

# UD2: Definición de esquemas y vocabularios en XML

## 2.1.1 DTDs

LMSXI

IES de Teis

Vigo

Curso 2021-2022

# Documentos XML válidos

- Hasta ahora, hemos creado documentos XML bien formados, es decir, sintácticamente correctos de acuerdo al estándar de XML definido por el W3C
- Un **documento XML válido está bien formado** y, además, se ajusta **a una estructura definida**. Esa estructura se puede definir en una de estas 2 opciones:
  - DTD
  - Esquema XML

# DTD

- Document Type Definition
- Permite crear un vocabulario o dialecto XML: Contiene especificaciones sobre:
  - qué elementos y atributos debe tener un documento XML
  - número de veces que pueden aparecer elementos (a grandes rasgos)
  - obligatoriedad/opcionalidad de atributos
- No es un documento XML, pero tiene una sintaxis propia para definir:
  - Elementos
  - Atributos
  - Entidades

# Definición de DTD

- **Interna:** dentro del documento XML que define

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

```
<!DOCTYPE nombre_Elemento_Raiz [  
  declaraciones  
>
```

→ [2.1.1-agenda-dtd-in.xml](#)

- **Externa:** en un documento diferente:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<!DOCTYPE nombre_Elemento_Raiz SYSTEM "nombreArchivo.dtd">
```

- En el archivo externo iría directamente la declaración de elementos y atributos de la DTD

→ [2.1.2-agenda-dtd-ext.xml](#)

# Definición de DTD

- Además de interna y externa, existe también una referencia pública para DTDs cuyo uso fue aceptado de forma general.

`<!DOCTYPE raíz PUBLIC "ID_público" "URL">`

- En la URL puede encontrarse la definición de la DTD
  - Es el caso, por ejemplo, de versiones anteriores a HTML5:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
    "http://www.w3.org/TR/html4/strict.dtd">
```

# DTD: Declaración de elementos

`<!ELEMENT nombre_elemento (contenido)>`

- Un DTD debe incluir tantas declaraciones de elementos como tipos de elementos (de etiquetas) tenga
- El contenido puede ser:
  - **Vacío:** Se utiliza la palabra clave EMPTY

`<!ELEMENT nombre_elemento EMPTY>`

- **Texto genérico (incluye números y otros símbolos):** Se utiliza la palabra clave #PCDATA
- **Otro(s) subelementos**
- **Contenido mixto:** (texto y elementos anidados)
- Cualquier otro tipo de contenido: palabra clave **ANY** (no recomendado, solo para pruebas)

# DTD: Declaración de elementos vacíos

```
<!DOCTYPE ejemplo_vacio [  
    <!ELEMENT ejemplo_vacio (vacio)>  
    <!ELEMENT vacio EMPTY>  
]>  
<ejemplo_vacio>  
    <vacio/>  
</ejemplo_vacio>
```

→ [2.1.3-ejemplo\\_vacio.xml](#)

# DTD: Declaración de elementos con texto

```
<!DOCTYPE ejemplo_texto [  
    <!ELEMENT ejemplo_texto (texto)>  
    <!ELEMENT texto (#PCDATA)>  
>
```

```
<ejemplo_texto>  
    <texto>Aquí texto o números 1 2 3 u  
    otros caracteres : , & ~ . </texto>  
</ejemplo_texto>
```

→ [2.1.4-ejemplo\\_texto.xml](#)



# DTD: Declaración de elementos con subelementos

- Los subelementos deben ir separados por comas y el orden indica el orden en que deben aparecer
- Si los subelementos pueden alternarse se utiliza | para separar las posibles opciones

```
<!ELEMENT contacto (nombre, apellido1,  
apellido2, (telefonos|email))>
```

Un contacto tendrá como subelementos:

- nombre, apellido1, apellido2 y
- o un elemento *telefonos* o un elemento *email*

→ [2.1.5-agenda-email-dtd-in.xml](#)

# DTD: Declaración de elementos con subelementos

- A su vez, habrá que declarar los subelementos como elementos y su contenido

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

- Si un elemento (o grupo de elementos) se puede repetir se utiliza:

- ? => 0 ó 1 vez

- + => 1 o más veces

- \* => 0 o más veces

```
<!ELEMENT telefonos (telefono)+>
```

# DTD: Declaración de elementos con subelementos de contenido mixto

- Contenido mixto: texto y/o subelementos dentro de un elemento
  - Los datos textuales deben aparecer en primer lugar (#PCDATA)
  - El grupo de subelementos debe ir separado con | para marcar la elección
  - El grupo debe aparecer con \* (0 o más veces)

<!-- Contenido mixto-->

```
<!ELEMENT item (#PCDATA | id | nombre |  
descripcion | precio)*>
```

→ [2.1.6-pedidos-mixto-dtd-in.xml](#)

# DTD: Declaración de elementos con subelementos de contenido mixto

```
1  <!DOCTYPE pedido [  
2  <!ELEMENT pedido (item+)>  
3  <!-- Contenido mixto -->  
4  <!ELEMENT item (#PCDATA | id | nombre | descripcion | precio)*>  
5  <!ELEMENT id (#PCDATA)>  
6  <!ELEMENT nombre (#PCDATA)>  
7  <!ELEMENT descripcion (#PCDATA)>  
8  <!ELEMENT precio (#PCDATA)>  
9  ]>  
10 <!-- Supongamos que cada item de pedido tiene un origen diferente y puede venir con XML estructurado:  
11     Id, nombre, descripcion, precio, un único texto o combinaciones de esos elementos -->  
12 <pedido>  
13     <item>  
14         <id>123</id>  
15         <nombre>Ratón Logitech</nombre>  
16         <descripcion>Óptico inalámbrico ergonómico</descripcion>  
17         <precio>13.50€</precio>  
18     </item>  
19     <item>  
20         457:Teclado mecánico, 34.95€  
21     </item>  
22     <item>  
23         Small metallic device for assisting in flotalating.  
24         <nombre>Metallic Flotalator</nombre>  
25         <precio>.50</precio>  
26     </item>  
27 </pedido>  
28
```

# DTD: Declaración de atributos: Tipos

```
<!ATTLIST nombre_elemento nombre_atributo  
tipo carácter >
```

- Tipos de atributos:
  - CDATA: Texto sin restricciones. Es muy general y no distingue textos, números, booleanos, etc.
  - NMTOKEN: Texto con restricciones (no pueden contener espacios o comas), pero sí comenzar por dígitos, contener :, -, .
  - NMTOKENS: Varios NMTOKEN separados por espacios en blanco

→ [2.1.7-articulos-dtd-in.xml](#)

# DTD: Declaración de atributos: Tipos

```
<!ATTLIST nombre_elemento nombre_atributo  
tipo carácter >
```

- Tipos de atributos:
  - **Enumeración:** Un atributo puede tomar un n.º limitado de valores entre paréntesis separados por |

```
<!ATTLIST telefono tipo (movil|fijo)  
#REQUIRED>
```

→ 2.1.1-agenda-dtd-in.xml

# DTD: Declaración de atributos: Tipos

```
<!ATTLIST nombre_elemento nombre_atributo  
tipo carácter >
```

- Tipos de atributos
  - **ID**: Identificador del elemento. Debe ser único en todo el documento XML y debe comenzar por una letra o guion bajo
  - **IDREF**: Representa el valor de otro atributo ID de otro elemento. Para que sea válido el documento, el ID del otro elemento debe existir

```
<!ATTLIST empleado  
    id ID #REQUIRED  
    jefe IDREF #IMPLIED  
>
```

# DTD: Declaración de atributos: Tipos

```
<!ATTLIST nombre_elemento nombre_atributo  
tipo carácter >
```

- Tipos de atributos
  - **IDREFS**: Representa múltiples ID de otros elementos, separados por espacios. Para que sea válido deben existir los Ids en el documento XML

```
<!ATTLIST departamento jefes IDREFS #REQUIRED>
```

→ 2.1.9-banco-dtd-in.xml



## DTD: Declaración de atributos: Carácter

```
<!ATTLIST nombre_elemento nombre_atributo  
tipo carácter >
```

- Carácter puede ser:
  - Un valor textual entre comillas dobles o simples:  
Representa el valor por defecto de **un atributo opcional**

```
<!ATTLIST nombre sponsor CDATA 'Nike'>
```

- #REQUIRED: Es obligatorio que exista el atributo en el documento XML. No se le asigna un valor por defecto

```
<!ATTLIST nombre id ID #REQUIRED >
```

## DTD: Declaración de atributos: Carácter

```
<!ATTLIST nombre_elemento nombre_atributo  
tipo carácter >
```

- Carácter puede ser:

- #FIXED: Si aparece el atributo se le asigna un valor fijo

```
<!ATTLIST nombre nacionalidad CDATA  
#FIXED "es" >
```

- #IMPLIED: De carácter opcional y sin valor por defecto. Si hubiese valor por defecto, no se escribe #IMPLIED

```
<!ATTLIST nombre internacional CDATA  
#IMPLIED >
```

# DTD: Declaración de atributos: Carácter

```
<!ATTLIST nombre_elemento nombre_atributo  
tipo carácter >
```

## → 2.1.10-deportistas-dtd-in.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE deportistas [
3  <!ELEMENT deportistas (nombre)*>
4  <!ELEMENT nombre (#PCDATA)>
5  <!ATTLIST nombre
6      |      sponsor CDATA 'Nike'
7      |      id ID #REQUIRED
8      |      nacionalidad CDATA #FIXED 'es'
9      |      internacional CDATA #IMPLIED
10     |      category (futbol|tenis|volleyball) #REQUIRED
11     |      >
12  <!-- sponsor: atributo de nombre de tipo carácter y valor por defecto Nike -->
13  <!-- id: atributo obligatorio de nombre de tipo ID -->
14  <!-- nacionalidad: atributo de nombre de tipo carácter y valor fijo "es" (si está presente) -->
15  <!-- internacional: atributo opcional de nombre de tipo carácter -->
16  <!-- category: atributo obligatorio de nombre enumerado con 3 posibles valores -->
17  ]>
18  <deportistas>
19      <nombre id="A0001" internacional="yes" category="futbol" nacionalidad="es">Cristiano Ronaldo</nombre>
20      <nombre id="A0002" category="futbol" sponsor="Adidas" nacionalidad="es">Hugo Mallo</nombre>
21      <nombre id="A0003" category="tenis" internacional="yes">Rafa Nadal</nombre>
22      <nombre id="A0004" category="volleyball" internacional="yes" sponsor="Puma"> Mario Ferrera </nombre>
23  </deportistas>
```



## DTD: Limitaciones

- No son documentos XML y, no se puede comprobar si están bien formados
- No se pueden fijar restricciones sobre los tipos de datos (números, cadenas de caracteres, valores booleanos, longitudes, número de decimales, etc.)
- No soportan espacios de nombres
- No se pueden enumerar los posibles valores de los elementos
- Los elementos no pueden tener valores por defecto
- El control sobre el número de elementos que pueden aparecer es limitado

# Herramientas de validación

- Notepad ++ con plugin XML Tools:
  - Plugins > XML Tools > Validate now o CTRL+Alt+Shift+M
  - Si aparece el mensaje “DTD está prohibido”
    - Versión 3.1.1.6 => Plugins > XML Tools > Options, hay que cambiar el valor Prohibit DTD y a false
  - Esto puede variar en función de la versión del plugin

# Herramientas de validación

- Visual Studio Code:
  - Botón derecho sobre el fichero > Paleta de Comandos > XML: Revalidate current XML file
    - También podemos establecer un atajo de teclado en la rueda de engranaje, por ejemplo, el mismo que en XML Tools: CTRL+Alt+Shift+M



## Referencias

- *Validación de documentos XML* de Victor M. Lourido Estévez. Margarita Pin Rodríguez
- *Validación con DTD del Material* da profesora Marta Rey López 2020-2021