

Unidad1. Lenguaje de marcas

1. Concepto y características generales, ventajas para el tratamiento de la información.
2. Clasificación e identificación de los más relevantes. Utilización en distintos ámbitos.
3. XML, características propias, etiquetas.
4. Herramientas de edición.
5. Elaboración de documentos XML bien formados, estructura y sintaxis.

1.- Concepto y características generales, ventajas para el tratamiento de la información

Estos lenguajes combinan **información textual** con marcas o **anotaciones** relativas a la estructura del texto o a la forma de representarlo.

```
<noticia ambito="local">  
  <lugar >Sevilla</lugar>  
  <fecha>21/09/2021</fecha>  
  <titular>Otro accidente mortal en  
  Avenida Luis Montoto</titular>  
  <descripcion>Un motorista de 31 años ha  
  muerto en la noche de este sábado al  
  chocar con una señal de tráfico.  
  </descripcion>  
</noticia>
```

XML

01prueba.xml

Se diferencian de los lenguajes de programación en que no tienen funciones aritméticas o variables.

02prueba.html

PostScript

```
newpath
% Inicio cursor
100 100 moveto
% Dibujo rectángulo
300 100 lineto
300 250 lineto
100 250 lineto
100 100 lineto
stroke
```

<section>

```
<h2>Otro accidente mortal en Avenida
  Luis Montoto</h2>
<p>Sevilla</p>
<p>21/09/2021</p>
<div>Un motorista de 31 años ha muerto
  en la noche de este sábado al
  chocar con una señal de
  tráfico.</div>
</section>
```

HTML

DocBook (basado en SGML/XML)

```
<article lang="es">  
  <title>Documentación Empleo de maquinaria</title>  
  <articleinfo>  
    <author><firstname>Susana</firstname>  
      <surname>Martínez</surname></author>  
  </articleinfo>  
  <section>  
  
    </section>  
</article>
```

La declaración de tipo de documento.

Todo lenguaje de marcas está definido en un documento denominado DTD (*Document Type Definition*) donde se definen las marcas, los elementos utilizados por dicho lenguaje y sus correspondientes etiquetas y atributos. Es decir, su **sintaxis**.

Se debe indicar esta información al principio, es lo que se conoce como la **declaración del documento**.

En el caso de HTML5, siempre es `<!DOCTYPE html>`

En XML, por ejemplo, podría ser

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

* Lo veremos con detalle más adelante.

Ventajas de los lenguajes de marcas.

Comunicación de datos. Si la información se transfiere en un lenguaje de marcas, cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en ese lenguaje y otra aplicación recibir esta información y trabajar con ella.

Migración de datos. Si tenemos que mover los datos de una base de datos a otra sería muy sencillo si las dos trabajasen en un mismo formato.

Portabilidad.

Reutilización.

Adaptabilidad.

Existen editores avanzados

Desventajas de los lenguajes de marcas.

- Complejidad.
- Diseño lento. Un **número excesivo de etiquetas** puede dificultar el **mantenimiento** o corrección.
- Lenguaje **estático**.
- La **interpretación de cada navegador** o herramienta de visualización puede ser distinta.

Evolución de los lenguajes de marcas. SGML.HTML.XML

GML (*Lenguaje de marcas generalizado, IBM*)

Incluir anotaciones en los documentos electrónicos como se hacía en el papel

Finales de los 60

SGML (*lenguaje estándar ISO 8879*)

Consiguió que se compartiera información entre sistemas informáticos pero requería de un *software* muy complejo.

HTML (*Hipertexto*)

En el CERN crearon un lenguaje de marcado que permitiera compartir información en las redes de ordenadores y posteriormente Internet.

Finales de los 80

<https://www.w3.org/TR/html/>

Se extendió con mucha rapidez debido a la sencillez de sus sintaxis y del software necesario para interpretarlo.

Tim Berners-Lee combinó el
ASCII y el SGML

Presentaba algunos inconvenientes:

- La estructura y el diseño están mezclados en el documento.
- No permitía contenido dinámico.
- El número de etiquetas limitaban la flexibilidad.

Las etiquetas semánticas nos indican cual es el contenido que contienen, en lugar de cómo se debe formatear



XML (*Extendido*)

<https://www.w3.org/TR/xml/>

La W3C intenta dotar a la *Web* de un lenguaje potente con una **estructura semántica**.

Mediados de los 90

El nuevo lenguaje de marcas extendido sería más sencillo que SGML y más potente que HTML.

No incluye ninguna información relativa al diseño, convirtiéndose rápidamente en el estándar para el intercambio de datos en la Web.

XML realmente es un conjunto de estándares relacionados entre sí:

- XSL. Hojas de estilo.
- XML *Linking Language*. Define enlaces entre documentos.
- XML *Namespaces*. Define contextos de actuación.
- XML *Schemas*. Define restricciones. Los más usados son las DTD.

- Lenguajes **procedurales o de procedimiento**. Orientados también a la presentación. El programa que representa el documento debe interpretar el código en el mismo orden en que aparece.

Por ejemplo, TeX, LaTeX o Postscript .

Lenguaje de descripción de páginas muy usado por las impresoras en talleres gráficos profesionales

La mayoría de los documentos científicos o libros técnicos con fórmulas matemáticas se escriben en LaTeX.

si $x=0$ entonces $y^2=4p+7$

se vería...

Si $x=0$ entonces $y^2=4p+7$

- Lenguajes **descriptivos o semánticos**. Describen las diferentes partes en las que se estructura el documento pero sin especificar cómo deben representarse. Las marcas sólo indican que es lo que se está representando. Son los más empleados.

Una cuestión muy importante es que todos los documentos codificados en XML puedan ser tratados como bases de datos.

Por ejemplo, SGML y sus derivados, HTML, XML, XHTML, etc.

Uno de los lenguajes basados en XML es el formato COLLADA que se emplea para definir escenas de modelos tridimensionales.

```
<?xml version="1.0"?>
```

```
<contact-info>
```

```
  <contact1>
```

```
    <name>Tanmay Patil</name>
```

```
    <company>TutorialsPoint</company>
```

```
    <phone>(011) 123-4567</phone>
```

```
  </contact1>
```

```
  <contact2>
```

```
    <name>Manisha Patil</name>
```

```
    <company>TutorialsPoint</company>
```

```
    <phone>(011) 789-4567</phone>
```

```
  </contact2>
```

```
</contact-info>
```

03baseDeDatos.xml

base de datos XML

Formada por registros
con campos que
contienen datos de dos
empleados.



Clasificación de los lenguajes de marcas según ámbito de aplicación.

- Documentación electrónica.
 - RTF (*rich text format*).
 - TeX. Ecuaciones matemáticas complejas.
 - Wikitexto.
 - DocBook.
- Tecnología de Internet.
 - HTML,XHTML. Creación de páginas web.
 - RSS.
- Intercambio de información entre diversos sistemas.
 - VoiceXML.
 - MusicXML.

3.- XML, características propias, etiquetas.

Etiqueta (*tag*). Texto que va entre los símbolos < y >. Las hay de inicio y de fin (entre </ y >)


Elemento. Estructura que nos permitirá organizar el contenido del documento cuando se interprete el mismo. Constan de:

- Etiqueta de inicio.
- Etiqueta final
- Todo lo que haya entre ambas.

Atributo. Es un par *nombre="valor"* que puede estar dentro de la etiqueta de inicio de un elemento indicando propiedades concretas.

Etiqueta de apertura

Etiqueta de cierre



```
<noticia ambito="local">  
  <lugar>Sevilla</lugar>  
  <fecha>21/09/2021</fecha>  
  <titular>Otro accidente mortal en Avenida Luis  
    Montoto</titular>  
  <descripcion>Un motorista de 31 años ha muerto en la  
    noche de este sábado al chocar con una señal de  
    tráfico.</descripcion>  
</noticia>
```

NOTICIA es un elemento que contiene cuatro elementos hijos en su interior además de un atributo que indica el ámbito de la noticia.

Puede haber elementos con contenido mixto.

```
<persona>
```

```
    <nombre>Rubén</nombre> vive en <ciudad>Salamanca</ciudad>
```

```
</persona>
```

- Todos los nombres de etiquetas **distinguen mayúsculas de minúsculas**.
- El primer carácter tiene que ser una letra o un guion bajo “_”.
- Puede incluir letras minúsculas, letras mayúsculas, números, puntos “.”, guiones medios “-” y guiones bajos “_”.
- Puede contener los dos puntos (:) pero se reserva para definir espacios de nombres.

Detrás del nombre de una etiqueta se permite escribir un espacio en blanco o un salto de línea.



```
<ciudad >Pamplona</ciudad  
>
```

El proceso de creación de un documento XML pasa por la siguientes etapas:

- Especificación de requisitos.
- Diseño de etiquetas.
- Marcado de los documentos.

Todo documento XML tiene dos partes: **prólogo** (opcional) y el **ejemplar** (obligatorio)

Podemos incluir comentarios usando la siguiente sintaxis:

<!-- Texto del comentario -->

Podemos poner comentarios donde queramos **menos antes del prólogo y dentro de una etiqueta.**

El prólogo.

Debe preceder al ejemplar del documento para facilitar el procesado.

Se divide en dos partes:

1. Declaración XML. Si se incluye es la primera línea siempre.

`<?xml atributos ?>`

Puede tener hasta tres atributos muy concretos (si se ponen debe ir en este orden) :

- *version*, para indicar la versión.
- *encoding*, para indicar la codificación de caracteres.
- *standalone*, para indicar la autonomía del documento, es decir, si necesita de otro para su interpretación.

`<?xml version="1.0" encoding="iso-8859-1" standalone="no" ?>`
`encoding="utf-8"`

El prólogo.

2. La declaración del tipo de documento.

<!DOCTYPE nombre del elemento raíz**>**

Por ejemplo, **<!DOCTYPE** cuadros**>**

<?xml *version="1.0" encoding="iso-8859-1" standalone="no" ?>*

<!DOCTYPE biblioteca**>**

El ejemplar.

Contiene los datos reales del documento. Formada por elementos anidados.
Por ejemplo,

```
<biblioteca>
```

```
  <libro>
```

```
    <titulo>La búsqueda incansable </titulo>
```

```
    <autor>Sebastián Torres</autor>
```

```
    <autor>Anna Sánchez</autor>
```

```
    <editorial>Ediciones Plum</editorial>
```

```
    <isbn>978-2-7460-4344-1</isbn>
```

```
    <edicion>2</edicion>
```

```
    <paginas>540</paginas>
```

```
  </libro>
```

```
</biblioteca>
```

04biblioteca.xml

El ejemplar.

- Sólo puede haber un *elemento raíz*.
- Todos los elementos deben tener etiqueta de inicio y de cierre. En el caso de ser elementos vacíos se podrían sustituir las dos etiquetas por una sola del estilo *<elemento/>*.
- No pueden encadenarse elementos. Todos anidados.
- Los nombres de las etiquetas de inicio y cierre deben ser idénticos y no pueden tener espacios ni empezar por “:” ni por la cadena xml.

Los caracteres especiales

05mediosDeTransporte.xml

Carácter	Cadena
>	>
<	<

Carácter	Cadena
&	&
"	"

Carácter	Cadena
'	'

El ejemplar.

Los atributos. Dan información extra sobre la etiqueta.

06atributos.xml

- Permiten añadir propiedades a los elementos.
- No siguen ninguna jerarquía, es decir, no pueden contener otros elementos o atributos.
- No aportan ninguna estructura lógica.
- Los valores deben ir entre comillas simples o dobles.

```
<fechaPrestamo dia= "11" mes= "marzo" año="2019"/>
```

```
<fechaPrestamo dia= "11" mes= "marzo" año="2019"/>
```

```
<fechaDevolucion dia= "30" mes= "marzo" año="2019"/>
```

El ejemplar.

```
<producto codigo="G45">  
  <nombre color="negro" precio="10.99">Sombrero</nombre>  
</producto>
```

Si, por ejemplo, el atributo *codigo* se quisiera representar como un elemento, se podría escribir:

```
<producto>  
  <codigo>G45</codigo>  
  <nombre color="negro" precio="10.99">Sombrero</nombre>  
</producto>
```


Características generales de XML

- Es compatible con protocolos que ya funcionan, como HTTPS.
- Todo documento que verifique las reglas de XML está conforme con SGML.
- El marcado de XML es legible para los humanos.
- El diseño XML es formal y conciso.
- XML es extensible, adaptable y aplicable a una gran variedad de situaciones.
- Todo documento XML se compone exclusivamente de datos de marcado y caracteres (contenido) entremezclados.

4.- Herramientas de edición.

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 para construir un documento XML. Podría emplearse el bloc de notas o cualquier editor de texto plano.

Para crear documentos completos pueden emplearse **editores XML** especializados o **editores avanzados**. Todo esto facilitará la edición y mantenimiento del código.

<https://geekflare.com/es/best-xml-editors/>

Nosotros, empezaremos con el editor avanzando Visual Studio Code.

Procesadores XML

Para interpretar el código XML se puede usar cualquier navegador ya que incluyen lo necesario para acceder a su contenido y su estructura.

Uno de los elementos que los navegadores llevan incorporados son el *parser* o analizador XML que se encarga de comprobar se cumplan todas las normas establecidas.

Para publicar un documento XML en Internet se emplean los procesadores XSLT que generarán archivos HTML a partir de un documento XML

Validadores XML

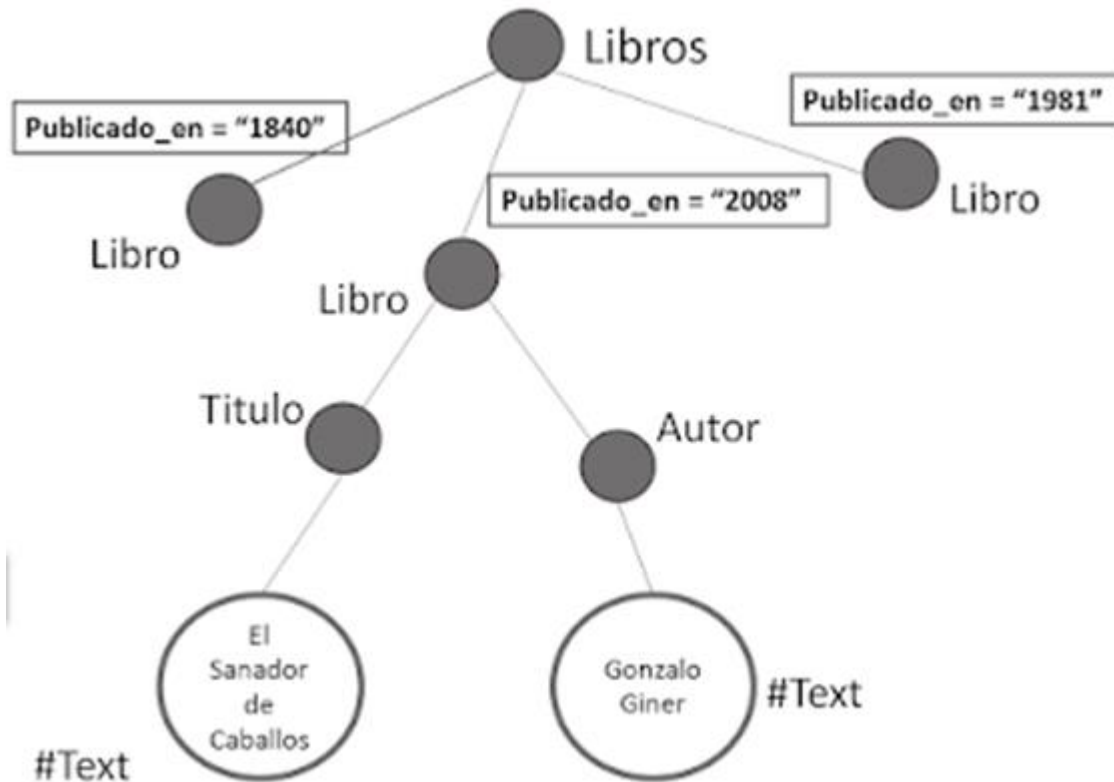
https://www.w3schools.com/xml/xml_validator.asp

<https://validator.aborla.net/index.php5?lang=es>

XML Copy Editor. <https://xml-copy-editor.sourceforge.io/>

Editor avanzado Visual Studio Code. <https://code.visualstudio.com/>

Árbol DOM



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

```
<!DOCTYPE libros>
```

```
<libros>
```

07libros.xml

```
<libro publicadoEn="1840">
```

```
<titulo>El Capote</titulo>
```

```
<autor>Nikolai Gogol</autor>
```

```
</libro>
```

```
<libro publicadoEn="2008">
```

```
<titulo>El Sanador de Caballos</titulo>
```

```
<autor>Gonzalo Giner</autor>
```

```
</libro>
```

```
<libro publicadoEn="1981">
```

```
<titulo>El Nombre de la Rosa</titulo>
```

```
<autor>Umberto Eco</autor>
```

```
</libro>
```

```
</libros>
```

5.- Elaboración de documentos XML bien formados, estructura y sintaxis.

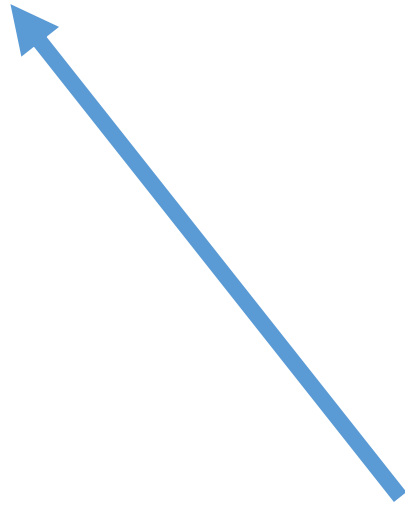
Documentos bien formados.

Son sintácticamente correctos. Cumplen las reglas de sintaxis del lenguaje.

- Documento debe tener definido un prólogo con la declaración completa.
- Existe un único elemento raíz donde el resto de elementos y contenidos están anidados.
- Cumplir reglas de sintaxis del lenguaje XML.

Documentos válidos.

Además de bien formados cumplen los requisitos de la definición de estructura que se haya indicado en la definición del documento.



* Más adelante veremos estas definiciones de estructuras.