1. Introducción

El leguaje **XML**, o **Lenguaje de Etiquetas Extendido**, es un lenguaje de etiquetas, creadas por el programador, estas estructuran y guardan de forma ordenada la información. No representa datos por sí mismo, solamente organiza la estructura.

El 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.

Al igual que en **HTML** un documento **XML** es un documento de texto, en este caso con extensión **".xml"**, compuesto de parejas de etiquetas, estructuradas en árbol, que describen una función en la organización del documento, que puede editarse con cualquier editor de texto y que es interpretado por los navegadores Web.

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 los **URL**.
- 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.
- 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.

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 los caracteres "<" y ">" o "&" y ";"

Los **datos carácter** son los que forman la verdadera información del documento **XML**.

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 *no son interpretados* por el intérprete **XML**. Estos se incluyen entre las cadenas "<!-" y "-->", pueden estar en cualquier posición en el documento salvo:

- Antes del prólogo.
- Dentro de una etiqueta.

Los documentos **XML** pueden estar formados por una *parte opcional llamada prólogo* y otra parte *obligatoria llamada ejemplar*.

2. El prólogo

Si se incluye, el prólogo **debe preceder al ejemplar del documento**. Su inclusión facilita el procesado de la información del ejemplar. El prólogo está dividido en **dos partes**:

- **La 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:
  - *Declaración la versión de XML usada para elaborar el documento.*
    Para ello se utiliza la etiqueta:

    ```
    <?xml versión= “1.0” ?>
    ```

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

- *Declaración de la codificación empleada para representar los caracteres.*
  Determina el conjunto de caracteres que se utiliza en el documento. Para ello se escribe:

  ```
  <?xml versión= “1.0” encoding=”iso-8859-1” ?>
  ```

  En este caso se usa el código `iso-8859-1` (`Latin-1`) que permite el uso de acentos o caracteres como la ñ.
- *Declaración de la autonomía del documento.*
  Informa de si el documento necesita de otro para su interpretación. Para declararlo hay que definir el prólogo completo:

  ```
  <?xml versión= “1.0” encoding=”iso-8859-1” standalone=”yes” ?>
  ```

  En este caso, el documento es independiente, de no ser así el atributo standalone hubiese tomado el valor "no".

- **La declaración del tipo de documento**, define qué tipo de documento estamos creando para ser procesado correctamente. Toda declaración de tipo de documento comienza por la cadena:

  ```
  <!DOCTYPE Nombre_tipo ...>
  ```

Los códigos más importantes son:

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
ISO -8859-6	Árabe
ISO -8859-7	Griego
ISO -8859-8	Hebreo
ISO -8859-9	Turco
ISO-8859-10	Lapón. Nórdico, esquimal
EUC-JP oder Shitf_JIS	Japonés

3. El ejemplar. Los elementos.

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

Los elementos son los distintos bloques de información que permiten definir la estructura de un documento **XML**. Están delimitados por una etiqueta de apertura y una etiqueta de cierre. A su vez, los elementos pueden estar formados por otros elementos y/o por atributos.

Ejemplo

Dado el siguiente código XML...

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

El ejemplar es el elemento < libro >, que a su vez está compuesto de los elementos < autor >, < editorial >, < isbn >, < edicion > y < paginas >

En realidad, **el ejemplar es el elemento raíz (root) de** un documento XML. **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.

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. Dichas reglas son:

- En todo documento **XML** debe existir un elemento raíz, y sólo uno.
- Todos los elementos tienen una etiqueta de inicio y otra de cierre. En el caso de que en el documento existan elementos vacíos, 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 "/>. Es decir, puede sustituirse por:
- Al anidar elementos hay que tener en cuenta que no puede cerrarse un elemento que contenga algún otro elemento que aún no se haya cerrado.
- Los nombres de las etiquetas de inicio y de cierre de un mismo elemento han de ser idénticos, respetando las mayúsculas y minúsculas. Pueden ser cualquier *cadena alfanumérica* que *no contenga espacios* y *no comience ni por el carácter dos puntos*, ":", *ni por la cadena "xml"* ni ninguna de sus versiones en que se cambien mayúsculas y minúsculas ("XML", "XmL", "xML",...).
- El contenido de los elementos no puede contener la cadena "]]>" por compatibilidad con SGML. Además no se pueden utilizar directamente los caracteres mayor que, >, menor que, <, ampersand, &, dobles comillas, ", y apostrofe, ‘. En el caso de tener que utilizar estos caracteres se sustituyen por las siguientes cadenas:

Carácter	Cadena
>	& gt;
<	& lt;
&	& amp;
"	& quot;
'	& apos;

- Para utilizar caracteres especiales, como £, ©, ®,... hay que usar las expresiones &#D; o &#H; donde D y H se corresponden respectivamente con el número decimal o hexadecimal correspondiente al carácter que se quiere representar en el código **UNICODE**. Por ejemplo, para incluir el carácter de Euro, €, se usarían las cadenas € o €

Debes conocer

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.

ASCII imprescindible en HTML y XML

3.1. Atributos

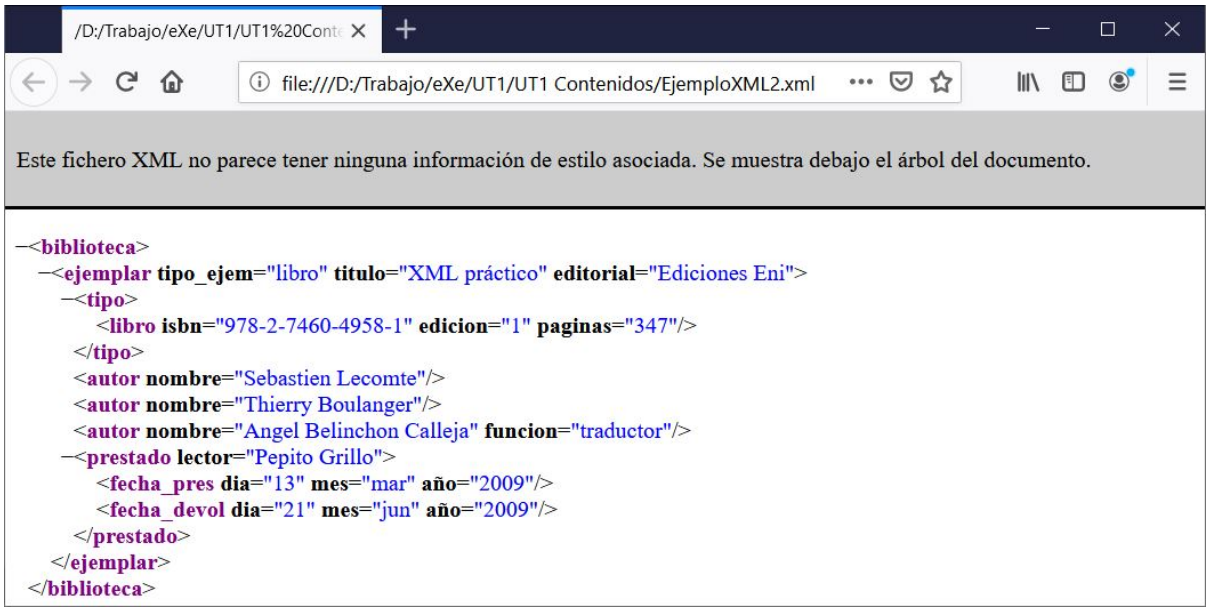
Permiten añadir propiedades a los elementos de un documento. Los atributos no pueden organizarse en ninguna jerarquía, no pueden contener ningún otro elemento o atributo y no reflejan ninguna estructura lógica.

No se debe utilizar un atributo para contener información susceptible de ser dividido.

Ejemplo: Dado el siguiente código XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE biblioteca >
<biblioteca>
    <ejemplar tipo_ejem="libro" titulo="XML práctico" editorial="Ediciones Eni">
        <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paginas="347"></libro> </tipo>
        <autor nombre="Sebastien Lecomte"></autor>
        <autor nombre="Thierry Boulanger"></autor>
        <autor nombre="Angel Belinchon Calleja" funcion="traductor"></autor>
        <prestado lector="Pepito Grillo">
            <fecha_pres dia="13" mes="mar" año="2009"></fecha_pres>
            <fecha_devol dia="21" mes="jun" año="2009"></fecha_devol>
        </prestado>
    </ejemplar>
</biblioteca>
```

Al abrir el documento anterior con el navegador obtendremos:



Los **nombres de los elementos aparecen en color morado**, los **atributos en negro** y sus **valores en azul**.

Como se observa en el ejemplo, **los atributos se definen y dan valor dentro de una etiqueta de inicio** o de elemento vacio, a continuación del nombre del elemento o de la definición de otro atributo, siempre separado de ellos por un espacio. Los valores del atributo van precedidos de un igual que sigue al nombre del mismo y tienen que definirse entre comillas simples o dobles.

Los nombres de los atributos han de cumplir las mismas reglas que los de los elementos, y no pueden contener el carácter menor que, <.

4. Documentos XML bien formados

Todos los documentos **XML** deben verificar las reglas sintácticas que define la recomendación del **W3C** para el estándar **XML**. Esas normas básicas son:

- El documento ha de tener definido un prólogo con la declaración xml completa (no es obligatorio, es una recomendación).
- Existe un único elemento raíz para cada documento: es un solo elemento en el que todos los demás elementos y contenidos se encuentran anidados.
- Hay que cumplir las reglas sintácticas del lenguaje **XML** para definir los distintos elementos y atributos del documento

Ejemplo - ¿Está bien formado el siguiente documento XML?

```
<?xml version="1.0"?>
<mensaje>
    <destinatario>Tomas</ destinatario>
    <remitente>Juan</ remitente>
    <asunto>
    <contenido> No olvides ir a recogerme al aeropuerto mañana por la mañana!</contenido>
</mensaje>
```

No, la etiqueta < asunto> sigue abierta y las etiquetas de cierre de los elementos destinatario y remitente no son iguales (tienen un espacio).
El prólogo no tiene una declaración XML completa, pero esto no es un error, es una recomendación.

5. Utilización de espacios de nombres en XML

Permiten definir la pertenencia de los elementos y los atributos de un documento XML al contexto de un vocabulario XML. De este modo se resuelven las ambigüedades que se pueden producir al juntar dos documentos distintos, de dos autores diferentes, que han utilizado el mismo nombre de etiqueta para representar cosas distintas.

Los espacios de nombres también conocidos como name spaces, permiten dar un nombre único a cada elemento, indexándolos según el nombre del vocabulario adecuado además están asociados a un URI que los identifica de forma única.

En el documento, *las etiquetas ambiguas se sustituyen* por otras en las que el nombre del elemento está *precedido de un prefijo*, que determina el contexto al que pertenece la etiqueta, seguido de dos puntos, :. Esto es:

<prefijo:nombre_etiqueta></prefijo:nombre_etiqueta>

Esta etiqueta se denomina "nombre cualificado". Al definir el prefijo hay que tener en cuenta que **no se pueden utilizar espacios** ni **caracteres especiales** y que no puede comenzar por **un dígito**.

Antes de poder utilizar un prefijo de un espacio de nombres, para resolver la ambigüedad de dos o más etiquetas, es necesario declarar el espacio de nombres, es decir, asociar un índice con el URI asignado al espacio de nombres, mediante un atributo especial xmlns. Esto se hace entre el prólogo y el ejemplar de un documento XML y su sintaxis es la siguiente:

< conexion>://< direccionservidor>/< apartado1>/< apartado2>/...

Para saber más

Los espacios de nombres tienen una recomendación en XML - http://www.w3.org/TR/REC-xml-names/

5.1. Ejemplo: Utilización de espacios de nombres

Supongamos dos documentos que organizan la información sobre los profesores y los alumnos del Ciclo Formativo.

XML de alumnos:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<!DOCTYPE alumnos>
<alumnos>
    <nombre>Fernando Fernández González</nombre>
    <nombre>Isabel González Fernández</nombre>
    <nombre>Ricardo Martínez López</nombre>
</alumnos>
```

XML de profesores

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<!DOCTYPE profesores>
<profesores>
    <nombre>Pilar Ruiz Pérez</nombre>
    <nombre>Tomás Rodríguez Hernández</nombre>
</profesores>
```

Si uniéramos los dos documentos en uno único, sin usar espacios de nombres, no se distinguirían los profesores de los alumnos ya que en los dos casos la etiqueta < nombre> se llama igual.

Para resolverlo necesitamos definir un espacio de nombres para cada contexto.

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<!DOCTYPE miembros>
<alumnos xlmns:alumnos=”http://DAM/alumnos”>
<profesores xlmns:profesores=”http://DAM/profesores”>
<asistentes>
    <alumnos:nombre>Fernando Fernández González</alumnos:nombre>
    <alumnos:nombre>Isabel González Fernández</alumnos:nombre>
    <alumnos:nombre>Ricardo Martínez López</alumnos:nombre>
    <profesores:nombre>Pilar Ruiz Pérez</profesores:nombre>
    <profesores:nombre>Tomás Rodríguez Hernández</profesores:nombre>
</asistentes>
```

Para saber más

Tutoriales del w3shools.com

- Tutorial XML
- Validador en línea de documentos bien formados
- Documentación Web de Mozilla Developer Network

Extras

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