Unidad 2. XML

1. INTRODUCCIÓN

XML es un **metalenguaje**, de propósito general desarrollado por la **W3C** y la base para la construcción de otros lenguajes más específicos. Define las reglas de construcción de los documentos.
Se puede utilizar para **intercambiar información entre sistemas**, guarda **configuraciones de una aplicación** o difundir información mediante **RSS**.

Fue **lanzado en 1998** y de **formato abierto**. La **extensión** es **.xml**.

Tiene fundamentos robustos, una sintaxis muy simple. Se utiliza para, crear ficheros de configuración, diseño de interfaces gráficas, publicación de contenidos, etc.
Es una **herramienta poderosa** por las tecnologías que tiene alrededor, como validadores, transformadores, editores, etc.

2. ESTRUCTURA Y SINTAXIS

Los documentos XML constan de dos partes:
- **Prólogo**: contiene la información relativa al conjunto del documento.
- **Cuerpo**: recoge los elementos con la información propiamente dicha.
- **Comentarios**: es opcional y deben empezar por **<!--** y terminar por **-->**

En el **prólogo** se **declara** la **versión** del XML, el **encoding**, que es la codificación utilizada y el **standalone**, que indica si el documento es independiente (yes) o no (no) a un DTD (Document Type Definition)

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

El **cuerpo** del documento está formado por los **elementos**, que son el componente fundamental de XML.
Los elementos están formados por una etiqueta de apertura y, opcionalmente, una de cierre y deben tener el mismo nombre. Además pueden tener o no uno o más atributos.

<nombre-elemento nombre-atributo=”valor1” nombre- atributo-2=”valor2”></nombre-elemento>

Las **reglas** para la asignación de nombres son:
- Diferencian entre mayúsculas y minúsculas.
- Deben comenzar con una letra o un guion bajo.
- Los nombres de elementos son idénticos en las etiquetas de apertura y cierre.
- No pueden contener espacios.

Las **reglas** de creación de atributos son:
- Deben tener asignado un valor.
- Los valores siempre van entrecomillados.
- Diferencian entre mayúsculas o minúsculas.
- Deben comenzar con una letra o guion bajo.
- Pueden estar formados por caracteres alfanuméricos, guiones bajos y puntos.

En el **DTD** están definidas las **entidades**. Estas entidades se utilizan para representar información haciendo referencias de ella en lugar de incluirlas directamente. Esta fragmentación aporta mejor estructura y facilita el trabajo en grupo.

El **CDATA** es una sección que contiene un conjunto de caracteres que no debe ser tratado por el validador.

3. VALIDACIÓN DE XML

Para que un XML esté **bien formado** tiene que cumplir las siguientes reglas:
- Debe haber uno y solo un elemento raiz.
- Todos los elementos deben estar cerrados.
- Los elementos tienen que estar anidados correctamente.
- Todos los valores de los atributos están entrecomillados.
- Los nombres de elementos y atributos han de cumplir con sus respectivas reglas.

El **DTD** interno se forma empezando con:

<**!DOCTYPE nombre-elemento-raiz[**
Elementos-y-sus-relaciones
]>

Si el **DTD** es externo tendremos que decirle a nuestro documento dónde mirar, eso se hace de la siguiente manera:

<!**DOCTYPE nombre-elemento-raiz SYSTEM "nombre-archivo-externo">**

Si declaramos una entidad en nuestro DTD, podemos poner:

<!ENTITY nombre-entidad "texto de reemplazo">

Esto hará que si en nuestro documento utilizamos **&nombre-entidad**, transformará ese código en el texto de reemplazo que hemos mencionado anteriormente.

Los elementos se declaran de la siguiente forma:

<!**ELEMENT nombreElemento (contenido)>**

El contenido de nuestro elemento puede contener los siguientes tipos:
- **EMPTY**: debe estar vacío.
- **ANY**: puede contener cualquier contenido.
- **#PCDATA**: puede tener datos de tipo carácter.
- **nombreElemento**: puede contener el elemento indicado

El **nombreElemento** puede tener varios usos:
- **?**: puede contener ninguna o solo una ocurrencia.
- +: puede contener una o más ocurrencias.
- *: puede contener ninguna o más ocurrencias.
- **nombreElemento1, nombreElemento2**: debe contener todos los elementos mencionado.
- **nombreElemento1 | nombreElemento2**: debe contener uno u otro elemento.

Los **atributos** también forman parte de nuestro DTD y la sintaxis es la siguiente:

<!ATTLIST nombreElemento nombre-atributo tipo-atributo valor-atributo>

Los **tipos** de atributos son los siguientes:
- **CDATA**: cadena de caracteres.
- **(valor1 | valor2 | …)**: lista de posibles valores.
- **ID**: un identificador único.
- **IDREF**: una referencia a un identificador único de otro elemento.
- **IDREFS**: lista de referencias separadas por espacios a identificadores de otros elementos.
- **NMTOKEN**: un nombre XML válido.
- **ENTITY**: una referencia a una entidad.

Los **valores** que le damos a los atributos son:
- **valor**: valor por defecto del atributo.
- **#REQUIRED**: indica que el atributo es obligatorio.
- **#IMPLIED**: el atributo es opcional.
- **#FIXED valor**: fija el valor del atributo.

El **XML Schema** es el lenguaje utilizado para describir la estructura, relaciones y restricciones de los documentos XML. Utiliza la extensión **.xsd** ya que su nombre técnico es XML Schema Definition.

Con el XSD se consigue un mayor nivel de **precisión** en el establecimiento de reglas de validación.
La primera versión fue en 2001 por la W3C.

Se creó para cubrir las carencias de los DTD.

Su **estructura** sería la siguiente:

<?xml version=”1.0” encoding= “UTF-8”?>
<xs:schema xmlns:ns=”http://www.w3.org/2001/XMLSchema”>

Existen **dos** tipos de elementos:
- **simpleType**: marca que un elemento es simple “solo texto”.
- **complexType**: indica que el elemento es complejo y contiene atributos, secuencias, etc “es padre de”.

Siendo **name** el atributo principal de todo elemento, existen además varios atributos que vamos a utilizar:
- **type**: el tipo de dato.
- **default**: permite asignar valores por defecto.
- **fixed**: determina el valor del atributo en caso de que éste exista.
- **minOccurs**: numero mínimo inclusive de ocurrencias del elemento.
- **maxOccurs**: numero máximo inclusive de ocurrencias del elemento.

En los casos de minOccurs y maxOccurs, para que sea **ilimitado** debemos poner el valor **“unbounded”**.

Los elementos pueden contener subelementos, de los cuales existen 3 tipos:
- **xs:sequence**: indica una secuencia de elementos obligatorios. Deben aparecer el el orden marcado.
- **xs:choice**: señala elementos alternativos. Solo debe aparecer uno.
- **xs:all**: indica una secuencia de elementos opcionales. No tienen que aparecer todos ni en el mismo orden.

Los elementos pueden tener **restricciones**, si dicho elemento solo tiene restricciones y no atributos, puede ser de tipo simpleType:

<xs:restriction base=”tipoDato”>
    <xs:nombreFaceta value=”valor”/>

A nombre de la restricción se le conoce como **facetas** y existen estos tipos:
- **xs:length**: determina una longitud fija.
- **xs:minLength**: establece una longitud mínima.
- **xs:maxLength**: establece una longitud máxima.
- **xs:totalDigits**: determina el numero máximo de dígitos que puede tener un numero.
- **xs:fractionDigits**: establece el máximo numero de decimales que puede tener un numero.
- **xs:minExclusive**: determina que el valor debe ser mayor al indicado.
- **xs:maxExclusive**: determina que el valor debe ser menos al indicado.
- **xs:minInclusive**: fija que el valor debe ser igual o mayor al indicado.
- **xs:maxInclusive**: fija que el valor debe ser igual o menos al indicado.
- **xs:enumeration**: establece una lista de valores posibles.
- **xs:whiteSpace**: determina cómo tratar los espacios en blanco, tabulaciones y saltos de lineas.
- **xs:pattern**: fija un patrón de caracteres permitidos.

En los XSD se puede indicar el **tipo de dato** preciso que contiene un elemento y existen varios tipos:
- **xs:string**: cadera de caracteres.
- **xs:integer**: números enteros.
- **xs:decimal**: números decimales, usando el punto como separador no la coma.
- **xs:boolean**: tipo de dato lógico, usando los valores “true” o “false”.
- **xs:date**: fechas en formato “AAAA-MM-DD”.
- **xs:time**: horas en formato “hh:mm:ss”.
- **xs:duration**: periodo de tiempo en formato PnYbMnDnHnMnS”.
    - **P**: inicio del periodo.
    - **nY**: numero de años.
    - **nM**: numero de meses.
    - **nD**: numero de días.
    - **T**: inicio del tiempo.
    - **nH**: numero de horas.
    - **nM**: numero de minutos.
    - **nS**: numero de segundos.