estudios de postgrado en computación

Bases de datos avanzadas

Universidad de Los Andes

Postgrado en Computación

Prof. Isabel M. Besembel Carrera

Unidad II. Sesión 9. XML, DTD, DOM.

UNIVERSIDAD DE LOS ANDES

Notación

Conectores

,	Secuencia ordenada (y)
&	Secuencia no ordenada (y) agregación
|	Escogencia (o exclusivo)

Indicadores de ocurrencias

+	1 o varios (1..n)
?	0 ó 1 (0 .. 1)
*	0 o varios (0 ..n)

{ } grupo de símbolos

Negritas símbolos de SGML o XML, elementos terminales del vocabulario

Itálicas símbolos del usuario, elementos no terminales del vocabulario

Entre comillas o apóstrofes “ “ o ‘ ‘ valor

Intervalos valorInicial-valorFinal

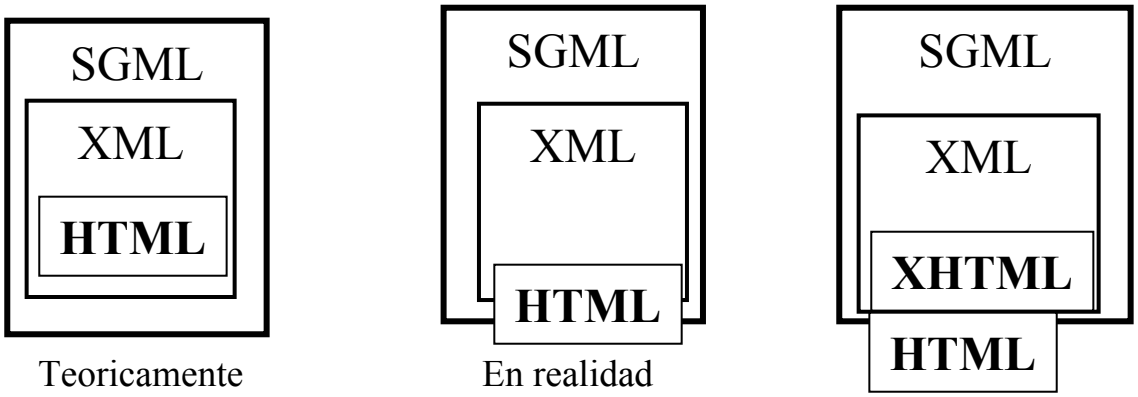
UNIVERSIDAD DE LOS ANDES

Etiqueta para marcar la información textual

XML …

- eXtensible Markup Language (publicado 1998, revisado 2000)
  - ✓ Lenguaje extensible de SGML que permite definir marcas
  - ✓ Meta-lenguaje que permite generar otros lenguajes de marcas
- Objetivo:

  Permitir que SGML pueda ser servido, recibido y procesado en Web como lo es HTML



Tomado de J-M Pinon

UNIVERSIDAD DE LOS ANDES

XML ...

- Concebido para su utilización en Internet e Intranet
- Soporta una gran cantidad de aplicaciones
- Es compatible con el núcleo de SGML
- Es simple, el número de características opcionales es mínimo
- Distingue entre minúsculas y mayúsculas
- ¿Por qué existe XML?
  - Porque SGML no es apto para la gestión hiperdocumental
  - SGML es muy complejo y lleno de opciones inútiles
  - SGML está mal adaptado a la Web, no tiene mecanismos de hiperenlaces
  - No existen visualizadores/editores de SGML de dominio público, las soluciones propietarias son costosas

UNIVERSIDAD DE LOS ANDES

XML …

- HTML está limitado a la presentación de hipertexto
- HTML es una aplicación SGML fija
  - No es posible definir nuevas marcas
  - No se adapta fácilmente a nuevas aplicaciones de los clientes

HTML :

```
<html>
<head><title>
 Dialogo extraterrestre
  </title></head>
  <body bgcolor="white"
        text="blue">
   <P> Hola tierra! </P>
   <P> Identifiquese y aterrice! </P>
  </body>
</html>
```

XML :

```
<?xml version="1.0" standalone="yes">
 --------------------
 <piezaDeTeatro>
   <titulo> Diálogo extraterrestre
   </titulo>
   <conversacion>
      <Saludo> Hola tierra!
      </Saludo>
      <Respuesta> Identifiquese y aterrice!
      </Respuesta>
   </conversacion>
 </piezaDeTeatro >
</xml>
```

Inicio de marcación

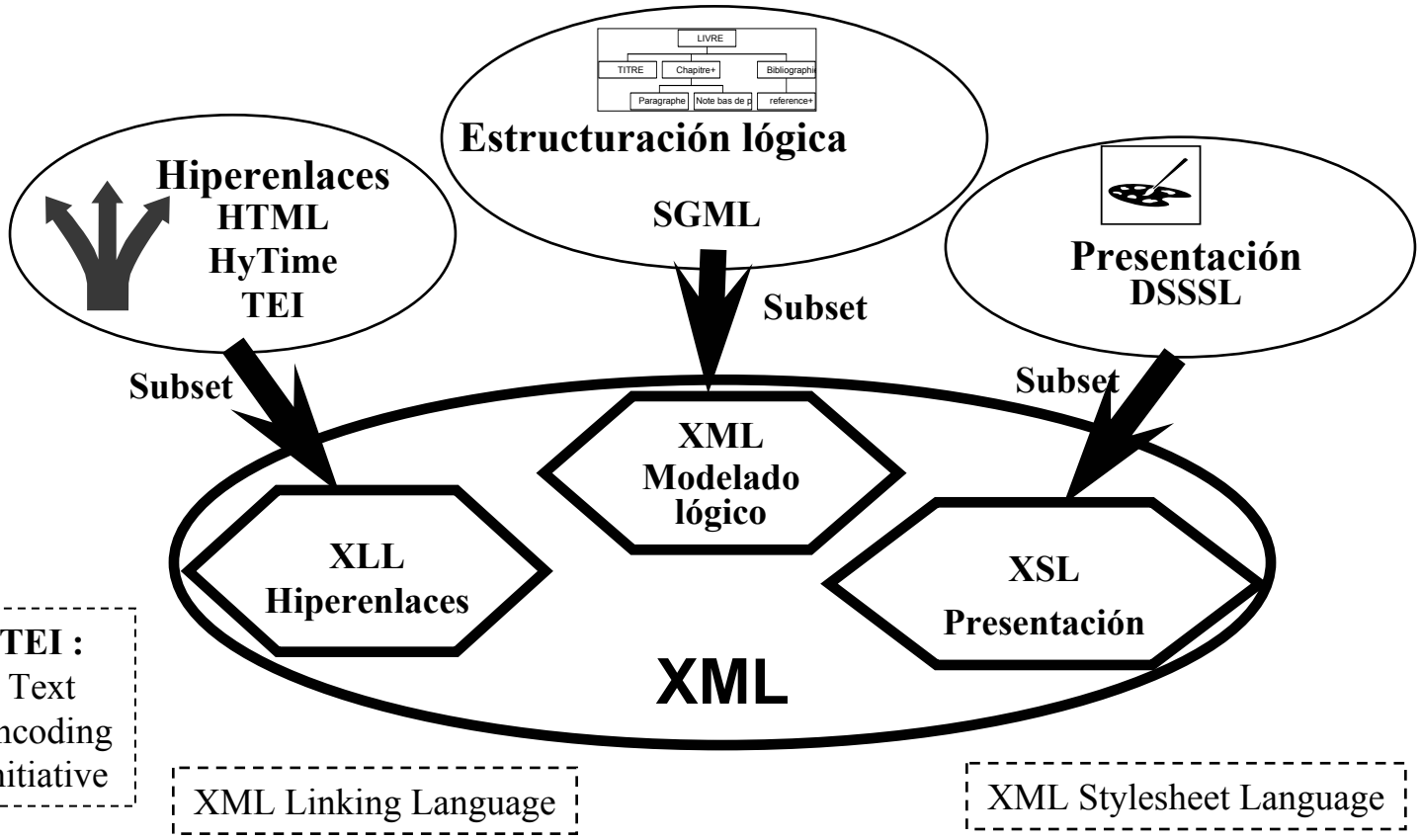
Elemento XML: inicio, contenido y fin

Fin de marcación

Tomado de J-M Pinon

UNIVERSIDAD DE LOS ANDES

XML ...



Tomado de J-M Pinon

UNIVERSIDAD DE LOS ANDES

XML ...

Gráficas y multi-media: **SMIL**, **SVG**

Llamadas remotas y B2B: **XML-RPC**, **SOAP**

Web y la publicación tradicional: **XHTML**, **DocBook**

Tecnología inalámbrica y voz: **WML**, **VoxML**

Aplicaciones

XML

Enlazamiento y señales: Xlink, Xpointer, XPath

Estilo y transformación: XSLT, XSL

Elementos de base y modelos de objetos: XML Info set, XML 1.0 + espacio de nombres, Esquema XML, DOM, SAX, XQuery

Interfaces de programación

Modelado complejo de información

Consultas sobre una colección de documentos

UNIVERSIDAD DE LOS ANDES

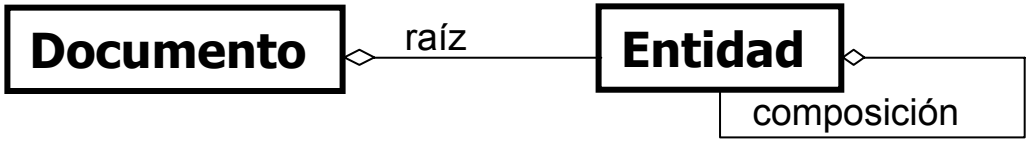
XML ...

- Diseñado para permitir la implementación e interoperabilidad con SGML y HTML
- Procesador XML: usado para leer documentos XML y permitir el acceso a su contenido y estructura
- Metas:
  - ✓ Usable en Internet, compatible con SGML y soporte de varias aplicaciones
  - ✓ Número mínimo de rasgos adicionales y facilidad de escribir programas que procesen documentos XML (claros y legibles)
  - ✓ Diseño XML formal, conciso y de fácil creación
  - ✓ No importa la minimalidad del conjunto de marcas

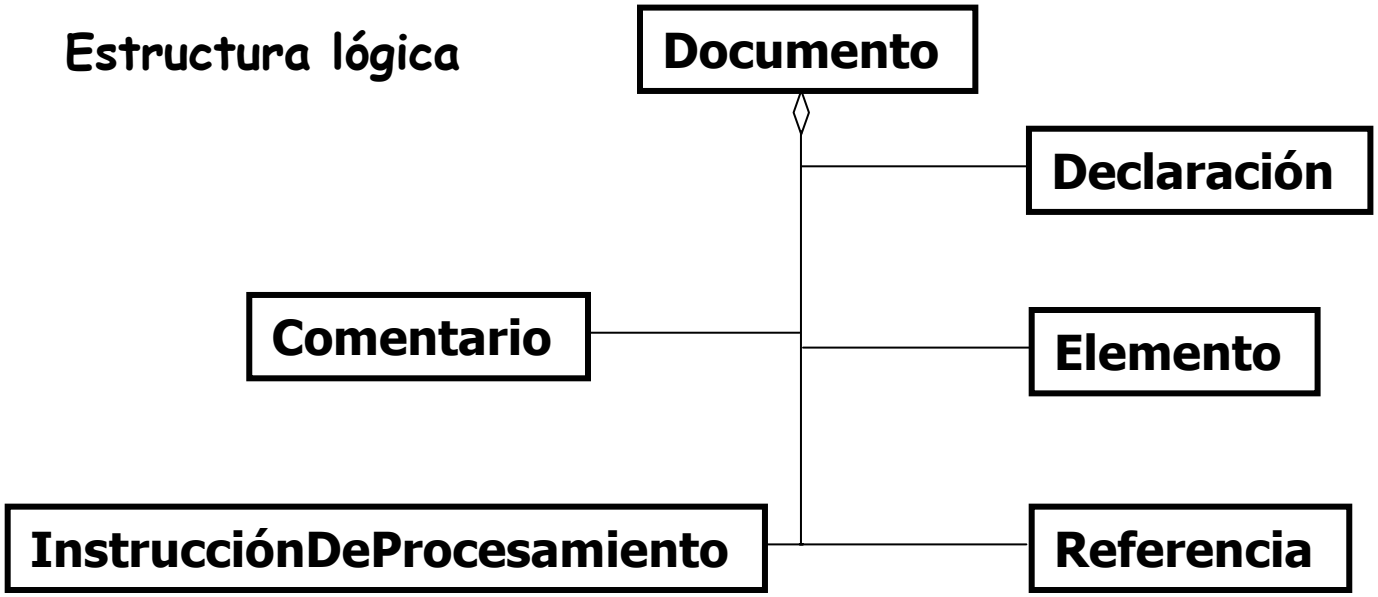
UNIVERSIDAD DE LOS ANDES

XML ...

Estructura física



Estructura lógica



UNIVERSIDAD DE LOS ANDES

Documento bien formado

- Un documento XML es un DBF si:
  - ✓ Cumple con las etiquetas de producción del documento
  - ✓ Cumple con todas las restricciones de la especificación XML
    - ∃ un elemento raíz
    - ∀ elemento ∈ documento sus marcas de inicio y fin están anidadas apropiadamente
    - Para cada elemento C ≠ raíz ∈ documento, ∃ elementos P, H ≠ C ∈ documento / C ⊂ contenido(P) y C ⊄ contenido(H), P es el padre de C y C es el hijo de P
  - ✓ Cada una de las entidades referenciadas directa o indirectamente en el documento están bien formadas

UNIVERSIDAD DE LOS ANDES

Documento XML

- Una entidad contiene texto (secuencia de caracteres) que pueden ser marcas o datos

Documento ::= Prólogo Elemento Otro *

- Estructura: árbol de elementos
- Elemento: **<***nombre listaDeAtributos***>** *contenido* **<***Inombre***>**
- *Atributo = "valor"*
- caracter (Car): unidad atómica de texto

Car ::= { #x9 | #xA | #xD | #x20-#xD7FF | #xE000-#xFFFD | #x10000-#x10FFFF }

UNIVERSIDAD DE LOS ANDES

Ejemplo

Prólogo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE drama SYSTEM "drama.dtd">
<!- -  Versión XML para el curso de Sistemas Multimedia - ->
```

Comentario

Marca de inicio

Elemento raíz y su contenido

```
<drama>
        <titulo> Ricardo II </titulo>
        <autor> William Shakespeare </autor>
        <fecha publicacion="1595"/>
        <acto numero="1">
                <escena numero="1">
                        <accion> Entra el rey Ricardo, con otros nobles
                        </accion>
                <!– aquí van otros elementos … -->>
                </escena>
        </acto>
</drama>
```

Atributo

Elemento vacío

Contenido de texto

Marca de fin

UNIVERSIDAD DE LOS ANDES

XML …

S ::= { #x20 | #x9 | #xD | #xA } +

- Espacio en blanco S consiste de 1 o más blancos (#x20), enter (#x9 o #xD) o tabulador (#xA)
- Nombre: es un token que comienza con una letra o : o _ y continua con letras, digitos, - , . , _ , :
- Se distingue entre mayúsculas y minúsculas
- Reservados: {X | x} {M | m} {L | l}
- Uso de los caracteres < > & “ ‘ dentro del texto deben ir codificados < > & & quot; & apos; respectivamente
- Soporta dos tipos de nulos: elemento ausente y elemento vacío

UNIVERSIDAD DE LOS ANDES

XML …

- & y < sólo en marcas, dentro de comentarios, secciones CDATA o instrucciones de procesamiento, de lo contrario usar código
- Preservación de los espacios en blanco dentro de los elementos: **xml:space={'preserve' | 'default'}**
- Lenguaje: **xml:lang={**'en-US' | 'sp' | 'fr' | 'en-GB' |} IETF RFC 1766
- Referencias a caracteres: inclusión de algún caracter determinado dentro del documento. Ejemplo: &#64, &#x4E
- Referencias a entidades: inclusión de un valor de entidad en el documento. Ejemplo: &amp, &apos, &gt

<![CDATA[<ejemplo> ilustración de una sección CDATA </ejemplo> **]]>**

UNIVERSIDAD DE LOS ANDES

Declaración XML

declaración XML : **<?xml,** *version, code, ref-externas,* **?>**
versión: **version =** *num-versión*
código: **encoding =** *ref-cod-ISO*
ref-externa: **standalone =** *{* **yes** | **no** *}*

Todas las declaraciones necesarias para el tratamiento del documento están incluidas en el archivo

Las declaraciones necesarias para el tratamiento del documento deben ser importadas

UNICODE (4 bytes : 4 294 967 296) o un subconjunto (2 bytes : 65 536)

Declaraciones de entidades, de elementos .....

Tomado de J-M Pinon

UNIVERSIDAD DE LOS ANDES

Ejemplo XML

```
<INVENTARIO>
        <LIBRO>
                <TITULO>Aprenda XML ya</TITULO>
                <AUTOR>Young, Michael J.</AUTOR>
                <EDITORIAL>McGrawHill-Microsoft</EDITORIAL>
                <ANO>2000</ANO>
                <ISBN>0-7356-1020-7</ISBN>
                <TITULOINGLES>XML Step-by-step</TITULOINGLES>
        </LIBRO>
        <LIBRO>
                <TITULO>Precálculo</TITULO>
                <AUTOR>Faires, J. Douglas y DeFranza, James</AUTOR>
                <EDITORIAL>International Thomson</EDITORIAL>
                <ANO>2001</ANO>
                <ISBN>970-686-032-0</ISBN>
                <TITULOINGLES>Precalculus</TITULOINGLES>
        </LIBRO>
        <LIBRO>
                <TITULO>Multimedia. Manual de referencia</TITULO>
                <AUTOR>Vaughan, Tay</AUTOR>
                <EDITORIAL>MaGraw-Hill Osborne Media</EDITORIAL>
                <ANO>2002</ANO>
                <ISBN>84-481-3626-8</ISBN>
                <TITULOINGLES>Multimedia: Making it work</TITULOINGLES>
        </LIBRO>
</INVENTARIO>
```

UNIVERSIDAD DE LOS ANDES

XML …

- Comentarios: en cualquier parte del documento fuera de las marcas y NO deben contener --

  Comentario ::= '**<- -**' { Car } '**- ->**'

- Ejemplo: **<--** Este es un comentario válido **-->**

  <-- Comentario inválido **--->**

- Un documento es válido si tiene una declaración y si cumple con todas las restricciones de su versión
- Declaración de tipo de documento (DTD): pueden estar incluidas o no dentro del documento. Si existen ambas se toman en cuenta las dos
- Declaración de marca: es una declaración de elemento, de lista de atributo, de entidad o de notación

UNIVERSIDAD DE LOS ANDES

XML …

- Símbolo: es la representación del caracter
- Pueden existir muchas representaciones para el mismo caracter
- Ejemplo: A A *A* A A
- Código UNICODE: UTF-8 y UTF-16
  - ✓ [0000-007F] caracteres comunes inglés
  - ✓ [0370-03FF] griego y cóptico
  - ✓ [0600-06FF] arábigos
  - ✓ [2600-26FF] signos matemáticos
- <?xml version="1.0" encoding="UTF-8"?>
- Documento válido: si es un DBF y cumple con la marcación de un DTD específico
- DTD: metadatos para definir los elementos de un documento

UNIVERSIDAD DE LOS ANDES

Modelos de procesamiento

- **Orientados a sucesos:** barre el documento XML secuencialmente y emite llamadas a funciones manejadoras de sucesos cuyos parámetros contienen los datos de la parte del documento

    1. Iniciar el analizador sintáctico registrando los manejadores
    2. Iniciar la lectura del documento
    3. Mientras no sea fin de documento y para cada marca

        Emitir las llamadas a los manejadores adecuados

- **Orientados al modelo de objetos:** barre el documento XML secuencialmente construyendo el árbol de objetos, éstos con sus atributos y métodos

    1. Iniciar el analizador sintáctico
    2. Iniciar la lectura del documento
    3. Recorrer el árbol resultante, buscando y tratando la información

UNIVERSIDAD DE LOS ANDES

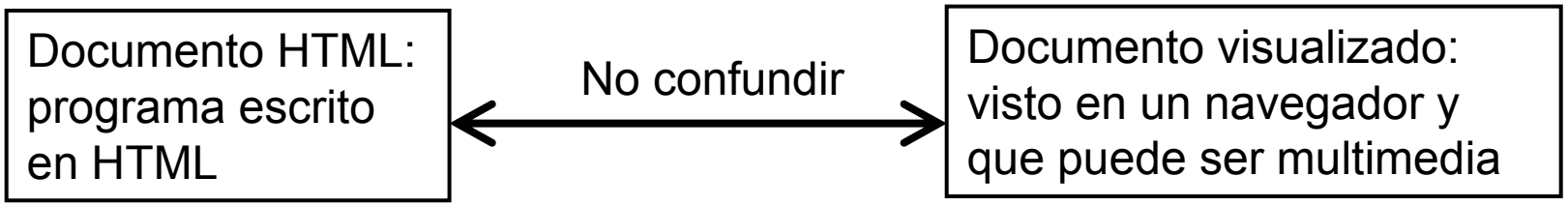
DTD …

- Document Type Definition
- Define la estructura lógica-física de un documento Web
  - ✓ Estructura simple con un número predefinido y reducido de tipos de elementos SGML o marcas
  - ✓ Elemento: nombre, atributos y texto o hipertexto
  - ✓ Formato: <marca atributo="argumento"> texto </marca>
    - Nodo: <html> contenido del nodo </html>
    - Encabezado: <head> encabezado </head>
    - Cuerpo: <body> contenido del documento </body>
    - Párrafo: contenido del párrrafo
    - Ancla: <a> texto del ancla
    - Imagen: <img> texto de la imagen </img>
    - Etc.

UNIVERSIDAD DE LOS ANDES

DTD ...

- Formatos de imágenes (jpg, gif, etc.) están predefinidos
- **Documento válido:** documento bien formado, cuyo prólogo referencia o contiene una DTD y el resto se compone de un árbol de elementos que **cumple** con la DTD.
- Contiene texto ASCII, por lo que puede ser escrito utilizando cualquier editor de texto
- URL: //host.dominio.puerto/camino
  - Ejemplo: //pgcomp.ing.ula.ve, //150.185.130.40:8080

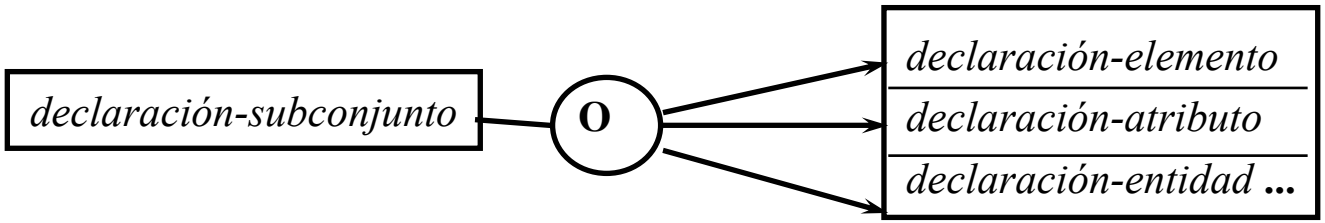


Tomado de J-M. Pinon

UNIVERSIDAD DE LOS ANDES

DTD

<! DOCTYPE, *nom-dtd* , *identificador_externo*?,
{ [, { *declaración-subconjuntoss* }+ , **] }? , >**



Ejemplo 1:
<! DOCTYPE *carta* **[** <--descripción de las componentes de una carta --> **]>**

Ejemplo 2: *(referencia a un DTD externo)*
<! DOCTYPE *carta* **SYSTEM** "cartas.dtd" **>**

Analogía: DTD es como una clase y el documento es una instancia de la clase. DTD es el esquema de la BD y el documento es su instancia. Los documentos deben cumplir con las reglas establecidas en el DTD

UNIVERSIDAD DE LOS ANDES

Elemento …

Declaración de elemento:
<!ELEMENT, *nom-elemento*, *modelo-contenido* >

- **1er car :** alfabético o _
- **luego:** alfanumérico o _ o .
- mayúscula ≠ minúscula
- **:** sentido particular
- no debe comenzar por xml

Cadena de datos analizables que puede contener elementos anidados

modelo-contenido: contenido-declarado | contenido-compuesto | contenido-mixto

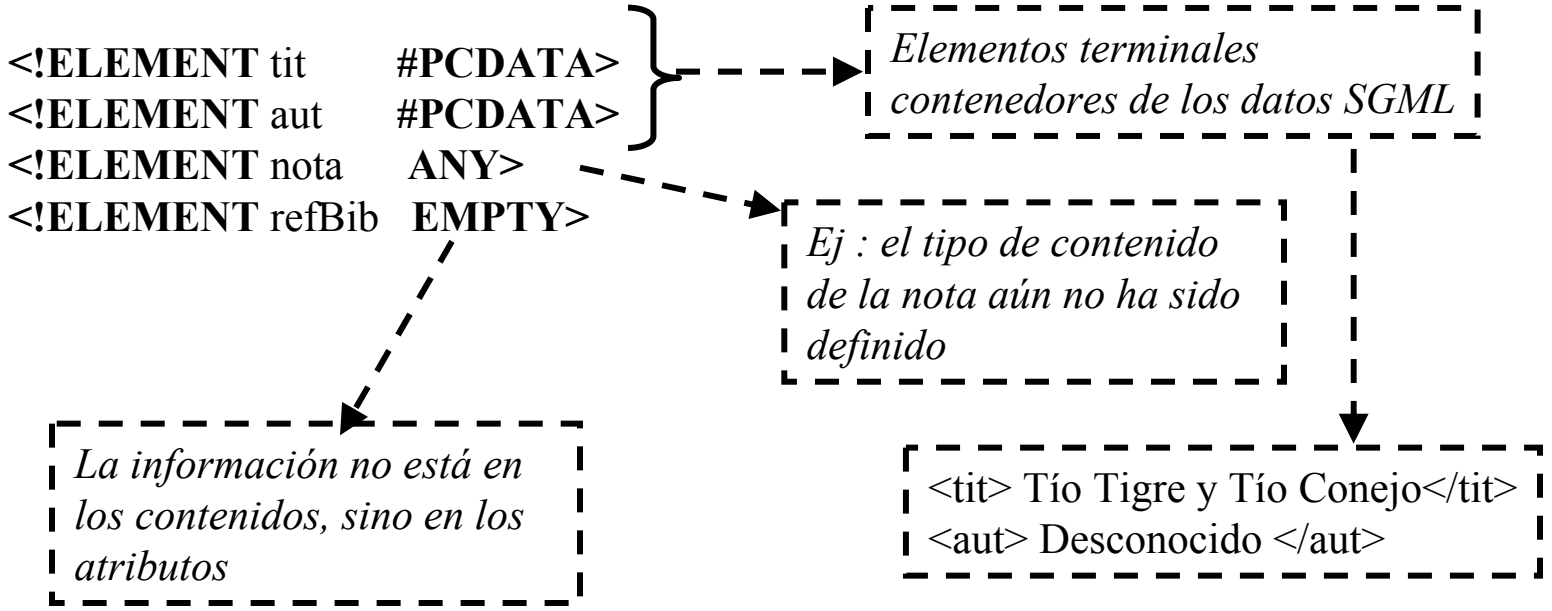
contenido-declarado: **#PCDATA | EMPTY | ANY**

Parsed Character DATA

UNIVERSIDAD
DE LOS ANDES

Elemento …

➔ **EMPTY** : elemento vacío (*solo la marca inicial aparece : <marca/>)*
➔ **#PCDATA** : texto analizable *(puede contener marcas, no tiene elementos hijos)*
➔ **ANY** : cualquier contenido *(generalmente aún no definido)*



UNIVERSIDAD
DE LOS ANDES

Elemento

contenido-compuesto: **(** ,
{ {nom-elemento , ind-occ? }, {connect, nom-element , ind-occ? } }|*
{ {contenido-compuesto, ind-occ?} , {connect, contenido-compuesto, ind-occ?}}* **,)**

<!ELEMENT linea **(**frase+**)>**
<!ELEMENT encabezado **(**tit-doc, ss-tit**?**, autor+, resumen**)>**
<!ELEMENT parrafo **(**titulo, linea+, **(**linea | figura**)*)>**

contenido-mixto: **(,**
{ {contenido-declarado, ind-occ?} , {connect , contenido-compuesto, ind-occ?}} |*
{ {modelo-contenido, ind-occ?}, {connect, modelo-contenido, ind-occ}}* **,)**

<!ELEMENT parrafo **(**titulo, **(#PCDATA** | enfasis**)+** | linea+ | parrafo+**)>**
<!ELEMENT linea (#PCDATA | lista-ord+ | enfasis**)+>**

UNIVERSIDAD
DE LOS ANDES

Ejemplo DTD

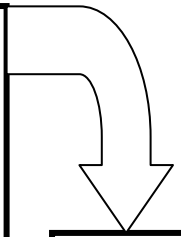
```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<!- - nombre del archivo        -->
<!-- OJO === ISO-8859-1 para español -->
<!DOCTYPE  INVENTARIO [
          <!ELEMENT INVENTARIO  (LIBRO)*>
                    <!ELEMENT LIBRO  (TITULO, AUTOR, EDITORIAL,
                              ANO, ISBN, TITULOINGLES, LUGAR, ESTADO)>
                              <!ELEMENT TITULO  (#PCDATA)>
                              <!ELEMENT AUTOR  (#PCDATA)>
                              <!ELEMENT EDITORIAL  (#PCDATA)>
                              <!ELEMENT ANO  (#PCDATA)>
                              <!ELEMENT ISBN  (#PCDATA)>
                              <!ELEMENT TITULOINGLES (#PCDATA)>
]>
```

UNIVERSIDAD DE LOS ANDES

Atributos …

- Características de un elemento en oposición a las componentes de los elementos
- Pares de nombre-valor dentro de las marcas de inicio y las de elementos vacíos

```
<producto>
        <agotado> si </agotado>
        <fabricante>ACME </fabricante>
        <nombre> tarjeta madre </nombre>
        <desc> Tarjeta principal </desc>
</producto>
```



```
<producto agotado="si">
        <fabricante>ACME </fabricante>
        <nombre> tarjeta madre </nombre>
        <desc> Tarjeta principal </desc>
</producto>
```

UNIVERSIDAD DE LOS ANDES

Atributos …

Los elementos XML pueden poseer ninguno, uno o varios atributos
Ellos contienen la información del tratamiento, que generalmente no se despliega
Los atributos se pueden declarar separados del elemento que los utiliza

definición de la lista de atributos: **<! ATTLIST,** *nom-elemento, nom-atributo, tipo-atributo, valorPorOmi?,* **>**

CDATA	cualquier cadena de caracteres entre ‘ ’(no ‘]]>’)
ID	identificador o clave de elemento XML
IDREF | IDREFS	referencia(s) a uno (o varios) ID
ENTITY | ENTITIES	referencia(s) a una (o varias) entidades
NMTOKEN | NMTOKENS	nombre(s) simbólico(s) cualquiera *('privado','público',...)*
NOTATION	notación utilisada para una entidad no XML *('JPEG ...)*

tipo-atributo=‘CDATA’ | (‘ID’ | ‘IDREF’ | ‘IDREFS’ | ‘ENTITY’ | ‘ENTITIES’ | ‘NMTOKEN’ | ‘NMTOKENS’) | (‘NOTATION’ (nombreNotacion | nombreNota1*) | (token | token1*))

UNIVERSIDAD DE LOS ANDES

DTD ...

valorPorOmi	valor por omisión del atributo
#FIXED *'v'*	el atributo tiene un solo valor *'v'*,
#REQUIRED	el valor debe ser colocado ya que se requiere de él
#IMPLIED	atributo implicado

Ejemplos

```
<!ATTLIST artículo_ley lengua CDATA #FIXED 'Español'>
```

```
<!ELEMENT fecha (#PCDATA)>
<!ATTLIST fecha formato (ANSI | ISO | EN-exp | ES-exp) #REQUIRED >
.....
<fecha formato = "ISO">2000-04-01</fecha>
```

```
<!ELEMENT seccion  (#PCDATA | xref )*>
<!ATTLIST seccion ident ID #IMPLIED >
<!ELEMENT xref  EMPTY>
<!ATTLIST xref ref IDREF #REQUIRED >
.....
<seccion ident = S425> tr : bli bli ... contenido sección tr .... blo blo </seccion>
<seccion > bla bla ... ver seccion 'tr' <xref ref = 'S425'> .... blu blu </seccion>
```

UNIVERSIDAD
DE LOS ANDES

Ejemplo 1

```
<!- - DTD simple para una lista de películas - ->
<!ELEMENT peliculas (pelicula*)>
        <!ELEMENT pelicula ( titulo, director+)>
        <!ATTLIST pelicula
                ident     ID        #REQUIRED
                censura  (A | B | C | D)     'B'>
                <!ELEMENT titulo (#PCDATA)>
                <!ELEMENT director (nombre, inicial?, apellido)>
                        <!ELEMENT nombre (#PCDATA)>
                        <!ELEMENT inicial (#PCDATA)>
                        <!ELEMENT apellido (#PCDATA)>
```

No existe forma de especificar números decimales o cierto tipo de cadenas

UNIVERSIDAD DE LOS ANDES

Ejemplo 2

```
…. DTD
<!ELEMENT artista (nombre, pais, influencia*)>
<!ATTLIST artista                ident        ID          #REQUIRED>
…
<!ELEMENT influencia EMPTY>
<!ATTLIST influencia   de        IDREF       #REQUIRED>
```

```
<?xml version="1.0">
<!DOCTYPE listaDeArtistas SYSTEM "artistas.dtd" >
<listaDeArtistas>
        <artista ident="soto" >
                <nombre> Jesús Rafael Soto </nombre>
                <pais> Venezuela </pais>
        </artista>
         <artista ident="cruzdiez">
                <nombre> Carlos Cruz Díez </nombre>
                <pais> Venezuela </pais>
                <influencia de="soto"/>
        </artista>
</listaDeArtistas>
```

UNIVERSIDAD DE LOS ANDES

Tokens

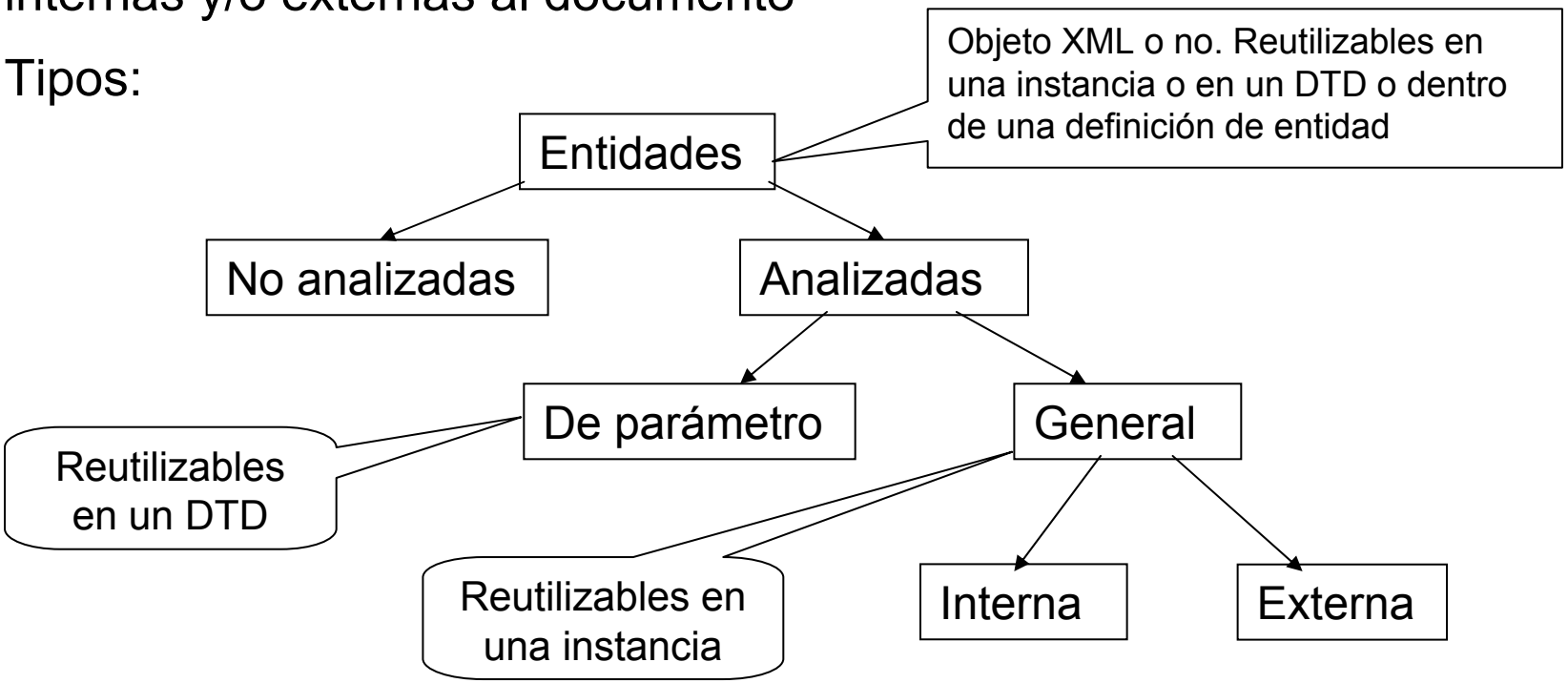
- NMTOKEN y NMTOKENS: para atributos cuyos valores son nombres de tokens (cadenas de caracteres **sin espacios incluidos**)
- NMTOKENS: para una lista de tokens separados por blancos

```
<!ELEMENT fechaDeNacimiento EMPTY>
        <!ATTLIST fechaDeNacimiento
                año         NMTOKEN         #REQUIRED>
<!ELEMENT residencia EMPTY>
        <!ATTLIST residencia
                dir         NMTOKENS        #REQUIRED>
----
<fechaDeNacimiento año="1990"/>
<residencia dir="Av. 5. Nro. 19-18. Mérida"/>
```

UNIVERSIDAD DE LOS ANDES

Entidades

- ENTITY y ENTITIES: para crear referencias a informaciones internas y/o externas al documento
- Tipos:



UNIVERSIDAD DE LOS ANDES

Entidades externas

- Entidades generales **externas**: fragmentos de texto que se pueden reemplazar **dentro del documento**
- Función: servir de mecanismo de creación de módulos
- Uso: se referencian en cualquier parte del documento por su nombre antecedido con &

Declaración:

<!ENTITY nombreDeLaEntidad {**SYSTEM** | **PUBLIC**} **“**identEntidad**”>**

PUBLIC para referencias a archivos o URLs de dominio público

```
<!ENTITY productos SYSTEM “productos.txt”>
----
<?xml version=“1.0”?>
<!DOCTYPE producto SYSTEM “producto.dtd”>
<producto>
        &productos;
</producto>
```

UNIVERSIDAD DE LOS ANDES

Entidades externas

Entidades no XML: siempre deben ser externas

declaración : **<!NOTATION** *nombreDeNotación* **SYSTEM** *URL* >

Declaración de la entidad: **<!ENTITY** *nombreDeEntidad* **SYSTEM** *URL*
NDATA *nombreDeNotación* >

La referencia a una entidad no XML se realiza con un atributo de tipo ENTITY asociado a un elemento vacío

dentro del DTD	..... **<!NOTATION** jpeg **SYSTEM** ".....\programas\displayjpeg.exe" **>** **<!ELEMENT** acceso **EMPTY>** **<ATTLIST** acceso imagen **ENTITY #REQUIRED>**
dentro del DTD	..... **<!ENTITY** imagAcceso **SYSTEM** "....\dic\acceso.jpg" **NDATA** jpeg **>** .....
dentro de la instancia	..... <acceso imagen= 'imagAcceso' /> .....

UNIVERSIDAD DE LOS ANDES

Entidades internas

- Entidades generales **internas**: fragmentos de texto que se pueden reemplazar **dentro del documento** y cuya definición está dentro del DTD
- Uso: mejorar la capacidad modular y de reutilización

Declaración:

<!ENTITY nombreDeLaEntidad **"**textoDeReemplazo**">**

```
<?xml version="1.0"?>
<!DOCTYPE inventario [
        <!ELEMENT inventario (libro)*>
        <!ENTITY editorialMG "McGraw-Hill">
        …… ]>
<inventario>
        <libro>
                <titulo> Aprenda XML ya</titulo>
                <autor>Young, Michael J.</autor>
                <editorial> &editorialMG; </editorial>
                <año>2000</año>
                <ISBN>0-7356-1020-7</ISBN>
                <tituloIngles>XML Step-by-step</tituloIngles>
</libro>
</inventario>
```

UNIVERSIDAD
DE LOS ANDES

Entidades internas

dentro del DTD	**<!ENTITY R "**Región **" >** **<!ENTITY CA "**Coordillera Andina**" >** **<!ENTITY RCA "&**R; **&**CA;**" >**
dentro de la instancia	**<parr>** La **&RCA;** está en el Occidente del país ...</parr>
Texto generado	La Región Coordillera Andina está en el Occidente del país .....

UNIVERSIDAD DE LOS ANDES

Entidades de parámetro

- Parámetro: reutilización de fragmentos de texto **dentro del DTD**
- Uso: reciclar fragmentos del modelo de contenido. Sólo se pueden usar dentro del DTD

☞ *Una entidad de parámetro puede hacer referencia a otras entidades de parámetro sin límites, pero los ciclos están prohibidos*

Declaración interna: ojo

<!ENTITY % nombreDeLaEntidad “textoDeReemplazo”**>**

EJEMPLO

```
<!ENTITY % atMeses “(1|2|3|4|5|6|7|8|9|10|11|12|Ene|Feb|Mar|Abr|May|Jun|Jul|Ago|Sep|Oct|Nov|Dic)”>
<!ELEMENT peliculas (pelicula*)>
        <!ELEMENT pelicula ( titulo, fecha, director+)>
        <!ATTLIST pelicula
                ident       ID          #REQUIRED
                censura     (A | B | C | D)‘B’>
                <!ELEMENT titulo (#PCDATA)>
                <!ELEMENT fecha EMPTY>
                <!ATTLIST   fecha
                            %atMeses;
                            año         NMTOKEN   #REQUIRED>
                <!ELEMENT director (nombre, inicial?, apellido)>
```

UNIVERSIDAD DE LOS ANDES

Entidades de parámetro

- Externas: el uso es igual a las internas, pero se definen dentro de un recurso externo o archivo XML

Declaración externa:
<!ENTITY % nombreDeLaEntidad **SYSTEM** "URL"**>**

URL designa el archivo donde se encuentra la declaración de la entidad
Referencia: igual que las anteriores
%nombreDeLaEntidad**;**

<!ENTITY % reporte **SYSTEM** ' ....\5IF\TP\entidades_IF.xml' >

UNIVERSIDAD DE LOS ANDES

DTD

- Internos: el DOCTYPE está dentro del documento .xml
- Externos: el DOCTYPE está en un archivo separado .dtd que debe ser incluido dentro del documento .xml
- Secciones condicionales: permiten incluir o ignorar fragmentos del DTD al encerrarlos en una marcación especial

Declaración de secciones condicionales:
<![INCLUDE[sección de DTD incluida **]]>**
<![IGNORE[sección de DTD excluida **]]>**

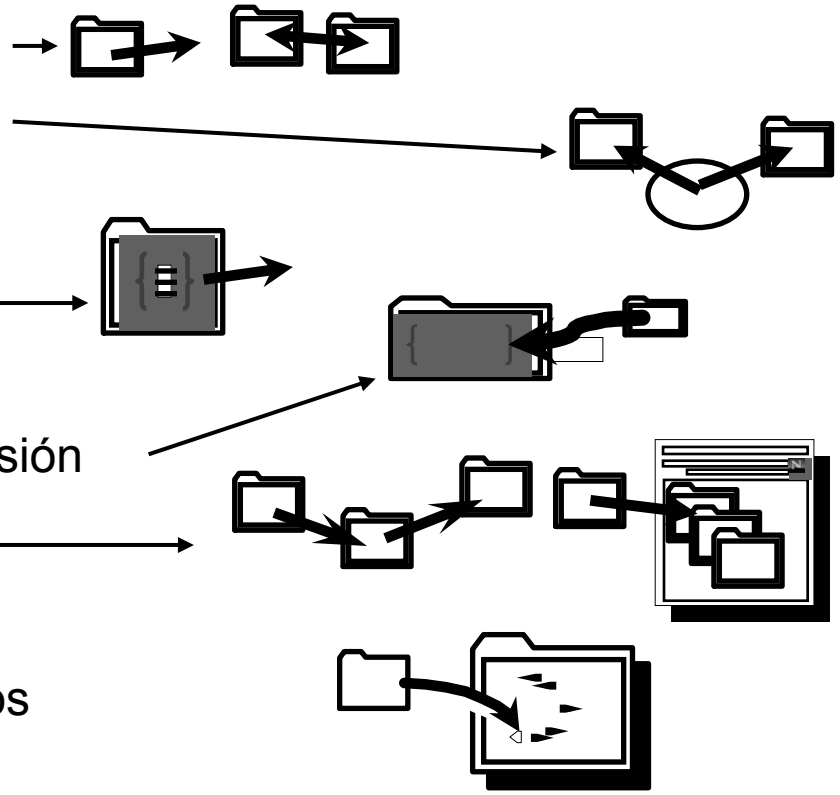
EJEMPLO

<!ENTITY % borrador **"INCLUDE">**
<!ENTITY % final **"IGNORE">**
<![%borrador**;[<!ELEMENT** libro **(** comentario*, titulo, cuerpo, anexos**?)>]]>**
<![%final**;[<!ELEMENT** libro **(** titulo, cuerpo, anexos**?)>]]>**

UNIVERSIDAD DE LOS ANDES

XLink …

- Enlaces en XML, tomado en gran parte de HyTime
  - ✓ Uni y bi-direccionales
  - ✓ Hacia otro documento
  - ✓ De tipo agregado
  - ✓ De transferencia e inclusión
  - ✓ N-arios
  - ✓ Dinámicamente anclados



UNIVERSIDAD DE LOS ANDES

XLink …

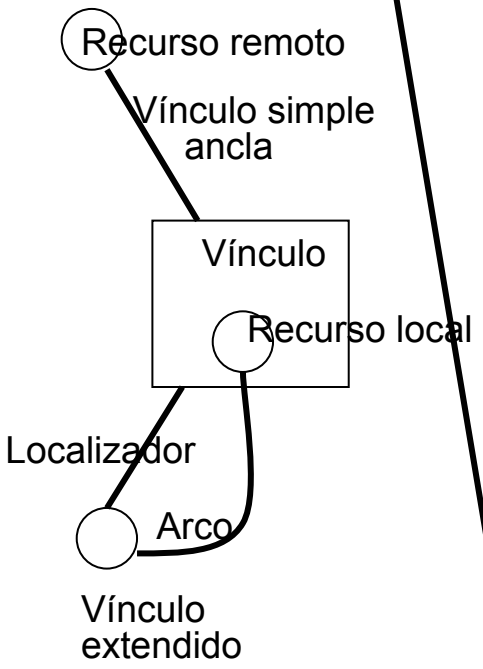
- Elementos de vinculación: marcas de elementos arbitrarios para unir cualquier cantidad de recursos locales y remotos
- **Objetivo:** creación de vínculos XML que tengan significado y que se puedan usar de maneras complejas
- **Recurso:** cualquier cosa que se pueda direccionar en la red incluyendo información XML interna en el recurso
- **Localizadores:** elementos que señalan recursos externos
- **Arcos:** relaciones entre recursos

```
<ejemplo xlink:type='extended'>
<termino xlink:type='resource' xlink:role='padres'> Padres </termino>
<definicion xlink:type='locator' xlink:role='padresDeMozart' xlink:href='padresMozart.xml'>
<asignacion xlink:type='arc' xlink:from='padres' xlink:to='padresDeMozart'/>
</ejemplo>
```

UNIVERSIDAD DE LOS ANDES

```
<!ELEMENT datosArco EMPTY >
<!ATTLIST  datosArco
            xlink:type  (arc)  #FIXED 'arc'
            xlink:title   CDATA  #IMPLIED
            xlink:show  (new | replace | embed | other | none)  #IMPLIED
            xlink:from    NMTOKEN    #IMPLIED
            xlink:to      NMTOKEN    #IMPLIED >
```

XLink ...



Elemento de vinculación: XLink proporciona *atributos globales* que marquen *cualquier* elemento

Vínculo extendido: (el más importante) está marcado con el tipo *extended* y puede contener:

- Cero o más recursos locales (*resource*)
- Cero o más localizadores que apuntan a recursos remotos (*locator*)
- Cero o más arcos que unen recursos según sus roles (*arc*)
- Un título opcional

Recursos locales: debe tener *role* (naturaleza del vínculo) y *title* (descripción para el usuario)

Recursos remotos: no son vínculos, son apuntadores a recursos remotos, deben tener *href* y pueden tener *role* y *title*

Arcos: son las reglas de recorrido desde el recurso marcado como desde (*from*) hasta (*to*). El valor del atributo debe ser el nombre del rol del vínculo extendido. No deben haber arcos duplicados. Atributos adicionales llamados de comportamiento.

UNIVERSIDAD
DE LOS ANDES

XLink …

Atributos de comportamiento:

xlink:show ⇔determina la presentación deseada del recurso final, con los valores:

- new: abre una nueva ventana y muestra el recurso
- replace: carga el recurso en la misma ventana
- embed: inserta el recurso, por ejemplo un sonido
- none: depende de la aplicación
- other: presentación no restringida por XLink

xlink:actuate ⇔determina el momento del recorrido hacia el recurso final, con los valores:

- onLoad: carga el recurso final cuando encuentra el recurso de inicio
- onRequest: carga el recurso final cuando se hace click en el vínculo
- none: depende de la aplicación
- other: presentación no restringida por XLink

Título: tiene una función doble, semántica y de presentación, se usa para colocar títulos complejos que no se pueden tener de valores de atributos

☞ **Se recomienda evitar la saturación de código colocando los elementos de XLink en los DTD**


```
<!ancla      xlink:type ='simple'
             xlink:href='pagina1.html'> comentario del ancla
</ancla>
```

XLink

- **Vínculos simples**: tiene la funcionalidad de un recurso, un localizador y un arco. Similares a los vínculos HTML.
- Soporte XLink en los navegadores es limitado
- Herramienta XLink2HTML XSLT de conversión

```
<!ELEMENT algo       (#PCDATA)>
<!ATTLIST algo xmlns:xlink CDATA #FIXED  "http://www.w3.org/1999/xlink"
               xlink:type              (resource) #FIXED  'resource'
               xlink:title             CDATA      #IMPLIED
               xlink:role              NMTOKEN              'algomas' >
<!ELEMENT            datos (foo, bar) >
<!ATTLIST  datos
               xlink:type               (locator)  #FIXED  'locator'
               xlink:href              CDATA      #REQUIRED
               xlink:title             CDATA      #IMPLIED
               xlink:role              NMTOKEN              'datosAlgomas' >
```

UNIVERSIDAD DE LOS ANDES

XPointer …

- XML Pointer Language
- Estándar para identificadores de fragmentos en referencias URI
  - ✓ Limitaciones del mecanismo en HTML:
    - Necesidad de marcar el punto dentro del documento antes de apuntar a él
    - Sólo se puede marcar un punto y no un rango
    - Sólo se puede hacer con un ancla que no depende de la estructura del documento
  - ✓ Especificación para XML:
    - Apuntar a rangos completos y a nodos específicos
    - Evitar el uso de marcas para señalar el fragmento en el destino
    - Sintaxis para incluir expresiones del lenguaje en un URI

UNIVERSIDAD DE LOS ANDES

XPointer …

- Conceptos básicos:
  - ✓ Recurso secundario: elemento que se encuentra dentro del documento
  - ✓ Punto: fragmento o posición antes de un carácter específico
  - ✓ Rango: identificación de una selección contigua que se encuentra entre dos puntos determinados
  - ✓ Conjunto de ubicación: lista ordenada de nodos, puntos y rangos de documentos
- Selecciona partes de un documento XML mediante una expresión
- Mantiene la estructura: eje, prueba de nodo y conjunto opcional de XPath
- Cualquier expresión debe tener los paréntesis equilibrados, sino se equilibran con ^

UNIVERSIDAD DE LOS ANDES

XPointer …

- Crea los enlaces hacia cualquier posición en el documento
- Posición absoluta: elemento cuyo id = t123
- Posición relativa: la 5ta frase del párrafo que contiene ‘tacata’ bajo título
- Constitución:
  - Primer término: reenvío hacia una posición absoluta
    - Raíz del documento destino (URL)
    - Elemento de un documento
      - Elemento que contiene un apuntador: destino identificado con **origin()**
      - Elemento distinguido con un identificador: **#id(t23)**
    - Ancla con nombre
  - Seguido de una cascada de reenvíos relativos:
    - reenvío hacia un nodo definido por su posición en el árbol

UNIVERSIDAD DE LOS ANDES

XPointer …

- **child: root().child(**2,capitulo**)** ⇔ 2do capítulo del documento
- **descendant: id(**cap3**).descendant(**5, par**)** ⇔ 5ta ocurrencia del elemento par en cualquier nivel dentro del elemento identificado con cap3
- **ancestor: descendant(**3, nom**).ancestor(**1, dir**)** ⇔ elemento dir que contiene la 3era ocurrencia del elemento nom
- **psibling** o **fsibling**: nodo hermano precedente o siguiente
- **preceding** o **following**: nodo precedente o siguiente

- Reenvío sobre el valor de un atributo
  - **child(**10, dir**).attr(**ciudad**)** ⇔ valor del atributo ciudad en la 10ma dir
- Reenvío sobre una cadena de caracteres
  - **string (nroDeOcurrencia, cadenaBuscada, posición, longitud)** ⇔ **string(**1, **"**Caracas**"**, 3, 2**):** selecciona "ra" en la 1era ocurrencia de Caracas
- Reenvío sobre un intervalo delimitado por dos reenvíos
  - **span(ter1-ter2)** ⇔ **id(**t25**).span(child(**2**), child(**4**)):** los hijos 2,3 y 4 del elemento t25

UNIVERSIDAD DE LOS ANDES

XPointer

✓ Un nombre se trata como argumento de la función **id()**
Pepe **xpointer(id(“Pepe”))**
✓ secuencia secundaria
/1/2 **root().child(**1**).child(**2**)**
✓ **Caracteres de escape:** cualquier caracter reservado de XML se sustituye con su representación en hexadecimal (%HH)
miCaso**.xml#xpointer(id(‘**casa**’))**
%20xpointer(//*%58@id=‘R**%C3’%5D)**
✓ **Punto de nodo:** nodo con elementos secundarios cuyo identificador es un índice a partir de cero.
✓ **Punto de caracter:** nodo sin elementos secundarios donde el índice indica el número del caracter a partir de uno

✓ **Rango:** ambos puntos deben estar en el mismo documento XML
✓ TipoDeNodo::= ‘comentario’ | ‘text’ | ‘node’ | ‘processing-instruction’ | ‘point’ | ‘range’
✓ **Ubicación:** puede ser un nodo, punto o rango
✓ Conjunto de ubicación: conjunto de nodos
✓ **Funciones adicionales** a las de XPath:
- **range-to(**expresión**):conjUbica**
- **string-range(conjUbic,** cadena**,** número**?,** número1**?):conjUbica**
  string-range(//pelea**/**campeon,’Ali’,1,2**)**
- **range(conjUbica):conjUbica**
- **range-inside(conjUbica):conjUbica**
- **start-point(conjUbica):conjUbica**
- **here():conjUbica**
- **unique():booleano**

UNIVERSIDAD DE LOS ANDES

¿Cómo definir las DTD?

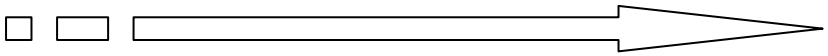
- Reglas clásicas:
  - ✓ Modularidad: usar entidades parametrizadas para definir dentro las entidades con partes reutilizables
  - ✓ Precedencia: agrupamiento de las declaraciones de entidades para utilizar una entidad solamente luego de su definición
  - ✓ Abstracción: aislamiento de las entidades significativas del mundo real para asociarlas a las entidades parametrizadas y definir los modelos de contenido
  - ✓ Especificidad: evitar las DTD muy generales que no aportan nada en términos de descripción de contenidos
  - ✓ Simplicidad: tratar que las DTD queden simples dividiendo las DTD complejas

UNIVERSIDAD DE LOS ANDES

Método basado en UML

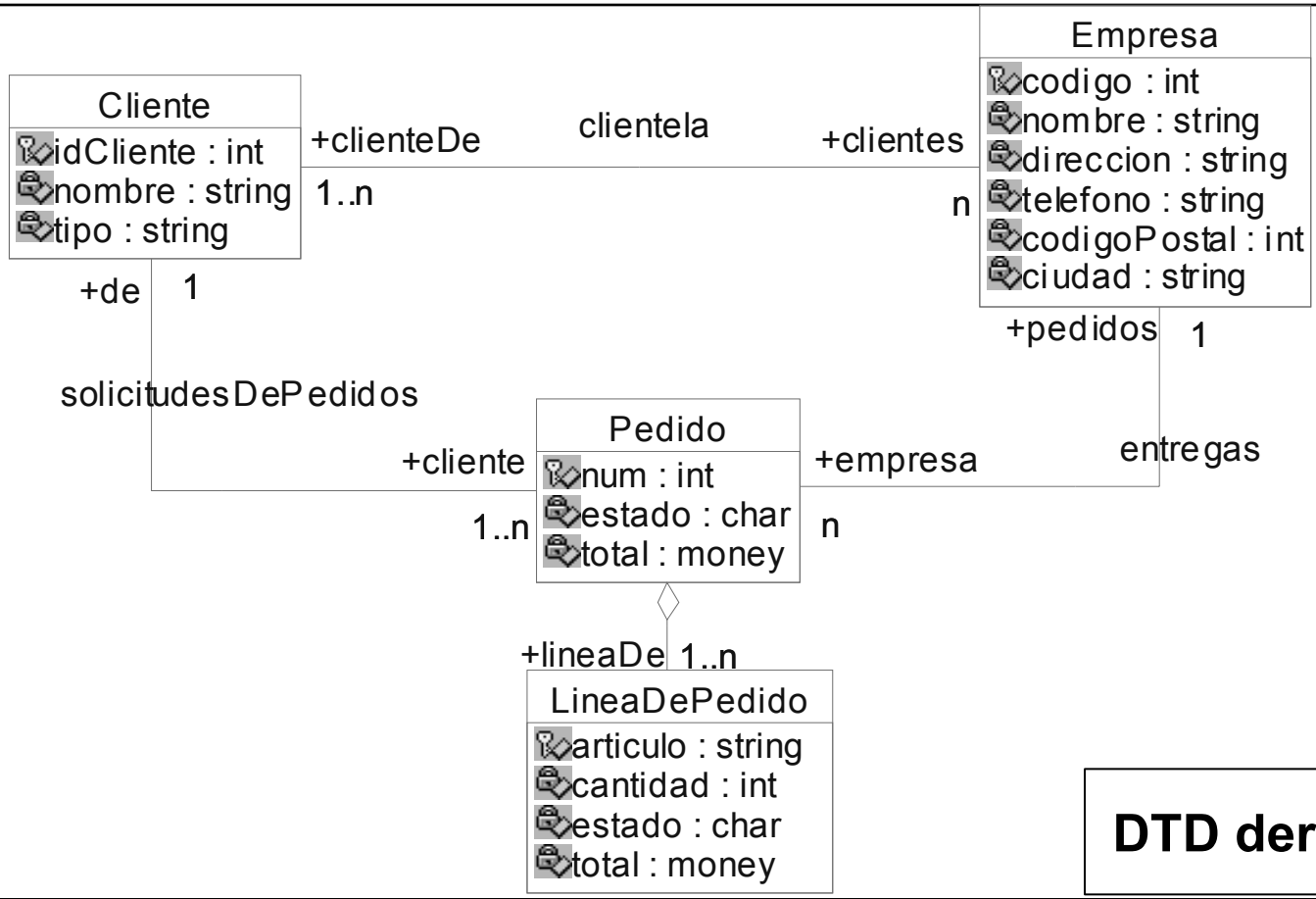
- Representar cada tipo de dato básico (int, char, float, etc.) como una entidad
- Representar cada clase como una entidad XML, cuyos atributos generales son atributos XML o elementos XML
- Escoger el orden de anidamiento jerárquico de las clases y generar el elemento raíz agregando las entidades básicas en el orden correcto
- Con el fin de no perder las relaciones, es posible agregar hiperenlaces XML

Ejemplo



UNIVERSIDAD DE LOS ANDES

Ejemplo de modelo UML



DTD derivada

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE umlEmpresa [
        <!-- Tipos basicos -->
<!ENTITY % int " (#PCDATA) " >
<!ENTITY % char " (#PCDATA) " >
<!ENTITY % string " (#PCDATA) " >
<!ENTITY % money " (#PCDATA) " >
        <!--        Clase Pedido -->
<!ENTITY % Pedido "
        <!ELEMENT Pedido (estadoP, totalP)>
                <!ATTLIST Pedido num ID #REQUIRED >
        <!ELEMENT       estadoP %char; >
        <!ELEMENT       totalP %money; > " >
        <!--        Clase LineaDePedido             -->
<!ENTITY % LineaDePedido "
        <!ELEMENT LineaDePedido(articulo, cantidad, estadoL?, totalL?) >
        <!ELEMENT articulo %string; >
        <!ELEMENT cantidad %int; >
        <!ELEMENT estadoL %char; >
        <!ELEMENT totalL %money; > "        >
```

```
        <!-- Definicion de umlEmpresa -->
<!ELEMENT umlEmpresa (Pedido, LineaDePedido+)*>
%Pedido;
%LineaDePedido;
]>
```

UNIVERSIDAD DE LOS ANDES

Limitaciones de las DTD

- No ofrecen tipos de datos simples a excepción del texto
- Son difíciles de interpretar e implican bastantes conversiones de tipo
- Son difíciles de traducir a esquemas orientados por objetos y viceversa
- No se definen en XML sino en su propio lenguaje

Las DTD definen solamente la gramática de las marcas y no atacan el problema de definir un esquema para el documento

UNIVERSIDAD DE LOS ANDES

Esquemas XML …

- Estandarizados en 2001 (XML Schema 1.0)
- Tipos simples:
  - ✓ Cadenas de caracteres:
    - string, normalisedString (sin return, tabs y espacios normalizados) y token (palabra que debe pertenecer a una lista predefinida)
  - ✓ Bytes y lógicos
    - byte (byte con signo), unsignedByte y boolean (True, False, 1, 0)
  - ✓ Binario
    - base64Binary y hexBinary (hexadecimal)
  - ✓ Enteros
    - integer [-126789,126789], positiveInteger [1, 212], negativeInteger [-1, -212], nonNegativeInteger [0, 212], nonPositiveInteger [-212, 0], int [-1, 21256754], unsignedInt [0, 1267896754], long [-1, 326789875432355], unsignedLong [0, 12678967543233], short [1, 12678], unsignedShort [0, 12678]

UNIVERSIDAD DE LOS ANDES

Esquemas XML …

- Reales:
  - decimal [-1.23, 1000.00], float y double
- Fechas:
  - time (milisegundos), dateTime (fechaThora), duration (añomesdíaThoraminutossegundos), date (formato US), gMonth (número del mes), gYear (año), gYearMonth (año-mes), gDay (día), gMonthDay (mes-día)
- Nombres y URI:
  - Name (marca), QName (espacio de nombres:marca) y anyURI
- Identificadores y referencias simples y múltiples:
  - ID, IDREF y IDREFS
- Tipos para las DTD:
  - ENTITY, ENTITIES, NOTATION, NMTOKEN y NMTOKENS

UNIVERSIDAD DE LOS ANDES

Esquemas XML …

- Tipos complejos: tipos compuestos construidos con secuencias, escogencias o todos
  - ✓ **<xsd:sequence>** colección ordenada de elementos con tipo
  - ✓ **<xsd:all>** colección desordenada de elementos con tipo
  - ✓ **<xsd:choice>** escogencia entre elementos con tipo

```
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<xsd:complexType  name="Direccion">
        <xsd:sequence>
                <xsd:element name="num" type="xsd:string"/>
                <xsd:element name="calle" type="xsd:string"/>
                <xsd:element name="ciudad" type="xsd:string"/>
        </xsd:sequence>
</xsd:complexType>
```

UNIVERSIDAD DE LOS ANDES

Esquemas XML …

- Cláusulas:
  - ✓ **<xsd:element>** asocia un tipo a un elemento. Atributos: name, type, ref, minOccurs, maxOccurs
  - ✓ **<xsd:attribute>** asocia un tipo a un atributo. Atributos: name y type
  - ✓ **<xsd:simpleType>** especializa un tipo básico escogido entre varios tipos básicos
  - ✓ **<xsd:complexType>** define un tipo complejo como una secuencia, alternativa o conjunto de tipos.
- Especialización por restricción: especializa un tipo simple o complejo con una restricción.

```
<xsd:restriction base="xsd:string">
<xsd:maxLength value="32"/>
</xsd:restriction>
```

```
<xsd:restriction base="xsd:string">
<xsd:pattern value="\d{3}-[A-Z]{2}"/>
</xsd:restriction>
```

UNIVERSIDAD DE LOS ANDES

Esquemas XML …

- Derivación de tipos por extensión: agregar al tipo complejo otro elemento
- Prohibición de derivación: colocar el atributo final a extension con el valor restriction, extension o #all

```
<complexType name="DireccionPais">
  <complexContent>
    <extension base="Direccion">
        <sequence>
            <element name="pais" type="string"/>
        </sequence>
    </extension>
  </complexContent>
</complexType>
```

Herramientas gráficas de generación de esquemas

XMLSpy
XML Authority
EditML Pro
eWebEditPro
Xeena (IBM)
XMetal (SoftQuad)

UNIVERSIDAD DE LOS ANDES

Esquemas XML …

- Reutilización de las declaraciones: **<element ref=“**nombre**”/>**
- Contenidos mixtos: Mixed Content mezcla de tipos
- Contenidos vacíos: Empty Content
- AnyType cualquier tipo aún no se especifica
- Attribute Groups definición de grupos de atributos
- Nil Values hacer explícito un valor nulo, también con nillable=“true”
- Specifying Uniqueness especificación de unicidad en una colección
- Defining Keys & their References definición de claves y sus restricciones referenciales
- Namespaces, Schemas & Qualification definición de espacios de nombres, calificación de los elementos y asociación de esquemas con espacios de nombres
- Abstract Elements & Types definición de elementos abstractos que serán luego reemplazados por elementos derivados de las instancias

UNIVERSIDAD DE LOS ANDES

Ejemplo de esquema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<xsd:element name="nombreC" type="xsd:string"/>
<xsd:element name="tipo" type="xsd:string"/>
<xsd:element name="nombreE" type="xsd:string"/>
<xsd:element name="direccion" type="xsd:string"/>
<xsd:element name="telefono" type="xsd:string"/>
<xsd:element name="codigoPostal" type="xsd:unsignedShort"/>
<xsd:element name="ciudad" type="xsd:string"/>
<xsd:element name="estadoP" type="xsd:unsignedByte"/>
<xsd:element name="totalP" type="xsd:float"/>
<xsd:element name="cantidad" type="xsd:unsignedShort"/>
<xsd:element name="estadoL" type="xsd:unsignedByte"/>
<xsd:element name="totalL" type="xsd:float"/>
```

UNIVERSIDAD DE LOS ANDES

Ejemplo de esquema

```
<xsd:element name="Cliente">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="nombreC" />
      <xsd:element ref="tipo" />
    </xsd:sequence>
    <xsd:attribute name="idCliente" type="xsd:unsignedInt" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Pedido">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="estadoP" />
      <xsd:element ref="totalP" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="num" type="xsd:unsignedInt" use="required"/>
  </xsd:complexType>
</xsd:element>
```

UNIVERSIDAD DE LOS ANDES

Ejemplo de esquema

```
<xsd:element name="Empresa">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="nombreE" />
      <xsd:element ref="direccion" />
      <xsd:element ref="telefono" />
      <xsd:element ref="codigoPostal" minOccurs="1000"/>
      <xsd:element ref="ciudad" />
    </xsd:sequence>
    <xsd:attribute name="codigo" type="xsd:unsignedInt" use="required"/>
  </xsd:complexType>
</xsd:element>
```

UNIVERSIDAD DE LOS ANDES

Ejemplo de esquema

```
<xsd:element name="LineaDePedido">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="cantidad" minOccurs="1" />
      <xsd:element ref="estadoL" />
      <xsd:element ref="totalL" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="articulo" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="articulos">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="LineaDePedido" maxOccurs="unbounded" />
    <xsd:sequence>
  <xsd:complexType>
<xsd:element>
```

Etc….. Completar