# Tema 1. Características de los lenguajes de marcas

# 1.- Introducción

Los lenguajes de marcas están en plena expansión. Apoyándose en el gran éxito que han representado las páginas web han ido apareciendo nuevas tecnologías basadas en lenguajes de marcas que permiten a los usuarios y a los programas conseguir resultados con los cuales antes no se podía soñar.

Los nuevos lenguajes de marcas intentan ir un poco más allá de la simple representación de datos que ofrecía HTML, y que eran ideales para las personas, a nuevos sistemas que puedan hacer que los programas puedan recuperar la información que se encuentra en los ordenadores y procesarla de manera automática.

En esta unidad veremos las características y la evolución de los lenguajes de marcas.

# 2. -Qué son los lenguajes de marcas


Para entender por qué surgieron los lenguajes de marcas, debemos comenzar entendiendo cómo el ordenador almacena y codifica la información.

Una definición poco estricta de lo que es un ordenador podría ser que "es una máquina electrónica que recibe y procesa datos para convertirlas en información útil".

**Uno de los componentes básicos en un sistema informático son los datos, que se pueden introducir y que hace el sistema para almacenarlos, utilizarlos posteriormente o mostrarlos de nuevo.**

Por lo tanto, una de las tareas básicas que hacen los ordenadores es almacenar la información que le proporcionamos para poder ser procesada posteriormente. Esta información puede ser de muchos tipos diferentes (texto, imágenes, vídeos, música...) pero lo realmente importante será de qué manera lo almacena el ordenador para poderla tratar posteriormente de manera eficiente para generar más información.

# 3.-Formas de representar la información en el ordenador

El ordenador es una ***máquina digital,*** por lo tanto sólo es capaz de representar información utilizando el sistema binario de numeración (0 y 1). Esto obliga a que, para poder almacenar información en un ordenador, previamente haya que codificarla en forma de números binarios.

El problema de los números binarios es que están muy alejados del ser humano; es decir, que las personas no estamos capacitadas para manejar información en binario. Nosotros usamos el sistema decimal para los números y formas de representación mucho más complejas para otro tipo de información (como el texto, las imágenes, la música, etc.).

Sin embargo, actualmente un ordenador es capaz de manejar información de todo tipo: música, imágenes, texto, etc. Esto es posible porque se ha conseguido que casi cualquier tipo de información sea codificable en binario.

Los seres humanos tenemos la capacidad de diferenciar claramente lo que es un texto de una imagen, lo que es un número de una canción, etc. Pero en un ordenador todo es más complicado, porque todo es binario.

Desde los inicios de la informática, la codificación (el paso de información humana a información digital) ha sido problemática debido a la falta de acuerdo en la representación. Pero hoy día ya tenemos numerosos estándares.

Fundamentalmente la información que un ordenador maneja son números y texto. Pero curiosamente a nivel formal se consideran datos binarios a cualquier tipo de información representable en el ordenador, que no es texto (imagen, sonido, vídeo, etc.), aunque como ya hemos comentado, en realidad toda la información que maneja un ordenador es binaria, incluido el texto.

Por tanto, podemos **representar la información** de dos maneras diferentes:

## 3.1. Datos binarios

Cualquier dato que no sea texto, se considera dato binario. Por ejemplo: música, vídeo, imagen, un archivo Excel, un programa, etc.

La forma de codificar ese tipo de datos a su forma binaria es muy variable. Por ejemplo en el caso de las imágenes, cada punto (píxel) de la imagen se codifica utilizando su nivel de rojo, verde y azul. De modo que una sola imagen produce millones de dígitos binarios.

En cualquier caso sea cual sea la información que estamos codificando en binario, para poder acceder a dicha información, el ordenador necesita el software que sepa como decodificar la misma, es decir saber qué significa cada dígito binario para traducirle a una forma más humana. Eso sólo es posible utilizando el mismo software con el que se codificó o bien otro software pero que sea capaz de entender la información codificada.

## 3.2. Texto plano

El texto es quizá la forma más humana de representar información. Antes de la llegada del ordenador, la información se transmitía mediante documentos o libros en papel. Esa forma de transmitir es milenaria y sigue siendo la forma más habitual de transmitir información entre humanos; incluso con la tecnología actual aplicaciones como Twitter, Whatsapp, etc; siguen usando el texto como formato fundamental para transmitir información.

# 4.-Formas de codificar el texto plano

Hemos comentado que los datos binarios están codificados siguiendo una serie de normas según el tipo de dato que almacene. Con el texto plano ocurre lo mismo, también existe el problema de cómo codificar ese texto en forma de dígitos binarios para hacerlo representable en el ordenador. La forma habitual ha sido codificar cada carácter en una serie de números binarios. De modo que, por ejemplo el carácter A fuera por ejemplo 01000001 y la B el 01000010.

El problema surgió por la falta de estandarización, la letra A se podía codificar distinto en diferentes ordenadores y así nos encontrábamos con un problema al querer pasar datos de un ordenador a otro. Poco a poco aparecieron estándares para intentar que todo el hardware y software codificara los caracteres igual. A ésto se le conoce como **Codificación de Caracteres**.

Podemos encontrar diferentes **Conjuntos de Caracteres** que realicen la Codificación de Caracteres. Los más utilizados en los lenguajes de marca son:

- ASCII.
- ISO 8859.
- UNICODE.

| Byte | Cod. | Char | Byte | Cod. | Char | Byte | Cod. | Char | Byte | Cod. | Char |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 00000000 | 0 | Null | 00100000 | 32 | Spc | 01000000 | 64 | @ | 01100000 | 96 | ` |
| 00000001 | 1 | Start of heading | 00100001 | 33 | ! | 01000001 | 65 | A | 01100001 | 97 | a |
| 00000010 | 2 | Start of text | 00100010 | 34 | " | 01000010 | 66 | B | 01100010 | 98 | b |
| 00000011 | 3 | End of text | 00100011 | 35 | # | 01000011 | 67 | C | 01100011 | 99 | c |
| 00000100 | 4 | End of transmit | 00100100 | 36 | $ | 01000100 | 68 | D | 01100100 | 100 | d |
| 00000101 | 5 | Enquiry | 00100101 | 37 | % | 01000101 | 69 | E | 01100101 | 101 | e |
| 00000110 | 6 | Acknowledge | 00100110 | 38 | & | 01000110 | 70 | F | 01100110 | 102 | f |
| 00000111 | 7 | Audible bell | 00100111 | 39 | ' | 01000111 | 71 | G | 01100111 | 103 | g |
| 00001000 | 8 | Backspace | 00101000 | 40 | ( | 01001000 | 72 | H | 01101000 | 104 | h |
| 00001001 | 9 | Horizontal tab | 00101001 | 41 | ) | 01001001 | 73 | I | 01101001 | 105 | i |
| 00001010 | 10 | Line feed | 00101010 | 42 | * | 01001010 | 74 | J | 01101010 | 106 | j |
| 00001011 | 11 | Vertical tab | 00101011 | 43 | + | 01001011 | 75 | K | 01101011 | 107 | k |
| 00001100 | 12 | Form Feed | 00101100 | 44 | , | 01001100 | 76 | L | 01101100 | 108 | l |
| 00001101 | 13 | Carriage return | 00101101 | 45 | - | 01001101 | 77 | M | 01101101 | 109 | m |
| 00001110 | 14 | Shift out | 00101110 | 46 | . | 01001110 | 78 | N | 01101110 | 110 | n |
| 00001111 | 15 | Shift in | 00101111 | 47 | / | 01001111 | 79 | O | 01101111 | 111 | o |
| 00010000 | 16 | Data link escape | 00110000 | 48 | 0 | 01010000 | 80 | P | 01110000 | 112 | p |
| 00010001 | 17 | Device control 1 | 00110001 | 49 | 1 | 01010001 | 81 | Q | 01110001 | 113 | q |
| 00010010 | 18 | Device control 2 | 00110010 | 50 | 2 | 01010010 | 82 | R | 01110010 | 114 | r |
| 00010011 | 19 | Device control 3 | 00110011 | 51 | 3 | 01010011 | 83 | S | 01110011 | 115 | s |
| 00010100 | 20 | Device control 4 | 00110100 | 52 | 4 | 01010100 | 84 | T | 01110100 | 116 | t |
| 00010101 | 21 | Neg. acknowledg | 00110101 | 53 | 5 | 01010101 | 85 | U | 01110101 | 117 | u |
| 00010110 | 22 | Synchronous idle | 00110110 | 54 | 6 | 01010110 | 86 | V | 01110110 | 118 | v |
| 00010111 | 23 | End trans. block | 00110111 | 55 | 7 | 01010111 | 87 | W | 01110111 | 119 | w |
| 00011000 | 24 | Cancel | 00111000 | 56 | 8 | 01011000 | 88 | X | 01111000 | 120 | x |
| 00011001 | 25 | End of medium | 00111001 | 57 | 9 | 01011001 | 89 | Y | 01111001 | 121 | y |
| 00011010 | 26 | Substitution | 00111010 | 58 | : | 01011010 | 90 | Z | 01111010 | 122 | z |
| 00011011 | 27 | Escape | 00111011 | 59 | ; | 01011011 | 91 | [ | 01111011 | 123 | { |
| 00011100 | 28 | File separator | 00111100 | 60 | < | 01011100 | 92 | \ | 01111100 | 124 | \| |
| 00011101 | 29 | Group separator | 00111101 | 61 | = | 01011101 | 93 | ] | 01111101 | 125 | } |
| 00011110 | 30 | Record Separator | 00111110 | 62 | > | 01011110 | 94 | ^ | 01111110 | 126 | ~ |
| 00011111 | 31 | Unit separator | 00111111 | 63 | ? | 01011111 | 95 | _ | 01111111 | 127 | Del |

# 5.-Ventajas y desventajas de utilizar archivos con datos binarios o texto plano

### Ventajas de los archivos binarios

✔ Ocupan menos espacio que los archivos de texto, ya que optimizan mejor su codificación a binario (por ejemplo el número 213 ocupa un solo byte y no tres como ocurriría si fuera un texto).

✔ Son más rápidos de manipular por parte del ordenador (se parecen más al lenguaje nativo del ordenador).

✔ Permiten el acceso directo a los datos. Los archivos de texto siempre se manejande forma secuencial, más lenta.

✔ En cierto modo permiten cifrar el contenido que de otra forma sería totalmente visible por cualquier aplicación capaz de entender textos (como el bloc de notas). Es decir los datos no son fácilmente entendibles.

**Ventajas de los archivos de texto**

✔ Son ideales para almacenar datos para exportar e importar información a cualquier dispositivo electrónico ya que cualquier es capaz de interpretar texto.

✔ Son directamente modificables, sin tener que acudir a software específico.

✔ Su manipulación es más sencilla que la de los archivos binarios.

✔ Son directamente transportables y entendibles por todo tipo de redes.

# 6.-El problema de compartir datos y los archivos texto plano como solución

L o s ***problemas relacionados con el intercambio de información entre aplicaciones y máquinas informáticas*** es tan viejo como la propia informática. El problema parte del hecho de haber realizado un determinado trabajo con un software en un ordenador concreto y después querer pasar dicho trabajo a otro software en ese u otro ordenador.

Los archivos binarios tienen la complicación de que para hacer ese proceso, el origen y el destino de los datos deben comprender cómo codificar y decodificar la información. En la informática actual eso es aún más problema al tener una necesidad de ***disponibilidad global del trabajo y además la posibilidad de ver dicho trabajo en diferentes dispositivos,*** como ordenadores, o teléfonos móviles por ejemplo.

Por ello poco a poco han aparecido ***formatos binarios de archivo que han sido estándares de facto*** (aunque no han sido reconocidos por ningún organismo de estándares) como por ejemplo el formato documental ***PDF***, el formato de imagen ***JPEG***, la música ***MP3*** o el formato ***MPEG*** de vídeo. Pero sigue habiendo empresas que utilizan formato propio por la idea de que sus formatos de archivo están directamente relacionados con la calidad de su software es decir razonan que el ***software que fabrican es muy potente y necesitan un formato binario propio compatible con esa potencia***. De ahí que muchas veces la opción para exportar e importar datos sea utilizar conversores, capaces de convertir los datos de un formato a otro (por ejemplo de Word a Open Office; de MP3 a MOV de Apple, etc.).

Sin embargo, ***hay un formato de archivo que cualquier dispositivo es capaz de entender: El texto.*** La cuestión es que para los archivos llamados de texto, sólo son capaces de almacenar texto plano; es decir sólo texto sin indicar ningún formato o añadir información no textual.

Debido a la facilidad de ser leído con cualquier aparato, se intenta que el ***propio texto sirva para almacenar otros datos***. Evidentemente no es posible usar texto para almacenar por ejemplo imágenes, pero sí otras cosas. Para ello dentro del archivo habrá contenido que no se interpretará como texto sin más que simplemente se debe mostrar, sino que hay texto en el archivo que se marca de manera especial haciendo que signifique otra cosa. Desde hace muchos años hay ***dos campos en los que esta idea ha funcionado bien: en las bases de datos y en los procesadores de texto.*** Actualmente el éxito de Internet ha permitido trasladar esta tecnología a otros campos.

Hay un ***problema*** con el texto, puesto que al ser formato tan universal, y ser su contenido siempre visible; es ***peligroso*** como fuente para almacenar datos confidenciales, ya que quedaría expuesto a cualquier persona. Los datos binarios no son del todo seguros, pero como requieren del software que entienda el formato binario concreto hacen que su contenido quede menos expuesto.

# 7.-Aparición de los lenguajes de marcas

Como se ha comentado en el apartado anterior, *el problema de la exportación de datos ha puesto en entredicho a los archivos binarios como fuente para exportar e importar información.* En su lugar parece que los archivos de texto poseen menos problemas (excepto el del cifrado de su información, que queda demasiado descubierta). Por ello se ha intentado que los archivos de texto plano (archivos que sólo contienen texto y no otros datos binarios) pudieran servir para almacenar otros datos como, por ejemplo, detalles sobre el formato del propio texto u otras indicaciones.

Los procesadores de texto fueron el primer software en encontrarse con este dilema. Puesto que son programas que sirven para escribir texto parecía que lo lógico era que sus datos se almacenaran como texto. Pero necesitan guardar datos referidos al formato del texto, tamaño de la página, márgenes, etc. La solución clásica ha sido guardar la información de formato de forma binaria, lo que provoca los ya comentados problemas.

***Algunos procesadores de texto optaron por guardar toda la información como texto, haciendo que las indicaciones de formato no se almacenen de forma binaria sino textual.*** Dichas indicaciones son caracteres marcados de manera especial para que así un programa adecuado pueda traducir dichos caracteres no como texto sino como operaciones que finalmente producirán mostrar el texto del documento de forma adecuada..

La idea del **marcado** procede del inglés **marking up** término con el que se referían a la técnica de marcar manuscritos con lápiz de color para hacer anotaciones como, por ejemplo, la tipografía a emplear en las imprentas. Este mismo término se ha utilizado para los documentos de texto que contienen comandos u anotaciones.

Las posibles anotaciones o indicaciones incluidos en los documentos de texto han dado lugar a lenguajes (entendiendo que en realidad ***son formatos de documento y no lenguajes en el sentido de los lenguajes de programación de aplicaciones***) llamados **lenguajes de marcas**, **lenguajes de marcado** o **lenguajes de etiquetas**.

# 8.- Definición y clasificación de lenguajes de marcas

Los **lenguajes de marcas** (también llamados **lenguajes de marcado**) son aquellos que ***combinan la información, generalmente textual, que contiene un documento con marcas o anotaciones relativas a la estructura del texto o a la forma de representarlo***. El lenguaje de marcas es el que especifica cuales serán las etiquetas posibles, donde deben colocarse y el significado que tendrá cada una de ellas. Así mismo, la presencia de etiquetas o marcas intercaladas en el contenido hace explicita la estructura del documento o cualquier información adicional que se quiera resaltar. Por otro lado, hay que tener en cuenta que las propias etiquetas o marcas generalmente no se suelen presentar al usuario final, ya que este suele estar interesado en el propio contenido del documento.

A continuación, se muestra un ejemplo en el que mediante una serie de marcas o etiquetas se ha representado una información relativa a una noticia:

**EJEMPLO 1.1**

```
<noticia>
<lugar>Madrid</lugar>
<fecha>27/08/2010</fecha>
<desc>Se ha inaugurado una estación de
tren</desc> </noticia>
```

**¿SABÍAS QUE…?**

Los lenguajes de marcas han de diferenciarse de los lenguajes de programación. El lenguaje de marcas no tiene funciones aritméticas o variables, como sí poseen los lenguajes de programación.

# 9.-Tipos de lenguajes de marcas

Los lenguajes de marcas se suelen dividir en tres grupos si bien hay que tener en cuenta que existen lenguajes que combinan características de más de un grupo:

- **Lenguajes orientados a presentación**. Este tipo de lenguajes son los usados tradicionalmente por los procesadores de texto como puede ser Microsoft Word y codifican como ha de presentarse el documento, por ejemplo, indicando que una determinada palabra debe presentarse en fuente itálica o que se debe dejar un espacio de 10 puntos al terminar el párrafo. Generalmente las marcas de los lenguajes orientados a presentación se ocultan al usuario lo que permite obtener un efecto WYSIWYG (*What You See Is What You Get.* Lo que ves es lo que obtienes). Este tipo de lenguajes de marcas no suelen ser flexibles ni reusables.

En ellos al texto común se añaden palabras encerradas en símbolos especiales que contienen indicaciones de formato que permiten a los traductores de este tipo de documentos generar un documento final en el que el texto aparece con el formato indicado. Es el caso de HTML en el que se indica cómo debe presentarse el texto (y no por ejemplo lo que significa el mismo) también se considera así los archivos generados por los procesadores de texto tradicionales en los que al texto del documento se le acompaña de indicaciones de formato (como negrita, cursiva, etc.).

**¿SABÍAS QUE…?**

En Word de Office o Writer de LibreOffice puedes ver las marcas pulsando el icono ¶ de la interfaz de Microsoft Word.

- **Lenguajes procedurales**. En este tipo de lenguajes las etiquetas son también orientadas a presentación pero se integran dentro de un marco procedural que permite definir macros (secuencias de acciones) y subrutinas. 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. Es el caso de LaTeX o PostScript donde por ejemplo se puede indicar una fórmula matemática.

**¿SABÍAS QUE…?**

La mayoría de los documentos científicos, artículos de investigación o libros técnicos que contienen fórmulas matemáticas se escriben con Latex.

**¿SABÍAS QUE…?**

PostScript es un lenguaje de descripción de páginas (en inglés PDL, *Page Description Language*), utilizado en muchas impresoras y, de manera usual, como formato de transporte de archivos gráficos en talleres de impresión profesional.

- **Lenguajes descriptivos**. Este tipo de lenguajes no definen qué se debe hacer con un trozo o sección del documento sino que por el contrario las marcas ***sirven para indicar qué es esa información***, ***esto es, describen que es lo que se esta representando***. La mayoría de los lenguajes de marcas que se usan hoy en día se encuentran dentro de este grupo como por ejemplo, el SGML y sus derivados (HTML, XML, etc.) que se verán a en el curso.

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 el 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).

**¿SABÍAS QUE…?**

El formato COLLADA está basado en XML y se utiliza para definir escenas de modelos tridimensionales, como el de los videojuegos.

# 10.- Evolución de los lenguajes de marcas

Los lenguajes de marcas comenzaron a usarse a finales de la década de los 60 para poder introducir anotaciones dentro de documentos electrónicos, de la misma forma que se hacia cuando la documentación estaba en papel. De esta posibilidad de incorporar marcas es de donde reciben su nombre. Es en esas fechas cuando se estandariza el lenguaje

**SGML** (*Standard Generalized Markup Language*), que es un descendiente directo del lenguaje **G M L** propuesto por IBM. Este lenguaje surgió para permitir compartir información por parte de sistemas informáticos. Este estándar tuvo una gran aceptación pero no consiguió asentarse del todo debido principalmente a su complejidad lo que provocaba que el software que usara SGML terminaba siendo excesivamente extenso y complejo.

A finales de los 80 dentro del CERN (*Conseil Européen pour la Recherche Nucléaire*) se creo un lenguaje de marcado pensado para compartir información usando las redes de computadores y, de forma más general, a través de Internet. Este lenguaje se basaba en algunos principios de de SGML y lo denominaron HTML (*Hyper-text Markup Language*). La aparición de este lenguaje supuso de alguna manera una revolución en la forma de compartir información, gracias principalmente a la sencillez de sus sintaxis y del software necesario para interpretarlo. En poco tiempo el lenguaje HTML se extendió y empezó a crecer de forma en ocasiones descontrolada y casi siempre influenciado por razones meramente comerciales.

A mediados de los años 90 el consorcio **W3C** (*World Wide Web Consortium) comenzó* una iniciativa para intentar dotar a la *web* de un lenguaje más potente y que pudiera dar una estructura semántica a la misma. Para ello se marcaron el objetivo de crear un nuevo lenguaje de marcas basado en SGML y que fuera sencillo como HTML. Finalmente, en el 1998, W3C hizo publico un nuevo estándar que denominaron XML (*eXtended Markup Language*), más sencillo que SGML y más potente que HTML.

**¿SABÍAS QUE…?**

HTML es el lenguaje de marcas predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

**P1.1. Evolución de los Lenguajes de Marcas**

# 11.-Cronología de los lenguajes de marcas

### Goldfarb

Se considera a Charles Goldfarb como al padre de los lenguajes de marcas. Se trata de un investigador de IBM que propuso ideas para que los documentos de texto tuvieran la posibilidad de indicar el formato del mismo. Al final ayudó a realizar el lenguaje GML de IBM el cual puso los cimientos del futuro SGML ideado por el propio Goldfarb.

### TeX y LaTeX

En la década de los 70, Donald Knuth (uno de los ingenieros informáticos más importantes de la historia, padre del análisis de algoritmos) creó para producir documentos científicos utilizando una tipografía y capacidades que fueran iguales en cualquier computadora, asegurando además una gran calidad en los resultados.

Para ello apoyó a TeX con tipografía especial (fuentes Modern Computer) y un lenguaje de definición de tipos (METAFONT). TeX ha tenido cierto éxito en la comunidad científica gracias a sus 300 comandos que permiten crear documentos con tipos de gran calidad, para ello se necesita un programa capaz de convertir el archivo TeX a un formato de impresión.

El éxito de TeX produjo numerosos derivados de los cuales el más popular es (LaTeX). Se trata de un lenguaje que intenta simplificar a TeX, fue definido en 1984 por Leslie Lamport, aunque después ha sido numerosas veces revisado. Al utilizar comandos de TeX y toda su estructura tipográfica, adquirió rápidamente notoriedad y sigue siendo utilizado para producir ***documentos con expresiones científicas, de gran calidad.***

***La idea es que los científicos se centren en el contenido y no en la presentación.***

Ejemplo de código LaTeX:

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\Ejemplo}
\begin{document}
Este es el texto ejemplo de \LaTeX{}
```

```
Con datos en \emph{cursiva} o \textbf{negrita}.
Ejemplo de f\'ormula
\begin{align}
E &= mc^2
\end{align}
\end{document}
```

Que con un traductor daría lugar al siguiente resultado:

Este es el texto ejemplo de LaTeX
Con datos en *cursiva* o **negrita**. Ejemplo de fórmula

$$E = mc^2 \tag{1}$$

**RTF**

RTF es el acrónimo de Rich Text Format (Formato de Texto Enriquecido) un lenguaje ideado por Microsoft en 1987 para producir documentos de texto que incluyan anotaciones de formato.

Actualmente se trata de un formato aceptado como texto con formato y en ambiente Windows es muy utilizado como formato de intercambio entre distintos procesadores por su potencia. El procesador de texto Word Pad incorporado por Windows lo utiliza como formato nativo. Ejemplo:

```
{\rtf1\ansi\ansicpg1252\deff0\deflang3082{\fonttbl{\f0\fnil\fcharset0 Calibri;}}
\viewkind4\uc1\pard\sa200\sl276\slmult1\lang10\f0\fs22 soy \i cursiva\i0\par}
```

Produce el resultado:

soy *cursiva*

**SGML**

Se trata de la versión de GML que estandarizaba el lenguaje de marcado y que fue ***definida finalmente por ISO como estándar mundial en documentos de texto con etiquetas de marcado.*** La estandarización la hace el subcomité SC24 que forma parte del comité JTC1 del organismo IEC de ISO que se encarga de los estándares electrónicos e informáticos (en definitiva se trata de una norma ISO/IEC JTC1/SC24, concretamente la 8879).

***Su importancia radica en que es el padre del lenguaje XML y la base sobre la que se sostiene el lenguaje HTML.***

En SGML las etiquetas que contienen indicaciones para el texto se colocan entre símbolos < y >. Las etiquetas se cierran con el signo /. Es decir las reglas fundamentales de los lenguajes de etiquetas actuales ya las había definido SGML.

En realidad (como XML) no es un lenguaje con unas etiquetas concretas, sino que se trata de un lenguaje que sirve para definir lenguajes de etiquetas; o más exactamente es un lenguaje de marcado que sirve para definir formatos de documentos de texto con marcas. ***Entre los formatos definidos mediante SGML, sin duda HTML es el más popular.***

**PostScript**

Se trata de un lenguaje de descripción de páginas. De hecho es el más popular. Permite crear documentos en los que se dan indicaciones potentísimas sobre como mostrar información en el dispositivo final. Se inició su desarrollo en 1976 por John Warnock y dos años más tarde se continuo con la empresa Xerox, hasta que en 1985 el propio Warnock funda Adobe Systems y desde esa empresa se continua su desarrollo.

Es en realidad todo un lenguaje de programación que indica la forma en que se debe mostrar la información que puede incluir texto y el tipo de letra del mismo, píxeles individuales y formas vectoriales (líneas, curvas). Sus posibilidades son muy amplias.

Ejemplo:

```
%!PS

/Courier          % Elige el tipo de letra
```

```
20 selectfont        % Establece el tamaño de la letra y
                     % la toma como el tipo de letra en uso
72 500 moveto        % Coloca el cursor en las coordenadas
                     %      72, 500 (contando los píxeles desde
                     %      la esquina izquierda de la página)
(Hola mundo!) show   % Escribe el texto entre paréntesis,
showpage             % Imprime el resultado
```

**HTML**

Tim Bernes Lee utilizó SGML para definir un nuevo lenguaje de etiquetas que llamó Hypertext Markup Language (lenguaje de marcado de hipertexto) para ***crear documentos transportables a través de Internet en los que fuera posible el hipertexto***; es decir, la posibilidad que determinadas palabras marcadas de forma especial permitieran abrir un documento relacionado con ellas.

A pesar de tardar en ser aceptado, ***HTML fue un éxito rotundo y la causa indudable del éxito de Internet.*** Hoy en día casi todo en Internet se ve a través de documentos HTML, que popularmente se denominan páginas web.

Inicialmente estos documentos se veían con ayuda de intérpretes de texto (como por ejemplo el Lynx de Unix) que simplemente coloreaban el texto y remarcaban el hipertexto. Después el software se mejoró y aparecieron navegadores con capacidad más gráfica para mostrar formatos más avanzados y visuales.

**XML**

Se trata de un subconjunto de SGML ideado para mejorar el propio SGML y con él definir lenguajes de marcado con sintaxis más estricta, pero más entendibles. Su popularidad le ha convertido en el lenguaje de marcado más importante de la actualidad y en el formato de documentos para exportación e importación más exitoso.

**JSON**

Abreviatura de ***JavaScript Object Notation***, Se trata de una notación de datos procedente del lenguaje JavaScript estándar (concretamente ECMA Script de 1999). En el año 2002 se le daba soporte desde muchos de los navegadores y su fama ha sido tal que ahora se ha convertido en una notación independiente de JavaScript que compite claramente con XML.

Se trata de una notación que realmente no se considera lenguaje de marcas, ya que no hay diferencia en el texto a través de etiquetas, sino que se ***basa en que el texto se divide en dato y metadato***. De modo que el símbolo de los dos puntos separa el metadato del dato. Por otro lado los símbolos de llave y corchete permiten agrupar de manera correcta los datos.

Ejemplo:

```
{
 "nombre": "Jorge",
 "apellido1": "Sánchez",
 "dirección":
 {
  "calle": "C/ Falsa nº 0",
  "localidad": "Palencia",
  "código Postal": "34001",
  "país":"España"
 },
 "teléfonos": [
  {
  "tipo": "fijo",
  "número": "999 999 999"
  },
  {
  "tipo": "móvil",
  "number": "666 666 666"
  }
 ]
}
```

# 12.- Etiquetas, elementos y atributos

Existen tres términos comúnmente usados para describir las partes de un documento de lenguajes de marcas: etiquetas, elementos y atributos.

- Una **etiqueta** (*tag*) es un texto que va entre el símbolo menor que (<) y el símbolo mayor que (>). Existen etiquetas de inicio (como <nombre>) y etiquetas de fin (como </nombre>).

  Los **elementos** representan estructuras mediante las que se organizara el contenido del documento o acciones que se desencadenan cuando el programa navegador interpreta el documento. Constan de la etiqueta de inicio, la etiqueta de fin y de todo aquello que se encuentra entre ambas. Algunos elementos no tienen contenido. Se les denomina elementos vacíos y no deben llevar etiqueta de fin.

- Un **atributo** es un par nombre-valor que se encuentra dentro de la etiqueta de inicio de un elemento e indican las propiedades que pueden llevar asociadas los elementos.

Fíjate en el texto siguiente:

```
<direccion>
    <nombre>
        <titulo>Mrs.</titulo>
        <nombre> Mary </nombre>
        <apellidos> McGoon </apellidos>
    </nombre>
    <calle> 1401 Main Street </calle>
    <ciudad estado="NC"> Anytown</ciudad>
    <codigo-postal> 34829 </codigo-postal>
</direccion>
```

En el ejemplo anterior, el elemento <*nombre*> contiene tres elementos hijos: <*titulo*>, <*nombre*> y <*apellidos*> y *estado* es un atributo del elemento <*ciudad*>.

## P1.2. Etiquetas, elementos y atributos

# 13.- Organizaciones desarrolladoras

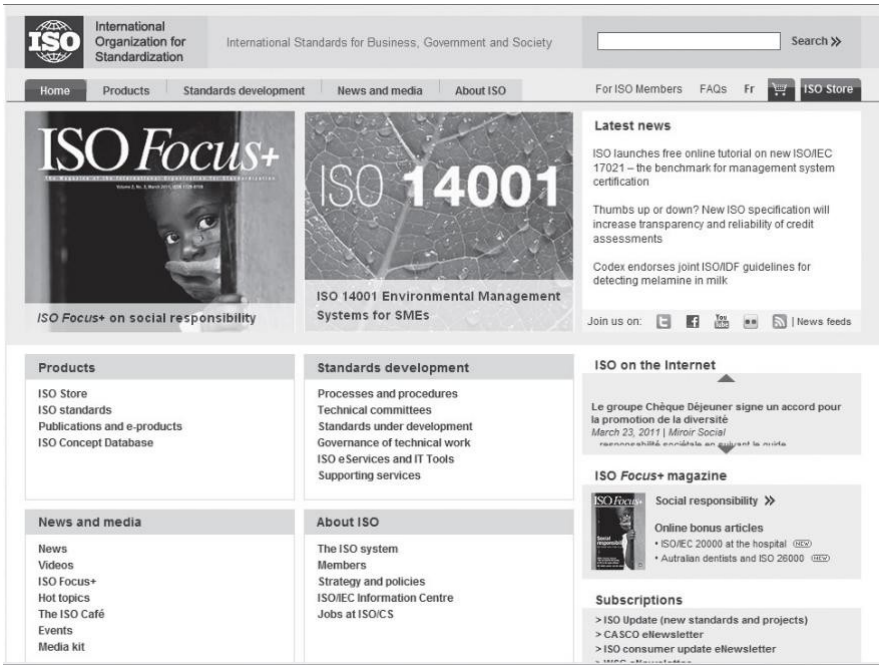
Dentro de las organizaciones que se han encargado de desarrollar los lenguajes de marcas se encuentran:

**Organización Internacional para la Estandarización** (**ISO**, *International Organization for Standardization*).

Se formo después de la Segunda Guerra Mundial (23 de febrero de 1947) y es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. S u función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.

Es una red de los institutos de normas nacionales de 163 países, sobre la base de un miembro por país, con una Secretaria Central en Ginebra (Suiza) que coordina el sistema.

Las normas desarrolladas por ISO son voluntarias, ya que es un organismo no gubernamental y no depende de ningún otro organismo internacional, por tanto, no tiene autoridad para imponer sus normas a ningún país. El contenido de los estándares esta protegido por derechos de copyright y para acceder a ellos el publico en general ha de comprar cada documento.




Esta organización después del éxito que tuvo **GML** y , después de un largo proceso, publicó en 1986 el *Standard Generalized Markup Languaje* (**SGML**) con rango de Estándar Internacional con el código **ISO 8879**.

***World Wide Web Consortium** (**W3C**). El W3C se creo en 1994 por Tim Berners-Lee en el MIT, actual sede central del consorcio. Posteriormente se unió, en abril de 1995, el INRIA en Francia, reemplazado por el ERCIM en el 2003 como el huésped europeo del consorcio y la Universidad de Keiō (Shonan Fujisawa Campus) en Japón en septiembre de 1996 como huésped asiático. **Su función principal es tutelar el crecimiento y organización de la web.***


Su primer trabajo fue normalizar el lenguaje HTML, el lenguaje de marcas con el que se escriben las paginas web. Al crecer el uso de la web, crecieron las presiones para ampliar el HTML. El W3C decidió que la solución no era ampliar el HTML, sino crear unas reglas para que cualquiera pudiera crear lenguajes de marcas adecuados a sus necesidades, pero manteniendo unas estructuras y sintaxis comunes que permitieran compatibilizarlos y tratarlos con las mismas herramientas. Ese conjunto de reglas es el XML, cuya primera versión se publico en 1998.

## P1.3. Organizaciones desarrolladoras

# 14.- Utilización de lenguajes de marcas en entornos web

Una **página web** es un documento electrónico adaptado para la *World Wide Web* que, normalmente, forma parte de un sitio web.

Esta compuesta, principalmente, por información (solo texto o módulos multimedia) así como por hiperenlaces; además, puede contener o asociar datos de estilo para especificar como debe visualizarse, y también aplicaciones embebidas para hacerla interactiva.

Las paginas web están escritas en un lenguaje de marcas que proporciona la capacidad de manejar e insertar hiperenlaces, generalmente, HTML.

El contenido de la pagina puede ser predeterminado (**página web estática**) o generado en el momento de su visualización o al solicitarla a un servidor web (**página web dinámica**).

Respecto a la estructura de las paginas web, algunos organismos, en especial el W3C, suelen establecer directivas con la intención de normalizar el diseño, para así facilitar y simplificar la visualización e interpretación del contenido.

# 15.- Gramáticas

Todo documento de un lenguaje de marcas tiene en común una gramática que define el marcado permitido en esa clase, el marcado requerido y como debe ser utilizado dicho marcado en la instancia del documento.

### DTD

E l estándar define esta gramática mediante la **D T D** (**Definición de Tipo de Documento**) que establece las reglas de formación del lenguaje formal, es decir, que combinaciones de símbolos elementales son sistemáticamente correctas.

En la DTD se identifica la estructura del documento, es decir, aquellos elementos que son necesarios en la elaboración de un documento o un grupo de documentos estructurados de manera similar. Contiene las reglas de dichos elementos: el nombre, su significado, donde pueden ser utilizados y que pueden contener.

La especificación del W3C para HTML 4.0 contempla tres DTD:

**DTD estricta** (*HTML 4.0 Strict DTD*): incluye todos los elementos y atributos que no han sido declarados "desaprobados" (*deprecated*), interpretando la expresión en el sentido de que no se recomienda ya su uso proponiéndose nuevos y mejores recursos para hacer lo mismo.

**DTD transicional o flexible** -*loose*- (*HTML 4.0 Transitional DTD*): incluye todo lo que la anterior más los elementos y atributos desaprobados (*deprecated*).

**DTD para documentos con marcos** (*HTML 4.0 Frameset DTD*): engloba todo lo incluido en la transicional más lo relativo a la creación de documentos con marcos (*frames*).

Recuerde que aunque la especificación recomienda ceñirse a los recursos de la DTD estricta, utilizar el resto de los elementos y atributos no es incorrecto.

La DTD es el formato de esquema nativo (y el más antiguo) para validar documentos XML, heredado de SGML. Utiliza una sintaxis no-XML para definir la estructura o modelo de contenido de un documento XML valido:

- Define todos los elementos.
- Define las relaciones entre los distintos elementos.
- Proporciona información adicional que puede ser incluida en el documento (atributos, entidades, notaciones).
- Aporta comentarios e instrucciones para su procesamiento y representación de los formatos de datos.

Es el método más sencillo usado para validar, y por esta razón presenta varias limitaciones, ya que no soporta nuevas ampliaciones de XML y no es capaz de describir ciertos aspectos formales de un documento a nivel expresivo.

Las DTD pueden ser internas o externas a un documento, o ambas cosas a la vez.

## 15.1.-Esquema XML

**XML Schema** es la evolución de la DTD descrita por el W3C, también denominado XSD (*XML Schema Definition*). **Es un lenguaje de esquema más complejo y más potente**, **basado en la gramática para proporcionar una potencia expresiva mayor que la DTD.** Utiliza sintaxis XML, cosa que le permite especificar de forma más detallada un extenso sistema de tipos de datos. A diferencia de las DTD, soporta la extensión del documento sin problemas.

A la hora de la validación del documento, la utilización de XSD supone un gran consumo en recursos y tiempo debido a su gran especificación y complejidad en la sintaxis (los esquemas son más difíciles de leer y de escribir).

Después de validar el documento con XML Schema, es posible expresar su estructura y contenido en términos del modelo de datos usado por el esquema de validación. Esta funcionalidad, conocida como *Post-Schema-Validation Infoset* (PSVI), se puede utilizar para transformar el documento en una jerarquía de objetos, a los cuales se puede acceder a través de un lenguaje de programación orientada a objetos (OOP).

El modelo de datos de XML Schema incluye:

- El vocabulario (nombres de elemento y atributo).
- El contenido modelo (relaciones y estructura).

  Los tipos de datos.