aprenderaprogramar.com

¿Qué es y para qué sirve el lenguaje de etiquetas XML (extensible markup language)? (DV00202A)

Sección: Divulgación

Categoría: Lenguajes y entornos

Fecha revisión: 2029

Autor: Walter Sagástegui Lescano

Resumen: Este artículo explica en qué consiste el lenguaje XML: Lenguaje de Marcado Extensible. A diferencia del HTML, separa el contenido de la presentación, y se está convirtiendo en un estándar de intercambio de datos.

¿QUÉ ES EL LENGUAJE XML?

XML (Extensible Markup Language) es un lenguaje de etiquetas, es decir, cada paquete de información está delimitado por dos etiquetas como se hace también en el lenguaje HTML, pero XML separa el contenido de la presentación. Explicaremos esto con el siguiente ejemplo:

```
<H1>Mateo</H1>          <--- HTML
<Nombre>Mateo</Nombre>  <--- XML
```

<H1> y <Nombre> son etiquetas. Ambas encierran el texto o paquete de información "Mateo". La etiqueta <H1> es de HTML, y se encarga de mostrar visualmente el texto "Mateo" en la página web en un tamaño determinado pero no dice nada del significado de "Mateo": si es una ciudad o un nombre, por ejemplo. En cambio la etiqueta <Nombre> es de XML y nos dice que "Mateo" es un nombre de persona, por lo tanto XML se preocupa del significado del texto que encierra y no de la apariencia de cómo se muestre el texto en la página web. Por eso se dice que XML es un lenguaje de etiquetas, que como hemos dicho anteriormente, separa el contenido de la presentación. Lo mismo se puede definir el lenguaje XML usando palabras mas técnicas pero con el mismo significado que la definición anterior: "XML describe el sentido semántico de los datos dejando de lado la presentación".

¿POR QUÉ ES ÚTIL EL LENGUAJE XML PARA LOS PROGRAMAS INFORMÁTICOS?

Un programa informático puede estar escrito en Java, Visual Basic y cualquier otro lenguaje. En esencia, todos los programas procesan información, entendiéndose por información "dato + significado". Para el caso que estamos viendo, el dato en el ejemplo sería "Mateo" y el significado es un "nombre de persona". Por lo tanto un documento escrito en XML tendría la información que necesitan los programas para procesar.

XML se plantea como un lenguaje estándar para el intercambio de información entre diferentes programas de una manera segura, fiable y libre, ya que no pertenece a ninguna compañía. Podemos ver por qué el XML es tan interesante para el intercambio de datos con el siguiente ejemplo:

Mateo nació el 15.10.2009 en la ciudad de Madrid con un peso de 3.1 kg y una estatura de 45 cm.

Maribel nació el 11.09.1976 en la ciudad de Sevilla con un peso de 3 Kg y una estatura de 40 cm.

Analizando el texto, nos encontramos que hay datos como "Madrid" y su correspondiente significado, que es una "ciudad" y otros más en un formato humano, tan sólo entendible por personas, no por los programas. Por tanto, podemos convertir el texto tanto en una base de datos "tradicional" como en un archivo o documento XML, que son formatos que los programas ya podrían entender, de la siguiente manera:

En formato tabla (base de datos "tradicional"):

Nombre	Fecha	Ciudad	Peso	Estatura
Mateo	15.10.2009	Madrid	3.1	45
Maribel	11.09.1976	Sevilla	3	40

En formato XML:

```
<Datos-Nacimiento>
  <Persona>
    <Nombre>Mateo</Nombre>
    <Fecha>15.10.2009</Fecha>
    <Ciudad>Madrid</Ciudad>
    <Peso>3.1Kg</Peso>
    <Estatura>45cm</Estatura>
  </Persona>

  <Persona>
    <Nombre>Maribel</Nombre>
    <Fecha>11.09.2009</Fecha>
    <Ciudad>Sevilla</Ciudad>
    <Peso>3Kg</Peso>
    <Estatura>40cm</Estatura>
  </Persona>
</Datos-Nacimiento>
```

Muchas instituciones públicas ya están utilizando XML para almacenar su información, siguiendo estos criterios de organización de datos. Esto es lo que llamamos una base de datos en XML.

Otra posibilidad interesante del XML es que a partir de un documento en XML se pueden generar archivos PDF y en otros formatos. De esta forma, la información puede ser presentada de una manera visual para su lectura por las personas y el XML sólo quedaría para ser entendido por los programas; aunque si hacemos un esfuerzo, vemos que es fácil para una persona extraer la información de un documento XML directamente.

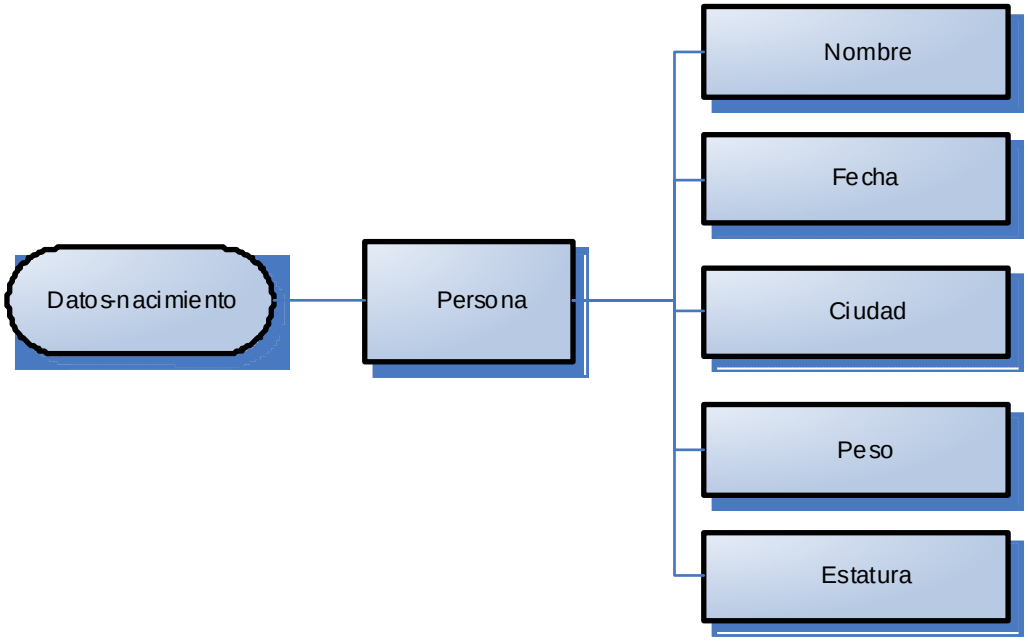
REGLAS BÁSICAS PARA ESCRIBIR CORRECTAMENTE UN DOCUMENTO XML

- Una etiqueta de apertura siempre tiene su contraparte de cierre con "/". Por ejemplo

Incorrecto: <Nombre>Mateo

Correcto: <Nombre>Mateo</Nombre>

- Sólo puede haber un elemento raíz, en el que estén contenidos todos los demás, como una especie de estructura jerárquica.



- El acrónimo “XML”(o “xml” o “xMl”, etc.) no puede usarse como caracteres iniciales de un nombre de etiqueta o atributo.

- El XML es sensible al tipo de letra utilizado (“case-sensitive”), es decir, trata las mayúsculas y minúsculas como caracteres diferentes. Por ejemplo, no es lo mismo <automóvil> que <Automóvil>

- Una etiqueta vacía, es la que no tiene contenido, por lo que se cerraría al final en la misma etiqueta de apertura. Por ejemplo:

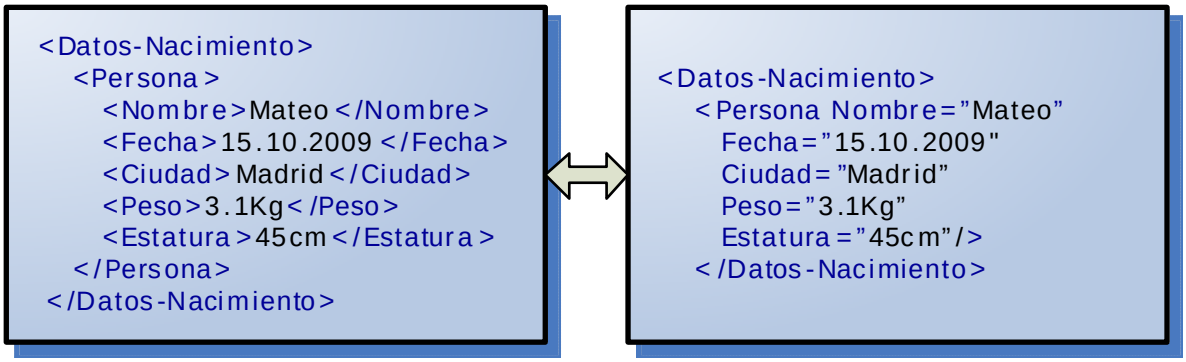
```
<Persona nombre=”walter”/>

<parámetro />
```

Las etiquetas pueden tener atributos, que son una manera de incorporar características o propiedades a las etiquetas de un documento. El atributo consta de dos partes: La propiedad del elemento y el valor de la propiedad, que siempre va entre comillas doble (“) o simple (‘). Por ejemplo: modelo y color serian atributos de la etiqueta Vehiculo

```
<Vehiculo marca=’Toyota’ modelo="45 TC" color="plomo">En venta</Vehiculo>
```

- Una etiqueta con contenido, puede modelarse como una etiqueta vacía con atributos. Por ejemplo:



EJEMPLO

A continuación daremos un ejemplo de forma práctica, en donde crearemos un documento XML que contenga la descripción de algunos partidos jugados por los equipos de fútbol de la liga española. Usando las características válidas para un documento XML podemos escribir.

```
<Liga>

    <Partido número="1">
        <Local>Barcelona</Local>
        <Visita>Real Madrid</Visita>
        <Goles-Local>2</Goles-Local>
        <Goles-Visita>3</Goles-Visita>
    </Partido>
    <Partido número="2">
        <Local>Barcelona</Local>
        <Visita>Sevilla</Visita>
        <Goles-Local>2</Goles-Local>
        <Goles-Visita>1</Goles-Visita>
    </Partido>
</Liga>
```

A partir de los datos "Goles-Local" y "Goles-Visita" podríamos realizar cálculos, por ejemplo, si:

Goles-Local > Goles-Visita ----> Ganó el equipo local

Goles-Local < Goles-Visita ----> Perdió el equipo local

Goles-Local = Goles-Visita ----> Hubo empate entre el equipo local y el visitante.