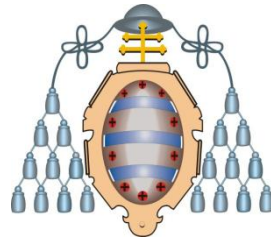
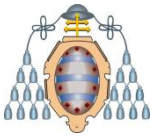


# *DTDs*

## *Declaraciones de tipo de documentos*



Departamento de Informática  
Universidad de Oviedo



## Documento válido

Se puede incluir una declaración del tipo de documento

```
<?xml version="1.0"?>
<!DOCTYPE poema SYSTEM "poema.dtd">
<poema fecha="Abril de 1915"
      lugar="Granada">

<titulo>Alba</titulo>

<verso>Mi corazón oprimido</verso>
<verso>siente junto a la alborada</verso>
<verso>el dolor de sus amores</verso>
<verso>y el sueño de las distancias. </verso>
</poema>
```

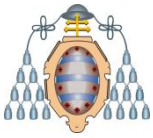
### poema.dtd

```
<!ELEMENT poema (titulo,verso*)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT verso (#PCDATA)>
<!ATTLIST poema fecha CDATA #REQUIRED
              lugar CDATA #IMPLIED>
```

## Documento válido

Está bien formado y

La estructura encaja con la declaración del tipo de documento

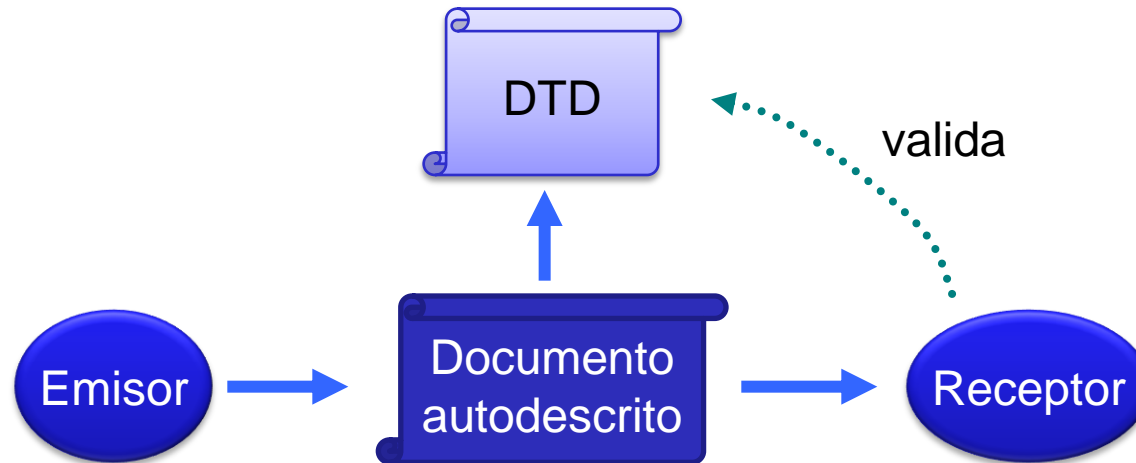


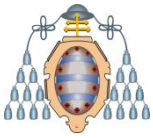
# *Declaración Tipo Documento*

## *DTD*

La DTD permite especificar la estructura del documento

La DTD puede estar separada del documento

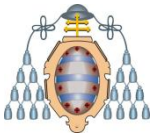




## *DTD interna*

### DTD interna: dentro del mismo documento XML

```
<?xml version="1.0" ?>
<!DOCTYPE pedido [
  <!ELEMENT pedido    (producto*) >
  <!ELEMENT producto  (nombre,cantidad)>
  <!ELEMENT nombre     (#PCDATA)>
  <!ELEMENT cantidad   (#PCDATA)>
]>
<pedido>
  <producto>
    <nombre>Rotulador RX2</nombre>
    <cantidad>20</cantidad>
  </producto>
  <producto>
    <nombre>Grapadora Lin</nombre>
    <cantidad>2</cantidad>
  </producto>
</pedido>
```

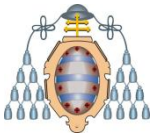


## DTD externa: Haciendo referencia a una URI

```
<?xml version="1.0" ?>
<!DOCTYPE pedido SYSTEM "http://www.ej.org/pedidos.dtd">
<pedido>
  <producto>
    <nombre>Rotulador RX2</nombre>
    <cantidad>20</cantidad>
  </producto>
  <producto>
    <nombre>Grapadora Lin</nombre>
    <cantidad>2</cantidad>
  </producto>
</pedido>
```

<http://www.ej.org/pedidos.dtd>

```
<!ELEMENT pedido      (producto*) >
<!ELEMENT producto    (nombre,cantidad)>
<!ELEMENT nombre      (#PCDATA)>
<!ELEMENT cantidad    (#PCDATA)>
```



Hace referencia a un identificador público

Un procesador puede disponer de un catálogo de identificadores públicos

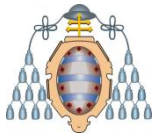
Liberaría de buscar en Internet el DTD para la validación

```
<?xml version="1.0"?>
<!DOCTYPE HTML PUBLIC
    "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3c.org/TR/REC-html/strict.dtd">
<HTML>
  <HEAD><TITLE>Ejemplo HTML</TITLE></HEAD>
  <BODY>
    <H1>HTML de ejemplo</H1>
  </BODY>
</HTML>
```



Es posible añadir nuevas definiciones a la validación externa

```
<?xml version="1.0" ?>
<!DOCTYPE pedido SYSTEM "http://www.ej.org/pedidos.dtd" [
  <!-- ATTLIST producto id CDATA #IMPLIED -->
]>
<pedido>
  <producto>
    <nombre>Rotulador RX2</nombre>
    <cantidad>20</cantidad>
  </producto>
  <producto>
    <nombre>&grapa;</nombre>
    <cantidad>2</cantidad>
  </producto>
</pedido>
```



# *Tipos de declaraciones*

## ELEMENT

Elementos del documento XML

## ATTLIST

Lista de atributos de un elemento

## ENTITY

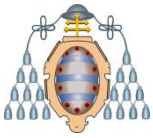
Entidades ( $\approx$ variables o macros)

## NOTATION

Definen tipos de contenidos

Facilitan la inclusión de formatos binarios (imágenes, vídeos, sonidos, ...)



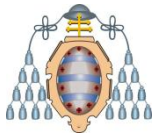


# Contenido de Elementos

(?) = 0, 1 elemento  
(\*) = 0 ó más elementos  
(+) = 1 ó más elementos  
(|) = alternativa  
(,) = secuencia  
EMPTY = vacío  
ANY = cualquier estructura de subelementos  
#PCDATA = cadena de caracteres analizados

```
<!ELEMENT pizza (ingrediente*, inventor?)>  
<!ELEMENT servicio (domicilio | restaurante) >  
<!ELEMENT ingrediente EMPTY>  
<!ELEMENT inventor (#PCDATA)>  
<!ELEMENT sección (título, (contenido | sección+))>
```

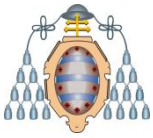
Recursividad



## Ejercicio

Definir una DTD para validar el siguiente documento

```
<libro>
<titulo>El Teclado</titulo>
<autor>Eduardo Santos</autor>
<seccion>
  <nombre>Introduccion</nombre>
  <contenido>
    <texto>Esta es la Introduccion</texto>
  </contenido>
</seccion>
<seccion>
  <nombre>Primer tomo</nombre>
  <contenido>
    <seccion><nombre>S1</nombre></seccion>
    <seccion><nombre>S2</nombre></seccion>
  </contenido>
</seccion>
</libro>
```



## Contenido mixto

Contenido que puede mezclar texto con elementos

Ejemplo:

**<comentario>**

Este texto está relacionado con el

**<enlace url="http://www.quijote.com">Quijote</enlace>** y ha sido  
realizado por **<persona>Leopoldo Alas Clarín</persona>**

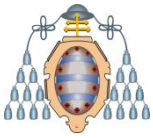
**</comentario>**

**<!ELEMENT comentario (#PCDATA | enlace | persona)\* >**

**<!ELEMENT enlace (#PCDATA)>**

**<!ELEMENT persona (#PCDATA)>**

**<!ATTLIST enlace url CDATA #REQUIRED>**



# Atributos

## Tipos de datos

CDATA = Cadena de caracteres

NMTOKEN = Palabra (sin espacios)

NMTOKENS = Lista de palabras

Enumeración separada por |

## Valor de los Atributos

#REQUIRED Obligatorio

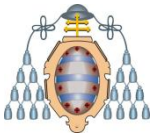
#IMPLIED Opcional

#FIXED Constante

Valor Valor por defecto

```
<!ATTLIST pizza nombre CDATA #REQUIRED>
<!ATTLIST ingrediente nombre CDATA #REQUIRED
                    calorías CDATA #IMPLIED>
<!ATTLIST precio moneda (euros|dólares) #REQUIRED
                    valor CDATA #REQUIRED>
<!ATTLIST conOrégano (sí|no) "sí" >
<!ATTLIST impuesto tipo CDATA #FIXED "IVA">
```

```
<pizza nombre="4 estaciones" >
  <ingrediente nombre="Jamón" />
  <precio moneda="euros" valor="7" />
</pizza>
```



# *Integridad referencial*

ID e IDREF permiten definir integridad referencial básica para valores de atributos

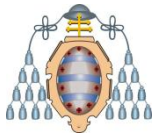
ID = Nombre único (sin duplicados)

No puede empezar por dígito

IDREF = Su valor debe apuntar a un ID ya existente

```
<!ATTLIST persona código ID #REQUIRED>  
<!ATTLIST dueño código IDREF #REQUIRED>
```

```
<persona código="23" nombre ="Juan" />  
<persona código="35" nombre ="Pepe" />  
<persona código="37" nombre ="Luis" />  
  
<dueño código="35" />
```



# Entidades Generales

Entidades: Asignan nombres a ciertos elementos (similar a variables)

Se denotan por &entidad;

No se admite recursividad

```
<!ENTITY marg "Pizza Margarita">  
<!ENTITY queso "<ingrediente nombre='queso' />" >
```

```
<pizza nombre="&marg;" precio="7">  
&queso;  
</pizza>
```

```
<pizza nombre="Pizza Margarita" precio="7">  
<ingrediente nombre='queso' />  
</pizza>
```

Entidades numéricas: Código numérico del carácter

```
&#x2200; ∇ &#8707; ∃
```

Entidades predefinidas: Permiten incluir etiquetas sin analizar

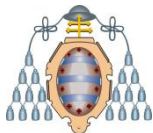
```
&lt; < &quot; “ &apos; ‘  
&gt; > &amp; &
```



## *Ejercicio*

Indicar cuál es el documento  
lógico equivalente a:

```
<!DOCTYPE prueba [  
  <!ENTITY p1 "Hola" >  
  <!ENTITY p2 "&p1; &p1;" >  
  <!ENTITY p3 "&p2; &p2;" >  
  <!ENTITY p4 "&p3; &p3;" >  
  <!ENTITY p5 "&p4; &p4;" >  
  <!ENTITY p6 "&p5; &p5;" >  
  <!ENTITY p7 "&p6; &p6;" >  
  <!ENTITY p8 "&p7; &p7;" >  
  <!ENTITY p9 "&p8; &p8;" >  
  <!ENTITY p10 "&p9; &p9;" >  
  <!ENTITY p11 "&p10; &p10;" >  
  <!ENTITY p12 "&p11; &p11;" >  
  <!ENTITY p13 "&p12; &p12;" >  
  <!ENTITY p14 "&p13; &p13;" >  
  <!ENTITY p15 "&p14; &p14;" >  
>  
<prueba>  
  &p15;  
</prueba>
```



## Entidades externas

Permiten usar archivos externos (Incluir otros documentos XML)

pizzas.xml

```
<pizzas>
<pizza nombre="4 Quesos" precio="7">
  <ingrediente nombre="Jamón" />
  <ingrediente nombre="Queso" />
</pizza>
...
</pizzas>
```

personal.xml

```
<personal>
<trabajador
  nombre=" Benito Alcaparra" >
  ...
</trabajador>
...
</personal>
```

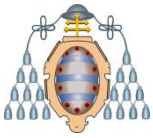
establecimiento.dtd

```
<!ELEMENT establecimiento ANY>
<!ENTITY personal SYSTEM "personal.xml">
<!ENTITY pizzas SYSTEM "pizzas.xml">
```

establecimiento.xml

```
<establecimiento
  nombre="Pizzería Al Capone">
  &personal;
  &pizzas;
</establecimiento>
```

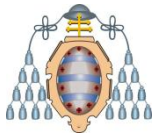




Las entidades externas combinadas con notaciones permiten incluir archivos externos de datos binarios

```
<!NOTATION gif SYSTEM "gifEditor.exe">  
<!ENTITY dibujo SYSTEM "logotipo.gif" NDATA gif>
```

```
<información>  
  <logotipo>&dibujo;</logotipo>  
</información>
```



Permiten dar nombres a partes de un DTD

Se denotan por %entidad;

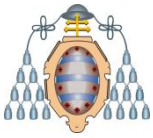
```
<!ENTITY establecimiento (nombre,dueño?,calle,número?,ciudad,país,códigoPostal)>  
<!ENTITY persona (dni, nombre,calle,número?,ciudad,país,códigoPostal)>
```

```
<!ENTITY %localización "calle,número?,ciudad,país,códigoPostal" >  
<!ENTITY establecimiento (nombre,dueño?,%localización;)>  
<!ENTITY persona (dni, nombre, %localización;)>
```

Entidades externas: Permiten incluir elementos externos en una DTD

Aplicación: Dividir la definición de una DTD en varios documentos

```
<!ENTITY %persona SYSTEM "persona.dtd">  
<!ENTITY %establecimiento SYSTEM "establecimiento.dtd">  
%persona;  
%establecimiento;
```



## *Características de DTD's*

Especifican estructura del documento:

Elementos, atributos, anidamientos, etc.

Integridad referencial mínima (ID, IDREF)

Mecanismo sencillo de abstracción

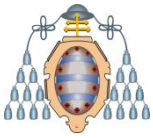
Entidades  $\approx$  Macros

Inclusión de documentos externos

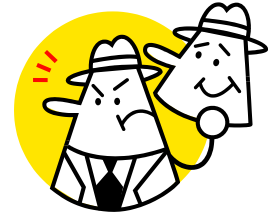
Integrados en XML (Parte de la especificación)

Sencillos de comprender ( $\approx$  Expresiones regulares)





# Limitaciones de DTD's



La Sintaxis **no es XML** (difíciles de manipular)

No soportan **Espacios de nombres**

No permiten especificar **tipos de datos** (por ejemplo: enteros, flotantes, fechas, etc.

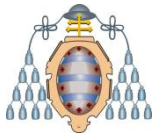
No permiten especificar **secuencias no ordenadas**

$((e1,e2,e3)|(e1,e3,e2)|(e2,e1,e3)|...(e3,e2,e1))$

No hay soporte para declaraciones **sensibles al contexto**: Los elementos se definen todos a nivel de documento, ejemplo, contenido con el mismo nombre cuya estructura cambia en diferentes contextos

Soporte limitado para **Referencias cruzadas**, no es posible formar claves a partir de varios atributos o de elementos

**No son extensibles** (una vez definido, no es posible añadir nuevos vocabularios a un DTD)



*Fin de la Presentación*

