

# UD1.3 Descripción de XML

## Introducción

XML (*eXtensible Markup Language*) es un lenguaje de etiquetas o un lenguaje de marcado que **estructura** y guarda de forma ordenada la **información**. No representa datos por sí mismo, solamente organiza la estructura. En XML, las etiquetas son creadas por el programador.

XML **ahorra tiempos de desarrollo** y proporciona **ventajas**, dotando a webs y a aplicaciones de una forma realmente potente de guardar la información. Además, se ha convertido en un formato universal que ha sido asimilado por todo tipo de sistemas operativos y dispositivos móviles.

**Para ampliar conocimientos sobre XML:**

- [Tutorial de XML](#) de W3C Schools
- [Introducción a XML](#) de Mozilla Developer Network (MDN)

## Características del documento XML

Un documento XML es un **documento de texto** que tiene las siguientes características:

- La extensión de fichero es `.xml`.
- Está compuesto de parejas de etiquetas estructuradas en árbol, que describen una función en la organización del documento.
- Puede editarse con cualquier editor de texto.
- Es interpretado por los navegadores web.

## Características del lenguaje XML

Las características básicas de XML son:

- Dado que XML se concibió para trabajar en la web, es directamente compatible con protocolos que ya funcionan (como HTTP y HTTPS).
- Todo documento que verifique las reglas de XML está conforme con SGML.
- No se requieren conocimientos de programación para realizar tareas sencillas en XML.
- Los documentos XML son fáciles de crear.

- La difusión de los documentos XML está asegurada, ya que cualquier procesador de XML puede leer un documento de XML.
- El marcado de XML es legible para los humanos (*human readable*).
- El diseño XML es formal y conciso.
- XML es extensible, adaptable y aplicable a una gran variedad de situaciones.
- XML es orientado a objetos.
- Todo documento XML se compone exclusivamente de datos de marcado y datos carácter entremezclados.

El **proceso de creación** de un documento XML pasa por varias etapas, en las que el éxito de cada una de ellas se basa en la calidad de la anterior. Estas etapas son:

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

## Marcado

El **marcado** en XML son etiquetas que se añaden a un texto para estructurar el contenido del documento. Esta información extra permite a los ordenadores *interpretar* los textos. El marcado es todo lo que se sitúa entre:

- `<` y `>` en el caso de elementos.
- `&` y `;` en el caso de entidades.

Más adelante veremos qué son los elementos y las entidades.

### EJEMPLO DE MARCADO: ELEMENTO

```
<body>
```

### EJEMPLO DE MARCADO: ENTIDAD

```
&gt;
```

Los datos que se encuentran **encerrados entre marcas** son los que forman la **verdadera información** del documento XML.

Por ejemplo, en el siguiente documento, el **dato** es `12.39`:

```
<precio>12.39</precio>
```

El marcado puede ser tan rico como se quiera. Puede ser interesante detectar necesidades futuras y crear documentos con una estructura fácilmente actualizables

## Comentarios

Los documentos XML pueden tener **comentarios**, que se utilizan para incluir información para el desarrollador, ya que éstos **no son interpretados** por el interprete XML.

Se incluyen entre las cadenas `<!--` y `-->`.

```
<!-- Ejemplo de comentario -->
```

Pueden estar en cualquier posición en el documento excepto:

- Antes del prólogo.
- Dentro de una etiqueta.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Comentario correcto -->
<b>
  <c cantidad="4">cccc<!-- Comentario correcto --></c>
  <d cantidad="2">dd</d><!-- Comentario correcto -->
</b>
```

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

<b>
  <c cantidad="4" <!-- Comentario incorrecto -->>cccc</c>
  <d cantidad="2">dd</d>
</b>
```

Los comentarios no pueden contener dos guiones medios seguidos (`--`).

```
<!-- Comentario incorrecto -- (contiene dos guiones medios seguidos
dentro del comentario) -->
```

# Estructura de un documento XML

Los documentos XML están formados por las siguientes partes:

- Un **prólogo** (opcional)
- Un **ejemplar** (obligatorio)

En el siguiente ejemplo, las dos primeras líneas se corresponden con el prólogo, mientras que el resto se corresponden con el ejemplar:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE libro>
<libro>
  <titulo>XML practico</titulo>
  <autor>Sebastien Lecomte</autor>
  <autor>Thierry Boulanger</autor>
  <editorial>Ediciones Eni</editorial>
  <isbn>978-2-7460-4958-1</isbn>
  <edicion>1</edicion>
  <paginas>347</paginas>
</libro>
```

## Prólogo

El prólogo es una parte **opcional** de un documento XML que, si se incluye, debe preceder al ejemplar. Su inclusión facilita el procesamiento de la información del ejemplar.

El **prólogo** está dividido en dos partes:

- La declaración XML.
- La declaración del tipo de documento.

## Declaración XML

En el caso de incluirse, ha de ser **la primera línea del documento**. De no ser así, se genera un error que impide que el documento sea procesado. El hecho de que sea opcional, permite el procesamiento de documentos HTML y SGML como si fueran XML. Si fuera obligatoria, éstos deberían incluir una declaración de versión XML que no tienen.

El prólogo puede tener tres funciones:

1. Declaración la versión de XML usada para elaborar el documento.

2. Declaración de la codificación empleada para representar los caracteres.
3. Declaración de la autonomía del documento.

De base, una declaración XML tiene el siguiente aspecto:

```
<?xml ?>
```

## Versión de XML

La declaración de la versión de XML usada para elaborar el documento se indica con el atributo `version`:

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

En este caso, se indica que el documento fue creado para la versión `1.0` de XML.

## Codificación de caracteres

La declaración de la codificación empleada para representar los caracteres se indica con el atributo `encoding`:

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

En este caso, se indica que el conjunto de caracteres que se utiliza en el documento es UTF-8, el cual permite el uso de acentos o caracteres como la ñ.

### Estándares de codificación de caracteres:

Estándar ISO	Código de país
UTF-8 (Unicode)	Conjunto de caracteres universal
ISO-8859-1 (Latin-1)	Europa occidental, Latinoamérica
ISO-8859-2 (Latin-2)	Europa central y oriental
ISO-8859-3 (Latin-3)	Sudoeste de Europa
ISO-8859-4 (Latin-4)	Países Escandinavos, Bálticos
ISO-8859-5	Cirílico

Estándar ISO	Código de país
ISO-8859-6	Árabe
ISO-8859-7	Griego
ISO-8859-8	Hebreo
ISO-8859-9	Turco
ISO-8859-10	Lapón. Nórdico, esquimal

## Autonomía del documento

La declaración de la autonomía del documento informa de si el documento necesita de otro para su interpretación. Para esto, se utiliza el atributo `standalone`:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
```

En este caso, se indica que el documento es independiente. De no ser así, el atributo `standalone` hubiese tomado el valor `no`.

Para declarar el atributo `standalone`, hay que definir el prólogo completo.

## Declaración del tipo de documento

Define qué tipo de documento estamos creando con la finalidad de ser procesado correctamente. Toda declaración de tipo de documento comienza por `<!DOCTYPE` seguido del nombre del tipo y el caracter `>`. Por ejemplo:

```
<!DOCTYPE html>
```

## Ejemplar

Es la **parte principal** de un **documento XML**, ya que contiene los datos reales del documento. Está formado por elementos anidados.

## Elementos

Los **elementos** son los distintos bloques de información que permiten definir la estructura de un documento XML. Por ejemplo:

```
<libro>XML práctico</libro>
```

Los elementos están delimitados por una etiqueta de **apertura** y una etiqueta de **cierre**. En el ejemplo mostrado:

- Etiqueta de apertura: `<libro>`
- Etiqueta de cierre: `</libro>`

A su vez, los elementos pueden estar formados por otros **elementos** y/o por **atributos**. Por ejemplo:

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

```
<!DOCTYPE libro>
```

```
<libro>
```

```
<titulo>XML practico</titulo>
```

```
<autor>Sebastien Lecomte</autor>
```

```
<autor>Thierry Boulanger</autor>
```

```
<editorial>Ediciones Eni</editorial>
```

```
<isbn>978-2-7460-4958-1</isbn>
```

```
<edicion>1</edicion>
```

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

```
<libro>
```

En este caso, el ejemplar es el elemento `<libro>`. A su vez, `<libro>` está compuesto de los elementos:

- `<autor>`
- `<editorial>`
- `<isbn>`
- `<edicion>`
- `<paginas>`

En realidad, el ejemplar es el **elemento raíz** (*root*) de un documento XML (en este ejemplo, el elemento `<libro>`). Todos los datos de un documento XML han de pertenecer a un

elemento del mismo.

Los **nombres** de las etiquetas han de ser **autodescriptivos**, lo que facilita el trabajo que se hace con ellas.

## Sintaxis

La formación de elementos ha de cumplir ciertas normas para que queden perfectamente definidos y que el documento XML al que pertenecen pueda ser interpretado por los procesadores XML sin generar ningún error fatal.

A continuación, se describen las reglas de sintaxis del lenguaje XML.

### Único elemento raíz

En todo documento XML debe existir **un elemento raíz** (*root element*), y sólo uno.

Por ejemplo, el siguiente ejemplo sería un XML no válido (tiene dos elementos raíz):

```
<libro>XML Práctico</libro>
```

```
<cuaderno>Lenguaje de marcas</libro>
```

El hecho de que tenga un elemento raíz, permite representar los datos en forma de árbol.

Supongamos el siguiente documento XML:

```
<persona>
```

```
<nombre>Elsa</nombre>
```

```
<mujer/>
```

```
<fecha-de-nacimiento>
```

```
<día>18</día>
```

```
<mes>6</mes>
```

```
<año>1996</año>
```

```
</fecha-de-nacimiento>
```



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

```
</persona>
```

Se podría representar gráficamente de la siguiente manera:

### Probar en el navegador

Documento XML

## Etiqueta de apertura y cierre

Todos los elementos tienen una etiqueta de inicio y otra de cierre. Por ejemplo:

```
<libro>XML Práctico</libro>
```

Lo que sería incorrecto es lo siguiente (falta la etiqueta de cierre):

```
<libro>XML Práctico
```

En el caso de que en el documento existan **elementos vacíos** (*empty elements*), se pueden sustituir las etiquetas de inicio y cierre por una de elemento vacío. Ésta se construye como la etiqueta de inicio, pero sustituyendo el carácter `>` por `/>`. Por ejemplo, supongamos el siguiente elemento:

```
<libro></libro>
```

Sería equivalente a:

```
<libro/>
```

## Anidación de elementos

Al **anidar elementos** (introducir unos dentro de otros), hay que tener en cuenta que no puede cerrarse un elemento que contenga algún otro elemento que aún no se haya cerrado.

Por ejemplo, esto es incorrecto (se cierra el elemento `libro` antes de `titulo`):

```
<libro>
```

```
<titulo>XML Práctico</libro>
```

```
</titulo>
```

Lo correcto es:

```
<libro>
```

```
<titulo>XML Práctico</titulo>
```

```
</libro>
```

## Nomenclatura de etiquetas

Los nombres de las etiquetas de **inicio** y de **cierre** de un mismo elemento han de ser **idénticos, respetando las mayúsculas y minúsculas**. Por ejemplo:

```
<cuaderno></cuaderno>
```

Sin embargo, no sería válido lo siguiente:

```
<Libro></libro>
```

Las normas de sintaxis básicas en relación a los nombres de etiquetas son:

- Todos los nombres de los elementos son sensibles a letras minúsculas y mayúsculas (*case sensitive*).
- Pueden contener letras minúsculas, letras mayúsculas, números, puntos (.), guiones medios (-) y guiones bajos (\_).
- Asimismo, pueden contener el carácter dos puntos (:). Pero, su uso se reserva para definir espacios de nombres.
- El primer carácter tiene que ser una letra o un guion bajo (\_).
- Las letras no inglesas (á, Á, ñ, Ñ, etc.) están permitidas. Pero, al igual que el carácter guion medio (-) y el punto (.), se recomienda no utilizarlos para reducir posibles incompatibilidades o errores en programas que no los interpreten bien.
- No puede comenzar por la cadena xml, ni ninguna de sus versiones en que se cambien mayúsculas y minúsculas (XML, Xml, xML, etc.).

## Contenido de los elementos

No se pueden utilizar directamente los caracteres >, <, &, " y ' en el contenido de los elementos, ya que son caracteres reservados.

### Uso de comillas simples y dobles

En el caso de las comillas simples (') y dobles ("), existe un caso donde sí se pueden utilizar: en **atributos**. Aunque respetando ciertas restricciones.

En el caso de tener que utilizar alguno de estos caracteres, se sustituyen por las siguientes cadenas:

Caracter	Entidad	Representación
>	gt	&gt;
<	lt	&lt;
&	amp	&amp;
"	quot	&quot;
'	apos	&apos;

Al nombre que hace referencia a un caracter se le denomina **entidad**. Por ejemplo, la entidad `gt` hace referencia al caracter `>`.

Cuando utilizamos una entidad en un documento XML debe hacerse mediante su representación (ver tabla), es decir, el nombre de la entidad entre los símbolos `&` y `;`. Por ejemplo:

```
<libro>Elemento &lt;libro&lt;</libro>
```

El contenido del elemento `libro` sería:

```
Elemento <libro>
```

## Caracteres especiales

Para utilizar caracteres especiales (como £, ©, ®, etc.) hay que usar las expresiones `&#D;` o `&#H;`, donde `D` y `H` se corresponden respectivamente con el número decimal o hexadecimal asociado al caracter que se quiere representar en el **código UNICODE**.

Por ejemplo, para incluir €, se usarían las cadenas `&#8364;` o `&#x20AC;`.

### Caracteres ASCII

En el siguiente enlace encontrarás una tabla con los caracteres ASCII, el nombre HTML, y el número HTML de cada uno de ellos que te será imprescindible a la hora de realizar documentos en HTML y XML.

<http://ascii.cl/es/codigos-html.htm>

