

U.T. III.- Definición del Tipo de Documento.

Índice.

[1. Documentos XML válidos](#)

[2. Documentos de validación](#)

[3. Referencia a una DTD en un documento XML](#)

[4. Declaraciones](#)

4.1. Declaración de entidades

[4.2. Declaración de elementos](#)

[4.3- Declaración de atributos](#)

1.- Documentos XML válidos.

- Como ya hemos visto, que un documento XML esté *bien formado* asegura que las reglas XML se cumplen y que no existe ninguna incoherencia en el uso del lenguaje. Sin embargo, esto no es suficiente para asegurar la homogeneidad entre documentos XML que almacenen el mismo conjunto de datos con el mismo objetivo.
- Esto provoca problemas para su tratamiento automatizado, tanto a la hora de almacenarlos en una BD, relacional o XML nativa, como a la hora de transmitirlos, tanto el equipo emisor como el receptor tendrán que conocer no sólo qué datos van a compartir, sino cómo van a estar éstos empaquetados.

Ejemplo 301.

En el ejercicio 206 vimos varias posibilidades de almacenar en un XML la información contenida en una factura.

Los documentos estaban bien formados y almacenaban el contenido de la factura respetando su estructura, pero eran heterogéneos, lo que dificulta su almacenamiento, consulta y transmisión automatizadas.

- Se hace pues necesaria la creación de unas reglas (una gramática) que aseguren que todos los documentos XML que contengan un determinado conjunto de información sean homogéneos, es decir, tengan los mismos elementos y atributos, con el mismo nombre y la misma estructura.
- Se dice en este caso que el documento XML es "**válido**" con respecto a la gramática para él definida.
- Durante el proceso de validación se comprueba:
 - o Qué elementos o atributos se permiten en un documento del tipo definido en el esquema.
 - o La estructura de los elementos y atributos (elementos anidados, atributos obligatorios u opcionales, etc.)
 - o El orden de los elementos.
 - o Los valores de los datos de atributos y elementos (según enumeraciones, rangos de valores delimitados, tipo de dato correcto (formato de fecha, números enteros, etc.)
 - o La unicidad de valores dentro de un documento.
 - Por ejemplo, referencias de productos que no pueden repetirse.

[Volver](#)

2.- Documentos de validación.

- Las técnicas más populares para validar documentos XML son :
- **DTD (Document Type Definition)**. Es el modelo más antiguo, heredado del SGML, es el más popular y sencillo, aunque sus detractores critican que su sintaxis no sea XML.
 - o Una **definición de tipo de documento** o **DTD** define la estructura válida de un documento XML: los elementos, atributos, entidades, notaciones, etc., que pueden aparecer, el orden y el

número de veces que pueden aparecer, cuáles pueden ser hijos de cuáles, etc.

- o El parser XML verifica si un documento XML es válido, es decir, si cumple las reglas especificadas en su DTD asociado.
- **XML Schema.** Es el nombre oficial otorgado a la recomendación del W3C, que elaboró el primer lenguaje de esquema externo al documento XML, dado que la DTD forma parte de él.
 - o XML Schema permite más capacidades de validación y es coherente con el lenguaje XML, dado que en sí mismo es un documento XML.
 - o Se utiliza junto con XQuery (lenguaje de consulta de BBDD XML nativas) y fundamentalmente en los servicios Web.
 - o La principal desventaja es su complejidad y que plantea problemas de compatibilidad con otros softwares.
- **Relax NG.** Es un modelo creado por OASIS, basado también en la sintaxis XML, muy intuitivo y más fácil de entender que el XML Schema.
 - o Es el menos difundido de los tres.

[Volver](#)

3.- Referencia a una DTD en un documento XML.

- Se puede definir la estructura que debe cumplir un documento XML mediante código DTD insertado en el propio documento, mediante un enlace a un documento-plantilla externo o ambas cosas.
- En cualquier caso la declaración se lleva a cabo mediante la instrucción de procesamiento **<!DOCTYPE>**
 - o El nombre del documento XML que contenga o referencie una DTD debe coincidir con el de su elemento raíz.

- La declaración de una **DTD interna** debe seguir la siguiente sintaxis:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE elemento_raíz [
    declaraciones
]>
<elemento_raíz>
    ...
</elemento_raíz>
```

Ejemplo 302.

Fichero **nota.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nota[
    <!ELEMENT nota (para,de,cabecera,cuerpo)>
    <!ELEMENT para (#PCDATA)>
    <!ELEMENT de (#PCDATA)>
    <!ELEMENT cabecera (#PCDATA)>
    <!ELEMENT cuerpo (#PCDATA)>
]>
<nota>
    <para>José</para>
    <de>Juan</de>
    <cabecera>Recordatorio</cabecera>
    <cuerpo>Reunión el lunes a primera hora</cuerpo>
</nota>
```

o La desventaja de utilizar una DTD interna es que esa definición sólo sirve para el documento en el que se encuentra, por lo que realmente no define *tipos* de documentos.

- La declaración de una **DTD externa** debe seguir la siguiente sintaxis:

```
<!DOCTYPE elemento_raíz SYSTEM "/url/nombrefichero.dtd">
```

o De este modo crearemos un documento-plantilla externo que puede ser referenciado por cualquier documento XML que deba ajustarse a sus reglas, es decir, que pertenezca a un determinado *tipo* de documentos.

Ejemplo 303.

Fichero **nota.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nota SYSTEM "nota.dtd">
<nota>
    <para>José</para>
    <de>Juan</de>
    <cabecera>Recordatorio</cabecera>
    <cuerpo>Reunión mañana a primera hora</cuerpo>
</nota>
```

Fichero **nota.dtd**:

```
<!ELEMENT nota (para,de,cabecera,cuerpo)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT cabecera (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
```

- la palabra **SYSTEM** indica que el documento DTD es de uso privado, si queremos que sea de uso público debemos utilizar **PUBLIC**:

```
<!DOCTYPE elemento_raíz PUBLIC fpi "/url/fichero.dtd">
```

- o Donde **fpi (Identificador Público Formal)** es el nombre de un repositorio de DTDs públicos. En el caso de no encontrar la DTD en dicho repositorio el parser la buscaría en la dirección especificada.
- La declaración de una **DTD interna** y una **externa** debe seguir la siguiente sintaxis:

```
<!DOCTYPE elemento_raíz SYSTEM "/path/fichero.dtd" [declaraciones]>
```

- o El parser da prioridad a la DTD interna, que nos permite redefinir partes de la DTD externa.

[Volver](#)

4.- Declaraciones.

- Las DTDs describen la estructura de los documentos XML mediante declaraciones.
- Los principales tipos de declaraciones son:
 - o De entidades.
 - o De elementos.
 - o De atributos.

4.1.- Declaración de entidades.

- Una **entidad** consiste en un nombre y su valor, son similares a las constantes en los lenguajes de programación. Se declaran y utilizan igual que en un documento XML.

[Volver](#)

4.2.- Declaración de elementos.

- Una declaración de elementos indica los elementos permitidos en un documento y su contenido (que puede ser texto, vacío u otros elementos).
- Sigue la siguiente sintaxis:

<!ELEMENT nombreelemento contenido>

- o Donde **contenido** es una expresión que describe el contenido del elemento.
- Para definir el contenido de un elemento se pueden utilizar los términos **EMPTY**, **(#PCDATA)**, **ANY** o expresiones más complejas.
 - o **EMPTY**. Indica que se trata de un elemento vacío, es decir, no puede tener contenido.

Declaración DTD:

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo EMPTY>  

```

Definiciones válidas en el documento XML:

- a) **<ejemplo></ejemplo>**
- b) **<ejemplo/>**

Definiciones **no** válidas en el documento XML:

- c) **<ejemplo>Esto es un ejemplo</ejemplo>**
- d) **<ejemplo><titulo></titulo></ejemplo>**

- o **(#PCDATA)** Indica que el elemento puede contener texto o estar vacío.

Declaración DTD:

```
<DOCTYPE ejemplo [  
  <!ELEMENT ejemplo (#PCDATA)>
```

Definiciones válidas en el documento XML:

- a) **<ejemplo/>**
- b) **<ejemplo>Esto es un ejemplo</ejemplo>**

Definiciones **no** válidas en el documento XML:

- c) **<ejemplo><titulo></titulo></ejemplo>**

- o **ANY** Indica que el elemento puede estar vacío, contener texto u otros elementos.

Declaración DTD:

```
<!DOCTYPE ejemplo [  
    <!ELEMENT ejemplo ANY>  
    <!ELEMENT titulo EMPTY>
```

Definiciones válidas en el documento XML:

- a) `<ejemplo/>`
- b) `<ejemplo>Esto es un ejemplo</ejemplo>`
- c) `<ejemplo><titulo></titulo></ejemplo>`

- Para especificar los elementos que puede o debe contener un elemento se utilizan los siguientes conectores y modificadores:

- o **,** La coma indica que el elemento contiene los elementos que se indican y en el mismo orden.

Declaración DTD:

```
<!DOCTYPE ejemplo [  
    <!ELEMENT ejemplo (a,b)>  
        <!ELEMENT a EMPTY>  
        <!ELEMENT b EMPTY>  
>
```

Definiciones válidas en el documento XML:

- a) `<ejemplo><a/></ejemplo>`

Definiciones **no** válidas en el documento XML:

- a) `<ejemplo><a/></ejemplo>`
- b) `<ejemplo><a/><c/></ejemplo>`
- c) `<ejemplo><a/></ejemplo>`

- o **|** La ó exclusiva indica que el elemento contiene uno sólo de los dos elementos implicados.

Declaración DTD:

```
<!DOCTYPE ejemplo [  
    <!ELEMENT ejemplo (a|b)>  
        <!ELEMENT a EMPTY>  
        <!ELEMENT b EMPTY>  
>
```

Definiciones válidas en el documento XML:

- a) `<ejemplo><a/></ejemplo>`
- b) `<ejemplo></ejemplo>`

Definiciones **no** válidas en el documento XML:

- c) `<ejemplo><a/></ejemplo>`
- d) `<ejemplo></ejemplo>`

- o **?** La interrogación indica que el elemento puede estar 0 ó 1 vez.

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo (a|b?)>  
    <!ELEMENT a EMPTY>  
    <!ELEMENT b EMPTY>  
>
```

Definiciones válidas en el documento XML:

- a) `<ejemplo><a/></ejemplo>`
- b) `<ejemplo></ejemplo>`
- c) `<ejemplo></ejemplo>`

Definiciones **no** válidas en el documento XML:

- d) `<ejemplo></ejemplo>`

- o ***** El asterisco Indica que el elemento que le precede puede estar 0 ó más veces.

```
<DOCTYPE ejemplo [  
  <!ELEMENT ejemplo (a*,b)>  
    <!ELEMENT a EMPTY>  
    <!ELEMENT b EMPTY>  
>
```

Definiciones válidas en el documento XML:

- a) `<ejemplo></ejemplo>`
- b) `<ejemplo><a/></ejemplo>`
- c) `<ejemplo><a/><a/></ejemplo>`

Definiciones **no** válidas en el documento XML:

- d) `<ejemplo></a/></ejemplo>`
- e) `<ejemplo><a/></ejemplo>`

- o **+** Indica que el elemento puede estar 1 ó más veces.

```
<DOCTYPE ejemplo [  
  <!ELEMENT ejemplo (a+,b)>  
    <!ELEMENT a EMPTY>  
    <!ELEMENT b EMPTY>  
>
```


Definiciones válidas en el documento XML:

- a) `<ejemplo><a></ejemplo>`
- b) `<ejemplo><a><a></ejemplo>`

Definiciones **no** válidas en el documento XML:

- c) `<ejemplo></ejemplo>`
- d) `<ejemplo></ejemplo>`
- e) `<ejemplo><a></ejemplo>`

- o **()** Permite agrupar expresiones.

```
<DOCTYPE ejemplo [  
  <!ELEMENT ejemplo (a,(a|b))>  
    <!ELEMENT a EMPTY>  
    <!ELEMENT b EMPTY>  
>
```

Definiciones válidas en el documento XML:

- a) `<ejemplo><a><a></ejemplo>`
- b) `<ejemplo><a></ejemplo>`

Definiciones **no** válidas en el documento XML:

- c) `<ejemplo></ejemplo>`
- d) `<ejemplo></ejemplo>`
- e) `<ejemplo><a></ejemplo>`

Ejercicio 301.

Los siguientes documentos XML no son válidos, modificarlos, lo mínimo posible, para que lo sean. La DTD no debe modificarse.

numeros.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE numeros [  
  <!ELEMENT numeros (#PCDATA)>  
>  
  
<numeros>  
  <numero>25</numero>  
</numeros>
```

letras.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE letras [  
  <!ELEMENT letras (letra)>  
  <!ELEMENT letra (#PCDATA)>  
>  
  
<letras>
```

```
<letra>m</letra>
<letra>uve doble</letra>
</letras>
```

colores.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE colores [
  <!ELEMENT colores (color*)>
  <!ELEMENT color (#PCDATA)>
]>

<colores>
  <color>azul marino</color>
  negro
  <color>amarillo</color>
</colores>
```

flores.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE flores [
  <!ELEMENT flores (flor+)>
  <!ELEMENT flor (#PCDATA)>
]>

<flores>
</flores>
```

animales.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE animales [
  <!ELEMENT animales (animal*)>
  <!ELEMENT animal (#PCDATA)>
]>

<animales>
  <perro>Caniche</perro>
  <gato>Siamés</gato>
</animales>
```

escritores.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE escritores [
  <!ELEMENT escritores (escritor*)>
  <!ELEMENT escritor (nombre, nacimiento)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT nacimiento (#PCDATA)>
]>
```

```

<escritores>
  <escritor>
    <nombre>Mario Vargas Llosa</nombre>
    <nacimiento>28 de marzo de 1936</nacimiento>
  </escritor>
  <escritor>
    <nacimiento>1 de abril de 1929</nacimiento>
    <nombre>Milan Kundera</nombre>
  </escritor>
</escritores>

```

musicos.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE musicos [
  <!ELEMENT musicos (musico*)>
  <!ELEMENT musico ((nombre | apodo), fechaNacimiento)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT apodo (#PCDATA)>
  <!ELEMENT fechaNacimiento (#PCDATA)>
]>

<musicos>
  <musico>
    <nombre>Antonio Vivaldi</nombre>
    <apodo>El cura pelirrojo</apodo>
    <fechaNacimiento>4 de marzo de 1678</fechaNacimiento>
  </musico>
  <musico>
    <nombre>Johann Sebastian Bach</nombre>
    <apodo>El viejo peluca</apodo>
    <fechaNacimiento>21 de marzo de 1685</fechaNacimiento>
  </musico>
</musicos>

```

agenda.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE agenda [
  <!ELEMENT agenda (contacto*)>
  <!ELEMENT contacto (nombre, telefonoFijo*, telefonoMovil+)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT telefonoFijo (#PCDATA)>
  <!ELEMENT telefonoMovil (#PCDATA)>
]>

<agenda>

```

```

<contacto>
  <nombre>Ayuntamiento</nombre>
  <telefonoFijo>010</telefonoFijo>
</contacto>
<contacto>
  <nombre>Emergencias</nombre>
  <telefonoFijo>112 (Unión Europea)</telefonoFijo>
  <telefonoMovil>Desconocido</telefonoMovil>
  <telefonoFijo>911 (Estados Unidos)</telefonoFijo>
</contacto>
</agenda>

```

sistemaSolar.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sistemaSolar [
  <!ELEMENT sistemaSolar (cuerpo*)>
  <!ELEMENT cuerpo (planeta|satelite|asteroide)>
  <!ELEMENT planeta (#PCDATA)>
  <!ELEMENT satelite (#PCDATA)>
  <!ELEMENT asteroide (#PCDATA)>
]>

<sistemaSolar>
  <cuerpo>
    <planeta>Tierra</planeta>
    <satelite>Luna</satelite>
  </cuerpo>
  <asteroide>Ceres</asteroide>
</sistemaSolar>

```

Ejercicio 302.

Con los mismos documentos XML del ejercicio 301, modificar ahora sólo su DTD interna, lo mínimo posible, para hacerlos válidos. (excepto colores.xml)

[Volver](#)

4.3- Declaración de atributos.

- Una declaración de atributos indica los atributos permitidos en cada elemento y su tipo o valores permitidos.
- Una declaración de atributos sigue la siguiente sintaxis:

<!ATTLIST nomElemento nomAtributo tipoAtributo valorAtributo>

- o **nomElemento** Nombre del elemento para el que se define el atributo.
 - o **nomAtributo** Nombre del atributo
 - o **tipoAtributo** Tipo de datos de los valores que se pueden dar al atributo.
 - o **valorAtributo** Es el valor predeterminado del atributo, aunque también puede indicar otras cosas.
 - o Todos los campos son obligatorios.
- Para definir varios atributos para un mismo elemento se pueden utilizar una o varias declaraciones de atributos.

La siguiente declaración

```
<!ATTLIST nombreElemento
    nombreAtributo1 tipoAtributo1 valorAtributo1
    nombreAtributo2 tipoAtributo2 valorAtributo2
>
```

sería equivalente a

```
<!ATTLIST nombreElemento
    nombreAtributo1 tipoAtributo1 valorAtributo1
>
<!ATTLIST nombreElemento
    nombreAtributo2 tipoAtributo2 valorAtributo2
>
```

- **tipoAtributo** puede ser:
 - o **CDATA** El atributo puede contener cualquier secuencia de caracteres o estar vacío.

```
<!DOCTYPE ejemplo [
    <!ELEMENT ejemplo EMPTY >
    <!ATTLIST ejemplo color CDATA #REQUIRED >
]>
```

Definiciones válidas en el documento XML:

- a) `<ejemplo color=""></ejemplo>`
- b) `<ejemplo color="rojo"></ejemplo>`
- c) `<ejemplo color="azul marino #000080"></ejemplo>`

Definiciones **no** válidas en el documento XML

- e) `<ejemplo/>`
- f) `<ejemplo alto="10px"></ejemplo>`

- o **NMTOKEN** El atributo no puede estar vacío y sólo puede contener letras dígitos y los caracteres `.` (punto) `-` (guión) `_` (subrayado) y `:` (dos puntos)

```
<!DOCTYPE ejemplo [  
    <!ELEMENT ejemplo EMPTY>  
    <!ATTLIST ejemplo color NMTOKEN #REQUIRED >  

```

Definiciones válidas en el documento XML:

- a) `<ejemplo color="rojo"></ejemplo>`

Definiciones **no** válidas en el documento XML

- b) `<ejemplo color=""></ejemplo>`
- c) `<ejemplo color="azul marino"></ejemplo>`
- d) `<ejemplo alto="10px"></ejemplo>`

- o **NMTOKENS** El atributo puede contener los mismos caracteres que en el caso anterior más espacios en blanco.

```
<!DOCTYPE ejemplo [  
    <!ELEMENT ejemplo EMPTY>  
    <!ATTLIST ejemplo color NMTOKENS #REQUIRED >  

```

Definiciones válidas en el documento XML:

- a) `<ejemplo color="rojo"></ejemplo>`
- b) `<ejemplo color="azul marino"></ejemplo>`

Definiciones **no** válidas en el documento XML

- c) `<ejemplo color=""></ejemplo>`
- d) `<ejemplo color="azul marino #000080"></ejemplo>`

- o **valores** El atributo debe contener uno de los términos indicados en una lista.

- La lista debe ir entre paréntesis, con los términos separados por la barra vertical |

```
<!DOCTYPE ejemplo [
  <!ELEMENT ejemplo EMPTY>
  <!ATTLIST ejemplo color (azul|blanco|rojo) #REQUIRED>
]>
```

Definiciones válidas en el documento XML:

a) `<ejemplo color="rojo"></ejemplo>`

Definiciones **no** válidas en el documento XML

b) `<ejemplo color=""></ejemplo>`

c) `<ejemplo color="verde"></ejemplo>`

- o **ID** El valor del atributo debe ser único, por lo que no podrá repetirse en otros elementos o atributos.

- El valor de un atributo de tipo **ID** no puede comenzar por un número ni tener espacios en blanco.

```
<!DOCTYPE ejemplo [
  <!ELEMENT ejemplo (libro*)>
  <!ELEMENT libro (#PCDATA)>
  <!ATTLIST libro codigo ID #REQUIRED >
]>
```

Definiciones válidas en el documento XML:

a) `<ejemplo>`

`<libro codigo="L1">El Poema del Mío Cid</libro>`

`<libro codigo="L2">La Odisea</libro>`

`</ejemplo>`

Definiciones **no** válidas en el documento XML

b) `<ejemplo>`

`<libro codigo="1">El Poema del Mío Cid</libro>`

`<libro codigo="L2">La Odisea</libro>`

`</ejemplo>`

c) `<ejemplo>`

`<libro codigo="L1">Poema del Mío Cid</libro>`

`<libro codigo="L1">La Odisea</libro>`

`</ejemplo>`

- o **IDREF** Su valor debe coincidir con el valor del atributo **ID** de otro elemento.

```
<!DOCTYPE ejemplo [
  <!ELEMENT ejemplo ((libro|prestamo)*)>
  <!ELEMENT libro (#PCDATA)>
    <!ATTLIST libro codigo ID #REQUIRED>
  <!ELEMENT prestamo (#PCDATA)>
    <!ATTLIST prestamo codlibro IDREF #REQUIRED>
]>
```

Definiciones válidas en el documento XML:

a) `<ejemplo>`
`<libro codigo="L1">El Poema del Mío Cid</libro>`
`<prestamo codlibro="L1">Antonio Pérez</prestamo>`
`</ejemplo>`

Definiciones **no** válidas en el documento XML

b) `<ejemplo>`
`<libro codigo="L1">El Poema del Mío Cid</libro>`
`<prestamo codlibro="L2">La Odisea</prestamo>`
`</ejemplo>`

Consideraciones:

- o Un mismo elemento no puede tener 2 atributos de tipo **ID**.
- o Varios elementos pueden tener atributos de tipo **IDREF** que hagan referencia a un mismo atributo **ID**.

```
<!DOCTYPE ejemplo [
  <!ELEMENT ejemplo ((libro|prestamo|venta)*)>
  <!ELEMENT libro (#PCDATA)>
  <!ELEMENT prestamo (#PCDATA)>
  <!ELEMENT venta (#PCDATA)>
    <!ATTLIST libro codigo ID #REQUIRED>
    <!ATTLIST prestamo codlibro IDREF #REQUIRED>
    <!ATTLIST venta codlibro IDREF #REQUIRED>
]>
```

```
<ejemplo>
  <libro codigo="L1">El Poema del Mío Cid</libro>
  <prestamo codlibro="L1">Antonio Pérez</prestamo>
  <venta codlibro="L1">Juan Fernández</venta>
</ejemplo>
```

- o Un mismo elemento puede tener 2 o más atributos de tipo **IDREF**, cada uno de ellos hará referencia a un atributo **ID** de un elemento distinto.

```
<!DOCTYPE ejemplo [
```



```

<!ELEMENT ejemplo ((libro|revista|prestamo)*)>
<!ELEMENT libro (#PCDATA)>
<!ELEMENT revista (#PCDATA)>
<!ELEMENT prestamo (#PCDATA)>
    <!ATTLIST libro codigo ID #REQUIRED>
    <!ATTLIST revista codigo ID #REQUIRED>
    <!ATTLIST prestamo codlibro IDREF #REQUIRED>
    <!ATTLIST prestamo codrevista IDREF #REQUIRED>
]>

```

```

<ejemplo>
    <libro codigo="L1">El Poema del Mío Cid</libro>
    <revista codigo="S1">Historia de la Ciencia</revista>
    <prestamo codlibro="L1" codrevista="S1">Antonio
Pérez</prestamo>
</ejemplo>

```

- o **IDREFS** el valor del atributo está formado por una serie de valores separados por espacios que coinciden con el valor del atributo **ID** de otros elementos.

```

<!DOCTYPE ejemplo [
    <!ELEMENT ejemplo ((libro|prestamo)*)>
    <!ELEMENT libro (#PCDATA)>
        <!ATTLIST libro codigo ID #REQUIRED>
    <!ELEMENT prestamo (#PCDATA)>
        <!ATTLIST prestamo libro IDREFS #REQUIRED>
]>

```

Definiciones válidas en el documento XML:

```

a) <ejemplo>
    <libro codigo="L1">Poema del Mío Cid</libro>
    <libro codigo="L2">La Odisea</libro>
    <prestamo libro="L1 L2">Antonio Pérez</prestamo>
</ejemplo>

```

```

b) <ejemplo>
    <libro codigo="L1">Poema del Mío Cid</libro>
    <libro codigo="L2">La Odisea</libro>
    <prestamo libro="L1">Antonio Pérez</prestamo>
</ejemplo>

```

```

c) <ejemplo>
    <libro codigo="L1">Poema del Mío Cid</libro>
    <libro codigo="L2">La Odisea</libro>
    <prestamo libro="L2">Antonio Pérez</prestamo>
</ejemplo>

```

Definiciones **no** válidas en el documento XML

d) `<ejemplo>`

`<libro codigo="L1">Poema del Mío Cid</libro>`

`<libro codigo="L2">La Odisea</libro>`

`<prestamo libro="L3">Antonio Pérez</prestamo>`

`</ejemplo>`

- **valorAtributo** puede ser:

- o **#REQUIRED**: el atributo es obligatorio.

```
<!DOCTYPE ejemplo [  
  <ELEMENT ejemplo EMPTY>  
  <!ATTLIST ejemplo color CDATA #REQUIRED>  

```

Definiciones válidas en el documento XML:

a) `<ejemplo color=""></ejemplo>`

b) `<ejemplo color="amarillo"></ejemplo>`

c) `<ejemplo color="azul marino #000080"></ejemplo>`

Definiciones **no** válidas en el documento XML

d) `<ejemplo></ejemplo>`

- o **#IMPLIED**: el atributo no es obligatorio.

```
<!DOCTYPE ejemplo [  
  <ELEMENT ejemplo EMPTY>  
  <!ATTLIST ejemplo color CDATA #IMPLIED>  

```

Definiciones válidas en el documento XML:

a) `<ejemplo></ejemplo>`

b) `<ejemplo color=""></ejemplo>`

c) `<ejemplo color="amarillo"></ejemplo>`

d) `<ejemplo color="azul marino #000080"></ejemplo>`

Definiciones **no** válidas en el documento XML

e) `<ejemplo alto="10px"></ejemplo>`

- o **#FIXED valor**: el atributo tiene el valor especificado y no es obligatorio.

```
<!DOCTYPE ejemplo [  
  <ELEMENT ejemplo EMPTY>  
  <!ATTLIST ejemplo color CDATA #FIXED "verde">  

```

Definiciones válidas en el documento XML:

- a) `<ejemplo></ejemplo>`
- b) `<ejemplo color="verde"></ejemplo>`

Definiciones **no** válidas en el documento XML

- c) `<ejemplo color=""></ejemplo>`
- d) `<ejemplo color="amarillo"></ejemplo>`

- o **"valor"**: el atributo no es obligatorio y puede cambiar el valor predeterminado.

- De esta manera, si el atributo no se especifica su valor no quedaría indeterminado como en el caso de **#IMPLIED**

```
<!DOCTYPE ejemplo [  
    <!ELEMENT ejemplo EMPTY>  
    <!ATTLIST ejemplo color CDATA "verde">  
>
```

Definiciones válidas en el documento XML:

- a) `<ejemplo></ejemplo>`
- b) `<ejemplo color=""></ejemplo>`
- c) `<ejemplo color="verde"></ejemplo>`
- d) `<ejemplo color="amarillo"></ejemplo>`

Ejercicio 303.

Los siguientes documentos XML no son válidos, modificarlos, lo mínimo posible, para que lo sean. La DTD no debe modificarse.

persona.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE persona [  
    <!ELEMENT persona EMPTY>  
    <!ATTLIST persona nombre CDATA #IMPLIED>  
>  
  
<persona dni="03141592E" />
```

pelicula.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE pelicula [  
    <!ELEMENT pelicula EMPTY>  
    <!ATTLIST pelicula titulo CDATA #IMPLIED>  
>
```

```
<pelicula titulo="La diligencia" genero="oeste" />
```

cuadros.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cuadros [
  <!ELEMENT cuadros (cuadro*)>
  <!ELEMENT cuadro EMPTY>
  <!ATTLIST cuadro titulo ID #REQUIRED>
  <!ATTLIST cuadro autor CDATA #REQUIRED>
]>

<cuadros>
  <cuadro titulo="Adán y Eva" autor="Alberto Durero" />
  <cuadro autor="Lucas Cranach, el viejo" titulo="Adán y Eva" />
</cuadros>
```

listacompra.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE listacompra [
  <!ELEMENT listacompra (item*)>
  <!ELEMENT item EMPTY>
  <!ATTLIST item nombre CDATA #REQUIRED>
  <!ATTLIST item cantidad CDATA #REQUIRED>
]>

<listacompra>
  <leche cantidad="12 litros" ></leche>
  <pan cantidad="3 barras de cuarto" />
</listacompra>
```

jugadoresfutbol.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jugadoresfutbol [
  <!ELEMENT jugadoresfutbol (jugador*)>
  <!ELEMENT jugador EMPTY>
  <!ATTLIST jugador nombre NMTOKENS #REQUIRED>
  <!ATTLIST jugador codigo ID #REQUIRED>
]>

<jugadoresfutbol>
  <jugador nombre="Alfredo Di Stéfano" codigo="1"/>
  <jugador nombre="Edison Arantes do Nascimento, Pelé" codigo="2" />
  <jugador nombre="Diego Armando Maradona" codigo="3" />
  <jugador nombre="Johan Cruyff" codigo="4" />
</jugadoresfutbol>
```

futbol.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE futbol [
  <!ELEMENT futbol ((jugador|equipo)*)>
  <!ELEMENT jugador EMPTY>
  <!ATTLIST jugador nombre NMTOKENS #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="Maradona" /
  >
  <equipo nombre="Futbol Club Barcelona" jugadores="Cruyff, Maradona" /
  >
</futbol>
```

Ejercicio 304.

Con los mismos documentos XML del ejercicio 303, modificar ahora sólo su DTD interna, lo mínimo posible, para hacerlos válidos.

Ejercicio 305.

Dados los siguientes documentos XML, crear una DTD externa válida para cada uno de ellos. La DTD deberá ajustarse lo máximo posible al contenido actual de los documentos.

a)

```
<?xml version="1.0" encoding="UTF-8"?>
<ies>
  <nombrecentro>IES Ágora</nombrecentro>
  <web>http://www.iesagora.juntaextremadura.net</web>
  <ciclos>
    <ciclo id="ASIR">
      <nombre>Administración de Sistemas Informáticos en Red</nombre>
      <grado>Superior</grado>
      <decretotitulo>2009</decretotitulo>
    </ciclo>
    <ciclo id="DAM">
      <nombre>Desarrollo de Aplicaciones Multiplataforma</nombre>
      <grado>Superior</grado>
      <decretotitulo>2010</decretotitulo>
    </ciclo>
  </ciclos>
</ies>
```

```

<ciclo id="DAW">
  <nombre>Desarrollo de Aplicaciones Web</nombre>
  <grado>Superior</grado>
  <decretotitulo>2010</decretotitulo>
</ciclo>
<ciclo id="SMR">
  <nombre>Sistemas Microinformáticos y Redes</nombre>
  <grado>Medio</grado>
  <decretotitulo>2008</decretotitulo>
</ciclo>
</ciclos>
</ies>

```

b)

```

<biblioteca>
  <libro>
    <titulo>La vida esta en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechapublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechanacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechapublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechanacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechapublicacion año="1969"/>
  </libro>
</biblioteca>

```

c)

```

<?xml version="1.0" encoding="UTF-8"?>
<cd>
  <artista>
    <nombre>Concha Piquer</nombre>
    <fechanaci>08/12/1906</fechanaci>
    <lugarnaci>Valencia</lugarnaci>
  </artista>
  <titulocd>Grandes Éxitos</titulocd>
  <fechacrea>01/01/2014</fechacrea>
  <genero>Canción Española</genero>
  <cancion>
    <titulo>Ojos verdes</titulo>
    <duracion tiempo="minutos">2.5</duracion>
  </cancion>
  <cancion>
    <titulo>Tatuaje</titulo>
    <duracion tiempo="minutos">2.9</duracion>
  </cancion>

```

```

<cancion>
  <titulo>Suspiros de España</titulo>
  <duracion tiempo="minutos">3.2</duracion>
</cancion>
</cd>

```

d)

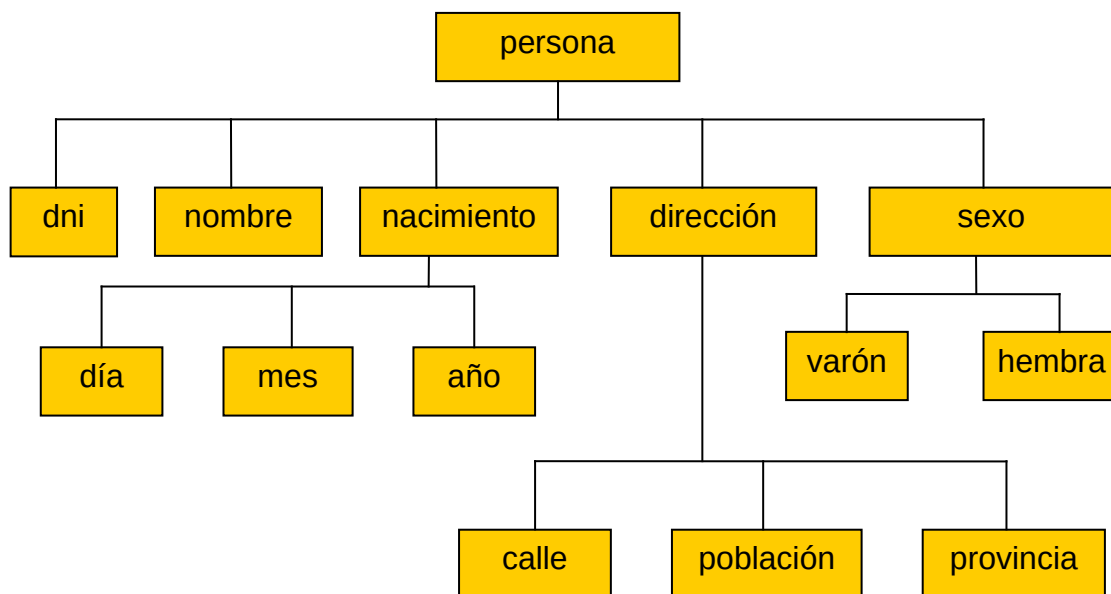
```

<?xml version="1.0" encoding="UTF-8"?>
<hoteles>
  <hotel>
    <nombre>Senator</nombre>
    <direccion>
      <calle>Alcalá</calle>
      <numero>10</numero>
    </direccion>
    <ciudad>Madrid</ciudad>
    <pais>España</pais>
    <estrellas>5</estrellas>
    <descripcion>
      <parrafo>Hotel de lujo</parrafo>
      <parrafo>Situado en el centro de Madrid</parrafo>
    </descripcion>
    <habitaciones>
      <individuales>20</individuales>
      <dobles>30</dobles>
      <triples>15</triples>
      <suties>8</suties>
    </habitaciones>
  </hotel>
  <hotel>
    <nombre>Extremadura</nombre>
    <direccion>
      <calle>Ruta de la Plata</calle>
      <numero>1</numero>
    </direccion>
    <ciudad>Cáceres</ciudad>
    <pais>España</pais>
    <estrellas>4</estrellas>
    <descripcion>
      <parrafo>Hotel funcional</parrafo>
      <parrafo>Situado a la entrada de Cáceres</parrafo>
    </descripcion>
    <habitaciones>
      <individuales>10</individuales>
      <dobles>40</dobles>
      <triples>5</triples>
      <suties>16</suties>
    </habitaciones>
  </hotel>
</hoteles>

```

Ejercicio 306.

Dado el siguiente árbol de nodos:



Crear un documento XML que contenga información sobre dos personas.

Crear también una DTD externa que lo valide.

Teniendo en cuenta que:

1. Todos los nodos son obligatorios excepto nacimiento y sexo.
 2. No podrá haber dos personas con el mismo dni.
 3. Una persona puede tener más de una dirección.
 4. Los nodos varón y hembra no pueden estar a la vez dentro de un mismo nodo sexo y no pueden tener contenido.
 - 5.- Debéis restringir los tipos de contenido lo máximo posible.
- a) El documento XML sólo contendrá como atributo el nodo dni.
- b) El documento XML sólo contendrá los siguientes elementos: persona, nombre, nacimiento, dirección, sexo, varón y hembra.

Ejercicio 307.

Se desea modelar en XML la información de los distintos modelos de coche que vende un concesionario.

En un momento dado la lista de modelos puede estar vacía o contener 1 o varios modelos diferentes.

Cada modelo se identifica mediante una referencia única y puede incluir una descripción del mismo.

Además, para cada modelo se puede incluir una lista de otros modelos con los que está relacionado (los otros modelos deben existir en el ejemplar del documento).

El motor también puede incluir una referencia a algún modelo relacionado.

Construir una DTD que cumpla estos requisitos y crear un ejemplar de documento válido para esa DTD con la siguiente información:

Modelo	Modelos relacionados	Motor	Motor Relacionado con Modelo	Descripción
M556				
M555	M444, M556	1.998cc	M556	Cuatro cilindros, 112 CV. Cambio manual Consumo medio de 8.5l a los 100 Km. Precio de 19.950€
M557				
M154				
M444	M555, M154	2000cc	M556	Cuatro cilindros, 128 CV. Cambio automático. Consumo medio de 9.5l a los 100 Km. Precio de 20.950€

Ejercicio 308.

Crear un documento XML, junto con su DTD, que almacene información sobre alumnos, con las restricciones que se indican a continuación:

- Por cada alumno se almacenará su número de matrícula, que será único, nombre, apellidos, domicilio y teléfono.
 - o Todos los nodos son obligatorios y el alumno puede tener más de un teléfono.
 - o Se desea reflejar, para cada número de teléfono almacenado, si es propio, del padre, de la madre o del tutor. Ningún otro valor será permitido.
- El alumno puede pertenecer opcionalmente a uno de los equipos deportivos del centro.

- Se desea almacenar el origen del alumno, la provincia si es español, o el país si es extranjero.
- Añadir opcionalmente comentarios sobre los alumnos.
 - o Crearlos de manera que puedan contener cualquier información de tipo texto junto con otros elementos.
- Se desea conocer el sexo de los alumnos, hombre o mujer, aunque no será obligatorio que aparezca esta información.
- Almacenar las notas mediante los elementos evaluación y calificación, que aparecerán en ese orden y varias veces.
- Algunos alumnos son ayudados por otro u otros alumnos, por lo que deberá reflejarse en la información referente a un alumno el o los alumnos que le apoyan.

[Volver](#)