

Lenguajes de Marcas y Sistemas de Gestión de Información

Características del Lenguaje de Marcas

1. Lenguaje de Marcas

¿ Qué es un Lenguaje de Marcas?

Un lenguaje de marcas es un modo de codificar un documento donde, junto con el texto, se incorporan etiquetas, marcas o anotaciones con información adicional relativa a la estructura del texto o su formato de presentación

Ejemplo

Aspecto de un documento realizado con un lenguaje de marcas:

```
<carta>
```

```
  <fecha>22/11/2006</fecha>
```

```
  <presentación>Estimado cliente:</presentación>
```

```
  <contenido>bla bla bla bla ...</contenido>
```

```
  <firma>Don Jose Gutiérrez González</firma>
```

```
</carta>
```

Clasificación de los lenguajes de marcas

- De presentación: Define el formato del texto
- De procedimientos: Se orienta también a la presentación, pero en este caso, el programa que representa el documento debe interpretar el código en el mismo orden en que aparece.
- Descriptivo o semántico: Describen las diferentes partes en las que se estructura el documento, pero sin especificar cómo deben representarse.

- Ejemplo de presentación:

Ejemplo: `Texto en negrita`, `<p>Párrafo</p>`.

- Ejemplo semántico descriptivos:

Ejemplo: `<libro><titulo>Aprendiendo XML</titulo><autor>Juan Pérez</autor></libro>`



- Documentación electrónica

- RTF
- TeX
- Wikitexto
- Docbook

- Tecnologías de internet

- HTML y XHTML



- RSS



Otros lenguajes especializados

- MathML
- VoiceXML
- MusicXML

- ¿Por qué creemos que necesitamos lenguajes de marcas en lugar de formatos de texto sin estructura?
- ¿Qué ventajas ofrecen las etiquetas a la hora de representar contenido o datos?

2. Evolución de los lenguajes de marcas

Los lenguajes de marcas surgieron como lenguajes formados por el conjunto de códigos de formato que los procesadores de textos introducen en los documentos para dirigir el proceso de presentación mediante una impresora.

Ejemplo

Código de marcas anterior a GML. Las etiquetas son de invención propia.

Dado el siguiente documento:

`<times 14><color verde><centrado>` Este texto es un ejemplo para mostrar la utilización primitiva de las marcas `</centrado></color></times 14>`

`<color granate><times 10><cursiva>`Para realiza este ejemplo se utilizan etiquetas de nuestra invención. `</cursiva>` Las partes importantes del texto pueden resaltarse usando la `<negrita>negrita</negrita>`, o el `<subrayar>subrayado</subrayar></times 10></color>`

Al imprimirlo se obtendría:

Este texto es un ejemplo para mostrar la utilización primitiva de las marcas

*Para realiza este ejemplo se utilizan etiquetas de nuestra invención. Las partes importantes del texto pueden resaltarse usando la **negrita**, o el subrayado*

Posteriormente, se añadieron como medio de presentación a la pantalla. Los códigos de estilo anteriores ya no aparecen, y se emplean otros medios para marcados, ahora se automatiza y basta con pulsar una combinación de teclas, o un simple botón, para lograr los resultados requeridos.

2.1 GML (Generalized Markup Language)

Uno de los problemas que se conoces es la falta de estandarización en los formatos de información usados por los distintos programas.

Fue creado por IBM en los años 60 para estandarizar el formato de documentos legales, GML fue el primer lenguaje de marcas que permitía describir documentos de manera que pudieran ser interpretados independientemente de la máquina o software.

2.2 SGML (Standard Generalized Markup Language)

SGML es la evolución de GML y se convirtió en un estándar ISO en 1986. Fue un lenguaje complejo y costoso de implementar, pero se usaba en aplicaciones industriales importantes.

Ventajas de SGML: Flexibilidad para describir documentos de cualquier tipo. Sin embargo, requería herramientas caras y una gran cantidad de conocimientos técnicos.

Ejemplo

Documento SGML sencillo:

```
<email>
  <remitente>
    <persona>
      <nombre> Pepito </nombre>
      <apellido> Grillo </apellido>
    </persona>
  </remitente>
  <destinatario>
    <direccion> pinocho@hotmail.com </direccion>
  </destinatario>
  <asunto>¿quedamos?</asunto>
  <mensaje> Hola, he visto que ponen esta noche la película que querías ver. ¿Te apetece
  ir?</mensaje>
</email>
```

2.3 HTML (HyperText Markup Language)

En 1989, Tim Berners-Lee creó el HTML como una simplificación de SGML para la Web. HTML utiliza etiquetas simples para estructurar texto y otros medios en una página web.

HTML es una combinación de dos estándares ya existentes: ASCII y SGML.

Desventajas de HTML:

- No soporta tareas de impresión y diseño.
- El lenguaje no es flexible, ya que las etiquetas son limitadas.
- No permite mostrar contenido dinámico.
- La estructura y el diseño están mezclados en el documento.

Ejemplo

Documento HTML

```
<html>
  <head>
    <title> Ejemplo de código HTML</title>
  </head>
  <body bgcolor="#ffffff">
    <p></p>
    <p>
      <b>20 de octubre de 2010</b>
    </p>
    <p><b> Bienvenido al modulo de "Lenguajes de Marcas y Sistemas de Gestión de
Información" </b></p>
    <p> En este curso aprender&aacute;s, entre otras cosas:<br/>
      <ul>
        <li>Las ventajas que ofrece XML </li>
        <li>La creaci&oacute;n de documentos bien formados </li>
        <li>La creaci&oacute;n de DTD</li>
      </ul>
    </p>
  </body>
</html>
```



2.4 XML (eXtensible Markup Language)

Para superar las limitaciones de HTML, en 1998 el W3C introdujo XML, un lenguaje flexible que permite la definición de etiquetas propias, sin restricciones de presentación.

XML es un metalenguaje caracterizado por:

- Permite definir etiquetas propias.
- Permite asignar atributos a las etiquetas.
- Utilizar un esquema para definir de forma exacta las etiquetas y atributos.
- La estructura y el diseño son independientes.

XML es, en realidad, un conjunto de estándares relacionados entre sí, que son:

- XSL
- XML Linking Language
- XML Namespaces
- XML Schemas

¿Por qué XML se convirtió en una solución importante para el intercambio de datos en la web y entre sistemas?

3. Comparación entre HTML y XML

XML

- Es un perfil de SGML.
- Especifica cómo deben definirse conjuntos de etiquetas aplicables a un tipo de documento.
- Modelo de hiperenlaces complejo.
- El navegador es una plataforma para el desarrollo de aplicaciones.
- Fin de la guerra de los navegadores y etiquetas propietarias.

HTML

- Es una aplicación de SGML.
- Aplica un conjunto limitado de etiquetas sobre un único tipo de documento.
- Modelo de hiperenlaces simple.
- El navegador es un visor de páginas.
- El problema de la "No compatibilidad" y las diferencias entre navegadores ha alcanzado un punto en el que la solución es difícil.

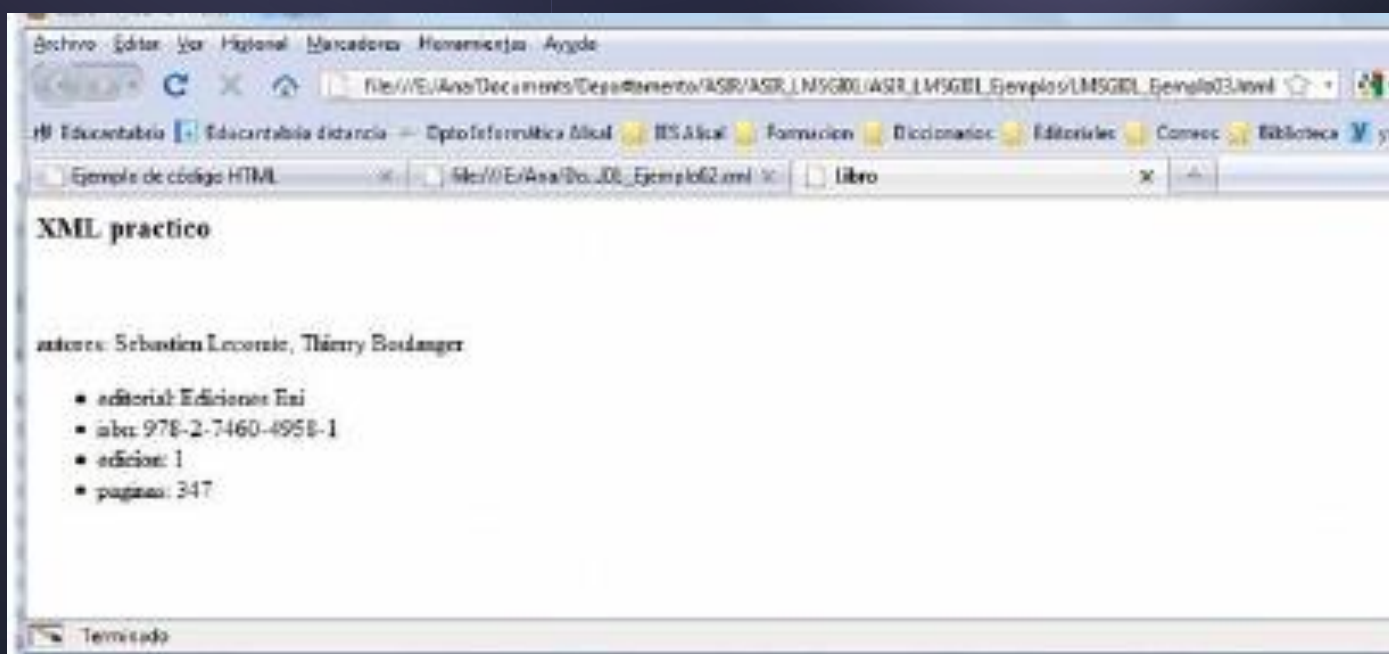
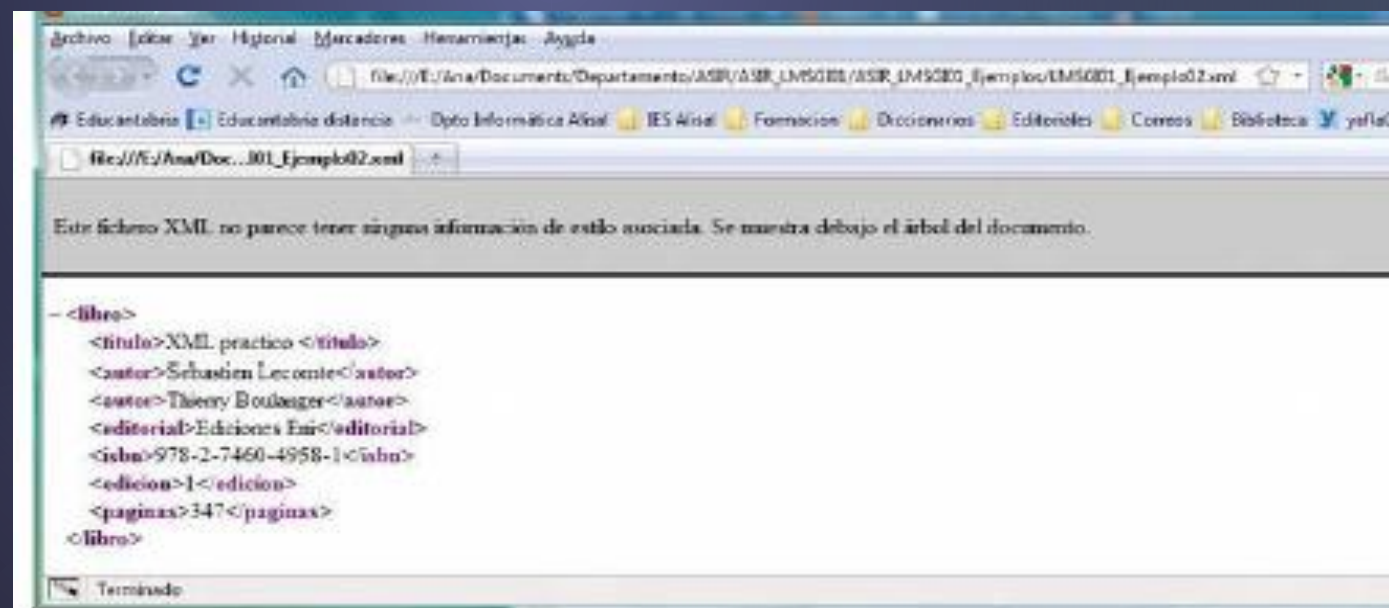
Ejemplo

Fichero XML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE libro>
<libro>
  <titulo>XML practico </titulo>
  <autor>SebastienLecomte</autor>
  <autor>Thierry Boulanger</autor>
  <editorial>Ediciones Eni</editorial>
  <isbn>978-2-7460-4958-1</isbn>
  <edicion>1</edicion>
  <paginas>347</paginas>
</libro>
```

Fichero HTML

```
<html>
  <head>
    <title>Libro</title>
  </head>
  <body>
    <h3>XML practico</h3><br>
    <p>autores: Sebastien Lecomte,
    Thierry Boulanger</p>
    <ul>
      <li>editorial: Ediciones Eni</li>
      <li>isbn:978-2-7460-4958-1</li>
      <li>edicion: 1 </li>
      <li>paginas: 347</li>
    </ul>
  </body>
</html>
```

4. Comparación de XML con SGML

XML

- Su uso es sencillo.
- Trabaja con documentos bien formados, no exige que estén validados.
- Facilita el desarrollo de aplicaciones de bajo coste.
- Es muy utilizado en informática y en más áreas de aplicación.
- Compatibilidad e integración con HTML.
- Formateo y estilos fáciles de aplicar.
- No usa etiquetas opcionales

SGML

- Su uso es muy complejo.
- Sólo trabaja con documentos válidos.
- Su complejidad hace que las aplicaciones sean muy costosas.
- Solo se utiliza en sectores muy específicos.
- No hay una compatibilidad con HTML definida.
- Formateo y estilos relativamente complejos.

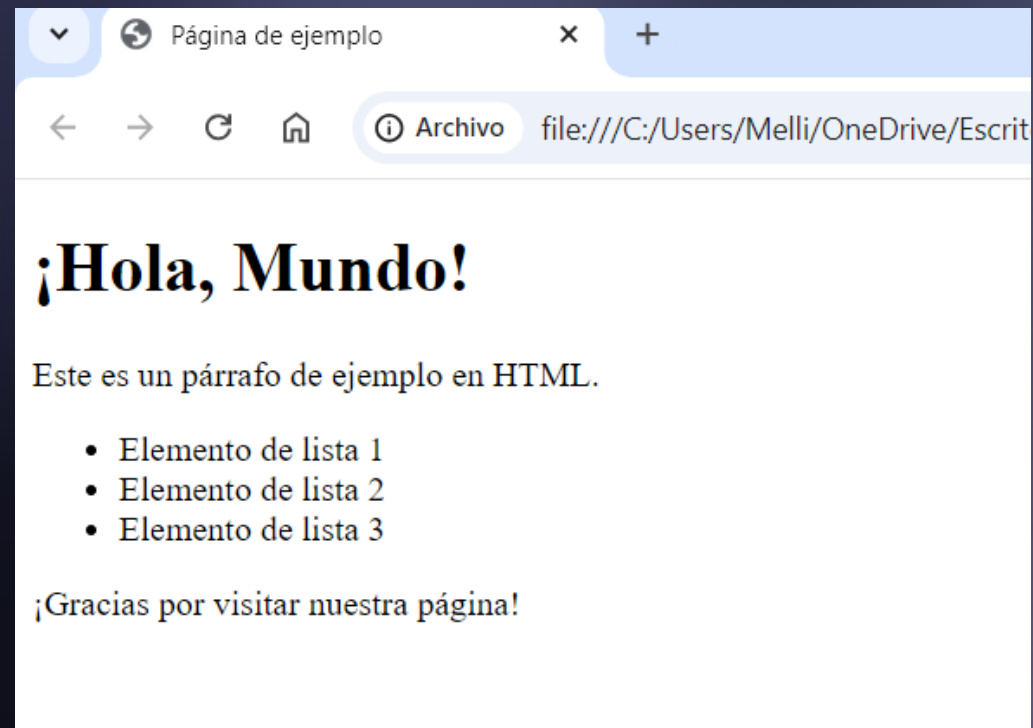
- ¿En qué situaciones sería más útil XML que HTML?
- ¿Cómo se relaciona la flexibilidad de XML con su uso en aplicaciones modernas?

5. Etiquetas

- Los lenguajes de marcas utilizan una serie de etiquetas especiales intercaladas en un documento de texto sin formato. Dichas etiquetas serán interpretadas por los intérpretes del lenguaje y ayudan al procesamiento del documento.
- Las etiquetas se escriben encerradas entre ángulos, es decir, < y >.
- Normalmente se utilizan dos etiquetas: una de inicio y otra de fin para indicar el efecto que queríamos presentar. La diferencia es que la de cierre lleva una barra inclinada "/".

Ejemplo de HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Página de ejemplo</title>
</head>
<body>
  <h1>¡Hola, Mundo!</h1>
  <p>Este es un párrafo de ejemplo en HTML.</p>
  <ul>
    <li>Elemento de lista 1</li>
    <li>Elemento de lista 2</li>
    <li>Elemento de lista 3</li>
  </ul>
  <p>¡Gracias por visitar nuestra página!</p>
</body>
</html>
```



Explicación de elementos

- DOCTYPE: Declara el tipo de documento como HTML5.
- Html Lang="es" : Indica que el contenido está en español.
- Meta charset = "UTF-8" : Asegura la correcta visualización de caracteres especiales.
- Meta name = "viewport" : Ayuda a la página a ser compatible con dispositivos móviles.
- H1: Título principal de la página.
- P: Párrafos de texto.
- Ul y Li: Lista no ordenada con elementos

Ejercicio para clase

1.- Escribe en un documento PDF el código correspondiente a un documento XML, de forma simple, en el que se describa una colección de libros, que contenga al menos tres libros y con los siguientes elementos: Título, autor, género, año de publicación y precio.

2.- Realiza un documento XML, de forma simple, sobre una colección de álbumes, con al menos tres, que contenga los siguientes elementos: Título, artista, género, lanzamiento y precio.

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
  <book>
    <title>El Quijote</title>
    <author>Miguel de Cervantes</author>
    <genre>Novela</genre>
    <published>1605</published>
    <price currency="EUR">19.99</price>
  </book>
  <book>
    <title>Cien Años de Soledad</title>
    <author>Gabriel García Márquez</author>
    <genre>Realismo mágico</genre>
    <published>1967</published>
    <price currency="USD">25.99</price>
  </book>
  <book>
    <title>1984</title>
    <author>George Orwell</author>
    <genre>Ciencia ficción</genre>
    <published>1949</published>
    <price currency="GBP">15.50</price>
  </book>
</library>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<coleccionMusica>
  <album>
    <titulo>Thriller</titulo>
    <artista>Michael Jackson</artista>
    <genero>Pop</genero>
    <lanzado>1982</lanzado>
    <precio moneda="USD">15.99</precio>
  </album>
  <album>
    <titulo>The Dark Side of the Moon</titulo>
    <artista>Pink Floyd</artista>
    <genero>Rock</genero>
    <lanzado>1973</lanzado>
    <precio moneda="GBP">12.50</precio>
  </album>
  <album>
    <titulo>Back to Black</titulo>
    <artista>Amy Winehouse</artista>
    <genero>Soul</genero>
    <lanzado>2006</lanzado>
    <precio moneda="EUR">10.99</precio>
  </album>
```

Explicación de la estructura

1. `<?xml version="1.0" encoding="UTF-8"?>`: Define la versión de XML y el tipo de codificación de caracteres.
2. `<library>`: Elemento raíz que contiene la colección de libros.
3. `<book>`: Cada libro está descrito en un elemento `<book>`.
4. `<title>`, `<author>`, `<genre>`, `<published>`, `<price>`: Son elementos hijos de `<book>` que describen atributos específicos del libro.
5. Atributo `currency` en `<price>`: Indica la moneda del precio.

6. Herramientas básicas de XML

Para trabajar en XML es necesario editar los documentos y luego procesarlos, por tanto, necesitamos dos tipos de herramientas:

- Editores XML.
- Procesadores XML

6.1 Editores XML

- Una característica de los lenguajes de marcas es que se basan en la utilización de ficheros de texto plano por lo que podemos utilizar un procesador de texto normal.
- Notepad++ o Sublime Text.
- Para documentos algo más complejos sería conveniente usar algún editor XML. Uno de ellos, creado por el W3C es Amaya, que soporta HTML, CSS y XML.

6.2 Procesadores XML

Para interpretar el código XML se puede utilizar cualquier navegador. Los procesadores permiten leer los documentos XML y acceder a su contenido y escritura.

Un procesador es un conjunto de módulos de software entre los que se encuentra un analizador de XML que comprueba que el documento cumple las normas establecidas para que pueda abrirse.

7. XML: Estructura y sintaxis

El XML es lenguaje de etiquetas, creadas por el programador, que estructuran y guardan de forma ordenada la información.

El XML ahorra tiempos de desarrollo y proporciona ventajas, dotando a webs y a aplicaciones de una forma realmente potente de guardar información.

Al igual que en HTML, un documento XML es un documento de texto, con extensión “.xml”, compuesto de parejas de etiquetas, estructuradas en árbol, que describen una función en la organización del documento.

7.1 Características básicas de XML

- Es directamente compatible con protocolos que ya funcionan, como HTTP y los URL.
- Todo documento que verifique las reglas de XML está conforme con SGML.
- No se requieren conocimientos de programación para realizar tareas sencillas en XML.
- Los documentos XML son fáciles de crear.
- La difusión de los documentos XML está asegurada ya que cualquier procesador de XML puede leer un documento de XML.
- El marcado de XML es legible y aplicable a una gran variedad de situaciones.
- El diseño XML es formal y conciso.
- XML es extensible, adaptable y aplicable a una gran variedad de situaciones.
- XML es orientado a objetos.
- Todo documento XML se compone exclusivamente de datos de marcado y datos carácter entremezclados.

El proceso de creación de un documento XML pasa por varias etapas:

- Especificación de requisitos.
- Diseño de etiquetas.
- Marcado de los documentos.

Los datos carácter son los que forman la verdadera información del documento XML.

Los documentos XML pueden tener comentarios, que no son interpretados por el intérprete XML. Estos se incluyen entre las cadenas "`<!--`" y "`-->`"

No se pueden escribir ni Antes del Prologo ni Dentro de una etiqueta.

7.2 El prólogo

El prólogo debe preceder al ejemplar del documento. Su inclusión facilita el procesamiento de la información del ejemplar. El prólogo está dividido en dos partes:

- La declaración XML: Ha de ser la primera línea del documento, de no ser así se generará un error que impide que el documento sea procesado.

El prólogo puede tener tres funciones:

- Declaración de la versión de XML usada.
- Declaración de la codificación.
- Declaración de la autonomía.

Estándar ISO	Código de país
UTF-8 (Unicode)	Conjunto de caracteres universal
ISO -8859-1 (Latin-1)	Europa occidental, Latinoamérica
ISO -8859-2 (Latin-2)	Europa central y oriental
ISO -8859-3 (Latin-3)	Sudoeste de Europa
ISO -8859-4 (Latin-4)	Países Escandinavos, Bálticos
ISO -8859-5	Cirílico
ISO -8859-6	Árabe
ISO -8859-7	Griego
ISO -8859-8	Hebreo
ISO -8859-9	Turco
ISO-8859-10	Lapón. Nórdico, esquimal
EUC-JP oder Shift JIS	Japonés

Declaración de la versión de XML usada:

```
<?xml versión="1.0"?>
```

Declaración de la codificación empleada:

```
<?xml versión="1.0" encoding=iso-8859-1?>
```

Declaración de la autonomía del documento:

```
<?xml versión="1.0" encoding="iso-8859-1" standalone="no" ?>
```

Declaración del tipo de documento: <!DOCTYPE Nombre_tipo...>

7.3 El ejemplar. Los elementos

Es la parte más importante de un documento XML, ya que contiene los datos reales del documento. Está formado por elementos anidados.

Los elementos son los distintos bloques de información que permiten definir la estructura de un documento XML.

En realidad, el ejemplar es el elemento raíz de un documento XML. Todos los datos de un documento XML han de pertenecer a un elemento de este.

La formación de elementos ha de cumplir ciertas normas para que queden perfectamente definidos, las cuáles son:

- En todo documento XML debe existir un elemento raíz, y solo uno.
- Todos los elementos tienen una etiqueta de inicio y otra de cierre. Si existen elementos vacíos se pueden sustituir las etiquetas de apertura y cierre por una de elemento vacío.
- Al anidar elementos hay que tener en cuenta que no puede cerrarse un elemento que contenga algún otro elemento que no haya sido cerrado.
- Los nombres de las etiquetas de apertura y de cierre han de ser idénticos.

- El contenido de los elementos no puede contener la cadena "]]>" por compatibilidad con SGML.
- No pueden aparecer directamente: <, >, &, " y '.

Carácter	Cadena	Carácter	Cadena	Carácter	Cadena
>	>	&	&	'	'
<	<	"	"		

- Para utilizar caracteres especiales se han de escribir mediante expresiones &#D o &#H, que indican el número decimal o hexadecimal correspondiente al carácter.

Para € se utilizarían € o €.

7.3.1 Atributos

Permiten añadir propiedades a los elementos del documento.

No pueden organizarse en ninguna jerarquía, no pueden contener ningún otro elemento o atributo y no reflejan ninguna estructura lógica.

No se debe utilizar un atributo para contener información susceptible de ser dividido.

Ejemplo

Dado el siguiente código XML:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?  
<!DOCTYPE biblioteca >  
<biblioteca>  
  <ejemplar tipo_ejem="libro" titulo="XML práctico" editorial="Ediciones Eni">  
    <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paginas="347"></libro> </tipo>  
    <autor nombre="Sebastien Lecomte"></autor>  
    <autor nombre="Thierry Boulanger"></autor>  
    <autor nombre="Angel Belinchon Calleja" funcion="traductor"></autor>  
    <prestado lector="Pepito Grillo">  
      <fecha_pres dia="13" mes="mar" año="2009"></fecha_pres>  
      <fecha_devol dia="21" mes="jun" año="2009"></fecha_devol>  
    </prestado>  
  </ejemplar>  
</biblioteca>
```

Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

```
-<biblioteca>
  -<ejemplar tipo Ejem="libro" titulo="XML práctico" editorial="Ediciones Eni">
    -<tipo>
      <libro isbn="978-2-7460-4958-1" edicion="1" paginas="347"/>
    </tipo>
    <autor nombre="Sebastien Lecomte"/>
    <autor nombre="Thierry Boulanger"/>
    <autor nombre="Angel Belinchon Calleja" funcion="traductor"/>
  -<prestado lector="Pepito Grillo">
    <fecha_pres dia="13" mes="mar" año="2009"/>
    <fecha_devol dia="21" mes="jun" año="2009"/>
  </prestado>
</ejemplar>
</biblioteca>
```

8. Documentos XML bien formados

Todos los documentos XML deben verificar las reglas sintácticas que define la recomendación del W3C para el estándar XML. Estas normas son:

- El documento ha de tener definido un prólogo con la declaración xml completa.
- Existe un único elemento raíz para cada documento.
- Hay que cumplir las reglas sintácticas del lenguaje XML para definir los distintos elementos y atributos del documento.