

UD2: Definición de esquemas y vocabularios en XML

2.1.2.DTD-ENTIDADES

LMSGI

IES de Teis

Vigo

Curso 2021-2022

DTD: Entidades

- Además de atributos y elementos, es posible declarar ENTIDADES
- Las entidades son pares clave-valor que permiten:
 - Declarar constantes reutilizables en un documento XML
 - Hacer referencia a objetos externos (imágenes, ficheros con texto o XML, páginas web, etc.)

DTD: Tipos de entidades (1/2)

Es posible clasificar las entidades en XML desde distintos puntos de vista y la clasificación no es exclusiva:

- Según su origen:
 - **Predefinidas:** `&`; `<`; `>`; `'`; `"`; y las numeradas en decimal y hexadecimal que vimos en la UD1 como `&` o `&`;
 - **Definidas por el usuario:** en función de su tipología, cambia la sintaxis
- Según la ubicación de la referencia:
 - **Internas:** Hacen referencia a una declaración en el propio documento XML.
 - **Externas:** Referencia un fichero externo por su URL. Puede contener XML u otro tipo de contenido.

DTD: Tipos de entidades (2/2)

Según el tipo de contenido de la referencia:

- **Analizables:** Referencian a documentos de texto o XML
- **No analizables:** Otro tipo de documentos: imágenes, audio, vídeo, etc.

Entidades internas

- También conocidas como *macros* o *constantes de texto*
- Definición:

`<!ENTITY nombre_entidad "definicion_Entidad" >`

- Son válidas comillas dobles o simples **para todas las definiciones**
 - El nombre de la entidad sigue las mismas reglas que el nombre de los elementos en XML
- Referencia: `&nombre_entidad;`
- La referencia **se reemplazará por la cadena de texto usada en su definición.**

→ 2.1.2.1-entidades-internas-firma.xml

```
1  <!DOCTYPE email [  
2      <!ENTITY contenido_firma "Juan Antonio Sánchez López">  
3      <!ELEMENT email (firma)>  
4      <!ELEMENT firma (#PCDATA)>  
5  ]>  
6  <email>  
7      <firma>&contenido_firma;</firma>  
8  </email>
```

Entidades externas: Públicas

- Públicas:

`<!ENTITY nombre_entidad PUBLIC "public_Id" "URI">`

Los procesadores de XML pueden usar el id público para almacenar una copia local del recurso accesible vía la URI.

→ **2.1.2.2-entidades-externas-publica-sorteos.xml**

```
1 <?xml version="1.0" standalone="no"?>
2 <!DOCTYPE sorteos [
3     <!ELEMENT sorteos (items*)>
4     <!ENTITY doc_externo PUBLIC "-//W3C//TEXT copyright//EN" "https://www.juegosonce.es/rss/sorteos2.xml">
5
6     <!-- Declaración de los elementos que se usan en el recurso externo -->
7
8 ]>
9 <sorteos>&doc_externo;</sorteos>
```

- Podemos comprobar que efectivamente resuelve las entidades si cambiamos alguna declaración de la DTD.

Entidades externas: Privadas (1/2)

- Privadas o locales a la propia red:

`<!ENTITY nombre_entidad SYSTEM "URI">`

- `2.1.2.3-entidades-externas-locales.xml` y `2.1.2.3.1-firma-externa.xml`

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <!DOCTYPE email [
3      <!ELEMENT email (firma)>
4      <!ELEMENT firma (#PCDATA) >
5      <!ENTITY firma-externa SYSTEM "2.1.2.3.1-firma-externa.xml">
6  ]>
7  <!-- Puede mostrarse un error con Visual Studio Code:
8      The content of element type "email" is incomplete, it must match "(firma)"
9      , pero con Notepad++ y XML Tools sí se valida. -->
10 <email>
11     &firma-externa;
12 </email>
```

Entidades externas: Privadas (2/2)

→ 2.1.2.3-entidades-externas-locales.xml (en Firefox)

```
-<!--  
  Puede mostrarse un error con Visual Studio Code:  
  The content of element type "email" is incomplete, it must match "(firma)"  
  , pero con Notepad++ y XML Tools sí se valida.  
-->  
<email> </email>
```

- No se puede visualizar en el navegador por motivos de seguridad (ataques XML External Entity Injection o XXE), pero Notepad++ y XML Tools permiten validarlo si la opción Prohibit DTD es False.
 - Si cambiamos el contenido de 2.1.2.2.1-firma-externa.xml renombrando firma a firma1, veremos que el validador de XML Tools detecta el cambio.
 - Un procesador de XML reemplazará la referencia a la entidad por su contenido:

```
<email>  
  <firma>  
    Juan Antonio Sánchez López  
  </firma>  
</email>
```


Entidades externas no analizables (1/2)

- En las entidades externas **no analizables**, los documentos referenciados no contienen texto sino **contenido binario** (imágenes, audio, vídeo, etc.).
- Se utiliza la palabra reservada **NDATA** en la definición de la entidad
- Además, se necesita una declaración **NOTATION** para asociar algún tipo de aplicación para gestionar el contenido binario designado por NDATA
- NOTATION debe declararse antes de que una entidad pueda utilizarla.

```
<!NOTATION webp SYSTEM "C:\Program Files (x86)\Google\Chrome\
Application\chrome.exe">
```

```
<!ENTITY dibujo SYSTEM "2.1.2.4.1-pc.webp" NDATA webp>
```

Entidades externas no analizables (2/2)

- [2.1.2.4-entidades-externas-locales-no-analizables-in.xml](#) y [2.1.2.4.1-pc.webp](#)

```
1  <!DOCTYPE email [  
2      <!ELEMENT email (#PCDATA)>  
3      <!-- Atención al tipo de atributo ENTITY -->  
4      <!ATTLIST email adjunto ENTITY #IMPLIED>  
5      <!NOTATION webp SYSTEM "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe">  
6      <!ENTITY dibujo SYSTEM "2.1.2.4.1-pc.webp" NDATA webp>  
7  ]>  
8  
9  <email adjunto="dibujo">Aquí el texto</email>
```

- Se indica con **NDATA webp** que la aplicación encargada de visualizar el archivo de la entidad dibujo es chrome.exe
- En la práctica, este tipo de asociaciones **no se suele usar** porque el analizador de XML no es capaz de invocar a la aplicación con solo esa información.
- Se utiliza como un **tipo de atributos ENTITY** (adicional a los ya vistos de CDATA, NMTOKEN, ID, ...).
- También pueden declararse en una DTD externo: [2.1.2.4.2-entidades-externas-locales-no-analizables-out.xml](#) y [2.1.2.4.2.1-dibujo.dtd](#)

Entidades paramétricas o de parámetros (1/3)

- Permiten nombrar partes de un DTD para su posterior reutilización
- Declaración:

`<!ENTITY % nombre_entidad "(elemento1, elemento2, elemento3?)">`

- ¡Ojo al espacio entre % y el nombre_entidad!
- Permiten comillas dobles o simples en la definición

- Reutilización:

`%nombre_entidad;`

- **Deben declararse antes de usarse**
- No se pueden reutilizar en DTD internas, pero sí declararse

→ 2.1.2.7-entidades-parametro-in-no.xml

- Pueden hacer referencia a otras entidades, pero no referencias recursivas (a sí mismas).

Entidades paramétricas o de parámetros (2/3)

- Ejemplo para reutilizar atributos:
- ➔ [2.1.2.5-entidades-parametro-at.xml](#) y [2.1.2.5.1-envios.dtd](#)

```
1  <!ELEMENT envios (origen,destino)>
2  <!ENTITY % direccion
3      "calle CDATA #REQUIRED
4      | numero CDATA #IMPLIED
5      | poblacion CDATA #REQUIRED">
6  <!ELEMENT origen EMPTY>
7  <!ELEMENT destino EMPTY>
8  <!ATTLIST origen %direccion;>
9  <!ATTLIST destino %direccion;>
```

```
1  <!DOCTYPE envios SYSTEM "2.1.2.5.1-envios.dtd">
2  <envios>
3      <origen calle="Calle Torres" numero="5" poblacion="Vigo" />
4      <destino calle="Gran vía" numero="25" poblacion="Marín" />
5  </envios>
```

Entidades paramétricas o de parámetros (3/3)

- Ejemplo para reutilizar elementos:

→ 2.1.2.6-entidades-parametro-el.xml y 2.1.2.6.1-direccion.dtd

```
1 <!ELEMENT carta (emisor, receptor)>
2 <!ENTITY % datos_direccion "tipo_via, nombre_via, numero?, bloque?, piso, letra">
3 <!-- La entidad datos_personales hace referencia a su vez a la entidad datos_direccion -->
4 <!ENTITY % datos_personales "nombre, apellidos, %datos_direccion;">
5 <!ELEMENT emisor (%datos_personales;)>
6 <!ELEMENT receptor (%datos_personales;)>
7
8 <!ELEMENT nombre (#PCDATA)>
9 <!ELEMENT apellidos (#PCDATA)>
10
11 <!ELEMENT tipo_via (#PCDATA)>
12 <!ELEMENT nombre_via (#PCDATA)>
13 <!ELEMENT numero (#PCDATA)>
14 <!ELEMENT bloque (#PCDATA)>
15 <!ELEMENT piso (#PCDATA)>
16 <!ELEMENT letra (#PCDATA)>
```

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE carta SYSTEM "2.1.2.4.1-direccion.dtd">
3 <carta>
4   <emisor>
5     <nombre>Ana </nombre>
6     <apellidos>López Mar</apellidos>
7     <tipo_via>Avenida</tipo_via>
8     <nombre_via>de Vigo</nombre_via>
9     <numero>2</numero>
10    <piso>1</piso>
11    <letra>D</letra>
12  </emisor>
13  <receptor>
14    <nombre>Beatriz </nombre>
15    <apellidos>Luz Serrano</apellidos>
16    <tipo_via>Calle</tipo_via>
17    <nombre_via>San Juan</nombre_via>
18    <piso>Bajo</piso>
19    <letra>E</letra>
20  </receptor>
21 </carta>
```

Atributo *standalone*

- El valor “yes” indica que no hay declaraciones externas que afectan a la información que se pasará del procesador de XML a una aplicación final como pueden ser los valores por defecto de atributos o las declaraciones de entidades
- El valor “no” indica que **hay o puede haber** declaraciones externas que afecten a la información que se pasará del procesador de XML a una aplicación final
- Si no hay declaraciones externas, el atributo standalone por sí mismo no tiene ningún significado
- Si hay declaraciones externas y *standalone* no se ha especificado, se asume el valor *no*

Declaraciones mixtas de DTD: interna y externa

- Es posible combinar una DTD externa con declaraciones internas

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

```
[<!-- declaraciones internas -->]
```

```
>
```

- Las declaraciones mixtas:
 - Permiten reutilizar la DTD externa entre distintos documentos XML
 - Las DTDs internas pueden personalizar las DTDs externas:
 - Redefiniendo entidades por parámetro (tanto de elementos como de atributos), aplicándose las internas sobre las externas
 - Pueden especificar el orden de los elementos y subelementos
 - Pero no permiten redefinir elementos ni atributos con declaraciones ATTLIST o ELEMENT
 - [2.1.2.8.1-dtd-mixta.xml](#) y [2.1.2.8.2-carta.dtd](#)

Secciones condicionales en una DTD (1/4)

- Sintaxis:
 - `<![IGNORE [declaraciones]]>`: Se ignoran las declaraciones entre []
 - `<![INCLUDE [declaraciones]]>`: Se incluyen las declaraciones entre []
- Solo es posible utilizarlas en **DTD externas**
- Las declaraciones en las secciones condicionales deben ser **completas**, no afecta a fragmentos de declaraciones
- Suelen combinarse con entidades por parámetro en una DTD mixta de forma que:
 - En la DTD interna se declara la entidad paramétrica
 - En la DTD externa se reutiliza la entidad paramétrica
- De esta forma, se puede reutilizar la misma DTD externa para casuísticas similares, pero con ligeras diferencias que se establecen en la DTD interna

Secciones condicionales en una DTD (2/4)

- Ejemplos: Si fuese posible tener dos versiones de subelementos para persona, con una versión básica y otra ampliada
- **2.1.2.9.1.1-datos-basicos.xml** y **2.1.2.9.2.1-datos-ampliados.xml**

```
3 <persona>
4   <nombre>Elsa</nombre>
5   <edad>23</edad>
6 </persona>
```

```
3 <persona>
4   <nombre>Elsa</nombre>
5   <apellidos>López Ruíz</apellidos>
6   <edad>23</edad>
7   <ciudad>León</ciudad>
8 </persona>
```

- Una primera versión se podría usar sendos ficheros DTD con el mismo contenido, pero intercambiando convenientemente la sección de IGNORE e INCLUDE

Secciones condicionales en una DTD (3/4)

→ 2.1.2.9.1.2-DTD-condic-datos-basicos.dtd

```
2  <![ INCLUDE [
3  |   <!ELEMENT persona (nombre, edad)>
4  ]]>
5
6  <![ IGNORE[
7  |   <!ELEMENT persona (nombre, apellidos, edad, ciudad)>
8  ]]>
9
10 <!ELEMENT nombre (#PCDATA)>
11 <!ELEMENT apellidos (#PCDATA)>
12 <!ELEMENT edad (#PCDATA)>
13 <!ELEMENT ciudad (#PCDATA)>
```

→ 2.1.2.9.2.2-DTD-condic-datos-ampliados.dtd

- Ambos ficheros tienen el mismo contenido y solo cambia la sección que se incluye o ignora

```
2  <![ IGNORE [
3  |   <!ELEMENT persona (nombre, edad)>
4  ]]>
5
6  <![ INCLUDE[
7  |   <!ELEMENT persona (nombre, apellidos, edad, ciudad)>
8  ]]>
9
10 <!ELEMENT nombre (#PCDATA)>
11 <!ELEMENT apellidos (#PCDATA)>
12 <!ELEMENT edad (#PCDATA)>
13 <!ELEMENT ciudad (#PCDATA)>
```

Secciones condicionales en una DTD (4/4)

- Una versión más refinada podría utilizar un único documento DTD externo que incluye:
 - Secciones condicionales que sustituyen INCLUDE E IGNORE con la reutilización de entidades de parámetro (aún no declaradas)
 - Una DTD mixta con:
 - la DTD externa única: `2.1.2.9.3.3-DTD-condicionales-datos-persona.dtd`
 - DTD interna con la declaración de la entidad por parámetro que especifica qué opción (básica o ampliada) se va a utilizar en ese documento XML:
 - `2.1.2.9.3.1-datos-persona-basicos.xml`
 - `2.1.2.9.3.2-datos-persona-ampliados.xml`

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*