UNIDAD 1 INTRODUCCIÓN A LOS LENGUAJES DE MARCAS

1 LENGUAJES DE MARCAS

Se puede decir que los archivos de marcas son una manera diferente de almacenar información en ordenadores que se añade a los modos de almacenar la información por medio de archivos binarios o archivos de texto.

Los ficheros de marcas toman como base los archivos de texto para aprovecharse de las características más interesantes de este tipo de archivos:

- La facilidad de creación y lectura.
- El cumplimiento de estándares de almacenamiento definidos y públicos.

Como los archivos de texto siempre están almacenados en algún código de caracteres conocido (ASCII, UTF-8, etc.) se consigue que puedan ser transportados y leídos en cualquier plataforma, sistema operativo o programa que pueda interpretar estos códigos de caracteres. Por lo tanto, los lenguajes de marcas se aprovecharán de esta característica, al estar basados en el formato de texto.

Además, de rebote, también tendrán la ventaja de que podrán ser abiertos y creados con los programas de edición de texto estándar. Desde editores tan simples como el Bloc de notas de los sistemas Windows o Gedit de sistemas Unix hasta editores más complejos como el Microsoft Word, pasando por editores especializados en XML como el Oxygen XML Editor.

Los ficheros de marcas, por lo tanto, se aprovechan de una de las grandes ventajas de los archivos de texto sobre los archivos binarios, ya que estos últimos requieren ser abiertos con un programa específico que pueda interpretar el formato.

Pero los archivos de marcas no solo intentan aprovechar de las características de los ficheros de texto sino que también intentan conseguir las características más interesantes de los ficheros binarios, como:

- La incorporación de metadatos.
- La definición de la estructura de los datos.

Esto hace que los lenguajes de marcas adquieran una de las características más interesantes de los ficheros binarios, que es la posibilidad de incorporar información sobre los datos -metadatos- pero intentando que afecte lo menos posible la legibilidad del documento.

También permiten definir los datos y su estructura de manera que sea sencillo para un programa poderlas interpretar.

Gracias a las ventajas que ofrecen los lenguajes de marcas, estos se han convertido rápidamente en una de las maneras habituales de representar datos y se pueden encontrar continuamente en las tareas habituales con ordenadores:

- El exponente más popular es Internet -el Web-, que está basado totalmente en los lenguajes de marcas.
- Muchos de los programas de ordenador que utilice habitualmente utilizan en algún momento alguna u otra forma de algún lenguaje de marcas para almacenar sus datos de configuración o de resultados.
- Internamente los formatos de documentos de Microsoft Office o de OpenOffice o LibreOffice están basados en lenguajes de marcas.
- Microsoft Visual Studio guarda su configuración utilizando lenguajes de marcas.

1.1 ¿Qué es un lenguaje de marcas?

Un lenguaje de marcas es un lenguaje en el que sus partes se diferencian entre si mediante señales. E s una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

Las marcas también están formadas de texto, pero que es interpretado cuando se muestra el documento, y suelen llamarse también etiquetas.

El lenguaje de marcas más popular es HTML.

Las marcas son una serie de códigos que se incorporan a los documentos electrónicos para determinar el formato, la forma en que se han de imprimir, la estructura de los datos, etc. Por lo tanto, son **anotaciones que se incorporan a los datos pero que no forman parte de ellos.**

Las marcas, por tanto, deben ser fácilmente distinguibles del texto normal (por su posición, porque siguen algún tipo de sintaxis, etc.). Las marcas más usadas son las que están formadas por textos descriptivos y están rodeadas de los símbolos de "más pequeño" (<) y "mayor" (>) y normalmente suele haber una al principio y al final:

```
<nombre > Manel Puig Garcia </ nombre >
```

Estas marcas pueden ser indentadas para indicar estructuras de datos:

```
<persona >
   <nombre > Manel Puig Garcia </ nombre >
   <nombre > Pedro González Puigdevall </ nombre >
   <nombre > María Pozos Canadell </ nombre >
</ persona >
```

Pero hay muchas otras formas de marcas. Otra idea consiste en encontrar alguna combinación de caracteres que salga raramente en el lenguaje habitual. El TeX utiliza las barras invertidas para indicar el inicio de las marcas:

```
\ Section {Personas}
\ Begin {itemize}
\ Item Manel Puig Garcia
\ Item Pedro González Puigdevall
\ Item María Pozos Canadell
\ End {itemize}
```

Otros lenguajes de marcas usan caracteres no habituales en determinadas posiciones para indicar que son marcas. Por ejemplo con Wiki Markup los caracteres "=" en la primera posición de una línea se usan para indicar que el texto es un título de apartado y el *por las listas de puntos:

```
= Personas =
* Manel Puig Garcia
* Pedro González Puigdevall
* María Pozos Canadell
```

La idea general es que es necesario que las marcas sean fácilmente identificables para podernos aprovechar de las ventajas que ofrecen los lenguajes de marcas.

1.2 Clasificación

Existen dos clases de lenguajes de marcas:

- **Marcas de procedimientos**: estas marcas se utilizan para la presentación del texto, interpretándose cada una en el orden que en aparecen. Por ejemplo, la marca que se agrega inmediatamente antes de un texto para que se vea en **negrita**. Luego debe existir la marca correspondiente que termine o cierre la negrita. Otras marcas de procedimientos pueden ser centrar texto, cambio de tamaño de fuente, cambios de estilos, etc. Algunos lenguajes de marcas de procedimiento son nroff, troff, TeX, PostScript, HTML, etc.

- **Marcas descriptivas:** También llamadas marcado descriptivo, o semántico. Aquí se utilizan las marcas para describir fragmentos de texto sin especificar cómo deben representarse. Algunos lenguajes diseñados para esto son el SGML y el XML. En los lenguajes de marcas descriptivas el formato está separado del contenido, permitiendo flexibilidad a la hora de reformatear un texto.

1.3 Características de los lenguajes de marcas

Los lenguajes de marcas son una manera de codificar un documento de texto de manera que por medio de las marcas (el equivalente de los metadatos de los archivos binarios) se incorpora información relativa a cómo se debe representar el texto, sobre qué estructura tienen los datos que contiene, etc.

Los lenguajes de marcas han destacado por una serie de características que los han convertido en los tipos de lenguajes más usados en la informática actual para almacenar y representar los datos. Entre las características más interesantes que ofrecen los lenguajes de marcas se encuentran:

- Que se basan en el texto plano.
- Que permiten utilizar metadatos.
- Que son fáciles de interpretar y procesar.
- Que son fáciles de crear y suficientemente flexibles para representar datos muy diversos.

Las aplicaciones de Internet y muchos de los programas de ordenador que se utilizan habitualmente utilizan de alguna manera u otra algún lenguaje de marcas.

Basados en texto plano

Los lenguajes de marcas se basan en texto plano sin formato. Estos caracteres pueden estar codificados en diferentes códigos de caracteres: ASCII, ISO-8859-1, UTF-8, etc.

Una de las ventajas que intentan aportar los lenguajes de marcas es que se pueden interpretar directamente y esto sólo es posible si usamos el formato de texto, ya que los binarios requieren un programa para interpretarlos. Pero además tienen la ventaja de que son independientes de la plataforma, del sistema operativo o del programa.

El hecho de que estén basados en formato de texto hace que sean fáciles de crear y modificar para que sólo requieren un simple editor de textos.

Uso de metadatos

Las marcas se intercalan entre el contenido del documento, por lo que generalmente estas etiquetas suelen ser descriptivos de qué es lo que indica el contenido de los datos que contienen.

Estas marcas son la forma en que se añaden los metadatos a los documentos de texto y cómo se consiguen superar las limitaciones del formato de texto y conseguir algunas de las ventajas de los ficheros binarios.

Facilidad de proceso

Los lenguajes de marcas permiten que el procesamiento de los datos que contengan pueda ser automatizado de alguna manera, ya que el archivo contiene la estructura de los datos que contiene.

El hecho de incluir la estructura permitirá que un programa pueda interpretar cada uno de los datos de un fichero de marcas para representarlo o tratarlo convenientemente, ya que muestran la estructura de los datos que contienen.

Posteriormente un programa podrá interpretar gracias a las marcas que es lo que significa cada uno de los datos del documento.

Facilidad de creación y representación de datos diversos

A pesar de que fueron pensados para contener datos de texto, los lenguajes de marcas han demostrado que son capaces de contener datos de muchos tipos diferentes.

Actualmente se están utilizando archivos de marcas para representar imágenes vectoriales, fórmulas matemáticas, crear páginas web, ejecutar funciones remotas mediante servicios web, representar música o sonidos, etc.

Y sin importar qué tipo de datos se representen siempre habrá la posibilidad de crear estos archivos desde un editor de texto básico.

1.4 Ejemplos de lenguajes de marcas

- Darwin Information Typing Architecture (DITA)
- DocBook
- Extensible HyperText Markup Language (XHTML)
- Extensible Markup Language (XML)
- Standard Generalized Markup Language (SGML)
- HyperText Markup Language (HTML)
- Lilypond (sistema para notación musical)
- Maker Interchange Format (MIF)
- Mathematics Markup Language (MathML)
- Microsoft Assistance Markup Language (MAML)
- Music Extensible Markup Language (MusicXML)
- Rich Text Format (RTF)
- S1000D (Especificación internacional para documentación técnica relacionada al área comercial y

militar).

- TeX, LaTeX (utilizado generalmente en matemáticas y publicaciones académicas).
- Text Encoding Initiative (TEI). (formato XML para publicaciones digitales)
- Wireless Markup Language (WML), Wireless TV Markup Language (WTVML)
- XHTML Basic (subconjunto de XHTML para dispositivos portátiles, para reemplazar a WML, XHTML MP y C-HTML).

Veamos alguno de ellos con más detalle:

SGML (*Standard Generalized Markup Language***)**

Es un estándar internacional publicado por la ISO (Organización Internacional de Estándares). SGML estableció dos reglas principales:

- La sintaxis que debía utilizarse para diseñar un conjunto de marcas aplicables a cada tipo de documento.
- La forma en la que se deben intercalar marcas en el texto de un documento para identificar sus partes estructurales.

El conjunto de marcas que se pueden utilizar con cada tipo de documento constituye una DTD o definición de tipo de documento. El concepto de DTD se ha reutilizado también en le lenguaje XML.

HTML (*Hyper Text Markup Language*)

Es una aplicación del lenguaje SGML que especifica cómo se deben codificar los documentos para distribuirlos en la Web. Su origen se remonta a comienzo de los años noventa, cuando Tim Berners Lee, del CERN, desarrollaron el World Wide Web.

HTML era independiente de plataformas hardware o software específicas, lo que le convertía en la solución idónea para los problemas de intercambio de documentación en formato electrónico.

HTML presenta algunas limitaciones:

- Incapacidad para presentar las características tipográficas y presentaciones complejas de los documentos.
- La falta de capacidad expresiva del lenguaje, debido a que sólo se puede utilizar un número limitado de marcas predefinidas en la especificación.

XML (*eXtensible Markup Language*)

Comenzó a desarrollarse en 1996 por el W3C (el comité encargado de normalizar y controlar el desarrollo de los estándares para la Web) con un claro propósito: diseñar un lenguaje de marcas optimizado para poder ser utilizado en Internet. Debía combinar la simplicidad de HTML, con la capacidad expresiva de SGML. XML es un lenguaje para representar datos e información. XML se dice extensible porque podemos crear nuestras propias etiquetas, en lugar de estar sujetos a un conjunto de ellas como ocurre en HTML.

XHTML (*eXtensible Hyper Text Markup Language*)

Se presenta como una redefinición del lenguaje HTML utilizando la sintaxis de XML. El W3C lo publicó en el año 1999 y es la última versión del lenguaje HTML, tras la versión 4.0.

Se han incluido todas las etiquetas HTML pero siguiendo las directrices de XML. Es decir que , entre otras cosas, cada etiqueta que se abra debe cerrarse con un orden.

2 XML

2.1 Un pequeño ejemplo

Antes de nada veamos un pequeño ejemplo de un documento XML. En primer lugar vamos a construir un archivo llamado filmoteca.css que contendrá la definición de estilos a aplicar en el documento. El archivo debe estar en la misma ubicación que va a estar el archivo xml, casablanca.xml.

Los ejemplos aparecen como adjuntos en la presente unidad.

Pero, como ocurre con HTML, no existe consenso universal sobre cómo interpretar archivos CSS, cada cliente Web lo hace a su manera. Abre el archivo con Firefox e Internet Explorer y comprueba la diferencia.

2.2 Características

XML ha pasado de se considerado una alternativa a HTML, a convertirse en el lenguaje con mayor impacto en el desarrollo de aplicaciones informáticas para Internet e Intranet.

Las principales características que ofrece XML son:

◆**Conjunto de marcas abiertas y ampliables**: podemos definir nuevas marcas para codificar la estructura y contenido de distintos tipos de documentos.

◆**Distinción entre la estructura y la presentación de los documentos:** las marcas de un documento XML no indican nada sobre cómo debe presentarse el documento. Para indicar cómo se debe presentar un documento en pantalla o en papel,será necesario crear una hoja se estilo aparte y asociarla al documento. Para crear hojas de estilo para documentos XML disponemos de dos alternativas; las hojas de estilo CSS, ya utilizadas con páginas HTML, y las XSL, diseñadas específicamente para XML.

◆**Gestión de hipervínculos avanzada**: los hipervínculos XML pueden crear una relación entre más de dos documentos.

◆**Modularidad**: decimos que un documento es modular cuando está formado por distintos archivos XML que se presentan como si se tratara de un único documento.

2.3 Estructura

Partes de un documento XML

Un documento XML puede presentar tres partes diferentes: el prólogo, el cuerpo y el epílogo.

PRÓLOGO

Es una parte opcional.

La primera línea se encarga de presentar el tipo de documento, la versión de la norma a la que se adhiere, la codificación de carácter (US-ASCII, UTF-8, UTF-7, UCS-2, EUCJP, Big5, ISO-8859-1, ISO-8859-7, etc. y otras características.

```
<?xml version=”1.0” encoding=”ISO-8859-1”?>
```

La etiqueta anterior está compuesta de forma diferente al resto de las etiquetas que aparecen en el documento. A los símbolos “<” y “>” se les añade un símbolo de cierre de interrogación. Estas instrucciones se conocen como i*nstrucciones de proceso*.

El conjunto de caracteres ISO-8859-1 referenciado en esta declaración incluye todos los caracteres usados en la mayoría de los lenguajes de Europa Occidental. Si no se especifica encoding el parser XML asume que los caracteres pertenecen al conjunto UTF-8, un estándar Unicode que soporta virtualmente cada carácter e ideograma de cualquier lenguaje del mundo.

La segunda línea define el tipo de documento, especificando que DTD valida y define los datos que contiene.

Ejemplos de prólogos

```
<?xml version="1.0" encoding="UTF-7"?>
<!DOCTYPE mensaje SYSTEM “mensaje.dtd”>

<?xml version="1.0" encoding=“ISO-8859-1"?>
<!DOCTYPE HTML PUBLIC “-/ /W3C/ /DTD HTML 3.2 Final/ /EN”>

<?xml version="1.0" encoding=“Big5"?>
Otra instrucción de proceso sería:
          <?xml-stylesheet type="text/css" href="filmoteca.css"?>
```

CUERPO

Los datos que un documento XML nos ofrece están en lo que se concoce como cuerpo. Es un árbol único de elementos marcados, con anidamiento estricto.

En nuestro ejemplo, es todo lo que va entre <pelicula> y </pelicula>, este elemento se conoce como la raíz del elemento.

EPÍLOGO

Situado a continuación del cuerpo, puede estar compuesto de instrucciones de proceso como las del prólogo, a excepción de declaraciones xml o de tipo de documento.

2.4 Etiquetas

Los elementos XML pueden tener contenido (más elementos, caracteres, o ambos a la vez), o bien ser elementos vacíos.

Un elemento con contenido es, por ejemplo:

```
<nombre>Fulano Mengánez</nombre>
```

Para diferenciar entre los diferentes elementos que componen un documento XML hemos de utilizar etiquetas. Las etiquetas están compuestas por un nombre y unos atributos, debe existir una de apertura y otra de cierre y deben de estar correctamente situadas.

Todas las etiquetas que aparecen en nuestros documentos XML deben seguir unas normas muy sencillas:

- Deben de estar delimitadas por los símbolos < y >.
- El nombre de la etiqueta debe comenzar por cualquier letra, un guión bajo (_) o dos puntos (:). A partir de ese momento, podemos utilizar también el guión (-)
- Las mayúsculas y las minúsculas son diferentes, algo que no es cierto en HTML, donde <BR> y <br> son la misma etiqueta. En un documento XML <cliente> y <Cliente> son etiquetas diferentes.
- Cada etiqueta de apertura debe tener una de cierre.

Las etiquetas también pueden estar vacías. Un elemento vacío, es el que no tiene contenido. Se pueden escribir de dos maneras diferentes:

```
<actor></actor>
```

o de forma abreviada:

```
<actor/>
```

La sintaxis de HTML permite etiquetas vacías tipo <hr>. En HTML reformulado para que sea un documento XML bien formado, se debería usar <hr/>.

2.5 Comentarios

Un documento XML puede contener anotaciones en forma de comentario. Los comentarios no son parte del contenido de información del documento, y pueden ser ignorados por los procesadores XML. Los comentarios se escriben como

<!-- ...*texto del comentario*... -->

El texto de un comentario no puede contener la secuencia --.

2.6 Atributos

Las etiquetas pueden aprovecharse para incluir otros datos, utilizando atributos. Si has visto HTML ya conocerás algo de los atributos. Por ejemplo, la etiqueta HTML para crear un enlace a otra página podría ser el siguiente:

```
<a href=http://www.yahoo.es>Enlace a yahoo</a>
```

En este caso, la etiqueta a se refiere a un enlace y href es un atributo.

En XML no es diferente a HTML, la etiqueta denota el nombre del elemento, y el atributo sus propiedades. La diferencia es que XML es más estricto que HTML.

```
<gato raza="Persa">Micifú</gato>
```

Al igual que en otras cadenas literales de XML, los atributos pueden estar marcados entre comillas verticales (') o dobles ("). Cuando se usa uno para delimitar el valor del atributo, el otro tipo se puede usar dentro.

```
<verdura clase="zanahoria" longitud='15" y media'>

<cita texto="'Hola buenos dias', dijo él">
```

Los atributos se usan para diferenciar entre elementos del mismo tipo. Por ejemplo:

```
<gato raza="Persa">Micifú</gato>

<gato raza="Siames">Milu</gato>
```

2.7 Anidamiento de elementos

El contenido de los elementos no está limitado a sólo texto; los elementos pueden contener otros elementos, que a su vez pueden contener texto u otros elementos, y así sucesivamente.

Un documento XML es un árbol de elementos. No hay límite para la profundidad del árbol, además de que los elementos pueden repetirse.

Al elemento que está dentro de otro se le conoce como hijo. El elemento en el que éste está contenido se conoce como padre.

```
<nombre>

    <nombre-pila>Juan</nombre-pila>

    <apellido>Pérez</apellido>

</nombre>
```

El documento debe tener un solo elemento raíz. Dicho de otro modo, todos los elementos del documento deben ser hijos de un solo elemento.

2.8 Entidades predefinidas

En XML 1.0, se definen cinco entidades para representar caracteres especiales y que no se interpreten como marcado por el procesador XML. Es decir, que así podemos usar el carácter "<" sin que se interprete como el comienzo de una etiqueta XML, por ejemplo.

Las entidades son:

Entidad	Caracter
&amp;	&
&lt;	<
&gt;	>
&apos;	'
&quot;	"

2.9 Secciones CDATA

Existe otra construcción en XML que permite especificar datos, utilizando cualquier carácter, especial o no, sin que se interprete como marcado XML. Las secciones CDATA se utilizan para indicarle al analizador que un determinado fragmento del documento XML no debe ser analizado. Existen caracteres que no pueden ser incluidos directamente como contenido, como son (<) (>) o (&), para los que tenemos que usar expresiones equivalentes.

En ocasiones, utilizar estos equivalentes no resulta práctico ni deseable. El caso más frecuente que se suele presentar es aquel en el que queremos incluir código fuente en algún lenguaje de programación, como por ejemplo JavaScript, o incluso utilizar HTML.

Un elemento CDATA debe comenzar con <![CDATA[y terminar con]]>

```
<ejemplo>
<![CDATA[
<HTML>
<HEAD><TITLE>Rock & Roll</TITLE></HEAD>
]]>
</ejemplo>
```

2.10 Elaboración de documentos bien formados

Se dice que un documento XML está bien formado cuando cumple las reglas sintácticas indicadas. Los procesadores XML pueden rechazar cualquier documento que no esté bien formado.

2.11 Utilización de espacios de nombres en XML

El poder de XML proviene de su flexibilidad, del hecho de que podamos definir nuestras propias etiquetas para describir nuestros datos. Veamos un ejemplo:

```
<direccion>
      <nombre>
          <titulo>Sra.</titulo>
          <nombre-pila>Maria</nombre-pila>
          <apellidos>Martin López</apellidos>
      </nombre>
   <calle>Mayor 14</calle>
```

```
   <ciudad>Madrid</ciudad>
      <codigo-postal>34829</codigo-postal>
</direccion>
```

El documento incluye el elemento <titulo> para el título o tratamiento de cortesía, una elección perfectamente razonable como nombre de un elemento. Si creamos una librería online, podría elegir el crear un elemento <titulo> para almacenar el título de un libro. Todas son elecciones razonables, pero todas ellas crean elementos con el mismo nombre. ¿Cómo saber si, dado un elemento <titulo> se refiere a una persona o a un libro ? Con los *namespace*s.

Para usar un namespace, definimos un prefijo *namespace* y lo mapeamos a una cadena particular.

Así es cómo se deberían definir prefijos namespace para nuestros dos elementos <titulo>:

```
<?xml version="1.0"?>
<resumen_usuario
  xmlns:direccion="http://www.xyz.com/direcciones/"
  xmlns:libros="http://www.zyx.com/libros/"
>
... <direccion:nombre><titulo>Mrs.</titulo> ... </direccion:nombre> ...
... <libros:titulo>El Señor de los Anillos</libros:titulo> ...
```

En este ejemplo, los dos prefijos del namespace son direccion y libros.

Hay que darse cuenta que la definición de un namespace para un elemento particular significa que todos los elementos hijos pertenecen al mismo namespace. El primer elemento <titulo> pertenece al namespace direccion debido a que su elemento padre <direccion:Nombre>, ya pertenecía a ese namespace.

El punto final: La cadena de una definición de namespace es solo una cadena. Si, esa cadena parece una URL, pero no lo es. Podríamos definir xmlns:direccion="pepe" y funcionaría exactamente igual, Lo único que importa acerca de la cadena del namespace es que sea única; por esto la mayor parte de las definiciones de namespace parecen URLs. Los parser XML no van a la dirección http://www.zyx.com/libros/ para buscar una DTD o un Esquema, simplemente usan esos textos como cadenas.

2.12 Herramientas software

Disponemos de las siguientes herramientas para comprobar que un documento este bien formado:

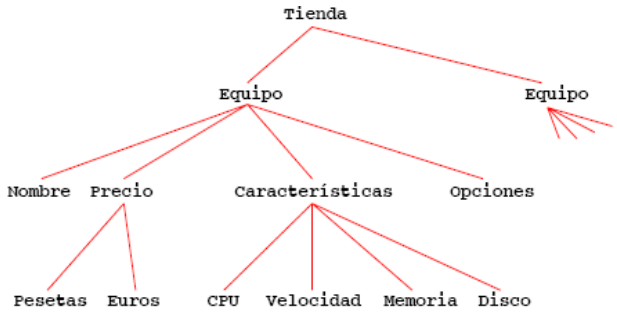
- Un cliente Web
- Un editor de XML
- Un validador online: www.w3c.org

EJERCICIO

Crea un documento XML denominado Equipos.xml que contenga la información sobre dos equipos de una tienda de informática; sobre cada producto se deben proporcionar los siguientes datos: nombre, precio, características y opciones. El precio se debe proporcionar tanto en pesetas como en euros.

Las características que se han de indicar son: CPU, velocidad, memoria y espacio de disco.

Una vez creado el documento XML visualizarlo en IE y Firefox sin hoja de estilo. Validarlo en www.w3c.org


Tienda
Equipo
Equipo
Nombre
Precio
Características
Opciones
Pesetas
Euros
CPU
Velocidad
Memoria
Disco