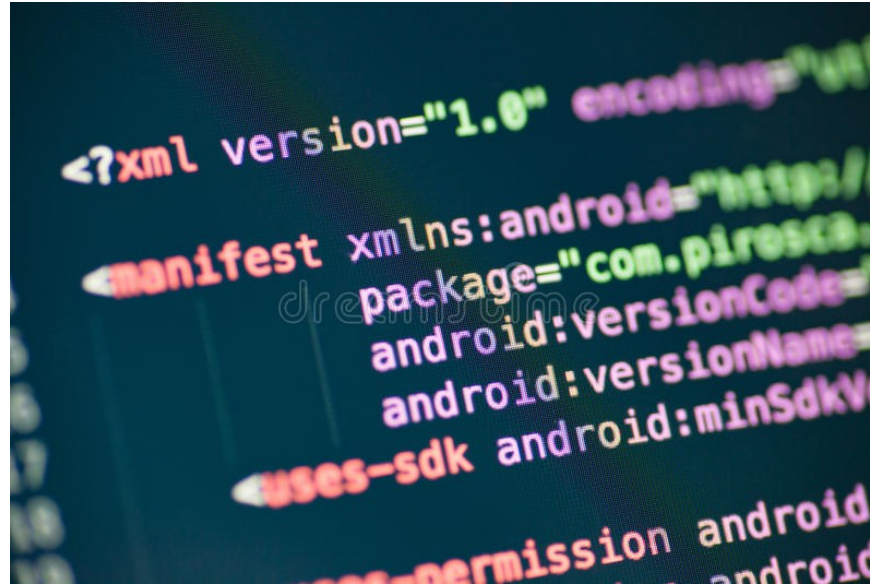


Lenguajes de marcas y sistemas de gestión de la información



UT05 – DTD y XML Schema

2 – DTD – Validación de atributos – Entidades – Limitaciones de DTD

DTD - Declaración de atributos

Para declarar los atributos de un elemento se utiliza "ATTLIST", que tiene la siguiente forma:

```
<!ATTLIST nom-elem nom-atr tipo-atr valor-atr>
```

Donde:

- nom-elem: elemento para el que estamos declarando atributos
- nom-atr: nombre del atributo declarado
- tipo-atr: tipo de dato del atributo
- valor-atr: valor del atributo / características adicionales del atributo.

DTD - Declaración de atributos - Ejemplo

Ejemplo de DTD para un fichero de alumnos:

```
<!DOCTYPE alumnos [  
    <!ELEMENT alumnos (alumno*)>  
    <!ELEMENT alumno (#PCDATA)>  
    <!ATTLIST alumno      dni ID #REQUIRED  
                        fechaNacimiento CDATA #IMPLIED>  
]>
```

Donde cada alumno:

- Sólo puede contener texto (definido en el ELEMENT del alumno)
- Debe tener un atributo dni, obligatorio. Este atributo es de tipo ID, que debe cumplir ciertas normas que veremos enseguida.
- Puede tener un atributo fechaNacimiento, opcional, que sólo puede tener texto (CDATA = Character Data)

DTD - Atributos - Tipos de datos

Tipo	El atributo contiene...
CDATA	Cadena de caracteres
(v1 v2 v3 ...)	Uno de los valores de la lista
ID	Un identificador único. Prohibido dos ID iguales en todo el documento.
IDREF	Un valor de otro atributo tipo ID, que aparezca en el mismo documento.
IDREFS	Como el anterior, pero varios ID separados por espacios
ENTITY	Un nombre de entidad
ENTITIES	Uno o varios nombres de entidad separados por espacios
NMTOKEN	Un nombre con sintaxis XML válida
NMTOKENS	Lista de nombres con sintaxis XML válida separados por espacios

DTD - Atributos - Valor / características

Valor	Significa...
"valor-defecto"	El atributo es opcional. Si el elemento no tiene el atributo, se asume que el valor del atributo es "valor-defecto".
#IMPLIED	El atributo es opcional. No hay un valor por defecto.
#REQUIRED	El atributo es obligatorio.
#FIXED "valor"	El atributo es opcional. Pero, si se incluye, debe ser un valor específico. No se puede usar ningún otro valor.

DTD - Declaración de atributos - Ejemplo

Los atributos de un elemento se pueden definir con una sola declaración ATTLIST, o con varias. El ejemplo anterior puede escribirse así:

```
<!DOCTYPE alumnos [  
    <!ELEMENT alumnos (alumno*)>  
    <!ELEMENT alumno (#PCDATA)>  
    <!ATTLIST alumno      dni ID #REQUIRED>  
    <!ATTLIST alumno      fechaNacimiento CDATA #IMPLIED>  
>
```

No hay diferencias entre una forma y la otra.

Es una cuestión de preferencias, facilidad de lectura, y peso del documento

DTD - Declaración de atributos -

Ejemplo

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

```
<!DOCTYPE deportistas [  
    <!ELEMENT deportistas (futbol | f1 | tenis)*>  
    <!ELEMENT futbol (#PCDATA)>  
    <!ELEMENT f1 (#PCDATA)>  
    <!ELEMENT tenis (#PCDATA)>  
    <!ATTLIST f1 pais CDATA #REQUIRED>  
>
```

```
<deportistas>  
    <f1 pais="España">Fernando Alonso</f1>  
    <tenis>Rafael Nadal</tenis>  
    <f1 pais="Alemania">Sebastian Vettel</f1>  
</deportistas>
```

DTD - Declaración de atributos - Ejemplo

En el ejemplo anterior:

```
<!ELEMENT deportistas (futbol | f1 | tenis)*>  
<!ELEMENT futbol (#PCDATA)>  
<!ELEMENT f1 (#PCDATA)>  
<!ELEMENT tenis (#PCDATA)>  
<!ATTLIST f1 pais CDATA #REQUIRED>
```

Puede haber múltiples deportistas dentro de "deportistas". Cada deportista será "f1", "futbol", o "tenis". En este caso, al estar separados por barras y no comas, no hay que respetar un orden.

Los deportistas de tipo f1 deben tener obligatoriamente un atributo "país".

DTD - Declaración de atributos -

Ejemplo

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

```
<!DOCTYPE deportistas [  
    <!ELEMENT deportistas (futbol | f1 | tenis)*>  
    <!ELEMENT futbol (#PCDATA)>  
    <!ELEMENT f1 (#PCDATA)>  
    <!ELEMENT tenis (#PCDATA)>  
    <!ATTLIST f1 pais CDATA #IMPLIED>  
>
```

```
<deportistas>  
    <f1 pais="España">Fernando Alonso</f1>  
    <tenis>Rafael Nadal</tenis>  
    <f1>Sebastian Vettel</f1>  
</deportistas>
```

DTD - Declaración de atributos - Ejemplo

Si cambiamos la declaración del atributo "pais" en f1:

```
<!ATTLIST f1 pais CDATA #IMPLIED>
```

Ya no será obligatoria. Podemos tener, como en el ejemplo, deportistas de F1 sin país.

Y con este otro cambio:

```
<!ATTLIST f1 pais CDATA #FIXED "España">
```

Sólo podremos usar el atributo "pais" con el valor "España". Podemos no usarlo, pero si lo usamos, debe ser España.

DTD - Declaración de atributos -

Ejemplo

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE deportistas [
    <!ELEMENT deportistas (futbol | f1 | tenis)*>
    <!ELEMENT futbol (#PCDATA)>
    <!ELEMENT f1 (#PCDATA)>
    <!ELEMENT tenis (#PCDATA)>
    <!ATTLIST f1 pais (España|Italia) #IMPLIED>
]>
<deportistas>
    <f1 pais="España">Fernando Alonso</f1>
    <tenis>Rafael Nadal</tenis>
    <f1>Sebastian Vettel</f1>
</deportistas>
```

DTD - Declaración de atributos - Ejemplo

En este caso:

```
<!ATTLIST f1 pais CDATA (España|Italia) #IMPLIED>
```

La declaración del atributo indica que es opcional, pero si lo usamos, tenemos que utilizar uno de los dos valores admitidos (España o Italia).

DTD - Declaración de atributos -

Ejemplo

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE ies [
  <!ELEMENT ies (estudios, expedientes)>
  <!ELEMENT estudios (titulo+)>
  <!ELEMENT expedientes (alumno)+>
  <!ELEMENT titulo (#PCDATA)> <!ATTLIST titulo cod ID #REQUIRED>
  <!ELEMENT alumno (#PCDATA)> <!ATTLIST alumno ciclo IDREF #REQUIRED>
]>
<ies>
  <estudios>
    <titulo cod="asir">Admon de sistemas informáticos en red</titulo>
    <titulo cod="daw">Desarrollo de aplicaciones web</titulo>
  </estudios>
  <expedientes>
    <alumno ciclo="asir">Marta</alumno>
    <alumno ciclo="daw">Antonio</alumno>
  </expedientes>
</ies>
```

DTD - Declaración de atributos -

Ejemplo

En este caso:

```
<!ELEMENT titulo (#PCDATA)> <!ATTLIST titulo cod ID #REQUIRED>
```

Al usar "ID" en el tipo del atributo "cod", el valor en estos atributos tiene que cumplir algunas reglas:

- Debe ser un nombre XML válido. Esto implica que tiene que empezar por letra o un guion bajo, y no contener espacios.
- Debe ser único en el documento. No puede haber dos atributos de tipo ID con el mismo valor. En todo el documento, no solo en elementos iguales.

DTD - Declaración de atributos -

Ejemplo

En este caso:

```
<!ELEMENT alumno (#PCDATA)> <!ATTLIST alumno ciclo IDREF #REQUIRED>
```

Al usar "IDREF" en el tipo del atributo "ciclo", el valor en estos atributos tiene que cumplir algunas reglas:

- Debe ser un nombre XML válido. Esto implica que tiene que empezar por letra o un guión bajo, y no contener espacios.
- IDREF define una relación, una referencia a otro elemento en el documento, así que su valor debe aparecer en un atributo de tipo "ID" dentro del documento.

DTD - Entidades

Ya hemos visto algunas entidades en XML, por ejemplo:

- & para representar el "ampersand" (&)
- < para representar el "menor que" (<)

DTD permite definir entidades personalizadas. Se puede hacer en el propio documento con ENTITY:

```
<!ENTITY nombre-entidad "Valor de la entidad">
```

Para usar la entidad, se usa &nombre-entidad; en el documento. Al procesar o visualizar el documento, cada aparición de &nombre-entidad; se sustituirá por el valor de esta entidad.

DTD - Entidades externas y paramétricas

Las entidades se pueden definir también en ficheros externos al DTD.

También hay otros tipos de entidades, denominadas paramétricas, que permiten cambiar el valor por el que se sustituyen.

No vamos a ver este tipo de entidades en este curso, pero si se quiere más información, hay referencia sobre ellas en los materiales del aula virtual.

DTD - Limitaciones

DTD es una tecnología anterior a XML, y en muchos casos es suficiente para validar un documento XML, pero tiene limitaciones:

- No se basa en XML, así que no se puede validar el DTD como XML.
- No se puede limitar tipos de dato (solo números, solo decimales, solo palabras, etc.)
- No hay soporte para namespaces (espacios de nombres).
- Imposible restringir valores válidos en elementos, solo en atributos.
- No se puede dar valor por defecto a elementos, solo a atributos.
- Imposible concretar cardinalidad de elementos (por ejemplo, de 4 a 7).

Para estos casos más avanzados se diseñó XSD