
XML. DTD.



1. Document Type Definition (DTD)	3
1.1.¿Qué es? ¿Cuál es su uso?	3
1.2. Asociar un XML con un DTD para su validación	4
1.3. Tipos de Declaraciones de un DTD	5
1.3.1. Declaración de un Elemento - ELEMENT	5
1.3.2. Declaración de un Atributo - ATTLIST	8
1.3.3. Vídeo: Como hacer un DTD básico.	11
1.3.4. Declaración de una Entidad - ENTITY	15
2. Enlaces de interés	18

1. Document Type Definition (DTD)

1.1. ¿Qué es? ¿Cuál es su uso?

Se utiliza para definir la **gramática** (estructura y sintaxis) de un documento XML. La función de un DTD es describir la estructura que debe tener un determinado documento XML, con la intención de mantener la consistencia entre los distintos documentos y permitir la validación de los documentos. Con esto se facilita la compartición de los documentos.

Un DTD, **puede definir**:

- Los **elementos que puede contener** el XML, sus etiquetas iniciales y finales.
- Sus **atributos y los valores válidos** para esos atributos.
- Su estructura indicando el **orden** en que se deben de poner **los distintos elementos** en el XML.

Ejemplo lista de personas. DTD:

```
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?,
                   sexo?, numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA)>
```

XML se puede corresponder con el DTD anterior:

```
<lista_de_personas>
  <persona>
    <nombre>Tim Berners-Lee</nombre>
    <fechanacimiento>8 de junio, 1955</fechanacimiento>
  </persona>
</lista_de_personas>
```

1.2. Asociar un XML con un DTD para su validación

Siempre debe estar situado en el **prólogo** y puede ser un **DTD interno o externo**:

- **DTD Interno**

```
<!DOCTYPE NOTICIA [...] >
```

```
<?xml version="1.0" ?>
<!DOCTYPE pedido [
  <!ELEMENT pedido (producto*) >
  <!ELEMENT producto (nombre,cantidad)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT cantidad (#PCDATA)>
]>
<pedido>
  <producto>
    <nombre>Rotulador RX2</nombre>
    <cantidad>20</cantidad>
  </producto>
  <producto>
    <nombre>Grapadora Lin</nombre>
    <cantidad>2</cantidad>
  </producto>
</pedido>
```

- **DTD Externo**

- **PUBLIC:** global, un estándar que se vaya a utilizar.

```
<!DOCTYPE elementoRaiz PUBLIC "urlDTD">
<!DOCTYPE elementoRaiz PUBLIC "nombreDTD" "urlDTD">
```

- **SYSTEM:** local, definido por el usuario(privado)

```
<!DOCTYPE elementoRaiz SYSTEM "urlDTD">
```

```
<?xml version="1.0" ?>
<!DOCTYPE pedido SYSTEM "http://www.ej.org/pedidos.dtd">
<pedido>
  <producto>
    <nombre>Rotulador RX2</nombre>
    <cantidad>20</cantidad>
  </producto>
  <producto>
    <nombre>Grapadora Lin</nombre>
    <cantidad>2</cantidad>
  </producto>
</pedido>
```

```
<?xml version="1.0"?>
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.0//EN"
  "http://www.w3c.org/TR/REC-html/strict.dtd">
<HTML>
  <HEAD><TITLE>Ejemplo HTML</TITLE></HEAD>
  <BODY>
    <H1>HTML de ejemplo</H1>
  </BODY>
</HTML>
```

```
http://www.ej.org/pedidos.dtd
<!ELEMENT pedido (producto*) >
<!ELEMENT producto (nombre,cantidad)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT cantidad (#PCDATA)>
```

```
<?xml version="1.0" ?>
<!DOCTYPE pedido SYSTEM "http://www.ej.org/pedidos.dtd" [
  <!ATTLIST producto fecha CDATA #IMPLIED >
]>
<pedido>
  <producto>
    <nombre>Rotulador RX2</nombre>
    <cantidad>20</cantidad>
  </producto>
  <producto fecha="Julio">
    <nombre>Grapa</nombre>
    <cantidad>2</cantidad>
  </producto>
</pedido>
```

- **DTD Mixta:** Podemos añadir nuevas definiciones a la validación externa, como en el siguiente ejemplo:

AC2.1: Implementar el DTD de lista de personas. A continuación crea un XML que cumpla con la especificación del XML (mínimo tres personas). Valida el XML con el DTD creado. Hazlo con DTD interno y con el DTD externo.

Validar documento XML usando Visual Studio Code, Netbeans o XML Copy Editor:

- <https://www.mclibre.org/consultar/xml/otros/xmlcopyeditor.html>

1.3. Tipos de Declaraciones de un DTD

Un DTD está compuesto por **declaraciones**:

- Que van encerradas entre: `<! >`
- Existen **4 tipos** de declaraciones, aunque las más usadas son las dos primeras, que son:
 - Elemento (**ELEMENT**)
 - Atributo (**ATTLIST**)
 - Entidad (**ENTITY**)
 - Notación (**NOTATION**)

1.3.1. Declaración de un Elemento - ELEMENT

Define un elemento(etiqueta) valido en el documento XML: `<!ELEMENT nombre (contenido)>`

El **nombre** de un elemento debe seguir las siguientes reglas:

- Primer carácter: letra | _ | :
- Resto de caracteres: letra | _ | : | dígitos | . | -

El **contenido** puede ser:

- **EMPTY**: significa que el elemento es vacío, es decir, que no puede tener contenido. Los elementos vacíos pueden escribirse con etiquetas de apertura y cierre sin nada entre ellos, ni siquiera espacios, o con una etiqueta vacía. EMPTY debe escribirse sin paréntesis.
- **ANY**: el elemento puede contener cualquier cosa (texto y otros elementos). Sin paréntesis.
- **(#PCDATA)**: el elemento puede contener texto, la información. Debe escribirse entre paréntesis.
- **Modelo de grupo**
- **Modelo mixto**

Modelo de grupo: representa una secuencia concreta de elementos que definen el contenido de un elemento. Puede estar formado por:

- **Lista**:
 - , —> de tipo secuencia, existen todas.
 - | —> de tipo elección, existen unas u otras.
- **Cuantificador**:
 - ? —> deben de haber 0 ó 1 ocurrencia.
 - + —> deben de haber 1 ó más ocurrencias.
 - * —> deben haber 0 ó más ocurrencias.
 - **sin cuantificador** —> debe de aparecer una vez.

(?)	= 0, 1 elemento
(*)	= 0 ó más elementos
(+)	= 1 ó más elementos
()	= alternativa
(,)	= secuencia
EMPTY	= vacío
ANY	= cualquier estructura de subelementos
#PCDATA	= cadena de caracteres analizados

AC2.2: Determinar si el fragmento XML se corresponde con la declaración mostrada.

```
<!ELEMENT algo (a, b?, c+, d*)>  
  
<algo>  
<a /><c /><c></c><d></d><d />  
</algo>
```

```
<!ELEMENT algo ((a | b), c?)>  
  
<algo>  
<a /><b />  
<c></c>  
</algo>
```

SI

```
<!ELEMENT algo ((a | b)+, c?)>  
  
<algo>  
<a /><b /><a /><a /><b /><c />  
</algo>
```

```
<!ELEMENT algo ((a, b)+, (c | d)+)>  
  
<algo>  
<a /><b /><a /><c /><d />  
</algo>
```

NO

AC2.3: Crear un elemento XML válido y uno no válido para cada uno de los elementos del DTD que aparecen en la imagen.

```
<!ELEMENT pizza (ingrediente*, inventor?)>  
<!ELEMENT servicio (domicilio | restaurante)>  
<!ELEMENT ingrediente EMPTY>  
<!ELEMENT inventor (#PCDATA)>  
<!ELEMENT sección (título, (contenido | sección+))>
```

Recursividad

AC2.4: Definir un DTD para validar el documento de la imagen. Hacerlo como DTD interno y externo.

```
<libro>  
  <título>La Primavera</título>  
  <autor>Juan Flores</autor>  
  <sección>  
    <nombre>Introducción</nombre>  
    <contenido>  
      <parrafo>Esta es la Introducción</parrafo>  
    </contenido>  
  </sección>  
  <sección>  
    <nombre>Conclusiones</nombre>  
    <contenido>  
      <parrafo>Las flores salen</parrafo>  
      <parrafo>La sangre se altera</parrafo>  
    </contenido>  
  </sección>  
</libro>
```

Modelo mixto: nos permite mezclar texto con otros elementos. Con este modelo **NO se puede restringir ni el orden, ni el número de ocurrencias**. Su definición siempre tiene que ser de la siguiente forma:

- Su primer elemento debe ser siempre #PCDATA.
- Los distintos elementos se deben separar con |
- Todo el grupo debe ser repetirle cero o más veces (*)

Veamos un ejemplo de la definición de una etiqueta p:

```
<!ELEMENT p (#PCDATA | strong | em)*>

<p>
Este <strong>elemento</strong> tiene
<em>contenido mixto</em>.
</p>
```

```
<comentario>
Este texto está relacionado con el
<enlace href="http://www.quijote.com">Quijote</enlace> y ha sido
realizado por <persona>Leopoldo Alas Clarín</persona>
</comentario>
```

```
<!ELEMENT comentario (#PCDATA | enlace | persona)* >

<!ELEMENT enlace (#PCDATA)>
<!ELEMENT persona (#PCDATA)>

<!ATTLIST enlace href CDATA #REQUIRED>
```

AC2.5: Crea un DTD que valide el siguiente documento XML.

```
<lista_videos>
<video>
    <titulo>La vida del Programador</titulo>
    <canal>Todo no es programar</canal>
    <visualizaciones>1231231</visualizaciones>
    <fecha_subida>20-10-2000</fecha_subida>
    <comentarios>
        <comentario>
            El mejor video que he visto
            <usuario>Luis Antonio</usuario>
        </comentario>
        <comentario>
            Este video está en todos sitios....
            <usuario>Luis Antonio</usuario>
            <enlace>www.enlace.com</enlace>
        </comentario>
    </comentarios>
</video>
</lista_videos>
```

1.3.2. Declaración de un Atributo - ATTLIST

Definen los distintos atributos que se pueden emplear con un determinado elemento. La definición de los atributos tienen la siguiente estructura:

```
<?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>
```

Forma 1:

```
<!ATTLIST nombreElemento nombreAtributo tipoAtributo valorInicial >
```

Forma 2:

```
<!ATTLIST nombreElemento nombreAtributo1 tipoAtributo1 valorInicial1 >
<!ATTLIST nombreElemento nombreAtributo2 tipoAtributo2 valorInicial2 >
```

Forma 3:

```
<!ATTLIST nombreElemento
  nombreAtributo1 tipoAtributo1 valorInicial1
  nombreAtributo2 tipoAtributo2 valorInicial2
>
```

En las que:

- "nombreElemento" es el nombre del elemento para el que se define un atributo.
- "nombreAtributo" es el propio nombre del atributo.
- "tipoAtributo" es el tipo de atributo .
- "valorPorDefecto" es el valor predeterminado del atributo.

Tipos de datos en atributos

CDATA = Cadena de caracteres
NMOKEN = Palabra (sin espacios)
NMOKENS = Lista de palabras
Enumeración separada por |

Valor de los Atributos

#REQUIRED	Obligatorio
#IMPLIED	Opcional
#FIXED	Constante
Valor	Valor por defecto

```
<!ATTLIST pizza nombre CDATA #REQUIRED>
<!ATTLIST ingrediente nombre CDATA #REQUIRED
  calorías CDATA #IMPLIED>
<!ATTLIST precio moneda (euros|dólares) #REQUIRED
  valor CDATA #REQUIRED>
<!ATTLIST impuesto tipo CDATA #FIXED "IVA">
```

```
<pizza nombre="4 estaciones" >
  <ingrediente nombre="Jamón" />
  <precio moneda="euros" valor="7" />
</pizza>
```

Tipos de Atributos (Extendido)	
Tipo	Descripción
CDATA	Puede contener cualquier carácter (sin restricciones).
ID (Identificación)	Su valor NO se puede repetir en otros elementos o atributos, debe de ser único. Su valor no puede empezar por un número.
IDREF	Se usa para hacer referencia a la identidad de otro elemento. Se utiliza para establecer conexiones entre los elementos. Su valor debe coincidir con el valor del atributo ID de otro elemento. Junto con ID, permiten definir integridad referencial básica para valores de atributos.
IDREFS	Se usa para referenciar múltiples identidades. Su valor es una serie de valores separados por espacios que coinciden con el valor de los atributos ID de otros elementos.
ENTITY	El nombre del atributo debe ser el nombre de una entidad externa.
ENTITIES	Representa una lista de entidades externas separadas por espacios en blanco.
NMTOKEN	El valor del atributo debes ser un nombre XML válido, por lo que puede contener letras, dígitos, punto, guión, subrayado y dos puntos. NO puede contener espacios en blanco
NMTOKENS	Puede contener los mismos caracteres que el NMTOKEN y además uno o más espacios en blanco, retornos de carro o tabuladores. El valor del atributo debe ser una lista de nombres XML válidos.
NOTATION	Su valor debe ser una anotación declarada en el documento DTD.
Grupo de valores enumerados	Permite definir valores de una lista. La lista se muestra entre paréntesis con los términos separados por la barra vertical " ". Los términos deben estar entre comillas simples o dobles si tienen espacios en blanco.

```
<!ATTLIST img
    print (yes | no) "yes">
```

```
<!ATTLIST persona código ID #REQUIRED>
<!ATTLIST dueño código IDREF #REQUIRED>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE futbol [
    <!ELEMENT futbol ((jugador|equipo)*)>
    <!ELEMENT jugador EMPTY>
    <!ATTLIST jugador nombre NMOKENS #REQUIRED>
    <!ATTLIST jugador codigo ID #REQUIRED>
    <!ELEMENT equipo EMPTY>
    <!ATTLIST equipo nombre CDATA #REQUIRED>
    <!ATTLIST equipo jugadores IDREFS #IMPLIED>
]>

<futbol>
    <jugador nombre="Alfredo Di Stéfano" codigo="ads"/>
    <jugador nombre="Edison Arantes do Nascimento" codigo="ean">
    <jugador nombre="Diego Armando Maradona" codigo="dam" />
    <jugador nombre="Johan Cruyff" codigo="jc" />
        <equipo nombre="Società Sportiva Calcio Napoli"
    jugadores="dam" />
        <equipo nombre="Futbol Club Barcelona" jugadores="jc dam" />
</futbol>
```

```
<persona código="23" nombre ="Juan" />
<persona código="35" nombre ="Pepe" />
<persona código="37" nombre ="Luis" />

<dueño código="35" />
```

Valores Iniciales o por defecto	
Valores	Descripción
# REQUIRED	El atributo es obligatorio, aunque no se especifica ningún valor predeterminado.
# IMPLIED	El atributo NO es obligatorio y NO tiene ningún valor predeterminado. Es un atributo OPCIONAL.
# FIXED "valor"	El atributo debe tener un valor fijo.
"valor"	El atributo tiene un valor por defecto.

Ejemplos: ¿Cuál será el válido?

```
<?xml version="1.0"?>
<!DOCTYPE address [
  <!ELEMENT address (company)*>
  <!ELEMENT company (#PCDATA)>
  <!ATTLIST company name NMOKEN #FIXED "tutorialspoint">
]>
<address>
  <company name="abc">we are a free online teaching faculty</company>
</address>
```

```
<?xml version="1.0"?>
<!DOCTYPE address [
  <!ELEMENT address (company)*>
  <!ELEMENT company (#PCDATA)>
  <!ATTLIST company name NMOKEN #FIXED "tutorialspoint">
]>
<address>
  <company name="tutorialspoint">we are a free online teaching faculty</company>
</address>
```

Más ejemplos: ¿Cuál de los siguientes documentos XML validaría con esta DTD?

```
<!ELEMENT d (p*)>
<!ELEMENT p (#PCDATA|v)*>
<!ELEMENT v (#PCDATA)>
<!ATTLIST d m CDATA #IMPLIED>
```

1 <d>
<p m="3"><v>x</v></p>
</d>

3 <d m="3">
<p><v>x</v>=2+<v>y</v></p>
</d>

2 <d>
<p>x</p>=2+<v>y</v>
</d>

4 <d m="3">
<p>x=2+<v>y</v>
</d>

AC2.6: Implementa el DTD que valide con el siguiente XML.

```
<futbol>
  <jugador nombre="Alfredo Di Stéfano" codigo="ads"/>
  <jugador nombre="Edison Arantes do Nascimento" codigo="ean" />
  <jugador nombre="Diego Armando Maradona" codigo="dam" />
  <jugador nombre="Johan Cruyff" codigo="jc" />
    <equipo nombre="Società Sportiva Calcio Napoli" jugadores="dam" />
    <equipo nombre="Futbol Club Barcelona" jugadores="jc dam" />
  </futbol>
```

Más ejemplos:

```
<?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>
```

```
<!ATTLIST img
  src CDATA #IMPLIED
  align (left | center | right) "left"
  size NMOKEN #REQUIRED>
```

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

```
<!ATTLIST alumno
  nombre CDATA #REQUIRED
  matricula ID #REQUIRED
  curso CDATA #FIXED "Primero">
```

1.3.3. Vídeo: Como hacer un DTD básico.

Vídeo. Como hacer un DTD básico: <https://www.youtube.com/watch?v=fPU1ex7bSgg>

AC2.7: Ejercicios que debes corregir sin modificar la DTD. Comprobar que ya son documentos válidos usando Visual Studio Code, XML Copy Editor o Netbeans.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE viajes [
    <!ELEMENT viajes (austria, italia, francia)>
    <!ELEMENT austria (ciudad+)>
    <!ELEMENT italia (ciudad+)>
    <!ELEMENT francia (ciudad+)>
    <!ELEMENT ciudad (#PCDATA)>
]>

<viajes>
    <austria>
        <vienna/>
    </austria>
    <italia>
        <florence/>
        <roma/>
        <venecia/>
    </italia>
    <francia/>
</viajes>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bebidas [
    <!ELEMENT bebidas (bebida)*>
    <!ELEMENT bebida ((codigo | nombre), precio)>
    <!ELEMENT codigo (#PCDATA)>
    <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT precio (#PCDATA)>
]>

<bebidas>
    <bebida>
        <codigo>L45</codigo>
        <nombre>leche</nombre>
        <precio>8</precio>
    </bebida>
    <bebida>
        <precio>6</precio>
    </bebida>
</bebidas>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE geografia [
    <!ELEMENT geografia (paises, ciudades)>
    <!ELEMENT paises (pais)*>
        <!ELEMENT pais (#PCDATA)>
        <!ATTLIST pais codpais ID #REQUIRED>
    <!ELEMENT ciudades (ciudad)*>
        <!ELEMENT ciudad (#PCDATA)>
        <!ATTLIST ciudad pais IDREF #IMPLIED>
]>

<geografia>
    <paises>
        <pais codpais="1">Argentina</pais>
        <pais codpais="2">Austria</pais>
        <pais codpais="3">Japón</pais>
        <pais codpais="4">Noruega</pais>
    </paises>
    <ciudades>
        <ciudad pais="3">Osaka</ciudad>
        <ciudad>Oslo</ciudad>
        <ciudad pais="">Sevilla</ciudad>
        <ciudad pais="3">Tokio</ciudad>
        <ciudad pais="2">Viena</ciudad>
    </ciudades>
</geografia>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE articulos [
    <!ELEMENT articulos (articulo)+>
    <!ELEMENT articulo (#PCDATA)>
        <!ATTLIST articulo color CDATA #FIXED "rojo">
]>

<articulos>
    <articulo nombre="bolígrafo" color="rojo"/>
    <articulo nombre="cuaderno"/>
    <articulo nombre="rotulador" color="amarillo"/>
</articulos>
```

AC2.8: Ejercicios que debes corregir modificando la DTD. Comprobar que ya son documentos válidos usando Visual Studio Code, XML Copy Editor o Netbeans.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ciudades [
    <!ELEMENT ciudades (ciudad)*>
    <!ELEMENT ciudad (nombre, capital)>
    <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT capital EMPTY>
]>

<ciudades>
    <ciudad>
        <nombre>Buenos Aires</nombre>
        <capital/>
    </ciudad>
    <ciudad>
        <nombre>Liverpool</nombre>
    </ciudad>
    <ciudad>
        <nombre>Osaka</nombre>
    </ciudad>
    <ciudad>
        <nombre>Oslo</nombre>
        <capital/>
    </ciudad>
</ciudades>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE socios [
    <!ELEMENT socios (socio)*>
    <!ELEMENT socio ((numero, clave)+, nombre, edad)>
    <!ELEMENT clave (#PCDATA)>
    <!ELEMENT edad (#PCDATA)>
    <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT numero (#PCDATA)>
]>

<socios>
    <socio>
        <nombre>Ana</nombre>
        <edad>21</edad>
    </socio>
    <socio>
        <numero>598</numero>
        <nombre>Iker</nombre>
        <edad>19</edad>
    </socio>
    <socio>
        <clave>hY75Du</clave>
        <nombre>Iker</nombre>
    </socio>
</socios>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE saludos [
    <!ELEMENT saludos (saludo*)>
    <!ELEMENT saludo (#PCDATA)>
    <!ATTLIST saludo idioma (ALE | FRA | ITA) "ESP">
]>

<saludos>
    <saludo idioma="ALE">Hallo</saludo>
    <saludo>Hola</saludo>
    <saludo idioma="ITA">Ciao</saludo>
    <saludo idioma="FRA">Salut</saludo>
    <saludo idioma="POR">Olá</saludo>
</saludos>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cine [
    <!ELEMENT peliculas (pelicula)*>
    <!ELEMENT pelicula (#PCDATA)>
    <!ATTLIST pelicula codpel ID (P1 | P2 | P3 | P4) "P0">
    <!ELEMENT actores (actor)*>
    <!ELEMENT actor (#PCDATA)>
    <!ATTLIST actor filmografia IDREF #REQUIRED>
]>

<cine>
    <peliculas>
        <pelicula>Avatar</pelicula>
        <pelicula codpel="P1">Gran Torino</pelicula>
        <pelicula codpel="P2">Invictus</pelicula>
        <pelicula codpel="P3">Million dollar baby</pelicula>
        <pelicula codpel="P4">Oblivion</pelicula>
        <pelicula codpel="P5">Unforgiven</pelicula>
    </peliculas>
    <actores>
        <actor filmografia="P1 P3 P5">Clint Eastwood</actor>
        <actor filmografia="P2 P3 P4 P5">Morgan Freeman</actor>
        <actor filmografia="P4">Tom Cruise</actor>
    </actores>
</cine>
```

AC2.9: Se quiere definir un documento XML para representar los resultados de una liga de fútbol. La información que se quiere almacenar de cada partido es:

- El nombre del equipo local
- El nombre del equipo visitante
- Los goles marcados por el equipo local
- Los goles marcados por el equipo visitante

Escribe tres documentos que incluyan los siguientes resultados:

- Nottingham Presa: 0 - Inter de Mitente: 1
- Vodka Juniors: 3 - Sparta da Risa: 3
- Water de Munich: 4 - Esteaua es del grifo: 2
- Aston Birra: 3 - Nottinghan Miedo: 2
- Esfinter de Milan: 2 - Maccabi de Levantar: 2

Cada documento incluirá un DTD diferente para representar ese documento XML:

- Un DTD en la que no haya atributos, sino únicamente etiquetas
- Un DTD en la que los goles sean atributos
- Un DTD en la que toda la información se guarde en forma de atributos

1.3.4. Declaración de una Entidad - ENTITY

Las entidades se usan para definir atajos para caracteres especiales en documentos XML. **Asocia un nombre con un fragmento de contenido.** Una entidad es similar a una macro.

Ventajas:

- Facilita la escritura y lectura.
- Disminuye los posibles errores.
- Clarifica la estructura.
- Facilita el mantenimiento.

Sintaxis de declaración de entidades, las entidades se pueden declarar de forma interna o de forma externa. Sus sintaxis son:

- **Entidad Interna**, son las que se declaran dentro del mismo DTD.

Se declara: `<!ENTITY entidad "valor">`
Se usa: `&entidad;`

Ejemplo:

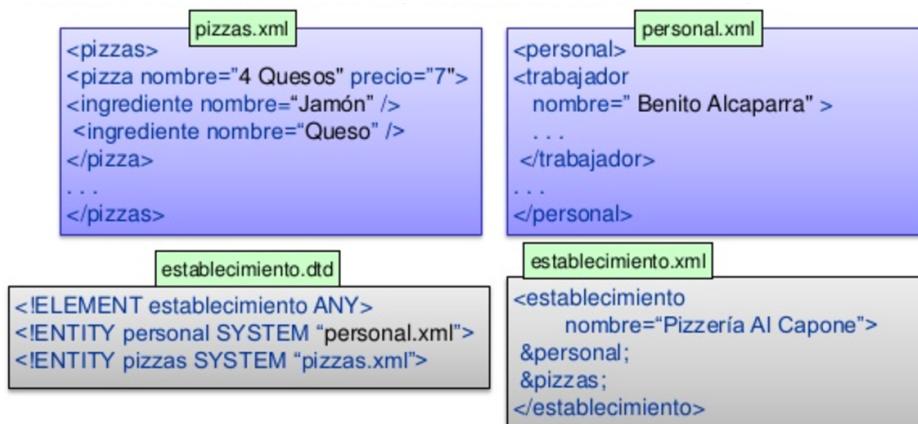
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE address [
<!ELEMENT address (#PCDATA)>
<!ENTITY name "Tanmay patil">
<!ENTITY company "TutorialsPoint">
<!ENTITY phone_no "(011) 123-4567">
]>
<address>
  &name;
  &company;
  &phone_no;
</address>
```

- **Entidad Externa**, son las entidades declaradas fuera del DTD. Se puede hacer referencia a ellas usando identificadores de sistema (SYSTEM) o bien identificadores públicos (PUBLIC).

Se declara: `<!ENTITY entidad SYSTEM "fichero.xml">`
 `<!ENTITY entidad PUBLIC "//BeginningXML//fichero.xml">`

Se usa: `&entidad;`

Ejemplo: Las entidades externas nos permiten **usar archivos externos para incluir otros documentos XML:**



Categorización, pueden ser de 4 tipos:

- Entidades **integradas**.
- Entidades de **carácter**.
- Entidades de **parámetro**.

- **Entidades integradas**, son las entidades con las que todo analizador XML debe de ser compatible.

Se pueden usar en cualquier. Existen 5 entidades integradas que son:

- Ampersand(&): &
- Single quote(comillas simples): '
- Mayor que: >
- Menor que: <
- Double quote(comillas dobles): "

- **Entidades de carácter**, se usan para nombrar alguna de las entidades que son representaciones simbólicas de información. Son caracteres difíciles o imposibles de escribir y que se pueden sustituir por entidades de carácter. Por ejemplo: @, #, \$, ©, ...

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE author[
  <!ELEMENT author (#PCDATA)>
  <!ENTITY writer "Tanmay patil">
  <!ENTITY copyright "&#169;">
]>
<author>&writer;&copyright;</author>
```

- **Entidades paramétricas**, asocian un nombre con un fragmento de DTD que se repite varias veces, pudiendo crear secciones de texto reutilizables.

Se declara: <!ENTITY % **nombre** contenido>

Se usa: %**nombre**;

Presenta **2 diferencias** con respecto a las generales:

- Se usa el % en lugar del &.
- **Solo pueden aparecer en el DTD, NO en el documento XML.**

Ejemplo:

```
<!ELEMENT residence (name, street, pincode, city, phone)>
<!ELEMENT apartment (name, street, pincode, city, phone)>
<!ELEMENT office (name, street, pincode, city, phone)>
<!ELEMENT shop (name, street, pincode, city, phone)>
```

Partiendo del DTD anterior, si ahora quisieramos modificarlo para añadir un elemento más por ejemplo country, tenemos que añadirlos a todas las declaraciones. Otra forma sería usando entidades paramédicas y nos quedaría:

```
<!ENTITY % area "name, street, pincode, city">
<!ENTITY % contact "phone">
```

```
<!ELEMENT residence (%area;, %contact;)>
<!ELEMENT apartment (%area;, %contact;)>
<!ELEMENT office (%area;, %contact;)>
<!ELEMENT shop (%area;, %contact;)>
```

Ejemplo:

```
<!ENTITY % direccion "via, numero?, ciudad, codigoPostal?, pais?">

<!ELEMENT establecimiento (direccion, personal, pizzas)>
|   <!ATTLIST establecimiento nombre CDATA #REQUIRED>

<!ELEMENT direccion (%direccion;)>
<!ELEMENT via (#PCDATA)>
|   <!ATTLIST via tipo CDATA #REQUIRED>
<!ELEMENT numero (#PCDATA)>
<!ELEMENT ciudad (#PCDATA)>
<!ELEMENT codigoPostal (#PCDATA)>
<!ELEMENT pais (#PCDATA)>

<!ELEMENT personal (trabajador+)>
<!ELEMENT trabajador ([nombre, direccion])>
<!ELEMENT nombre (#PCDATA)>

<!ELEMENT pizzas (pizza+)>
<!ELEMENT pizza EMPTY>
|   <!ATTLIST pizza nombre CDATA #REQUIRED>
```

```
<establecimiento nombre="Tres Cantos">
    <direccion>
        <via tipo="Calle">Los Barrancos</via>
        <ciudad>La Laguna</ciudad>
    </direccion>
    <personal>
        <trabajador>
            <nombre>Ana Ramos</nombre>
            <direccion>
                <via tipo="Calle">Los Barrancos</via>
                <ciudad>La Laguna</ciudad>
                <codigoPostal>38000</codigoPostal>
            </direccion>
        </trabajador>
        <trabajador>
            <nombre>Juan Soria</nombre>
            <direccion>
                <via tipo="Calle">Los Barrancos</via>
                <ciudad>La Laguna</ciudad>
            </direccion>
        </trabajador>
```

2. Enlaces de interés

- Validador OnLine: <http://www.xmlvalidation.com/>
- Información DTD: <https://www.mclibre.org/consultar/xml/lecciones/xml-dtd.html>
- Información DTD: <https://www.abrirllave.com/dtd/>
- Más información sobre los DTDs - <https://www.tutorialspoint.com/es/dtd/index.htm>