XML

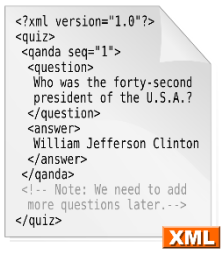
LIMA UD3 – Tema 1

IES Plurilingüe Antón Losada Diéguez

Tabla de contenido

XML

1. Introducción



XML (eXtensive Markup Language) es un lenguaje de marcas de propósito general usado principalmente para la transmisión de la información.

A diferencia de otros lenguajes de marcas, XML no tiene unas etiquetas propias, simplemente dispone de unas reglas para crear dichas etiquetas. Esto permite crear unas estructuras propias que se adapten a las necesidades del sistema en concreto.

Al igual que la mayoría de los lenguajes de marcas, la estructura es jerárquica, con los elementos unos dentro de otros representando un árbol con padres e hijos.

2. Reglas para un XML bien formado

Los XML tienen unas reglas básicas obligatorias de todo documento sin las cuales no funciona.

2.1. Prólogo

Lo primero que ha de aparecer en un documento XML es la línea llamada prólogo, que indica la versión de XML a utilizar y la codificación de caracteres a utilizar.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Aunque existe una versión 1.1 que añadió algunas mejoras a nivel de caracteres permitidos, su uso es muy escaso, ya que gran parte de esas mejoras se añadieron en revisiones posteriores de la versión 1.0.

2.2. Etiquetas

Las etiquetas de XML son muy similares a las de HTML anteriormente vistas, con la salvedad de que siempre tienen que cerrarse.

Toda etiqueta está conformada por una etiqueta de apertura y una de cierre o una de autocierre si no tiene contenido, pero nunca una de apertura solo.

```
<correo>...</correo>
```

Este ejemplo representa una etiqueta con contenido.

```
<correo/>
```

Aquellos elementos carentes de contenido se pueden estipular como una etiqueta de autocierre como la del ejemplo.

2.2.1. Nomenclatura

El nombre de la etiqueta está decidida por su creador, pero ha de seguir unas reglas:

- Distingue entre mayúsculas y minúsculas, *<Libro>* y *<libro>* son elementos distintos.
- Se pueden usar letras y números, pero el nombre ha de empezar siempre por una letra.
- Admite acentos y letras de todos los idiomas. Algunos lenguajes y lectores no los admitan, por lo que no suele ser recomendable su uso.
- Permite el uso de los caracteres '.' '-' y '_'.
- No puede empezar por xml en ninguna combinación de minúsculas y mayúsculas.

2.3. Contenido

El contenido de un elemento XML puede estar vacío, ser texto, otras etiquetas o una combinación de texto y etiquetas, también denominado contenido mixto.

```
<frutas>
    <pera>Fruta del género Pyrus</pera>
    <manzana>Fruta del género Malus domestica</manzana>
</frutas>
```

En este ejemplo puede verse que el elemento frutas contiene otros elementos y cada elemento fruta contiene solo texto.

En el caso del texto, este no puede tener los símbolos '<', '>' o '&'. Para poder utilizarlos, es necesario utilizar los códigos Unicode o los códigos de escape, de la misma forma que se vio en el tema de HTML básico.

2.4. Anidamiento

Se denomina anidamiento de etiquetas cuando unas se encuentran dentro de otras formando una estructura jerárquica.

Esta estructura debe respetar el orden de apertura y cierre de las mismas, siempre cerrando en orden inverso al que se crearon, de dentro a fuera.

```
<texto>
    <i>Hola gente</i>
</texto>
```

Correcto

```
<texto>
    <i>Hola gente</texto>
</i>
```

Incorrecto

2.5. Atributos

Todo elemento puede tener un número ilimitado de atributos, que se estipulan con *nombre="valor"* dentro de la etiqueta de apertura del elemento. Las comillas pueden ser simples '' o dobles "", pero usando las mismas para la apertura y el cierre.

Un mismo elemento no puede tener dos atributos iguales, pero sí puede repetirse un mismo atributo en distintos elementos.

```xml
<informe dia="2">
    <vendido>35 frutas</vendido>
    <ganado moneda="euro" simbolo="€">40</ganado>
</informe>
```

En este ejemplo se pueden ver varios elementos con atributo, incluyendo uno con dos atributos.

Los nombres de los atributos han de respetar las mismas reglas que los nombres de las etiquetas.

A mayores, no pueden contener las mismas comillas que engloban el valor, por lo que si es necesario utilizar una, hay que usar la otra para englobar.

```xml
<pasta tipo="Alfredo's"/>
```

2.6. Elemento raíz

Todo documento XML tiene lo que se denomina un elemento raíz, un único elemento del que salen los demás.

Habitualmente este elemento está nombrado como la entidad que representa el fichero, el significado del mismo.

```xml
<ordenadores>
    <ordenador>...</ordenador>
    <ordenador>...</ordenador>
</ordenadores>
```

Como se puede ver en el ejemplo, otras veces, el nombre es el plural de lo que agrupa, el conjunto de elementos que representa el documento.

2.7. Comentarios

Los comentarios son un tipo de etiqueta especial que permite añadir información para el lector del archivo y que no forma parte del documento.

```xml
<!-- Comentario -->
```

Este comentario es idéntico al del HTML, poniendo el texto entre los '--'.

2.8. Sección CDATA

El uso de elementos de escape de caracteres puede ser algo tedioso en ocasiones, además de hacer poco legible el documento sin el uso de un transformador.

Para ello existe la sección CDATA, en la que se ignoran todos los caracteres propios de la estructura de XML y se toma todo como texto plano.

```xml
<![CDATA[ ... ]]>
```

3. Estructura basada en etiquetas o basada en atributos

Existen dos vertientes fundamentales a la hora de crear un documento XML, dos ideas fundamentales, el representar los datos basándose principalmente en etiquetas o usando etiquetas y atributos de forma casi indistinta.

3.1. Basada en atributos

Este formato se basa en la idea de que las cualidades y propiedades de un elemento se guarden como atributos del mismo y los elementos queden reservados a las entidades.

```
<ordenador so="Windows"
           ram="16GB"
           grafica="Nvidia"
           cod="DAM1-03"
           />
```

Hay que tener en cuenta que los atributos no pueden repetirse, por lo que, si una cualidad se repite, obligatoriamente habría que usar un elemento.

3.2. Basada en elementos

Esta vertiente defiende la idea de que los atributos han de usarse solo para representar algo adicional, un metadato, un identificador del elemento, algo que no sea información relevante.

Por tanto, los atributos serían cosas que utilizaría un lenguaje de alto nivel, por ejemplo, para representar la información que contiene el elemento o para indicar las unidades, o el tipo de moneda, por ejemplo.

```
<ordenador cod="DAM1-03">
    <so>Windows</so>
    <so>Ubuntu</so>
    <ram>16 GB</ram>
    <grafica>Nvidia</grafica>
</ordenador>
```

En este ejemplo se puede ver la información de un ordenador y como, al usar elementos en vez de atributos, puede tener varios elementos repetidos.

3.3. Vertiente principal

Aunque no hay una forma estándar, hay cierta tendencia a usar la segunda, pero todo dependerá del sistema que use el documento y lo que se estipule, así como las necesidades en cada caso.

En el módulo se usará la segunda forma, ya que ofrece más libertad y más manejo desde lenguajes de alto nivel.

4. Espacio de nombres. Namespace

Como no hay un estándar definido y cada programador define sus propias etiquetas, puede ocurrir que los nombres coincidan para dos partes distintas del sistema, lo que podría acarrear conflictos si se intentan aunar documentos.

```xml
<empleado>
    <nombre>Manolo</nombre>
    <rango>Senior</rango>
    <departamento>Robótica</departamento>
</empleado>
 <proyecto>
     <nombre>Robodog</nombre>
     <inicio>5/6/2010</inicio>
     <presupuesto>5.000.000</presupuesto>
 </proyecto>
```

En este ejemplo se puede observar como en el documento de empleados de recursos humanos y en el de proyectos, existe una etiqueta nombre. Si estos dos XML se combinasen en uno único, por ejemplo que represente cada empleado y los proyectos en los que participa, habría dos elementos nombre con significados completamente distintos, lo que podría ocasionar problemas a la hora de manipular los datos.

Para resolver el problema, se utilizan los espacios de nombres, unos nombres que se estipulan en los elementos de un documento determinado que los identifican unequívocamente.

Estos se establecen mediante el atributo *xmlns* (XML NameSpace) en la etiqueta padre de los elementos que representa. El valor de este atributo es una URI y, habitualmente, se usa una URL del dominio de la empresa. Este enlace no tiene que existir, simplemente es un identificador.

```xml
<empleado xmlns="http://www.example.com/empleado">
    <nombre>Manolo</nombre>
    <rango>Senior</rango>
    <departamento>Robótica</departamento>
</empleado>

<proyecto xmlns="http://www.example.com/proyecto">
    <nombre>Robodog</nombre>
    <inicio>5/6/2010</inicio>
    <presupuesto>5.000.000</presupuesto>
</proyecto>
```

En este ejemplo se establecen dos espacios de nombre, empleado para todos los datos del empleado y proyecto.

4.1. Espacio de nombres con prefijo

En caso de necesitar usar distintos espacios de nombre en una etiqueta o, si se considera que puede quedar más legible, se puede estipular un prefijo a utilizar en todos los elementos que lo representan.

```xml
<trabajos xmlns:e="http://www.example.com/empleado"
xmlns:p="http://www.example.com/proyecto">
    <e:empleado>
        <e:nombre>Manolo</e:nombre>
    </e:empleado>

    <p:proyecto>
        <p:nombre>Robodog</p:nombre>
    </p:proyecto>
</trabajos>
```

En este ejemplo se puede ver la declaración de dos espacios de nombre con prefijo en un elemento raiz, uno para empleados y otro para proyectos

Una vez declarados, estos han de ponerse en los elementos que forman parte de ese espacio de nombres, anteponiendo el prefijo al nombre de la etiqueta de apertura y de cierre.

Los prefijos se pueden utilizar desde el mismo elemento en el que se declaran y todos sus hijos.

Ambos tipos de namespace se pueden combinar para un mismo conjunto de elementos, pero con una consideración, los que no tienen prefijo se usan como identificador de los elementos a los que no se le estipule ningún prefijo y solo puede haber un namespace de este tipo por elemento.