

# ÍNDICE

## XML

1. DEFINICIÓN DE XML
2. ESTRUCTURA DE UN DOCUMENTO XML
3. SINTAXIS GENERAL DE XML
4. CÓMO AGREGAR ATRIBUTOS
5. ESPACIOS DE NOMBRES
  - 5.1. UTILIZACIÓN DE ESPACIOS DE NOMBRES EN DOCUMENTOS

### 1. DEFINICIÓN DE XML

- ❑ **XML** es un *metalenguaje*, es decir, un lenguaje que permite la creación de otros lenguajes, sus gramáticas y vocabularios. Utiliza las marcas como método de transmisión de metadatos (información sobre un conjunto de datos). Los lenguajes de marcas usan etiquetas para delimitar y describir los datos.
- No debe pensarse que XML es sólo un metalenguaje para crear páginas web. XML sirve para el etiquetado y la estructuración de documentación. Es tan potente y flexible que muchos personas lo utilizan para otros fines. Algunas de estas otras aplicaciones las podemos englobar en estos tres grupos:
  1. Uso de XML para **describir metacontenidos** respecto a documentos o recursos en línea.
  2. Uso de XML para publicar e **intercambiar contenidos de bases de datos**.
  3. Uso de XML como formato para sistemas de **mensajería** con el fin de permitir la comunicación entre programas de aplicaciones.

## 2. ESTRUCTURA DE UN DOCUMENTO XML

- ❑ Todo documento XML se divide en tres partes fundamentales: **prólogo**, **cuerpo** y **epílogo**.

Prólogo	<pre>&lt;?xml version="1.0"?&gt; &lt;!--aquí pueden ir comentarios e instrucciones de procesamiento--&gt; &lt;!DOCTYPE libros SYSTEM <a href="http://miservidor.com/libros.dtd">http://miservidor.com/libros.dtd</a>&gt; &lt;!--aquí pueden ir más comentarios e instrucciones--&gt;</pre>
Cuerpo	<pre>&lt;libros&gt;   &lt;libro categoria="ficción"&gt;     &lt;autor&gt;J.R.R. Tolkien&lt;/autor&gt;     &lt;título&gt;El Señor de los Anillos&lt;/título&gt;     &lt;precio&gt;5000&lt;/precio&gt;   &lt;/libro&gt;   &lt;libro categoria="filosofía"&gt;     &lt;autor&gt;Schopenhauer&lt;/autor&gt;     &lt;título&gt;El arte de tener razón&lt;/título&gt;     &lt;precio&gt;2000&lt;/precio&gt;   &lt;/libro&gt; &lt;/libros&gt;</pre>
Epílogo	<pre>&lt;!--aquí pueden ir comentarios e instrucciones--&gt;</pre>

### 2. ESTRUCTURA DE UN DOCUMENTO XML

- ❑ **Prólogo:** Aunque no es obligatorio, los documentos XML pueden empezar con una serie de líneas que identifican al documento y proporcionan información sobre los archivos externos relacionados con él.
- La *declaración XML* `<?xml>`, es la forma por la que el SW identifica un documento XML. Se trata de una sentencia que aparece siempre como la primera línea del archivo XML, que especifica qué versión de XML se utiliza en el documento. En la declaración XML se puede especificar la codificación de caracteres utilizada en el documento, que para uso con lenguajes europeos (incluyendo el juego de caracteres especiales del castellano), se utilizará `UTF-7` o `ISO-8859-1`.
- Si el documento XML es completamente autosuficiente y no obtiene información de archivos externos como hojas de estilo o definiciones de tipo de documento (DTD), entonces el documento XML se considera autosuficiente (`standalone`).

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

- Si para el documento XML se ha creado una declaración de tipo de documento (DTD), que valida y define la información contenida en el archivo XML, es necesario especificarlo en el prólogo del documento.

```
<!DOCTYPE MENSAJE SYSTEM "mensaje.dtd">
```

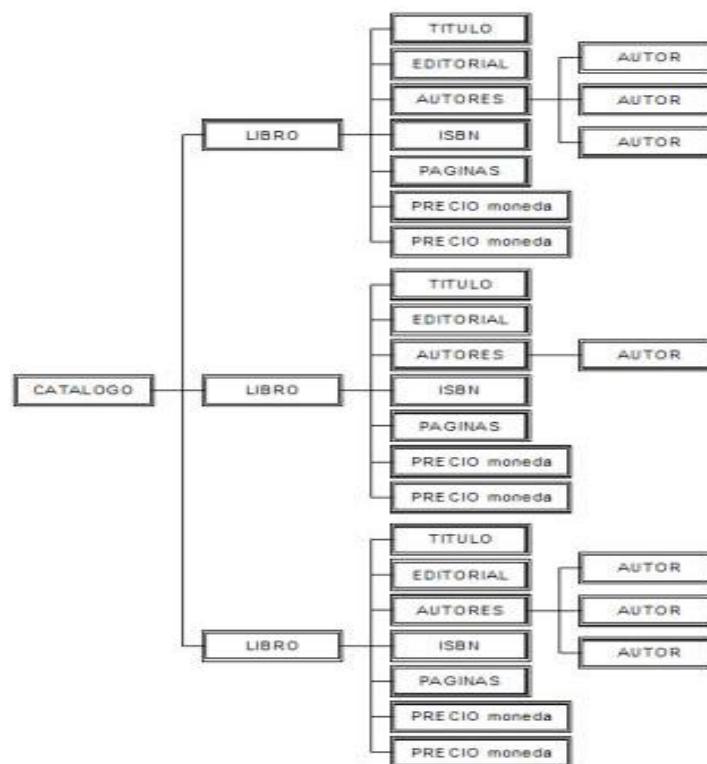
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final/EN" >
```

```
<!DOCTYPE LABEL SYSTEM "http://www.empresa.com/dtds/label.dtd">
```

### 2. ESTRUCTURA DE UN DOCUMENTO XML

- ❑ **Cuerpo:** El cuerpo del documento XML contiene uno o más elementos en forma de árbol jerárquico que pueden contener datos. Los documentos XML poseen un único elemento raíz que debe servir como base en la que se incluye todo el resto del documento.
- Se puede considerar que los elementos proporcionan información adicional acerca del contenido del documento. En otras palabras, se puede utilizar la expresividad de los nombres de elemento de XML para definir su contenido.

```
<CATALOGO>
<LIBRO>
<TITULO>MASTERING XML</TITULO>
<EDITORIAL>SYBEX</EDITORIAL>
<AUTORES>
<AUTOR>ANN NAVARRO</AUTOR>
<AUTOR>CHUCK WHITE</AUTOR>
<AUTOR>LINDA BURMAN</AUTOR>
</AUTORES>
<ISBN>0782122663</ISBN>
<PAGINAS>888</PAGINAS>
<PRECIO MONEDA ="PESETA">10.425</PRECIO>
<PRECIO MONEDA ="EURO">62,66</PRECIO>
</LIBRO>
.....
</CATALOGO>
```



## 2. ESTRUCTURA DE UN DOCUMENTO XML

- ❑ **Epílogo:** El epílogo es opcional y utilizado normalmente para situar comentarios e instrucciones de procesamiento.
  
- ❑ **Tipos de documentos:** Existen dos tipos de documentos XML: bien formados y válidos. La diferencia que existe entre los dos es la siguiente:
  - **Bien formados:** son todos los que cumplen las especificaciones del lenguaje respecto a las reglas sintácticas, sin estar sujetos a unos elementos fijados en una DTD. De hecho, los documentos XML deben tener una estructura jerárquica muy estricta y los documentos bien formados deben cumplirla.
  - **Válidos:** Además de estar bien formados, siguen una estructura y una semántica determinada por una DTD específica. Sus elementos, atributos y sobre todo la estructura jerárquica que define la DTD, debe ser respetada por el documento XML.

### 3. SINTAXIS GENERAL DE XML

- ❑ **Nombres de elemento** : A medida que se prepara un documento XML, es necesario tener en cuenta las diferentes reglas que existen para nombrar elementos. Los nombres de elemento de XML distinguen mayúsculas de minúsculas y no pueden contener espacios en blanco.
  - Todo nombre de elemento debe comenzar por: una letra o bien guión bajo ‘\_’ o ‘:’ siempre y cuando el principio no contenga la palabra ‘xml’ en cualquier combinación posible de mayúsculas y minúsculas.
- ❑ **Etiquetas de elemento de comienzo y de terminación**: La mayor parte de los elementos deben tener una etiqueta de comienzo y otra de terminación. Dichas etiquetas marcan la ubicación dentro del documento para colocar la información. Si los elementos no están escritos correctamente, el documento no estará bien formado. Si el documento no está bien formado, no se puede analizar adecuadamente.  
`<NOMBRE_DE_ELEMENTO>Aquí va el contenido. </NOMBRE_DE_ELEMENTO>`
- XML requiere que todos los elementos se cierren de forma apropiada utilizando una etiqueta de finalización o una etiqueta vacía con el formato adecuado.
- ❑ **Contenido de un elemento**: Los elementos pueden contener referencias a datos binarios (en forma de archivos externos) o datos de caracteres (*referencias a entidades o texto estándar*).

### 3. SINTAXIS GENERAL DE XML

- ❑ Las **referencias a entidades** son códigos que utilizan una serie de caracteres para dar instrucciones al visor de XML para que muestre un carácter que, en otro caso, estaría reservado para otros aspectos del documento XML. XML sólo admite cinco referencias a entidades.
- Los **datos de caracteres normales** (CDATA, *Regular Character Data*) permite especificar datos, utilizando cualquier carácter, especial o no, sin que se interprete como marcado XML. Existe una excepción y es la cadena "]]>" con el que termina el bloque CDATA. Esta cadena no puede utilizarse dentro de una sección CDATA.

Ejemplo primero usando entidades predefinidas y luego con un bloque CDATA

```
<parrafo>Lo siguiente es un ejemplo de HTML.</parrafo>
```

```
<ejemplo> &lt;html&gt; &lt;head&gt;
&lt;title&gt;Rock &amp; Roll&lt;/title&gt;
&lt;/head&gt;
</ejemplo>
```

```
<ejemplo> <![CDATA[ <html> <head>
<title>Rock & Roll</title>
</head> ]]>
</ejemplo>
```

REFERENCIA A ENTIDAD	CARÁCTER
&amp;	&
&apos;	'
&gt;	>
&lt;	<
&quot;	"

- Una sección CDATA se identifica con frecuencia utilizando el marcado `<![CDATA[ y ]]>`. **Es necesario encerrar los datos de tipo carácter entre estos elementos de marcado** para que el analizador sintáctico ignore todos los caracteres.



### 3. SINTAXIS GENERAL DE XML

- ❑ **Etiquetas de elementos vacíos:** A veces se da la circunstancia de que se desea crear un elemento sin contenido. Los elementos vacíos pueden contener atributos, o pueden servir para contener otro tipo de información. Las etiquetas de elementos vacíos incluyen una barra inclinada de cierre (/) dentro del elemento de apertura. El analizador sintáctico de XML reconoce el elemento vacío y pasa al siguiente elemento del documento.

```
<NOMBRE_DE_ELEMENTO_VACÍO/>
```

- ❑ **Elementos secundarios:** Igual que ocurre en HTML, es posible anidar elementos de XML. Pero, al contrario que HTML, que permite el anidamiento errático de elementos, los elementos de XML no se pueden superponer. Un documento XML bien formado requiere que las etiquetas de elementos de XML se apilen de forma ordenada.
- Los elementos vacíos pueden aparecer en cualquier parte. Pueden aparecer entre otras etiquetas de elemento o dentro de su contenido.

```
<DOCUMENTO>
<TITULO> Mi primer documento XML </TITULO>
<TEXT01> Hola Mundo! </TEXT01>
<TEXT02> Es sencillo crear documentos ajustados a las normas de XML. </TEXT02>
<FIN/>
</DOCUMENTO>
```

### 4. CÓMO AGREGAR ATRIBUTOS

- ❑ **Identificación de atributos** : Los atributos proporcionan más información acerca del contenido de los datos de un elemento XML determinado, agregando detalles adicionales o información auxiliar. Como los atributos sólo proporcionan información auxiliar o explicativa acerca del contenido de un elemento o del propio elemento, no es necesario especificarlos en una sentencia completa. Se les incrusta en la etiqueta de inicio del propio elemento.

```
<AREA_DE_CONOCIMIENTO codigo="570">Lenguajes</AREA_DE_CONOCIMIENTO>
```

- ❑ **Cuándo utilizar atributos** : En realidad, los atributos se pueden utilizar para mostrar toda la información que, de otro modo, aparecería dentro de los elementos secundarios. Depende de cada uno decidir qué información se coloca en los atributos y cuál en los elementos secundarios.

```
<AREA_DE_CONOCIMIENTO>
```

```
<CODIGO>570</CODIGO>
```

```
<TITULO> Lenguajes y Sistemas Informáticos. </TITULO>
```

```
<AREA_DE_CONOCIMIENTO/>
```

- Hay que tener en cuenta que, en general, se pone información en un atributo si se trata estrictamente de información auxiliar que no necesita utilizar ningún formato concreto para mostrarse en el documento final. Cuando se decide qué información es estrictamente auxiliar, hay que tener presente las siguientes reglas de elementos y atributos:
  - Los elementos permiten almacenar metadatos como atributos.
  - Los elementos son más sencillos de extender en el futuro.
  - Los atributos no poseen estructura.
  - Es necesario utilizar transformaciones como XSLT para mostrar atributos en los navegadores convencionales.

### 4. CÓMO AGREGAR ATRIBUTOS

- La cuestión fundamental acerca de la utilización de atributos es su ambigüedad y falta de extensibilidad. Al colocar mucha información como atributos se limita drásticamente la facilidad de uso de los datos y la cantidad de datos que se pueden agregar a un documento.
- Se elimina el propósito de incluir metadatos en los elementos y los atributos se convierten en el contenido.
- Debido a la libertad que proporciona XML, no hay por qué irse al extremo de utilizar solamente elementos ni al extremo opuesto de utilizar sólo atributos. Resulta sencillo encontrar el término medio entre ambos que permita la flexibilidad y extensibilidad necesarias dentro del documento.
- ❑ **Nombres de atributos:** Existen muy pocas restricciones para conseguir que la utilización de atributos forme parte de un documento bien formado. Los atributos que contienen errores y no siguen dichas reglas romperán el documento XML y provocarán la aparición de errores sintácticos. Los atributos deben seguir las siguientes reglas:
  - Los nombres de atributo pueden contener letras, números, puntos o guiones de subrayado. Se puede utilizar el símbolo “:” pero está reservado para su utilización con espacios de nombres.
  - Los valores de atributo deben aparecer entre comillas.
  - No es posible tener varios atributos con el mismo nombre en un único elemento.

### 4. CÓMO AGREGAR ATRIBUTOS

- ❑ **Valores de atributos** : Un aspecto importante de la utilización de atributos es la identificación de su valor. Se dispone de muchas opciones a la hora de identificar los valores de los atributos:
  - **Los valores pueden pertenecer a un conjunto especificado.** Pueden formar parte de una lista predefinida. Para imponer que los valores de un atributo formen parte de una lista específica es necesario proporcionar una DTD o un esquema de documento que contenga las reglas que utilizará la aplicación de XML para validarlos.
  - **Los valores pueden hacer referencia a otro archivo.** Piénsese en el atributo `src` del elemento `<IMG>` de HTML. XML puede hacer lo mismo con entidades cargadas adecuadamente.
  - **Los valores pueden ser de formato libre.** Además de hacer referencia a una lista específica o a un objeto binario como un archivo de imagen, los valores pueden tener un formato libre. Por ejemplo el atributo `alt` del elemento `<IMG>` de HTML.
- Hasta ahora se han utilizado atributos con elementos normales que disponen de marcadores de apertura y cierre. También se pueden utilizar atributos con elementos vacíos. Puede que incluso la utilización de atributos con elementos vacíos sea más importante, ya que sirven como medio para proporcionar toda la referencia y contenido de ese elemento.
- La utilización de atributos no está limitada a un único elemento. Se pueden utilizar atributos con el mismo nombre en diferentes elementos.

### 4. CÓMO AGREGAR ATRIBUTOS

- ❑ **Utilización de hojas de estilo con atributos :** Las hojas de estilo proporcionan un medio para configurar un documento XML de forma que tenga el aspecto deseado. La sentencia que permite a XML utilizar hojas de estilo es `<?xml-stylesheet?>`. Esta instrucción de procesamiento indica al SW de navegación, así como al analizador sintáctico, dónde encontrar una hoja de estilo para utilizarla con el documento XML.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="libros.xsl" ?>
<!DOCTYPE LIBROS SYSTEM "libros.dtd">
<LIBROS>
  <LIBRO>
    <TITULO>AutoSketch</TITULO>
    <AUTOR>Ramón Montero</AUTOR>
    <PRECIO>2.500</PRECIO>
  </LIBRO>
  <LIBRO>
    <TITULO>Windows 98</TITULO>
    <AUTOR>Jaime de Yraolagoitia</AUTOR>
    <PRECIO>3.250</PRECIO>
  </LIBRO>
  <LIBRO>
    <TITULO>Web Graphics</TITULO>
    <AUTOR>Ron Wodaski</AUTOR>
    <PRECIO>8.975</PRECIO>
  </LIBRO>
</LIBROS>
```

## 4. CÓMO AGREGAR ATRIBUTOS

- En esta instrucción de procesamiento existen dos pseudo atributos. El primer atributo, **type**, proporciona información acerca del tipo del medio MIME específico que se utiliza para identificar la hoja de estilo, en este caso una hoja de estilo XSL. El segundo atributo, **href**, se utiliza para identificar el nombre y la ubicación de la hoja de estilo que se va a utilizar.
- La instrucción de procesamiento `<?xml-stylesheet?>` debe seguir siempre a la sentencia `<?xml?>` y aparecer antes del elemento raíz del documento XML. Si se está utilizando una DTD, la instrucción de procesamiento de la hoja de estilo debe aparecer antes de la sentencia `<!DOCTYPE>`. Al contrario que en los documentos HTML, no es posible incrustar hojas de estilo dentro de documentos XML, sólo se pueden vincular. Esto, por supuesto, resulta útil para promocionar la utilización de hojas de estilo con varios documentos, que era una de sus intenciones originales.

### 5. ESPACIOS DE NOMBRES

- ❑ A medida que se trabaja con XML, llega un momento en que se desea incluir en un solo documento información que se encuentra en dos o más de ellos. Existen tres opciones: **(i)** no validar el documento y utilizar XML bien formado para ambos documentos, **(ii)** validarlo y volver a escribir la DTD de los dos conjuntos de marcas o **(iii)** utilizar espacios de nombres de XML (*XML Namespaces*).
- El problema aparece cuando existen elementos con el mismo nombre. El compilador, o agente de usuario, no reconocerá a qué DTD pertenece la definición de cada ocurrencia del elemento.
- Los **espacios de nombres** de XML, se han creado para tener la posibilidad de utilizar varios elementos con el mismo nombre dentro del mismo documento. Este sistema permite adjuntar una URL al nombre de un elemento para proporcionar al procesador de XML la posibilidad de relacionar los diferentes elementos con la fuente correcta. Esto es válido tanto para nombres de elemento como para nombres de atributo.

### 5.1. UTILIZACIÓN DE ESPACIOS DE NOMBRES EN DOCUMENTOS

- ❑ Cuando se desea declarar un espacio de nombres de XML en el documento actual se puede utilizar la notación siguiente:

```
<nombre_elemento xmlns:prefijo="url_del_espacio_de_nombres">
```

- Como se puede apreciar, es posible especificar un nombre de elemento o de atributo después del prefijo utilizado para identificar el espacio de nombres individual. El prefijo de espacio de nombres se utiliza, a continuación, con el prefijo `xmlns` para identificar el origen de la DTD utilizada para identificar por completo este atributo universal.
- No se pueden utilizar los prefijos `xml` ni `xmlns` para unir un espacio de nombres con un documento. El prefijo `xml` señala, de manera predeterminada, a la dirección `http://www.w3.org/xml/1998/namespace` del propio documento XML. El prefijo `xmlns` se utiliza para ligar elementos con espacios de nombres y, por tanto, no se puede utilizar como prefijo.
- No es necesario declarar un espacio de nombres para cada elemento secundario si ya se ha hecho para el elemento primario. El espacio de nombres se aplicará de forma automática al elemento secundario y a todos sus atributos a menos que se especifique otra cosa.
- **Nombres completos o calificados** (*qualified names*), son aquellos que combinan el prefijo y el nombre de la parte local, por ejemplo, `<mdoc:QUESTION>` o `<math:MI>`. Si se califica un nombre de elemento, entonces hay que aplicar el prefijo tanto a la etiqueta de comienzo como a la de finalización. Si se califica un atributo dentro de un elemento, basta con aplicar el prefijo al nombre del atributo.



### 5.1. UTILIZACIÓN DE ESPACIOS DE NOMBRES EN DOCUMENTOS

- Un ejemplo sería una instancia XML que contuviera referencias a un cliente y a un producto solicitado por éste. Tanto el elemento que representa el cliente como el que representa el producto pueden tener un elemento hijo llamado "numero\_ID". Las referencias al elemento "numero\_ID" podrían ser ambiguas, salvo que los elementos, con igual nombre pero significado distintos, se llevaran a espacios de nombres distintos que los diferenciaran.

```
<?xmlversion="1.0"?>
<cli:cliente      xmlns:cli='http://www.dominio.org/Espacio_de_nombres_XML/cliente'
xmlns:ped='http://www.dominio1.org/Espacio_de_nombres_XML/pedido'>
<cli:numero_ID>1232654</cli:numero_ID>
<cli:nombre>Fulanito de Tal</cli:nombre>
<cli:telefono>99999999</cli:telefono>
<ped:pedido>
<ped:numero_ID>6523213</ped:numero_ID>
<ped:articulo>Caja de herramientas</ped:articulo>
<ped:precio>187,90</ped:precio>
</ped:pedido>
</cli:cliente>
```

- Un espacio de nombres se declara usando el atributo XML reservado xmlns, cuyo valor debe ser un identificador uniforme de recurso. Por ejemplo:

```
xmlns="http://www.w3.org/1999/xhtml"
```