

# 1 Introducción a los Lenguajes de Marcas

## 1.1 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:

### **1.1.1 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.

### **1.1.2 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.

## **1.2 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 esto se le conoce como **Codificación de Caracteres**.

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

- ASCII
- ISO 8859
- UNICODE

### 1.3 **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 manejan de 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 cualquiera 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.

## **1.4 El problema de compartir datos y los archivos texto plano como solución**

Los 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. Eso, en muchos casos, ha sido un gran problema que ha obligado a que todos los trabajadores y trabajadoras hayan tenido que adaptarse al software de la empresa y por supuesto en toda la empresa utilizar dicho software. 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 dispositivos de todo tipo como mini ordenadores, PDA o incluso teléfonos móviles.

Por ello poco a poco han aparecido formatos binarios de archivo que han sido estándares (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 espolear 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.

## **1.5 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.**

Por tanto un lenguaje de marcado o **lenguaje de marcas** se puede definir como 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, su presentación, ...

### **1.6 Concepto y ventajas**

Las personas utilizamos en nuestro lenguaje hablado ciertos gestos, diferente entonación, pausas, etc. que acompañan al contenido que queremos transmitir. De modo parecido en el lenguaje escrito utilizamos tamaños de letra, negrita, numeración, viñetas, tablas, colores, etc. que también acompañan a la información para que sea más fácil de entender. En definitiva estamos aplicando un determinado formato a la información que transmitimos.

En el caso de los documentos que intercambiamos a través de Internet, como las páginas web, son los lenguajes de marcas quienes nos permiten aplicar dicho formato.

Un documento que contenga exclusivamente texto es perfectamente

legible por nosotros, aunque evidentemente, tedioso e inapropiado para publicar páginas web. Si le aplicamos formato mediante un lenguaje de marcas como por ejemplo HTML, obtenemos un archivo también legible, pero más difícil de interpretar. De esta tarea se encargará el navegador o mejor dicho el agente de usuario, quien interpreta las marcas de formato y las aplica convenientemente al texto para dar lugar a una página web, que será mucho más agradable de leer que el texto original.

Una "marca" es una señal colocada dentro de un texto, con el fin de delimitar una parte del mismo y en muchos casos, aplicarle un determinado formato (aunque existen marcas con otros propósitos).

Las marcas más comunes están formadas por una palabra que describe su función encerrada entre los símbolos menor que (<) y mayor que (>) como <html>.

Es muy habitual que aparezcan por parejas, una de comienzo y otra de fin.

Por ejemplo:

<h1> Hola, este texto aparece más grande </h1>

<h3> Este texto aparece más pequeño, adiós </h3>

Los lenguajes de marcas no son equivalentes a los lenguajes de programación, aunque se llamen también lenguajes. De hecho, no debemos utilizar la palabra "programar" cuando nos referimos a lenguajes de marcas, puesto que no disponen de los elementos típicos como variables, arrays, sentencias de control, funciones, etc.

Sin embargo, los lenguajes de marcas se pueden combinar dentro del mismo documento, con otros lenguajes como JavaScript o PHP, que sí son lenguajes de programación, con el objetivo de aportar funcionalidad y dinamismo a la página web.

Otro aspecto importante a tener en cuenta cuando hablamos de lenguajes de marcas es el destinatario de la información. Quizás lo más habitual, es un usuario final utilizando un navegador web en el pc de su casa, pero tenemos que considerar el resto de opciones que van en aumento, otros



## Introducción a los Lenguajes de Marcas

destinatarios podrían ser: usuarios en dispositivos móviles, usuarios con deficiencias visuales o motrices, usuarios de avanzada edad', un periférico como la impresora, los robots de los buscadores, etc. Por esta razón, es más correcto utilizar el término general, **agente de usuario** (user-agent) **en lugar de navegador**.

La presentación en la misma página web para cada uno de estos usuarios debe ser lógicamente muy distinta, así por ejemplo, un texto en negrita puede representarse respectivamente, por caracteres con mayor grosor, por un volumen más alto en el sintetizador de voz, por más puntos en el papel, etc.

La cuestión es que **el lenguaje de marcas debe ser independiente del destinatario final**, es el intérprete del lenguaje quien se encarga de representar las marcas de la forma adecuada. **HTML por ejemplo, no especifica en sus etiquetas cómo serán representadas más tarde por el navegador**. Esta es una de las razones por la que podemos encontrar ciertas diferencias en la visualización de una misma página, por parte de diferentes navegadores.

Por ejemplo, la siguiente tabla se visualiza diferente porque Firefox considera la barra / como salto de línea cuando no dispone de espacio para el texto y Explorer sin embargo, expande la columna. También podemos observar que Explorer asigna un grosor de letra algo mayor que Firefox.

Navegador	Texto	Salto de línea
M. Firefox	/MozillaFirefox /realizaunsaltodelíneaprevio	Espacio, Guión y Barra
I Explorer	/InternetExplorer /norealizasaltodelíneaconlabarra	Espacio y Guión

Vista con Firefox



Navegador	Texto	Salto de línea
M.Firefox	/MozillaFirefox/realizaunsaltodelíneaprevio	Espacio, Guión y Barra
I.Explorer	/InternetExplorer/norealizasaltodelíneaconlabarra	Espacio y Guión

Vista con Explorer

Por otro lado, para independizar aún más la representación de la página web de su contenido, se creó **CSS**, que **no es un lenguaje de marcas sino de estilos**. Mediante CSS podemos especificar con mayor precisión y eficacia la representación de la información, para cada intérprete y para diferentes soportes, como monitores, dispositivos móviles, papel, voz, etc.

Dado el auge de los dispositivos móviles, muchas páginas presentan diferentes versiones adaptadas al dispositivo que utilice el usuario, en este caso, se trata de documentos HTML diferentes o bien del mismo documento html, pero aplicándole una hoja de estilos distinta.

Por ejemplo, el sitio del Museo del Prado, cuando detecta como agente de usuario un dispositivo móvil, nos presenta una página diferente cuya dirección es <http://m.museodelprado.es>

## 1.7 SGML. El origen

En los años 60 las empresas de publicación y manejo de documentos electrónicos tenían el problema de **falta de compatibilidad entre aplicaciones**. El problema existente era que cada aplicación utilizaba sus propias marcas para describir los diferentes elementos, esto impedía el intercambio de documentos entre plataformas. Otra carencia importante era la separación entre estructura y aspecto del documento.

**IBM**, empresa pionera en investigación en informática y electrónica (más de 5.000 patentes en 100 años de historia) **intentó resolver estos problemas a través de un lenguaje de marcas denominado GML** (Generalized Markup Language).

**GML**, independiza el documento del dispositivo que lo va a utilizar, usando **marcas genéricas**. Por otro lado GML **incorpora marcas descriptivas** para la estructura del documento que permiten distinguir el texto, de las listas, las tablas, etc. El mismo documento puede, entonces, ser utilizado por varios dispositivos, simplemente especificando un perfil para cada uno.

En 1986 GML pasó a manos de **ISO** y se convirtió en **SGML** (ISO 8879), Standard Generalized Markup Language, software libre y de código abierto.

Es importante tener en cuenta que SGML no es estrictamente un lenguaje sino un metalenguaje, es decir, **un conjunto de normas que permiten crear otros lenguajes de marcas**. Esto se hace definiendo un vocabulario o conjunto de elementos a utilizar, y una gramática o conjunto de reglas que rigen el uso de los elementos y sus atributos.

**SGML**, por tanto, es un metalenguaje que **permite definir lenguajes de marcado**. **HTML** por ejemplo **es uno de los lenguajes** creados a partir de SGML.

**Ventajas de SGML:** **Reutilización de los datos**, **integridad** y **control** sobre los datos, portabilidad, **adaptabilidad**.

**Inconvenientes de SGML:** Alta **complejidad**

Un documento SGML **consta de 2 partes:**

- **prólogo:** **contiene la estructura**.
  - La **declaración**: indica que el documento es SGML y algunos parámetros.
  - La **definición de tipo de documento (DTD)**: indica la sintaxis particular del lenguaje creado.
- **La instancia de documento:** **contiene los datos**.

En los siguientes ejemplos se ha omitido la declaración SGML, la DTD sería el vocabulario y las reglas de uso, y la instancia de documento serían los datos.

**Ejemplo 1:** Los módulos del Ciclo ASIR

*Vocabulario:* asir, módulo, título, contenido, unidad.

*Reglas:* asir contiene varios módulos, un módulo tiene un elemento simple título y un elemento contenido, contenido tiene varias unidades y toda unidad debe estar en un contenido, las unidades son texto simple, detrás de cada unidad solo puede ir otra unidad o fin de contenido, detrás de un módulo solo puede ir otro módulo o fin de asir.

```
<asir>
  <módulo><título>Lenguajes de Marcas</título></módulo>
    <contenido>
      <unidad>Introducción</unidad>
      <unidad>HTML</unidad>
      <unidad>CSS</unidad>
      <unidad>XML</unidad>
      <unidad>XSLT</unidad>
      <unidad>Sindicación</unidad>
      <unidad>ERP</unidad>
    </contenido>
  </módulo>
</asir>
```

### **Ejemplo 2:** Mini HTML

*Vocabulario:* html, head, title, body, p.

*Reglas:* html contiene un elemento head y un elemento body, head contiene un elemento simple title, body contiene varios elementos p y todo elemento p debe estar en body, p es texto simple, detrás de p solo puede ir otro p o fin de body, detrás de body solo puede ir fin de html.

```
<html>
  <head><title>Mi página</title></head>
  <body>
    <p>Hola mundo</p>
    <p>Esta es mi página</p>
  </body>
</html>
```

## **1.8 Características de los lenguajes de marcas**

### **TEXTO PLANO**

Los archivos de texto plano son aquellos que están compuestos únicamente por caracteres de texto, a diferencia de los archivos binarios que pueden contener imágenes, sonido, archivos comprimidos, programas compilados, etc.

Estos caracteres se pueden codificar con distintos códigos dependiendo

del idioma o alfabeto que se necesite, por ejemplo: ASCII, ISO-8859-15, UTF-8.

Una de las principales ventajas de los archivos de texto plano es que pueden ser interpretados directamente por un simple editor de texto, a diferencia de los binarios que necesitan software específico (visores multimedia, descompresores, compiladores, etc.)

Esta característica hace que los documentos sean independientes del sistema operativo o programa con el que fueron creados, esto facilita la interoperabilidad, que constituye una importante ventaja para el intercambio de información en Internet.

### **COMPACIDAD**

Las instrucciones de marcado se mezclan con el propio contenido, por ejemplo, `<h2>Contenido</h2>`.

El código entre corchetes como `<h2>`, son instrucciones de marcado, también llamadas etiquetas. Esta etiqueta en concreto es una etiqueta de presentación, indica que el texto comprendido debe tener el formato asignado a la cabecera nº 2.

El texto entre las marcas es el propio contenido del documento.

### **INDEPENDENCIA DEL DISPOSITIVO FINAL**

El mismo documento puede ser interpretado de diferentes formas dependiendo del dispositivo final, así tendremos diferentes resultados Si se usa un dispositivo móvil, un ordenador de sobremesa o una impresora.

### **ESPECIALIZACIÓN**

Inicialmente los lenguajes de marcas se idearon para visualizar documentos de texto, pero progresivamente se han empezado a utilizar en muchas otras áreas como gráficos vectoriales, sindicación de contenidos, notación científica, interfaces de usuario, síntesis de voz, etc.

### **FLEXIBILIDAD**

Los lenguajes de marcas se pueden combinar en el mismo archivo con otros lenguajes, como HTML con PHP y JavaScript. Incluso hay etiquetas específicas para ello como es `<script>`.

XML ha permitido que se puedan combinar varios lenguajes de marcas diferentes en un mismo archivo, como en el caso de XHTML con MathML y SVG.

## 1.9 Clasificación de los lenguajes de marcas

Normalmente los lenguajes de marcas se suelen clasificar en tres tipos, atendiendo al tipo de marcas que utilizan:

- **De presentación:**

Indican el formato del texto o tipografía, sin especificar su estructura, por ejemplo aumentar el tamaño de la fuente, centrar o cambiar a negrita.

Esta categoría incluye los lenguajes de procedimiento que agrupan varias marcas de presentación en una macro. Por ejemplo, para formatear un título, debe haber una serie de directivas inmediatamente antes del texto indicando: tamaño de letra 16p, fuente Arial, negrita. Justo después del título debe haber etiquetas inversas que anulen el formato, para continuar con el texto normal.

El software que representa el documento debe interpretar el código en el mismo orden en que aparece.

Los procesadores de texto y en general las aplicaciones de edición profesional utilizan este tipo de marcado.

Ejemplos:

- nroff, troff, RTF
- TeX, Docbook (derivados de SGML)

- **Descriptivo, estructural o semántico:**

Indican las diferentes partes en las que se estructura el documento, pero sin especificar cómo deben representarse ni en qué orden.

XML es un metalenguaje expresamente diseñado para generar marcado descriptivo y los lenguajes derivados de XML con este

propósito son: EBML, ROF, XFML, OWL Y XTM. Aunque XML almacena información de todo tipo, los demás tienen contenido específico.

Estos lenguajes crean documentos con estructura en árbol que almacenan información, por eso son bases de datos, sin embargo no usan tablas ni respetan las reglas de integridad propias de las BD Relacionales, por ello se les llama bases de datos semiestructuradas.

Ejemplos:

- ASN.1, YAML.
- EBML, ROF, XFML, OWL, XTM (derivados de XML)

- **Híbridos**

Lenguajes que contienen marcas de los dos tipos anteriores indistintamente.

Ejemplos:

- HTML (derivado de SGML)
- XHTML, WML (derivados de XML)

Una clasificación de lenguajes de marcas atendiendo a su funcionalidad:

- **Para crear documentación electrónica:**

- RTF
- TeX
- Wikitexto
- DocBook

- **Tecnologías de Internet:**

- HTML, XHTML
- RDF (recurso-propiedad(relación)-valor)
- RSS

- **De propósito específico:**

- MathML, CML (fórmulas matemáticas)

- VoiceXML (síntesis de voz)
- SVG (gráficos vectoriales)
- MusicXML (partituras de música)

## 1.10 Organización y estándares

Podemos definir la normalización o estandarización como el proceso de especificación de normas, para garantizar el correcto funcionamiento de elementos contruidos de forma independiente.

Aplicado al contexto de los lenguajes de marcas, sería por ejemplo el desarrollo de páginas web atendiendo a las especificaciones oficiales del lenguaje utilizado.

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."

El W3C recibe ingresos de las cuotas de sus miembros, de becas de investigación, subvenciones y donaciones privadas. Por tanto, no se trata de una empresa con fines lucrativos sino de una comunidad heterogénea formada por diferentes organismos que son miembros, un grupo de documentación técnica y los grupos de trabajo formados por expertos, que son quienes fabrican principalmente los estándares.

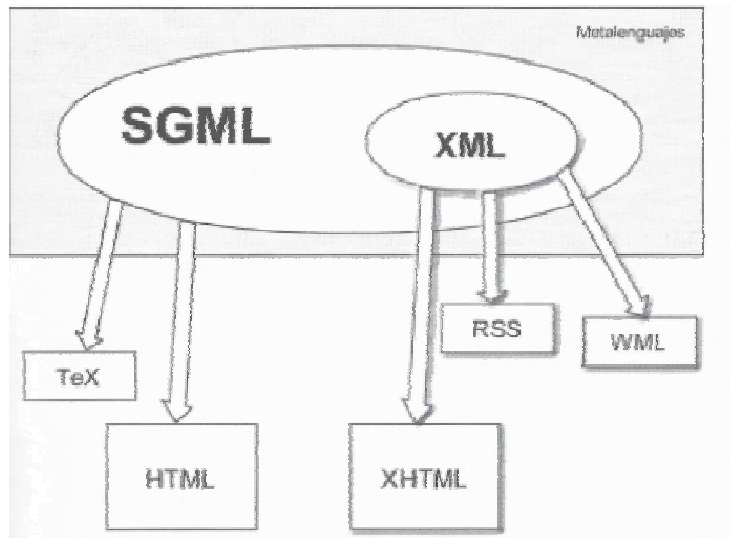
Toda organización de estándares pretende desarrollar normas que sean de amplio seguimiento por parte de la comunidad, para lo cual es imprescindible el consenso con las empresas involucradas como los navegadores, buscadores, desarrolladores web y fabricantes de dispositivos móviles.

Entre sus miembros se encuentran las principales empresas del sector como Microsoft, Apple o Google entre otras.



Además de producir estándares, la Comunidad W3C ha creado Software de Código Abierto, siendo el más conocido el validador W3C, que nos será de utilidad con HTML, CSS, contenido mobileOK y otras tecnologías.

### 1.11 Introducción a los principales lenguajes de marcas



El origen de los lenguajes de marcas como ya sabemos es SGML, del cual se derivan directamente algunos lenguajes como HTML. También se creó a partir de él por simplificación XML, otro metalenguaje más fácil de usar y entender. A partir de XML se han creado muchos lenguajes como XHTML, RSS y un largo etc.

### 1.12 Bibliografía

<http://www.ticarte.com/contenido/que-son-los-lenguajes-de-marcas>

Castro Ramos Juan Manuel, Rodríguez Sánchez, José Ramón. "Lenguaje de marcas y sistemas de gestión de información". Ed. Garceta