



El estándar XML

| | |
|-------------------------------|---|
| El estándar XML | Qué es XML y qué elementos lo componen |
| DOM | Acceso a los datos de un documento XML |
| XSL | Cómo dar formato a un documento XML con XSL |
| Aplicaciones Prácticas | Uso práctico de XML en entornos Java |



XML - Introducción

- **XML** (eXtended Markup Language) es un lenguaje que proporciona un formato para describir datos estructurados
- Es un lenguaje de marcas
- Es un subconjunto de **SGML** (Standard Generalized Markup Language)
- Definido por el W3C en 1998



XML - Características

- Lenguaje de marcas extensible, es un *metalenguaje*
- Es “case sensitive”
- Representación estructural de los datos en forma de árbol
- Define los datos en una aplicación, no su presentación
- Permite interoperatividad entre máquinas distintas (un estándar para el intercambio de datos)

XML - Ejemplo de documento XML

```
<?xml version="1.0"?>
<!DOCTYPE discos SYSTEM "cds.dtd">
<discos>
  <disco>
    <artista tipo="individual">Frank Sinatra</artista>
    <titulo numerodepistas="2">In The Wee Small Hours</titulo>
    <pistas>
      <pista>In The Wee Small Hours</pista>
      <pista>Mood Indigo</pista>
    </pistas>
    <precio>$12.99</precio>
  </disco>
  <disco>
    <artista tipo="band">The Offspring</artista>
    <titulo numerodepistas="2">Americana</titulo>
    <pistas>
      <pista>Welcome</pista>
      <pista>Have You Ever</pista>
    </pistas>
    <precio>$12.99</precio>
  </disco>
</discos>
```



XML - Elementos XML

- Un documento XML está formado por **tags** o **etiquetas** para la estructuración de los datos.

Por ejemplo: <disco>

- Un **elemento XML** está formado por una etiqueta de inicio y otra etiqueta final. Entre ambas está el valor del dato al que se refiere el tag.

En este ejemplo, el elemento pista tiene como valor "Welcome":

```
<pista>Welcome</pista>
```

XML - Elementos XML - Atributos

- Un elemento XML puede contener uno o más **atributos** en su tag inicial

En este ejemplo, el atributo sería "tipo", y su valor "individual":

```
<artista tipo="individual">Frank Sinatra</artista>
```

- Un elemento XML puede no contener un valor, solamente atributos. En este caso se puede escribir con un solo tag terminado en /.

Esta expresión

```
<precio valor="$12.99"></precio>
```

equivale a

```
<precio valor="$12.99"/>
```



XML - Documentos XML

Un **documento XML** es un conjunto de elementos XML anidados (con estructura de árbol)

Reglas a tener en cuenta:

- Todos los elementos deben cerrarse
- Los elementos deben estar limpiamente anidados
- Un documento XML debe tener un único elemento raíz



XML - Reglas de Construcción de Nombres XML

En la construcción de nombres XML (atributos y elementos XML) se deben tener en cuenta las siguientes reglas:

- Un nombre XML puede contener los siguientes caracteres: Letras (A-Z, a-z), dígitos (0-9), guión bajo (_), guión (-) y punto (.)
- Debe empezar con una letra o guión bajo
- Puede usar dos puntos (:) sólo en *namespaces*
- No puede contener espacios en blanco



XML - Documento XML Bien Formado

Se dice que un documento XML está **bien formado** si cumple las siguientes reglas:

- Todo tag debe tener su correspondiente tag final.
- Los elementos deben quedar debidamente anidados
- Debe haber exactamente un elemento raíz
- Los valores de los atributos deben estar entre comillas
- Un mismo elemento no debe tener más de un atributo con el mismo nombre
- Los comentarios no deben colocarse dentro de los tags
- Los caracteres & y < deben ser precedidos por un \ si forman parte de los datos de un elemento o atributo.

Validar un documento XML —————> *Con un Parser XML*

XML - Ejercicio 1

¿Es el siguiente documento XML un documento bien formado?

```
<empleados>
  <empleado nombre="Juan" apellido="Pérez" apellido="Sánchez">
    <direccion calle="López Peñalver">
  </empleado>
  <Empleado nombre="Alberto">
    <direccion calle="López Peñalver">
  </Empleado>
  <empleado>
    <localidad>Málaga</localidad>
  </Empleado>
</empleados>

<oficinas>
  <oficina 1tlf="951000000" 2tlf="951000001">
    <ciudad>Madrid</ciudad>
  </oficina>
  <oficina>
    <ciudad telefono="12312312">Barcelona</ciudad>
  </oficina>
</oficinas>
```

XML - Ejercicio 2

Crear un documento XML válido que contenga los siguientes datos:

| Vuelo | Origen | Destino | Salida | Llegada |
|--------------|---------------|----------------|---------------|----------------|
| IB777 | Málaga | Madrid | 8:00 | 8:50 |
| IB888 | Madrid | Barcelona | 9:00 | 10:10 |
| IB999 | Madrid | Málaga | 12:00 | 12:50 |

| Código Avión | Vuelos |
|---------------------|---------------|
| 1234 | IB777,IB999 |
| 1122 | IB888 |



XML - Namespaces

- Un **XML namespace** es una colección de nombres que pueden ser usados en elementos o atributos.
- Un namespace distingue nombres de forma única en la Web, para evitar conflictos.
- Un namespace puede ser identificado por una:
 - URI (Universal Resource Identifier)
 - URL (Uniform Resource Locator)
 - URN (Uniform Resource Number)

XML - Namespaces - Declaración

Se declara con el atributo xmlns en el elemento deseado.
Existen dos tipos de declaración:

- Declaración por defecto. No hace falta usar prefijo, todos los nombres pertenecen al namespace declarado:

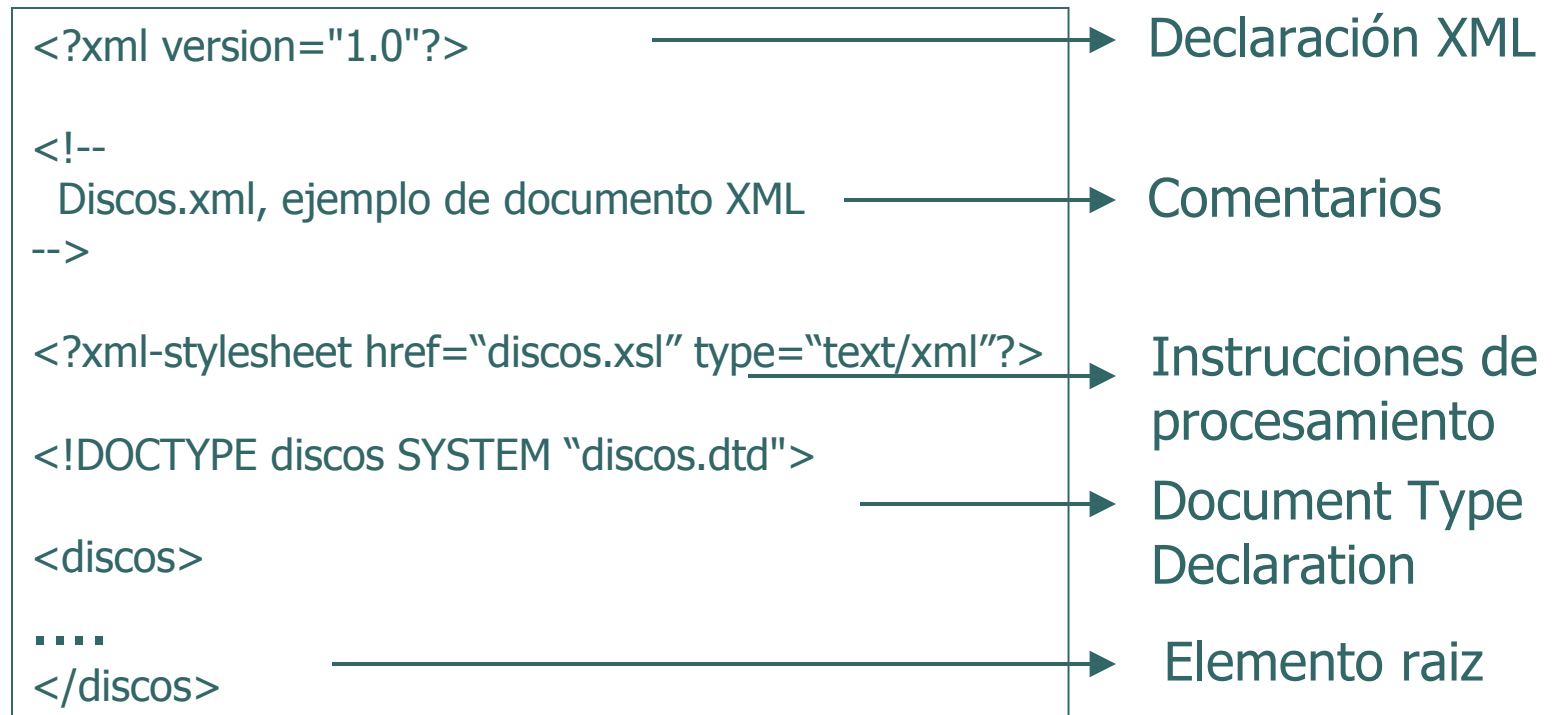
```
<BOOK xmlns="urn:BookLovers.org:BookInfo">  
  <TITLE>A Suitable Boy</TITLE>  
  <PRICE currency="US Dollar">22.95</PRICE>  
</BOOK>
```

- Declaración explícita. Cada nombre debe usar un prefijo especificando el namespace al que pertenece:

```
<BOOKS>  
  <bk:BOOK xmlns:bk="urn:BookLovers.org:BookInfo"  
    xmlns:money="urn:Finance:Money">  
    <bk:TITLE>A Suitable Boy</bk:TITLE>  
    <bk:PRICE money:currency="US Dollar">22.95</bk:PRICE>  
  </bk:BOOK>  
</BOOKS>
```

XML - Prólogo del Documento XML

Antes del elemento raíz del documento XML pueden aparecer una serie de elementos, formando la siguiente estructura:





XML - Prólogo del Documento XML

Declaración

- No es necesaria para que el documento sea bien formado
- Incluye los siguientes atributos:
 - version: versión de XML
 - encoding: codificación de los caracteres
 - standalone: especifica si no es necesaria una validación por DTD

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

Comentarios

- Pueden aparecer en cualquier sitio excepto dentro de un tag

```
<!--  
    Discos.xml, ejemplo de documento XML  
-->
```



XML - Prólogo del Documento XML

Instrucciones de procesamiento

- Asocia el documento XML con algún elemento externo, como por ejemplo una hoja de estilos xsl.

```
<?xml-stylesheet href="discos.xsl" type="text/xml"?>
```

Document Type Declaration

- Referencia a un DTD (Document Type Definition), el cual valida el documento XML.

```
<!DOCTYPE discos SYSTEM "discos.dtd">
```




XML - DTDs (Document Type Definition)

- Un **DTD** es un documento que especifica las restricciones que debe cumplir un documento XML.
- Un documento *válido* es aquel que cumple las restricciones especificadas en su DTD. Un documento es *inválido* si no las cumple o simplemente no referencia a ningún DTD.
- Permite la declaración de tres tipos de estructuras:
 - Elementos
 - Atributos
 - Entidades
- Debido a sus limitaciones se creó el lenguaje *XML Schema*.



XML - DTDs - Declaración

Un DTD se puede declarar de 2 formas:

- En un fichero aparte

```
<!DOCTYPE discos SYSTEM "discos.dtd">
```

- En el propio documento XML (en el prólogo):

```
<!DOCTYPE empleado  
[  
  <!ELEMENT empleado (nombre, direccion)>  
  <!ELEMENT nombre (#PCDATA)>  
  <!ELEMENT direccion (ciudad, calle)>  
  <!ELEMENT ciudad (#PCDATA)>  
  <!ELEMENT calle (#PCDATA)>  
]  
>
```

XML - DTDs - Declaración de Elementos

- Un Elemento se declara especificando su nombre y los elementos que lo componen:

```
<!ELEMENT empleado (nombre, direccion)>
```

- Si el elemento no contiene ningún otro elemento, se especifica el tipo del valor que contiene

```
<!ELEMENT nombre (#PCDATA)>
```

➤ #PCDATA permite caracteres alfanuméricos

- Así, el siguiente documento XML sería válido para el DTD anterior:

```
<empleado>  
  <nombre>Antonio</nombre>  
  <direccion>  
    <ciudad>Málaga</ciudad>  
    <calle>Larios</calle>  
  </direccion>  
</empleado>
```



XML - DTDs - Declaración de Elementos

- Si separamos con comas los elementos hijo, el orden de aparición de éstos debe cumplirse

```
<!ELEMENT empleado (nombre, direccion)>
```

- Si separamos los elementos con una barra (|), con que aparezca sólo uno de ellos ya se cumplirá la condición

```
<!ELEMENT empleado (nombre | direccion)>
```

- También pueden agruparse con paréntesis:

```
<!ELEMENT empleado ((nombre | apellido) , direccion)>
```



XML - DTDs - Declaración de Elementos

Modificadores

- Los siguientes modificadores especifican el número de elementos que deben aparecer:

| | |
|---|---------|
| * | 0 ó más |
| + | 1 ó más |
| ? | 0 ó 1 |

- Si no se muestra ningún modificador, debe aparecer exactamente 1 elemento.

Ejemplos:

```
<!ELEMENT empleado (nombre , apellido? , direccion+)>
```

```
<!ELEMENT empleado (nombre , apellido , (direccion | telefono)*)>
```



XML - DTDs - Declaración de Elementos

Elementos Vacíos

- Si un elemento no contiene valor, se declara con la palabra EMPTY

```
<!ELEMENT direccion EMPTY>
```

Contenido ANY

- Si un elemento puede tener cualquier tipo de valor o de elementos compuestos, se declara con la palabra ANY

```
<!ELEMENT otrosDatos ANY>
```

XML - DTDs - Ejercicio 1

Crear un DTD que valide la siguiente estructura:

```
<proveedores>
  <proveedor>
    <nombre>Pascal</nombre>
    <productos>
      <producto>Leche</producto>
      <producto>Flan</producto>
    </productos>
  </proveedor>
  <proveedor>
    <nombre>Capitán Pesconovo</nombre>
    <productos>
      <producto>Merluza</producto>
    </productos>
    <telefono>
      952888777
    </telefono>
  </proveedor>
</proveedores>
```

XML - DTDs - Declaración de Atributos

- Un elemento puede tener 1 o más atributos. Por ejemplo, para declarar los atributos *ciudad*, *calle* y *numero* del elemento *direccion*, se haría de la siguiente forma:

```
<!ATTLIST direccion
    ciudad CDATA #FIXED
    calle CDATA #REQUIRED
    numero CDATA #IMPLIED
>
```

```
<direccion ciudad="Málaga" calle="Larios"
numero="5"/>
```

- La expresión `#REQUIRED` indica que el atributo es obligatorio
- La expresión `#IMPLIED` indica que el atributo es opcional
- La expresión `#FIXED` indica que el atributo no es modificable
- La expresión `CDATA` es el tipo de dato (datos alfanuméricos)
- Se puede especificar un valor por defecto:

```
<!ATTLIST direccion pais CDATA "España">
```


XML - DTDs - Declaración de Atributos

Tipos de atributos

- Enumeración. Limita los posibles valores del atributo.

```
<!ATTLIST empleado sexo (hombre | mujer) #REQUIRED>
```

- Tipos de atributos simple.
 - CDATA. Caracteres alfanuméricos
 - ID. Identificador único para un elemento.
 - IDREF. Referencia a un número identificador contenido en el atributo ID de algún otro elemento

```
<!ATTLIST empleado  
  empid ID #REQUIRED  
  supervisor IDREF #IMPLIED  
>
```

```
<empleado empid="A120">  
</empleado>  
<empleado empid="A240" supervisor="A120">  
</empleado>
```

XML - DTDs - Declaración de Entidades

- Una **entidad** es una asociación entre el nombre de la entidad y su valor. La entidad puede aparecer en el valor de un elemento o atributo, y será sustituido por su valor.
- Hay 5 entidades predefinidas:
 - < (<)
 - & (&)
 - > (>)
 - " (")
 - ' (')
- Se pueden definir entidades nuevas:

```
<!ENTITY empresa "Coritel BPM">
```

y usarlas de la siguiente forma

```
<empleado empresa="&empresa;">
```

XML - DTDs - Ejercicio 2

Crear un DTD que valide la siguiente estructura:

```
<almacen empresa="&empresa;">
  <productos>
    <producto id="A191">
      <nombre>Teclado</nombre>
      <precio valor="20" moneda="euros"/>
      <caracteristicas>15 pulgadas</caracteristicas>
    </producto>
    <producto id="A167">
      <nombre>Monitor 14'</nombre>
      <precio valor="20000" moneda="pesetas"/>
    </producto>
    <producto id="A180" nohay="A167">
      <nombre>Monitor 15'</nombre>
      <precio valor="20000" moneda="pesetas"/>
    </producto>
  </productos>
</almacen>
```