1. Introducción

Un "lenguaje de marcas" **es un modo de codificar un documento** donde, junto con el texto, se **incorporan etiquetas**, marcas o anotaciones **con información adicional** relativa a la estructura del texto o su formato de presentación. Permiten hacer explícita la estructura de un documento, su contenido semántico o cualquier otra información lingüística o extralingüística que se quiera hacer patente.

Todo lenguaje de marcas está definido en un documento denominado **DTD** (**Document Type Definition**). En él se establecen las marcas, los elementos utilizados por dicho lenguaje y sus correspondientes etiquetas y atributos, su sintaxis y normas de uso.

Ejemplo

Aspecto de un documento realizado en un lenguaje de marcas

```
<carta>
        <fecha>22/11/2006</fecha>
        <presentacion>Estimado cliente:</presentacion>
        <contenido>bla bla bla bla …</contenido>
        <firma>Don José Gutiérrez González</firma>
</carta>
```

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

- **De presentación**: Define el formato del texto.
- **De procedimientos**: Orientado también a la presentación pero, en este caso, el programa que representa el documento debe interpretar el código en el mismo orden en que aparece.
- **Descriptivo o semántico**: Describen las diferentes partes en las que se estructura el documento pero sin especificar cómo deben representarse.

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

- **RTF** (Rich Text Format): Formato de Texto Enriquecido, fue desarrollado por Microsoft en 1987. Permite el intercambio de documentos de texto ente distintos procesadores de texto.
- **TeX**: Su objetivo es la creación de ecuaciones matemáticas complejas.
- **Wikitexto**: Permite la creación de páginas wiki en servidores preparados para soportar este lenguaje.
- **DocBook**: Permite generar documentos separando la estructura lógica del documento de su formato. De este modo, dichos documentos, pueden publicarse en diferentes formatos sin necesidad de realizar modificaciones en el documento original.

Tecnologías de internet:

- **HTML**, **XHTML**: (Hypertext Markup Language, eXtensible Hypertext Markup Language): Su objetivo es la creación de páginas web.
- **RSS**: Permite la difusión de contenidos web

Otros lenguajes especializados:

- **MathML** (Mathematical Markup Language): Su objetivo es expresar el formalismo matemático de tal modo que pueda ser entendido por distintos sistemas y aplicaciones.
- **VoiceXML** (Voice Extended Markup Language) tiene como objetivo el intercambio de información entre un usuario y una aplicación con capacidad de reconocimiento de habla.
- **MusicXML**: Permite el intercambio de partituras entre distintos editores de partituras.

2. Evolución de los lenguajes de marcas

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

- Los lenguajes de marcas.
  Paralelamente, también, surgen otros lenguajes informáticos orientados a la representación, almacenamiento y consulta eficiente de grandes cantidades de datos:
- Lenguajes y sistemas de bases de datos.

Los lenguajes de marcas surgieron, inicialmente, como lenguajes formados por el conjunto de códigos de formato que los procesadores de texto introducen en los documentos para dirigir el proceso de presentación (impresión) mediante una impresora. Como en el caso de los lenguajes de programación, inicialmente estos códigos de formato estaban ligados a las características de una máquina, programa o procesador de textos concreto y,

en ellos, inicialmente no había nada que permitiese al programador (formateador de documentos en este caso) abstraerse de las características del procesador de textos y expresar de forma independiente a éste la estructura y la lógica interna del documento.

Ejemplo

Código de marcas anterior a GML. Las etiquetas son de invención propia.

Dado el siguiente documento:

```
<times 14><color verde><centrado> Este texto es un ejemplo para mostrar la utilización primitiva de las marcas</centrado></color>
</times 14>
<color granate><times 10><cursiva>Para realizar este ejemplo se utilizan etiquetas de nuestra invención. </cursiva>
Las partes importantes del texto pueden resaltarse usando la
<negrita>negrita</negrita>, o el <subrayar>subrayado</subrayar></times 10></color>
```

Al imprimirlo se obtendría:

Este texto es un ejemplo para mostrar la utilización primitiva de las marcas

Para realiza este ejemplo se utilizan etiquetas de nuestra invención. Las partes importantes del texto pueden resaltarse usando la **negrita** , o el subrayado

Posteriormente, se añadieron este tipo de características, como medio de presentación a la pantalla. Los códigos de estilo de visualización anteriores ya no aparecían, y se empleaban otros medios para marcados, distintos de la inclusión a mano de cadenas formateadoras. Ese proceso se automatizó y bastaba pulsar una combinación de teclas, o un botón, para lograr los resultados requeridos. Aunque esto era sólo una abstracción, para su uso interno, las aplicaciones seguían utilizando marcas, para delimitar aquellas partes del texto que tenían un formato especial.

Este marcado estaba exclusivamente orientado a la presentación de la información, aunque pronto se percataron de las posibilidades del marcado y le dieron nuevos usos que resolvían una gran variedad de necesidades. De este modo apareció el formato generalizado.

2.1.- GML (Generalized Markup Language).

Uno de los problemas que se conocen desde hace décadas en la informática es la falta de estandarización en los formatos de información usados por los distintos programas.

Para resolver este problema, en los años sesenta **IBM** encargó a Charles F. Goldfab la construcción de un sistema de edición, almacenamiento y búsqueda de documentos legales. Tras analizar el funcionamiento de la empresa llegaron a la conclusión de que para realizar un buen procesado informático de los documentos había que establecer un formato estándar para todos los documentos que se manejaban en la empresa. Con ello se lograba gestionar cualquier documento en cualquier departamento y con cualquier aplicación, sin tener en cuenta dónde ni con qué se generó el documento. Dicho formato tenía que ser válido para los distintos tipos de documentos legales que utilizaba la empresa, por tanto, debía ser flexible para que se pudiera ajustar a las distintas situaciones.

El formato de documentos que se creó como resultado de este trabajo fue **GML**, cuyo objetivo era describir los documentos de tal modo que el resultado fuese independiente de la plataforma y la aplicación utilizada.

2.2.- SGML (Standard Generalized Markup Language).

El formato **GML** evolucionó hasta que en 1986 dio lugar al estándar **ISO 8879** que se denominó **SGML**. Éste era un lenguaje muy complejo y requería de unas herramientas de software caras. Por ello su uso ha quedado relegado a grandes aplicaciones industriales.

Ejemplo

Documento SGML sencillo:

```
<email>
        <remitente>
                <persona>
                <nombre> Pepito </nombre>
                <apellido> Grillo </apellido>
                </persona>
        </remitente>
        <destinatario>
                <direccion> pinocho@hotmail.com </direccion>
        </destinatario>
        <asunto>¿quedamos?</asunto>
        <mensaje> Hola, he visto que ponen esta noche la película que querías ver. ¿Te apetece ir?</mensaje>
</email>
```

2.3.- HTML (HyperText Markup Language).

En 1989/90 Tim Berners-Lee creó el **World Wide Web** y se encontró con la necesidad de organizar, enlazar y compatibilizar gran cantidad de información procedente de diversos sistemas. Para resolverlo creó un lenguaje de descripción de documentos llamado **HTML**, que, en realidad, era una combinación

de dos estándares ya existentes:

- **ASCII**: Es el formato que cualquier procesador de textos sencillo puede reconocer y almacenar. Por tanto es un formato que permite la trasferencia de datos entre diferentes ordenadores.
- **SGML**: Lenguaje que permite dar estructura al texto, resaltando los títulos o aplicando diversos formatos al texto.

HTML es una versión simplificada de **SGML**, ya que sólo se utilizaban las instrucciones absolutamente imprescindibles. Era tan fácil de comprender que rápidamente tuvo gran aceptación, logrando lo que no pudo **SGML**

HTML se convirtió en un estándar general para la creación de páginas web. Además, desde su creación, tanto las herramientas de software como los navegadores, que permiten visualizar páginas **HTML**, no han parado de mejorar.

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

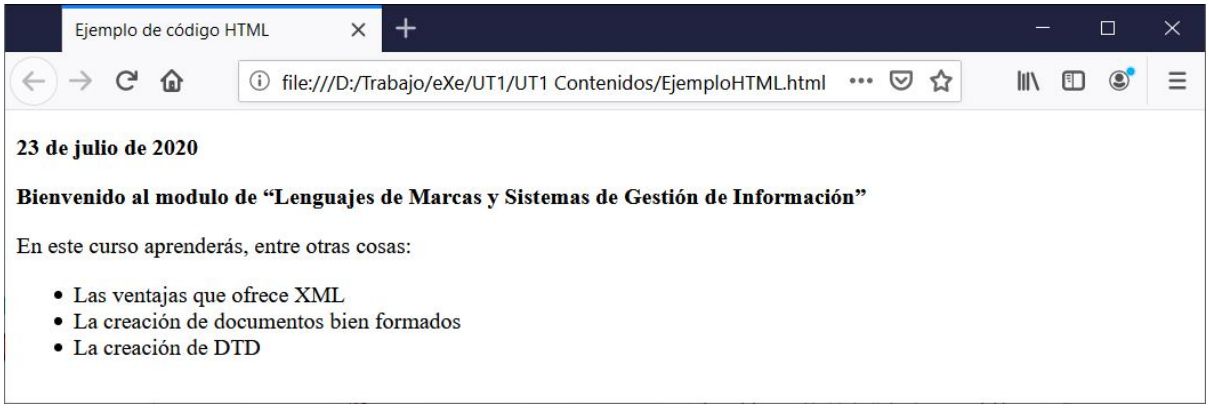
- No soporta tareas de impresión y diseño.
- El lenguaje no es flexible, ya que las etiquetas son limitadas.
- No permite mostrar contenido dinámico.
- La estructura y el diseño están mezclados en el documento.

Ejemplo

Ejemplo: Documento HTML

```
<html>
        <head>
                <title> Ejemplo de código HTML</title>
        </head>
        <body>
                <p></p>
                <p>
                        <b>23 de julio de 2020</b>
                </p>
                <p><b> Bienvenido al modulo de “Lenguajes de Marcas y Sistemas de Gestión de Información” </b></p>
                <p> En este curso aprender&aacute;s, entre otras cosas:<br/>
                <ul>
                        <li>Las ventajas que ofrece XML </li>
                        <li>La creaci&oacute;n de documentos bien formados </li>
                        <li>La creaci&oacute;n de DTD</li>
                </ul>
                </p>
        </body>
</html>
```

En el navegador se visualizaría así:



2.4.- XML (eXtensible Markup Language).

Para resolver estos problemas de **HTML** el **W3C** establece, en 1998, el estándar internacional **XML**, un lenguaje de marcas puramente estructural que **no incluye ninguna información relativa al diseño**. Está convirtiéndose con rapidez en estándar para el intercambio de datos en la Web. A diferencia de **HTML** las etiquetas indican el significado de los datos en lugar del formato con el que se van a visualizar los datos.

XML es un metalenguaje caracterizado por:

- Permitir definir etiquetas propias.
- Permitir asignar atributos a las etiquetas.
- Utilizar un esquema para definir de forma exacta las etiquetas y los atributos.

- La estructura y el diseño son independientes.

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

- **XSL**, eXtensible Style Language. Permite definir hojas de estilo para los documentos XML e incluye capacidad para la transformación de documentos.
- **XML Linking Language**, incluye Xpath, Xlink y Xpointer. Determinan aspectos sobre los enlaces entre documentos XML.
- **XML Namespaces**. Proveen un contexto al que se aplican las marcas de un documento de XML y que sirve para diferenciarlas de otras con idéntico nombre válidas en otros contextos.
- **XML Schemas**. Permiten definir restricciones que se aplicarán a un documento XML. Actualmente los más usados son las **DTD** (Document type Definition).

Ejemplo

Documento XML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE biblioteca">
<biblioteca>
        <ejemplar tipo_ejem="libro" titulo="XML practico" editorial="Ediciones Eni">
        <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paginas="347"></libro> </tipo>
        <autor nombre="Sebastien Lecomte"></autor>
        <autor nombre="Thierry Boulanger"></autor>
        <autor nombre="Ángel Belinchon Calleja" funcion="traductor"></autor>
        <prestado lector="Pepito Grillo">
            <fecha_pres dia="13" mes="mar" año="2009"></fecha_pres>
            <fecha_devol dia="21" mes="jun" año="2009"></fecha_devol>
        </prestado>
    </ejemplar>
    <ejemplar tipo_ejem="revista" titulo="Todo Linux 101. Virtualización en GNU/Linux" editorial="Studio Press">
        <tipo>
            <revista>
                <fecha_publicacion mes="abr" año="2009"></fecha_publicacion>
            </revista>
        </tipo>
        <autor nombre="Varios"></autor>
        <prestado lector="Pedro Picapiedra">
            <fecha_pres dia="12" mes="ene" año="2010"></fecha_pres>
        </prestado>
    </ejemplar>
</biblioteca>
```

2.5. XML vs HTML

A continuación encontrarás una tabla comparativa de ambos lenguajes

XML	HTML
- Es un perfil de **SGML**.	- Es una aplicación de **SGML**.
- Especifica cómo deben definirse conjuntos de etiquetas aplicables a un tipo de documento.	- Aplica un conjunto limitado de etiquetas sobre un único tipo de documento.
- Modelo de hiperenlaces complejo.	- Modelo de hiperenlaces simple.
- El navegador es una plataforma para el desarrollo de aplicaciones.	- El navegador es un visor de páginas.
- Fin de la guerra de los navegadores y etiquetas propietarias.	- El problema de la 'no compatibilidad' y las diferencias entre navegadores ha alcanzado un punto en el que la solución es difícil.

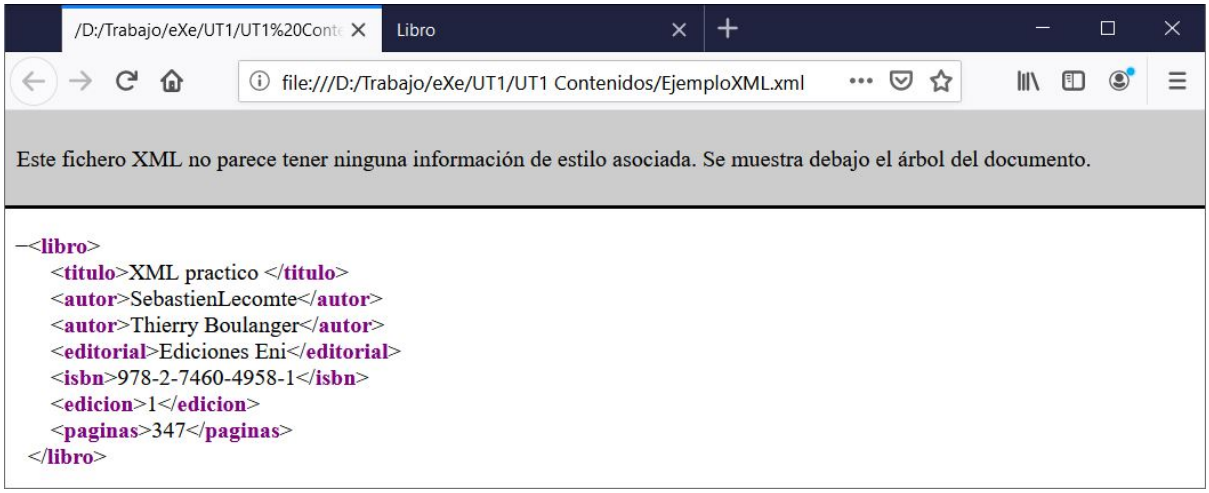
Ejemplo XML vs. HTML

Ejemplo fichero con código **XML**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE libro>
<libro>
    <titulo>XML practico </titulo>
    <autor>SebastienLecomte</autor>
    <autor>Thierry Boulanger</autor>
    <editorial>Ediciones Eni</editorial>
    <isbn>978-2-7460-4958-1</isbn>
    <edicion>1</edicion>
```

```
    <paginas>347</paginas>
</libro>
```

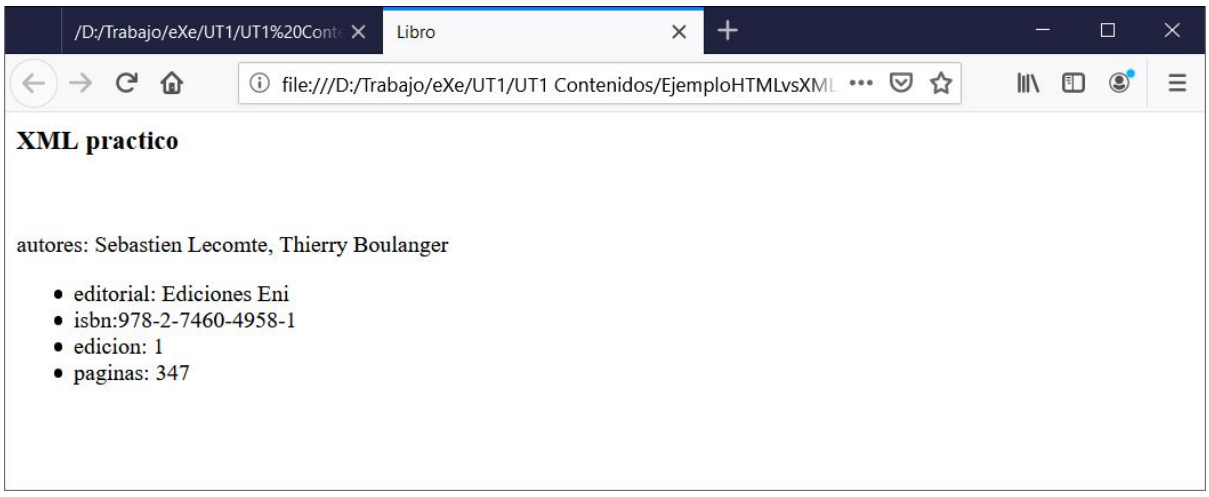
Visualización en Navegador fichero código **XML**



Ejemplo de fichero con código **HTML**

```
<html>
    <head>
        <title>Libro</title>
    </head>
    <body>
        <h3>XML practico</h3><br>
        <p>autores: Sebastien Lecomte,
        Thierry Boulanger</p>
        <ul>
            <li>editorial: Ediciones Eni</li>
            <li>isbn:978-2-7460-4958-1</li>
            <li>edicion: 1 </li>
            <li>paginas: 347</li>
        </ul>
    </body>
</html>
```

Visualización en Navegador fichero código **HTML**



2.6. Comparación de XML con SGML

XML	SGML
Uso sencillo	Uso complejo
Trabaja con documentos bien formados. No exige que estén validados	Solo trabaja con documentos válidos
Facilita el desarrollo de aplicaciones de bajo coste	Su complejidad hace que las aplicaciones informáticas para procesar SGML sean

XML	SGML
	muy costosas
Es muy utilizado en informática y en más áreas de aplicación	Solo se utiliza en sectores muy específicos
Compatibilidad e integración con HTML	No hay una compatibilidad con HTML definida
Formato y estilos fáciles de aplicar	Formateo y estilos relativamente complejos
No usa etiquetas opcionales	

Para saber más

La recomendación de **XML** publicada por el W3C es pública y accesible en:
https://www.w3.org/TR/xml/

2.7.- Etiquetas.

Los lenguajes de marcas utilizan una serie de etiquetas especiales intercaladas en un documento de texto sin formato. Dichas etiquetas serán posteriormente interpretadas por los intérpretes del lenguaje y ayudan al procesado del documento.

Las etiquetas **se escriben encerradas entre ángulos**, es decir < y >. Normalmente, **se utilizan dos etiquetas: una de inicio y otra de fin** para indicar que ha terminado el efecto que queríamos presentar. La única diferencia entre ambas es que la de cierre lleva una barra inclinada "/" antes del código.

< etiqueta>texto que sufrirá las consecuencias de la etiqueta< /etiqueta>

Ejemplo: Etiqueta HTML de subrayado (Underline)

```
<u>Esto está subrayado</u>
```

En el navegador el texto se verá:

Esto está subrayado

Las últimas especificaciones emitidas por el W3C indican la necesidad de que vayan escritas **siempre en minúsculas** para considerar que el documento está correctamente creado.

2.8. Herramientas de edición

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

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

Para crear documentos **XML** complejos e ir añadiendo datos es conveniente usar algún editor **XML**. Estos nos ayudan a crear estructuras y etiquetas de los elementos usados en los documentos, además algunos incluyen ayuda para la creación de otros elementos como **DTD**, hojas de estilo **CSS** o **XSL**, ...
El W3C ha desarrollado un editor de **HTML**, **XHTML**, **CSS** y **XML** gratuito cuyo nombre es:
Amaya (https://www.w3.org/Amaya/.

Procesadores XML
Para interpretar el código **XML** se puede utilizar cualquier navegador. Los procesadores de **XML** permiten leer los documentos **XML** y acceder a su contenido y estructura. Un procesador es un conjunto de módulos de software entre los que se encuentra un **parser** o analizador de **XML** que comprueba que el documento cumple las normas establecidas para que pueda abrirse. Estas normas pueden corresponderse con las necesarias para trabajar sólo con documentos de tipo válido o sólo exigir que el documento esté bien formado, primeros se conocen como validadores y los segundos como no validadores. El modo en que los procesadores deben leer los datos **XML** está descrito en la recomendación de **XML** establecida por **W3C**.

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

Puesto que XML se puede utilizar para el intercambio de datos entre aplicaciones, hay que recurrir a motores independientes que se ejecutan sin que nos demos cuenta. Por ejemplo JAXP de Sun.

Para saber más

Información sobre analizadores XML - http://xml.coverpages.org/index.html

expat - XML Parser Toolkit - http://www.jclark.com/xml/expat.html