



## Unidad 4

XML: creación, validación y utilización

### CONTENIDOS

- 4.1. Introducción, evolución y estado actual**
- 4.2. Estructura y sintaxis de XML**
- 4.3. Validación de XML**
- 4.4. XML aplicado**
- 4.5. Herramientas de visualización y edición de XML**

### 4.1. Introducción, evolución y estado actual

- XML (eXtensible Markup Language, “lenguaje de marcas extensible”, en castellano).
- XML es un metalenguaje (permite construir otros lenguajes y dialectos) de propósito general.
- XML no define etiquetas: define reglas.
- Tiene como orígenes los lenguajes GML y SGML.
- Características:
  - Sencillo.
  - Robusto.
  - Formato abierto (no tiene coste).



- Las versiones de XML son gestionadas por el W3C

Año	Versión	Observaciones
1998	XML 1.0	Primera edición
2000	XML 1.0	Segunda edición
2004	XML 1.0	Tercera edición
2004	XML 1.1	Primera edición
2006	XML 1.0	Cuarta edición
2006	XML 1.1	Segunda edición
2008	XML 1.0	Quinta edición

Figura 4.1. Versiones de XML.

## 4.2. Estructura y sintaxis de XML

- Un documento XML tiene la siguiente estructura:
  - Declaración.
    - Versión.
    - Codificación.
    - Indicador de dependencias.

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

- Contenido.
  - Elemento raíz.
  - Resto de elementos.

- Un documento XML es una estructura en forma de árbol.
- Contiene una raíz.
- De la raíz descienden elementos que a su vez pueden ser contenedores de otros elementos.



Figura 4.2. XML representa una estructura en forma de árbol.

- **Las relaciones entre los elementos de un documento XML son las siguientes:**
  - 1. Existe un único elemento raíz (root).**
  - 2. Cada uno de los elementos descendientes directos de un elemento se llama hijo (children).**
  - 3. El elemento ascendiente directo de un elemento se llama padre (parent).**
  - 4. Los elementos que tienen un padre común se denominan hermanos (sibling).**

- Los elementos tienen una etiqueta de apertura, una etiqueta de cierre y, opcionalmente, uno o más atributos.
- Los atributos están compuestos de pares parámetro-valor.

```
<nombre-elemento atributo-1="valor1" atributo-2="valor2" >  
</nombre-elemento>
```

- Los elementos pueden estar vacíos, contener texto o a otros elementos.



- **Reglas para asignar nombre a los elementos o etiquetas:**
  1. **Diferencian entre mayúsculas y minúsculas.**
  2. **Deben comenzar con una letra o un guion bajo.**
  3. **Los nombres de elementos tienen que ser idénticos en las etiquetas de apertura y de cierre.**
  4. **Los nombres pueden estar formados por caracteres alfanuméricos, guiones, guiones bajos y puntos.**
  5. **Los nombres no pueden contener espacios (interpreta como nombre del elemento la primera palabra y el resto como parámetros)**

- **Reglas para la creación de atributos:**
  1. **Deben tener asignado un valor.**
  2. **Los valores siempre van entrecomillados, admitiendo comillas simples y dobles (deben coincidir el tipo de comilla de cierre con el de apertura).**
  3. **Diferencian entre mayúsculas y minúsculas.**
  4. **Deben comenzar con una letra o un guion bajo.**
  5. **Pueden estar formados por caracteres alfanuméricos, guiones, guiones bajos y puntos.**

```
<?xml version="1.0" encoding="UTF-8"?>
<animales>
  <animal id="1">
    <nombre>Tiburón blanco</nombre>
    <nombrecientifico>Carcharodon
    carcharias</nombrecientifico>
  </animal>
  <animal id="2">
    <nombre>Atún rojo</nombre>
    <nombrecientifico>Thunnus thynnus</nombrecientifico>
    extinción</estadodeconservacion>
  </animal>
</animales>
```

Figura 4.3. Ejemplo de documento XML.

- **Comentarios.**
  - **Contienen información que no es procesada ni interpretada.**
  - **Comienzan por <!--**
  - **Terminan por -->**

**<!-- Este texto es un comentario -->**

- **Espacios de nombres.**
  - **Permiten resolver las ambigüedades en caso de que varios elementos tengan el mismo nombre.**
  - **Declaración:**  
`xmlns:nombre-espacio-nombres=URL`
  - **Uso:**  
`<nombre-espacio-nombres:nombre-elemento>`
  - **Ejemplo:**  
`xmlns:autorestecnicos='http://unaurlcualquiera';`  
`<autorestecnicos:autor>`

- **Entidades:**
  - **Son un mecanismo para representar información dentro de un documento mediante referencias en lugar de incluirla directamente.**
  - **Tipos:**
    - **Generales**
      - **Internas. Contienen el valor en el propio documento.**
      - **Externas. Contienen el valor en un documento externo.**
    - **De parámetros**

- **Entidades internas predefinidas:**

Entidad	Significado	Carácter
<b>&amp;lt;</b>	Símbolo de “menor que”	<
<b>&amp;gt;</b>	Símbolo de “mayor que”	>
<b>&amp;amp;</b>	Símbolo del ampersand (en español et)	&
<b>&amp;apos;</b>	Comilla simple	'
<b>&amp;quot;</b>	Comilla doble	“

**Figura 4.4. Entidades predefinidas.**

- **Sección CDATA:**
  - **Contiene un conjunto de caracteres que no debe ser tratado por el analizador.**
  - **Sintaxis:**
    - **Comienzo:** `<![CDATA[`
    - **Fin:** `]]>`



- Sección CDATA:

- Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<ficha>
  <nombre>Lenguaje C</nombre>
  <info-html>
    <![CDATA[
      <h1>LENGUAJES DE PROGRAMACIÓN</h1>
      <hr>
      <a href="url-enlace">C</a>
    ]]>
  </info-html>
</ficha>
```

### 4.3. Validaciones de XML

- **XML dispone de mecanismos de validación.**
- **Dos perspectivas:**
  - **XML bien formado. Es conforme a las reglas generales de XML.**
  - **XML válido. Está bien formado y conforme a las reglas definidas por el creador o diseñador del documento.**

- **XML bien formado. Reglas:**
  - **Debe haber uno y solo un elemento raíz.**
  - **Todos los elementos deben estar cerrados.**
  - **Los elementos deben estar anidados correctamente: no se pueden intercalar aperturas y cierres de elementos distintos.**
  - **Todos los valores de los atributos están entrecomillados.**
  - **Los nombres de elementos y atributos deben cumplir con sus respectivas reglas.**

- **XML válido. Alternativas:**
  - **Alternativa 1: Debe cumplir las reglas especificadas en un DTD.**
  - **Alternativa 2: Debe cumplir las reglas especificadas en un XML-Schema.**

- **DTD.**
  - ***Document Type Definition*** (Definición de Tipo de Documento).
  - **Contiene una serie de reglas que sirven para validar un documento XML.**
  - **Alternativas de uso:**
    - **DTD interno. Incluido en el propio documento XML.**
    - **DTD externo. Almacenado en un fichero externo y referenciado desde el documento XML.**

- DTD interno. Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE aviso [
  <!ELEMENT aviso (de,para,mensaje)>
  <!ELEMENT de (#PCDATA)>
  <!ELEMENT para (#PCDATA)>
  <!ELEMENT mensaje (#PCDATA)>
]>
<aviso>
  <de>David</de>
  <para>Rosalía</para>
  <mensaje>Mañana nos vemos en el estudio a las 10.</mensaje>
</aviso>
```

- DTD externo. Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE aviso SYSTEM "dtdExterno.dtd">
<aviso>
    <de>David</de>
    <para>Rosalía</para>
    <mensaje>Mañana nos vemos en el estudio a las 10.</mensaje>
</aviso>
```

- **DTD. Entidades.**
  - En un DTD, una entidad es el equivalente a una constante en programación.
  - **Declaración:**

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

- **Uso:**

```
&nombre-entidad;
```



- DTD. Entidades.

- Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fichas [
  <!ENTITY poblacion "Tejeda de Tiétar, Cáceres">
]>
<fichas>
  <ficha>
    <empresa nombre="Tractores Toledo"></empresa>
    <direccion>&poblacion;</direccion>
  </ficha>
</fichas>
```

- DTD. Entidades.

- Declaración entidad externa privada:

```
<!ENTITY provincia SYSTEM "provincia.txt">
```

- Uso:

```
<poblacion>&poblacion;</poblacion>
```

- **DTD. Anotaciones:**
  - Se utilizan para identificar el formato de entidades que no son XML y que, por lo tanto, no se van a procesar o como valor válido para un atributo.
- **DTD. Elementos:**
  - Un elemento en un DTD indica en qué condiciones debe encontrarse dicho elemento en el documento XML que se está validando.

```
<!ELEMENT nombre-elemento contenido>
```

- **DTD. Elementos. Tipos de contenido.**

Tipo de contenido	Descripción
<b>EMPTY</b>	Indica que el elemento referenciado debe estar vacío.
<b>ANY</b>	Indica que el elemento referenciado puede contener cualquier contenido.
<b>(#PCDATA)</b>	Indica que el elemento referenciado puede datos de tipo carácter (Parsed Character Data).
<b>(nombreElemento)</b>	Indica que el elemento referenciado puede contener al elemento indicado.
<b>(nombreElemento1, nombreElemento2, ...)</b>	Indica que el elemento referenciado puede contener una sucesión de elementos indicados como una lista separada por comas.

**Figura 4.5. Tipos de contenido de los elementos de DTD.**

- **DTD. Elementos. Cardinalidades.**

Notación	Descripción	Ejemplo
<b>(nombreElemento)</b>	Una única ocurrencia del elemento.	<!ELEMENT aviso (de)>
<b>(nombreElemento?)</b>	Cero o una única ocurrencia del elemento.	<!ELEMENT aviso (de?)>
<b>(nombreElemento+)</b>	Una o más ocurrencias del elemento.	<!ELEMENT aviso (mensaje+)>
<b>(nombreElemento*)</b>	Cero o más ocurrencias del elemento.	<!ELEMENT aviso (mensaje*)>
<b>(nombreElemento1, nombreElemento2,...)</b>	Debe contener todos los elementos de la lista.	<!ELEMENT aviso (de, para, mensaje)>
<b>(nombreElemento1   nombreElemento2)</b>	Debe contener uno u otro elemento	<!ELEMENTO aviso (de   para, mensaje)>

**Figura 4.6. Cardinalidades de los elementos de DTD.**

- **DTD. Atributos.**
  - Los DTD permiten restringir las reglas y condiciones de los atributos usados en los documentos XML que validan.
  - Sintaxis:

```
<!ATTLIST elemento nombre-atributo tipo-atributo valor-atributo>
```

- **DTD. Tipos de atributos.**

Tipo	Descripción
CDATA	Cadena de caracteres.
(valor1   valor2   ...)	Una lista de posibles valores.
ID	Un identificador único
IDREF	Una referencia a un identificador único de otro elemento
IDREFS	Una lista de referencias a identificadores de otros elementos
NMTOKEN	Un nombre XML válido.
NMTOKENS	Una lista de nombres XML válidos separadas por espacios.
ENTITY	Una referencia a una entidad
ENTITIES	Una referencia a un conjunto de entidades
NOTATION	Un nombre de una notación
xml:lang	Indica el idioma del contenido.
xml:space	Indica qué hacer con espacios, tabulaciones y retornos de carro múltiples.

**Figura 4.7. Tipos de atributo admitidos por los DTD.**

- **DTD. Valores de atributos.**

Valor	Descripción
<b>valor</b>	El valor por defecto del atributo
<b>#REQUIRED</b>	Indica que el atributo es obligatorio
<b>#IMPLIED</b>	Indica que el atributo es opcional
<b>#FIXED valor</b>	El valor del atributo

**Figura 4.8. Valores admitidos por los atributos de DTD.**



- **XML-Schema.**
  - Es el lenguaje utilizado para describir la estructura, relaciones y restricciones de los documentos XML.
  - Mejora la precisión proporcionada por los DTD.
  - Se definen en ficheros XML como extensión .xsd.
  - Se referencias desde el documento XML a validar:

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

```
<element-raiz
```

```
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
```

```
xs:noNamespaceSchemaLocation="fichero-esquema.xsd">
```

- XML-Schema. Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<mensaje
  xmlns:xs="http://www.w3.org/2001/XMLSchema-
instance"
  xs:noNamespaceSchemaLocation="mensaje.xsd">
  <de>Anna</de>
  <para>Rocío</para>
  <mensaje>Ya estoy en la biblioteca.</mensaje>
</mensaje>
```

Figura 4.9. Ejemplo de documento XML a validar.

- XML-Schema. Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="mensaje">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="de" type="xs:string"/>
        <xs:element name="para" type="xs:string"/>
        <xs:element name="mensaje" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 4.10. Ejemplo de XML de validación.

- XML-Schema. Estructura.

- Declaración del documento XML.

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

- Declaración del elemento raíz y del espacio de nombres.

- ```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

- Declaración de los elementos.

- Declaración de los atributos.

- **XML-Schema. Tipos de elementos.**
  - **Locales.**
    - **Son hijos de elementos que no son raíz y solo se pueden utilizar una vez.**
  - **Globales.**
    - **Son hijos del elemento raíz y pueden ser reutilizados.**

- XML-Schema. Tipos de elementos.
  - Simples.
    - Pueden contener textos (cadenas de caracteres, fechas, valores lógicos, etc.).

```
<xs:element name=""></xs:element>
```

- Complejos.
  - Pueden contener a otros elementos, estar vacíos o contener otros elementos y textos.

```
<xs:element name="" type="">  
  <xs:complexType/>  
</xs:complexType>  
</xs:element>
```

- **XML-Schema. Sub-elementos.**
  - Son contenidos por elementos.
  - Tipos:
    - xs:sequence. Secuencia de elementos obligatorios.
    - xs:choise. Secuencias de elementos de los que debe aparecer uno.
    - xs:all. Secuencia de elementos opcionales.

```
<xs:element name="direccion">
  <xs:complexType>
    <xs:choice>
      <xs:element name="particular" type="xs:string"/>
      <xs:element name="profesional" type="xs:string"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Figura 4.11. Ejemplo de restricción de tipo “complexType” y “choise”.

- **XML-Schema. Atributos.**
  - Se definen a nivel de elemento.
  - Componentes:
    - Tipo (type): por defecto, anySimpleType.
    - Uso (use): required, optional o prohibited.
    - Valor por defecto (default).
    - Valor fijo si existe (fixed).

```
<xs:attribute name="" type="" />
```



- **XML-Schema. Tipos de datos.**

Tipo (utilizando xs como prefijo)	Descripción
xs:string	Cadena de caracteres.
xs:integer	Números enteros.
xs:decimal	Números decimales.
xs:boolean	Tipo de dato lógico.
xs:date	Fecha.
xs:time	Instante de tiempo en formato “hh:mm:ss”.
xs:duration	Un periodo de tiempo.

**Figura 4.12. Tipos de datos de XML-Schema.**

- **XML-Schema. Facetas.**
  - **Permiten detallar las restricciones de los atributos.**

Faceta	Descripción
<b>xs:length</b>	Determina una longitud fija.
<b>xs:minLength</b>	Determina una longitud mínima.
<b>xs:maxLength</b>	Determina una longitud máxima.
<b>xs:totalDigits</b>	Determina el máximo número de dígitos que puede tener un número.
<b>xs:fractionDigits</b>	Determina el máximo número de decimales que puede tener un número.
<b>xs:minExclusive</b>	Determina que el valor debe ser mayor que el valor indicado.
<b>xs:maxExclusive</b>	Determina que el valor debe ser menor que el valor indicado.

**Figura 4.13. Facetas (1 de 2).**

- **XML-Schema. Facetas.**
  - **Permiten detallar las restricciones de los atributos.**

Faceta	Descripción
<b>xs:minInclusive</b>	Determina que el valor debe ser mayor o igual que el valor indicado.
<b>xs:maxInclusive</b>	Determina que el valor debe ser menor o igual que el valor indicado.
<b>xs:enumeration</b>	Determina una lista de valores posibles.
<b>xs:whiteSpace</b>	Determina como tratar los espacios en blanco, las tabulaciones y los saltos de línea.
<b>xs:pattern</b>	Determina un patrón de caracteres permitidos.

**Figura 4.14. Facetas (2 de 2).**

- XML-Schema. Comentarios.
  - Se indican con el elemento `<annotation>`

```
<xs:annotation>  
  <xs:appinfo>Validador de mensajes</xs:appinfo>  
  <xs:documentation xml:lang="es">  
    Este esquema valida mensajes privados seguros.  
  </xs:documentation>  
</xs:annotation>
```

- **Otras alternativas de validación:**
  - **RELAX NG**
  - **Schematron**

### 4.4. XML aplicado

- **La versatilidad de XML es tal que se admiten usos en múltiples entornos:**
  - **XML en aplicaciones, como estructura de datos.**
  - **XML en la Web, como herramienta de diseño de páginas web.**
  - **XML Servicios Web, como formato de intercambio.**

### 4.5. Herramientas de visualización y edición de XML

- Editores de texto.
- Navegadores web.
- XML Notepad.
- Liquid Studio.
- XMLPad.
- Eclipse.
- Editores web.