Introducción a XML

Índice

1. Introducción

XML significa lenguaje de marcas extensivo (Extensible Markup Language). Es un lenguaje usado para estructurar información en un documento o en general en cualquier fichero que contenga texto, como por ejemplo una tabla de datos.

Ha ganado muchísima popularidad en los últimos años debido a ser un estándar abierto y libre, creado por el Consórcio World Wide Web, W3C (los creadores de la www).

El XML fue propuesto en 1996, y la primera especificación apareció en 1998. Desde entonces su uso ha tenido un crecimiento acelerado, que se espera que continúe durante los próximos años.

2. Ventajas del XML

Antes de ser lanzado el XML, ya existían otros lenguajes de marcas, como por ejemplo el HTML (Hyper Text Markup Language), basados en el lenguaje generalizado de marcas SGML (Standard Generalized Markup Language). El problema con el SGML es que por ser muy flexible y muy general, se torna difícil el análisis sintáctico de un documento y la especificación de la estructura.

El XML tiene la ventaja de poder ser más exigente en cuanto a la organización del documento, lo cual resulta en documentos mejor estructurados.

Por ser posible exigir la estructura que deben tener un tipo determinado de documentos, se vuelve posible extraer información de varios documentos automáticamente, por ejemplo para crear bases de datos o listados con información sobre todos los documentos.

El XML ha servido para definir un gran número de lenguajes de marcado particulares, tales como:

- XHTML: revisión de HTML para adaptarlo a XML
- SVG: descripción de gráficos vectoriales
- DocBook: esquema general de documentos
- MathML: descripción de fórmulas matemáticas
- ... y otros miles de lenguajes ...

3. Conceptos básicos

Los ficheros XML son ficheros de texto, que en principio está en código Unicode, pero se pueden usar otros alfabetos como el latin-1. Existen cinco caracteres especiales en XML: los símbolos menor que, <, mayor que, >, las comillas dobles, ", el apóstrofe ' y el caracter &. Los símbolos mayor que y menor que se usan para delimitar las marcas que dan la estructura al documento.

Cada marca tiene un nombre; veamos un ejemplo: la marca <figura>, que puede tener uno o más *atributos*: <figura fichero="foto1.jpg" tipo="jpeg"> tiene dos atributos, "fichero" y "tipo". Los atributos toman valores que tienen que estar entre comillas o entre apóstrofes.

Cuando sea necesario usar uno de los 5 caracteres especiales en el texto, para evitar que sean interpretados de forma especial se usan las siguientes *entidades*: <, >, ", ', &, para <, >, ", ' y &, respectivamente.

Una diferencia importante con SGML, y en particular HTML, es que los nombres de las marcas y de sus atributos distinguen entre mayúsculas y minúsculas; <a> y <A> serian dos marcas diferentes. Normalmente se suelen usar únicamente minúsculas para los nombres de las marcas y de sus atributos. Otra diferencia sobresaliente con SGML es que en XML ninguna marca se puede dejar abierta; o sea, por cada marca, por ejemplo debería existir una marca correspondiente que indica donde termina el contenido de la marca. En el siguiente ejemplo:

<refrán>El que mucho abarca, poco aprieta</refrán>

El contenido de la marca "refrán" esta claramente delimitado entre <refrán> y </refrán>. Si una marca cualquiera no contiene ningún texto, por ejemplo <hr></hr>, se puede abreviar de la siguiente forma: <hr/>, pero nótese que la primera forma también es válida, en cambio escribir únicamente <hr> o <hr> daría un error.

4. Documento XML

La estructura general de un documento XML está formada por tres partes:

- **Prólogo**, opcional: Conteniendo una secuencia de instrucciones de procesamiento y/o declaración de tipo de documento.

- **Cuerpo**: Un árbol único de elementos marcados, con anidamiento estricto.

- **Epílogo**, opcional: En general se omite, ya que no está claro para qué sirve. Está pensado para contener instrucciones de procesamiento, pero resulta poco intuitivo poner estas instrucciones al final.

Además puede haber comentarios en cualquier parte.

Intuitivamente, el contenido de información del documento es el cuerpo. El prólogo y el epílogo sirven para facilitar la interpretación del documento

4.1. Prólogo: Declaración XML

La declaración XML es una instrucción de procesamiento especial. Es opcional. Cuando existe debe ser la primera instrucción del prólogo. Su formato es:

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

• **version:** atributo obligatorio. Indica la versión de XML usada en el documento. Actualmente la versión más recomendada es la 1.0.

• **encoding:** atributo opcional, recomendado. Indica la forma en que se ha codificado el documento. y debe ser un valor IANA válido. Por defecto 'UTF-8' o 'UTF-16'. Actualmente el más recomendado es 'ISO-8859-1', permite representar acentos y diéresis.

Los nombre 'xml', 'version', ... deben escribirse en minúsculas. Los valores pueden escribirse en minúsculas o mayúsculas ('ISO-8859-1' = 'iso-8859-1').

4.2. Prólogo: Declaración de tipo de documento

La declaración de tipo de documento es opcional. Se escriben en el prólogo, y tiene un formato especial, distinto de las marcas y de las instrucciones de procesamiento. El formato es uno de los siguientes:

• *<!DOCTYPE nombre-elemento PUBLIC public-ID system-ID ... >*
• *<!DOCTYPE nombre-elemento SYSTEM system-ID ... >*

Donde:

- nombre elemento es el nombre del elemento principal (elemento raíz del cuerpo)
- public ID es un identificador asociado al lenguaje de marcado particular
- system-ID es una referencia a un DTD o XSD externo
- DTD: definición de tipo de documento (Document Type Definition)
- XSD: definición de esquema XML (Xml Schema Definition)

Ejemplo de documento XML sin incluir declaración de tipo de documento o documento XML sin DTD

```
<!-- Prólogo. Declaración XML a través de una instrucción de procesamiento

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

    <!-- Cuerpo  -->

    <personas>    <!--  Donde personas es elemento raíz del cuerpo  -->
    <persona>
        <nombre>Rosa</nombre>
        <apellido1>Casas</apellido1>
        <apellido2>Rueda</apellido2>
    </persona>
    <persona>
        <nombre>Sergio</nombre>
        <apellido1>Fuentes</apellido1>
        <apellido2>Río</apellido2>
    </persona>
    </personas>
```

5. Comentarios

Un documento XML puede contener anotaciones en forma de comentario. Los comentarios no son parte del contenido de información del documento, y pueden ser ignorados por los procesadores XML. Los comentarios se escriben como:

<!-- *...texto del comentario...* -->

El texto de un comentario no puede contener la secuencia --.

6. Instrucciones de procesamiento

Son directivas que pueden ser interpretadas por los procesadores XML. El formato de una instrucción de procesamiento es:

<?*nombre ... texto de la instrucción ...* ?>

7. Nombres

En XML se utilizan nombres que deben estar formados de la siguiente manera:

- Inicial : letra _ : (letra, subrayado, dos puntos)
- Resto: letra _ : - . (lo mismo más: guión, punto)
- Se distinguen mayúsculas y minúsculas

Un nombre simple sólo contiene letras, subrayado y guiones. Los caracteres punto y dos puntos se usan en nombres cualificados.

8. Elementos

Son fragmentos de información delimitados por marcas, de la siguiente manera:

- Marca inicial: <x ....>
- Contenido: texto u otros elementos.
- Marca final: </x>

9. Marcas

Sirven para delimitar los elementos que componen el documento XML. Un elemento queda delimitado por una marca inicial y otra final. Si el elemento no tiene contenido, se puede escribir en forma abreviada como una sola marca. El formato de las marcas es:

- Marca inicial: <nombre atributos_opcionales>
- Marca final: </nombre>
- Elemento vacío: <nombre atributos_opcionales /> (equivale a <nombre ...></ nombre>)

10. Literales

Sirven para delimitar fragmentos de texto, de acuerdo con las siguientes reglas:

- Delimitados por comillas simples o dobles: 'ejemplo' "ejemplo"
- Se puede usar la otra dentro del literal: "Roger O'Connors dijo 'Sí' al votar"
- Si hay que usar el delimitador dentro del literal se usa la referencia a entidad '
(') o " (")

11. Atributos

Son fragmentos de información que forman parte de la marca inicial de un elemento. La sintaxis es:

- <nombre_marca nombre_atributo = 'valor' nombre_atributo = "valor" ...>
- No puede haber dos atributos con el mismo nombre en la misma marca
- Los valores de los atributos se dan como literales, entre comillas o apóstrofos.

12. Tipos de documentos XML

Existen dos tipos de documentos XML:

- Documentos XML bien formados
- Documentos XML válido

Se dice que un **documento XML está bien formado** cuando cumple las reglas sintácticas indicadas. Los procesadores XML pueden rechazar cualquier documento que no esté bien formado.

Un **documento XML válido** es un documento que está bien formado, y además cumple con la definición de un lenguaje de marcado particular. Es decir, el cuerpo del documento tiene una estructura de elementos compatible con el lenguaje concreto al que corresponde.

13. Definición de tipo de documento: DTD

Para mantener compatibilidad con SGML, el estándar XML mantiene el metalenguaje DTD de definición de lenguajes particulares de marcado. Las siglas DTD significan *Document Type Definition*, y se refieren, por tanto, a la definición de un tipo o esquema de documento.

La definición del tipo de documento puede hacerse:

- En el propio documento, dentro del DOCTYPE. Documento XML con DTD interno.
- En un fichero separado, y poner la referencia en el DOCTYPE. Documento XML con DTD externo.

El lenguaje DTD permite definir elementos, atributos, entidades y notaciones (estas últimas se utilizan poco).

- Los **elementos** configuran la estructura general de un documento XML, y se anidan unos dentro de otros formando un árbol. Para cada elemento se define su nombre y la estructura de su contenido (texto u otros elementos).

- Los **atributos** son fragmentos de información asociados a un elemento. Tienen nombre y su contenido es siempre texto. No pueden anidarse.

- Las **entidades** son similares a las *macros* de ciertos lenguajes de programación. Son fragmentos de texto constantes a los que se puede hacer referencia mediante un nombre. Sirven para simplificar la escritura de documentos y DTDs en los que aparecen repetidamente ciertos fragmentos de texto.

- Las **notaciones** sirven para delimitar contenido no XML dentro de un documento XML.

El formato general de una definición elemental en una DTD es:

<!*clase parámetros ...*>

Donde clase será ELEMENT, ATTLIST, ENTITY o NOTATION, y los parámetros dependerán de la clase de definición.

13.1. Definición de elementos

Los parámetros de una definición de elemento son su nombre y el esquema de su contenido. El formato de la definición puede ser uno de los siguientes:

<!ELEMENT *nombre* ANY >
<!ELEMENT *nombre* EMPTY >
<!ELEMENT *nombre* (*expresión regular*) >
<!ELEMENT *nombre* (*expresión regular*)*repetición* >
<!ELEMENT *nombre* (#PCDATA) >
<!ELEMENT *nombre* (#PCDATA | *nombre* | *nombre* ...)* >

La primera forma define un elemento cuyo contenido puede ser cualquiera. La segunda forma define un elemento sin contenido.

Las formas tercera y cuarta definen elementos compuestos que contienen otros elementos y cuya estructura debe ajustarse a la expresión regular que se indica. La expresión regular debe estar formada por nombres de elementos y los metacaracteres de agrupación "(" ")", de secuencia y alterativa "," "|" y de repetición "+" "?" "*". Nótese que siempre es necesario un nivel externo de paréntesis. Además estas formas de expresión no pueden contener el símbolo #PCDATA.

Las dos últimas formas definen elementos con lo que se denomina *mixed content*, formado por texto solo o entremezclado con otros elementos. El nombre especial #PCDATA (que indica contenido de texto) debe aparecer siempre al principio de la expresión, que contendrá sólo ese término o será una repetición de una alternativa simple.

Ejemplo de documento XML con DTD interno

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE personas [
<!ELEMENT personas (persona+)>
<!ELEMENT persona (nombre,apellido1,apellido2)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellido1 (#PCDATA)>
<!ELEMENT apellido2 (#PCDATA)>
]>
<personas>
<persona>
        <nombre>Rosa</nombre>
        <apellido1>Casas</apellido1>
        <apellido2>Rueda</apellido2>
</persona>
<persona>
        <nombre>Sergio</nombre>
        <apellido1>Fuentes</apellido1>
        <apellido2>Río</apellido2>
</persona>
</personas>
```

Ejemplo de documento XML con DTD externo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE personas SYSTEM "personas.dtd">
<personas>
<persona>
        <nombre>Rosa</nombre>
        <apellido1>Casas</apellido1>
        <apellido2>Rueda</apellido2>
</persona>
<persona>
        <nombre>Sergio</nombre>
        <apellido1>Fuentes</apellido1>
        <apellido2>Río</apellido2>
</persona>
</personas>
```

Contenido del fichero “personas.dtd”

```
<!ELEMENT personas (persona+)>
<!ELEMENT persona (nombre,apellido1,apellido2)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellido1 (#PCDATA)>
<!ELEMENT apellido2 (#PCDATA)>
```

13.2. Definición de atributos

Los parámetros de una definición de atributos son el nombre del elemento al que corresponden y los nombres y descripciones de contenido de los atributos. El conjunto de atributos de un elemento puede declararse en una sola definición, o por partes, en varias definiciones separadas. El formato de una definición de atributos es el siguiente:

<!ATTLIST *elemento*
nombre tipo tratamiento_por_defecto
nombre tipo tratamiento_por_defecto
...
>

- *elemento*: es el nombre del elemento al que corresponden los atributos
- *nombre*: es el nombre del atributo
- *tipo*: CDATA, (*valor* | *valor* | ...), ID, IDREF, IDREFS, NMTOKEN, NMTOKENS
- *tratamiento_por_defecto*: #REQUIRED, #IMPLIED, #FIXED *valor_por_defecto*, *valor_por_defecto*

Un valor del tipo CDATA corresponde a un valor de texto, en general. Un valor del tipo enumerado (valor | valor | ...) especifica uno entre varios posibles *nmtoken*. Un valor del tipo ID es un nombre que debe ser único en todo el documento, y que sirve para identificar el elemento. Un valor del tipo IDREF es un nombre que debe aparecer como valor de un atributo ID en algún elemento del documento. Un valor del tipo IDREFS es una lista de IDREF separados por espacio en blanco. Un valor del tipo NMTOKEN es similar a un nombre, sin la restricción del carácter inicial. Un valor del tipo NMTOKENS es una lista de NMTOKEN separados por espacio en blanco.

Un atributo #REQUIRED debe aparecer siempre. Un atributo #IMPLIED es opcional, sin un valor por defecto. Un atributo #FIXED debe tener el valor indicado si no se omite, y si se omite se asumirá el valor indicado. Si no se indica #REQUIRED ni #IMPLIED ni #FIXED sino sólo un valor por defecto, el atributo es opcional y si se omite se asume el valor por defecto.

13.3. Definición de entidades

Como ya se ha dicho las entidades son valores constantes a los que se puede hacer referencia mediante un nombre. Hay dos clases de entidades:

- *General entities*: para ser usadas en el contenido del documento
- *Parameter entities*: para ser usadas en la DTD

Una *general entity* se define como:

<!ENTITY *nombre valor_de_sustitución* >

El *valor_de_sustitución* puede ser un texto literal, que a su vez puede contener referencias a otras entidades. A la entidad se hace referencia con &nombre; en el contenido del documento.

Una *parameter entity* se define como:

<!ENTITY % *nombre valor_de_sustitución* >

El *valor_de_sustitución* puede ser un texto literal, que a su vez puede contener referencias a otras entidades. A la entidad se hace referencia con %nombre; en la DTD.