

# Unidad 6. Validación de Documentos XML

## 6.1 DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ARTICLES [
<ELEMENT ARTICLES (ARTICLE)*>
<ELEMENT ARTICLE (ARTICLEDATA)*>
<ELEMENT ARTICLEDATA (TITLE, AUTHOR)>
<ELEMENT TITLE (#PCDATA)>
<ELEMENT AUTHOR (#PCDATA)>
]>
<ARTICLES>
  <ARTICLE>
    <ARTICLEDATA>
      <TITLE>XML Demystified</TITLE>
      <AUTHOR>Jaidev</AUTHOR>
    </ARTICLEDATA>
  </ARTICLE>
</ARTICLES>
```



JJ Taboada León  
IES San Sebastián, Departamento de Informática  
LENGUAJE DE MARCASY SGI  
Curso 2014 / 2015

# Guión del tema

- ¿Qué es un DTD?
- Declaración de DTD
- Declaración de Elementos
- Declaración de Atributos

# ¿Qué es un DTD?

- (DTD) significa Document Type Definition (Definición del tipo de Documento )
- ¿Qué define?
  - Los elementos y atributos que pueden aparecer en el documento XML.
- Un DTD puede ser declarado en línea dentro de un documento XML, o como una referencia externa.

# Declaración de DTD interna

- Debe seguir la siguiente sintaxis

`<!DOCTYPE elemento_raíz [declaración de elementos]>`

- Ejemplo de documento XML con una DTD interna:

```
<?xml version="1.0"?>
<!DOCTYPE nota [
<!ELEMENT nota (para,de,cabecera,cuerpo)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT cabecera (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
]>
<nota>
<para>Jose</para>
<de>Juani</de>
<cabecera>Recordatorio</cabecera>
<cuerpo>NO me olvides este finde</cuerpo>
</nota>
```

# DTD externa

- Debe seguir la siguiente sintaxis

`<!DOCTYPE elemento raíz SYSTEM "archivo">`

nota.xml



```
<?xml version="1.0"?>
<!DOCTYPE nota SYSTEM "nota.dtd">
<nota>
<para>Jose</para>
<de>Juani</de>
<cabecera>Recordatorio</cabecera>
<cuerpo>NO me olvides este finde</cuerpo>
</nota>
```

nota.dtd



```
<!ELEMENT nota
(para,de,cabecera,cuerpo)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT cabecera (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
```

# ¿Por qué utilizar una DTD?

- Con una DTD, cada uno de los archivos XML se puede llevar a una descripción de su propio formato.
- Con una DTD, grupos independientes de personas se ponen de acuerdo para utilizar una DTD estándar para intercambiar datos.
- Su aplicación puede utilizar una norma DTD para verificar que los datos que recibimos del mundo exterior es válida.
- También puede utilizar un DTD para verificar sus propios datos.

# Los Bloques de Construcción de documentos XML

- Visto desde un punto de vista DTD, todos los documentos XML (y los documentos HTML) están compuestos por los siguientes elementos:

- Elementos
- Atributos
- Entidades
- PCDATA
- CDATA

# Elementos

- Los elementos son los **bloques de construcción principales** de documentos HTML y documentos XML.
  - Ejemplos de elementos HTML son "body" y "tabla".
  - Ejemplos de elementos XML podría ser "nota" y "mensaje". Los elementos pueden contener texto, otros elementos, o estar vacío. Ejemplos de vacío elementos HTML son "h", "br" y "img".

Ejemplos:

```
<body>texto</body>
```

```
<mensaje>texto</mensaje>
```



# Atributos

- Los atributos proporcionan **información adicional acerca de los elementos** .
  - Los atributos se colocan siempre dentro de la etiqueta de apertura de un elemento. Atributos siempre vienen en pares de nombre / valor.
  - El siguiente elemento "img" tiene información adicional acerca de un archivo de origen:

```

```

# Entidades

- Algunos caracteres tienen un significado especial en XML, como el signo menor que (<) que define el inicio de una etiqueta XML.
- HTML entity: "&nbsp;“. Esta entidad "no-breaking-space" representa la inserción de un espacio en blanco. Las entidades son transformadas por su valor cuando el documento XML es parseado por el parser de XML.

Entidades principales

Entidad	Caracter
&lt;	<
&gt;	>
&amp;	&
&quot;	“
&apos;	'

# Declaración de elemento

- En una DTD, los elementos XML se declaran con una declaración del elemento con la siguiente sintaxis:

<!ELEMENT nombre-elemento categoría>

o

<!ELEMENT nombre-elemento (elemento)>

## Declaración de distintos tipos de elementos y ocurrencia o aparición

- Elementos vacío → EMPTY
- Elemento PCDATA → #PCDATA
- Elemento con cualquier contenido → ANY
- Elemento con hijos → (hijo1, hijo2...)
- Aparición de un solo elemento → (hijo)
- Ocurrencia de un elemento como mínimo → (hijo+)
- 0 o más ocurrencia de un elemento → (hijo\*)
- 0 o una ocurrencia de un elemento → (hijo?)
- Uno u otro contenido → (hijo1 | hijo2)
- Contenido mixto → elemento (#PCDATA | hijo1 | hijo2 | hijo3 | hijo4)\*

# Elementos vacío →EMPTY

- Los elementos vacíos se declaran con la palabra clave EMPTY

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

Ejemplo DTD:

```
<!ELEMENT br EMPTY>
```

Ejemplo XML :

```
<br />
```

[←volver](#)

## Elementos de datos → PCDATA

- Los elementos con los datos de caracteres se analizan con la declaración # PCDATA entre paréntesis:

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

Ejemplo DTD:

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

Ejemplo XML :

```
<de>Juani</de>
```

[← volver](#)

# Elemento con cualquier contenido → ANY

- Los elementos que pueden tener cualquier contenido son declarados con ANY. Puede contener cualquier combinación de los datos apta para su procesamiento:

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

Ejemplo DTD:

```
<!ELEMENT nota ANY>
```

.....

Ejemplo XML :

```
<nota>
```

```
<para>Jose</para>
```

```
<de>Juani</de>
```

```
<asunto>Recordatorio</asunto>
```

```
<mensa>NO me olvides este finde</mensa>
```

```
</nota>
```

[←volver](#)

## Elemento con hijos → (hijo1,hijo2...)

- Los elementos con uno o más hijos se declaran con el nombre de los elementos de los hijos entre paréntesis:

```
<!ELEMENT nombre-elemento (child1)>  
<!ELEMENT nombre-elemento (child1,child2,...)>
```

Ejemplo DTD:

```
<!ELEMENT nota (para,de,asunto,mena)>  
<!ELEMENT para (#PCDATA)>  
<!ELEMENT de (#PCDATA)>  
<!ELEMENT asunto (#PCDATA)>  
<!ELEMENT mena (#PCDATA)>
```

Ejemplo XML :

```
<nota>  
  <para>Jose</para>  
  <de>Juani</de>  
  <asunto>Recordatorio</asunto>  
  <mena>NO me olvides este finde</mena>  
</nota>
```

[← volver](#)

# Aparición de un solo elemento → (hijo)

- En esta declaración el elemento secundario o hijo "mensaje" debe aparecer una vez, y sólo una vez dentro del elemento "nota".

**<!ELEMENT nombre-elemento (hijo)>**

Ejemplo DTD:

```
<!ELEMENT nota (mensaje)>
```

.....

Ejemplo XML :

```
<nota>
```

```
<mensaje>NO me olvides este finde</mensaje>
```

```
</nota>
```



# Ocurrencia de un elemento como mínimo → (hijo+)

- En esta declaración utilizamos el signo + para indicar que el elemento secundario mensaje debe aparecer una o más veces dentro del elemento "nota".

```
<!ELEMENT nota (mensaje+)>
```

Ejemplo DTD:

```
<!ELEMENT nota (mensaje+)>
```

.....

Ejemplo XML :

```
<nota>
```

```
  <mensaje>NO me olvides este finde</mensaje>
```

```
  <mensaje>quedamos a las 21h</mensaje>
```

```
</nota>
```

## 0 o más ocurrencia de un elemento → (hijo\*)

Usamos el signo \* para indicar que el elemento secundario "mensaje" puede aparecer cero o más veces dentro del elemento "nota".

```
<!ELEMENT nombre-elemento (hijo*)>
```

Ejemplo DTD:

```
<!ELEMENT nota (mensaje*)>
```

.....

Ejemplo XML :

```
<nota>
```

```
</nota>
```

# 0 o una ocurrencia de un elemento → (hijo?)

- Con el signo ?, el elemento secundario "mensaje" puede aparecer cero o una vez dentro del elemento "nota".

**<!ELEMENT nombre-elemento (hijo?)>**

Ejemplo DTD:

```
<!ELEMENT nota (mensaje?)>  
.....
```

Ejemplo XML :

```
<nota>  
    <mensaje>NO me olvides este finde</mensaje>  
    <mensaje>quedamos a las 21h</mensaje>  
</nota>
```

Esto es erróneo

[← volver](#)

## Uno u otro contenido → (hijo1 | hijo2)

- El ejemplo declara que el elemento "nota" debe contener los elementos "para", "de", "asunto", y el elemento "mensaje" o "cuerpo".

```
<!ELEMENT nombre-elemento (hijo1, hijo2, (hijo3 | hijo4)) >
```

Ejemplo DTD:

```
<!ELEMENT nota (para,de,asunto,(mensaje | cuerpo))>
```

.....

Ejemplo XML :

```
<nota>  
<para>Jose</para>  
<de>Juani</de>  
<asunto>Recordatorio</asunto>  
<cuerpo>NO me olvides este finde</cuerpo>  
</nota>
```

# Contenido mixto → elemento

(#PCDATA | hijo1 | hijo2 | hijo3 | hijo4)\*

- El ejemplo define que el elemento "nota" puede contener cero o más ocurrencias de datos de: caracteres analizados, o "para", o "de", o "asunto", o "mensaje".

```
<!ELEMENT nombre-elemento (#PCDATA | hijo1 | hijo2 | hijo3 | hijo4)* >
```

Ejemplo DTD:

```
<!ELEMENT nota (#PCDATA | para | de | asunto | mensaje | cuerpo)*>
```

.....

Ejemplos XML :

```
<nota>
```

```
<para>Jose</para>
```

```
<para>Antonio</para>
```

```
<para>Maria</para>
```

```
</nota>
```

```
<nota>para antonio</nota>
```

# Ejercicios

- Realizar los ejercicios 1 y 2 del documento que se adjunta

# Declaración de atributos

- En una DTD, los atributos se declaran con una declaración ATTLIST.
- `<!ATTLIST elemento atributo tipo-atributo valor>`

Ejemplo DTD

```
<!ATTLIST pago tipo CDATA "cheque">
```

Four red arrows point from the general syntax above to the specific example below. The first arrow points from 'elemento' to 'pago', the second from 'atributo' to 'tipo', the third from 'tipo-atributo' to 'CDATA', and the fourth from 'valor' to '"cheque"'.

<!-- This diagram shows four red arrows pointing from the general syntax above to the specific example below. The first arrow points from 'elemento' to 'pago', the second from 'atributo' to 'tipo', the third from 'tipo-atributo' to 'CDATA', and the fourth from 'valor' to '"cheque"'. -->

Ejemplo XML

```
<pago tipo="cheque" />
```

# El tipo de atributo puede ser:

Tipo	Descripción
CDATA	El valor es un dato de carácter
( <i>en1</i>   <i>en2</i>   ..)	El valor debe ser uno de una lista enumerada
ID	El valor es un identificador único
IDREF	El valor es el identificador de otro elemento
IDREFS	El valor es una lista de identificadores de otros
NMTOKEN	El valor es un nombre XML válido
NMTOKENS	El valor es una lista de nombres XML válidos
ENTITY	El valor es una entidad
ENTITIES	El valor es una lista de entidades
NOTATION	El valor es un nombre de una notación
xml:	El valor es un valor predefinido xml



# Valor por defecto

Valor	Explicación
<i>valor</i>	El valor predeterminado del atributo
# REQUIRED	El atributo es necesario
# IMPLIED	El atributo no es necesario
# FIXED <i>valor</i>	El valor del atributo es fijo

# Ejemplos

DTD:

```
<!ELEMENT cuadrado EMPTY>  
<!ATTLIST cuadrado ancho CDATA "0">
```

XML:

```
<cuadrado ancho="100" />
```

DTD:

```
<!ATTLIST persona numero CDATA #REQUIRED>
```

XML válido:

```
<persona numero="5677" />
```

XML inválido:

```
<persona />
```

DTD:

```
<!ATTLIST contacto fax CDATA #IMPLIED>
```

XML válido:

```
<contacto fax="555-667788" />
```

XML válido

```
<contacto />
```

# ¿Usar elementos o atributos?

- Usar los atributos lo menos posible
- Algunos de los problemas con los atributos son:
  - Los atributos no pueden contener varios valores (elementos secundarios pueden)
  - Los atributos no son fácilmente extensibles (para futuros cambios)
  - Los atributos no pueden describir las estructuras (elementos secundarios pueden)
  - Los atributos son más difíciles de manipular por el código del programa
  - Los valores de los atributos no son fáciles de probar con una DTD
- Sólo utilizar los atributos como identificador único

```
<mensajes>
  <nota id="n100">
    <para>Jose</para>
    <de>Juani</de>
    <cabecera>Recordatorio</cabecera>
    <cuerpo>NO me olvides este finde</cuerpo>
  </nota>
</mensajes>
```

# Entidades

- Las entidades son las variables utilizadas para definir los accesos directos a texto estándar o caracteres especiales.
- Pueden ser internas o externas

Ejemplo DTD:

```
<!ENTITY escritor "Donald Duck.">  
<!ENTITY copyright "Copyright W3Schools.">
```

Ejemplo XML :

```
<autor>&escritor;&copyright;</autor>
```

& entidad ;



Ejemplo DTD:

```
<!ENTITY escritor SYSTEM "http://www.w3schools.com/entities.dtd">  
<!ENTITY copyright SYSTEM "http://www.w3schools.com/entities.dtd">
```

Ejemplo XML :

```
<autor>&escritor;&copyright;</autor>
```

# Ejemplos de XML y DTD (I)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE DOCUMENT SYSTEM "cliente.dtd">
<libreta-direcciones>
  <entrada>
    <nombre>John Doe</nombre>
    <direccion>
      <calle>34 Fountain Square Plaza</calle>
      <region>OH</region>
      <codigo-postal>45202</codigo-postal>
      <localidad>Cincinnati</localidad> <pais>US</pais>
    </direccion>
    <telefono>513-555-8889</telefono>
    <correo-e href="mailto:jdoe@emailaholic.com"/>
  </entrada>
</libreta-direcciones>
```

Cliente.xml

Cliente.dtd

```
<!ELEMENT libreta-direcciones (entrada*)>
<!ELEMENT entrada (nombre, direccion, telefono, correo-e)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT direccion (calle, region, codigo-postal, localidad, pais)>
<!ELEMENT calle (#PCDATA)>
<!ELEMENT region (#PCDATA)>
<!ELEMENT codigo-postal (#PCDATA)>
<!ELEMENT localidad (#PCDATA)>
<!ELEMENT pais (#PCDATA)>
<!ELEMENT telefono (#PCDATA)>
<!ELEMENT correo-e EMPTY>
<!ATTLIST correo-e href CDATA #IMPLIED>
```

# Ejemplos de XML y DTD (II)

```
<?xml version="1.0"?>
<!DOCTYPE Productos SYSTEM "Productos.dtd">
<Productos>
  <Items>
    <Cantidad>2</Cantidad>
    <Descripcion>Hamburguesas</Descripcion>
    <Precio>€2.25</Precio>
    <Total>€$4.5</Total>
  </Items>
  <Items>
    <Volumen>2 Litros</Volumen>
    <Descripcion>Coca Cola</Descripcion>
    <Precio>€1.25</Precio>
    <Total>€1.5</Total>
  </Items>
  <Impuesto>€0.25</Impuesto>
  <Entrega>A domicilio</Entrega>
  <Total>€6.25</Total>
</Productos>
```

Productos.xml

Productos.dtd

```
<!ELEMENT Productos (Items*, Impuesto?, Entrega,
Total)>
  <!ELEMENT Items ((Cantidad|Volumen), Descripcion,
Precio, Total)>
    <!ELEMENT Cantidad (#PCDATA)>
    <!ELEMENT Volumen (#PCDATA)>
    <!ELEMENT Descripcion (#PCDATA)>
    <!ELEMENT Precio (#PCDATA)>
    <!ELEMENT Total (#PCDATA)>
  <!ELEMENT Impuesto (#PCDATA)>
  <!ELEMENT Entrega (#PCDATA)>
  <!ELEMENT Total (#PCDATA)>
```

# Ejemplos de XML y DTD (III)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Libro SYSTEM "Libro.dtd">
<Libro isbn="isbn-861003-11-0"
  fecha = "Noviembre 1999"
  paginas = "750">
  <Titulo>Aprendiendo XML</Titulo>
  <Abstract>Metodo de aprender XML</Abstract>
  <Categorias>
    <Categoria>Internet</Categoria>
    <Categoria>Publicacion Web</Categoria>
    <Categoria>XML</Categoria>
  </Categorias>
  <Precio moneda="eur">34.5</Precio>
</Libro>
```

Libro.xml

Libro.dtd

```
<!ELEMENT Libro (Titulo, Abstract,
Categorias, Precio?)>
<!ELEMENT Categorias (Categoria,
Categoria, Categoria)>
<!ELEMENT Titulo (#PCDATA)>
<!ELEMENT Abstract (#PCDATA)>
<!ELEMENT Categoria (#PCDATA)>
<!ELEMENT Precio (#PCDATA)>
<!ATTLIST Precio moneda (dol|eur|lib)
#REQUIRED>
<!ATTLIST Libro isbn ID #REQUIRED
nivel CDATA #IMPLIED
fecha CDATA #REQUIRED
paginas CDATA #REQUIRED
autor IDREFS #IMPLIED
editor IDREFS #IMPLIED>
```

# Ejemplos de XML y DTD (IV)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE cuentas SYSTEM "Banco.dtd">
<cuentas>
  <copropietario>John Doe</copropietario>
  <copropietario>Jack Smith</copropietario>
  <cheques>
    <saldo>170.00</saldo>
    <transaccion>-100.00</transaccion>
    <transaccion>-500.00</transaccion>
    <cargos>4.00</cargos>
  </cheques>
  <copropietario>John Doe</copropietario>
  <ahorros>
    <saldo>5000.00</saldo>
    <intereses>212.50</intereses>
  </ahorros>
</cuentas>
```

Banco.xml

Banco.dtd

```
<!ENTITY % cuenta "(saldo,
transaccion*)">
<!ELEMENT cuentas (copropietario+,
(chèques|ahorros))+>
<!ELEMENT copropietario (#PCDATA)>
<!ELEMENT cheques (%cuenta;,
cargos)>
<!ELEMENT ahorros
(%cuenta;,intereses)>
<!ELEMENT cargos (#PCDATA)>
<!ELEMENT intereses (#PCDATA)>
<!ELEMENT saldo (#PCDATA)>
<!ELEMENT transaccion (#PCDATA)>
```



# Ejercicios

- Realizar los ejercicios 3 y 4 del documento que se adjunta

# Unidad 6

## Validación de documentos XML

### 6.1 DTD

Fin de la presentación

¡Gracias!