

## Ejemplo lista de clase

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<IELEMENT listaAlumnos (alumno)+>
<IELEMENT alumno (nombre, apellidos, nota?, asignatura *)>
<IELEMENT nombre (#PCDATA)>
<IELEMENT apellidos (#PCDATA)>
<IELEMENT nota (#PCDATA)>
<IELEMENT asignatura (#PCDATA | curso)*>
<IELEMENT curso (#PCDATA)>
```

listaClase.dtd

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE listaAlumnos SYSTEM "listaClase.dtd">
<listaAlumnos>
  <alumno>
    <nombre>Balta</nombre>
    <apellidos>Fdez Manjón</apellidos>
    <asignatura>paw</asignatura>
    <asignatura>lp3</asignatura> </alumno>
  </alumno>
  <nombre>Antonio</nombre>
  <apellidos>Navarro</apellidos>
  <asignatura>IG1 <curso>2</curso></asignatura></alumno>
</listaAlumnos>
```

30

©Baltasar Fernández Manjón

## Declaración de atributos en una DTD

- Los atributos se utilizan para asociar pares nombre-valor a los elementos.
- La especificación de los atributos debe aparecer sólo dentro de las etiquetas de inicio o de las etiquetas de elementos vacíos.
- La declaración comienza con ATTLIST seguida por el nombre del elemento al que pertenecen los atributos, seguida por la definición de los atributos individuales.
- El orden de los atributos es indiferente

```
<!ATTLIST nombreElemento
  nombreAtributo1 tipoAtributo1 característica1 valorPorDefecto1
  .....
  nombreAtributoN tipoAtributoN característicaN valorPorDefectoN>
```

31

©Baltasar Fernández Manjón

## Ejemplo lista de clase

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<IELEMENT listaAlumnos (alumno)+>
<IELEMENT alumno (nombre, apellidos, nota?, asignatura *)>
<IELEMENT nombre (#PCDATA)>
<IELEMENT apellidos (#PCDATA)>
<IELEMENT nota (#PCDATA)>
<IELEMENT asignatura (#PCDATA | curso)*>
<IELEMENT curso (#PCDATA)>
```

listaClase.dtd

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE listaAlumnos SYSTEM "listaClase.dtd">
<listaAlumnos>
  <alumno>
    <nombre>Balta</nombre>
    <apellidos>Fdez Manjón</apellidos>
    <asignatura>paw</asignatura>
    <asignatura>lp3</asignatura> </alumno>
  </alumno>
  <nombre>Antonio</nombre>
  <apellidos>Navarro</apellidos>
  <asignatura>IG1 <curso>2</curso></asignatura></alumno>
</listaAlumnos>
```

30

©Baltasar Fernández Manjón

## Declaración de atributos en una DTD

- Los atributos se utilizan para asociar pares nombre-valor a los elementos.
- La especificación de los atributos debe aparecer sólo dentro de las etiquetas de inicio o de las etiquetas de elementos vacíos.
- La declaración comienza con ATTLIST seguida por el nombre del elemento al que pertenecen los atributos, seguida por la definición de los atributos individuales.
- El orden de los atributos es indiferente

```
<!ATTLIST nombreElemento
  nombreAtributo1 tipoAtributo1 característica1 valorPorDefecto1
  .....
  nombreAtributoN tipoAtributoN característicaN valorPorDefectoN>
```

31

©Baltasar Fernández Manjón

## Características de los atributos

- La característica de los atributos indica si un atributo es necesario o no y como debe comportarse un analizador (parser) si no aparece el atributo en un documento.
- Posibles valores
  - #REQUIRED
    - El atributo debe estar siempre presente en la instancia del documento
  - #IMPLIED
    - El atributo es opcional
  - #FIXED
    - El atributo es opcional. Si aparece debe coincidir con el valor por defecto. Si no aparece el parser puede proporcionarle el valor por defecto
  - Valor por defecto (sin palabra clave)
    - El atributo es opcional. Si aparece debe ser un valor adecuado para su tipo de atributo. Si no aparece el parser puede proporcionarle el valor por defecto

32

©Baltasar Fernández Manjón

## Tipos de atributos

- CDATA
  - Datos formados por caracteres (cadenas de texto)
  - Con < y & es necesario usar entidades como carácter de escape (&amp; ; ó &lt; ;)
- Valores enumerados
  - Se enumera el conjunto de los posibles valores permitidos
  - Se puede proporcionar un valor por defecto
- ID
  - Identificador único para cada instancia de elemento.
- IDREF
  - Una referencia a un elemento con un atributo de tipo ID
- IDREFS
  - Una lista de valores IDREF separados por blanco(s)

33

©Baltasar Fernández Manjón

## Tipos de atributos

- NMTOKEN
  - ❑ Cadena de texto que cumple las reglas de un identificador
- NMTOKENS
  - ❑ Lista de valores NMTOKEN separados por blancos
- ENTITY
  - ❑ El nombre de una entidad predefinida
- ENTITIES
  - ❑ Lista de nombres de entidades separados por blancos
- NOTATION
  - ❑ Un tipo de notación que se declara explícitamente en la DTD

## Atributos CDATA

- Un atributo de tipo CDATA puede contener cualquier texto (secuencia de caracteres)
  - ❑ Se puede incluir referencias a entidades predefinidas (p.e. &lt; y a entidades internas
  - ❑ No se puede incluir referencias a entidades externas

```
<!ELEMENT elemConAtributos (#PCDATA)>
<!/ATTLIST elemConAtributos
aaa CDATA #REQUIRED
bbb CDATA #IMPLIED
ccc CDATA #FIXED "valor:PorDefecto" >
```

## Tipos de atributos

- NMTOKEN
  - ❑ Cadena de texto que cumple las reglas de un identificador
- NMTOKENS
  - ❑ Lista de valores NMTOKEN separados por blancos
- ENTITY
  - ❑ El nombre de una entidad predefinida
- ENTITIES
  - ❑ Lista de nombres de entidades separados por blancos
- NOTATION
  - ❑ Un tipo de notación que se declara explícitamente en la DTD

## Atributos CDATA

- Un atributo de tipo CDATA puede contener cualquier texto (secuencia de caracteres)
  - ❑ Se puede incluir referencias a entidades predefinidas (p.e. &lt; y a entidades internas
  - ❑ No se puede incluir referencias a entidades externas

```
<!ELEMENT elemConAtributos (#PCDATA)>
<!/ATTLIST elemConAtributos
aaa CDATA #REQUIRED
bbb CDATA #IMPLIED
ccc CDATA #FIXED "valor:PorDefecto" >
```

## Atributos con valores enumerados

- Se enumera el conjunto de cadenas de caracteres que forman los posibles valores permitidos
  - ❑ Estas cadenas no pueden contener espacios en blanco
- Se puede proporcionar un valor por defecto

```
<!ELEMENT elemento ANY>
<!/ATTLIST elemento
tipo (difícil/fácil) #IMPLIED
curso (1/2/3/4/5) "1">
.....
<elemento curso="2">contenido del elemento</elemento>
<elemento tipo="fácil"/>
```

## Atributo ID – identificador de elemento

- Proporciona un identificador único para una instancia de un elemento
  - ❑ Debe ser un identificador XML válido (p.e. No puede empezar por un número y no debe tener espacios en blanco)
  - ❑ Debe ser único dentro de un documento
  - ❑ Un elemento no puede tener más de un ID
  - ❑ Este atributo debe declararse como #IMPLIED o #REQUIRED

```
<!ELEMENT alumno (nombre, apellidos, nota?, asignatura)*>
<!/ATTLIST alumno
IDalumno ID #REQUIRED
repetidor CDATA #REQUIRED
sexo (h|m) "h">
.....
<alumno IDalumno="Alumno1" sexo="h" repetidor="no">
<nombre>Balta</nombre>
<apellidos>Fdez Manjón</apellidos>
<asignatura>paw</asignatura>
<asignatura>lp3</asignatura>
</alumno>
```



## Entidades – contenido reemplazable

- Es un mecanismo para asignar un nombre a un contenido, de modo que cuando se utilice el nombre se reemplace por dicho contenido
  - Simplifican la legibilidad, mantenibilidad y reusabilidad
- Distintos categorías principales de entidades (por uso)
  - Entidades generales
    - Utilizadas en documentos XML
  - Entidades paramétricas
    - Utilizadas en las DTDs
- Clasificación de las entidades (por contenido)
  - Entidades analizables (parsed entities)
    - Pueden contener cualquier contenido bien formado (llamado texto de reemplazamiento)
  - Entidades no analizables (unparsed entities)
    - Contienen datos no XML (p.e. Gráficos)

©Baltasar Fernández Manjón 42

## Entidades

- Clasificación de las entidades (por almacenamiento)
  - Internas
    - El texto de reemplazamiento se incluye en la propia definición
  - Externas
    - El contenido de reemplazamiento se almacena en un recurso externo (p.e. Fichero)
- Referencias a entidades analizables
  - Todas las entidades analizables se incluyen mediante una referencia a dicha entidad
  - Sintaxis de la referencia
    - Generales
      - ♦ &nombreReferencia;
    - Paramétricas
      - ♦ %nombreReferencia;
- Las entidades no analizables sólo se pueden usar como valores de un atributo ENTITY o ENTITIES
  - Las entidades no analizables deben tener asociada una notación

©Baltasar Fernández Manjón 43

## Entidades generales

- Algunas veces llamadas simplemente entidades
- El nombre de la entidad es sustituido por 'texto a reemplazar' dentro del documento XML
- Entidades analizables internas
  - Declaración
    - <!ENTITY nombreEntidad "texto de reemplazamiento" >

```
<!ELEMENT noticia ANY>
<!ENTITY universidad "UCM">
<!-- referencias a caracteres -->
<!ENTITY caracter "&#60;"> <!-- valor en decimal -->
<!ENTITY copyrigh "&#x00A9;"> <!-- valor en hexadecimal -->
.....
<noticia>
&universidad; es la universidad mas grande de España
&caracter;
</noticia>
```

©Baltasar Fernández Manjón 44

## Entidades paramétricas

- Usado sólo en DTD
- Facilitan la lectura y modificación de la DTD

```
<!ENTITY % string "CDATA">
<!ELEMENT persona EMPTY>
<!ATTLIST persona
    nombre %string; #REQUIRED
    apellidos %string; #REQUIRED
>
<persona nombre="Jose" apellidos="Pérez">
```

©Baltasar Fernández Manjón 45

## Entidades analizables externas

- El texto de reemplazamiento se almacena en un fichero externo
- El fichero externo debe incluir una declaración de codificación si no se utiliza el valor por defecto (UTF-8 ó UTF-16)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
No aceptamos ninguna responsabilidad por los posibles
PROBLEMAS ocasionados por nuestro software
disclaimer.txt
```

```
<?xml version="1.0" encoding="ISO-8859-1" standalone = "yes"?>
<!DOCTYPE noticia [
<!ELEMENT noticia ANY>
<!ENTITY disclaimer SYSTEM 'disclaimer.txt'>
]>
<noticia>
esto es texto &disclaimer;
</noticia>
```

## Entidades externas no analizables

- Una entidad externa no analizable es una referencia a un recurso externo (p.e. Una imagen)
- La notación asociada a la entidad es una referencia al gestor o aplicación que procesa el recurso externo (p.e. Un visualizador de imágenes)
- Declaración
  - <!ENTITY nombre SYSTEM "URL" NDATA tipo\_notación>
  - <!ENTITY nombre PUBLIC "localización1" "URL" NDATA tipo\_notación>El nombre debe coincidir con una declaración de notación para ser válido

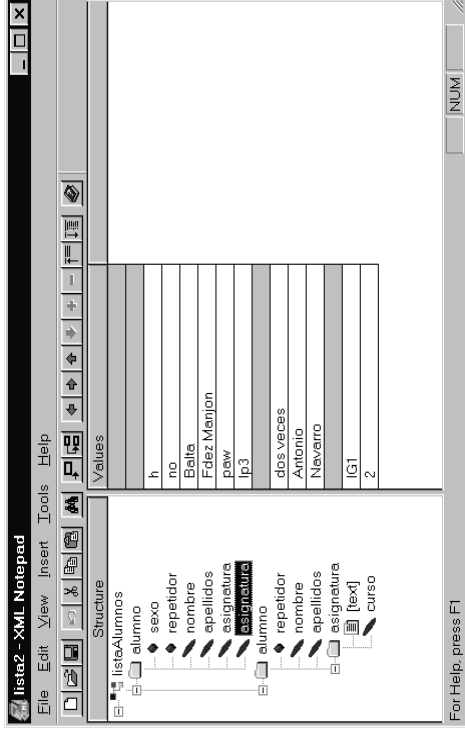
## Secciones CDATA

- Los bloques CDATA permiten especificar un bloque de texto que no se quiere que sea analizado por el procesador XML.
  - Pueden aparecer en cualquier lugar donde puedan datos de caracteres
  - No se pueden anidar ni estar vacías
- Sintaxis
  - <! [CDATA [ contenido\_no\_analizable ] ]>

```
<! [CDATA [
<libro>esto no será analizado</libro>
]]>
```

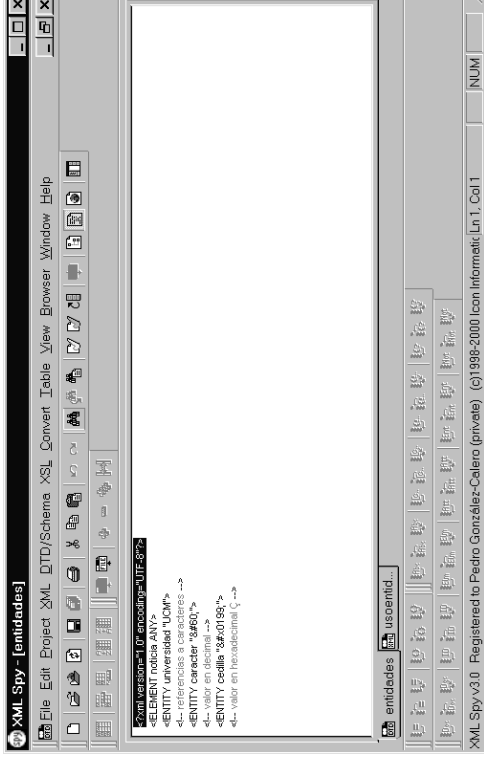
## Tipos de analizadores (parsers)

- Hay dos tipos de analizadores XML
  - Analizadores no validadores
  - Analizadores validadores
- Analizadores no validadores
  - Comprueban únicamente que un documento está bien formado
  - Internet Explorer 5.X
- Analizadores validadores (validating parsers)
  - Además de comprobar que un documento está bien formado, comprueban que este documento sigue las reglas de sintaxis y de estructura especificadas en un esquema de documento (por ejemplo una DTD)
  - XMLNotepath
  - XMLSpy



### ¿Elemento o Atributo?

- No es fácil saber cuando modelar un dato como un elemento o como un atributo de un elemento existente. En general:
- Utilizar un elemento si:
  - El dato es complejo y debería descomponerse en otros elementos mas simples
  - El dato ocupa varias líneas
  - El valor del dato cambia con frecuencia
- Utilizar un atributo si:
  - El dato es simple y su valor cambia muy poco (o nunca)
  - El dato puede tener sólo un número pequeño de valores posibles y estos son fijos y conocidos



### Problemas y limitaciones de las DTDs

- Las DTDs no están definidas utilizando XML
- En las DTDs no se diferencia entre los distintos tipos de datos posibles
  - No existe el concepto de tipo de datos
  - <persona edad="cualquiercosa">
- Como las DTDs se diseñaron para crear gramáticas documentales no son la mejor opción para las aplicaciones orientadas al intercambio de datos en las que el tipo de los datos es crucial.
  - No es posible establecer restricciones
  - <persona edad="- 13">