2. La tecnología XML

En este capítulo[1], se examinan las generalidades del lenguaje de codificación de documentos XML (*Extensible Markup Language*, o en español, lenguaje de etiquetado extensible). En esencia, trata de qué se puede hacer y qué no se puede hacer con documentos XML. Además, se mencionan cuáles son las principales ventajas y diferencias que hay al utilizar este tipo de etiquetado de documentos, comparados con documentos HTML. Por último, se enumeran las características que deben cumplir los documentos de texto para que éste sea un documento bien formado, y posteriormente validarlo usando una DTD o un esquema XML.

Generalidades

Qué no es XML

Antes de empezar a describir lo que son los documentos XML, y las características que deben tener los documentos de texto para ser considerados como XML, voy a hacer algunas precisiones, para explicar lo que es y no es un documento XML y lo que se puede y no se puede hacer con ellos.

[1] La información presentada aquí se obtuvo de revisar las siguientes fuentes: el libro de MacDonald, Matthew (*Office 2003 XML for Power Users*, Apress, 2004) y las siguientes páginas:

- http://es.wikipedia.org/wiki/XML
- http://es.wikipedia.org/wiki/Validacion_xml
- http://manuales.dgsca.unam.mx/xml/Qu%E9%20es%20DTD.htm
- http://www.sidar.org/recur/desdi/traduc/es/xml/xml10p/xml10p.htm
- http://www.w3.org/TR/xml11/
- http://es.geocities.com/alba1509/Fase3/teoria.html
- http://www.scribd.com/doc/6974950/XML
- http://www.webtaller.com/construccion/lenguajes/xml/lecciones/xml-10-puntos.php

Primero, XML no es un lenguaje de programación. Los documentos XML no son programas y no pueden “correr” o “ejecutar”. Segundo, aunque los documentos XML tienen una estructura arbórea, y esa característica se utilizó en esta tesis para que a partir de ella se pueda crear una base de datos relacional, XML en un sentido estricto sólo es una forma de delimitar piezas de datos, no es una base de datos.

Por otra parte, es importante mencionar que aunque XML y HTML tienen ciertas semejanzas, empezando por los caracteres ML (*Markup Language*) del nombre, son lenguajes de codificación diferentes. Tienen en común que utilizan etiquetas y atributos, (<etiqueta atributo="valor" …> delimitada por picoparéntesis < >). La diferencia consiste en que HTML tiene un conjunto (relativamente cerrado) de etiquetas con estructuras especificadas por quienes diseñaron el lenguaje, dicho de otra forma, se especifica lo que cada etiqueta y atributo significan, también nos indican la apariencia que deben tener los documentos en los navegadores (el texto, su tamaño, tipo de letra, color, su ubicación, y hasta el fondo que debe tener el texto). En cambio, XML utiliza las etiquetas para delimitar y estructurar piezas de datos según las especificaciones y necesidades del usuario y deja la interpretación de los datos, completamente, a la aplicación que los lee. Por ejemplo, si en un documento HTML se observa una etiqueta "<p>", esto quiere decir que se trata de un párrafo, en cambio si encontramos la etiqueta "<p>", en un documento XML, no necesariamente se trata de un párrafo; es decir, dependiendo de cómo se quiera usar, puede tratarse de un precio, un parámetro,

una persona, u otra cosa. Además, si quisiéramos marcar un párrafo lo podríamos hacer también llamándolo <parrafo>, debido a que en los documentos XML cada usuario tiene la libertad de especificar el nombre de las etiquetas según convenga.

En pocas palabras se puede decir que XML es un lenguaje similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. Conviene hacer notar que XML es un formato que permite la lectura de datos a través de diferentes aplicaciones y no se creo para sustituir a HTML, debido a las diferencias antes mencionadas.

Una vez que ya se examinaron las generalidades de los documentos XML y también se mostraron algunas diferencias que existen entre un documento HTML y uno XML, empezaré a describir las características que deben cumplir los documentos de texto para que puedan ser llamados documentos XML.

XML

La especificación o documento que define cómo diseñar y aplicar las etiquetas y los atributos en XML es *XML 1.0*. Alrededor de esta especificación se crearon diferentes módulos opcionales que tienen colecciones de etiquetas y atributos, o modelos para especificar tareas que pueden tanto presentar la información u otra cosa. Por ejemplo:

- *Xlink,*
- *CSS,*

- *XSL ,*
- *XSLT,*
- *DOM ,*
- *XML Namespaces,*
- *Esquemas XML (XML Schemas),*
- entre otros.

Respecto a los documentos XML, éstos están formados por un prólogo y por el cuerpo del documento. El prólogo es una etiqueta donde se especifica la versión XML, el tipo de documento y otras cosas (`<?xml version="1.0" … ?>`). Esta primera etiqueta contiene la información de la versión de manera obligatoria, (hay que mencionar que aunque la versión más utilizada es la 1.0, ya está disponible la versión 1.1 de XML[2]), dentro de la misma etiqueta también se incluye opcionalmente la codificación de caracteres utilizada (*encoding*); ésta hace referencia al modo en que se representan internamente los caracteres, normalmente UTF-8 ó UTF-16 ó Unicode. Y por último, también en esta primera etiqueta, se puede incluir la declaración independiente (standalone) que indica al procesador XML si un documento es independiente (standalone="yes") o se basa en información de fuentes externas, es decir, si depende de declaraciones de marca externas como una DTD externa (standalone="no"), esta es la opción por defecto.

A continuación se muestra un ejemplo de la primera etiqueta:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

[2] http://www.w3.org/TR/xml11/

Después de esta etiqueta sigue la declaración de tipo de documento. Ésta enlaza el documento con su DTD (definición de tipo de documento) o esquema XML. Opcionalmente, el DTD puede estar incluido en la propia declaración o ambas cosas al mismo tiempo.

Luego sigue el cuerpo del documento. La primera característica indispensable que debe tener para que pueda considerarse documento XML es que debe contener un y sólo un elemento raíz para cada documento; esto es, un solo elemento en el que todos los demás elementos estén contenidos dentro de éste. Estos elementos contenidos se encuentran anidados y correctamente cerrados, es decir, se basa en una estructura jerárquica y su función es la misma que el elemento raíz de un documento HTML `<HTML>Contenido</HTML>`.

Todas las etiquetas (que representan elementos o entidades) utilizadas en el documento se declaran en una DTD interna o externa, o en un esquema XML. Todos los elementos, atributos y entidades que se utilicen deben escribirse con una sintaxis correcta, según esta DTD o esquema XML. Algunas de las características más importantes que debe tener un documento XML se mencionan a continuación.

Todos los elementos deben estar delimitados por una etiqueta inicial y otra final con el mismo nombre en el siguiente formato:

```
<etiqueta></etiqueta >
```

Los documentos siguen una estructura estrictamente jerárquica respecto a las etiquetas que delimitan sus elementos. Una etiqueta debe estar correctamente incluida dentro de otra. Esto quiere decir que las etiquetas deben anidarse correctamente. Obviamente, todos los elementos deben estar cerrados de la manera apropiada.

```
<documento>
          <titutlo>Historia</titulo>
          <fechaPublicacion>1821</fechaPublicacion>
</documento >
```

Los atributos de una etiqueta en XML deben estar contenidos dentro de esta y los valores de dichos atributos deben ir entre comillas dobles. Los elementos vacíos deben terminar con '/' (autocierre), <ejemplo/>, o añadiendo una etiqueta de fin, <ejemplo></ejemplo>, y no puede haber etiquetas no cerradas. Los siguientes ejemplos contienen dos atributos *tipo* y *pags,* en el primero existe una etiqueta de cierre, y la segunda con autocierre, ambos son correctos:

```
<otro tipo="acta" pgs="1"></otro>

<otro tipo="acta" pgs="1"/>
```

XML es sensible a mayúsculas y minúsculas y los nombres de las etiquetas pueden ser alfanuméricos. También hay que hacer notar que existe un conjunto de caracteres que se pudieran ser considerados como espacios en blanco (espacios, tabulaciones, saltos de línea, entre otros) que en los documentos XML, cada uno de estos se consideran caracteres diferentes en el marcado XML.

Las etiquetas, y todos los elementos que se encuentran dentro de los picoparéntesis, son partes del documento que el procesador XML puede entender. El resto del documento, es la información que se espera sea leída por los usuarios.

A continuación se presenta un ejemplo sencillo, en dónde se cumple con las condiciones que se mencionaron anteriormente:

```
<?xml version="1.0" encoding=" UTF-8" ?>
      <referencias>
            <referencia id="chem0">
                  <lugar>Apatzingan</lugar>
                  <otro tipo="decreto" pgs="1"/>
                  <corta><i>Decreto constitucional para la libertad de la
          América Mexicana</i></corta>
            </referencia>

            <referencia id="chem1">
                  <otro tipo="acta" pgs="1"/>
                  <corta><i>Acta    de    independencia    del    Imperio
          Mexicano</i></corta>
                  </referencia>
      </referencias>
```

Si un documento de texto cumple con las condiciones anteriores que son las especificaciones de marcado XML, entonces se dice que dicho documento esta “bien formado” (*well formed*). Al cumplir con esta característica dichos documentos pueden ser analizados por un parser, para verificar que siga la sintaxis XML. Hay que mencionar que no son sinónimos que un documento sea “bien formado” y que sea válido.

La sintaxis XML establece los requisitos mínimos que debe cumplir un documento XML. Si se quiere tener un mayor control del contenido de los documentos y aprovechar dicha información de una mejor manera, es necesario establecer un conjunto de definiciones más adecuadas a nuestras necesidades, este proceso es llamado validación de los documentos XML.

Validación XML

La validación es la parte más importante dentro de esta exposición, porque determina si un documento creado se ajusta a las restricciones descritas en el esquema utilizado para su elaboración. Es cierto que se pueden utilizar documentos que no se encuentren asociados a ningún esquema y por lo tanto no tendríamos necesidad de validarlos. Sin embargo, en la mayoría de los casos conviene que estén validados para aprovechar al máximo las ventajas de los documentos XML.

La validación XML es la comprobación de que un documento en lenguaje XML está bien formado y se ajusta a una estructura definida. Un documento bien formado sigue las reglas de sintaxis de XML, pero un documento válido además de cumplir con lo anterior respeta las normas establecidas por su DTD o esquema XML utilizado. Controlar el diseño de documentos a través de esquemas aumenta su nivel de fiabilidad, consistencia y precisión, logrando con esto un mejor manejo entre diferentes aplicaciones y usuarios. Cuando creamos documentos XML válidos logramos que estos se ajusten de una mejor manera, a las necesidades que nosotros requerimos.

Métodos de validación

Aunque ya se mencionó la importancia de validación o validar los documentos XML, conviene mencionar que existen varios métodos para validar los documentos XML. Los métodos más usados son la *DTD* de XML versión 1.0, el *XML Schema* de W3C, aunque no son los únicos (ver por ejemplo: RELAX NG, Schematron).

A continuación se explican de una forma más detallada los DTD y los esquemas XML que se utilizan para validad un documento XML, describiendo algunas de sus principales características y ventajas.

DTD

La DTD (document type definition) es el formato de esquema nativo (y el más antiguo) para validar documentos XML, heredado de SGML. Dichos esquemas utilizan una sintaxis diferente a la de XML para definir la estructura o modelo de contenido de un documento XML válido:

- Define todos los elementos.
- Define todas las relaciones entre los distintos elementos.
- Proporciona toda información adicional que pueda ser incluida en el documento (atributos, entidades, notaciones).
- Aporta comentarios e instrucciones para su procesamiento y para la representación de los formatos de datos.

Es el método más antiguo usado para validar. Las DTDs pueden estar asociadas a un documento XML de manera interna o externa, o de ambas

maneras a la vez; esto es, parte de una puede estar contenida dentro del documento XML, mientras que la otra puede estar en un archivo de texto separado.

Algunas de las principales desventajas de este tipo de esquema de validación, es que el DTD es poco flexible para definir elementos con contenido mixto, es decir, que incluyan otros elementos además de texto. Además, se complica indicar a qué tipo de dato (número, fecha, moneda) ha de corresponder un atributo o el texto de un elemento.

Ejemplo de un DTD:

```
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA)>
```

Esquema XML (*XML Schema*)

Ya que se explicó brevemente cuales son las característica de un DTD, sin embargo hay que destacar que existe un formato que nos facilita la elaboración de esquemas, y nos permite especificar además de estructura, también anidación, restricciones y tipos de dato complejos con base en tipos de dato más simples, cosa que no se puede realizar utilizando una DTD. El esquema XML se creó para solucionar algunas limitaciones que se presentaban en los DTD, especialmente en lo referente a lo complicado que pudiera ser definir tipos de datos que no sean de

texto puro y la falta de jerarquización en las DTD. Tanto los DTD como el esquema XSD, están descritas por el W3C (Consorcio World Wide Web) que es un consorcio donde se desarrollan estándares de WEB. El esquema XML, también llamado XSD (*XML Schema Definition*), es un lenguaje de representación más completo y más poderoso que el de una DTD y debido a que se pueden declarar un número mayor de tipos de datos, además de que tienen una estructura jerárquica lo que facilita la creación de este tipo de documentos; asimismo utiliza una sintaxis parecida a la de XML, lo que le permite especificar de forma más detallada un sistema, gracias a que cuenta con un extenso de tipos de datos y se pueden crear los propios. A diferencia de las DTDs, soporta la extensión del documento sin mayor problema.

Una de las desventajas es que a la hora de validar, no todos los parser contienen la información necesaria para poder validar un documento que haya utilizado un esquema XSD, al contrario de los DTD que están contenidos en casi todos los programas que se utilizan para validar un documento XML, además de que debido a sus características las definiciones pueden ser complejas, lo que provoca mayor gasto de recursos al momento de validar, pero es una desventaja menor comparada con la gran cantidad de cosas que nos permite realizar.

Hay que mencionar que el uso de las DTD y esquemas XML, son los más utilizados actualmente, pero no son los únicos y existen otros “esquemas descriptivos” que nos facilitan ciertas tareas especificas, aunque con los esquemas XSD se puede definir casi la totalidad de las necesidades que

deseemos. En algunas ocasiones resulta muy complicado ó redundante la definición de algunos tipos de datos, para estos casos especiales se crean otros esquemas que nos dan atajos para este tipo de necesidades especificas; hay algunos que son reconocidos por la W3C, y hay otros que a pesar de su amplia utilización no se consideran todavía dentro de las recomendaciones de la W3C, algunos de ellos son: RELAX NG, Schematron, Namespace Routing Language (NRL), Document Schema Definition Languages (DSDL), Document Definition Markup Language (DDML), Document Structure Description (DSD), SGML, Schema for Object-Oriented XML (SOX).

Ejemplo esquema XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3c.org/2001/XMLSchema">
     <xsd:element name="Libro">
         <xsd:complexType>
            <xsd:sequence>
                 <xsd:element name="Título" type="xsd:string"/>
                 <xsd:element       name="Autores"       type="xsd:string"
maxOccurs="10"/>
                 <xsd:element name="Editorial" type="xsd:string"/>
            </xsd:sequence>
            <xsd:attribute name="precio" type="xsd:double"/>
        </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

Ventajas de los esquemas XML frente a los DTDs

Algunas de las ventajas que tienen los esquemas XSD, mas importantes y que hacen que sea mas robusto comparado con un DTD son las siguientes:

- Su sintaxis está basada en la de XML, al contrario que los DTDs, en la que su sintaxis puede resultar algo confusa, especialmente si no se ha manejado este tipo de esquema con anterioridad.

- Se puede manejar en términos generales como cualquier otro documento XML, ya que este tipo de esquemas tienen una estructura jerárquica.
- Permiten especificar los tipos de datos, ya que no están limitados a los que se definan por el propio esquema, sino que pueden existir tipos de datos complejos creados por la persona que usa el esquema XML.
- Son extensibles.

Ventajas de la tecnología XML

Las ventajas que se tiene al utilizar documentos XML, pueden enumerarse en una gran cantidad, sin embargo existen algunas que son de mayor importancia. Comenzando por el hecho de que el lenguaje es extensible y es definido por la persona que lo utiliza, lo cual nos permite generar etiquetas entendibles por el usuario que las crea adaptándose a sus necesidades, y una vez diseñado el lenguaje y puesto a funcionar, es posible aumentarlo usando nuevas etiquetas para describir partes de texto de una mejor manera o para describir algún tipo de información que no había sido considerada o que no existía, de manera que se puedan usar las definiciones sencillas, a la vez que se usen definiciones más detalladas, y ambas, tanto nuevas como antiguas convivan sin mayor problema.

Otra de las ventajas de utilizar este tipo de documentos, es que debido a la estructura arbórea propia de los documentos XML, es sencillo de entender de lo que tratan los documentos aún sin conocerlos detalladamente, ya que tales documentos presentan un orden. Añadiendo que actualmente existe una gran cantidad de aplicaciones que nos permiten validar los documentos de una manera rápida y sencilla.

Por último hay que agregar que existen diferentes opciones para generar formatos de texto a partir de la información previamente codificada en XML, creando a partir de un documento XML, diferentes formatos de texto según lo necesitemos.

Aquí se termina el capítulo que abarca de forma general lo que son los documentos XML, y sus características que sirvieron de base para realizar el análisis, tanto de los documentos XML como de los esquemas XSD, para posteriormente crear el algoritmo que me permitiera desarrollar el constructor automático de la base de datos. En el siguiente capítulo se abordará lo referente a la aplicación de la tecnología XML en el *Corpus Histórico del Español en México*, mostrando de una forma general un documento y cómo está codificado.