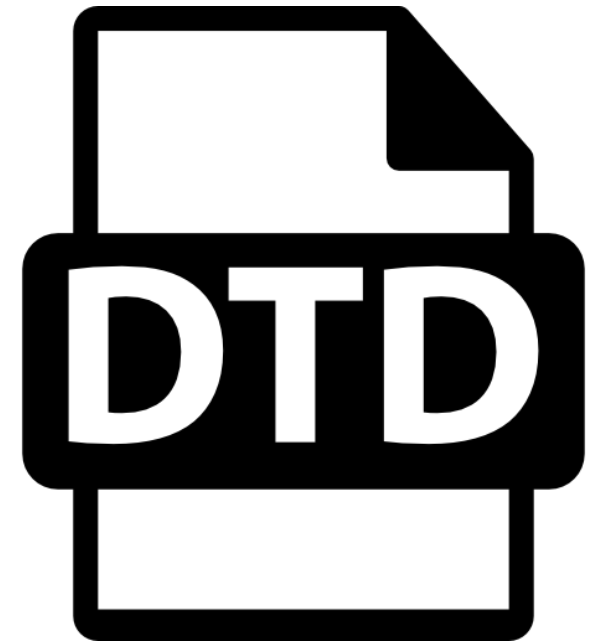


Lenguaje de Marcas y Sistemas de Gestión de Información.

UT 04.02- Validación de
Documentos XML.
Document Type Definition (DTD).



UT 04.02- Validación de Documentos XML. DTD.

1.- Definición de DTD.



- **DTD (Document Type Definition)** es una tecnología que permite comprobar **si la estructura de un documento XML se corresponde con las especificaciones técnicas necesarias.**
- Los documentos XML deben cumplir una serie de reglas sintácticas definidas en el estándar para estar bien formados.
- Además, se pueden establecer por parte del programador una serie de descripciones formales, que definen ciertas propiedades y restricciones adicionales: qué elementos y atributos están permitidos en el documento, que valores pueden tomar, etc.
- **Si documento XML cumple estas especificaciones y está libre de errores sintácticos (bien formado), entonces el documento XML es válido.**
- DTD también es útil para **detectar y corregir errores en la fase de elaboración del documento.** ¿Qué haríamos con un pedido en el que no se ha especificado el destinatario?.

UT 04.02- Validación de Documentos XML. DTD.

2.- Declaración de marcado de DTD.

2.1.- Declaración del tipo de documento.

- Las declaraciones de marcado incluidas en la DTD pueden ser:
 - **Internas.** Se encuentran dentro del propio documento XML. Se denomina **subconjunto interno**.
 - **Externas.** Se encuentran en un fichero separado. Se denomina **subconjunto externo**.
 - **Mixtas.** Una combinación de ambas.

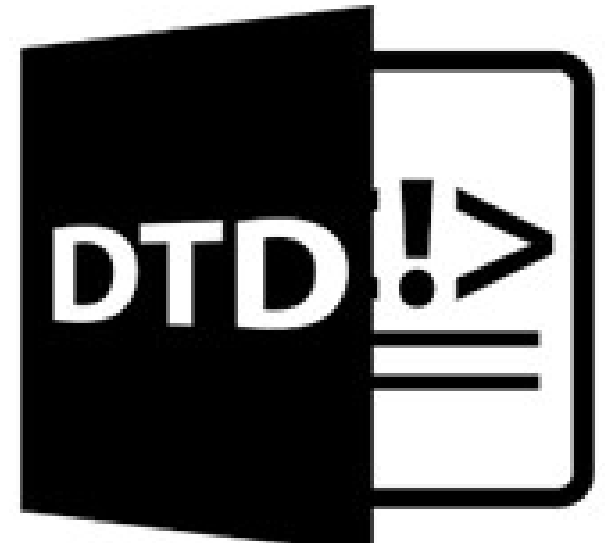
Declaración del tipo de documento.

- La declaración del tipo de documento especifica el nombre del tipo, y se incluye dentro del prólogo del documento XML mediante la referencia **!DOCTYPE** de la siguiente manera:

<!DOCTYPE root >

donde root es el **elemento raíz** del documento XML.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<!DOCTYPE casasRurales ... >  
<casasRurales>  
    ...  
</casasRurales>
```



UT 04.02- Validación de Documentos XML. DTD.

2.- Declaración de marcado de DTD.

2.2.- Definición del tipo de documento.



- La declaración de documento se complementa con una **definición de tipo de documento (DTD)**. La definición de tipo de documento indica las cualidades que posee.
- La DTD define los tipos de elementos, atributos, entidades y notaciones que se podrán utilizar en el documento.
Además, define una serie de restricciones estructurales y de contenido, valores por defecto, etc.
- Para hacer esto existen una serie de **declaraciones de marcado**. estas pueden de los siguientes tipos:
 - Tipos de elementos.
 - Listas de atributos para los tipos de elementos.
 - Entidades.
 - Notación.

UT 04.02- Validación de Documentos XML. DTD.

2.- Declaración de marcado de DTD.

2.3.- Definición interna de documento (I).

- Las declaraciones de marcado **están incluidas dentro del documento XML.**
- Para referenciar un DTD interno, **el atributo standalone en la declaración XML debe tener como valor “yes”.**
- En este caso, dichas declaraciones se incluyen dentro de unos corchetes que siguen a la declaración del tipo del documento. El formato es el siguiente:

```
<!DOCTYPE NombreXML [  
    ...  
>
```

- Estas declaraciones **pertenecen únicamente a un documento y que no es posible usarla en otros documentos.**



UT 04.02- Validación de Documentos XML. DTD.

2.- Declaración de marcado de DTD.

2.3.- Definición interna de documento (II). Ejemplo.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>

<!DOCTYPE casasRurales [
  <!ELEMENT casasRurales (casa)*>
  <!ELEMENT casa (dirección, descripción, estado, tamaño)>
  <!ELEMENT dirección (#PCDATA) >
  <!ELEMENT descripción (#PCDATA) >
  <!ELEMENT estado (#PCDATA) >
  <!ELEMENT tamaño (#PCDATA) >
]>

<casasRurales>
  <casa>
    <dirección>Calle Las Palmas 23. La Sagra. Toledo</dirección>
    <descripción>Se trata de una casa del siglo XVII.</descripción>
    <estado>El estado de conservación es magnífico.</estado>
    <tamaño>259</Tamaño>
  </casa>
  <casa>
    <dirección>Calle Or 5, Fregenal de la Sierra. Badajoz.</dirección>
    <descripción>Típica casa extremeña.</descripción>
    <estado>Recientemente restaurada</estado>
    <tamaño>180</Tamaño>
  </casa>
</casasRurales>
```


UT 04.02- Validación de Documentos XML. DTD.

2.- Declaración de marcado de DTD.

2.3.- Definición interna de documento (III).

- La definición del tipo anterior especifica que:
 - El tipo de documento casasRurales está formado por **varios elementos** de tipo casa.
 - Los elementos de tipo casa contienen a su vez elementos de tipo dirección, descripción, estado y tamaño, **en este orden y sin faltar ninguno**.
 - El contenido de estos elementos está formado exclusivamente por **datos carácter (#PCDATA)**.

A continuación, se incluye el contenido del documento.

- **Normas a la hora de declarar los elementos dentro de un DTD:**
 - La declaración del tipo de documento debe aparecer al inicio del documento. No está permitido en ningún otro lugar del documento.
 - La declaración de DOCTYPE, las declaraciones de elemento deben empezar con un signo de exclamación.
 - **El nombre en la declaración del tipo de documento debe coincidir con el tipo de elemento del elemento raíz.**



UT 04.02- Validación de Documentos XML. DTD.

2.- Declaración de marcado de DTD.

2.4.- Definición externa de documento (I).



- Las declaraciones de marcado también pueden encontrarse **fuera del documento XML en otro fichero**.
- Las declaraciones de marcado externas pueden referenciarse de dos maneras:
 - Mediante una declaración explícita de subconjunto externo.
 - Mediante entidades parámetro externas.
- Las declaraciones externas se comparten entre múltiples documentos XML que pertenecen al mismo tipo.
- Para el subconjunto externo la declaración del tipo del documento puede tomar alguna de las siguientes formas:
 - `<!DOCTYPE NombreXML SYSTEM "URI" >`
 - `<!DOCTYPE NombreXML PUBLIC "id_publico" "URI" >`
- Generalmente, vamos a usar la primera opción. La URI suele ser el nombre del fichero donde está recogido el DTD. Suele tener por convención extensión .dtd aunque no es obligatorio.

UT 04.02- Validación de Documentos XML. DTD.

2.- Declaración de marcado de DTD.

2.4.- Definición externa de documento (II). Ejemplo.

casasRurales.dtd

```
<!ELEMENT casasRurales (Casa)*>
<!ELEMENT casa (dirección, descripción, estado, tamaño)>
<!ELEMENT dirección (#PCDATA) >
<!ELEMENT descripción (#PCDATA) >
<!ELEMENT estado (#PCDATA) >
<!ELEMENT tamaño (#PCDATA) >
```

casasRurales.xml

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE Casas_Rurales SYSTEM "casasrurales.dtd">
<casasRurales>
  <casa>
    <dirección>Calle Las Palmas 23. La Sagra. Toledo</dirección>
    <descripción>Se trata de una casa del siglo XVII.</descripción>
    <estado>El estado de conservación es magnífico.</estado>
    <tamaño>259</Tamaño>
  </casa>
  <casa>
    <dirección>Calle Or 5, Fregenal de la Sierra. Badajoz.</dirección>
    <descripción>Típica casa extremeña.</descripción>
    <estado>Recientemente restaurada</estado>
    <tamaño>180</Tamaño>
  </casa>
</casasRurales>
```

UT 04.02- Validación de Documentos XML. DTD.

3.- Declaración de tipos de elementos.

- Definen **qué elementos están permitidos para un documento y cuál es su contenido**.
- Para que un documento XML sea de tipo válido se debe cumplir que:
 - Los elementos están en las declaraciones de tipos.
 - El contenido de cada elemento se ajusta a lo declarado en su tipo.
- Las declaraciones de tipos de elementos permitirán la detección de ciertos errores: inclusión de elementos no declarados, contenidos no válidos en un elemento, elementos obligatorios olvidados, elementos repetidos, etc.
- El formato básico de una declaración de tipo de elemento tiene la siguiente sintaxis:
<!ELEMENT' nombre contenido>
- Los tipos de elementos se declaran de uno en uno. No es posible tampoco declarar un tipo de elemento dos veces.
- Hay cuatro modelos posibles de declaración de contenido: **EMPTY, ANY, Mixed y children**.



UT 04.02- Validación de Documentos XML. DTD.

3.- Declaración de tipos de elementos.

3.1.- El contenido ANY.



- El elemento declarado como tipo ANY **puede contener cualquier combinación de datos carácter y elementos sin ninguna restricción.**

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>

<!DOCTYPE ejemploANY [
  <!ELEMENT ejemploANY ANY>
]>

<ejemploANY>
  <pera>Conferencia</pera>
  <manzana>Golden</manzana>
  <manzana>Reineta</manzana>
  <naranja>de zumo</naranja>
</ejemploANY>
```

UT 04.02- Validación de Documentos XML. DTD.

3.- Declaración de tipos de elementos.

3.2.- El contenido EMPTY.

- **El elemento no puede tener contenido.**
- Normalmente un elemento sin contenido tendrá atributos que aporten información y den sentido al elemento.

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<!DOCTYPE discos [  
  <!ELEMENT disco EMPTY>  
  
<discos>  
  <disco título="Forever Changes" anyo="1967"/>  
  <disco título="Exile On Main St." anyo="1972"/>  
  <disco título="Pet Sounds" anyo="1966"/>  
</discos>
```



UT 04.02- Validación de Documentos XML. DTD.

3.- Declaración de tipos de elementos.

3.3.- El contenido Mixed.

- Se trata de elementos formados **por datos carácter sólo o entremezclados con otros elementos.**

Elemento que contiene sólo datos carácter.

- Su sintaxis es:

<!ELEMENT nombre (#PCDATA)>

- Los elementos declarados como de tipo PCDATA no pueden contener los caracteres siguientes: “<”, “&” o “[]>”.
- No existe forma de limitar el valor de un elemento PCDATA a un determinado grupo de caracteres.

Elemento que contiene datos carácter y otros elementos.

- Su sintaxis es la siguiente:

<!ELEMENT nombre (#PCDATA | elemento_1 | elemento_2 | ...) *>

El asterisco al final es obligatorio.

- La aparición de los distintos elementos o de los datos carácter en un elemento concreto no es obligatoria y podrían no estar presentes.
- No se pueden establecer restricciones adicionales sobre el orden o sobre el número.

UT 04.02- Validación de Documentos XML. DTD.

3.- Declaración de tipos de elementos.

3.4.- El contenido Mixed. Ejemplo.



- Como se puede ver en este ejemplo, el elemento disco puede contener una combinación de texto, elemento título y elemento anyo. Cualquiera de estos tres elementos puede estar presente.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE listado_discos [
  <!ELEMENT discos ANY>
  <!ELEMENT disco (#PCDATA | título | anyo)*>
  <!ELEMENT autor (#PCDATA)>
  <!ELEMENT título (#PCDATA)>
  <!ELEMENT canción (#PCDATA)>
]>
<discos>
  <disco>El disco <título>Forever Changes</título> es del
  año <anyo>1967</anyo></disco>
  <disco><título>Exile On Main St.</título></disco>
</discos>
```


UT 04.02- Validación de Documentos XML. DTD.

3.- Declaración de tipos de elementos.

3.5.- El contenido Children. Alternancia de elementos (I).

- Este tipo se utiliza fundamentalmente para **agrupar otros elementos y formalizar estructuras. Sólo pueden contener otros elementos, y no datos carácter.**
- En la declaración del tipo se incluye el patrón que deben seguir los elementos que contiene. Hay varias posibilidades, estas son las siguientes:

- **Secuencia de elementos.** El elemento se compone de una secuencia de elementos que deben aparecer obligatoriamente.

<!ELEMENT nombre (elem1, ... , elemN)>

- **Alternativa de elementos.** Se especifica que únicamente un elemento de la lista puede formar parte del contenido en cada ocurrencia del elemento.

<!ELEMENT nombre (elem1 | ... | elemN)>

- **Combinación de modelos.** Se pueden utilizar paréntesis para agrupar secuencias y alternativas de modelos:

<!ELEMENT nombre_de_elemento (elem1 | (elem2, elem3))>



UT 04.02- Validación de Documentos XML. DTD.

3.- Declaración de tipos de elementos.

3.5.- El contenido Children. Alternancia de elementos (II).

- En este caso el elemento **menu** tiene como hijos **entrante** y **primero** que son obligatorios, seguidos por un elemento **segundo** o un elemento **postre**.



menus.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE menus [
  <!ELEMENT menu (menu)* >
  <!ELEMENT menu(entrante,primero,(segundo|postre))>
  <!ELEMENT entrante (#PCDATA)>
  <!ELEMENT primero (#PCDATA)>
  <!ELEMENT segundo (#PCDATA)>
  <!ELEMENT postre (#PCDATA)>
]>

<menus>
  <menu>
    <entrante>Ensalada de Pimientos</entrante>
    <primero>Paella de Marisco</primero>
    <postre>Natillas caseras</postre>
  </menu>
  <menu>
    <entrante>Croquetas de Bacalao</entrante>
    <primero>Habas con Jamón</primero>
    <segundo>Pescadilla a la plancha</segundo>
  </menu>
</menus>
```

UT 04.02- Validación de Documentos XML. DTD.

3.- Declaración de tipos de elementos.

3.5.- El contenido Children. Cardinalidad de elementos.

- Es posible especificar la frecuencia de repetición adjuntando al elemento un carácter después de su nombre o del cierre del paréntesis en su definición.
- Las distintas posibilidades son las siguientes:
 - **Interrogación (?)**. El elemento puede aparecer una vez o no aparecer **(0:1)**.
 - **Signo mas (+)**. Una o más repeticiones, como mínimo habrá una ocurrencia de ese elemento **(1:N)**.
 - **Asterico (*)**. Varias ocurrencias, pudiendo no aparecer **(0:N)**.
- No es posible especificar la aparición de un número concreto de veces (tres, cinco, veinte).
- En este ejemplo cada alumno tiene los siguientes elementos: un nombre (obligatorio), uno varios tutores legales, y beca (una o ninguna).

alumnos.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOC TYPE  alumnos [
  <!ELEMENT alumnos (alumno)*>
  <!ELEMENT alumno (nombre, tutor+, beca?)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT tutor (#PCDATA)>
  <!ELEMENT beca (#PCDATA)>
]>
```

UT 04.02- Validación de Documentos XML. DTD.

3.- Declaración de tipos de elementos.

3.5.- El contenido Children. Uniendo ambas cosas.

- **<!ELEMENT aviso (parrafo)>.**

El elemento aviso sólo puede contener un parrafo y solo uno.

- **<!ELEMENT aviso (titulo, parrafo)>.**

El elemento aviso debe obligatoriamente de contener un elemento titulo seguido de un parrafo. Deben de aparecer en ese orden.

- **<!ELEMENT aviso (parrafo | grafico)>.**

El elemento aviso puede contener un elemento parrafo o un elemento grafico indistintamente.

- **<!ELEMENT aviso (titulo, (parrafo | grafico))>.**

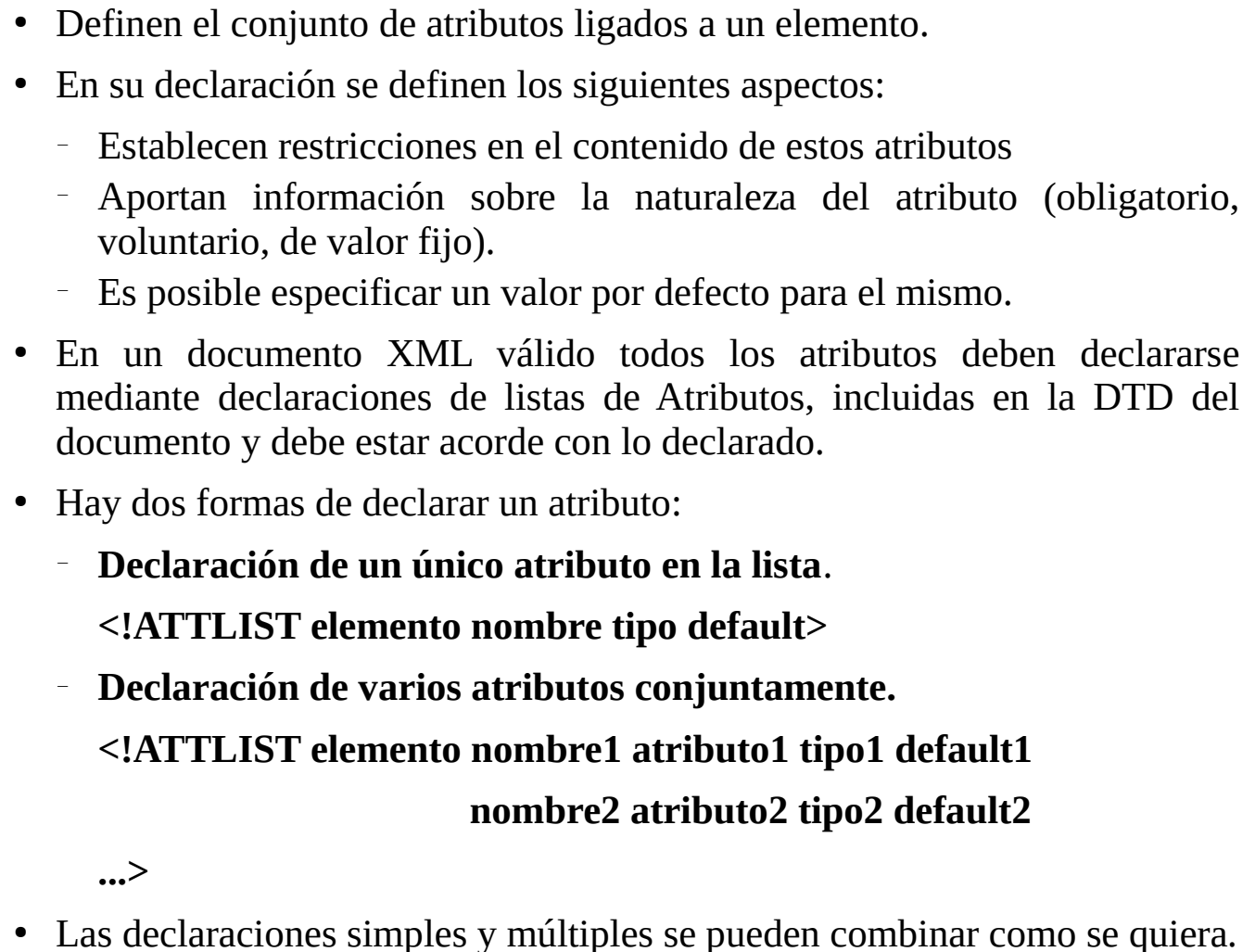
El elemento aviso debe contener un titulo seguido de uno y solo uno de los siguientes elementos: un elemento parrafo o un elemento grafico indistintamente.

- **<!ELEMENT aviso (titulo?, (parrafo+, grafico)*)>.**

El elemento aviso puede tener o no tener un solo titulo y puede tener cero o una serie de conjuntos de combinaciones de elementos parrafo (como mínimo uno) y un solo grafico.



4.1.- Definición de un atributo.



UT 04.02- Validación de Documentos XML. DTD.

4.- Declaración de atributos.

4.1.- Definición de un atributo. Ejemplo.

- En este ejemplo el elemento mascota tiene dos atributos: su fecha de nacimiento (fechaNac) y la especie a la que pertenece ambos atributos tienen como valor por defecto la cadena vacía.

mascotas.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE mascotas [
  <!ELEMENT mascotas (mascota)*>
  <!ELEMENT mascota (#PCDATA)>
  <!--ATTLIST mascota fechaNac CDATA ""-->
  <!--ATTLIST mascota especie CDATA ""-->
]>

<mascotas>
  <mascota fechaNac="06/05/2015" especie="perro">Canelo</mascota>
</mascotas>
```

mascotas.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE mascotas [
  <!ELEMENT mascotas (mascota)*>
  <!ELEMENT mascota (#PCDATA)>
  <!--ATTLIST mascota fechaNac CDATA "" especie CDATA ""-->
]>

<mascotas>
  <mascota fechaNac="06/05/2015" especie="perro">Canelo</mascota>
</mascotas>
```


UT 04.02- Validación de Documentos XML. DTD.

4.- Declaración de atributos.

4.2.- Tipo de un atributo. El tipo DATA.

- El tipo **CDATA (Character DATA)** son cadenas de texto. Admite cualquier cadena de texto (salvo los caracteres anteriormente indicados), es un valor válido para un atributo CDATA.
- Para declarar un atributo de tipo cadena, lo único que hay que hacer es añadir la palabra CDATA a continuación de su nombre en la declaración de una lista de atributos.

`<!ATTLIST mascota especie CDATA "">`

catalogo.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE catalogo [
  <!ELEMENT catalogo (articulo)*>
  <!ELEMENT articulo #PCDATA>
  <!ATTLIST articulo codigo CDATA>
]>

<catalogo>
<articulo codigo="5621">Tornillo 3mm</articulo>
<articulo codigo="2548">Tuerca 6mm</articulo>
</catalogo>
```

UT 04.02- Validación de Documentos XML. DTD.

4.- Declaración de atributos.

4.2.- Tipo de un atributo. El tipo ENUMERATION.

- Ofrecen una lista de posibles valores que puede tomar el atributo. Los valores **son nombres de tipo token**.
- Para declarar un atributo de tipo ENUMERATION hay que añadir a continuación del nombre del atributo una lista con los posibles valores que puede tomar el atributo separados entre sí por un carácter | y encerrados entre paréntesis.

<!ATTLIST elemento atributo (valor1 | valor2 | ...) default>

mascotas.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mascotas [
  <!ELEMENT mascotas (mascota)*>
  <!ELEMENT mascota (#PCDATA)>
  <!ATTLIST mascota fechNac CDATA "">
  <!ATTLIST mascota especie CDATA "">
  <!ATTLIST mascota sexo (macho | hembra) "macho">
]>
<mascotas>
  <mascota fechNac="06/05/2015" especie="perro" sexo="macho">Canelo</mascota>
</mascotas>
```

UT 04.02- Validación de Documentos XML. DTD.

4.- Declaración de atributos.

4.2.- Tipo de un atributo. El tipo TOKEN (I).

- Son semejantes a los tipos CDATA, con la diferencia que estos tipos son más restrictivos, ya que existen ciertos caracteres que no pueden formar del valor del atributo (“ “, “,”, “!”, “;”, “/”, “\”, etc).
- Hay varios tipos de atributo TOKEN. Entre ellos están los siguientes: 🎬

Atributos de tipo ID.

- Asignar un identificador único a cada uno de los elementos. Tienen las siguientes restricciones:
 - Los valores que pueden tomar son nombres XML.
 - El valor del atributo ID deberá ser único dentro del documento XML. **No puede haber dos atributos ID con el mismo valor, aunque los atributos tengan distintos nombres y se refieran a tipos de elementos diferentes.**
 - Un elemento puede tener un único atributo de tipo ID.

Atributos de tipo IDREF e IDREFS.

- Los atributos **IDREF** se utilizan para referenciar elementos identificados con un atributo ID. Cada referencia de un tipo IDREF debe estar recogida en un atributo de tipo ID, es decir, los valores que tomen estos atributos deben ser referencias a valores de atributos ID que existan en el documento.
- Los atributos **IDREFS** son análogos a los IDREF con la diferencia de que pueden tomar varios valores, separados por espacios.

UT 04.02- Validación de Documentos XML. DTD.

4.- Declaración de atributos.

4.2.- Tipo de un atributo. El tipo TOKEN (II).

Atributos de tipo ENTITY y ENTITIES.

- Los objetos (ficheros de imágenes, sonido, etc.) en un documento XML se gestionan a través de atributos entidad.
- Incluyen en su declaración el nombre del fichero que contiene los datos y una notación que identifica la aplicación que es capaz de interpretarlos.

Atributos de tipo NMTOKEN y NMTOKENS.

- Los atributos **NMTOKEN** son análogos a los de tipo CDATA, pero con la diferencia de que sus valores válidos sólo pueden ser nombres tipo token.
- Los atributos de tipo **NMTOKENS** son idénticos a los de tipo NMTOKEN con la diferencia de que aceptan varios valores al mismo tiempo. Los distintos valores se separan utilizando espacios.
- Ejemplos.

```
<!ATTLIST mascota DNIMascota ID>  
<!ATTLIST mascota padre IDREF>  
<!ATTLIST mascota madre IDREF>  
<!ATTLIST mascota nombresAnteriores NMTOKENS "">  
<!ATTLIST mascota fotografia ENTITIES "">
```

UT 04.02- Validación de Documentos XML. DTD.

4.- Declaración de atributos.

4.3.- Declaración por defecto de un atributo.

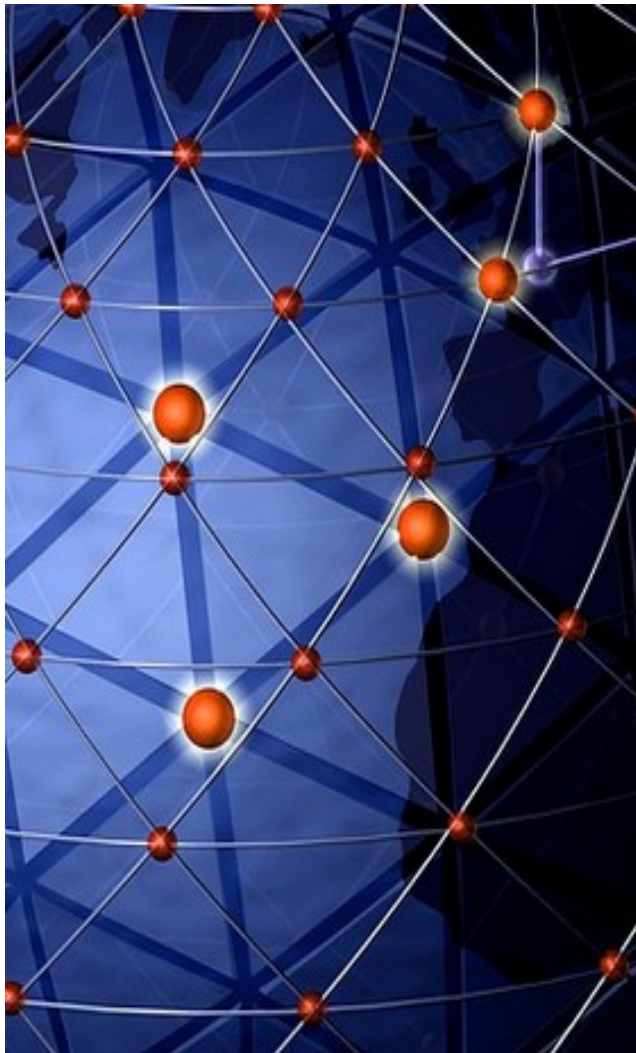
- Los posibles valores son:
 - **#REQUIRED**. Indica que en el ejemplar del documento se debe asignar obligatoriamente un valor al atributo. No hay valor por defecto.
 - **#IMPLIED**. Es opcional incluir el atributo en el elemento. No hay valor por defecto. El analizador de XML pasará un valor en blanco a la aplicación, que puede poner su propio valor por defecto. 🎬
 - **"default"**. Si se especifica un valor por defecto, el atributo queda inicializado a ese valor si no se especifica ningún otro valor en el ejemplar del documento. El valor por defecto va entre comillas pueden simples o dobles.
 - **#FIXED "valor"**. Estos atributos tienen siempre un valor fijo. No es posible modificar el valor de un atributo FIXED en el contenido del documento XML.

```
<!ATTLIST mascota DNIMascota ID #REQUIRED>  
<!ATTLIST mascota padre IDREF #IMPLIED>  
<!ATTLIST mascota madre IDREF #IMPLIED>  
<!ATTLIST mascota nombresAnteriores NMTOKENS "">  
<!ATTLIST veterinario CDATA "Desconocido">
```

UT 04.02- Validación de Documentos XML. DTD.

4.- Declaración de atributos.

4.4.- Entidades. Definición. Entidades ya existentes (I).



- Tienen una serie de funciones entre las que están las siguientes:
 - Permiten indicar abreviaturas de texto.
 - Referenciar elementos externos.
 - Utilizar caracteres que de otra forma serían inválidos en el documento.

Entidades ya existentes.

- En XML están definidas las siguientes entidades:
 - El símbolo de menor (<). “<”.
 - El símbolo de mayor (>). “>”.
 - El ampersand: &. “&”.
 - La comilla simple ('). “'”.
 - La comilla doble ("). “"”.
- **No hay que declararlas en ningún DTD**, todos los analizadores de XML estándar conocen estas entidades
- **Ejemplo.**

<autor>Leopoldo Alas 'Clarín'</autor>

El texto PCDATA del elemento autor es **Leopoldo Alas 'Clarín'.**

UT 04.02- Validación de Documentos XML. DTD.

4.- Declaración de atributos.

4.4.- Entidades. Definición. Entidades ya existentes (II).

Entidades para referencias a caracteres especiales.

- La etiqueta inicial `<?xml>` permite indicar el juego de caracteres que utiliza un documento XML (Unicode, UTF8, etc). No obstante, hay veces que es necesario escribir un carácter especial que no está contenido en el teclado de algunos entornos.
- Esto se puede hacer mediante su código en el juego de caracteres que utiliza el documento.
- Para ello, se usa la sintaxis **&#número; (decimal)** o **&#número; (hexadecimal)**.
- **Ejemplo.**

`<calle>Kantstraße, Berlín</calle>`

`<calle>Kantstra&#EF;e, Berlín</calle>`

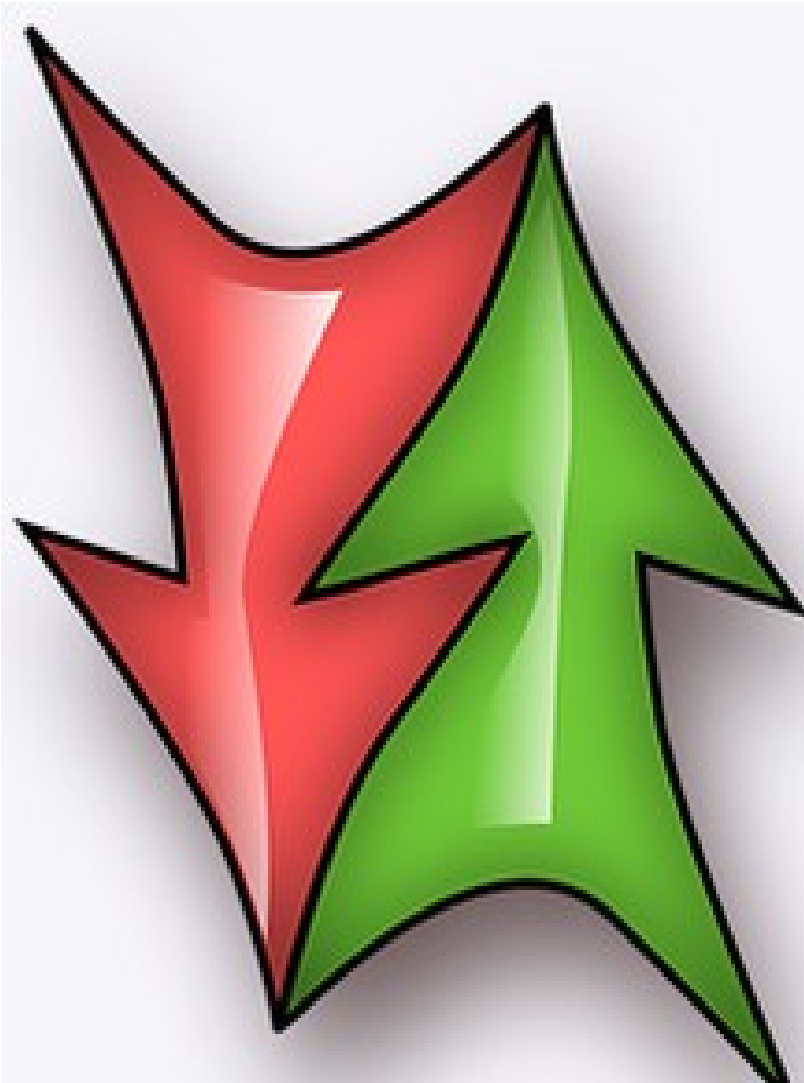
Tiene como contenido **Kantstraße, Berlín.**



UT 04.02- Validación de Documentos XML. DTD.

4.- Declaración de atributos.

4.4.- Entidades. Definición. Entidades generales.



- Se usan como **abreviaturas** que **aparecerán en el documento XML**.
- Se utilizan con fines tales como la de facilitar la escritura de nombres repetitivos.
- La sintaxis para la **declaración de una entidad general** es:

<!ENTITY nombre "texto">

- Para usar una **entidad general** en un documento XML la entidad declarada, se usa con la nomenclatura **"&nombre;"**.
- Ejemplo.

<!ENTITY mayor "Calle Mayor Principal" >

<dirección>&mayor; 18</dirección>

- Un uso muy interesante es usar entidades que hacen referencia a archivos externos.

<!ENTITY path SYSTEM "direccion.txt" >

El uso de la entidad **&path;** en un documento XML provocará que en dicho documento se añada el contenido del archivo **dirección.txt**.