datos entre cualquier tipo de organizaciones. Todos estos proyectos se circunscriben dentro del movimiento del software libre y están liderados por el creador del XML: Jon Bosak.

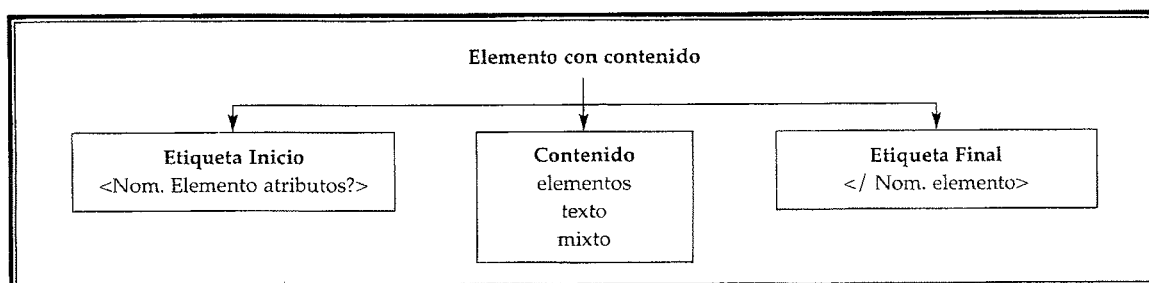
## 2.5. REQUISITOS PARA DESARROLLAR DOCUMENTOS XML.

- Editores XML (Ejem. Visual XML).
- Editor DTD (Ejem. EZDTD).
- Parsers XML (Ejem. parser de IBM).

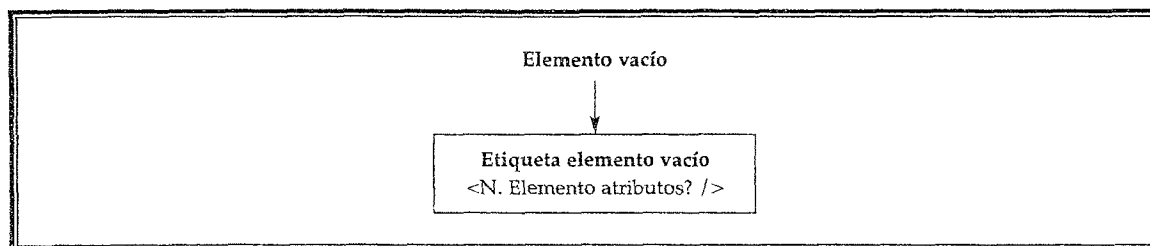
## 2.6. COMPONENTES DE UN DOCUMENTO XML.

### 2.6.1. Elementos.

Todo documento XML se compone de uno o más elementos, cuyos límites están delimitados por etiquetas de comienzo y etiquetas de fin en el caso de que tengan contenido:



En el caso de elementos con contenido, las etiquetas de comienzo se componen del símbolo menor que «<», el nombre del tipo de elemento, los atributos si los tiene y el símbolo mayor que «>». Mientras que las etiquetas de fin se componen del símbolo menor que seguido de contrabarra «</>», el nombre del tipo del elemento y el símbolo mayor que «>».



En el caso de ser un elemento vacío, sólo hay una etiqueta de elemento vacío que se forma del símbolo menor que «<», el nombre del tipo de elemento, los atributos si los tiene y se cierra con el símbolo «>». Es importante destacar este tipo de elementos, ya que hasta ahora en el SGML y, por tanto en el HTML entendido como aplicación SGML, los elementos vacíos sólo se representaban con una etiqueta de inicio.

### 2.6.2. Atributos.

Cada elemento puede tener atributos (propiedades) que nos ofrecen información sobre el elemento.

```
<p>Mi Primer <destacar importancia="1">documento XML</destacar></p>
```

Como podemos observar, la definición de un atributo está formada por el nombre del atributo, seguido del símbolo igual "=" y, entrecomillado, el valor del atributo.

### 2.6.3. Prólogo.

Los documentos XML pueden empezar con un prólogo, en el que esencialmente se define:

- Una declaración XML.
- Una declaración de tipo de documento.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE documento [
```

```
<!ELEMENT documento (p | imagen | ejemplo)*>
```

```
<!ELEMENT p (#PCDATA|destacar)*>
```

```
<!ELEMENT destacar (#PCDATA)>
```

```

<!ATTLIST destacar
    importancia CDATA #REQUIRED>

<!ELEMENT imagen EMPTY>

<!ATTLIST imagen
    fichero CDATA #REQUIRED>

<!ELEMENT ejemplo (#PCDATA)>

]>

```

En la declaración XML,

<?xml version="1.0" encoding="UTF-8"?>. Indicamos:

- Información sobre la versión de XML que estamos utilizando. Por el momento sólo puede ser la versión 1.0.
- Información sobre el tipo de codificación de caracteres que estamos utilizando. En nuestro caso es el código ASCII de 7 bits, que es un subconjunto del código Unicode denominado UTF-8. No hubiese sido necesario declararlo, ya que es el que los parsers manejan por defecto.

• **En la declaración del tipo de documento.**

```

<!DOCTYPE
documento [
<!ELEMENT
documento (p |
imagen | ejemplo)*>
<!ELEMENT p
(#PCDATA | destacar
)*>
<!ELEMENT destacar
(#PCDATA)>
<!ATTLIST destacar

```

Asociamos la DTD respecto de la cual construimos el documento. En nuestro ejemplo va implícita en el propio documento XML, aunque también puede hacerse externa al documento e incluso de una forma mixta. Si la hubiésemos escrito en un fichero "ejemplo.dtd" tendríamos que referenciarla de la siguiente manera:

```
<!DOCTYPE documento SYSTEM "ejemplo.dtd">
```

Aunque recordad que a diferencia del SGML, tenemos la posibilidad de no utilizarla.

Ambas partes del prólogo son opcionales, aunque en el caso de incluir ambas la declaración XML tiene que ir antes.

#### 2.6.4. Comentarios.

Mediante los cuales podemos proporcionar información que el parser no tendrá en cuenta.

```
<!-- Esto es un comentario -->
```

Los comentarios empiezan con los caracteres "<!--" y terminan con "-->" y pueden colocarse en cualquier sitio excepto dentro de las declaraciones, etiquetas y otros comentarios.

#### 2.6.5. CDATA.

Permiten integrar texto en un documento en XML que de otra forma sería interpretado como etiquetas. Es decir, estamos introduciendo texto que luego el procesador XML va a mostrar pero no va a procesar como marcado.

```
<![CDATA[
```

Aquí puedo poner lo que quiera.

```
] ]>
```

Los CDATA empiezan con los caracteres "<![CDATA[" y termina con "]]>".

Dentro de ellos podemos colocar cualquier cosa ya que no va a ser interpretado, con la salvedad de la cadena que indica el final de CDATA, "]]>", ya que el procesador al encontrársela entendería que la sección CDATA ya ha terminado con las nefastas consecuencias que esto puede tener.

#### 2.6.6. Entidades predefinidas.

En XML existen algunos caracteres reservados que no podemos utilizar para evitar problemas con el marcado, lo que no significa que no tengan que salir en nuestros documentos XML.

En nuestro ejemplo, nos aparece el caso cuando intentamos escribir la etiqueta <documento>

```
<p>Comienza con la etiqueta &lt;documento&gt;</p>
```

Ya hemos visto que una posible solución es la utilización de CDATA, pero sin duda es poco útil cuando simplemente queremos escribir un carácter.

Las entidades predefinidas son marcas XML que se utilizan para representar estos caracteres. El XML especifica cinco entidades predefinidas:

- & para el &
- < para el <
- > para el >
- ' para el '
- " para el "

Como podemos observar, se reconocen al ir entre los símbolos "&" y ">".

## 2.7. DOCUMENTOS XML BIEN FORMADOS.

Un objeto de texto es un documento XML bien formado si:

- Tomado como un todo, cumple la regla denominada «document».
- Respetar todas las restricciones de buena formación dadas en la especificación.
- Cada una de las entidades analizadas que se referencia directa o indirectamente en el documento está bien formada.

### 2.7.1. La regla "document".

Cumplir la regla "document" antes mencionada significa:

1. Que continene uno o más elementos.
2. Hay exactamente un elemento, llamado raíz, o elemento documento, del cual ninguna parte aparece en el contenido de ningún otro elemento.
3. Para el resto de elementos, si la etiqueta de comienzo está en el contenido de algún otro elemento, la etiqueta de fin está en el contenido del mismo elemento. Es decir, los elementos delimitados por etiquetas de principio y final se anidan adecuadamente mutuamente.

El siguiente ejemplo no es un documento XML bien formado:

Mi primer documento XML

ya que no contiene ningún elemento y, por tanto, está incumpliendo la regla número 1.

En cambio:

<p>Mi primer documento XML</p>

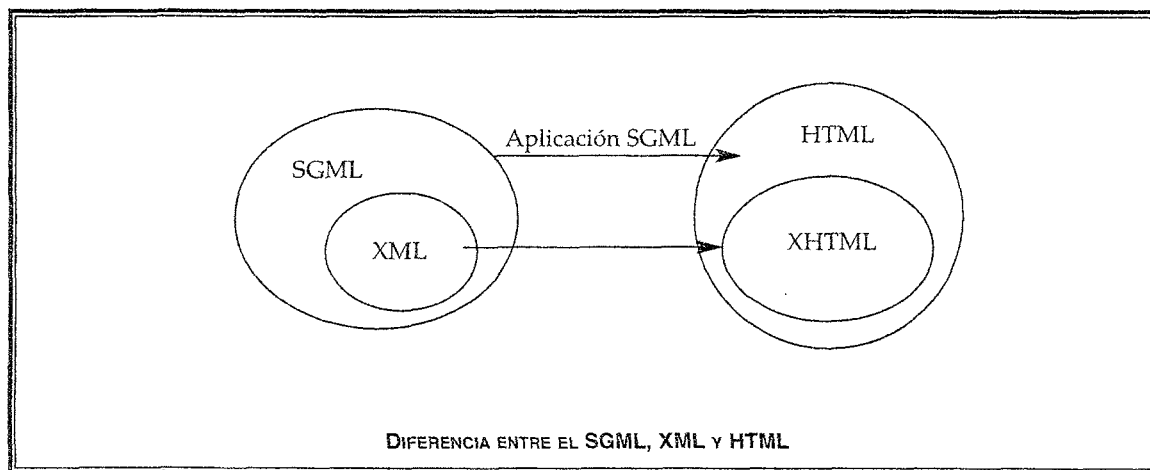
sí que lo es, al contener al menos el elemento "p". La principal razón por la que el procesador comprueba los elementos es para determinar si el documento tiene estructura de datos que pueda extraer. Un documento que carece de elementos no tiene estructura de datos. Un documento con al menos un elemento tiene estructura de datos.

### 3. HTML, XML, *VERSUS* SGML.

Como ya se ha indicado, tanto el XML como el HTML tienen su base en el SGML. El SGML (Standard Generalized Markup Language, ISO 8879) es el estándar internacional para la definición de la estructura y el contenido de diferentes tipos de documentos electrónicos. Es decir, es un metalenguaje que nos permite crear lenguajes para definir la estructura y el contenido de nuestros documentos. La definición de la estructura y el contenido de un tipo de documento se realiza en una DTD. En ella definimos los elementos que conformarán ese tipo de documentos y cómo tienen que estar organizados para que sea correcto.

Un ejemplo de DTD es por ejemplo la que define cómo tendrán que ser los documentos HTML. Por tanto, el HTML no es más que un tipo de documento SGML que se utiliza en la Web, y esto es importante, ya que aquí radica su principal diferencia con el XML.

El XML no es ningún tipo de documento SGML, sino que es una versión abreviada de SGML optimizada para su utilización en Internet. Esto significa que con él vamos a poder definir nuestros propios tipos de documentos (podremos definir nuestras propias etiquetas) y, por tanto, ya no dependeremos de un único e inflexible tipo de documento HTML.



El XML hay que considerarlo como un SGML optimizado para su utilización en Internet. "XML ofrece el 80 por 100 de las ventajas del SGML con un 20 por 100 de su complejidad". Y es que los diseñadores de XML intentaron dejar fuera sólo aquellas partes que raramente se utilizan. Esta reducción resultó ser muy importante: la especificación XML ocupa aproximadamente 30 páginas, frente a las 500 del SGML. Actualmente el W3C está involucrado en la redefinición del lenguaje HTML utilizando el XML, el resultado ha sido XHTML, el cual se pretende sea el estándar sobre el que evolucione el lenguaje de etiquetas más usado en la actualidad.



#### 4. CUADRO-RESUMEN CON ALGUNAS DE LAS DIFERENCIAS SIGNIFICATIVAS.

	HTML/DHTML	XML	SGML
Gramática	Fija y no ampliable	Extensible	Extensible
Estructura	Monolítica	Jerárquica	Jerárquica
Número de marcas	Fijas	Sin límite	Sin límite
Complejidad	Baja	Mediana	Alta
Diseño de páginas	Fijado por tags. Etiquetas con atributos CSS en DHTML	CSS ó DSSSL	DSSSL
Enlaces	Simple enlaces	Poderosos enlaces (XLL)	HyTime
Exportabilidad (formatos/aplicaciones)	No	Sí	Sí
Validación	Sin validación (8)	Pueden validarse	Obligatorio DTD
Búsquedas	Simple y a veces resuelta por scripts o CGI	Potente búsqueda. Con capacidad para personalizarla	Son posibles potentes búsquedas.
Indización/Catalogación de páginas web	Sólo lo permite los atributos de la etiqueta <META>, e implementaciones como DC.	Una descripción abierta y personalizable con el RDF.	Algún proyecto como TEI, DLI, etc.

#### BIBLIOGRAFÍA

- ALADRO GARCÍA, A. «El lenguaje XML: la nueva forma de estructurar los contenidos». Net Magazine, año IV, n.º 34, p. 74-77.
- MACE, S. [et al.]. «Tejer mejor la red». Byte España, n.º 38.
- PEÑA TRESANCOS, J. «Estándar XML 1.0: tecnologías para Internet». PC World, n.º 144.
- MACK, E. S.; PLATT, J. *HTML 4.0*. Madrid: Anaya Multimedia, 1998.
- CONNOLLY, D. Evolution of Web Data Formats [en línea]. <http://www.w3.org/Talks/9803xml-seattle/>
- BOSAK, J. «XML, Java, and the future of the Web» [en línea]. Sun Microsystems <http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm>.

- SPERBERG-MCQUEEN, C. M. What is XML and Why Should Humanists Care? [en línea]. Chicago: University of Illinois, 1998. <http://users.ox.ac.uk/~drh97/Papers/Sperberg.html>.
- KHARE, R.; RIFKIN, A. Capturing the State of Distributed Systems with XML [en línea]. Word Wide Web Journal, vol. 2, n.º 4, 1997, p. 207-218. Disponible también en:  
<http://www.cs.caltech.edu/~adam/papers/xml/xml-for-archiving.html>.
- CONNOLLY, D.; KHARE, ROHIT; RIFKIN, ADAM. The evolution of the Web documents: the ascent of XML [en línea]. Word Wide Web Journal, vol. 2, n.º 4, 1997, p. 119-128. Disponible también en: <http://www.cs.caltech.edu/~adam/papers/xml/ascent-of-xml.html>.
- LANDER, R. The new markup wave [en línea].  
[http://www.csclub.uwaterloo.ca/u/relander/XML/Wave/xml\\_mw.html](http://www.csclub.uwaterloo.ca/u/relander/XML/Wave/xml_mw.html).

Más información en:

- What is XML? : [http://www.gca.org/conf/xml/xml\\_what.htm](http://www.gca.org/conf/xml/xml_what.htm)
- POET XML Resource Library: [http://www.poet.com/xml/xml\\_lib.html](http://www.poet.com/xml/xml_lib.html)
- El site oficial del W3C: <http://www.w3.org/>
- XML.com : <http://www.xml.com/xml/pub>
- En The Summer Institute of Linguistics: <http://www.sil.org/sgml/xml.html>
- What the ¿XML!: <http://www.geocities.com/SiliconValley/Peaks/5957/xml.html>
- Abaitua, Joseba. Todo sobre SGML/XML. <http://www.deusto.es/~abaitua/konzeptu/sgml.htm>
- Grupo de Trabajo de SGML/XML. <http://www.promeko.com/informatica/gtsx/>

