

Tema 04 - DTD

Parte 2

9. DTD's

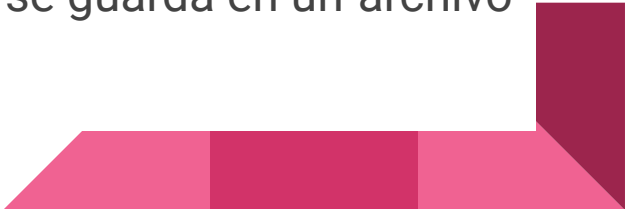
En una empresa tenemos dos programadores que programan así:

<u>Programador 1</u>	<u>Programador 2</u>
<pre><clientes> <cliente> <nombre>AcerSA</nombre> <cif>455321</cif> </cliente> <cliente> <nombre>ACME</nombre> <cif>455321</cif> </cliente> </clientes></pre>	<pre><clientes> <cliente> <cif>455321</cif> <nombre>AcerSA</nombre> </cliente> <cliente> <cif>455321</cif> <nombre>ACME</nombre> </cliente> </clientes></pre>

9. DTD's

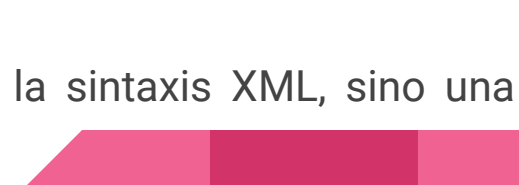
Está claro, que ninguno de los dos puede leer los archivos del otro. Por eso, resulta crítico ponerse de acuerdo en lo que se puede hacer, lo que puede aparecer y en qué orden debe hacerlo. Esto se hará mediante las DTD.

DTD significa Declaración de Tipo de Documento, y es un conjunto de reglas sintácticas para definir etiquetas. Nos indica qué etiquetas se pueden usar en un documento, en qué orden deben aparecer, cuáles pueden aparecer dentro de otras, cuáles tienen atributos, etc. Toda esta información se guarda en un archivo de texto con extensión .dtd.



9. DTD's

Un DTD:

- No es un documento XML y por lo tanto no necesita prólogo, salvo en ciertos casos especiales.
 - Crear un DTD es como crear nuestro propio lenguaje de marcado para una aplicación específica. Pueden ser parte del documento XML, pero se suele colocar aparte para poder reutilizarlo.
 - Puesto que XML es un sistema para definir lenguajes, quien necesite usar XML para intercambio de datos debe definir su propio DTD.
 - Un problema que presentan los DTDs es que no siguen la sintaxis XML, sino una propia.
- 

- Supongamos que en nuestros ficheros deseamos indicar que el elemento raíz es <listaclientes>.
- Dentro de <listaclientes> deseamos permitir uno o más elementos <cliente>.
- Dentro de <cliente> todos deberán tener <cif> y <nombre> y en ese orden.
- Dentro de <cliente> puede aparecer o no un elemento <diasentrega> para indicar que ese cliente exige un máximo de plazos.
- Como no todo el mundo usa plazos el <diasentrega> es optativo.
- Se puede indicar si un elemento aparece o no de forma opcional (usando ?)

¿Cuáles son válidos y cuáles no?

```
<listaclientes>
  <cliente>
    <cif>5676443</cif>
    <nombre>Mercasa</nombre>
  </cliente>
</listaclientes>
```

```
<listaclientes>
  <cliente>
    <cif>5676443</cif>
    <diasentrega>30</diasentrega>
  </cliente>
</listaclientes>
```

```
<listaclientes>
  <cliente>
    <cif>5676443</cif>
    <nombre>Mercasa</nombre>
    <diasentrega>30</diasentrega>
  </cliente>
  <cliente>
    <cif>5121554</cif>
    <nombre>Acer SL</nombre>
  </cliente>
</listaclientes>
```

```
<listaclientes>
  <cliente>
    <cif>5676443</cif>
    <nombre>Mercasa</nombre>
    <diasentrega>30</diasentrega>
  </cliente>
</listaclientes>
```

```
<listaclientes>
</listaclientes>
```

```
<listaclientes>
  <cliente>
    <nombre>Mercasa</nombre>
    <cif>5676443</cif>
  </cliente>
  <cliente>
    <cif>5121554</cif>
    <nombre>Acer SL</nombre>
  </cliente>
</listaclientes>
```

10. Definición de elementos en un DTD

Crear el DTD

Para ello creamos un fichero de texto con extensión .dtd.

Vamos describiendo cada elemento uno a uno desde el nodo raíz hasta el último elemento.



10. Definición de elementos en un DTD

Declaración DOCTYPE

Esta declaración es necesaria en el documento XML. Sirve para especificar a que DTD está unido el documento. Siempre debe ir colocado después del prólogo y antes del nodo raíz.

Hay tres formas de especificar un DTD:

- **SYSTEM:** Se utiliza en el caso de DTDs “personales”. Para especificar un fichero local:

```
<!DOCTYPE nodo_raiz SYSTEM "fichero.dtd">
```

- **PUBLIC:** Se utiliza en el caso de una DTD pública.

```
<!DOCTYPE nodo_raiz PUBLIC "-//OASIS//DTD Libro XML//EN" "../dtds/cap.dtd">
```

- También es posible incrustar la DTD internamente en el documento XML:

10. Definición de elementos en un DTD

DTD incrustado en el documento XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE capitulo
  [<!ELEMENT capitulo (titulo, apartado+)>
   <!ELEMENT titulo (#PCDATA)>
   <!ELEMENT apartado (#PCDATA)>
  ]>
<capitulo>
  <titulo>Introducción</titulo>
  <apartado>Primeros pasos con DTDs.</apartado>
</capitulo>
```

10. Definición de elementos en un DTD

Declaración de elementos

- Los elementos permitidos se especifican con ELEMENT, seguido del nombre y el tipo de elemento:

```
<!ELEMENT nombre_elemento tipo_elemento>
```

- **EMPTY**: significa que el elemento es vacío, es decir, que no puede tener contenido.
- **(#PCDATA)**: significa que el elemento puede contener texto. #PCDATA debe escribirse entre paréntesis.
- **ANY**: significa que el elemento puede contener cualquier cosa (texto y otros elementos). ANY debe escribirse sin paréntesis.

Dado

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

¿Cuáles de las siguientes etiquetas son correctas y cuáles no?

```
<ejemplo></ejemplo>
```

```
<ejemplo />
```

```
<ejemplo>Esto es un ejemplo</ejemplo>
```

```
<ejemplo><a></a></ejemplo>
```

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



`<ejemplo></ejemplo>`



`<ejemplo />`



`<ejemplo>Esto es un ejemplo</ejemplo>`

`<!-- ERROR: contiene texto -->`



`<ejemplo><a></ejemplo>`

`<!-- ERROR: contiene un elemento <a> -->`

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

```
<ejemplo />
```

```
<ejemplo>Esto es un ejemplo</ejemplo>
```

```
<ejemplo><a></a></ejemplo>
```

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?



```
<ejemplo />
```



```
<ejemplo>Esto es un ejemplo</ejemplo>
```



```
<ejemplo><a></a></ejemplo>
```

```
<!-- ERROR: contiene un elemento <a> -->
```

Dado

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo ANY>  
  <!ELEMENT a ANY>  

```

¿Cuál de los siguientes elementos son correctos y cuáles no?

`<ejemplo />`

`<ejemplo>Esto es un ejemplo</ejemplo>`

`<ejemplo>Esto es <a>un ejemplo</ejemplo>`

Dado

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo ANY>  
  <!ELEMENT a ANY>  

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



`<ejemplo />`



`<ejemplo>Esto es un ejemplo</ejemplo>`



`<ejemplo>Esto es <a>un ejemplo</ejemplo>`

10. Definición de elementos en un DTD

Declaración de elementos

- Para indicar que un elemento puede o debe contener otros elementos se deben indicar los elementos, utilizando los conectores y modificadores siguientes:

- **, (coma)**: significa que el elemento contiene los elementos en el orden indicado.

```
<!ELEMENT libro (titulo, autor, capitulo)>
```

libro debe contener un titulo, un autor y un capitulo exactamente y por ese orden.

- **| (o lógico)**: significa que el elemento contiene uno de los dos elementos.

```
<!ELEMENT libro (titulo | autor | capitulo)>
```

libro debe contener un titulo, un autor o un capitulo

- **? (opcional)**: significa que el elemento puede aparecer o no, pero sólo una vez.
- *****: significa que el elemento puede aparecer o no una o más veces.
- **+ (obligatorio)**: significa que el elemento tiene que aparecer una o más veces (no puede no aparecer).
- **()**: permite agrupar expresiones.

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

`<ejemplo><a /></ejemplo>`

`<ejemplo><a /><c /></ejemplo>`

`<ejemplo><a /></ejemplo>`

`<ejemplo><a /></ejemplo>`

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



<ejemplo><a /></ejemplo>



<ejemplo><a /><c /></ejemplo>

<!-- ERROR: contiene un elemento <c /> -->



<ejemplo><a /></ejemplo>

<!-- ERROR: falta el elemento -->



<ejemplo><a /></ejemplo>

<!-- ERROR: el orden no es correcto -->

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

<ejemplo><a /></ejemplo>

<ejemplo></ejemplo>

<ejemplo><a /></ejemplo>

<ejemplo></ejemplo>

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?



`<ejemplo><a /></ejemplo>`



`<ejemplo></ejemplo>`



`<ejemplo><a /></ejemplo>`

`<!-- ERROR: están los dos elementos -->`



`<ejemplo></ejemplo>`

`<!-- ERROR: no hay ningún elemento -->`

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

`<ejemplo><a /></ejemplo>`

`<ejemplo><a /></ejemplo>`

`<ejemplo></ejemplo>`

`<ejemplo></ejemplo>`

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



<ejemplo><a /></ejemplo>



<ejemplo><a /></ejemplo>



<ejemplo></ejemplo>

<!-- ERROR: falta el elemento <a /> -->



<ejemplo></ejemplo>

<!-- ERROR: el elemento aparece dos veces -->

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

<ejemplo></ejemplo>

<ejemplo><a /></ejemplo>

<ejemplo><a /><a /></ejemplo>

<ejemplo><a /></ejemplo>

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



`<ejemplo></ejemplo>`



`<ejemplo><a /></ejemplo>`



`<ejemplo><a /><a /></ejemplo>`



`<ejemplo><a /></ejemplo>`

`<!-- ERROR: el elemento <a /> aparece después de -->`

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

`<ejemplo><a /></ejemplo>`

`<ejemplo><a /><a /></ejemplo>`

`<ejemplo></ejemplo>`

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?



```
<ejemplo><a /><b /></ejemplo>
```



```
<ejemplo><a /><a /><b /></ejemplo>
```



```
<ejemplo><b /></ejemplo>
```

```
<!-- ERROR: falta el elemento <a /> -->
```

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

```
<ejemplo><a /><a /></ejemplo>
```

```
<ejemplo><a /><b /></ejemplo>
```

```
<ejemplo><a /></ejemplo>
```

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



<ejemplo><a /><a /></ejemplo>



<ejemplo><a /></ejemplo>



<ejemplo><a /></ejemplo>

<!-- ERROR: falta el elemento <a /> o -->

Ejemplo

Unos programadores necesitan un formato de fichero para que sus distintos programas intercambien información sobre ventas. El acuerdo al que han llegado es que su XML debería tener esta estructura:

- El elemento raíz será <listaventas>
- Toda <listaventas> tiene una o más ventas.
- Toda <venta> tiene los siguientes datos:
- Importe.
- Comprador.
- Vendedor.
- Fecha (optativa).
- Un código de factura.

```
<listaventas>
  <venta>
    <importe>1500</importe>
    <comprador>Wile E.Coyote</comprador>
    <vendedor>ACME</vendedor>
    <codigofactura>E17</codigofactura>
  </venta>
  <venta>
    <importe>750</importe>
    <comprador>Elmer Fudd</comprador>
    <vendedor>ACME</vendedor>
    <fecha>27-2-2015</fecha>
    <codigofactura>E18</codigofactura>
  </venta>
</listaventas>
```

Posible solución:

```
<!DOCTYPE listaventas[
  <!ELEMENT listaventas (venta+)>
  <!ELEMENT venta (importe, comprador,
vendedor, fecha?, codigofactura)>
  <!ELEMENT importe (#PCDATA)>
  <!ELEMENT comprador (#PCDATA)>
  <!ELEMENT vendedor (#PCDATA)>
  <!ELEMENT fecha (#PCDATA)>
  <!ELEMENT codigofactura (#PCDATA)>

]>
```

Classroom

10. Definición de elementos en un DTD

Declaración de atributos

Los atributos permitidos para un elemento se especifican con ATTLIST y sigue la siguiente sintaxis:

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

Donde:

- **nombreElemento:** es el nombre del elemento para el que se define un atributo.
- **nombreAtributo:** es el nombre del atributo.
- **tipoAtributo:** es el tipo de datos.
- **valorInicialAtributo:** es el valor predeterminado del atributo (aunque también puede indicar otras cosas).

10. Definición de elementos en un DTD

Declaración de atributos

¡Importante! Para definir varios atributos de un mismo elemento, se puede utilizar una o varias declaraciones de atributos.

```
<!ATTLIST nombreElemento nombreAtributo1 tipoAtributo1 valorInicialAtributo1>  
<!ATTLIST nombreElemento nombreAtributo2 tipoAtributo2 valorInicialAtributo2>
```



```
<!ATTLIST nombreElemento  
    nombreAtributo1 tipoAtributo1 valorInicialAtributo1  
    nombreAtributo2 tipoAtributo2 valorInicialAtributo2  
>
```

10. Definición de elementos en un DTD

Declaración de atributos

Los tipos de definiciones de atributos que podemos encontrarnos son:

- **CDATA:** el atributo contiene caracteres (sin restricciones).
- **NMTOKEN:** el atributo sólo contiene letras, dígitos, y los caracteres ".", "-", "_" y ":".
- **NMTOKENS:** el atributo sólo contiene letras, dígitos, y los caracteres ".", "-", "_", ":" (como el tipo NMTOKEN) y también espacios en blanco.
- **valores concretos:** el atributo sólo puede contener uno de los términos de una lista. La lista se escribe entre paréntesis, con los términos separados por una barra vertical "|".
- **ID:** el valor del atributo (no el nombre) debe ser único y no se puede repetir en otros elementos o atributos.

10. Definición de elementos en un DTD

Declaración de atributos

Los tipos de definiciones de atributos que podemos encontrarnos son:

- **IDREF:** el valor del atributo debe coincidir con el valor del atributo ID de otro elemento.
- **IDREFS:** el valor del atributo es una serie de valores separados por espacios que coinciden con el valor del atributo ID de otros elementos.
- **ENTITY:** el valor del atributo es alguna entidad definida en la DTD.
- **ENTITIES:** el valor del atributo es alguna de las entidades de una lista de entidades definida en la DTD.
- **NOTATION:** el valor del atributo es alguna notación definida en la DTD.

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

`<ejemplo color="" />`

`<ejemplo color="amarillo" />`

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

`<ejemplo />`

`<ejemplo sabor="dulce" />`

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



`<ejemplo color="" />`



`<ejemplo color="amarillo" />`



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



`<ejemplo />`

`<!-- ERROR: falta el atributo "color", obligatorio debido al #REQUIRED -->`



`<ejemplo sabor="dulce" />`

`<!-- ERROR: el atributo "sabor" no está definido -->`

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

`<ejemplo color="" />`

`<ejemplo color="azul-marino" />`

`<ejemplo color="1" />`

`<ejemplo color="azul marino" />`

`<ejemplo color="#F0F0F0" />`

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



`<ejemplo color="" />`



`<ejemplo color="azul-marino" />`



`<ejemplo color="1" />`



`<ejemplo color="azul marino" />`

`<!-- ERROR: hay un espacio en blanco -->`



`<ejemplo color="#F0F0F0" />`

`<!-- ERROR: contiene el carácter # -->`

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?

`<ejemplo color="" />`

`<ejemplo color="1" />`

`<ejemplo color="azul marino" />`

`<ejemplo color="2*2" />`

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



`<ejemplo color="" />`



`<ejemplo color="1" />`



`<ejemplo color="azul marino" />`



`<ejemplo color="2*2" />`

`<!-- ERROR: hay un asterisco -->`

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

`<ejemplo color="" />`

`<ejemplo color="azul" />`

`<ejemplo color="verde" />`

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?



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



```
<ejemplo color="azul" />
```



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

```
<!-- ERROR: "verde" no está en la lista de valores -->
```

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
</ejemplo>
```

```
<ejemplo>  
  <libro codigo="1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
</ejemplo>
```

```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L1">Los preceptos de Ptah-Hotep</libro>  
</ejemplo>
```

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
</ejemplo>
```



```
<ejemplo>  
  <libro codigo="1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
</ejemplo>
```

<!-- ERROR: el valor de un atributo de tipo ID no puede empezar con un número -->



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L1">Los preceptos de Ptah-Hotep</libro>  
</ejemplo>
```

<!-- ERROR: no se puede repetir un atributo de tipo ID -->

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <prestamo libro="L1">Numa Nigerio</prestamo>  
</ejemplo>
```

```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <prestamo libro="L2">Numa Nigerio</prestamo>  
</ejemplo>
```

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <prestamo libro="L1">Numa Nigerio</prestamo>  
</ejemplo>
```



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <prestamo libro="L2">Numa Nigerio</prestamo>  
</ejemplo>
```

<!-- ERROR: el valor "L2" no es ID de ningún elemento -->

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
  <prestamo libro="L1 L2">Numa Nigerio</prestamo>  
</ejemplo>
```


```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
  <prestamo libro="L1">Numa Nigerio</prestamo>  
</ejemplo>
```

```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
  <prestamo libro="L3">Numa Nigerio</prestamo>  
</ejemplo>
```


Dado

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


¿Cuál de los siguientes elementos son correctos y cuáles no?



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
  <prestamo libro="L1 L2">Numa Nigerio</prestamo>  
</ejemplo>
```



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
  <prestamo libro="L1">Numa Nigerio</prestamo>  
</ejemplo>
```



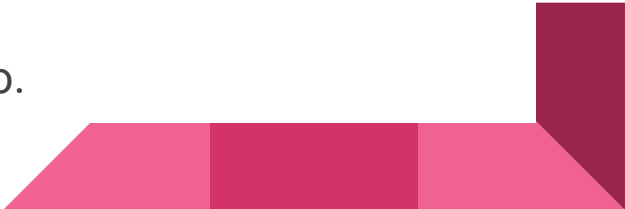
```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
  <prestamo libro="L3">Numa Nigerio</prestamo>  
</ejemplo>
```

<!-- ERROR: el valor "L3" no es ID de ningún elemento -->

10. Definición de elementos en un DTD

Declaración de atributos

Por otra parte, los atributos pueden tomar valores iniciales, que son los siguientes:

- **#REQUIRED:** el atributo es obligatorio, aunque no se especifica ningún valor predeterminado.
 - **#IMPLIED:** el atributo no es obligatorio y no se especifica ningún valor predeterminado.
 - **#FIXED valor:** el atributo tiene un valor fijo.
 - **valor concreto:** el atributo tiene un valor predeterminado.
- 

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?

`<ejemplo color="" />`

`<ejemplo color="amarillo" />`

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

`<ejemplo />`

Dado

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

```

¿Cuál de los siguientes elementos son correctos y cuáles no?



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



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



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



```
<ejemplo />
```

```
<!-- ERROR: falta el atributo "color" -->
```

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

<ejemplo />

<ejemplo color="" />

<ejemplo color="amarillo" />

<ejemplo color="azul marino #000080" />

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?



`<ejemplo />`



`<ejemplo color="" />`



`<ejemplo color="amarillo" />`



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

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

<ejemplo />

<ejemplo color="verde" />

<ejemplo color="" />

<ejemplo color="amarillo" />

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?



<ejemplo />



<ejemplo color="verde" />



<ejemplo color="" />

<!-- ERROR: el atributo "color" no tiene el valor "verde" -->



<ejemplo color="amarillo" />

<!-- ERROR: el atributo "color" no tiene el valor "verde" -->

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?

<ejemplo />

<ejemplo color="" />

<ejemplo color="amarillo" />

<ejemplo color="verde" />

Dado

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

¿Cuál de los siguientes elementos son correctos y cuáles no?



<ejemplo />



<ejemplo color="" />



<ejemplo color="amarillo" />



<ejemplo color="verde" />

Classroom