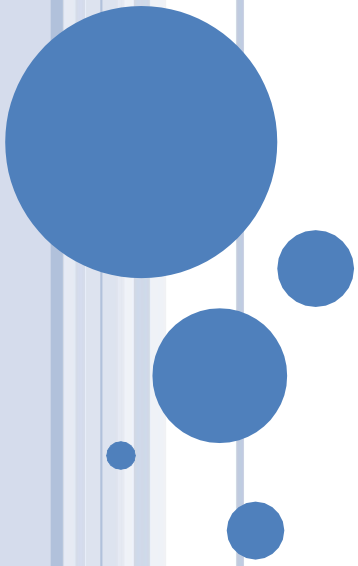


DTD

Lenguajes de Marcas y Sistemas de Gestión de la Información

CIFP La Laboral

Profesora: Inés Menéndez

¿Qué es un DTD?

- DTD significa Document Type Definition : Definición del Tipo de Documento
- ¿Qué define?
  - Los tipos de elementos, atributos y entidades permitidas que pueden aparecer en el documento XML
- Un DTD puede ser declarado en línea dentro de un documento XML, o como una referencia externa

¿QUÉ ES UN DTD?

- Los documentos XML que se ajustan a su DTD, se denominan “válidos”.
- Un documento XML “bien formado” simplemente respeta la estructura y sintaxis definidas por la especificación de XML. Un documento “bien formado” puede además ser “válido” si cumple las reglas de una DTD determinada.

DECLARACIÓN DE DTD INTERNA

- Debe seguir la siguiente sintaxis:

<!DOCTYPE elemento raíz [declaración de elementos]>

- Ejemplo de documento XML con una DTD interna:

```
<?xml version="1.0"?>
<!DOCTYPE mensaje[
<!ELEMENT mensaje (para, de, encabezado, cuerpo)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT encabezado (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
]>
<mensaje>
<para>Jaime</para>
<de>Margarita</de>
<encabezado>Recordatorio</encabezado>
<cuerpo>NO olvides comprar la comida del perro</cuerpo>
</mensaje>
```

DTD EXTERNA

- Debe seguir la siguiente sintaxis:

<!DOCTYPE elemento raíz SYSTEM “archivo”>

mensaje.xml

```
<?xml version=“1.0”?>
<!DOCTYPE mensaje SYSTEM “nota.dtd”>
<mensaje>
<para>Jaime</para>
<de>Margarita</de>
<encabezado>Recordatorio</encabezado>
<cuerpo>NO olvides comprar la comida del perro</cuerpo>
</mensaje>
```

nota.dtd

```
<!ELEMENT mensaje (para, de, encabezado, cuerpo)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT encabezado (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
```

¿Por qué utilizar una DTD?

- Con una DTD, cada uno de los archivos XML puede llevar una descripción de su propio formato.
- Con una DTD, grupos independientes de personas se ponen de acuerdo para utilizar una DTD estándar para intercambiar datos.
- Su aplicación puede utilizar una norma DTD para verificar que los datos que recibimos del mundo exterior es válida.
- También puede utilizar un DTD para verificar sus propios datos.

¿POR QUÉ NO? – LIMITACIONES DTD

- La sintaxis no es XML (son difíciles de manipular).
- En las DTDs no se diferencia entre los distintos tipos de datos posibles: No existe el concepto de tipo de datos
- 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, por ejemplo no se puede definir que un atributo 'edad' no pueda tomar un valor menor que cero.

DECLARACIÓN DE TIPO DE DOCUMENTO: <!DOCTYPE>

- Es la instrucción donde se indica qué DTD validará el XML. Aparece al comienzo del documento XML. El primer dato que aparece es el nombre del elemento raíz del documento XML.
- En función del tipo de DTD la sintaxis varía. Las características que definen el tipo son:
  - Ubicación: dónde se localizan las reglas del DTD.
    - Interno: las reglas aparecen en el propio documento XML.
    - Externo: las reglas aparecen en un archivo independiente.
    - Mixto: mezcla de los anteriores , las reglas aparecen en ambos lugares. Las reglas internas tienen prioridad sobre las externas.
  - Carácter: si es un DTD para uso privado o público.
    - Para uso privado se identifica por la palabra SYSTEM.
    - Para uso público: se identifica por la palabra PUBLIC. Debe ir acompañado del FPI (Identificador Público Formal), una etiqueta que identifica al DTD de manera 'universal'.

DECLARACIÓN DE TIPO DE DOCUMENTO: <!DOCTYPE>

- Ejemplo 1:

  Se declara un DTD respecto a un documento XML cuyo elemento raíz se llama *html* y que es de carácter público:

  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional/"EN"
  http://www.w3.org/TR/xhtml11/DTD/xhtml-transitional.dtd>

  Este ejemplo sigue la sintaxis:

  <!DOCTYPE elemento-raíz PUBLIC FPI URL>

  La url sólo existe cuando el DTD se encuentra declarado en un archivo externo, del que se da su ubicación.

- Ejemplo 2:

  Se declara el tipo de un documento XML cuyo elemento raíz se llama *<planetas>.* Se trata de un DTD de uso privado (SYSTEM) y la ubicación de las reglas del DTD es externa, en un archivo llamado *'planetas.dtd*', que se encuentra en la misma carpeta que el archivo XML.

  <!DOCTYPE planetas SYSTEM "planetas.dtd">

  Este ejemplo sigue la sintaxis:

  <!DOCTYPE elemento-raíz SYSTEM URL>

DECLARACIONES DTD

- Tipos de declaraciones:
  - ELEMENT
    - Declara un nombre de tipo de elemento y sus posibles subelementos
  - ATTLIST
    - Lista de atributos de un elemento
  - ENTITY
    - Declara referencias a caracteres especiales o a bloques de texto (similar a un #define) o también a contenido repetitivo que puede estar contenido en un recurso externo (similar a un #include).
  - NOTATION
    - Declara contenido no-XML externo (p.e. Imágenes) y la aplicación externa que gestiona dicho contenido
    - Facilitan la inclusión de formatos binarios (imágenes…)

DECLARACIÓN DE ELEMENTO

- Los elementos son la base de las marcas XML, y deben ajustarse a un tipo de documento declarado en una DTD para que el documento XML sea declarado válido.
- Las declaraciones de tipo de elemento deben empezar con <!ELEMENT seguidas por el identificador genérico del elemento que se declara:

<!ELEMENT nombre-elemento categoría>

O

<!ELEMENT nombre-elemento (elemento)>

Declaración de Elemento

- La declaración de un elemento en la DTD indica:
  - El nombre del elemento
  - El contenido que puede tener, también llamada 'declaración de contenido'.
  - La 'declaración de contenido' se escribe entre paréntesis, y en ella se puede indicar:
    - El nombre de otros elementos. No es necesario haber declarado un elemento para poder utilizarlo en la declaración de contenido de otro elemento en la misma DTD.
    - La palabra reservada #PCDATA, que indica que el elemento puede contener datos de tipo carácter.

ELEMENTO VACÍO: EMPTY

- Los elementos vacíos se declaran con la palabra clave EMPTY

```
<!ELEMENT nombre-elemento EMPTY>
```

- Ejemplo DTD:

```
<!ELEMENT br EMPTY>
```

- Ejemplo XML:

```
<br />
```

ELEMENTO CON CUALQUIER CONTENIDO: ANY

- Los elementos que pueden tener cualquier contenido son declarados con ANY. Puede contener cualquier combinación de los datos apta para su procesamiento:

```
<!ELEMENT nombre-elemento ANY>
```

- Ejemplo DTD:

```
<!ELEMENT mensaje ANY>
```

- Ejemplo XML:

```
<mensaje>
<para>Jaime</para>
<de>Margarita</de>
<encabezado>Recordatorio</encabezado>
<cuerpo>NO olvides comprar la comida del perro</cuerpo>
</mensaje>
```

ELEMENTOS DE DATOS: PCDATA

- Los elementos con datos de caracteres se analizan con la declaración (#PCDATA):

  <!ELEMENT nombre-elemento (#PCDATA)>

- Esta declaración de contenido indica que el elemento puede contener cualquier tipo de texto que no sea 'mark up', esto es cualquier letra menos < > &
- En lugar de estos caracteres se usarán las entidades < > ó &
- Las comillas simples y doble pueden sustituirse por las entidades " y '

- Ejemplo DTD:

  <!ELEMENT para (#PCDATA)>
- Ejemplo XML:

  <para>Jaime</para>

EJERCICIOS

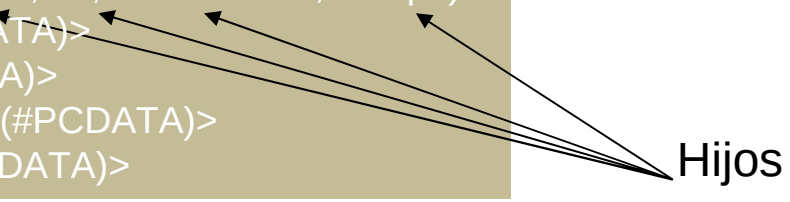
- Los elementos con uno o más hijos se declaran con el nombre de los elementos hijo entre paréntesis:

<!ELEMENT nombre-elemento (hijo1)>

<!ELEMENT nombre-elemento (hijo1, hijo2, …)>

Ejemplo DTD:

```
<!ELEMENT mensaje (para, de, encabezado, cuerpo)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT encabezado (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
```



Ejemplo XML:

```
<mensaje>
<para>Jaime</para>
<de>Margarita</de>
<encabezado>Recordatorio</encabezado>
<cuerpo>NO olvides comprar la comida del perro</cuerpo>
</mensaje>
```

EJERCICIOS

- El elemento hijo aparece una y sólo una vez:

  <!ELEMENT nombre-elemento (hijo)>

- El elemento hijo puede aparecer 0 o 1 vez:

  <!ELEMENT nombre-elemento (hijo?)>

- El elemento hijo puede aparecer 0 o más veces:

  <!ELEMENT nombre-elemento (hijo*)>

- El elemento hijo puede aparecer 1 o más veces:

  <!ELEMENT nombre-elemento (hijo+)>

- Lista de elección, sólo puede aparecer uno de los elementos:

  <!ELEMENT nombre-elemento (hijo1|hijo2)>

EJERCICIOS

https://www.youtube.com/watch?v=EfnWCeQNTQI

VIDEO DE LA UNIVERSIDAD DE ALICANTE MOSTRANDO LA CREACION DE UN DTD EXTERNO (SYSTEM)

EJERCICIOS

EJEMPLO 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE receta [
<!ELEMENT receta (titulo, ingredientes, procedimiento)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT ingredientes (#PCDATA)>
<!ELEMENT procedimiento (#PCDATA)>
]>
<receta>
<titulo>faf</titulo>
<ingredientes>fadfd</ingredientes>
<procedimiento>...</procedimiento>
</receta>
```

EXPLICACION: RECETA ESTA COMPUESTO DE LOS 3 ELEMENTOS OBLIGATORIAMENTE

EJERCICIOS

EJEMPLO 2: (SE OMITE LA DECLARACION COMPLETA DEL DTD POR BREVEDAD)

<!ELEMENT aviso (titulo?, (párrafo | grafico)*)>

<aviso>
<titulo>...</titulo>
<parrafo>...</parrafo>
<grafico>...</grafico>
<parrafo>...</parrafo>
<grafico>...</grafico>
</aviso>

EXPLICACION: AVISO SE COMPONE DE UN ELEMENTO TITULO QUE PUEDE NO ESTAR Y QUE PUEDE ESTAR SEGUIDO O NO DE UN BLOQUE DE O UN PARRAFO O UN GRAFICO

EJERCICIOS

EJEMPLO 3:

<!ELEMENT aviso (titulo?, (párrafo| grafico)*)>

```
<aviso>
        <titulo>
        </titulo>
        <parrafo>
        </parrafo>
        <parrafo>
        </parrafo>
        <grafico>
        </grafico>
</aviso>
```

EXPLICACION: AVISO ESTA FORMADO POR UN TITULO QUE PUEDE NO ESTAR O ESTAR UNA VEZ SOLO SEGUIDO DE UN BLOQUE DE PARRAFO Y GRAFICO COMO EN EL EJEMPLO ANTERIOR, EN ESTE CASO GRAFICO ESTA MUCHAS VECES

EJERCICIOS

A) DADO EL SIGUIENTE XML CREAR UN DTD QUE LO VALIDE:

```
<numeros>
 <numero>25</numero>
</numeros>
```

EJERCICIOS

A) SOLUCION:

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

EJERCICIOS

B) DADO EL SIGUIENTE XML CREAR UN DTD QUE LO VALIDE:

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

EJERCICIOS

B) SOLUCION:

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

EJERCICIOS

C) DADO EL SIGUIENTE XML CREAR UN DTD QUE LO VALIDE:

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

EJERCICIOS

C) SOLUCION:

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

EJERCICIOS

D) DADO EL SIGUIENTE XML CREAR UN DTD QUE LO VALIDE:

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

EJERCICIOS

D) SOLUCION:

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

EJERCICIOS

E) DADO EL SIGUIENTE XML CREAR UN DTD QUE LO VALIDE:

```
<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>
```

EJERCICIOS

E) SOLUCION:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE escritores [
<!ELEMENT escritores (escritor*)>
<!- - AL IMPORTAR EL ORDEN ES NECESARIO INCLUIR AMBAS COMBINACIONES - ->
<!ELEMENT escritor ((nombre, nacimiento) | (nacimiento, nombre))>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT nacimiento (#PCDATA)>
]>
```

EJERCICIOS

F) DADO EL SIGUIENTE XML CREAR UN DTD QUE LO VALIDE:

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

EJERCICIOS

F) SOLUCION:

```
<?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)>
]>
```

EJERCICIOS

G) DADO EL SIGUIENTE XML CREAR UN DTD QUE LO VALIDE:

```
<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>
```

EJERCICIOS

G) SOLUCION:

```
<?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)>
]>
```

EJERCICIOS

H) DADO EL SIGUIENTE XML CREAR UN DTD EXTERNO QUE LO VALIDE:

```
<nota>
<para>Pedro</para>
<de>Laura</de>
<titulo>Recordatorio</titulo>
<contenido>A las 7:00 en la puerta del teatro</contenido>
</nota>
```

EJERCICIOS

H) SOLUCION CON DTD EN FICHERO EXTERNO:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nota SYSTEM "DTDBASICO.dtd">
<nota>
   <para>Pedro</para>
   <de>Laura</de>
   <titulo>Recordatorio</titulo>
   <contenido>A las 7:00 en la puerta del teatro</contenido>
</nota>
```

EN DTDBASICO.DTD:

```
<!ELEMENT nota (para*, de, titulo, contenido)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT contenido (#PCDATA)>
```

EJERCICIOS

J) DADO EL SIGUIENTE XML CREAR UN DTD QUE LO VALIDE:

```
<matricula>
 <personal>
  <dni>99223366M</dni>
  <nombre>Juan Pardo Martín</nombre>
  <titulacion>Ingeniería Informática</titulacion>
  <curso_academico>1997/1998</curso_academico>
  <domicilios>
   <domicilio tipo="familiar">
    <nombre>C/ Principal nº1</nombre>
   </domicilio>
   <domicilio tipo="habitual">
    <nombre>C/ Secundaria nº2</nombre>
   </domicilio>
  </domicilios>
 </personal>
 <pago>
  <tipo_matricula>Matrícula Ordinaria</tipo_matricula>
 </pago>
</matricula>
```

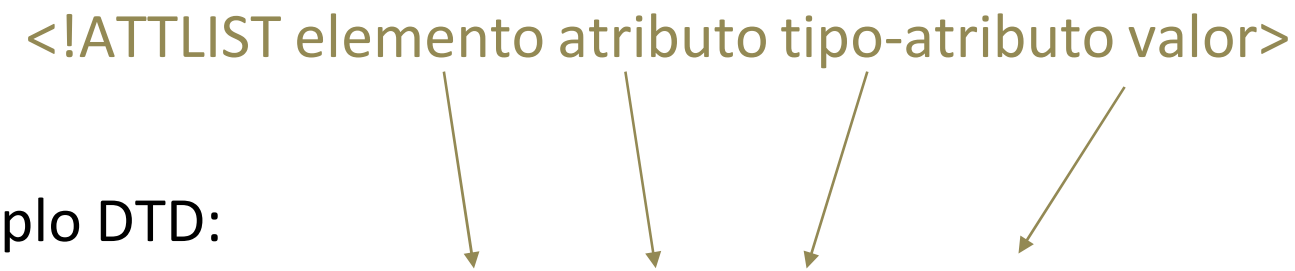
EJERCICIOS

J) SOLUCION CON DTD EN FICHERO EXTERNO:

<!ELEMENT matricula (personal, pago)+>
<!ELEMENT personal (dni, nombre, titulacion, curso_academico, domicilios)>
<!ELEMENT dni (#PCDATA)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT titulacion (#PCDATA)>
<!ELEMENT curso_academico (#PCDATA)>
<!ELEMENT domicilios (domicilio+)>
<!ATTLIST domicilio tipo (familiar|habitual) #REQUIRED>
<!ELEMENT domicilio (nombre)>
<!ELEMENT pago (tipo_matricula)>
<!ELEMENT tipo_matricula (#PCDATA)>

DECLARACIÓN DE ATRIBUTOS

- En una DTD los atributos se declaran con una declaración ATTLIST:



- Ejemplo DTD:

```
<!ATTLIST pago tipo CDATA “cheque”>
```

- Ejemplo XML:

```
<pago tipo=“cheque” />
```

Tipos de Atributo

Tipo	Descripción
CDATA	El valor es un dato de carácter
(en1 | en2 | …)	El valor debe ser uno de una lista enumerada
ID	El valor es un identificador único. No puede empezar por un número y no puede tener espacios en blanco.
IDREF	El valor es el identificador de otro elemento: apunta a un ID
IDREFS	El valor es una lista de identificadores separados por espacios
NMTOKEN	El valor es una palabra sin espacios: letras, números, puntos, guiones, subrayados y los dos puntos
NMTOKENS	El valor es una lista de múltiples NMTOKEN
ENTITY	El valor es una entidad
ENTITIES	El valor es una lista de entidades
NOTATION	El valor es un nombre de una notación
xml:	El valor es un valor predefinido xml

VALOR DE LOS ATRIBUTOS

Valor	Explicación
Valor	El valor predeterminado del atributo
#REQUIRED	El atributo es obligatorio y por tanto no tiene valor por defecto
#IMPLIED	El atributo es opcional y no se adopta automáticamente un valor por defecto
#FIXED valor	El valor del atributo es fijo

EJEMPLOS DE ATRIBUTOS

DTD:

```
<!ATTLIST contacto fax CDATA #IMPLIED>
```

XML:

```
<contacto fax="985-111111" />
```

DTD:

```
<!ATTLIST  persona numero CDATA #REQUIRED>
```

XML:

```
<persona numero="5666" />
```

DTD:

```
<!ATTLIST novela fecha CDATA #IMPLIED lugar CDATA "Gijón">
```

XML:

```
<novela fecha="1885" lugar "Oviedo">
```

EJEMPLOS DE ATRIBUTOS

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cine [
  <!ELEMENT cine (directores, peliculas)>
  <!ELEMENT directores (director)*>
    <!ELEMENT director (#PCDATA)>
      <!ATTLIST director coddir ID #REQUIRED>
  <!ELEMENT peliculas (pelicula)*>
    <!ELEMENT pelicula (#PCDATA)>
      <!ATTLIST pelicula presup CDATA #REQUIRED “100 millones”>

      <!ATTLIST pelicula direccion IDREF #REQUIRED>

      <!ATTLIST pelicula clasificacion CDATA #FIXED "TP" >
]>
```

CREA UN ID QUE SERA REFERENCIADO LUEGO

VALOR POR DEFECTO SI NO APARECE

REQUIERE QUE HAGA REFERENCIA A UN ID DECLARADO YA

ESTABLECE UN VALOR FIJO SIEMPRE PARA ESE ATRIBUTO

EJEMPLOS DE ATRIBUTOS

```
<cine>
  <directores>
    <director coddir="CE">Clint Eastwood</director>
    <director coddir="JC">James Cameron</director>
  </directores>
  <peliculas>
    <pelicula direccion="JC">Avatar</pelicula>
    <pelicula direccion="CE">Mystic River</pelicula>
    <pelicula direccion="JC" clasificacion=”TP” >Titanic</pelicula>
  </peliculas>
</cine>
```

¿USAR ELEMENTOS O ATRIBUTOS?

- Se recomienda usar los atributos lo menos posible.
- Algunos de los problemas con los atributos son:
  - No pueden contener varios valores
  - No son fácilmente extensibles, para futuros cambios
  - Son más difíciles de manipular por el código del programa
  - Los valores de los atributos no son fáciles de probar con una DTD
- Sólo utilizar los atributos como identificador único

ENTIDADES

- XML hace referencia a objetos (ficheros, páginas web, imágenes, …) que no deben ser analizados sintácticamente según las reglas de XML mediante el uso de entidades. Se declaran en la DTD con `<!ENTITY`
- Una entidad puede no ser más que una abreviatura que se utiliza como una forma corta de algunos textos. Al usar una referencia a esta entidad, el analizador sintáctico reemplaza la referencia con su contenido. En otras ocasiones es una referencia a un objeto externo o local.

TIPOS DE ENTIDADES

- Las entidades pueden ser:
  - Internas O Externas
  - Analizadas O No analizadas
  - Generales O Parámetro

TIPOS DE ENTIDADES

- Entidades generales internas:
  - Son básicamente abreviaturas definidas en la sección de la DTD del documento XML. Son siempre entidades analizadas, es decir, una vez reemplazada la referencia a la entidad por su contenido, pasa a ser parte del documento XML y como tal, es analizada por el procesador XML.

```
<!DOCTYPE texto[
<!ENTITY ovni "Objeto Volante No
  Identificado">
]>
<texto><titulo>Un día en la vida de un &ovni;
</titulo></texto>
```

TIPOS DE ENTIDADES

- Entidades generales externas analizadas:
  - Las entidades externas obtienen su contenido en cualquier otro sitio del sistema, ya sea otro archivo del disco duro, una página web o un objeto de una base de datos. Se hace referencia al contenido de una entidad así mediante la palabra SYSTEM seguida de un URI (Universal Resource Identifier)

```
<!ENTITY intro SYSTEM
  "http://server.com/intro.xml">
```

TIPOS DE ENTIDADES

- Entidades no analizadas:
  - Si el contenido de la entidad es un archivo MPEG o una imagen GIF o un fichero ejecutable EXE, el procesador XML no debería intentar interpretarlo como si fuera texto XML. Este tipo de entidades son siempre generales y externas.

```
<!ENTITY logo SYSTEM
  "http://server.com/logo.gif">
```

EJERCICIOS AVANZADOS

1) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML. HACERLO USANDO UN DTD INTERNO Y OTRO EXTERNO.

```
<bdusuarios>
<usuario clave="**" dni ="****">
<nombre>Jose Garcia</nombre>
<email>jose@jose.com</email>
<fechaNac/>
</usuario><usuario clave="**" dni ="****">
<nombre>Alfredo Fernandez</nombre>
<email>alf@here.com</email>
<fechaNac/>
        </usuario>
       </bdusuarios>
```

EJERCICIOS AVANZADOS

1) SOLUCION

```
<?xml version=”1.0” encoding=”UTF-8”?>
<!DOCTYPE bdusuarios[
<!ELEMENT bdusuarios(usuario)+>
<!ELEMENT usuario (nombre, email, fechaNac)>
<!ATTLIST usuario
   clave CDATA #REQUIRED
   dni    CDATA #REQUIRED>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT fechanac (#PCDATA)>
<!ELEMENT fechaNac EMPTY>
]>
```

EJERCICIOS AVANZADOS

2) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML. HACERLO USANDO UN DTD INTERNO Y OTRO EXTERNO.

```
<Direccion>
<Datos>fjasdfkjj</Datos>
<Datos><Calle>...</Calle></Datos>
</Direccion>
```

EJERCICIOS AVANZADOS

2) SOLUCION

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<!DOCTYPE Direccion [
   <!ELEMENT Direccion (Datos)*>
   <!ELEMENT Datos (#PCDATA | Calle)*>
   <!ELEMENT Calle (#PCDATA)>
]>
```

EJERCICIOS AVANZADOS

3) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML. HACERLO USANDO UN DTD INTERNO Y OTRO EXTERNO.

```
<Direccion>
<Datos>fjasdfkjj</Datos>
<Datos><Calle>...</Calle></Datos>
<Datos><Numero>...</Numero></Datos>
</Direccion>
```

EJERCICIOS AVANZADOS

3) SOLUCION

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<!DOCTYPE Direccion [
   <!ELEMENT Direccion (Datos)*>
   <!ELEMENT Datos (#PCDATA | Calle | Numero)*>
   <!ELEMENT Calle (#PCDATA)>
]>
```

EJERCICIOS AVANZADOS

4) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML. HACERLO USANDO UN DTD INTERNO Y OTRO EXTERNO.

```
<Direccion>
<Datos>fjasdfkjj</Datos>
<Datos>
   <Calle>...
   </Calle>
</Datos>
<Datos>
   <Numero>...
   </Numero>
</Datos>
</Direccion>
```

EJERCICIOS AVANZADOS

4) SOLUCION

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<!DOCTYPE Direccion [

    <!ELEMENT Direccion (Datos)*>
    <!ELEMENT Datos (#PCDATA | Calle | Numero)*>
    <!ELEMENT Calle (#PCDATA)>
    <!ELEMENT Numero (#PCDATA)>
```

EJERCICIOS AVANZADOS

5) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML. HACERLO USANDO UN DTD INTERNO Y OTRO EXTERNO.

```
<Direccion>
<Datos>fjasdfkjj</Datos>
<Datos>
        <Calle>...
        </Calle>
</Datos>
<Datos>
        <Calle>
        </Calle>
        <Numero>
        </Numero>
</Datos>
</Direccion>
```

EJERCICIOS AVANZADOS

5) SOLUCION

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<!DOCTYPE Direccion [

    <!ELEMENT Direccion (Datos)*>
    <!ELEMENT Datos (#PCDATA | Calle | Numero)*>
    <!ELEMENT Calle (#PCDATA)>
    <!ELEMENT Numero (#PCDATA)>
```

EJERCICIOS AVANZADOS

6) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML.

```
<Direccion>
<Datos>fjasdfkjj</Datos>
<Datos>
<Calle Barrio="arena">...</Calle></Datos>
<Datos>
<Calle Barrio="otro"></Calle>
<Numero></Numero>
</Datos>
</Direccion>
```

EJERCICIOS AVANZADOS

6) SOLUCION

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<!DOCTYPE Direccion [
   <!ELEMENT Direccion (Datos)*>
   <!ELEMENT Datos (#PCDATA | Calle|
Numero)*>
   <!ELEMENT Calle (#PCDATA)>
   <!ELEMENT Numero (#PCDATA)>
   <!ATTLIST Calle Barrio CDATA #REQUIRED>
]>
```

EJERCICIOS AVANZADOS

7) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML.

```
<Direccion>
<Datos>fjasdfkjj</Datos>
<Datos>
<Calle Barrio="arena">...
</Calle>
</Datos>
<Datos>
<Calle >
</Calle>
<Numero>
</Numero>
</Datos>
</Direccion>
```

EJERCICIOS AVANZADOS

7) SOLUCION

```
<?xml version="1.0" encoding="ISO-8859-1"
standalone="yes"?>
<!DOCTYPE Direccion [
   <!ELEMENT Direccion (Datos)*>
   <!ELEMENT Datos (#PCDATA | Calle| Numero)*>
   <!ELEMENT Calle (#PCDATA)>
   <!ELEMENT Numero (#PCDATA)>
   <!ATTLIST Calle Barrio CDATA #IMPLIED>
]>
```

EJERCICIOS AVANZADOS

7) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML.

```
<Direccion>
<Datos>fjasdfkjj</Datos>
<Datos> Esta es mi direccion
<Calle Barrio="arena">Piles
</Calle>
</Datos>
<Datos>
<Calle >
</Calle>
<Numero>
</Numero>
</Datos>
</Direccion>
```

EJERCICIOS AVANZADOS

8) SOLUCION

```
<?xml version="1.0" encoding="ISO-8859-1"
standalone="yes"?>
<!DOCTYPE Direccion [
   <!ELEMENT Direccion (Datos)*>
   <!ELEMENT Datos (#PCDATA | Calle| Numero)*>
   <!ELEMENT Calle (#PCDATA)>
   <!ELEMENT Numero (#PCDATA)>
   <!ATTLIST Calle Barrio CDATA #IMPLIED>
]>
```

EJERCICIOS AVANZADOS

9) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML.

```
<?xml version='1.0' encoding="ISO-8859-1"?>
<personas>
 <persona>
   <nombre>...</nombre>
   <apellidos>...</apellidos>
   <nacimiento dia="10" mes="Octubre" anio="2021"/>
   <direccion>
     <calle>aqui</calle>
     <poblacion>Gijon</poblacion>
     <provincia>Asturias</provincia>
     <cpostal>33203</cpostal>
   </direccion>
   <varon />
 </persona>
 <persona>
   <nombre>xxx</nombre>
```

```
        <apellidos>xxx</apellidos>
        <direccion>
          <calle>xxxx</calle>
          <poblacion>.....</poblacion>
          <provincia>....</provincia>
          <cpostal>xxxx</cpostal>
        </direccion>
        <direccion>
          <calle>xxxxx</calle>
          <poblacion>xxxx</poblacion>
          <provincia>xxxx</provincia>
          <cpostal>xxxx</cpostal>
        </direccion>
        <hembra />
      </persona>
    </personas>
```

EJERCICIOS AVANZADOS

9) SOLUCION.

```
<!ELEMENT personas [
 <!ELEMENT personas (persona+)>
 <!ELEMENT persona (nombre, apellidos, nacimiento?, direccion+,
(varon | hembra))>
 <!ELEMENT nombre (#PCDATA)>
 <!ELEMENT apellidos (#PCDATA)>
 <!ELEMENT nacimiento EMPTY>
 <!ATTLIST nacimiento
     dia    CDATA   #REQUIRED
     mes    CDATA   #REQUIRED
     anio   CDATA   #REQUIRED>
 <!ELEMENT direccion (calle,poblacion,provincia,cpostal)>
 <!ELEMENT calle (#PCDATA)>
 <!ELEMENT poblacion (#PCDATA)>
 <!ELEMENT provincia (#PCDATA)>
 <!ELEMENT cpostal (#PCDATA)>
 <!ELEMENT varon EMPTY>
 <!ELEMENT hembra EMPTY>
]>
```

EJERCICIOS AVANZADOS

10) REALIZAR EL DTD QUE VALIDE EL SIGUIENTE XML.

```
<actor nombre="Leonardo Sbaraglia" actorID="LS"/>
<actor nombre="Eusebio Poncela" actorID="EP"/>
<actor nombre="Willy Toledo" artistID="WT"/>
<actor nombre="Max Von Sydow" artistID="MVS"/>

<pelicula nombre="Intacto" peliculaID="IC" reparto="LS EP WT MVS"/>
```

EJERCICIOS AVANZADOS

10) SOLUCION.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE videoclub [
<!ELEMENT videoclub (actor*, pelicula)*>
<!ELEMENT actor (#PCDATA)>
<!ATTLIST actor nombre CDATA #REQUIRED>
<!ATTLIST actor actorID ID #REQUIRED>
<!ELEMENT pelicula EMPTY>
<!ATTLIST pelicula nombre CDATA  #REQUIRED>
<!ATTLIST pelicula peliculaID ID  #REQUIRED>
<!ATTLIST pelicula reparto IDREFS #REQUIRED>
]>
<videoclub>
<actor nombre="Leonardo Sbaraglia" actorID="LS"/>
<actor nombre="Eusebio Poncela" actorID="EP"/>
<actor nombre="Willy Toledo" actorID="WT"/>
<actor nombre="Max Von Sydow" actorID="MVS"/>

<pelicula nombre="Intacto" peliculaID="IC" reparto="LS EP WT MVS"/>
</videoclub>
```

EJERCICIOS AVANZADOS