https://www.mclibre.org/consultar/xml/lecciones/xml-conceptos-basicos.html

Ejemplo de documento XML

Un documento XML es similar a una página web, salvo que los nombres de las etiquetas y atributos no son los del HTML.

```
<?xml version="1.0" encoding="UTF-8"?>
<paises>
  <pais>
    <nombre>España</nombre>
    <capital>Madrid</capital>
  </pais>
  <pais>
    <nombre>Francia</nombre>
    <capital>París</capital>
  </pais>
</paises>
```

Por supuesto, la misma información se puede guardar con otra estructura:

```
<?xml version="1.0" encoding="UTF-8"?>
<paises>
  <pais nombre="España" capital="Madrid" />
  <pais nombre="Francia" capital="París" />
</paises>
```

Conceptos y vocabulario

Documento XML
Un documento XML es un documento de texto plano (sin formato).
Procesador XML (*XML processor*) y aplicación (*application*)
Cuando una aplicación necesita leer un documento XML, la aplicación recurre a un procesador XML. El procesador XML (o analizador XML, en inglés *XML parser*) es el que lee el documento, analiza el contenido y le pasa la información en un formato estructurado a la aplicación. La recomendación XML especifica lo que debe hacer el procesador, pero no entra en lo que hace después la aplicación con esa información.

Caracteres (*characters*)
Los documentos XML pueden estar codificados en distintos juegos de caracteres (UTF-8, ISO-8859-1, etc).

Marcas (*mark-up*) y contenido (*content*)
El texto que contiene un documento XML se divide en marcas y contenido. Las marcas pueden ser de dos tipos: etiquetas o referencias a entidades. Todo lo que no son marcas es contenido.

Elementos (*elements*)
Un elemento es un componente lógico de un documento que o bien comienza por una etiqueta de apertura y termina por la etiqueta de cierre correspondiente o que consiste en una única etiqueta vacía. El contenido de un elemento es todo lo que se encuentra entre las etiquetas de apertura y cierre, incluso si estos son también elementos en cuyo caso se llaman elementos hijos.

Etiquetas (*tags*)
Una etiqueta es un identificador que empieza por el carácter < y termina por >. Existen tres tipos de etiquetas:

- las etiquetas de apertura (*start-tag*), que empiezan por el carácter < y terminan por >. Por ejemplo:

  ```
  <apartado>
  ```

- las etiquetas de cierre (*end-tag*), que empiezan por los caracteres </ y terminan por >. Por ejemplo:

  ```
  </apartado>
  ```

- las etiquetas vacías (*empty tag*), que empiezan con el carácter < y terminan por />. Por ejemplo:

  ```
  <linea />
  ```

Atributos (*attributes*)
Un atributo es un componente de las etiquetas que consiste en una pareja nombre (*name*) / valor (*value*). Se puede encontrar en las etiquetas de apertura o en las etiquetas vacías, pero no en las de cierre. En una etiqueta no puede haber dos atributos con el mismo nombre. La sintaxis es siempre nombreAtributo="valorAtributo". Por ejemplo:

```
<profesor nombre="Bartolomé" apellidos="Sintes Marco" />
```

Comentarios (*comments*)
Un comentario es una etiqueta que comienza por . Los comentarios no pueden estar dentro de otras marcas y no pueden contener los caracteres "--". Dentro de un comentario las entidades de carácter no se reconocen, es decir, sólo se pueden utilizar los caracteres del juego de caracteres del documento. Por ejemplo:

```
<!-- Esto es un comentario -->
```

Declaración XML (*XML declaration*)
La declaración XML es una etiqueta que comienza por y que proporciona información sobre el propio documento XML. Aunque no es obligatoria es conveniente que aparezca, y debe aparecer siempre al principio del documento. No es una instrucción de procesamiento, pero tiene la misma sintaxis (empieza por).

La declaración xml indica el juego de caracteres del documento. El juego de caracteres que se utiliza en este curso es UTF-8:

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

En XML, el nombre del juego de caracteres se debe escribir en mayúsculas (UTF-8 en vez de utf-8).
La declaración xml también indica la versión XML utilizada. Aunque existe la versión XML 1.1, la versión más común sigue siendo la versión XML 1.0.
Se pueden utilizar otros juegos de caracteres, como ISO-8859-1 (Europeo occidental):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Es importante que el juego de caracteres que aparece en la declaración sea el juego de caracteres en que realmente está guardado el documento, porque si no el procesador XML puede tener problemas leyendo el documento.

Correcto

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Este texto está guardado como UTF-8 y declara que es UTF-8 -->
<prueba>
  <texto>á é í ó ú</texto>
</prueba>
```

Incorrecto

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Este texto está guardado como UTF-8 sin BOM y declara que es ISO-8859-1 -->
<prueba>
  <texto>á é í ó ú</texto>
</prueba>
```

Correcto

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Este texto está guardado como ISO-8859-1 y declara que es ISO-8859-1 -->
<prueba>
  <texto>á é í ó ú</texto>
</prueba>
```

Incorrecto

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Este texto está guardado como ISO-8859-1 y declara que es UTF-8 -->
<prueba>
  <texto>á é í ó ú</texto>
</prueba>
```

Nota:
En el caso del documento guardado como UTF-8 pero que declara ser ISO-8859-1, también le afecta que haya o no marca de orden de bytes (BOM).

La marca de orden de bytes son los primeros caracteres de un fichero UTF. Estos caracteres indican:

- el orden de los bytes en los formatos de 16 o 32 bytes. Cuando la codificación de un carácter ocupa más de un byte, los bytes que lo forman pueden estar en el orden natural de lectura (big endian) o en el contrario (little endian). Por ejemplo el carácter A, cuyo código es 0041, se puede guardar en UTF-16 como 00 41 (big endian) o como 41 00 (little endian).
- el formato del archivo en general. Para saber si un archivo está en formato UTF-8, 16 o 32 (es decir, si cada byte del archivo es un carácter, o los bytes se tienen que tomar de dos en dos, o de cuatro en cuatro), los programas sólo tienen que mirar los primeros bytes del archivo y reconocer la marca correspondiente.

En el caso de los archivos UTF-8, la marca de orden de bytes son los tres caracteres EF BB BF (que corresponden a los caracteres ï»¿). Pero como en un archivo UTF-8 no se necesita indicar el orden de los bytes, puesto que cada carácter ocupa solamente un byte, la marca es opcional, es decir, que hay archivos UTF-8 con BOM y archivos UTF-8 sin BOM. Los editores (como Eclipse PDT) no muestran estos caracteres y, suelen respetarlos al guardar de nuevo al archivo, pero otros (como el Bloc de notas de Windows) guardan los archivos UTF-8 siempre con BOM. Para ver esos caracteres, se puede utilizar un editor hexadecimal, que muestra todos los caracteres sin excepción.

El ejemplo siguiente muestra como los navegadores son capaces de mostrar correctamente el contenido si el archivo está guardado como UTF-8 con BOM. El motivo es que aunque el archivo declara que está guardado como ISO-8859-1, el navegador detecta el BOM y deduce que se trata de un archivo UTF-8:

Correcto

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Este texto está guardado como UTF-8 con BOM y declara que es ISO-8859-1 -->
<prueba>
  <texto>á é í ó ú</texto>
</prueba>
```

El ejemplo siguiente muestra como los navegadores no son capaces de mostrar correctamente el contenido si el archivo está guardado como UTF-8 sin BOM. El archivo se muestra como si fuera ISO-8859-1 ya que el archivo declara que está guardado como ISO-8859-1 y el navegador no detecta un BOM que le indique que realmente está guardado como UTF-8:

Incorrecto

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Este texto está guardado como UTF-8 sin BOM y declara que es ISO-8859-1 -->
<prueba>
  <texto>á é í ó ú</texto>
</prueba>
```

Definición de tipo de documento (DTD, *Document Type Definition*)
Una DTD es un documento que define la estructura de un documento XML: los elementos, atributos, entidades, notaciones, etc, que pueden aparecer, el orden y el número de veces que pueden aparecer, cuáles pueden ser hijos de cuáles, etc. El procesador XML la utiliza para verificar si un documento es válido, es decir, si el documento cumple las reglas del DTD.

Declaración de tipo de documento (DOCTYPE, *Document type declaration*)
Una declaración de tipo de documento es una etiqueta que comienza por y que indica la(s) DTD(s) que debe utilizar el procesador XML para validar el documento. La DTD puede estar incluida en el propio documento o ser un documento externo. Por ejemplo, el siguiente ejemplo muestra la declaración de tipo de documento que se debe incluir en los documentos XHTML 1.0 de tipo strict (en este caso, la DTD es un documento externo):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Instrucciones de procesamiento (PI, *processing instruction*)
Una instrucción de procesamiento en una etiqueta que empieza por "" y que contiene instrucciones dirigidas a las aplicaciones que leen el documento. Pueden aparecer en cualquier lugar del documento. Por ejemplo:

```
<?xml-stylesheet type="text/xsl" href="estilo.xsl" ?>
```

Entidades de carácter
En XML se utilizan varias entidades de carácter de HTML, para poder escribir en cadenas de texto los caracteres que delimitan las marcas o las cadenas de texto :

Referencia a entidad	Carácter
<	<
>	>
&	&
'	'

Referencia a entidad	Carácter
"	"

Secciones CDATA (CDATA *section*)

Una sección CDATA es una etiqueta que comienza por y cuyo contenido el procesador XML no interpreta como marcas sino como texto. Es decir, que si aparecen los caracteres especiales (< & " ') en una sección CDATA, el procesador XML no interpreta que empieza una marca, sino que lo considera un carácter más. Se suele utilizar en documentos en los que aparecen muchas veces esos caracteres especiales para no tener que estar utilizando las referencias a entidades (< y &) que dificultan la lectura del documento.

Incorrecto

```
<?xml version="1.0" encoding="UTF-8"?>
<prueba>
  <texto>Los caracteres < y & no pueden escribirse
  si no es como comienzo de marcas</texto>
</prueba>
```

Correcto

```
<?xml version="1.0" encoding="UTF-8"?>
<prueba>
  <texto>Los caracteres &lt; y &amp; no pueden escribirse
  si no es como comienzo de marcas</texto>
</prueba>
```

Correcto

```
<?xml version="1.0" encoding="UTF-8"?>
<prueba>
  <texto><![CDATA["Los caracteres < y & no pueden escribirse
  si no es como comienzo de marcas"]]></texto>
</prueba>
```

Referencias a entidades

En XML se pueden definir entidades y utilizarlas como se utilizan las entidades de carácter en el HTML. Una entidad se define mediante una etiqueta que comienza por y contiene el nombre y el valor de la entidad. Por ejemplo:

Como las entidades de carácter del HTML, para hacer referencia a una entidad se escribe sin espacios intermedios el carácter "&", el nombre de la entidad y el carácter ";". Al abrir el documento XML el procesador sustituye la referencia a la entidad por su valor. Por ejemplo:

Correcto

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE autor [
  <!ENTITY yo "Bartolomé Sintes Marco">
  <!ELEMENT autor (#PCDATA)>
]>
<autor>&yo;</autor>
```

Documentos bien formados

Un documento XML debe estar bien formado, es decir debe cumplir las reglas de sintaxis de la recomendación XML. Para que un documento esté bien formado, al menos debe cumplir los siguientes puntos:

- El documento contiene únicamente caracteres Unicode válidos.
- Hay un elemento raíz que contiene al resto de elementos.
- Los nombres de los elementos y de sus atributos no contienen espacios.
- El primer carácter de un nombre de elemento o de atributo puede ser una letra, dos puntos (:) o subrayado (_).
- El resto de caracteres pueden ser también números, guiones (-) o puntos (.).
- Los caracteres "<" y "&" sólo se utilizan como comienzo de marcas.
- Las etiquetas de apertura, de cierre y vacías están correctamente anidadas (no se solapan) y no falta ni sobra ninguna etiqueta de apertura o cierre.

- Las etiquetas de cierre coinciden con las de apertura (incluso en el uso de mayúsculas y minúsculas).
- Las etiquetas de cierre no contienen atributos.
- Ninguna etiqueta tiene dos atributos con el mismo nombre.
- Todos los atributos tienen algún valor.
- Los valores de los atributos están entre comillas (simples o dobles).
- No existen referencias en los valores de los atributos.

Si un documento XML no está bien formado, no es un documento XML. Los procesadores XML deben rechazar cualquier documento que contenga errores.

Documentos válidos

Un documento XML bien formado puede ser válido. Para ser válido, un documento XML debe:

- incluir una referencia a una gramática,
- incluir únicamente elementos y atributos definidos en la gramática,
- cumplir las reglas gramaticales definidas en la gramática.

Existen varios formas de definir una gramática para documentos XML, las más empleadas son :

- DTD (Document Type Definition = Definición de Tipo de Documento). Es el modelo más antiguo, heredado del SGML.
- XML Schema. Es un modelo creado por el W3C como sucesor de las DTDs.
- Relax NG. Es un modelo creado por OASIS, más sencillo que XML Schema.

Última modificación de esta página: 1 de mayo de 2018