

ÍNDICE

1. INTRODUCCIÓN DTD (DOCUMENT TYPE DEFINITION)	2
1.1 ¿QUÉ ES DTD?	2
1.2 INFERENCIA	2
1.3 TIPOS DE DTD	2
1.4 ¿DÓNDE Y CÓMO SE INCLUYE EL CÓDIGO DTD?	3
2. ELEMENTOS XML	4
2.1 DECLARACIÓN EN DTD	4
ACTIVIDAD 7. DTD INTERNOS Y EXTERNOS, PÚBLICOS Y PRIVADOS	5
2.2 CARDINALIDAD	6
ACTIVIDAD 8. CREAR DTD SIMPLE	6
ACTIVIDAD 9. CREAR XML Y DTD A PARTIR DE UN SUPUESTO “MUEBLES”	6
2.3 SECUENCIA DE ELEMENTOS	7
ACTIVIDAD 10. CREAR XML Y DTD A PARTIR DE UN SUPUESTO “CANTANTES”	10
3. ATRIBUTOS	11
3.1 DECLARACIÓN EN DTD	11
ACTIVIDAD 11. DTD NO VÁLIDOS CON ATRIBUTOS	14
4. ENTIDADES	15
4.1 REFERENCIAS A ENTIDADES INTERNAS	15
ACTIVIDAD 12. DEFINIR REFERENCIAS A ENTIDAD INTERNAS	15
4.2 NOTACIÓN	16

1. Introducción DTD (Document Type Definition)

1.1 ¿Qué es DTD?

Para que un documento XML sea válido, además de estar bien formado, debe respetar las reglas de la [gramática](#) que tenga definida.

Una forma de especificar la gramática es a través del DTD (Definición del Tipo de Documento): establece los elementos que pueden aparecer, su orden, cuáles son optativos, sus atributos, etc.

Actualmente la tendencia es emplear la definición de gramática a través de esquemas.

1.2 Inferencia

Cuando los documentos XML tienen muchos datos, los DTD pueden ser muy extensos, por lo que su escritura resulta tediosa.

Para facilitar el proceso, existen herramientas de [generación automática de DTD a partir de un documento XML](#), es lo que se conoce como inferencia.

Ejemplo: IDE XML Spy



1.3 Tipos de DTD

➡ **DTD Privado:** no es accesible externamente. La palabra **SYSTEM** identifica al DTD como privado.

➡ **DTD Público:** es accesible externamente para su uso compartido. Las palabras **PUBLIC** y **FPI** (identificador público de formato) identifican al DTD como público.

Un DTD público siempre debe ser DTD externo (declarado en un fichero aparte).

1.4 ¿Dónde y cómo se incluye el código DTD?

➔ **DTD Interno:** incluido en el propio documento XML. Un DTD interno siempre es privado.

Formato:

```
<!DOCTYPE elemento_raíz [reglas]>
```

Ejemplo:

```
<!DOCTYPE mensaje [  
  <!ELEMENT mensaje (contenido*)>  
  <!ELEMENT contenido (#PCDATA)>  


```

➔ **DTD externo:** codificado en un fichero externo. La ventaja de emplear este método es que una misma definición DTD puede emplearse para varios documentos XML. Puede ser público o privado. Tiene [URL](#) (Identificador de fichero externo).

Formato DTD privado:

```
<!DOCTYPE elemento_raíz SYSTEM URL>
```

Ejemplo:

```
<!DOCTYPE mensaje SYSTEM "msj.dtd">
```

Formato DTD público:

```
<!DOCTYPE elemento_raíz PUBLIC FPI URL>
```

FPI: Identificador Público de Formato

Ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
```

➔ **Forma mixta:** integra DTD interno y externo.

Formatos:

```
<!DOCTYPE elemento_raíz SYSTEM URL [reglas]>  
<!DOCTYPE elemento_raíz PUBLIC FPI URL [reglas]>
```

Ejemplos:

```
<!DOCTYPE mensaje SYSTEM "msj.dtd" [  
  <!ENTITY saludo "Buenos días">  
  <!ELEMENT contenido (#PCDATA)>  


```

```
<!DOCTYPE mensaje PUBLIC "-//W3C//DTD HTML 4.01//EN"  
  "http://www.w3.org/TR/html4/strict.dtd" [  
  <!ENTITY saludo "Buenos días">  
  <!ELEMENT contenido (#PCDATA)>  


```

2. Elementos XML

2.1 Declaración en DTD

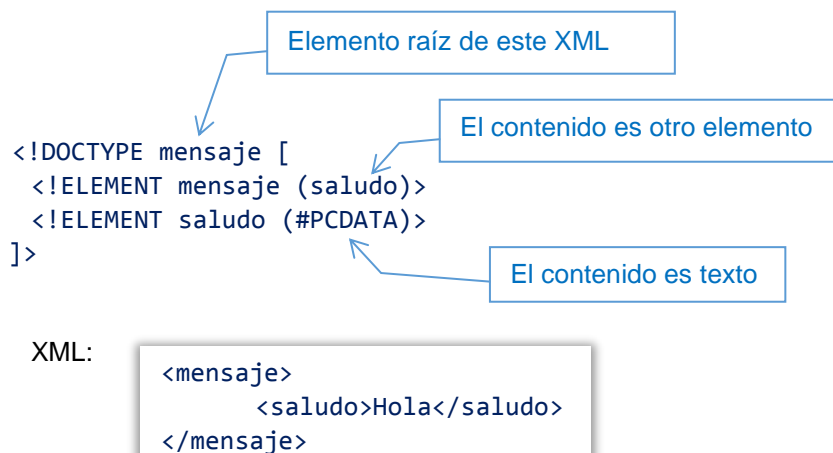
`<!ELEMENT nombre_elemento tipo_contenido>`

Tipos de contenido:

- (**#PCDATA**) Datos de tipo alfanumérico (texto). No pueden contener:
 - Otros elementos
 - `<` (se interpreta como comienzo de elemento)
 - `&` (se interpreta como entidad)
 - `]]>` (se interpreta con fin de CDATA)
- **ANY** No impone restricción (puede contener caracteres y otros elementos)
- **EMPTY** Sin contenido
- (**ELEMENTO**) Otro elemento o lista de elementos separados por comas

Hay que tener en cuenta que **el elemento raíz debe estar declarado como elemento**. Observa en el siguiente ejemplo como el elemento raíz “mensaje” está declarado como elemento con `<!ELEMENT ...>`.

Ejemplos:



```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mensaje [
  <!ELEMENT mensaje (saludo, autor)>
  <!ELEMENT saludo (#PCDATA)>
  <!ELEMENT autor EMPTY>
]>
```

El contenido son dos elementos

Elemento vacío

XML:

```
<mensaje>
  <saludo>Hola</saludo>
  <autor/>
</mensaje>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mensaje [
  <!ELEMENT mensaje (saludo, despedida)>
  <!ELEMENT saludo (#PCDATA)>
  <!ELEMENT despedida ANY>
  <!ELEMENT firma (#PCDATA)>
]>
```

Contenido sin restricción

El contenido de <despedida>
es texto y otro elemento

XML:

```
<mensaje>
  <saludo>Hola</saludo>
  <despedida>Hasta la próxima <firma>Gloria</firma></despedida>
</mensaje>
```

Actividad 7. DTD internos y externos, públicos y privados

- 7.1 Definir DTD interno y privado
- 7.2 Definir DTD externo y privado
- 7.3 Definir DTD mixto y privado
- 7.4 Definir DTD externo y público

2.2 Cardinalidad

- ? aparece 0 o 1 vez (opcional)
- * aparece 0 o n veces (opcional, repetición)
- + aparece 1 o n veces (obligatorio, repetición)

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mensaje [
  <!ELEMENT mensaje (fecha?, saludo, párrafo+, despedida)>
  <!ELEMENT saludo (#PCDATA)>
  <!ELEMENT fecha (#PCDATA)>
  <!ELEMENT párrafo (#PCDATA)>
  <!ELEMENT despedida (#PCDATA)>
]>
<mensaje>
  <saludo>Hola</saludo>
  <párrafo>Espero que estés bien</párrafo>
  <párrafo>Tengo ganas de verte</párrafo>
  <despedida>Hasta pronto</despedida>
</mensaje>
```

Actividad 8. Crear DTD simple

Actividad 9. Crear XML y DTD a partir de un supuesto “muebles”

2.3 Secuencia de elementos

(A,B)	B sigue a A (A y B obligatorios)
(A B)	A o B (pero no ambos)

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mensaje [
<!ELEMENT mensaje (saludo, párrafo+, (despedida|firma))>
<!ELEMENT saludo (#PCDATA)>
<!ELEMENT párrafo (#PCDATA)>
<!ELEMENT despedida (#PCDATA)>
<!ELEMENT firma (#PCDATA)>
]>
<mensaje>
  <saludo>Hola</saludo>
  <párrafo>Espero que estés bien</párrafo>
  <despedida>Hasta pronto</despedida>
</mensaje>
```

Despedida o firma, no ambos

Como se ha comentado anteriormente, si se utiliza ANY como contenido de un elemento, no hay ninguna restricción respecto a lo que puede contener.

En ocasiones puede ser necesario mantener flexibilidad en el contenido de un elemento, pero no una libertad completa. Si se quiere limitar el contenido de un elemento a un conjunto de otros elementos, pero sin imponer restricciones de cantidad u orden podría usarse el formato:

(#PCDATA | elemento1 | elemento2)*

El contenido serán caracteres, elemento1 o elemento2 o una combinación de los tres (sin determinar el orden). El asterisco (*) implica repetición.

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mensaje [
<!ELEMENT mensaje (saludo, párrafo*, despedida)>
<!ELEMENT autor (#PCDATA)*>
<!ELEMENT saludo (#PCDATA)>
<!ELEMENT párrafo (#PCDATA)>
<!ELEMENT firma (#PCDATA)>
<!ELEMENT despedida (#PCDATA | firma | párrafo)*>
]>
<mensaje>
  <saludo>Hola</saludo>
  <despedida>
    <párrafo>Hasta la próxima</párrafo>
    Un abrazo
    <firma>Jaime</firma>
    <párrafo>Adiós</párrafo>
  </despedida>
</mensaje>
```

Despedida será una combinación de caracteres, elementos firma y elementos párrafo, en cualquier orden y cantidad. No puede contener otros elementos como "autor"

En este caso "despedida" incluye:

- Elemento párrafo
- Caracteres
- Elemento firma
- Elemento párrafo

Para los siguientes casos, indica si los XML son válidos en función de su definición DTD. En caso de que no lo sean, indica el error.

a)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE discografia [
<!ELEMENT discografia (disco)>
<!ELEMENT disco (#PCDATA)>
]>
<discografia>
  <disco>A Broken Frame</disco>
  <disco>Construction Time Again</disco>
</discografia>
```

¿Es válido? SI/NO

b)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE discografia [
<!ELEMENT discografia (disco*)>
<!ELEMENT disco (#PCDATA)>
]>
<discografia>
  <disco>Speak & Spell</disco>
  <disco>A Broken Frame</disco>
  <disco>Construction Time Again</disco>
</discografia>
```

¿Es válido? SI/NO

c)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE discografia [
<!ELEMENT discografia (disco*)>
<!ELEMENT disco (titulo,fecha)>
]>
<discografia>
  <disco>
    <titulo>Construction Time Again</titulo>
    <fecha>1983</fecha>
  </disco>
  <disco>
    <titulo>Some Great Reward</titulo>
  </disco>
</discografia>
```

¿Es válido? SI/NO

d)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE discografia [
<!ELEMENT discografia (disco*)>
<!ELEMENT disco (titulo,fecha?)>
<!ELEMENT titulo ANY>
<!ELEMENT fecha (#PCDATA)>
]>
<discografia>
  <disco>
    <titulo>Some Great Reward</titulo>
    <fecha>1984</fecha>
  </disco>
  <disco>
    <titulo>Black Celebration <fecha>1986</fecha></titulo>
  </disco>
</discografia>
```

¿Es válido? SI/NO

e)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE discografia [
<!ELEMENT discografia (disco*)>
<!ELEMENT disco (titulo,fecha,sencillo*)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT fecha (#PCDATA)>
<!ELEMENT sencillo (fecha?,(titulo|promocional))>
<!ELEMENT promocional (#PCDATA|fecha)*>
]>
<discografia>
  <disco>
    <titulo>Music for the Masses</titulo>
    <fecha>1987</fecha>
  </disco>
  <disco>
    <titulo>Violator</titulo>
    <fecha>1990</fecha>
    <sencillo>
      <titulo>Enjoy the Silence</titulo>
    </sencillo>
    <sencillo>
      <promocional>Publicado en <fecha>1990</fecha></promocional>
    </sencillo>
  </disco>
</discografia>
```

¿Es válido? SI/NO

f)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE discografia [
  <!ELEMENT discografia (disco*)>
  <!ELEMENT disco (titulo,fecha,sencillo+)>
  <!ELEMENT titulo (#PCDATA)>
  <!ELEMENT fecha (#PCDATA)>
  <!ELEMENT sencillo (#PCDATA)>
]>
<discografia>
  <disco>
    <titulo>Music for the Masses</titulo>
    <fecha>1987</fecha>
  </disco>
  <disco>
    <titulo>Violator</titulo>
    <fecha>1990</fecha>
    <sencillo>Enjoy the Silence</sencillo>
    <sencillo>Policy of Truth</sencillo>
  </disco>
</discografia>
```

¿Es válido? SI/NO

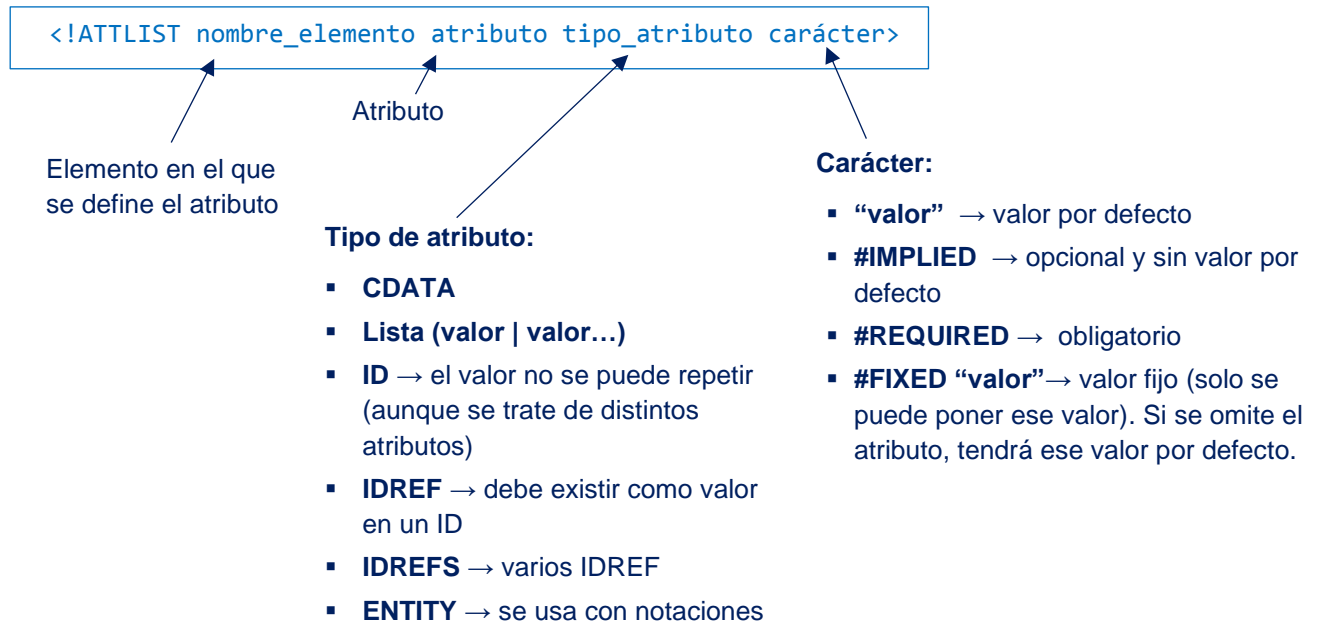
Actividad 10. Crear XML y DTD a partir de un supuesto “cantantes”

Ejercicios: Ejercicios DTD1 y DTD2.

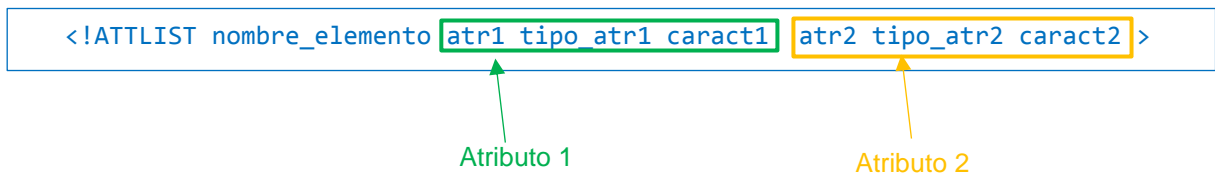
3. Atributos

3.1 Declaración en DTD

Para que a un elemento se le puedan asignar atributos, hay que declararlos en el DTD:



No existe limitación en el número de atributos que puede tener cada elemento. Pueden declararse varios atributos de un mismo elemento en un único <!ATTLIST..>:



A continuación se exponen una serie de ejemplos:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE equipo [
  <!ELEMENT equipo (persona*)>
  <!ELEMENT persona (#PCDATA)>
  <!ATTLIST persona fecha_nac CDATA "">
]>
<equipo>
<persona fecha_nac="12/03/97">Eva</persona>
</equipo>
```

valor por defecto en blanco

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tienda[
<!ELEMENT tienda (producto*)>
<!ELEMENT producto (#PCDATA)>
<!ATTLIST producto precio CDATA #REQUIRED>
]>
<tienda>
<producto precio="9.99">Estola</producto>
<producto>Caja Sorpresa</producto>
</tienda>
```

El atributo "precio" es de tipo texto y obligatorio

ERROR, "producto" no tiene el atributo precio

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE equipo [
<!ELEMENT equipo (persona*)>
<!ELEMENT persona (#PCDATA)>
<!ATTLIST persona fecha_nac CDATA "" género (hombre|mujer) "mujer" >
]>
<equipo>
  <persona>Lisa</persona>
  <persona género="hombre" fecha_nac="30/08/2009">Iván</persona>
</equipo>
```

El atributo "fecha_nac" es de tipo texto y no tiene valor por defecto (es opcional)

Al atributo "género" se le puede asignar una lista acotada de valores y su valor por defecto es "mujer"

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tienda[
<!ELEMENT tienda (producto*)>
<!ELEMENT producto (#PCDATA)>
<!ATTLIST producto peso CDATA #REQUIRED precio CDATA #IMPLIED>
]>
<tienda>
  <producto precio="9.99">Estola</producto>
  <producto peso="1 Kg">Caja Sorpresa</producto>
  <producto>Tocado</producto>
</tienda>
```

El atributo "peso" es obligatorio

El atributo "precio" es opcional sin valor por defecto

Este producto no tiene precio (es correcto)

Este producto es incorrecto porque no tiene peso

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tienda[
<!ELEMENT tienda (producto*)>
<!ELEMENT producto (#PCDATA)>
<!--ATTLIST producto precio CDATA #FIXED "20.99">
]>
<tienda>
  <producto precio="20.99">Tocado</producto>
  <producto>Estola</producto>
  <producto precio="14">Caja Sorpresa</producto>
</tienda>
```

El atributo "precio" tiene el valor fijo "20.99"

El valor del producto "Estola" será "20.99" aunque no se indique

ERROR, no tiene el valor correcto

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE oficina[
<!ELEMENT oficina (empleado|despacho)*>
<!ELEMENT empleado (#PCDATA)>
<!--ATTLIST empleado numEmpleado ID #REQUIRED>
<!--ELEMENT despacho EMPTY>
<!--ATTLIST despacho numDespacho ID #REQUIRED>
]>
<oficina>
  <empleado numEmpleado="e130">Javier López</empleado>
  <empleado numEmpleado="e100">Ana García</empleado>
  <despacho numDespacho="e100"/>
</oficina>
```

El valor del atributo "numEmpleado" no se puede repetir.
Es obligatorio, al ser #REQUIRED.

El valor del atributo "numDespacho" no se puede repetir.
Es obligatorio, al ser #REQUIRED.

ERROR, hay un ID repetido (aunque sea en distintos atributos)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE oficina[
<!--ELEMENT oficina (empleado*)>
<!--ELEMENT empleado (#PCDATA)>
<!--ATTLIST empleado numEmpleado ID #REQUIRED
numEmpleadoJefe IDREF #IMPLIED>
]>
<oficina>
  <empleado numEmpleado="e100">Ana García</empleado>
  <empleado numEmpleado="e130" numEmpleadoJefe="e100">Javier López</empleado>
</oficina>
```

El valor del atributo "numEmpleadoJefe" debe existir previamente en algún atributo de tipo ID (en este caso "numEmpleado").

Al ser #IMPLIED Es opcional sin valor por defecto

El valor "e100" debe existir como ID

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE oficina[
<!ELEMENT oficina (empleado|jefe)*>
<!ELEMENT empleado (#PCDATA)>
<!ELEMENT jefe EMPTY>
<!ATTLIST empleado numEmpleado ID #REQUIRED >
<!ATTLIST jefe numEmpleadoJefe IDREF #IMPLIED>
]>
<oficina>
  <empleado numEmpleado="e130" >Javier López</empleado>
  <empleado numEmpleado="e100" >Ana García</empleado>
  <jefe numEmpleadoJefe="e100"/>
</oficina>

```

Este ejemplo es como el anterior, pero el atributo "numEmpleadoJefe" pertenece a otro elemento distinto (jefe).

El valor "e100" debe existir como ID

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE oficina[
<!ELEMENT oficina (empleado*)>
<!ELEMENT empleado (#PCDATA)>
<!ATTLIST empleado numEmpleado ID #REQUIRED
numEmpleadoJefe IDREFS #IMPLIED>
]>
<oficina>
  <empleado numEmpleado="e100">Ana García</empleado>
  <empleado numEmpleado="e200">Ariel López</empleado>
  <empleado numEmpleado="e130" numEmpleadoJefe="e100 e200">Javier López</empleado>
</oficina>

```

El atributo "numEmpleadoJefe" puede tener varios valores, y todos deben existir previamente en algún atributo de tipo ID.

Al ser #IMPLIED Es opcional sin valor por defecto

Los valores "e100" y "e200" deben existir como ID (tiene 2 jefes)

Ver un ejemplo de atributo de tipo ENTITY en el apartado [Notación](#)

Actividad 11. DTD no válidos con atributos

Ejercicios: Ejercicio DTD3.

4. Entidades

4.1 Referencias a entidades internas

Las entidades permiten [definir constantes](#) en un documento XML. Cuando se usan dentro del documento XML se limitan por "&" y ";", por ejemplo &entidad;

¿Cómo trabaja el intérprete con ellos? Al procesar el documento XML, el intérprete sustituye la entidad por el valor que se le ha asociado en el DTD.

Hay una serie de entidades predefinidas, no es necesario declararlas. Permiten introducir caracteres especiales dentro del texto de XML:

Referencia a entidad	Carácter
<	<
>	>
&	&
'	'
"	"

Por otro lado, es posible definir textos particulares como constantes que podrán utilizarse en el texto XML.

Formato:

```
<!ENTITY nombre_entidad "valor">
```

Ejemplo:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE autor [
  <!ENTITY yo "Ana Martín Terranova">
  <!ELEMENT autor (#PCDATA)>
]>
<autor>&yo;</autor>
```

El contenido del elemento "autor" es "Ana Martín Terranova"

Actividad 12. Definir referencias a entidad internas

Ejercicios: Ejercicio DTD4.

4.2 Notación

Una notación describe un **formato de datos no XML** dentro de un documento XML, como por ejemplo imágenes y otros ficheros. **La utilidad es indicar el tipo de datos (declaración estándar) o una aplicación que puede interpretar estos datos.** Formato:

```
<!NOTATION tipo_dato SYSTEM aplicación>
```

Ejemplos: `<!NOTATION gif SYSTEM "aplicaciones/gifEditor.exe">` ➡ Define un programa para interpretar el tipo de dato

`<!NOTATION gif SYSTEM "image/gif">` ➡ Define el tipo de dato como estándar MIME image/gif

Una vez declarado el tipo de dato no xml, es posible utilizarlo en entidades con el formato:

```
<!ENTITY nombre_entidad SYSTEM "fichero" NDATA tipo_dato>
```

Se declara un fichero
como privado

Tipo de datos
del fichero

Un atributo de un elemento podrá tener como valor el nombre de la entidad, lo que le llevará a un fichero (definido en la entidad) que se interpretará de una forma definida en la notación:

```
<!NOTATION tipo_dato SYSTEM "declaración_tipo_dato">
<!ENTITY nombre_entidad SYSTEM "fichero" NDATA tipo_dato>
<!ATTLIST elemento atributo ENTITY #REQUIRED>
```

XML: `<elemento atributo="nombre_entidad"/>`

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE perros [
  <!ELEMENT perros (perro)*>
  <!ELEMENT perro EMPTY>
  <!ATTLIST perro foto ENTITY #REQUIRED>
  <!ENTITY canelo SYSTEM "IMG56.gif" NDATA gif>
  <!ENTITY brutus SYSTEM "IMG51.gif" NDATA gif>
  <!NOTATION gif SYSTEM "image/gif">
]>
<perros>
  <perro foto="canelo"/>
  <perro foto="brutus"/>
</perros>
```



Canelo (IMG56.gif)



Brutus (IMG51.gif)

Ejercicios: Ejercicio DTD5.