

DTD: Document Type Definition

M04 - Lenguaje de Marcas



Alicia Vázquez - @aliciaFPInf
alicia.vazquez@itb.cat

Bien formado vs válido

- Un documento está **bien formado** cuando sigue las reglas del XML.
- Un documento es **válido** si sigue las reglas del vocabulario y esquema que tiene asociado.
- Un documento puede estar bien formado pero no ser válido, en cambio, si es válido seguro que está bien formado

DATOS + ESTRUCTURA = INFORMACIÓN

DTD

- ❑ Definición de elementos con DTD
 - ❑ Elementos
 - ❑ Atributos
 - ❑ Entidades
-

DTD (Document Type Definition)

- Es la gramática del XML.
- Podemos escribir la definición del XML dentro del propio fichero xml o bien hacerlo en un fichero a parte que tendrás la extensión **.dtd**

Dentro del fichero xml definimos el código DTD

```
<?xml version="1.0"?>  
<!DOCTYPE nombreElementoRaiz [....códigoDTD....]>
```

DTD (Document Type Definition)

Fuera: Creamos un fichero **dtd** que luego enlazaremos desde el fichero XML. El fichero **dtd** puede ser público o del privado, que esté en nuestro sistema de archivos.

- **SYSTEM:** Para documento DTD privado, estará en el directorio raíz o en una URL nuestra.

```
<?xml version="1.0"?>
<!DOCTYPE nombreElementoRaiz SYSTEM "nombreFichero.dtd">
<!DOCTYPE raiz_FPxml SYSTEM fichero_FPxml.dtd">
```

- **PUBLIC:** Para documentos públicos, ya en uso (ej. HTML).

```
<?xml version="1.0"?>
<!DOCTYPE raíz PUBLIC “nombreDTD” “DTD_URL”>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Definición de un DTD

En un código DTD (tanto externo como interno) se pueden definir:

- Los **elementos** que se pueden utilizar en un documento XML. En esta definición se indica además **que pueden contener** dichos elementos.
- Los **atributos** que pueden poseer los elementos. Además incluso indicando sus posibles valores válidos.
- Las **entidades/restricciones** que puede utilizar el documento XML.

Un DTD permite definir la ESTRUCTURA que debe tener nuestro XML → El árbol.

DTD

- ❑ Definición de elementos con DTD
 - ❑ Elementos
 - ❑ Atributos
 - ❑ Entidades
-

ELEMENTOS

Instrucción

<!ELEMENT nombre_elemento DEF_CONTENIDO>

- **nombre_elemento:** Es el nombre del elemento que debe cumplir con las normas de nombre marcadas por XML.
- **DEF_CONTENIDO**
 - **EMPTY:** Es un elemento que no puede tener contenido, pero puede tener atributos.
 - **ANY:** Puede contener texto, otro tipo de datos y cualquier etiqueta. Además puede tener atributos.
 - **(#PCDATA):** Indica que contiene texto (Parseable Character DATA), datos de carácter analizados sintácticamente por el navegador.
 - **(hijo1, hijo2, hijo3):** Hace referencia al nombre/s de otros elementos que serán hijos del elemento que se está definiendo.

ELEMENTOS *<!ELEMENT nombre_elemento DEF_CONTENIDO>*

```
<!DOCTYPE persona [  
    <!ELEMENT persona (nombre, mujer, fecha_de_nacimiento, ciudad)>  
    <!ELEMENT nombre (#PCDATA)>  
    <!ELEMENT mujer EMPTY>  
    <!ELEMENT fecha_de_nacimiento (dia, mes, anyo)>  
    <!ELEMENT dia (#PCDATA)>  
    <!ELEMENT mes (#PCDATA)>  
    <!ELEMENT anyo (#PCDATA)>  
    <!ELEMENT ciudad (#PCDATA)>  
]>
```

```
<persona>  
    <nombre>Maria</nombre>  
    <mujer/>  
    <fecha_de_nacimiento>  
        <dia>21</dia>  
        <mes>10</mes>  
        <anyo>1998</anyo>  
    </fecha_de_nacimiento>  
    <ciudad>Bilbao</ciudad>  
</persona>
```

ELEMENTOS COMPLEJOS

<!ELEMENT nombre_elemento (hijo1, hijo2, hijo3)>

Conectores

| (indica que uno u otro)

, (la coma, indica que ambos elementos deben estar además informa de una secuencia)

`<!ELEMENT persona (nombre | apellido)>` La persona debe tener nombre o apellido (ambos daría error.)

`<!ELEMENT persona (nombre, apellido)>` La persona debe tener nombre y apellido

ELEMENTOS COMPLEJOS

<!ELEMENT nombre_elemento (hijo1, hijo2, hijo3)>

Cardinalidad

* : 0 a n veces

+ : 1 a n veces

? : 0 o 1 vez.

Sin símbolo: debe aparecer una y solo una vez.

```
<!ELEMENT libro (introduccion?, capitulo+, conclusion?)>
<!ELEMENT capitulo (autor*, parrafo+)>
<!ELEMENT libro (autor?, capitulo)+>
```

Nota: El orden en el que definamos a los “hijos” en el contenido importa!

Ejemplo

EN UN SOLO
FICHERO → XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE marcadores [
    <!ELEMENT marcadores (pagina)*>
    <!ELEMENT pagina (nombre, descripcion, url)>
    <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT descripcion (#PCDATA)>
    <!ELEMENT url (#PCDATA)>
]>

<marcadores>
    <pagina>
        <nombre>Openwebinars</nombre>
        <descripcion>Tutoriales de
informática.</descripcion>
        <url>http://www.openwebinars.com/</url>
    </pagina>
    <pagina>
        <nombre>Wikipedia</nombre>
        <descripcion>La enciclopedia
libre.</descripcion>
        <url>http://www.wikipedia.org/</url>
    </pagina>
</marcadores>
```

Ejemplo

EN DOS
FICHEROS →
DTD + XML

```
<!ELEMENT marcadores (pagina)*>
<!ELEMENT pagina (nombre, descripcion, url)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT descripcion (#PCDATA)>
<!ELEMENT url (#PCDATA)>
```

marcadores.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE marcadores SYSTEM "marcadores.dtd">
<marcadores>
    <pagina>
        <nombre>Openwebinars</nombre>
        <descripcion>Tutoriales de
informática.</descripcion>
        <url>http://www.openwebinars.com/</url>
    </pagina>
    <pagina>
        <nombre>Wikipedia</nombre>
        <descripcion>La enciclopedia
libre.</descripcion>
        <url>http://www.wikipedia.org/</url>
    </pagina>
</marcadores>
```

marcadores.xml

Ejercicio 1

Crea el DTD que valide este xml y comprueba que valide.

```
<Empresa>
  <Empleado>
    <Apellido>Perez</Apellido>
    <Nombre>Juan</Nombre>
    <NumEmpleado>1234567890</NumEmpleado>
    <Email>juanperez@gmail.com</Email>
    <Telefono>666 555 444</Telefono>
    <Direccion>
      <Calle>Carrer de Pau Claris, 121</Calle>
      <Ciudad>Barcelona</Ciudad>
      <CP>08009</CP>
    </Direccion>
  </Empleado>
</Empresa>
```

Ejercicio 2

Basándose en el XML del ejercicio 2, haz varias versiones de DTD y los cambios pertinentes que se tengan que hacer en el XML.

3.1 Dos apellidos

3.2 Email o Telefono, no las dos cosas

3.3 Varios Empleados

Ejercicio 3

Dado siguiente DTD escribe un document XML que valide contra el siguiente DTD

```
<!ELEMENT matricula (ciclo,alumnos)>
<!ELEMENT ciclo (#PCDATA)>
<!ELEMENT alumnos (alumno)+>
<!ELEMENT alumno (nombre,year,asignaturas)>

<!ELEMENT nombre (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT asignaturas (asignatura)+>
<!ELEMENT asignatura (#PCDATA)>
```

DTD

- Definición de elementos con DTD
 - Elementos
 - Atributos
 - Entidades
-

ATRIBUTOS

Instrucción

```
<!ATTLIST nombre_elemento  
nombre_atributo TYPE OBLIGATION "valor_por_defecto">
```

- **nombre_elemento:** Es el nombre del elemento al que hace referencia el atributo
- **nombre_atributo:** Es el nombre del atributo que debe cumplir con las normas de nombre marcadas por XML.

ATRIBUTOS type - CDATA

`<!ATTLIST nombre_elemento nombre_atributo TYPE OBLIGATION>`

TYPE Es el tipo de valores que podemos asignar al atributo.

CDATA: texto (Character Data), su contenido no es procesado, lo que significa que puede contener cualquier valor (incluidos símbolos prohibidos en los #PCDATA como <, >, &, etc).

```
<?xml version = "1.0"?>
<!DOCTYPE address [
  <!ELEMENT address ( name )>
  <!ELEMENT name ( #PCDATA )>
  <!ATTLIST name id CDATA #REQUIRED>
]>
<address>
  <name id="123">
    Tanmay Patil
  </name>
</address>
```

ATRIBUTOS obligation

<!ATTLIST nombre_elemento nombre_atributo TYPE OBLIGATION>

#REQUIRED : indica que **siempre** hay que darle un valor a ese atributo y no puede tener un valor por defecto. El atributo es **obligatorio**.

```
<?xml version = "1.0"?>
<!DOCTYPE address [
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name id CDATA #REQUIRED>
]>
<address>
  <name id="123">
    Tanmay Patil
  </name>
</address>
```

ATRIBUTOS obligation

<!ATTLIST nombre_elemento nombre_atributo TYPE OBLIGATION >

#IMPLIED : Si el atributo que se declara no tiene un valor predeterminado, no tiene valor fijo, ni es necesario que esté. El atributo es **opcional**.

@aliciaFPlnf

```
<?xml version = "1.0"?>
<!DOCTYPE address [
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name id CDATA #IMPLIED>
]>
<address>
  <name />
</address>
```

ATRIBUTOS obligation

```
<!ATTLIST nombre_elemento nombre_atributo TYPE OBLIGATION "valor_por_defecto">
```

#FIXED : Se usan cuando se quiere especificar que el valor del atributo está **fijado** y no se puede cambiar. Si toma otra valor el XML no será válido. En este caso debemos definir qué valor tiene que tener para ser validado.

valor_por_defecto Contiene un valor predeterminado. Los valores se pueden rodear con comillas independientes ('') o dobles("").

```
<!DOCTYPE persona [  
    <!ELEMENT persona (#PCDATA)>  
    <!ATTLIST persona nacionalidad CDATA #FIXED "Española">  
]>
```

```
<persona nacionalidad="Española"> Maria</persona>
```

Ejercicio 4

Basándonos en el ejemplo de la página 9, cambia los ficheros de tal manera que **dia**, **mes**, **anyo** sean atributos **dia** y **mes** requeridos y **anyo** con valor “2000” por defecto.

Ejercicio 5

Basándonos en el Ejercicio02 XML cambia los ficheros de tal manera que:

- **NumEmpleado** pase a ser un atributo de **Empleado**.
- **Calle, CP y Ciudad**, Atributos de Dirección siendo **Ciudad** un atributo opcional y **Calle y CP**, obligatorios.

ATRIBUTOS type

<!ATTLIST nombre_elemento nombre_atributo TYPE OBLIGATION VALOR_POR_DEFECTO>

TYPE Es el tipo de valores que podemos asignar al atributo puede contener:

- **CDATA:** Puede contener cualquier cadena de caracteres.
- **ENUMERATION:** Se utiliza para definir un valor dentro de un conjunto de valores.
- **ID:** Se utiliza como identificador de un elemento y su valor será único.
- **IDREF o IDREFS:** Referencias a IDs que han de existir. El plural es para definir listas de ID's
- **ENTITY o ENTITIES** Permiten definir constantes .
- **NMTOKEN o NMTOKENS:** Cadenas de caracteres permitidos por xml.
- **NOTATION:** Notaciones declaradas anteriormente.

ATRIBUTOS type

<!ATTLIST nombre_elemento nombre_atributo TYPE OBLIGATION VALOR_POR_DEFECTO>

ID: identificador único en todo el XML(combinación de cifras y/o letras). Solo puede haber un atributo de tipo ID en el elemento.

Los IDs son especialmente útiles para las herramientas de maquetación, análisis y programación de aplicaciones XML. Ya que permiten **diferenciar de manera única** a cada elemento. Es como la clave en BBDD.

ATRIBUTOS type

IDREF : Los atributos IDREF contienen como valor el identificador de otro elemento. Es decir será una referencia a otro elemento, a un **ID** de otro elemento.

Debe existir un atributo **ID** en el documento XML cuyo valor coincida con el especificado en un **IDREF**. Es como la **clave foranea** en BBDD.

@aliciaFPInf

ATRIBUTOS type

```
<!-- Archivo directorio.dtd -->
<!ELEMENT directorio (persona)+ >
<!ELEMENT persona (#PCDATA) >
<!ATTLIST persona id  ID #REQUIRED madre  IDREF #IMPLIED padre  IDREF
#IMPLIED>
```

```
<!-- Archivo directorio1.xml-->
<!DOCTYPE directorio SYSTEM "directorio.dtd">
<directorio>
    <persona id="p1">Pedro</persona>
    <persona id="p2">Marisa</persona>
    <persona id="p3" madre="p2" padre="p1">Carmen</persona>
</directorio>
```

Carmen es la hija de Pedro y Marisa, según el código anterior, ya que los atributos padre y madre de tipo **IDREF** contienen los **ID** de Pedro y Marisa.

ATRIBUTOS type

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE classificacio [
<!ELEMENT classificacio (equip*)>
<!ELEMENT equip (#PCDATA) >
<!ATTLIST equip posicio ID #REQUIRED>
]>
```

```
<classificacio>
  <equip posicio="primer">F.C.Barcelona</equip>
  <equip posicio="primer">Reial Madrid</equip>
</classificacio>
```

No valida al tener el mismo valor!!

Ejercicio 6

1. Escribir un DTD que valide una librería de libros que contenga un ISBN único, ten en cuenta que el parámetro ID tiene que empezar por un carácter no numérico

```
<?xml version="1.0" encoding="UTF-8"?>
<libreria>
    <libro ISBN="A9788497592581" titulo="El origen de las especies"/>
    <libro ISBN="A978849759251" titulo="La Celestina"/>
    <libro ISBN="A8481301264" titulo="El nombre de la Rosa"/>
</libreria>
```

2. Ahora añade una compra / un ticket en el cual se esté haciendo referencia a uno de los libros de la librería (IDREF).

ATRIBUTOS type

IDREFS: Igual que el anterior (**IDREF**), solo que, en este caso, se permite indicar varias referencias (que deben existir en el documento XML) a otros ID, separadas por espacios.

```
<!DOCTYPE directorio [  
<!ELEMENT directorio (persona)+>  
<!ELEMENT persona (#PCDATA)>  
<!ATTLIST persona id ID #REQUIRED padres IDREFS #IMPLIED > ]>  
<directorio>  
  <persona id="p1">Pedro</persona>  
  <persona id="p2">Marisa</persona>  
  <persona id="p3" padres="p1 p2">Carmen</persona>  
</directorio>
```

Ejercicio 6

3. Usa IDREFS en lugar de IDREF para poder poner en un solo ticket varias compras.

ATRIBUTOS type

Enumeración de valores Se trata de atributos a los que se indica una serie de posibles valores. El valor del atributo debe coincidir con alguno de la lista. Los valores posibles se indican entre paréntesis separados por el símbolo |.

```
<!ATTLIST nombre_elemento nombre_atributo (valor1 | valor2 | valor3 | valor4)  
“valor_por_defecto”>
```

ATRIBUTOS type

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE deportistas [
    <!ELEMENT deportistas (futbol*, f1*, tenis*)>
    <!ELEMENT futbol (#PCDATA)>
    <!ELEMENT f1 (#PCDATA)>
        <!ATTLIST f1 pais ( ESP | FRA | ITA | ALE ) "ESP">
    <!ELEMENT tenis (#PCDATA)>
]>

<deportistas>
    <f1 pais="ALE">Sebastian Vettel</f1>
    <f1>Fernando Alonso</f1>
    <f1 pais="ESP">Carlos Sainz</f1>
    <tenis>Rafael Nadal</tenis>
</deportistas>
```

Ejercicio 7

Escribir un DTD que permita únicamente en los atributos de la etiqueta curso los meses del año y los cursos 2019 | 2020 | 2021.

```
<matricula>
    <curso year="2019" mes="abril"/>
</matricula>
```

Ejercicio 8*

Escribir un DTD que valide una biblioteca de libros que contengan un ISBN único, usuarios que contengan un DNI único y préstamos de libros a usuarios . Utiliza Un DTD empleando los atributos ID i IDREF . Además clasifica los libros según si son: clásico, novela o teatro, siendo por defecto un clásico.

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro ISBN="A978849759251" titulo="La Celestina" tipo="clasico"/>
    <libro ISBN="A8481301264 " titulo="El nombre de la Rosa"/>
    <libro ISBN="A9788497592581" titulo="Don Juan" tipo="teatro"/>

    <usuario DNI="Z47673211" nombre="Antonio"/>
    <usuario DNI="Z47673212" nombre="Maria"/>

    <prestamo DNI="Z47673211" ISBN="A8481301264 A9788497592581"/>
</biblioteca>
```

ATRIBUTOS type

NMTOKEN : Los atributos NMTOKEN contienen una cadena de caracteres, pudiendo contener letras **minúsculas**, letras **mayúsculas**, **números**, puntos ".", guiones medios "-", guiones bajos "_" o el carácter dos puntos ":".

No se pueden escribir **espacios en blanco** ni caracteres especiales, tales como: *, \$, %, &, ?, @...

@alic

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE usuarios [
    <!ELEMENT usuarios (usuario)*>
    <!ELEMENT usuario (#PCDATA)>
    <!ATTLIST usuario clave NMTOKEN #REQUIRED>
]>

<usuarios>
    <usuario clave="123456789">Ana</usuario>
    <usuario clave="ab-c-d-fg">Iker</usuario>
    <usuario clave="A1_B2..C3">Elsa</usuario>
</usuarios>
```

ATRIBUTOS type

NMTOKENS El atributo puede contener varios valores de tipo **NMTOKEN** separados por espacios.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE usuarios [
    <!ELEMENT usuarios (usuario)*>
    <!ELEMENT usuario (#PCDATA)>
    <!ATTLIST usuario codigos NMTOKENS #REQUIRED>
]>

<usuarios>
    <usuario codigos="1234 567 89">Ana</usuario>
    <usuario codigos="ab c-d fg">Iker</usuario>
    <usuario codigos="A1:B2">Elsa</usuario>
</usuarios>
```

ATRIBUTOS type

ENTITY

El valor de un atributo será una **entidad** de la cual se indica el nombre.

ENTITIES

El valor del atributo será una lista de nombres de **entidades** separadas por espacios.

@alicia@Plnf

DTD

- ❑ Definición de elementos con DTD
 - ❑ Elementos
 - ❑ Atributos
 - ❑ Entidades
-

ENTIDADES

Las entidades son elementos XML que permiten indicar abreviaturas de texto (o referencias a elementos externos abreviadas) o utilizar caracteres que de otra forma serían inválidos en el documento.

- **ENTIDADES YA EXISTENTES**

Estas entidades **no** hay que declararlas en ningún DTD, todos los analizadores de XML estándar conocen estas entidades.

```
<autor>Leopoldo Alas  
&apos;Clarín&apos;</autor>
```

ENTIDAD	SIGNIFICADO
<	El símbolo de menor (<)
>	El símbolo de mayor (>)
&	El ampersand (&)
'	La comilla simple ('')
"	La comilla doble ("")

ENTIDADES

Para definir una **Entidad** nueva:

<!ENTITY nombre_Entity CONTENIDO>

Asociamos un **nombre** a un **valor**, que luego será por lo que sustituiremos, igual que si fuera una constante.

@aliciaFPlnf

ENTIDADES contenido

Texto de sustitución de la entidad: Representa el contenido por el que se sustituirá la referencia a la entidad dentro del documento:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE frase [
    <!ELEMENT frase (#PCDATA)>
    <!ENTITY color "azul">
    <!ENTITY cielo "El cielo es &color;.">
]>
<frase>&cielo;</frase>
```

This XML file does not appear to have any style

```
<frase>El cielo es azul.</frase>
```

ENTIDADES contenido

Entidad Externa: La ubicación del archivo que contiene la entidad. Debe ir precedido por las palabras **SYSTEM** y posteriormente la URI/URL correspondiente al archivo dónde encontraremos el valor de dicha entidad.

```
<!ENTITY entity-name SYSTEM "URI/URL">  
  
<!ENTITY writer SYSTEM "https://www.w3schools.com/entities.dtd">  
<!ENTITY copyright SYSTEM "https://www.w3schools.com/entities.dtd">  
  
<author>&writer; &copyright;</author>
```

Ejercicio 9: Utilización de un DTD

- Crear el DTD siguiente:

carnet.dtd

```
<!ELEMENT carnet (persona+)>
<!ELEMENT persona EMPTY>
<!ATTLIST persona
  nombre CDATA #REQUIRED
  apellidos CDATA #IMPLIED telefono CDATA #REQUIRED>
```

- Crea dos documentos XML que sea validado por este DTD.

Referencias

Referencia oficial: https://www.w3schools.com/xml/xml_dtd_intro.asp

Ejemplos de cada propiedad del DTD: <http://www.mclibre.org/consultar/xml/lecciones/xml-dtd.html>

Manual XML: <https://jorgesanchez.net/manuales/xml/introduccion-lenguajes-de-marcas.html>

@aliciaFPlnf