

1.1. SGML



Índice

Objetivos	3
Introducción	4
El origen de los lenguajes de marcas.....	4
Tipos de lenguajes de marcas	5
Marcas de procesamiento	5
Marcas de descripción	5
El estándar SGML	6
Características de los lenguajes de marcas.....	7
Beneficios de los lenguajes de marcas	9
SGML	10
Estructura.....	10
Estructura natural	10
Estructura lógica	11
Estructura física	12
Componentes SGML	12
XML.....	14
Conceptos	14
Objetivos de XML	14
Componentes XML.....	14
Estructura de un XML.....	15
XSL.....	17
Despedida	18
Resumen.....	18

Objetivos

En esta unidad perseguimos los siguientes objetivos:

- Aprender qué son los lenguajes de marcas.
- Conocer los orígenes de SGML.
- Conocer la estructura en lenguajes de marcas a través de su definición.
- Conocer los lenguajes genéricos, en concreto SGML.
- Conocer el uso de DTD para generar tipos de lenguajes.
- Conocer ejemplos de estructuras en los lenguajes que derivan de SGML.

Introducción

El origen de los lenguajes de marcas

Los lenguajes de marcado se remontan al uso de la imprenta y a cómo los autores indicaban, a través de marcas, cómo se debían formatear los textos.

Normalmente **indicaban el estilo de la letra, la disposición de los títulos, los párrafos y demás elementos de formato.**

Las especificaciones o explicaciones se anotaban junto con el texto. Estas marcas especificaban operaciones tipográficas y la configuración que debía realizar el que manejaba la imprenta sobre cada elemento. Por lo tanto, eran instrucciones de formato que se aplicaban de manera global o individual sobre cada párrafo, palabra o bloque de texto.

Una vez que surge la informática se decide que las marcas deben ser estándares y escritas en ASCII, un código de 7 bits que representa los caracteres basado en el alfabeto latino.

• • • • = Ignorar la corrección y dejarlo como estaba:

en el ~~Quijote~~ Cervantes...

() = Quitar el espacio de más: la noticia

) (= Añadir un espacio; en este caso

[] = Cambiar el orden de letras, palabras o frases: cno

[] = Marcado en el margen izquierdo, alinear verticalmente

Con las líneas contiguas
En párrafos, sangría izquierda

a
a
a

[] = Marcado en el margen derecho, alinear verticalmente

Con las líneas contiguas
En párrafos, sangría derecha

b
b
b

[] = Sangrar línea

[] = Alinear a margen izquierdo

Ejemplo de símbolos de marcado de un texto.

¿Quieres saber más?

En el siguiente enlace puedes encontrar un listado con los símbolos más utilizados en la corrección de textos. <https://correctordetextos.com/signos.htm>

Tipos de lenguajes de marcas

Existen distintos tipos de lenguajes de marcado, que se combinan en la práctica.

Marcas de presentación

Este tipo de marcas **son códigos incrustados en el texto del documento** que producen el efecto WYSIWYG (*What You See Is What You Get*, “Lo que ves es lo que obtienes”), utilizados por los sistemas de procesamiento de textos tradicionales.

Dicho marcado se suele incluir en el documento de manera que el usuario no lo ve.

```
[mi_shortcode atributo="valor"]  
[mi_shortcode atributo="valor"]Contenidos[/mi_shortcode]
```

- Este tipo de marcado es útil para maquetar la presentación de un documento para su lectura, pero **es insuficiente para el procesamiento automático de la información**.
- El marcado de presentación resulta **más fácil de elaborar**, sobre todo para cantidades pequeñas de información.

Marcas de procesamiento

Describen las reglas de procesamiento del texto y cómo se imprimirá. **Un ejemplo de lenguajes que usan este sistema son PostScript y LaTeX.**

```
\begin{verbatim}  
Escribes tu código aquí  
\end{verbatim}
```

- Sus símbolos o marcas **indican la clase de operaciones tipográficas** que deben ser aplicadas a cada uno de los elementos del documento electrónico para dar formato al texto.
- Se utilizan **para configurar la apariencia física de los documentos**, tanto en pantalla como impreso.

Marcas de descripción

Es un lenguaje que conforma la estructura de un documento, identifica cada parte de este y el tipo de elementos que constituye cada parte.

A cada marca se le puede asignar un comportamiento determinado para dar forma al documento.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>title</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
  </head>
  <body>
    <header>
      <h1>Título de la WEB</h1>
    </header>
    <nav>
      <a href="http://dominio.com/seccion2.html">IR SECCIÓN 2</a>
      <a href="http://dominio.com/seccion2.html">IR SECCIÓN 3</a>
    </nav>
    <section>
      <article>
        <h2>CONTENIDO PRINCIPAL</h2>
        <p>Este es el contenido principal de mi web</p>
        <div>
          <p>Aquí tenéis una imagen.</p>
          
        </div>
      </article>
    </section>
    <aside>
      <h3>Banner de publicidad</h3>
      <a href="http://dominio-externo.com">
        
      </a>
      <h3>Testimonios</h3>
      <p>Me gusta mucho esta página.</p>
    </aside>
    <footer>
      <h4>Avisos legales</h4>
      <a href="http://dominio.com/aviso-legal">Política de cookies</a>
      <h4>Redes sociales</h4>
      <a href="http://facebook.com/mi-pagina-de-facebook">Mi Facebook</a>
    </footer>
  </body>
</html>

```

- Determinan **tanto la estructura lógica del documento electrónico como la descripción de su contenido.**
- Utilizan las **marcas o etiquetas para describir los fragmentos de texto.**
- Una de las virtudes del marcado descriptivo es su **flexibilidad**, ya que los fragmentos de texto se etiquetan tal como son, y no como deben aparecer.
- **Simplifica la tarea de formatear un texto**, debido a que la información está separada del propio contenido.

El estándar SGML

SGML surge de la necesidad de estandarizar los lenguajes de marcado. Los primeros lenguajes de marcado usaban el estándar ASCII y seguían unas reglas previamente establecidas, pero había muchos y de distintos fabricantes. **IBM fue el primero que**, preocupado por la gran cantidad de lenguajes de marcas de la época que hacían imposible el envío de documentos de

un sistema a otro, **definió el primer intento de estandarización, que fue el GML** (*Generalized Markup Language*).

En 1978 el Instituto Nacional Americano de Normalización (ANSI) diseñó la especificación del procesamiento de textos, creando el **estándar SGML**, que fue retomado en 1986 por la ISO y se transformó en la norma 8879. Las siglas del lenguaje SGML (*Standar Generalized Markup Language*) traducidas a "**Estándar de Lenguaje de Marcado Generalizado**", sirven para especificar las **reglas de etiquetado de documentos**, aunque deja libertad para diseñar las marcas a los distintos creadores o fabricantes.

```
<!ELEMENT DOCUMENTO - - (titulo, contenido, autor?) >
<!ELEMENT CONTENIDO - - (capitulo+) >

<!ELEMENT CAPITULO - (subtitulo, parrafo?)

<titulo>Lenguajes de Marcas</titulo>
<contenido>
  <capitulo>
    <subtitulo>Un sistema de transmisión de información</subtitulo>
    <parrafo>Sed ut perspiciatis unde omnis iste natus error sit voluptatem
accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo
inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
</parrafo>
  </capitulo>
</contenido>
```

En SGML la forma de definir las etiquetas es la siguiente: se usan "**nombres de elementos**" que se encuentran **delimitados por marcas y que indican el comienzo y final de los elementos u** objetos lógicos.

SGML tiene la característica de ser flexible ya que, por un lado, permite que se definan lenguajes de marcas de forma independiente y, por otro, facilita el intercambio y conservación de documentos y recursos digitales estructurados.

Por lo tanto, SGML puede ser considerado como un **metalenguaje** o *framework* general de descripción de marcado.

Las reglas de comportamiento de estos documentos son definidas en las DTD (descripción de tipo de documento) de SGML. Un ejemplo es el conocido HTML.

Características de los lenguajes de marcas

Las características más llamativas de los lenguajes de marcado son las siguientes.

Información y marcas

Son lenguajes que se **mezclan con los datos y les aportan semántica, formato y estructura**.

Texto plano

Los archivos que contienen lenguajes de marcado son básicamente archivos de texto, por lo que **se pueden editar desde un programa de tratamiento de textos**.

Compacto

Las marcas son escritas en un mismo fichero, por lo que **es fácil acceder a ellas e interpretarlas**.

Procesamiento sencillo

Cualquier intérprete que maneje lenguajes de marcas puede procesarlo, ya que es muy sencillo. Dichas marcas **solo definen datos, estructuras y formato**.

Flexibilidad

Se han podido extender a áreas tales como los gráficos vectoriales, los servicios web o las interfaces de usuario (Android, por ejemplo); esto permite que **en un solo documento se puedan mezclar distintos estándares**.

La estructura de un documento confeccionado con lenguaje de marcado suele dividirse en **cabecera y cuerpo**, y dependiendo del tipo de documento puede subdividirse cada zona en otras.

En concreto un documento HTML tendría:

1. Una parte inicial, donde se indica la versión del estándar HTML. En ella se especifica un DTD sobre el tipo de documento que viene a continuación.

```
<!DOCTYPE HTML PUBLIC"- //W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Ejemplo DTD para HTML 4.

2. La etiqueta <HTML> nos delimita el resto del documento, que tendrá:

- Una cabecera, delimitada por la etiqueta <HEAD>.
- Un cuerpo del documento, delimitado por la etiqueta <BODY>.

Dentro de estas partes se incluyen los siguientes elementos:

- Metadatos.
- Codificación de caracteres.
- Etiquetas o marcas.
- Elementos.
- Atributos.
- Comentarios.

Beneficios de los lenguajes de marcas

- **Permitir el intercambio de documentos** estructurados entre distintos sistemas.
- **Tener una estructura definida**; los documentos necesitan tenerla y que sea establecida con anterioridad.
- **Posibilitar el almacenamiento de la información** estructurada en el sistema como una base de datos.
- Tener un lenguaje **independiente de la representación**.
- Tener un lenguaje **independiente de la plataforma**.
- Tener un lenguaje **abstraído de la información**.

SGML

Estructura

Cuando hablamos de estructura en los lenguajes de marcas lo asociamos a varios conceptos intrínsecos a dichos lenguajes.

- Los documentos de marcas suelen tener **una estructura definida por su naturaleza** y que caracteriza al documento en cuestión. Por ejemplo, los documentos HTML contienen dos partes diferenciadas, que son el encabezado y el cuerpo.
- La estructura de un XML dependerá de cómo se haya **definido su DTD o Schema**.
- Los lenguajes de marcas establecen como condición indispensable que un documento tenga una **estructura lógica**. Eso implica que un documento contiene partes diferenciadas, pero relacionadas de alguna forma.
- Dentro del propio documento existe una **estructura asociada a la apariencia de los contenidos del mismo**. Esta estructura define los componentes físicos, el posicionamiento y la tipografía.

Estructura natural

Los distintos lenguajes de marcado necesitan ser configurados a través de unas definiciones explícitas. Estas definiciones, que en el caso de HTML, XHTML o XML se construyen a través de ficheros DTD y Schemas, nos indican la estructura natural de cada documento generado y sus estructuras lógicas y físicas.

Estas definiciones indican las partes de las que consta un tipo de fichero de un lenguaje de marcado en particular.

```
<?xml version="1.0"?>
<!DOCTYPE MENSAJE SYSTEM "mensaje.dtd">
<mensaje>
  <remite>
    <nombre>Alfredo Reino</nombre>
    <email>alf@ibium.com</email>
  </remite>
  <destinatario>
    <nombre>Bill Clinton</nombre>
    <email>president@whitehouse.gov</email>
  </destinatario>
  <asunto>Hola Bill</asunto>
  <texto>
    <parrafo>¿Hola qué tal? Hace <enfasis>mucho</enfasis> que no escribes. A
ver si llamas y quedamos para tomar algo.</parrafo>
  </texto>
</mensaje>
```

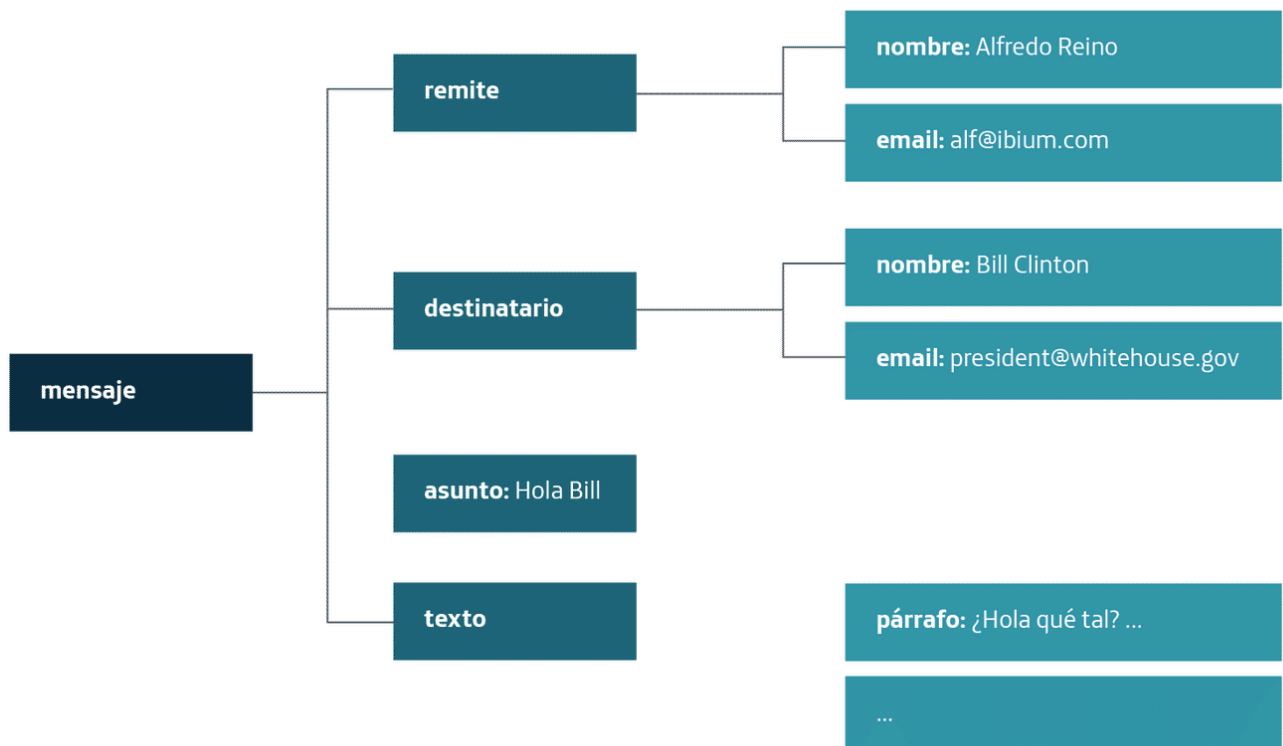
Estructura lógica

La estructura lógica está **formada por las partes que componen el documento y por sus relaciones**.

En el caso de documentos de presentación, la estructura lógica indica el orden de los contenidos y su relación entre ellos.

Por ejemplo, en el caso de una página web podríamos diferenciar entre:

- Una cabecera con información general sobre el contenido de la página, logotipo, título, etc.
- Un menú global de navegación.
- El cuerpo o contenido sustancial de la página.
- El pie de página que hace referencia a información añadida, como el *copyright*.
- Otros elementos, como búsqueda o navegación rápida.



Estructura física

La estructura física indica la **aparición del documento sobre la pantalla**, incluyendo sus componentes físicos, el posicionamiento de los elementos y la tipografía empleada, estilos, imágenes, etc.

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Un Título para el Browser de turno </TITLE>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
</HEAD><BODY>
<H1>Esto es el título. </H1>
<P>Esto es un párrafo con información</P>
<BR>Una línea más escribo en <STRONG> negrita </STRONG>!
<H3>Un subtítulo</H3>
<UL>
<LI>Esto es una lista no ordenada.</LI>
<LI>Las listas quedan mejor si tienen varios elementos.</LI>
</UL>
<P>más información</P>
</BODY>
</HTML>
```

Esto es el título.

Esto es un párrafo con información
Una línea más escribo en **negrita** !

Un subtítulo

- Esto es una lista no ordenada.
- Las listas quedan mejor si tienen varios elementos.

más información

Componentes SGML

Los componentes de un documento SGML son:

La declaración

Indica lo que contendrá el documento en cuanto a caracteres, delimitadores y características opcionales de SGML utilizadas, así como la sintaxis.

La DTD

Indica la estructura del documento y de los elementos que lo van a conformar.

La instancia del documento

Es el documento conformado y validado que contiene la información textual y las etiquetas que lo definen, que están declaradas en la DTD.

Ejemplos

A continuación te mostramos algunos ejemplos de un documento SGML. En ellos podemos ver que constan de una primera sección de código con el DTD, que define las reglas según las que se comportarán las etiquetas dentro del contenido.

```
<!ELEMENT DOCUMENTO - - (titulo, contenido, autor?)>  
<!ELEMENT CONTENIDO - - (capitulo+)>  
<!ELEMENT CAPITULO - (subtitulo, párrafo?)>
```

DTD para definir la estructura del documento

```
<documento>  
  <titulo>HIPERTEXTO</titulo>  
  <autor> MARIA JESUS LAMARCA </autor>  
  <contenido>  
    <capitulo>  
      <subtitulo>  
        EL NUEVO CONCEPTO DE DOCUMENTO EN LA CULTURA DE LA IMAGEN  
      </subtitulo>  
      <párrafo>  
        Doctorado: Fundamentos, Metodología y Aplicaciones de las  
        Tecnologías Documentales y Procesamiento de la Información.  
      </párrafo>  
    </capitulo>  
  </contenido>  
</documento>
```

Contenido del documento.

XML

Conceptos

XML surge del estándar SGML para definir la gramática de lenguajes específicos en la creación y estructuración de documentos grandes. También es usado para generar bases de datos documentales y para el intercambio de datos entre sistemas.

Su desarrollo comenzó en el año 1996, precisamente en el entorno empresarial, donde el HTML se había quedado corto respecto a las necesidades. En 1998 el W3C recoge la recomendación creando el estándar.

XML es un lenguaje estándar que sigue las recomendaciones de W3C y que se creó para poder estructurar datos. No define las etiquetas ni cómo se utilizan, sino un conjunto de reglas sintácticas para poder crear las etiquetas y los documentos. Por lo tanto, no es un lenguaje sino un meta lenguaje, y en la actualidad es el estándar universal para el intercambio de datos. Por otra parte, es independiente del navegador y cada etiqueta realiza siempre la misma función. Con XML se puede jerarquizar la información y estructurar los datos describiendo los contenidos.

Objetivos de XML

1. Servir para el **intercambio de datos en Internet** con cualquier lenguaje de programación.
2. Facilitar la **construcción de documentos estructurados** y de fácil sintaxis.
3. Generar documentos **legibles para el ser humano**.
4. **Generar de forma rápida estructuras de datos** con un lenguaje normalizado y extensible.
5. Ser **independiente del hardware y del software** que lo maneja.

Componentes XML

XML DTD

Define la estructura y componentes de un tipo de documento XML:

```
<!DOCTYPE nota[
  <!ELEMENT nota(to,from,heading,body)>
  <!ELEMENT para (#PCDATA)>
  <!ELEMENT de (#PCDATA)>
  <!ELEMENT asunto (#PCDATA)>
  <!ELEMENT cuerpo (#PCDATA)>
]>
```


XML Schemas

Define la estructura, igual que un DTD, y permite validar un documento.

```
<xs:element name="nota">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="para" type="xs:string"/>
      <xs:element name="de" type="xs:string"/>
      <xs:element name="asunto" type="xs:string"/>
      <xs:element name="cuerpo" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

- **XML Namespace:** especificaciones sobre cómo se conecta una URL con los atributos y *tags* de un XML.
- **DOM:** llamadas a funciones para manejar ficheros XML y HTML.
- **Xlink:** método para enlaces a un fichero XML.
- **XSL:** lenguaje para generación de hojas de estilo.
- **Xpointer:** definición de una sintaxis para la señalización de partes de un documento.

Elementos

Un sistema XML típico consta de tres tipos de archivos:

- **Datos XML y las etiquetas** que describen el significado y la estructura de los datos.
- **Esquemas XML**, que definen las reglas. Por ejemplo, un esquema podría servir para asegurarse de que los usuarios no pueden escribir texto en un campo de fecha.
- **Transformaciones XML**, que permiten usar datos en programas y archivos. Por ejemplo, una transformación podría agregar datos relativos a las compras a un libro de contabilidad, mientras que otra transformación podría insertar la misma información en un programa contable. Un estándar de W3C, como XSTL, nos ayuda a incorporar datos desde el documento hacia otro HTML.

Estructura de un XML

Cabecera

En esta zona se encuentran las **declaraciones XML**, que son instrucciones que identifican el archivo principal y los demás archivos asociados.

```
<?xml version="1.0" encoding="UTF-7"?>
```

Declaraciones XML

Raíz

Elemento **raíz o nodo principal** es el que contiene al resto de nodos y etiquetas, con sus datos para cada registro. También debe contener los atributos que se relacionan con dicho nodo principal.

Es un elemento imprescindible en todo documento XML para que esté bien formado.

```
<Disco>
...
</Disco>
```

Raíz o nodo principal.

Etiquetas

Etiquetas y datos **son los componentes principales** del archivo que cuelgan de cada nodo.

```
<autor>Los Beatles</autor>
<titulo>Let it Be</titulo>
<formato>Wav</formato>
<localizacion>\discos\pop</localizacion>
```

Etiquetas y datos XML.

Atributos

Son otra forma de incluir datos en los nodos del documento, aunque en este caso perdemos la posibilidad de jerarquizar la información.

Los atributos se usan frecuentemente **para identificar un nodo en concreto o transmitir información propia del nodo donde se encuentran**.

```
<autor dni="58585858F" ref="G58">Los Beatles</autor>
<titulo alias="LIT">Let it Be</titulo>
<formato>Wav</formato>
<localizacion>\discos\pop</localizacion>
```

Atributos XML.

Ejemplo:

```
<?xml version="1.0" encoding="UTF-7"?>
<Disco id="F478">
  <autor>Los Beatles</autor>
  <titulo>Let it Be</titulo>
  <formato>Wav</formato>
  <localizacion>\discos\pop</localizacion>
</Disco>
```

XSL

XSLT (*eXtensible Stylesheet Language for Transformations*) es un lenguaje que permite realizar transformaciones a un XML para obtener otro tipo de documento, como un HTML o un documento de texto plano.

La hoja de estilos XSLT no es más que otro XML con las reglas de transformación del documento. La extensión del archivo es *.xsl*.

Las ventajas de utilizar documentos XSLT frente a otro tipo de lenguajes de marcas como CSS son:

1. Se puede **cambiar el orden** de los elementos.
2. Se puede **operar con sus valores**.
3. Se pueden **agrupar sus elementos**.



XSLT permite realizar modificaciones a un XML para obtener otro tipo de documento.

```

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
    <h2>Notas de departamento</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>De</th>
        <th>Para</th>
        <th>Asunto</th>
        <th>Cuerpo</th>
      </tr>
      <xsl:for-each select="ListadoNotas/nota">
        <tr>
          <td><xsl:value-of select="de"/></td>
          <td><xsl:value-of select="para"/></td>
          <td><xsl:value-of select="asunto"/></td>
          <td><xsl:value-of select="cuerpo"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>

</xsl:stylesheet>
  
```

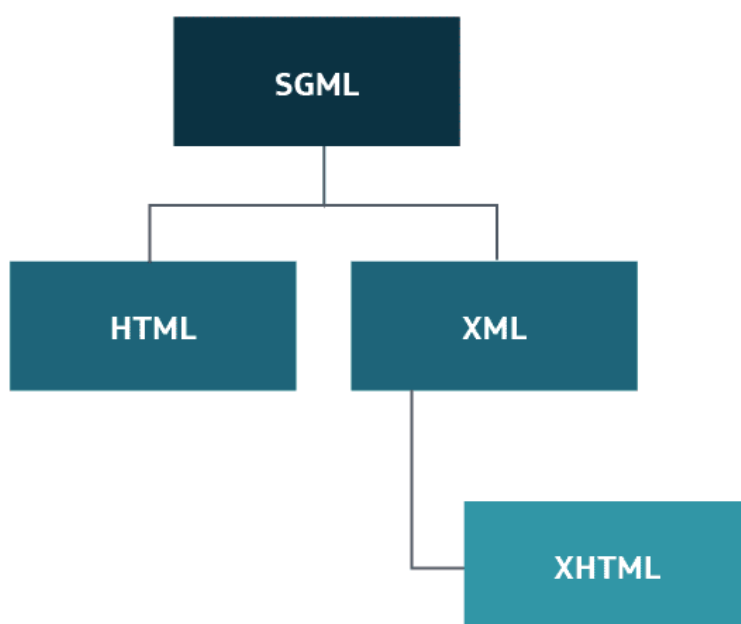
Ejemplo documento XSL.

Despedida

Resumen

Has terminado la lección, veamos los puntos más importantes que hemos tratado.

- Los lenguajes de marcas **se emplean para la transmisión de información**, tanto entre máquina y usuario, como entre diferentes sistemas informáticos.
- **El lenguaje SGML fue el más destacado y ha derivado en los lenguajes XML y HTML**, muy extendidos en los sistemas de información actuales, como la web.



- Con un lenguaje de marcas **podemos construir documentos complejos y/o transferir grandes cantidades de información estructurada** mediante etiquetas y atributos.
- En los lenguajes de marcas **se siguen las reglas especificadas por el proveedor de datos**. Estas reglas se definen en los DTD, que son parte fundamental de un documento de marcado, pues indican las normas para el almacenamiento o mostrado de los datos.