



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

4.3. Validación de XML



- Garantizar la correcta estructura de los documentos.
- XML permite determinar las reglas que debe cumplir un determinado documento escrito en este lenguaje.
- XML no tiene etiquetas predeterminadas.
- Proporciona una infraestructura sobre la que crear etiquetas y dotarlas de significado.
- Hay ciertas reglas generales de diseño que hay que cumplir y que dotan de robustez a los lenguajes creados con XML

XML válido y XML bien formado



- 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 conforme a las reglas definidas en su DTD (Document Type Definition) o XML Schema, aunque estas definiciones de reglas semánticas son opcionales.

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.

ACTIVIDAD PROPUESTA

Documento bien formado

Haz los cambios necesarios en el siguiente documento XML para que esté bien formado:

```
<?xml version="1.0" encoding="UTF-8"?>
<Videojuego>
  <titulo anyo="2017">Fortnite</titulo>
  <empresa>Epic Games</empresas>
  <motor>Unreal Engine</Motor>
</videojuego>
```

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.
- Las reglas contenidas en un DTD tienen que ver con la estructura del documento y los elementos y atributos que puede contener.

DTD



- 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



- Tanto los datos como las reglas de validación se encuentran en un único documento.
- Sencillo y compacto.
- Se crea incluyendo un elemento de tipo OCTYPE inmediatamente después de la declaración del documento XML. Sintaxis:

```
<!DOCTYPE nombre-elemento-raíz[  
    Elementos-y-sus-relaciones  
>
```

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



- Facilita la reutilización.
- Permite compartir las reglas de validación.
- Se recomienda, por lo tanto, utilizar DTD externos.
- La vinculación de un DTD a un documento XML se realiza añadiendo una referencia al fichero que contiene el DTD. Sintaxis:

```
<!DOCTYPE nombre-elemento-raíz SYSTEM "URI">
```

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. 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>`

Por ejemplo, si en un DTD se incluye la siguiente declaración:

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

Se está indicando que el documento XML debe tener un elemento `<de>` que contiene una cadena de caracteres.

DTD. Elementos. Tipos de contenido.



EMPTY

Indica que el elemento referenciado debe estar vacío.

ANY

Indica que el elemento referenciado puede contener cualquier contenido.

(#PCDATA)

Indica que el elemento referenciado puede datos de tipo carácter (Parsed Character Data).

(nombreElemento)

Indica que el elemento referenciado puede contener al elemento indicado.

(nombreElemento1, nombreElemento2, ...)

Indica que el elemento referenciado puede contener una sucesión de elementos indicados como una lista separada por comas.

DTD. Elementos. Cardinalidades.

En las relaciones entre elementos se pueden especificar cardinalidades para determinar cuántos elementos pueden estar contenidos dentro del elemento referenciado

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

DTD. Elementos. Cardinalidades.




Estas opciones para la declaración de relaciones entre elementos se pueden combinar y agrupar con paréntesis para asignar las cardinalidades. Por ejemplo, se puede declarar un elemento de la siguiente manera:

```
<!ELEMENT aviso (#PCTADA | mensaje | nota | correo | llamada)*>
```

El significado para el ejemplo anterior es el siguiente:

Un aviso puede tener cero o más ocurrencias del tipo texto libre, elemento mensaje, elemento nota, elemento correo o elemento llamada.

DTD. Elementos. Cardinalidades.Ejemplo



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cv [
  <!ELEMENT cv (nombre, direccion, telefono, fax?, email+, idiomas)>
  <!ELEMENT nombre {#PCDATA}>
  <!ELEMENT direccion {#PCDATA}>
  <!ELEMENT telefono {#PCDATA}>
  <!ELEMENT fax {#PCDATA}>
  <!ELEMENT email {#PCDATA}>
  <!ELEMENT idiomas (idioma*)>
  <!ELEMENT idioma {#PCDATA}>
]>
<cv>
  <nombre>Desmond Miles</nombre>
  <direccion>Black Hills, Dakota del Sur, Estados Unidos</direccion>
  <telefono>+1-555-73-66-58</telefono>
  <email>desmond.miles@servidordeprueba.org</email>
  <email>desmond.miles@otroservidor.fr</email>
  <idiomas>
    <idioma>Francés</idioma>
    <idioma>Inglés</idioma>
    <idioma>Italiano</idioma>
  </idiomas>
</cv>
```

DTD. Elementos. Cardinalidades.Ejemplo

Definición	Descripción
<!DOCTYPE cv	Indica que cv es el elemento raíz del documento.
<!ELEMENT cv (nombre, direccion, telefono, fax?, email+, idiomas)>	Señala que el elemento cv tiene que contener los siguientes elementos: un nombre , una dirección , un teléfono , opcionalmente un fax , uno o más email y un idiomas .
<!ELEMENT nombre (#PCDATA)>	Informa de que el elemento nombre puede contener texto.
<!ELEMENT direccion (#PCDATA)>	Indica que el elemento direccion puede contener texto.
<!ELEMENT telefono (#PCDATA)>	Señala que el elemento telefono puede contener texto.
<!ELEMENT fax (#PCDATA)>	Indica que el elemento fax puede contener texto.
<!ELEMENT email (#PCDATA)>	Informa de que el elemento email puede contener texto.
<!ELEMENT idiomas (idioma*)>	Señala que el elemento idiomas opcionalmente (podría no tener ninguno) puede tener varios elementos idioma .
<!ELEMENT idioma (#PCDATA)>	Indica que el elemento idioma puede contener texto.

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>`

```
<!ATTLIST direccion calle CDATA "Domicilio desconocido">
```

DTD. Atributos. TIPOS



Tipo	Descripción
CDATA	Cadena de caracteres.
(<i>valor1</i> <i>valor2</i> ...)	Una lista de posibles valores.
ID	Un identificador único.
IDREF	Una referencia a un identificador único de otro elemento.
IDREFS	Una lista de referencias separadas por espacios a identificadores de otros elementos.
NMTOKEN	Un nombre XML válido.
NMTOKENS	Una lista de nombres XML válidos separados 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	Señala si se han de respetar los espacios, las tabulaciones y los retornos de carro múltiples (valor «preserve») o eliminarlos (valor «default»).

DTD. Valores de atributos.



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

DTD. Atributos. Ejemplo 1



La siguiente declaración de atributos indica que el elemento idioma tiene un atributo nivel que puede tomar los valores bajo, medio y alto, asignándole por defecto el valor medio.

```
<!ATTLIST idioma nivel (bajo | medio | alto) "medio">
```

Aplicado al siguiente fragmento de XML:

```
<idiomas>
  <idioma nivel="bajo">Francés</idioma>
  <idioma>Inglés</idioma>
  <idioma>Italiano</idioma>
</idiomas>
```

DTD. Atributos. Ejemplo 1

```
-<cv>
  <nombre>Desmond Miles</nombre>
  <direccion>Black Hills, Dakota del Sur, Estados Unidos</direccion>
  <telefono>+1-555-73-66-58</telefono>
  <email>desmond.miles@servidordeprueba.org</email>
  <email>desmond.miles@otroservicor.fr</email>
  <idiomas>
    <idioma nivel="bajo">Francés</idioma>
    <idioma nivel="medio">Inglés</idioma>
    <idioma nivel="medio">Italiano</idioma>
  </idiomas>
</cv>
```


DTD. Atributos. Ejemplo 2

Si se modifica la declaración del atributo en el DTD sustituyendo el valor por defecto por **#REQUIRED**, cuando se realice una validación del documento se mostrarán tantos errores como elementos idioma carezcan del atributo nivel

```
<!ATTLIST idioma nivel (bajo | medio | alto) #REQUIRED>
```

```
19 <idiomas>
20     <idioma nivel="bajo">Francés</idioma>
21     <idioma>Inglés</idioma>
22     <idioma>Italiano</idioma>
23 </idiomas>
```

Attribute "nivel" is required and must be specified for element type "idioma".

DTD. Atributos. Ejemplo 3

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE cv [
3   <!ELEMENT cv (nombre, idiomas)>
4   <!ELEMENT nombre (#PCDATA)>
5   <!ELEMENT idiomas (idioma*)>
6   <!ELEMENT idioma (#PCDATA)>
7   <!ATTLIST idioma nivel (bajo | medio | alto) "medio">
8 ]>
9 <cv>
10   <nombre>Desmond Miles</nombre>
11   <categoria>Senior</categoria>
12   <idiomas>
13     <idioma codigo="fr" nivel="bajo">Francés</idioma>
14     <idioma codigo="en">Inglés</idioma>
15     <idioma codigo="it">Italiano</idioma>
16   </idiomas>
17 </cv>
```

Uso de los tipos de atributo **ID** e **IDREFS**. Se declara un atributo identificador en el elemento lengua y se le asigna el tipo **ID**, convirtiendo al atributo en un identificador de entidad {los valores no se pueden repetir). Por otra parte, se declara el atributo codigoPais del elemento idioma como una referencia a un identificador de otra entidad (sin especificar cuál) mediante el tipo **IDREF**. Como resultado de esta declaración, para que el documento XML sea válido, los valores de los atributos codigoPais deben coincidir con los valores de atributos que sean identificadores de otra entidad.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cv [
  <!ELEMENT cv (nombre, idiomas, lenguas)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT idiomas (idioma*)>
  <!ELEMENT idioma (#PCDATA)>
  <!ELEMENT lenguas (lengua*)>
  <!ELEMENT lengua (#PCDATA)>
  <!ATTLIST lengua identificador ID #REQUIRED>
  <!ATTLIST idioma nivel (bajo | medio | alto) "medio">
  <!ATTLIST idioma codigoPais IDREF #REQUIRED>
]>
<cv>
  <nombre>Desmond Miles</nombre>
  <idiomas>
    <idioma codigoPais="fr" nivel="bajo">Francés</idioma>
    <idioma codigoPais="en">Inglés</idioma>
    <idioma codigoPais="it">Italiano</idioma>
  </idiomas>
  <lenguas>
    <lengua identificador="fr">Francés</lengua>
    <lengua identificador="en">Inglés</lengua>
    <lengua identificador="it">Italiano</lengua>
  </lenguas>
</cv>
```

ACTIVIDAD RESUELTA

Validación de receta de cocina con DTD

Crear el **DTD** que valide un documento XML que almacene recetas de cocina. La estructura es la siguiente:

- El elemento principal es la receta.
- Una receta está formada por el nombre (un texto libre), un listado con uno o más ingredientes y un texto con las instrucciones de elaboración (un texto libre).
- Un ingrediente contiene texto libre.
- Las recetas tienen un atributo que indica el número de raciones.
- Las recetas tienen un atributo que indica la dificultad, aceptando los valores baja, media (por defecto) y alta.
- Los ingredientes tienen un atributo opcional con la cantidad del ingrediente.
- Los ingredientes tienen un atributo opcional con la unidad en la que están expresadas las cantidades.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE receta [
  <!ELEMENT receta (nombre,ingredientes,elaboracion)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT ingredientes (ingrediente+)>
  <!ELEMENT elaboracion (#PCDATA)>
  <!ELEMENT ingrediente (#PCDATA)>

  <!ATTLIST receta raciones CDATA #REQUIRED>
  <!ATTLIST receta dificultad (baja | media | alta) "media">
  <!ATTLIST ingrediente cantidad CDATA #IMPLIED>
  <!ATTLIST ingrediente unidad CDATA #IMPLIED>
]>
<receta raciones="6" dificultad="media">
  <nombre>Tortilla de patatas</nombre>
  <ingredientes>
    <ingrediente cantidad="1" unidad="kilogramo">Patata</ingrediente>
    <ingrediente cantidad="8" unidad="unidades">Huevo</ingrediente>
    <ingrediente cantidad="1" unidad="unidades">Cebolla</ingrediente>
    <ingrediente>Aceite de Oliva</ingrediente>
    <ingrediente>Sal</ingrediente>
  </ingredientes>
  <elaboracion>
    Se pelan las patatas...
  </elaboracion>
</receta>
```

ACTIVIDAD PROPUESTA

Validación de una especie animal con DTD

Crea el DTD que valide un XML que representa una especie animal y que tiene la siguiente estructura:

- El elemento principal contiene un atributo estado que puede tomar los valores extinto, amenazado y sin-peligro.
- El elemento principal contiene un nombre (texto), una familia (texto) y un listado de áreas de distribución.
- El nombre contiene un atributo con el nombre científico.
- El listado de áreas de distribución está formado por al menos, una zona (texto).

XML de ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<especie estado="amenazado">
  <nombre nombre-cientifico="Monachus monachus">Foca monje</nombre>
  <familia>Phocidae</familia>
  <distribucion>
    <zona>Costa Atlántica</zona>
    <zona>Costa Mediterránea</zona>
  </distribucion>
</especie>
```