Documentos bien formados

Estructura y sintaxis de XML

Un documento XML está formado, en principio, por lo que se conoce como “texto plano”, esto es, texto en el cual todos los caracteres se representan visualmente, sin existir caracteres no visibles exceptuando los de salto de línea, tabulador o espacio.
Los documentos escritos usando XML contendrán marcas para separar la información que estructura el documento de la información que se quiere almacenar. Para construir dichas marcas, en XML, se usan los caracteres “<” y “>” para delimitar el texto que se desea marcar, mientras que el carácter “/” sirve para indicar la etiqueta de finalización del marcado.

Ejemplo: <nombre>Elías</nombre>

Esta construcción se denomina “elemento” y constituye la base principal de los documentos XML. Además de los elementos, un documento XML puede contener otros tipos de información. A continuación, se especifican los componentes relevantes.

ETIQUETAS, ELEMENTOS Y ATRIBUTOS

Las etiquetas son el componente de XML que permite definir los elementos que conformarán un documento de la siguiente forma:

<etiqueta>Valor</etiqueta>

Como se puede observar los elementos[1] son la base de la estructura de los documentos; se formarán usando una etiqueta de inicio, otra de fin, delimitadas mediante los caracteres “<” y “>”, y que comparten el identificador textual pero añadiendo el carácter “/” al principio En medio de las etiquetas de inicio y fin del elemento se representará el contenido que se desee almacenar en ese elemento. Este contenido puede a su vez englobar otros elementos. Por otro lado, los elemento pueden no contener ningún valor, pero en ese caso se deberá usar solamente la etiqueta de finalización.

<direccion></direccion>

Se considera que el elemento XML engloba todo lo que se encuentra entre las correspondientes etiquetas de inicio y de fin y pueden contener tanto otros elementos como simplemente texto o una combinación de ambos.

```
<alumno>
        <nombre>Pablo</nombre>
        <apellido>Pérez</apellido>
        <telefono>915555555</telefono>
        <direccion></direccion>
</alumno>
```

[1] Ver elementos en <http://w3schools.com/xml/xml_elements.asp>

Los nombres de los elementos deberán empezar por una letra o bien por el carácter “_” o “:” siempre y cuando el principio no contenga la palabra “xml” en cualquier combinación posible de mayúsculas y minúsculas. Además, los nombres son *case sensitive* y sólo podrán contener letras, números y los caracteres “-”, “.”, “_”, “:”.

Los elementos pueden asimismo contener atributos, los cuales se especificarán en la etiqueta de inicio del elemento. El objetivo de los atributos es poder proporcionar una información adicional sobre un elemento concreto. La sintaxis pra representar los atributos consiste en especificar el nombre del atributo dentro de la etiqueta de inicio, a continuación un símbolo “=” y finalmente el valor del atributo delimitado por comillas dobles o por comillas simples.

Siguiendo con el ejemplo presentado previamente, el sexo del alumno anteriormente se había representado usando un elemento mientras que en el siguiente ejemplo se utiliza un atributo para denotar dicha característica. Además se ha añadido el atributo “fechaNacimiento” para el elemento alumno y el atributo “tipo” para el elemento teléfono.

```
<alumno sexo=”varón” fechaNacimiento=”5/6/1990”>
        <nombre>Pablo</nombre>
        <apellido>Pérez</apellido>
        <telefono tipo=”móvil”>915555555</telefono>
        <direccion>Ronda de Segovia 111</direccion>
</alumno>
```

Observamos que generalmente se pueden usar tanto atributos como nuevos elementos para representar información. Sin embargo, el uso de un número excesivo de atributos puede provocar que el documento XML sea menos legible, más difícil de mantener y, asimismo, difícilmente extensible. Además, hay que tener en cuenta que los atributos no pueden contener información en forma de árbol, esto es, no pueden contener otros elemento o atributos tal y como sucede con los elementos. De forma general, se puede establecer la recomendación de no usar atributos en exceso y dejarlos casi exclusivamente para representación de metadatos.

Siguiendo esta recomendación, en el ejemplo tanto el atributo “sexo” como el atributo “fechanacimiento” se pueden convertir en elementos. Además en este ejemplo se ha añadido un identificador del alumno como atributo:

```
<alumno id=”532”>
        <nombre>Pablo</nombre>
        <apellido>Pérez</apellido>
        <fechaNacimiento>
                <dia>5</dia>
                <mes>6</mes>
                <año>1990</año>
        </fechaNacimiento>
        <sexo>varón</sexo>
        <telefono tipo=”móvil”>915555555</telefono>
        <direccion>Ronda de Segovia 111</direccion>
</alumno>
```

NODOS

Un nodo está compuesto por una etiqueta, sus atributos y su contenido. El contenido es todo lo que está entre la etiqueta de apertura y la de cierre, lo que puede incluir a otros nodos.

Si en el contenido de un nodo hay otros nodos, nos referiremos a ellos como n*odos descendientes*. En el primer nivel de descendencia están los nodos hijos. Si estos nodos tienen a su vez otros nodos en su interior, serán hijos del inmediatamente anterior, y nietos del otro. Esta estructura de padres e hijos también es conocida como estructura arbórea o jerárquica.

Para que esta jerarquía esté correctamente descrita es necesario que las etiquetas se abran y cierren con orden y concierto. Cuando este orden se sigue, se dice que los nodos están correctamente *anidados*.

CONTENIDO

Entre las etiquetas de apertura y cierre podemos escribir otras etiquetas, como hemos indicado, pero también podemos escribir texto independiente.

Ejemplo: <actor>Jonathan Pryce</actor>

Sin embargo, existen algunas limitaciones sobre lo que puede ser escrito como contenido de un nodo. En concreto, si queremos utilizar los caracteres “ampersand” (&), “menor que” (<), “mayor que” (>), “comillas simples” (‘), o “comillas dobles” (“), deberemos utilizar las *entidades de carácter*.

CARACTERES ESPECIALES (entidades de carácter)

En XML algunos símbolos son reservados del lenguaje, por lo que para poder representarlos es necesario usar unos códigos. Estos se definen usando el símbolo “&” seguido de una palabra clave y terminados por punto y coma. Estas construcciones son entidades predefinidas. A continuación se detallan algunos de los más relevantes.

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

INSTRUCCIONES DE PROCESAMIENTO

Más allá de los propios datos contenidos en los ficheros XML y las etiquetas de marcado en un fichero XML, se pueden encontrar instrucciones especiales llamadas *instrucciones de procesamiento*. Éstas comienzan con “<?” y terminan con “?>”. Una de las instrucciones de

procesamiento más habituales es la que se usa para indicar que versión de XML se va a usar y cuál es la codificación de caracteres que se va a usar.

Ejemplo: <?xml version=”1.0” encoding=”UTF-8”?>

COMENTARIOS Y SECCIONES CDATA

Dentro de un documento XML se puede añadir información, que no pertenezca ni al marcado ni información de contenido del documento, y que sirve para documentarlo en forma de comentarios internos. La sintaxis de un comentario consta de un texto delimitado por una marca inicial “<!--” y una marca final “-->”.

Ejemplo: <!-- Esto es un comentario en XML -->

Los comentarios son elementos especiales y no necesitan ninguna marca de cierre. Además hay que tener en cuenta que dentro de un comentario no se pueden usar dos guiones seguidos”--”.

Además, en XML se encuentran disponibles las secciones CDATA, que permiten marcar un texto para que éste no sea procesado por el *parser*, es decir, no serán analizadas sintácticamente. CDATA proviene de “*Character DATA*” (datos de carácter) en contraposición a datos de marcado. La sintaxis de estas secciones se basa en la etiqueta de inicio “<![CDATA[“ y la etiqueta de fin “]]”.

Documentos XML bien formados

Una vez visto los elementos que pueden formar parte de un documento XML y sus características, el siguiente paso será establecer cuando un documento es correcto. En este sentido, en XML se puede hablar de documentos “bien formados” y documentos “válidos”.

Los documentos bien formados son aquellos que son sintácticamente correctos, es decir, que cumplen las reglas expuestas en los apartados previos. Sin embargo, los documentos válidos son aquellos que, además de estar bien formados, cumplen los requisitos de una definición de estructura.

Estructura general de un documento

Un documento XML consta de tres partes que son:

- cabecera
- declaración de la estructura que debe cumplir el documento
- elemento raíz

CABECERA

Esta parte del documento sirve para establecer unos parámetros globales al documento que permitirán su uso a nivel global en la red.

La cabecera es opcional y también sus componentes que si aparecen son como sigue:

<?xml version=”1.0” encoding=”UTF-8” standalone=”yes”?>

- el primero establece la versión del XML a utilizar, la 1.0.
- el segundo establece el juego de caracteres usado en el resto del documento.
- el tercero establece si el documento es “tal cual” (valor “yes”) o si su estructura debe ser comprobada en base a un fichero externo que la marque (valor “no”, valor por defecto).

DECLARACIÓN DE LA ESTRUCTURA QUE DEBE CUMPLIR EL DOCUMENTO

Esta declaración servirá para establecer cual es la estructura que debe tener un documento.

Esta sección es opcional y si no aparece significará que el documento sólo debe ser correcto frente a las reglas generales de XML, ya enunciadas y al formato general que estamos describiendo.

En caso de aparecer, esta sección puede tener tres distintos tipos de contenidos aunque siempre se ajustará al siguiente formato:

<!DOCTYPE nombre_elemento_raiz estructura>

Este componente tiene por nombre fijo DOCTYPE y va seguido de un nombre de elemento (definido por el usuario) que se creará siguiendo las normas definidas para estos nombres. Este nombre de elemento es el nombre del elemento raíz que debe tener el documento para ser considerado correcto. A continuación tendremos que incluir la estructura que queremos para el documento siguiendo unas normas que veremos más adelante.

Los tres tipos de definición DOCTYPE se diferencian por el contenido que tenemos en la estructura, siendo:
Definición de estructura interna. Este es el caso más sencillo, en el cual las definiciones que marcarán la estructura del documento se realizan de forma local en el mismo elemento DOCTYPE.

Definición de estructura externa. En este caso, las definiciones se almacenan en un fichero externo. Para poder usar este tipo necesitamos incluir en la cabecera la declaración standalone=”yes”. En el elemento DOCTYPE tendremos que referirnos al fichero que contiene las definiciones.

Definición de estructura externa e interna. Esta es una combinación de las anteriores en la que tenemos referencia a un fichero externo de definiciones y además definiciones internas a continuación. Es importante tener presente que las definiciones internas se imponen a las externas.

Ejemplo de los tres tipos de definiciones:

<?xml version=”1.0”?>
<!DOCTYPE raiz[...]>
<raiz> la la la (/raiz>

<?xml version=”1.0”?>

<!DOCTYPE raiz SYSTEM “hola.dtd”>
<raiz> la la la (/raiz>

<?xml version=”1.0”?>
<!DOCTYPE raiz SYSTEM “hola.dtd” [   ]>
<raiz> la la la (/raiz>

Espacios de nombres

Un espacio de nombres se define como una referencia IRI (*Uniform Resource Identifier*), que servirá para identificar los elementos que pertenecen a dicho espacio de nombres. Otra forma de verlo es que los elementos tendrán un nombre compuesto por dos partes: una primera con su nombre y una segunda con el nombre de espacio de nombres. Este nombre compuesto permitirá identificar de forma unívoca al elemento en cuestión y de esa forma conocer siempre a qué elemento se está refiriendo el documento.

La construcción de estos nombres extendidos se realiza uniendo el nombre del espacio de nombres y el nombre del elemento o atributo, usando como conector el símbolo “:”. Sin embargo, las referencias pueden ser largas, lo que va en detrimento de la legibilidad y claridad del documento, además de propiciar que se cometan errores más fácilmente. Igualmente, las URIs pueden contener caracteres no válidos en XML. Para solucionar este problema, en XML se puede asignar un sinónimo corto al espacio de nombres de forma que este sinónimo corto sea el que se use a lo largo del documento. El sinónimo se asignará usando el separador “:” y la etiqueta “xmlns”. En realidad, “xmlns” es un atributo reservado (recordamos que los atributos no pueden comenzar por “xml” en ninguna combinación de mayúsculas y minúsculas).

Ejemplo:
```
<elementoej xmlns:enej="http://dominioej.com/rutaej">
  <enej:elemento1>Texto 1</enej:elemento1>
  <enej:elemento2>Texto 2</enej:elemento2>
</elementoej>
```

Observamos, en el ejemplo anterior, que el sinónimo corto del espacio de nombres es “enej” y que su uso resulta más adecuado que el nombre completo del espacio de nombres "http://dominioej.com/rutaej". Además los elementos “elemneto1” y “elemento2” pertenecen al espacio de nombres “enej” al estar calificados con el sinónimo de dicho espacio.

ESPACIO DE NOMBRES POR DEFECTO

Si un espacio de nombres se declara sin su sinónimo correspondiente esto indicará que todos los elementos (incluido el elemento que declara el espacio de nombres) que contenga pertenecerán a dicho espacio de nombres. Por tanto, sería como definir un espacio de nombres por defecto para los elementos que no tengan espacio de nombres asignado.

Otro uso de los espacios de nombres que puede resultar de gran utilidad es dejar su declaración en blanco (xmlns=””), lo que indicaría que los elementos y atributos contenidos, por defecto, no pertenecen a ningún espacio de nombres. Por último, cuando se declara un espacio de nombres por defecto, o sin espacio de nombres por defecto (xmlns=””), pero un elemento contiene un prefijo de un espacio de nombres, el espacio de nombres que prevalecerá será éste último.

Bibliografía:

Zurdo, J.S. et al; "*Lenguajes de Marcas y Sistemas de Gestión de la Información*"; ed: Ra-Ma (2011).

Gutierrez Gallardo, J. D.; *Manual imprescindible de XML*; Anaya Multimedia, (2010).