

Introducción a los Lenguajes de Marcas

En este capítulo se introduce el concepto de Lenguaje de Marcas y su utilidad como medio de transmisión de información entre plataformas y aplicaciones. Se introducirán las características y ventajas de los lenguajes de marcas y se hará una breve introducción sobre el lenguaje de marca más utilizado: XML.



Introducción a los Lenguajes de Marcas by Rafael Lozano is licensed under a [Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 España License](https://creativecommons.org/licenses/by-nc-sa/3.0/es/).

Tabla de contenido

1	Introducción.....	1
1.1	Anatomía de una marca.....	1
1.2	Los lenguajes de marcas en un entorno web.....	3
1.3	Evolución histórica de los LM.....	5
1.3.1	El origen: GML.....	6
1.3.2	La estandarización: SGML.....	6
1.3.3	La popularización: HTML.....	7
1.3.4	La madurez: XML.....	8
1.4	Características generales de los LM.....	8
1.4.1	Texto plano.....	8
1.4.2	Compactibilidad.....	9
1.4.3	Independencia del dispositivo final.....	9
1.4.4	Especialización.....	9
1.4.5	Flexibilidad.....	9
1.5	Ventajas y desventajas de los LM.....	9
1.6	Clasificación de los LM.....	10
1.6.1	De presentación.....	10
1.6.2	Descriptivo, estructural o semántico.....	11
1.6.3	Híbrido.....	11
1.7	Ámbitos de aplicación.....	12
2	Introducción a XML.....	13
2.1	Estructura de un documento XML.....	14
2.2	Reglas sintácticas.....	15
2.3	Documento XML bien formado.....	15
2.4	Espacio de nombres.....	16
3	Organizaciones y estándares.....	16
4	Bibliografía.....	19

Introducción a los Lenguajes de Marcas

1 Introducción

Los documentos electrónicos constan de un conjunto de información al que se le ha aplicado un determinado formato para hacerlos más atractivos y comprensibles. Utilizamos tamaños de letra, estilos de letra, listas numeradas o con viñetas, tablas, colores, etc. La forma de aplicar este formato dependerá del documento en si y de la aplicación que lo ha generado, pero el principio básico es que un fragmento de texto del documento con un determinado formato disponga de información añadida definiendo dicho formato.

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

Un documento que contenga exclusivamente texto es perfectamente legible por nosotros, aunque evidentemente, tedioso, soso e inapropiado para publicar páginas web. Si le aplicamos formato mediante un LM, como por ejemplo HTML, obtenemos un archivo también legible pero más difícil de interpretar. Sin embargo, a la hora de mostrarlo por pantalla, la aplicación del formato al texto es tarea del agente de usuario (navegador web), el cual 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** que delimita un fragmento de texto para realizar algún procesamiento sobre ese texto.*

1.1 Anatomía de una marca

Generalmente, las marcas están formadas por una palabra o algún código nemotécnico

que describe su función encerrada entre los símbolos menor que (<) y mayor que (>) como por ejemplo <table>. Dependiendo del tipo de marca, es muy habitual que aparezcan por parejas, una de comienzo que precede a un fragmento de texto, y otra de fin justo después de dicho fragmento de texto, para delimitar la aplicación de esa marca al fragmento encerrado entre la apertura y el fin. Así, el fragmento de texto delimitado por la marca de inicio y de fin será la parte del documento a la que se tenga que aplicar el formato o función indicado por la marca en cuestión.



Figura 1.- Marca

Por ejemplo, supongamos el siguiente texto en el que aparece unas marcas de inicio y fin sobre un fragmento.

La equivalencia entre masa y energía es $E=mc^2$

Vemos que la marca ^{es el inicio y} es la marca de final. Entre el inicio y el fin se encuentra el texto 2. Esta marca se aplica únicamente al texto delimitado por el inicio y el fin y, en este caso, provocará que el 2 se vea como un **superíndice**. El resultado sería $E=mc^2$.

Un documento de mayor tamaño combinará su contenido con múltiples marcas diseminadas por todo el texto, las cuales indicarán algún tipo de procesamiento dentro del documento y, como vimos en el ejemplo anterior, este procesamiento se aplicará a fragmentos de texto concreto, aquellos delimitados por las marcas de inicio y final.

Por ejemplo, el siguiente documento es una página web realizada con uno de los lenguajes de marcas más populares: HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Mi página de prueba</title>
  </head>
  <body>
    <h1>Esto es un título de nivel 1</h1>
    <p>Esto es un sencillo ejemplo de documento electrónico
    realizado con un lenguaje de marcas como HTML. En el mismo
    podemos apreciar diferentes elementos identificados con una
```

```
marca o etiqueta.</p>
  <p>A lo largo de estos capítulos veremos que estructura
  tienen los documentos hechos con lenguajes de marcas y el
  significado de cada etiqueta</p>
  
</body>
</html>
```

Vemos que hay una serie de marcas, como por ejemplo: `<h1>Esto es un título de nivel 1</h1>` es un texto delimitado por una marca en la apertura (`<h1>`) y otra en el cierre (`</h1>`) para delimitar al texto sobre el que se aplica.

Las marcas pueden **incluir atributos**, los cuales se emplean para añadir alguna propiedad o característica al elemento de la marca. Los atributos generalmente tienen un valor, el cual se encierra entre comillas y se separa del atributo por el símbolo `=`. Los atributos solamente se incluyen, cuando proceda, en la marca de inicio, nunca en la de fin. Su sintaxis sería:

```
<marca atributo="valor">Texto</marca>
```

El nombre del atributo siempre se introduce en la marca de inicio, nunca en la de fin. Por ejemplo:

```
<p align="center">Párrafo con alineación centrada</p>
```

Puede haber varios separados por espacios. Por ejemplo:

```
<td colspan="2" rowspan="3">Celda combinada</td>
```

Las marcas para elementos que **no tienen contenido** no tienen cierre, únicamente apertura. A estos elementos se les conoce como **elementos vacíos**. En este caso la marca incluye el símbolo `/` antes del `>`. Por ejemplo ``

```

```

1.2 Los lenguajes de marcas en un entorno web

Un lenguaje de marca **NO es un lenguaje de programación**, aunque se llame también lenguajes. De hecho, no debemos utilizar la palabra programar cuando nos referimos a la confección de documentos electrónicos mediante LM, de igual forma que tampoco nos referimos a programar cuando creamos un documento electrónico con un procesador de textos. Un LM no es un lenguaje de programación puesto que no disponen de las estructuras de programación como variables, arrays, sentencias de control, funciones, etc.

Sin embargo, los LM 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 final de la información contenida en el documento. Quizás lo más habitual, es que el usuario final sea una persona utilizando un navegador web en un ordenador, pero aun así, tenemos que considerar que no todas las personas son iguales, por lo que cada tipo de persona necesitará tener la información del documento adaptada a sus necesidades. Por tanto, otros destinatarios podrían ser: usuarios en dispositivos móviles, usuarios con deficiencias visuales o motrices, usuarios de avanzada edad, etc. La presentación de 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.

Sin embargo, las personas pueden no ser el único destinatario de un documento realizado con algún LM. Existen multitud de tecnologías basadas en el intercambio de información entre dispositivos, o entre aplicaciones, empleando documentos confeccionados con LM, lo que daría lugar a que el destinatario no fuera una persona, sino otro tipo de entidad. Por ejemplo un dispositivo periférico como la impresora, los robots de los buscadores, archivos de configuración de servicios en un servidor, una aplicación accediendo a una base de datos, etc. Independientemente de si el destinatario es una persona o un dispositivo, es más correcto utilizar el término general agente de usuario, ya que el documento electrónico se procesa por una aplicación (por ejemplo el navegador web) el cual se encargará de interpretar su contenido bien para una persona, visualizando su contenido, bien para otra función específica.

Un agente de usuario es una aplicación que actúa como cliente realizando solicitudes a un servidor en un protocolo de red y encargándose de interpretar la respuesta recibida.

La cuestión es que el LM debe ser independiente del destinatario final. El agente de usuario contiene un intérprete del lenguaje que 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 otro lado, y en el caso de los documentos web, para independizar aún más la representación de la página web de su contenido, se creó CSS, que no es un LM 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, si vemos la web de la Real Academia de la Lengua en un PC con el navegador web veremos lo siguiente:

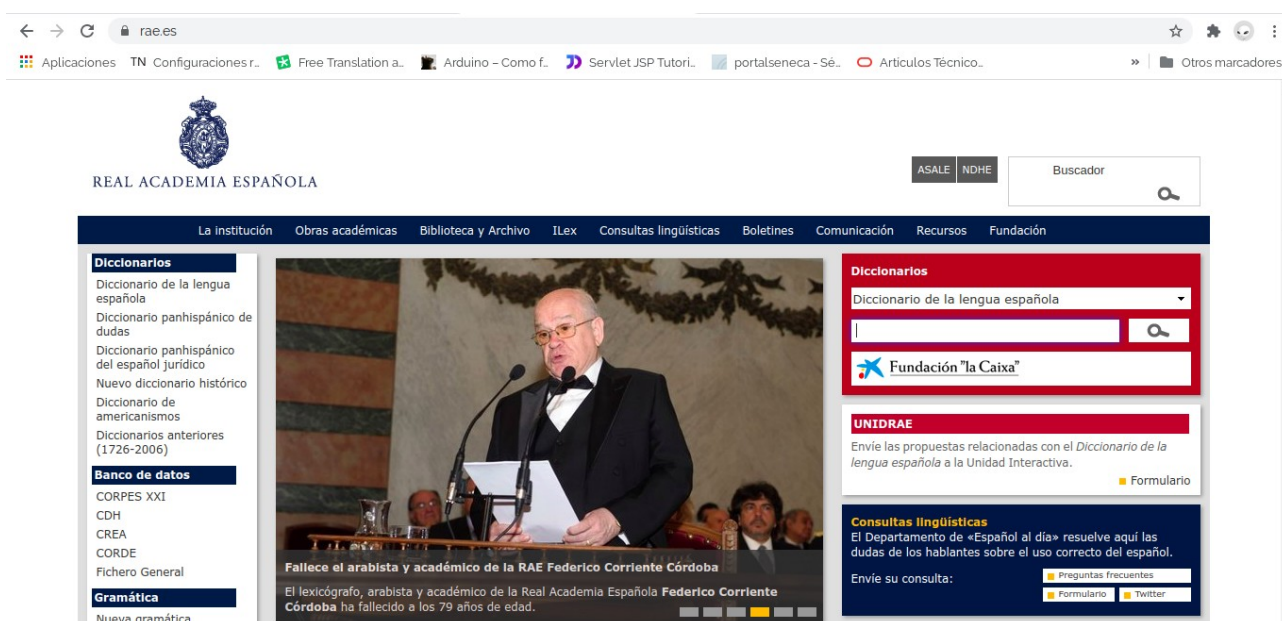


Figura 2: Web de la RAE vista desde un ordenador

Ahora bien, si visualizamos la misma página desde un móvil se verá lo siguiente:

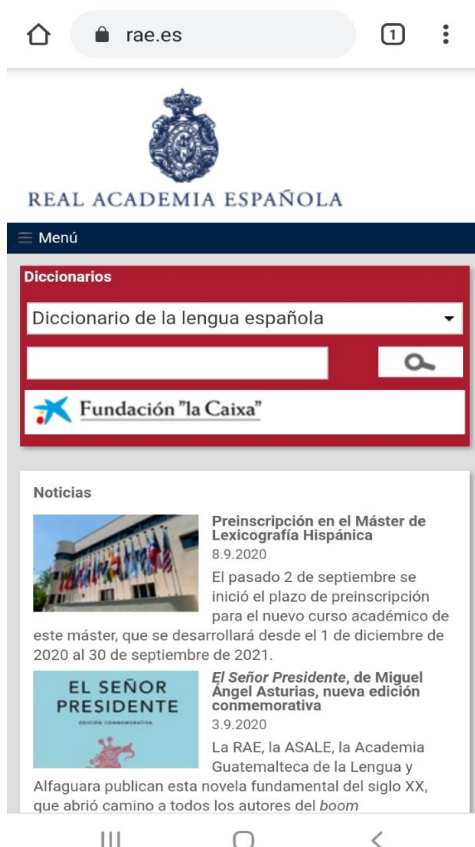


Figura 3: Web de la RAE sobre un móvil

1.3 Evolución histórica de los LM

En la década de los 60 las empresas de publicación y manejo de documentos

electrónicos tenían el problema de falta de compatibilidad entre aplicaciones. Cada aplicación utilizaba sus propias marcas para describir los diferentes elementos, esto impedía el intercambio de documentos entre diferentes aplicaciones y plataformas. Otra carencia importante era la separación entre estructura y aspecto del documento, es decir, el documento electrónico combinaba marcas para el contenido y el formato del documento.

1.3.1 El origen: GML

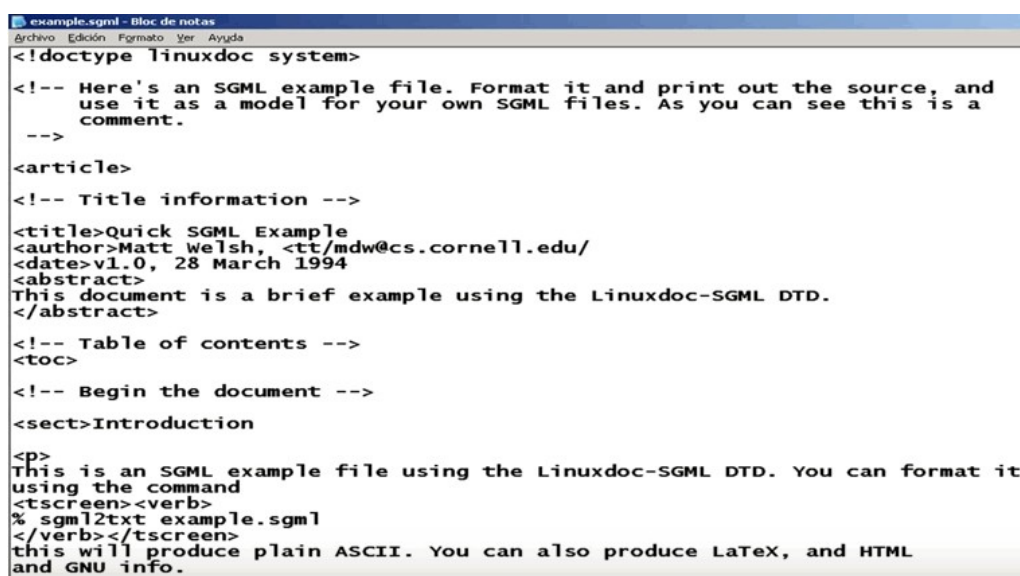
IBM intentó resolver estos problemas a través de un lenguaje de marcas denominado **GML (Generalized Markup Language)**. Éste independiza el documento del dispositivo que lo va a utilizar, usando marcas genéricas. Por otro lado GML incorpora marcas descriptivas para la estructura del documento que permiten 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. El marcado, por tanto, se centra en definir la estructura del texto y no su presentación visual.

El lenguaje GML fue un gran éxito y pronto se extendió a otros ámbitos, siendo adoptado por el gobierno de Estados Unidos, con lo que surgió la necesidad de estandarizarlo.

1.3.2 La estandarización: SGML

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

El SGML **especifica la sintaxis** para la inclusión de marcas en los textos, así como la sintaxis del documento indicando qué etiquetas están permitidas y dónde se pueden ubicar dentro del documento. Esta especificación de la sintaxis se realiza mediante el **Tipo de Definición de Documento (DTD) o esquema**. Esto permitía que un autor emplease cualquier marca que quisiera, eligiendo nombres para las etiquetas que tuvieran sentido tanto por el tema del documento como por el idioma. Por tanto, **SGML es un metalenguaje**, del que se derivan varios lenguajes específicos para su uso en el intercambio electrónico, manejo y publicación de documentos.



```

example.sgml - Bloc de notes
Archivo Edición Formato Ver Ayuda
<!doctype linuxdoc system>

<!-- Here's an SGML example file. Format it and print out the source, and
      use it as a model for your own SGML files. As you can see this is a
      comment.
-->

<article>

<!-- Title information -->

<title>Quick SGML Example
<author>Matt Welsh, <tt/mdw@cs.cornell.edu/
<date>v1.0, 28 March 1994
<abstract>
This document is a brief example using the Linuxdoc-SGML DTD.
</abstract>

<!-- Table of contents -->
<toc>

<!-- Begin the document -->

<sect>Introduction

<p>
This is an SGML example file using the Linuxdoc-SGML DTD. You can format it
using the command
<tscreen><verb>
% sgm12txt example.sgml
</verb></tscreen>
this will produce plain ASCII. You can also produce LaTeX, and HTML
and GNU info.
  
```

Figura 4: Ejemplo documento SGML

El SGML tuvo una gran aceptación y hoy día se emplea en campos en los que se requiere documentación a gran escala. A pesar de ello, resultó farragoso y difícil de aprender, como consecuencia de la ambición de los objetivos previstos. Su gran potencia era a la vez una ventaja y una desventaja. Por ejemplo, ciertas etiquetas podían tener solo principio, o solo final, o incluso ser obviadas, pensando en que los textos serían redactados a mano y que así se ahorrarían pulsaciones de teclas. Sin embargo fue un punto clave en el desarrollo de los lenguajes de marcas actuales, ya que la gran mayoría derivan de este.

1.3.3 La popularización: HTML

En 1991, parecía que los editores WYSIWYG (que almacenan los documentos en formatos **binarios propietarios**) abarcarían casi la totalidad del procesamiento de textos, relegando al SGML a usos profesionales o industriales muy específicos. Sin embargo, la situación cambió drásticamente cuando **Sir Tim Berners-Lee utilizó la sintaxis SGML para crear el HTML.**

Este lenguaje era similar a cualquier otro creado a partir del SGML, sin embargo resultó extraordinariamente sencillo. La flexibilidad y escalabilidad del marcado HTML fue uno de los principales factores, junto con el empleo de URLs y la distribución libre de navegadores, del éxito de la World Wide Web.

El HTML es hoy día el tipo de documento más empleado en el mundo. Su sencillez era tal que cualquier persona podía escribir documentos en este formato, sin apenas necesidad de conocimientos de informática. Esta fue una de las razones de su éxito, pero también condujo a un cierto caos. El crecimiento exponencial de la web en los años 90 produjo documentos en cantidades ingentes pero mal estructurados, problema agravado aún más por la falta de respeto por los estándares, por parte de diseñadores web y fabricantes de software.

1.3.4 La madurez: XML

La respuesta a los problemas surgidos en torno al HTML vino de la mano del XML (*eXtensible Markup Language*). El XML es un metalenguaje que permite crear etiquetas adaptadas a las necesidades (de ahí lo de «extensible»). El estándar define cómo pueden ser esas etiquetas y qué se puede hacer con ellas. Es además especialmente estricto en cuanto a lo que está permitido y lo que no, todo documento debe cumplir **dos condiciones: ser válido y estar bien formado.**

El XML fue desarrollado por el W3C con el objetivo principal de simplificar SGML para adaptarlo a un campo muy preciso: documentos en internet.

El nuevo lenguaje se extendió con rapidez, ya que todo documento XML es a su vez SGML. Los programas y documentos creados para y con SGML podían convertirse casi automáticamente al nuevo lenguaje. El XML simplificó radicalmente la complejidad del SGML, facilitando el aprendizaje y la implementación del nuevo estándar. Se solucionaron además viejos problemas, como los surgidos de la internacionalización, y la imposibilidad de validar un documento sin esquema. El acierto fundamental de este lenguaje es que logra un equilibrio entre simplicidad y flexibilidad.

El XML fue ideado en principio para entornos semiestructurados, como textos y publicaciones. Uno de los ejemplos más claros es el XHTML, la redefinición del HTML en clave XML, con las ventajas que ello supone. Sin embargo pronto se observó que sus virtudes podían ser útiles en campos bien distintos. Los lenguajes basados en XML tienen aplicaciones incontables, como en la transacción de datos entre servidores, intercambio de información financiera, fórmulas y reacciones químicas, y un largo etcétera.

1.4 Características generales de los LM

Entre las características de los LM podemos encontrar las siguientes:

1.4.1 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. Algunos lenguajes de presentación guardan la información en archivos binarios como '.doc' de MS Word donde solo una pequeña parte de la información es legible. Esto es una ventaja evidente respecto a los sistemas de archivos binarios, que requieren siempre de un programa intermediario para trabajar con ellos. Un documento escrito con lenguajes de marcado puede ser editado por un usuario con un sencillo editor de textos, sin perjuicio de que se puedan utilizar programas más sofisticados que faciliten el trabajo.

Los caracteres del documento se pueden codificar con distintos juegos de caracteres dependiendo del idioma o alfabeto que se necesite, por ejemplo: ASCII, ISO-8859-15, UTF-8.

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

1.4.2 Compactibilidad

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

El código entre símbolos mayor y menor, 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 de segundo nivel. El texto entre las marcas es el propio contenido del documento.

1.4.3 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.

1.4.4 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.

1.4.5 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.

1.5 Ventajas y desventajas de los LM

Entre las ventajas de los LM están:

- ✓ **Sencillez** → Con unos conocimientos mínimos se pueden utilizar LM para generar documentos electrónicos y definir su estructura.
- ✓ **Estructurado** → El contenido de los documentos electrónicos presenta una estructura lo que facilita su comprensión y tratamiento.
- Herramientas específicas → Se pueden generar documentos electrónicos mediante herramientas específicas que el usuario profano podría utilizar.
- Pocos recursos → Los documentos electrónicos generados mediante LM son pequeños, por tanto ocupan poco espacio y se transmiten muy rápido por Internet.
- Despliegue rápido → Un usuario tarda poco tiempo en generar los documentos electrónicos y subirlos a Internet para su distribución.
- Fácil aprendizaje → No se necesitan conocimientos específicos de sistemas o

aplicaciones para generar documentos electrónicos mediante lenguajes de marcas.

- Estándar → Todos los navegadores web y múltiples aplicaciones emplean documentos electrónicos generados mediante LM, lo que facilita la interoperabilidad.

Sin embargo, también adolecen de ciertas desventajas:

- Lenguaje estático → El contenido incluido en un documento electrónico se genera para luego almacenarse y distribuirse. Cualquier cambio requiere ser distribuido de nuevo.
- Falta de homogeneidad en la presentación → La interpretación de las marcas en cada navegador puede diferir.
- Información irrelevante → Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- Diseño lento → Antes de generar un documento electrónico hay que crear un diseño de su estructura para poder validarlo posteriormente.
- Etiquetas limitadas → Un documento electrónico no tiene la misma potencia y flexibilidad que otros tipos de documentos, como los generados por procesadores de texto.

1.6 Clasificación de los LM

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

1.6.1 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 Anal, 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, como por ejemplo RTF y TeX

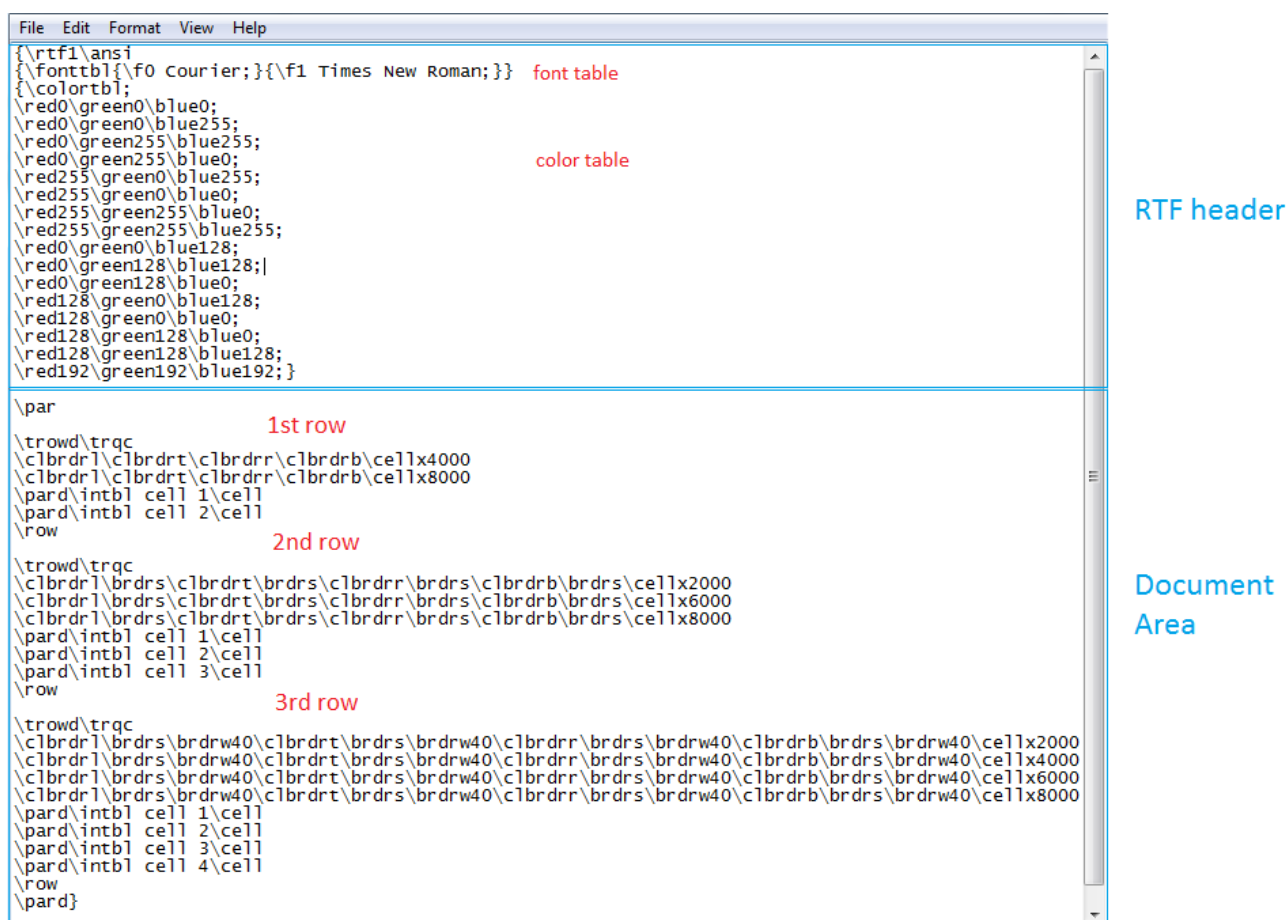


Figura 5: Ejemplo de documento RTF

1.6.2 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.

Estos lenguajes crean documentos con estructura jerárquica en árbol que almacenan información, por eso se les considera también bases de datos, pero de tipo jerárquico, no relacional. Por lo tanto no usan tablas ni respetan las reglas de integridad propias de las BD relacionales, y por ello se les llama bases de datos semiestructuradas.

XML es un metalenguaje expresamente diseñado para generar marcado descriptivo y los lenguajes derivados de XML con este propósito son: EBML, RDF, XFML, OWL y XTM. Aunque XML almacena información de todo tipo, los demás tienen contenido específico. Ejemplos: ASN.1, YAML y los derivados de XML.

1.6.3 Híbrido

Lenguajes que contienen marcas de los dos tipos anteriores indistintamente. Por ejemplo HTML (derivado de SGML), XHTML, WML (derivados de XML). En este caso el lenguaje permite tanto representar la estructura del documento como su presentación.

1.7 Ámbitos de aplicación

Las áreas de aplicación de los LM suelen ser las siguientes:

- Desarrollo web → El uso más extendido de documentos electrónicos es en forma de páginas web, generadas de forma estática o dinámica, confeccionados mediante uno de los lenguajes de marcas más populares: HTML
- Ámbito científico → En entornos científicos es muy común el uso de fórmulas matemáticas complejas. Para poder representar este tipo de información se creó el lenguaje MathML, basado en XML. Con él se pueden representar expresiones matemáticas como ecuaciones, límites, fórmulas físicas, etc.
- Gráficos vectoriales → Los formatos de imagen más habituales son jpg, png y gif. Todos ellos son mapas de bits que tienen el inconveniente de que cuando su tamaño crece pierden definición. Sin embargo, no ocurre lo mismo con los gráficos vectoriales escalables realizados con el lenguaje SVG que permite generar imágenes bidimensionales independientes de la resolución, lo que permite aumentar su tamaño sin perder definición.
- Metainformación → Un metacontenido no es más que la información relativa al contenido del documento, como su título, autor, tamaño del archivo, fecha de creación, historial de cambios, palabras clave, y demás información asociada. Se puede utilizar un metacontenido, por ejemplo, para realizar búsquedas, filtrar información, configuración de servicios y aplicaciones, etc. Además tenemos la posibilidad de definir estos metacontenidos fuera del documento para emplearlos en múltiples documentos.
- Bases de datos → Al estar la información altamente estructurada se puede crear una base de datos mediante un LM, la cual permitiría un acceso a la información extremadamente rápido. De hecho, algunos sistemas gestores de bases de datos relacionales incorporan funciones para transformar la información que almacenan a un formato de LM y también hay productos de bases de datos específicos que emplean LM en exclusiva.
- Intercambio de información → Compartir grandes volúmenes de información entre plataformas de software y sistemas operativos diferentes de forma sencilla, segura y, sobre todo, fiable.
- Mensajería → Respecto al intercambio de mensajes se busca la flexibilidad y sencillez que ofrece un LM para intercambiar documentos electrónicos a través de mensajes de correo electrónico. Tradicionalmente se ha empleado un formato de documento electrónico complejo y difícil de leer. El uso de un LM no solo facilita la legibilidad del documento, y por ende el desarrollo de aplicaciones para su interpretación, sino también la estructura de sus elementos tiene sentido en sí mismo.

2 Introducción a XML

El lenguaje XML es una **simplificación y adaptación del lenguaje SGML**, el cual permite definir lenguajes específicos. Por lo tanto, XML no es un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades, es decir, lo que hemos llamado un metalenguaje. Para describir la relación con SGML a menudo se utiliza la regla 80/20: 80% de funcionalidad y 20% de complejidad.

Algunos de los lenguajes que se basan en XML para su definición son XHTML, SVG, MathML, RSS, etc.

Como características podemos citar:

- ✓ **Extensible**, se pueden definir nuevas etiquetas
- ✓ **Versátil**, separa contenido, estructura y presentación
- ✓ **Estructurado**, se pueden modelar datos a cualquier nivel de complejidad
- ✓ **Validable**, cada documento se puede validar frente a un DTD/Schema
- ✓ **Abierto**, independiente de empresas, sistemas operativos, lenguajes de programación o entornos de desarrollo.
- ✓ **Sencillo**, fácil de aprender y de usar.

XML no se utiliza solo en Internet, sino que se está convirtiendo en un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos ligeras, editores de texto, hojas de cálculo, transacciones comerciales y en general donde se necesite almacenar información sin las restricciones de un SGBD relacional.

```
<?xml version="1.0" ?>
- <pedidos>
- <pedido cod="1">
  <fecha>01-01-2013</fecha>
  <pu>45.5</pu>
  <cantidad>2</cantidad>
  <descripcion>Botella de Vino</descripcion>
  <tipo>C</tipo>
</pedido>
- <pedido cod="2">
  <fecha>31-12-2012</fecha>
  <pu>25</pu>
  <cantidad>1</cantidad>
  <descripcion>Menu Ejecutivo</descripcion>
  <tipo>A</tipo>
</pedido>
</pedidos>
```

Figura 6: Ejemplo de documento XML

2.1 Estructura de un documento XML

La estructura general de un documento XML está formada por **dos partes**:

- ✓ **Prólogo (opcional)** → Contiene una secuencia de instrucciones de procesamiento y/o declaración del tipo de documento. Se puede dividir en dos partes:
 - **Declaración XML**. Establece la versión de XML, el tipo de codificación y si es un documento autónomo.
 - **Declaración de tipo de documento**. Establece la estructura del contenido que aparece en el cuerpo.
- ✓ **Cuerpo**: es el contenido de información del documento, organizado como un árbol jerárquico único de elementos marcados.

El estándar también permite la inclusión opcional de un **epílogo**, al final del documento, que puede contener **instrucciones de procesamiento**. Esta parte en general se omite dada su poca utilidad, ya que resulta poco intuitivo poner las instrucciones de procesamiento al final.

Las instrucciones de procesamiento se utilizan para enviar información a las aplicaciones que van a procesar el documento XML. Las instrucciones de procesamiento pueden aparecer en varios lugares del documento, por ejemplo entre el prólogo y el cuerpo, dentro del cuerpo o en el epílogo.

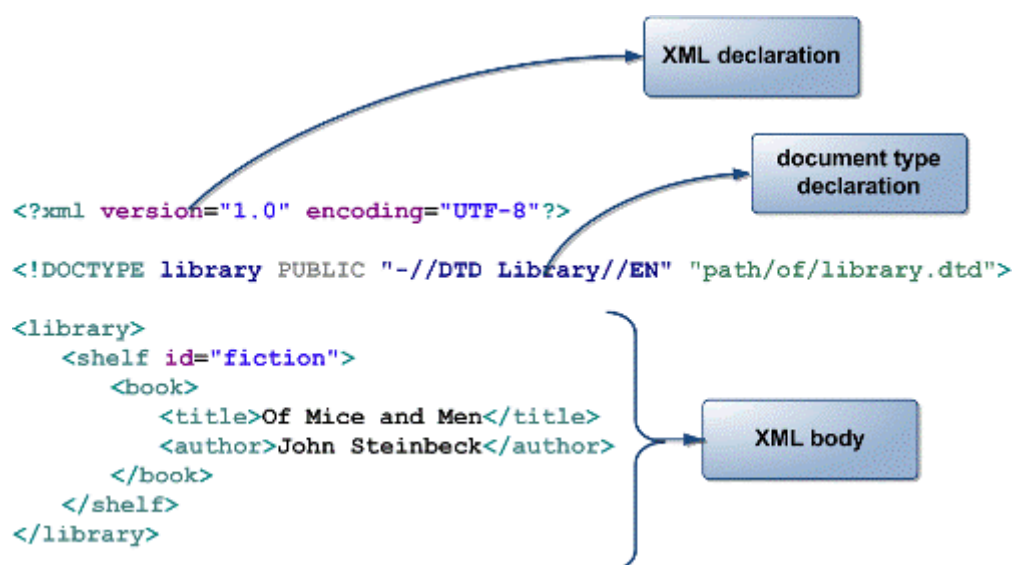


Figura 7: Estructura de documento XML

2.2 Reglas sintácticas

Una **sintaxis** es un conjunto de reglas que se deben aplicar a la hora de crear un documento XML. Estas reglas las dicta el W3C y entre ellas destacan:

- Hay dos tipos de construcciones: el marcado (entre `<...>` ó `&...;`) y los datos de tipo carácter (todo lo demás).
- El XML es **sensible a mayúsculas y minúsculas**.
- Todo documento XML se compone de elementos. Cada elemento está delimitado por una etiqueta de comienzo y otra de fin, a no ser que sea vacío. Los elementos vacíos constan de una única etiqueta. Los nombres de las etiquetas son arbitrarios y no pueden contener espacios.
- Cada elemento puede contener datos de tipo carácter, elementos, ambas cosas a la vez o puede estar vacío.
- En todo documento debe haber un elemento, llamado raíz de documento, que contenga a los demás. En el elemento raíz la etiqueta de inicio ha de ser la primera de todas y la de cierre la última de todas.
- Todos los elementos deberán estar correctamente anidados. No se puede mezclar la anidación de las etiquetas: los elementos deben abrirse y cerrarse por orden.
- Los elementos pueden tener atributos (propiedades) que nos ofrecen información sobre ellos. Los valores de los atributos deben ir entrecomillados. Tanto atributos como valores son arbitrarios.

2.3 Documento XML bien formado

Un documento XML bien formado, es aquel que cumple con todas las reglas sintácticas definidas para XML. Los procesadores de documentos XML pueden rechazar cualquier

documento que no esté bien formado.

Un procesador de documento XML es una aplicación software que lee un documento XML y realiza acciones sobre su contenido.

No hay que confundir un documento XML bien formado con un documento válido. Un documento XML válido es el que está bien formado, y además cumple con la definición de un lenguaje de marcado particular especificado para el documento. Es decir, el cuerpo del documento tiene una estructura de elementos compatible con el lenguaje concreto al que corresponde. Así, todo documento XML válido es un documento bien formado (todos los documentos XML tienen que estar bien formados), pero no ocurre al contrario.

Un procesador XML conforme con la especificación XML, comprobará que el documento cumple estas restricciones de buena formación. Cualquier restricción no cumplida será detectada y se tratará como un error fatal: el procesador informará a la aplicación y dejará de trabajar de una manera normal. Fundamentalmente, el objetivo de estas restricciones es asegurar que los documentos XML puedan ser interpretados por los procesadores XML sin ninguna ambigüedad, de manera que todos los elementos (y atributos) quedan perfectamente definidos.

2.4 Espacio de nombres

Un espacio de nombres XML es una recomendación W3C para proporcionar elementos y atributos con nombre único en un archivo XML. Un archivo XML puede contener nombres de elementos o atributos procedentes de más de un vocabulario XML. Si a cada uno de estos vocabularios se le da un espacio de nombres, un ámbito semántico propio, referenciado a una URI donde se listen los términos que incluye, se resuelve la ambigüedad existente entre elementos o atributos que se llamen igual. Los nombres de elementos dentro de cada espacio de nombres deben ser únicos.

La unicidad de nombres de elementos se resuelve mediante el espacio de nombres, no así la de atributos. No puede haber un elemento con dos atributos con el mismo nombre, aunque provengan de espacios de nombres diferentes.

3 Organizaciones y estándares

En el desarrollo de tecnologías relacionadas con los procesos industriales y telemáticos es fundamental la existencia de estándares emitidos por organizaciones de estandarización y reconocidos universalmente para garantizar la interoperabilidad de sistemas heterogéneos.

Normalización o estandarización es el proceso de especificación de normas que sirven de guía a fabricantes y desarrolladores para garantizar el correcto funcionamiento de elementos contruidos de forma independiente.

Aplicado al contexto de los lenguajes de marcas, sería el desarrollo de lenguajes

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 y otras tecnologías.

4 Bibliografía

WIKIPEDIA, *Espacio de nombres XML*. [acceso septiembre 2020]. Disponible en <https://es.wikipedia.org/wiki/Espacio_de_nombres_XML>

WIKIPEDIA, *Lenguaje de marcado*. [acceso septiembre 2020]. Disponible en <https://es.wikipedia.org/wiki/Lenguaje_de_marcado#Historia>

TODOXML, *Documentos XML bien formados* [acceso septiembre 2020]. Disponible en <<https://sites.google.com/site/todoxmldtd/referencia/referencia-de-xml/07-xml-bien-formados#:~:text=Un%20documento%20XML%20bien%20formado,formado%20con%20un%20documento%20v%C3%A1lido.>>>

ARRANZ,D., *Apuntes de HTML*. [acceso septiembre 2020]. Disponible en <https://www.dsi.uclm.es/personal/MiguelFGraciani/mikicurri/Docencia/LenguajesInternet0910/web_LI/Teoria/XML/Programaci%C3%B3n%20en%20castellano%20Apuntes%20de%20XML.%20Escribir%20XML.htm>

CASTRO,J.M. y RODRÍGUEZ, J.R., *Lenguajes de Marcas y Sistemas de Gestión de Información*. Ed Garceta 2012 – ISBN: 978-84-1545-217-1