TEMA 1. INTRODUCCIÓN A LOS LENGUAJES DE MARCA

1. Lenguajes de marcas.

Un "lenguaje de marcas" **es una forma 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. Permiten hacer explícita la estructura de un documento, su contenido semántico o cualquier otra información lingüística o extralingüística que se quiera hacer patente.

Los lenguajes de marca suelen confundirse con lenguajes de programación. Sin embargo, no son lo mismo, ya que el lenguaje de marcado no tiene funciones aritméticas o variables ni elementos para controlar el flujo del programa, como sí poseen los lenguajes de programación. Para cada lenguaje de marca, los desarrolladores de software pueden construir una aplicación para leer los documentos escritos en ese lenguaje. Por ejemplo, los navegadores web leerán los documentos HTML. Los documentos escritos en XML pueden leerse por medio de aplicaciones personalizadas.

Todo lenguaje de marcas está definido en un documento denominado **DTD (Document Type Definition).** Un DTD es un documento que define la estructura válida de un documento creado mediante un lenguaje de marca. En él se establecen las marcas, los elementos utilizados por dicho lenguaje y sus correspondientes etiquetas y atributos, su sintaxis y normas de uso.

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 José Gutiérrez González</firma>
</carta>
```

Aunque en la práctica, en un mismo documento pueden combinarse varios tipos diferentes de lenguajes de marcas, éstos se pueden clasificar como sigue:

- **Lenguajes de presentación:** Indica el formato del texto, es decir cómo ha de presentarse el documento. La etiquetas de marcado suelen estar ocultas al usuario. Es típico en los procesadores de texto.
- **Lenguajes de procedimientos:** Se trata de documentos en los que hay texto marcado especialmente que en realidad se interpreta como **órdenes a seguir** y así el archivo en realidad contiene instrucciones a realizar con el texto. Se utiliza sobre todo para presentar el documento bien. Es el caso de **LaTeX** o **PostScript o Html**.
- **Lenguajes descriptivo o semántico:** En ellos las marcas especiales permiten **dar significado al texto**, pero no indican cómo se debe presentar en pantalla el mismo. Sería el caso de **XML** (o de **SGML**) y **JSON** en los que la presentación nunca se indica en el documento; simplemente se indica una semántica de contenido que lo hace ideal para almacenar datos (por ejemplo, si el texto es un nombre de persona o un número de identificación fiscal).

Algunos ejemplos de lenguajes de marcado agrupados por su ámbito de utilización son:

- **Documentación electrónica**
  - **RTF** (Rich Text Format): Formato de Texto Enriquecido, fue desarrollado por Microsoft en 1987. Permite el intercambio de documentos de texto ente distintos procesadores de texto.
  - **TeX**: Su objetivo es la creación de ecuaciones matemáticas complejas.
- **Tecnologías de internet**
  - **HTML**, **XHTML**: (Hypertext Markup Language, eXtensible Hypertext Markup Language): Su objetivo es la creación de páginas web.
- Otros lenguajes especializados
  - **MathML** (Mathematical Markup Language): Su objetivo es expresar el formalismo matemático de tal modo que pueda ser entendido por distintos sistemas y aplicaciones.
  - **VoiceXML** (Voice Extended Markup Language) tiene como objetivo el intercambio de información entre un usuario y una aplicación con capacidad de reconocimiento de habla.
  - **MusicXML**: Permite el intercambio de partituras entre distintos editores de partituras.

2. Evolución de los lenguajes de marcas.

En los años 70 surgen unos lenguajes informáticos, distintos de los lenguajes de programación, orientados a la gestión de información. Con el desarrollo de los editores y procesadores de texto surgen los primeros lenguajes informáticos especializados en tareas de descripción y estructuración de información: los lenguajes de marcas.

Los lenguajes de marcas surgieron, inicialmente, como lenguajes formados por el conjunto de códigos de formato que los procesadores de texto introducen en los documentos. Estos códigos de formato estaban ligados a las características de una máquina, programa o procesador de textos concreto y, en ellos, inicialmente no había nada que permitiese al programador abstraerse de las características del procesador de textos y expresar de forma independiente a éste la estructura y la lógica interna del documento.

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 de visualización anteriores ya no aparecen, y se emplean otros medios para marcados, distintos de la inclusión a mano de cadenas formateadoras, ahora ese proceso se automatiza y basta pulsar una combinación de teclas, o pulsar un botón, para lograr los resultados requeridos. Aunque esto es sólo una abstracción, para su uso interno las aplicaciones siguen utilizando marcas para delimitar aquellas partes del texto que tienen un formato especial.

2.1 GML (Generalized Markup Language).

El primer lenguaje que se creó fue el lenguaje **GML** (General Markup Language o lenguaje de marcas generalizado), cuyo objetivo era describir los documentos de tal modo que el resultado fuese independiente de la plataforma y la aplicación utilizada. Este lenguaje se desarrolló para la edición, almacenamiento y búsqueda de documentos legales. Y rápidamente su uso se extendió a otros ámbitos.

2.2 SGML (Standard Generalized Markup Language).

El lenguaje **GML** evolucionó hasta que en 1986 dio lugar a un estándar que se denominó **SGML**. Este lenguaje surgió para permitir compartir información por parte de sistemas informáticos. Tuvo una gran aceptación, pero no consiguió asentarse del todo debido principalmente a su complejidad lo que provocaba que el software que generaba terminaba siendo excesivamente extenso y complejo. Por ello su uso ha quedado relegado a grandes aplicaciones industriales.

Ejemplo

Documento SGML sencillo:

```
<email>
    <remitente>
       <persona>
            <nombre> Pepito </nombre>
            <apellido> Grillo </apellido>
       </persona>
    </remitente>
    <destinatario>
         <dirección> pinocho@hotmail.com </dirección>
    </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 1989/90 **Tim Berners-Lee** creó el **World Wide Web** y también creó un lenguaje de descripción de documentos llamado **HTML.**

HTML se obtiene a partir de **SGML**. Se creó un lenguaje de marcado pensado para compartir información usando las redes de ordenadores y, de forma más general, a través de Internet. Era tan fácil de comprender que rápidamente tuvo gran aceptación logrando lo que no pudo SGML, **HTML** se convirtió en un estándar general para la creación de páginas web.

A pesar de todas estas ventajas **HTML** no es un lenguaje perfecto, sus principales desventajas son:

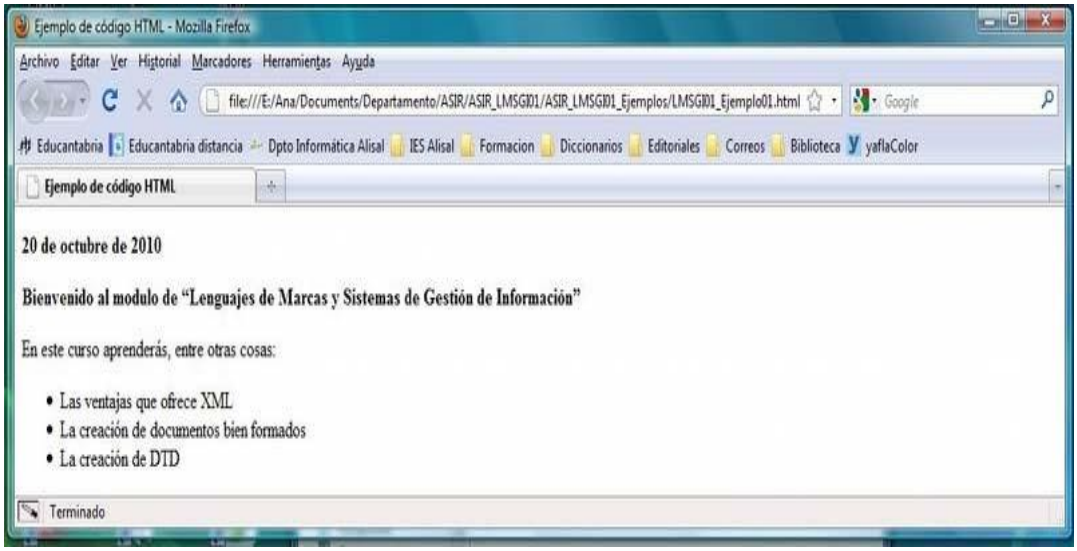
- El lenguaje no es flexible, ya que las etiquetas son limitadas.
- No permite mostrar contenido dinámico.
- La **estructura** y el **diseño** están mezclados en el documento.

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 módulo 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>
```

Al publicarlo en un navegador, por ejemplo, en el Firefox, tendríamos:



2.4 XML (eXtensible Markup Language).

El **W3C** establece, en 1998, el estándar internacional **XML**, un lenguaje de marcas puramente **estructural** que **no incluye ninguna información relativa al diseño**. Está convirtiéndose con rapidez en estándar para el intercambio de datos en la Web. **A**

diferencia de HTML las etiquetas indican el significado de los datos en lugar del formato con el que se van a visualizar los datos.

XML es un metalenguaje caracterizado por:

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

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

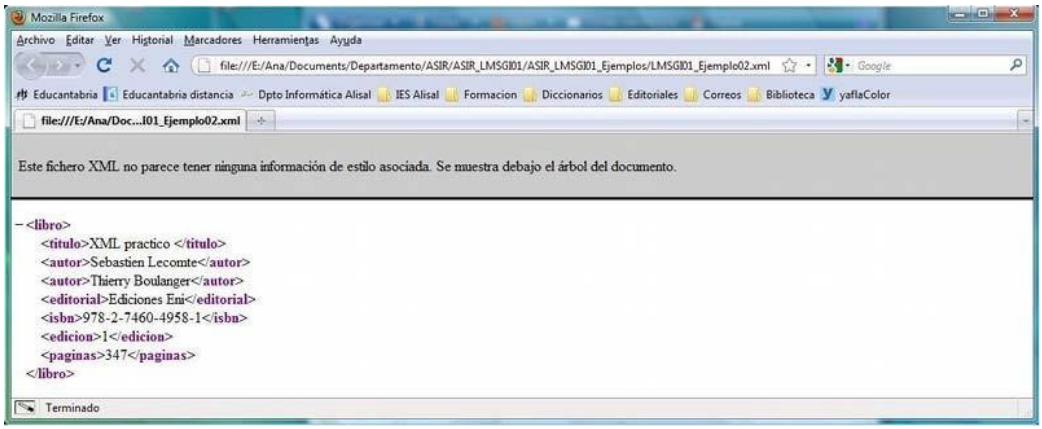
- **XSL**, eXtensible Style Language. Permite definir hojas de estilo para los documentos XML e incluye capacidad para la transformación de documentos.
- **XML Linking Language**, incluye Xpath, Xlink y Xpointer. Determinan aspectos sobre los enlaces entre documentos XML.
- **XML Schemas**. Permiten definir restricciones que se aplicarán a un documento XML. Actualmente los más usados son las DTD.

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>
```

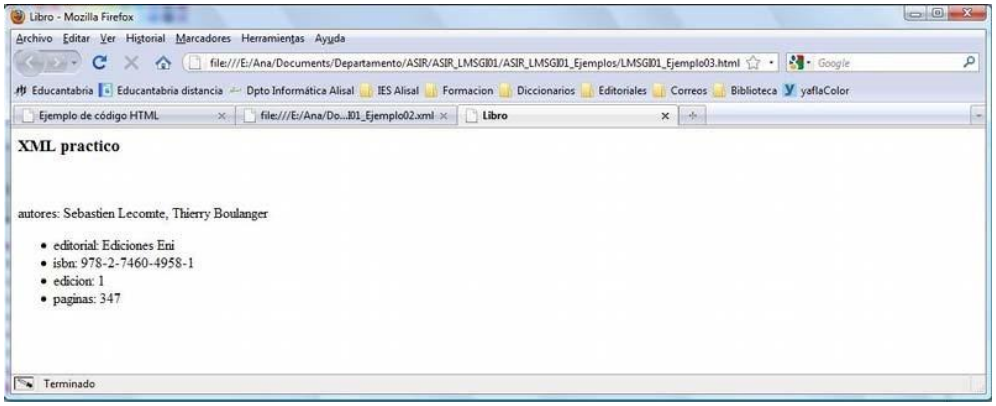
Al interpretar este fichero con un navegador, por ejemplo, Mozilla, se obtiene:



Fichero HTML

```
<html>
    <head>
        <title>Libro</title>
    <head>
     <body>
          <h3>XML práctico</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>páginas: 347</li>
        </ul>
   </ bodyy>
</ html>
```

Al interpretarlo con el navegador Mozilla Firefox tendremos:

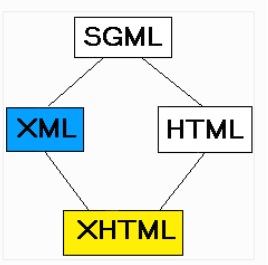


Como podemos ver el resultado que muestra el navegador con un documento **HTML** es muy diferente del que muestra con un fichero **XML**.

2.5 XHTML

XHTML (lenguaje extensible de marcado de hipertexto) surge en el año 2000, básicamente para expresar el lenguaje **HTML** como un lenguaje **XML** válido.

Las diferencias entre **HTML** y **XML** son: la sintaxis **HTML** está inspirada en la norma SGML (aunque no la cumple estrictamente), mientras que la sintaxis **XHTML** está basada en la recomendación **XML** (aunque tampoco la cumple estrictamente).



En general, la sintaxis XHTML es más "estricta", en el sentido de imponer restricciones en la forma de escribir etiquetas, atributos o valores, mientras que la sintaxis HTML es más "flexible".

Si un documento contiene errores de sintaxis (HTML o XHTML) se dice que es un documento **inválido**. Si las páginas web se sirven al navegador como **xhtml**, el navegador debe rechazar las páginas inválidas, pero si se sirven al navegador como **html**, los navegadores intentan mostrar el documento, aunque contenga errores. Aunque los navegadores a menudo consiguen mostrar documentos inválidos, se aconseja validar y corregir los documentos para asegurar que los navegadores puedan mostrarlos correctamente.

Por ejemplo:

- En **XHTML** para que un documento se muestre en el navegador tiene que estar "bien formado", es decir, tiene que cumplir todas las reglas sintácticas. En HTML, los documentos se muestran en el navegador, aunque contengan errores sintácticos (la visualización puede ser correcta o incorrecta, dependiendo del tipo de errores de la página).
- En XHTML no puede haber texto no incluido en alguna etiqueta, pero en HTML puede haberlo

3. Herramientas básicas para el uso de los lenguajes de marca.

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

3.1. Editores

Una característica de los lenguajes de marcas es que se basan en la utilización de ficheros de texto plano por lo que basta utilizar un procesador de texto normal y corriente para construir un documento XML o HTML.

Para crear documentos XML complejos e ir añadiendo datos es conveniente usar algún editor XML. Estos nos ayudan a crear estructuras y etiquetas de los elementos usados en los documentos, además algunos incluyen ayuda para la creación de otros elementos como DTD, hojas de estilo CSS o XSL, ... El W3C ha desarrollado un editor de HTML, XHTML, CSS y XML gratuito cuyo nombre es Amaya.

3.2 Procesadores

Para interpretar el código XML o HTML se puede utilizar cualquier navegador. Los procesadores de XML permiten leer los documentos XML y acceder a su contenido y estructura. 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. Estas normas pueden corresponderse con las necesarias para trabajar sólo con documentos de tipo válido o sólo exigir que el documento esté bien formado; los primeros se conocen como validadores y los segundos como no validadores. El modo en que los procesadores deben leer los datos XML está descrito en la recomendación de XML establecida por W3C.

Para publicar un documento XML en Internet se utilizan los procesadores XSLT, que permiten generar archivos HTML a partir de documentos XML.

Puesto que XML se puede utilizar para el intercambio de datos entre aplicaciones, hay que recurrir a motores independientes que se ejecutan sin que nos demos cuenta. Entre estos destacan "XML para Java" de IBM, JAXP de Sun, etc

4. Gramáticas

La **DTD** (Definición de Tipo de Documento) establece las reglas de formación del lenguaje, es decir, qué combinaciones de símbolos son sintácticamente correctas. La especificación del W3C para HTML 4.0 contempla:

- **DTD estricta**: incluye todos los elementos y atributos que no han sido declarados “desaprobados” (deprecated).
- **DTD transicional o flexible:** incluye todo lo de la anterior más los elementos y atributos desaprobados (deprecated).
- **DTD para documentos con marcos**: engloba todo lo incluido en la transicional más lo relativo a la creación de documentos con marcos (frames).

5. Organizaciones y estándares

La normalización o estandarización permite la creación de normas o estándares que establecen las características comunes que deben cumplir los productos y que deben ser respetadas por todos.

Aplicado al contexto de los lenguajes de marcas, sería por ejemplo el desarrollo de páginas web atendiendo a las especificaciones oficiales del lenguaje utilizado. Son un montón de normas para hacer bien una web.

Para la definición de estas normas existen organismos internacionales, nacionales incluso organizaciones privadas. Las organizaciones más importantes en materia de software son **W3C, ISO y Open Source.**

Según el propio W3C:

"El World Wide Web Consortium (W3C) es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo."

Bibliografía: apuntes José Luis Comesaña