



XML: eXtensible Markup Languaje

Tecnologías XML

DTD (*Document Type Definition*)

Dr. Juan Manuel Cueva Lovelle
Departamento de Informática
Universidad de Oviedo
cueva@uniovi.es

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- Declaración de listas de atributos
- Declaración de entidades
- Declaración de notaciones
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- **Declaración de Tipo de Documento**
- Declaración de elementos
- Declaración de listas de atributos
- Declaración de entidades
- Declaración de notaciones
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Declaración de Tipo de Documento (I): Document Type Definition

Grado en
Ingeniería
Informática
del Software

- **DTD es el conjunto de reglas estándar que deben cumplir un documento XML**
- Pueden ser vistos como “plantillas” o como gramáticas
- Se pueden almacenar en archivos **.dtd**
- El documento XML que se ajusta a su DTD, se denomina “**valido**”
≠ “**bien-formado**”
- Históricamente fue la primera forma de validar XML
- Es la forma más simple y menos potente de validar XML
- Se usará como paso intermedio para llegar a los XML Schema
- **Los DTD determinan:**
 - **Qué elementos están permitidos en un tipo de documento**
 - **Las relaciones entre los elementos y atributos**

Software y estándares para la Web

Declaración de Tipo de Documento (II): Vincular DTDs a documentos XML

Grado en
Ingeniería
Informática
del Software

- Hay **3 formas** de vincular un DTD con un documento XML:

- Incluyendo el **DTD al comienzo del XML** con una etiqueta **!DOCTYPE**

- *Ejemplo:*

```
<!DOCTYPE poema [  
  <!ELEMENT poema (titulo, verso+)>  
  . . .  
<poema>. . . </poema>
```

- Colocarlos **en un archivo aparte** utilizando la misma etiqueta **!DOCTYPE** pero con distinta sintaxis, y así se puede utilizar para validar múltiples documentos XML.

- *Ejemplo:*

```
<!DOCTYPE factura SYSTEM "factura.dtd">
```

- Algunos DTD pueden tener **identificadores públicos**

- *Ejemplo:*

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

Software y estándares para la Web

Declaración de Tipo de Documento (III): Ejemplo **DTD incrustado** *pizzas.xml*

Grado en
Ingeniería
Informática
del Software

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE pizzas [
  <!ELEMENT pizzas (pizza+)>
  <!ELEMENT pizza (ingrediente*)>
  <!ELEMENT ingrediente (#PCDATA)>
  <!ATTLIST pizza nombre CDATA #REQUIRED>
  <!ATTLIST pizza precio CDATA #REQUIRED>
  <!ATTLIST ingrediente nombre CDATA #REQUIRED>
]>
<pizzas>
  <pizza nombre="Barbacoa" precio="8,0€">
    <ingrediente nombre="Salsa Barbacoa"/>
    <ingrediente nombre="Mozzarella"/>
    <ingrediente nombre="Tocineta"/>
    <ingrediente nombre="Ternera"/>
  </pizza>

</pizzas>
```

pizzas.xml

Software y estándares para la Web

Declaración de Tipo de Documento (IV): Ejemplo **DTD incrustado** *libros.xml* (a)

Grado en
Ingeniería
Informática
del Software

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE libros [
<!ELEMENT libros (libro+)>
<!ELEMENT libro (título, autor+, año?, precio, editorial, clasificación, idioma)>
<!ELEMENT título (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT año (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ELEMENT editorial (#PCDATA)>
<!ELEMENT clasificación (#PCDATA)>
<!ELEMENT idioma (#PCDATA)>
<!ATTLIST libro isbn ID #REQUIRED>
<!ATTLIST precio moneda CDATA #REQUIRED>
]>
<libros>
<libro isbn="isbn-9788420633114">
<título>El Aleph</título>
<autor>Jorge Luis Borges</autor>
```

libros.xml

Software y estándares para la Web

Declaración de Tipo de Documento (V): Ejemplo **DTD incrustado** *libros.xml* (b)

Grado en
Ingeniería
Informática
del Software

```
<año>1946</año>  
<precio moneda="Euro">7,80</precio>  
<editorial>Alianza Editorial</editorial>  
<clasificación>Literatura</clasificación>  
<idioma>Español</idioma>  
</libro>
```

```
<libro isbn="isbn-9780470036662">  
<título>Domain-Specific Modeling. Enabling full code generation</título>  
<autor>Steven Kelly</autor>  
<autor>Juha-Pekka Tolvanen</autor>  
<año>2008</año>  
<precio moneda="Dolar USA">74,03</precio>  
<editorial>Wiley</editorial>  
<clasificación>Informática</clasificación>  
<idioma>Inglés</idioma>  
</libro>
```

```
</libros>
```

libros.xml

Software y estándares para la Web

Declaración de Tipo de Documento (VI): Ejemplo **DTD externo** *pizzas.dtd*

Grado en
Ingeniería
Informática
del Software

pizzas.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE pizzas SYSTEM "pizzas.dtd">
<pizzas>
  <pizza nombre="Barbacoa" precio="8,0€">
    <ingrediente nombre="Salsa Barbacoa"/>
    <ingrediente nombre="Mozzarella"/>
    <ingrediente nombre="Tocineta"/>
    <ingrediente nombre="Ternera"/>
  </pizza>
</pizzas>
```

pizzas.dtd

```
<!ELEMENT pizzas (pizza+)>
<!ELEMENT pizza (ingrediente*)>
<!ELEMENT ingrediente (#PCDATA)>
<!ATTLIST pizza nombre CDATA #REQUIRED>
<!ATTLIST pizza precio CDATA #REQUIRED>
<!ATTLIST ingrediente nombre CDATA #REQUIRED>
```

Software y estándares para la Web

Declaración de Tipo de Documento (VII): Ejemplo **DTD externo** *libros.dtd*

Grado en
Ingeniería
Informática
del Software

libros.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE libros SYSTEM "libros.dtd">
<libros>

<libro isbn="isbn-9788420633114">
<título>El Aleph</título>
<autor>Jorge Luis Borges</autor>
<año>1946</año>
<precio moneda="Euro">7,80</precio>
<editorial>Alianza Editorial</editorial>
<clasificación>Literatura</clasificación>
<idioma>Español</idioma>
</libro>

</libros>
```

Software y estándares para la Web

Declaración de Tipo de Documento (VIII): Ejemplo **DTD externo** *libros.dtd*

Grado en
Ingeniería
Informática
del Software

libros.dtd

```
<!ELEMENT libros (libro+)>
<!ELEMENT libro (título, autor+, año?, precio, editorial, clasificación, idioma)>
<!ELEMENT título (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT año (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ELEMENT editorial (#PCDATA)>
<!ELEMENT clasificación (#PCDATA)>
<!ELEMENT idioma (#PCDATA)>
<!ATTLIST libro isbn ID #REQUIRED>
<!ATTLIST precio moneda CDATA #REQUIRED>
```

Software y estándares para la Web

Declaración de Tipo de Documento (IX): Ejemplo **DTD público**

xhtml-1-1.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
<head>
<title>Recursos XML - Juan Manuel Cueva Lovelle</title>
<meta name="Author" content="Juan Manuel Cueva Lovelle"/>
</head>
<body>
<h1> Recursos de XML </h1>
</body>
</html>
```



Grado en
Ingeniería
Informática
del Software

Software y estándares para la Web

Declaración de Tipo de Documento (X): Tipos de declaraciones

Grado en
Ingeniería
Informática
del Software

- **ELEMENT**
 - Elementos del documento XML
- **ATTLIST**
 - Lista de atributos de un elemento
- **ENTITY**
 - Entidad (similares a las macros)
- **NOTATION**
 - Definen otros tipos de contenidos
 - Facilitan la inclusión de formatos binarios (imágenes, vídeos, sonidos, ...)

Software y estándares para la Web

Declaración de Tipo de Documento (XI): Documento XML válido

- **Un documento XML será válido si**
 - además de no tener errores de sintaxis (bien formado)
 - **cumple lo indicado en las declaraciones de elementos, atributos, entidades y notaciones, del DTD al que esté asociado**

Grado en
Ingeniería
Informática
del Software

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- **Declaración de elementos**
- Declaración de listas de atributos
- Declaración de entidades
- Declaración de notaciones
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Declaración de Elementos (I): ELEMENT

Grado en
Ingeniería
Informática
del Software

- Formato: **<!ELEMENT NombreElemento Regla >**
- La regla describe el contenido que es almacenado en el elemento:
 - **PCDATA**: indica que el contenido es solo texto, NO tiene anidación dentro de él. Ej. **<!ELEMENT enfasis (#PCDATA)>**
 - **ANY**: Puede tener cualquier contenido. Es mejor no usarla y estructurar adecuadamente los documentos
Ej. **<!ELEMENT batiburrillo ANY>**
 - **EMPTY**: puede ser vacío
Ej. **<!ELEMENT linea-horizontal EMPTY>**
 - **MIXED**: Puede tener caracteres o una mezcla de caracteres y sub-elementos especificados
Ej. **<!ELEMENT parrafo (#PCDATA|enfasis)*>**
- **ELEMENT**: Puede especificar uno o más sub-elementos contenidos en el elemento que se está definiendo
Ej. **<!ELEMENT html (head, body)>**

Software y estándares para la Web

Declaración de Elementos (II): Ejemplo ELEMENT

Grado en
Ingeniería
Informática
del Software

```
<!ELEMENT receta (titulo, ingredientes, procedimiento)>
```

Documento XML válido

```
<receta>  
  <titulo>Arroz cocido</titulo>  
  <ingredientes>Arroz y ajo</ingredientes>  
  <procedimiento>Cocer el arroz</procedimiento>  
</receta>
```

Documento XML no válido

```
<receta>  
  <párrafo>La siguiente receta me la pasó Álvaro</párrafo>  
  <titulo>Arroz cocido</titulo>  
  <ingredientes>Arroz y ajo</ingredientes>  
  <procedimiento>Cocer el arroz</procedimiento>  
</receta>
```

Error

Software y estándares para la Web

Declaración de Elementos (III): Cardinalidad de elementos

Grado en
Ingeniería
Informática
del Software

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

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

Software y estándares para la Web

Declaración de Elementos (IV): libros.xml y libros.dtd

Grado en
Ingeniería
Informática
del Software

```
<?xml version="1.0" encoding="UTF-8">
<libros>
  <libro isbn="isbn-9788420633114">
    <título>El Aleph</título>
    <autor>Jorge Luis Borges</autor>
    <año>1946</año>
    <precio>10 Euros</precio>
  </libro>
</libros>
```

- El DTD correspondiente es:

```
<!ELEMENT libros (libro+)>
<!ELEMENT libro (título, autor+, año?, precio)>
<!ELEMENT título (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT año (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ATTLIST libro isbn ID #REQUIRED>
```

Software y estándares para la Web

Declaración de Elementos (V): Modelos de contenido

Grado en
Ingeniería
Informática
del Software

- **Indicador de frecuencia**

- Siguen directamente a un identificador general, una secuencia o una opción
- No pueden ir precedidos por espacios en blanco
- Ejemplo:

`<!ELEMENT aviso (título?, (párrafo+, gráfico)*)>`

- **Agrupamientos** utilizando los paréntesis curvos:

Ejemplo:

`<!ELEMENT p (font+, (img, br?)*, a*, ul*, ol*)>`

el elemento p tiene 1 o más ocurrencias del elemento font, el elemento img seguido por el o los elementos br que halla pueden aparecer 0 o más veces, y así sucesivamente.

- Cuando hay elementos que pueden ser conjuntos **no ordenados**, se representan:

Ejemplo: `<!ELEMENT p (font | (img, br?) | a | ul | ol) +>`

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- **Declaración de listas de atributos**
- Declaración de entidades
- Declaración de notaciones
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Declaración de listas de atributos (I): ATTLIST

Grado en
Ingeniería
Informática
del Software

- Todos los atributos son declarados como:

```
<!ATTLIST elemento nombre-del-atributo TYPE Palabra-clave>
```

- Donde palabra clave puede ser:
 - **#REQUIRED**: Es obligatorio especificar el atributo. No tiene valor por defecto.
 - **#IMPLIED**: Se puede omitir el atributo, sin que se adopte automáticamente un valor por defecto
 - **#FIXED**: le da un valor por defecto al atributo

Software y estándares para la Web

Declaración de listas de atributos (II): Atributos y tipos de datos

Grado en Ingeniería Informática del Software

Atributos

#REQUIRED	Obligatorio
#IMPLIED	Opcional
#FIXED	Constante

Tipos de datos

- CDATA = Cadena de caracteres
- NMTOKEN = Palabra (sin espacios)
- NMTOKENS = Lista de palabras
- Enumeración separada por |
- ID = Nombre único (sin duplicados)
- IDREF = Su valor debe apuntar a un ID

```
<!ATTLIST pizza nombre CDATA #REQUIRED>

<!ATTLIST ingrediente nombre CDATA #REQUIRED
                  calorías CDATA #IMPLIED>

<!ATTLIST precio moneda (euros|dólares) #REQUIRED
                  valor CDATA #REQUIRED>

<!ATTLIST persona código ID #REQUIRED>

<!ATTLIST dueño código IDREF #REQUIRED>

<!ATTLIST impuesto tipo CDATA #FIXED "IVA">
```

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

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

<dueño código="35" />

<impuesto tipo="IVA" />
```

Software y estándares para la Web

Declaración de listas de atributos (III): Ejemplo

```
<!ELEMENT mensaje (de, a, texto)>  
<!!ATTLIST mensaje prioridad (normal | urgente) normal>  
<!ELEMENT texto (#PCDATA)>  
<!!ATTLIST texto idioma CDATA #REQUIRED>
```

```
<mensaje prioridad="urgente">  
  <de>Menganito</de>  
  <a>Juanita</a>  
  <texto idioma="español">  
    Lo que le tenga que contar Menganito a Juanita  
  </texto>  
</mensaje>
```


Software y estándares para la Web

Declaración de listas de atributos (IV): Tipos de atributos (a)

Grado en
Ingeniería
Informática
del Software

- **CDATA** (character data): texto

```
<!--ATTLIST mensaje fecha CDATA #REQUIRED-->  
<mensaje fecha="21 de Mayo de 2003">
```
- **Enumerados**: Sólo pueden contener un valor de entre un número reducido de opciones ("|")

```
<!--ATTLIST mensaje prioridad (normal | urgente) normal-->
```
- **ID y IDREF**
 - El tipo **ID** permite que un atributo determinado tenga un nombre único que podrá ser referenciado por un atributo de otro elemento que sea de tipo **IDREF**
 - Permite implementar un sistema de hipervínculos en un documento XML

```
<!--ELEMENT enlace EMPTY-->  
<!--ATTLIST enlace destino IDREF #REQUIRED-->  
<!--ELEMENT capitulo (parrafo)*-->  
<!--ATTLIST capitulo referencia ID #IMPLIED-->
```

- **NMTOKEN**

- Los atributos NMTOKEN (*NaMe TOKEN*) aceptan los caracteres válidos de XML: letras, números, puntos, guiones, subrayados y los dos puntos; pero NO PUEDEN TENER ESPACIOS EN BLANCO EN MEDIO. No es necesario que cumplan las reglas de los identificadores (por ejemplo pueden empezar por un número)
- Algunos autores los denominan *testigos de nombre XML*

```
<!ATTLIST mensaje fecha CDATA #REQUIRED>  
<mensaje fecha="15 de Diciembre de 1999">
```

```
<!ATTLIST mensaje fecha NMTOKEN #REQUIRED>  
<mensaje fecha="15-12-1999">
```

- **NMTOKENS**

Contiene uno o más testigos de nombre (NMTOKEN) separados por un espacio en blanco

```
<!ATTLIST mensajes fecha CDATA #REQUIRED>
```

```
<mensaje fecha="14 de Diciembre de 1996, 1 de Marzo de 2008, 25 de septiembre de 2012">
```

```
<!ATTLIST mensaje fecha NMTOKENS #REQUIRED>
```

```
<mensaje fecha="14-12-1996 01-03-2008 25-09-2012">
```

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- Declaración de listas de atributos
- **Declaración de entidades**
- Declaración de notaciones
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Declaración de entidades (I): ENTITY

Grado en
Ingeniería
Informática
del Software

- Las **entidades** se utilizan para hacer referencia a objetos (ficheros, páginas Web, imágenes,...) que no deben ser analizados sintácticamente según las reglas de XML
- Se declaran mediante “<!**ENTITY**”
- Puede usarse para declarar una abreviatura que se utiliza como una forma más corta de algunos textos
- En otras ocasiones es una referencia a un objeto externo o local

Software y estándares para la Web

Declaración de entidades (II): Tipos de entidades (a)

- Las entidades pueden ser:
 - Internas o Externas**
 - Analizadas o No analizadas**
 - Generales o Parámetro**
- Entidades generales internas:

```
<!DOCTYPE texto[  
<!ENTITY ovni "Objeto Volador No Identificado">  
>  
<texto>  
  <titulo> Durmiendo en clase, Juan Manuel vio un &ovni;  
  </titulo>  
</texto>
```

- Son básicamente abreviaturas definidas en la sección del DTD del documento XML. Son como macros.
- Son siempre entidades analizadas

Software y estándares para la Web

Declaración de entidades (II): Tipos de entidades (b)

Grado en
Ingeniería
Informática
del Software

- Entidades generales externas analizadas

- Permiten incluir documentos XML externos en el documento actual

```
<!ENTITY enlaceUniovi SYSTEM "http://www.uniovi.es/intro.xml">
```

- Entidades no analizadas

```
<!ENTITY logo SYSTEM "http://www.uniovi.es/logo.gif">
```

- Entidades parámetro Internas

```
<!DOCTYPE texto[  
  <!ENTITY % elemento-alf "!ELEMENT ALF (#PCDATA)">  
  %elemento-alf; ]>
```

- Entidades parámetro Externas

```
<!DOCTYPE texto[  
  <!ENTITY % elemento-alf SYSTEM "alf.ent">  
  ...  
  %elemento-alf; ]>
```

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- Declaración de listas de atributos
- Declaración de entidades
- **Declaración de notaciones**
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Declaración de notaciones (I)

Grado en
Ingeniería
Informática
del Software

- Las notaciones se utilizan para especificar el **formato de entidades externas** (datos no XML)
- Ejemplos:
 - Archivos de imágenes
 - Archivos de música
 - Archivos de video
- Estas entidades externas no serán analizadas por los procesadores XML
- En un DTD se pueden declarar dos tipos de notaciones
 - Privadas con **SYSTEM**
 - Públicas con **PUBLIC**
- **Sintaxis**

<!NOTATION nombre-de-la-notación SYSTEM "identificador-del-sistema">

<!NOTATION nombre-de-la-notación PUBLIC "identificador-público">

<!NOTATION nombre-de-la-notación PUBLIC "identificador-público" "identificador-del-sistema">

Software y estándares para la Web

Declaración de notaciones (II): Ejemplo con SYSTEM

Grado en
Ingeniería
Informática
del Software

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE frutas [
  <!ELEMENT frutas (fruta)*>
  <!ELEMENT fruta EMPTY>
  <!ATTLIST fruta foto ENTITY #REQUIRED>

  <!ENTITY manzana SYSTEM "manzana.gif" NDATA gif>
  <!ENTITY naranja SYSTEM "naranja.gif" NDATA gif>

  <!NOTATION gif SYSTEM "image/gif">
]>

<frutas>
  <fruta foto="manzana"/>
  <fruta foto="naranja"/>
</frutas>
```

Software y estándares para la Web

Declaración de notaciones (III): Ejemplo con SYSTEM

Grado en
Ingeniería
Informática
del Software

- En el DTD del ejemplo anterior se está indicando que los valores –datos– de las entidades (**manzana** y **naranja**) van a ser cargados desde una **URI** (*Uniform Resource Identifier*, Identificador Uniforme de Recurso).
- En este caso, se hace referencia a los archivos externos "**manzana.gif**" y "**naranja.gif**".
- Con **NDATA** (*Notation Data*) se ha asociado a las entidades manzana y naranja con la notación **gif**.
- La notación **gif** es una declaración del tipo **MIME** *image/gif*.

Software y estándares para la Web

Declaración de notaciones (IV): Ejemplo con PUBLIC

Grado en
Ingeniería
Informática
del Software

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE frutas [
  <!ELEMENT frutas (fruta)*>
  <!ELEMENT fruta EMPTY>
  <!ATTLIST fruta foto ENTITY #REQUIRED>

  <!ENTITY manzana SYSTEM "manzana.gif" NDATA gif>
  <!ENTITY naranja SYSTEM "naranja.gif" NDATA gif>

  <!NOTATION gif PUBLIC "GIF 1.0" "image/gif" >
]>

<frutas>
  <fruta foto="manzana"/>
  <fruta foto="naranja"/>
</frutas>
```

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- Declaración de listas de atributos
- Declaración de entidades
- Declaración de notaciones
- **Espacios de nombres**
- Proceso de validación de documentos XML con DTD
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Espacios de nombres (I)

Grado en
Ingeniería
Informática
del Software

- Permite crear nombres en un documento XML que son identificados por una URI (*Uniform Resource Identifier*)
- Ejemplo de XML valido que introduce la problemática de etiquetas con igual nombre.
- Homonimia: Mismo nombre con diferentes propósitos

```
<país nombre="Francia">  
  <capital>París</capital>  
</país>
```

```
<inversión>  
  <capital>1200€</capital>  
</inversión>
```

```
<inversiones>  
  <país nombre="Francia">  
    <capital>París</capital>  
    <capital>1200€</capital>  
  </país>  
  . . .  
</inversiones>
```



Software y estándares para la Web

Espacios de nombres (II)

Grado en
Ingeniería
Informática
del Software

- Son definidos dentro de los elementos del documento XML.
- Si un elemento tiene elementos hijos, ellos pueden heredar el *namespace* del padre o sobrescribir el mismo.
- Prefijo para referenciar el *namespace*
 - No puede llamarse ni *xmlns* ni *xml*.
- Declarados dentro de la etiqueta inicial de definición de un elemento
 - Ejemplo:

```
<namespacePrefix:elementName xmlns:namespacePrefix='URL'>
```

Software y estándares para la Web

Espacios de nombres (III)

Grado en
Ingeniería
Informática
del Software

- La solución consistirá en asociar a cada etiqueta una URI que identificaría el espacio de nombres al que pertenece
- La URI sirve simplemente para evitar ambigüedad, obteniendo de esta forma **un identificador global único**
- Conceptualmente se representaría como:

```
<[http://www.bolsa.com]:inversiones>  
<[http://www.geog.es]:pais [http://www.geog.es]:nombre="Francia">  
  <[http://www.geog.es]:capital>París</[http://www.geog.es]:capital>  
  
<[http://www.bolsa.com]:capital>1200€</[http://www.bolsa.com]:capital>  
</[http://www.geog.es]:pais>  
.  
.  
.  
</[http://www.bolsa.com]:inversiones>
```

Legibilidad...



Software y estándares para la Web

Espacios de nombres (IV)

Grado en
Ingeniería
Informática
del Software

- Para abreviar la sintaxis se asocian alias mediante `xmlns:alias="...URI..."`
- Todos los elementos en el ámbito de esa declaración que empiecen por `alias:nombre` pertenecen a dicho espacio de nombres

```
<b:inversiones xmlns:b="http://www.bolsa.com"
xmlns:geo="http://www.geo.es">
  <geo:país geo:nombre="Francia">
    <geo:capital>París</geo:capital>
  <b:capital>1200€</b:capital>
</geo:país>
. . .
</b:inversiones>
```

NOTA: Las URIs sólo se utilizan para que el nombre sea único, no son enlaces, ni tienen que contener información

- **Asignación dinámica:** Es posible ir asociando espacios de nombres a los elementos según van apareciendo

```
<bolsa:inversiones
  xmlns:bolsa="http://www.bolsa.com">
  <geog:país
    xmlns:geog="http://www.geog.es"
    geog:nombre="Francia">
    <geog:capital>París</geog:capital>
    <bolsa:capital>1200€</bolsa:capital>
  </geog:país>
  .
  .
  .
</bolsa:inversiones>
```

Software y estándares para la Web

Espacios de nombres (VI)

Grado en
Ingeniería
Informática
del Software

- Espacios de nombres por defecto: Mediante `xmlns=""` se define un espacio de nombres por defecto (sin alias)

```
<inversiones
  xmlns="http://www.bolsa.com">
  <geog:país
    xmlns:geog="http://www.geog.es"
    geog:nombre="Francia">
    <geog:capital>París</geog:capital>
    <capital>1200€</capital>
  </geog:país>
  . . .
</inversiones>
```

Se refiere a
<http://www.bolsa.com>

Software y estándares para la Web

Espacios de nombres (VII)

- Incorporación de los espacios de nombres a los DTDs
 - Hay que definir los espacios de nombre usados ya que los DTDs no los contemplan

Grado en
Ingeniería
Informática
del Software

```
<!DOCTYPE inversiones [  
<!ELEMENT inversiones (geog:país*)>  
<!ELEMENT geog:país (geog:capital,capital) >  
<!ELEMENT geog:capital (#PCDATA)>  
<!ELEMENT capital (#PCDATA)>  
<!ATTLIST inversiones  
      xmlns CDATA #FIXED "http://www.bolsa.com">  
<!ATTLIST geog:país  
      geog:nombre CDATA #REQUIRED  
      xmlns:geog CDATA #FIXED "http://www.geog.es">  
>
```

- Ampliamente utilizados para combinar vocabularios
- Facilitan la incorporación de elementos no previstos inicialmente
- Sintaxis *extraña* al principio
 - Uso de prefijos
 - URIs como elemento diferenciador...pero las URLs también sirven para acceder a recursos
- Complican los DTDs

Software y estándares para la Web

Esquema

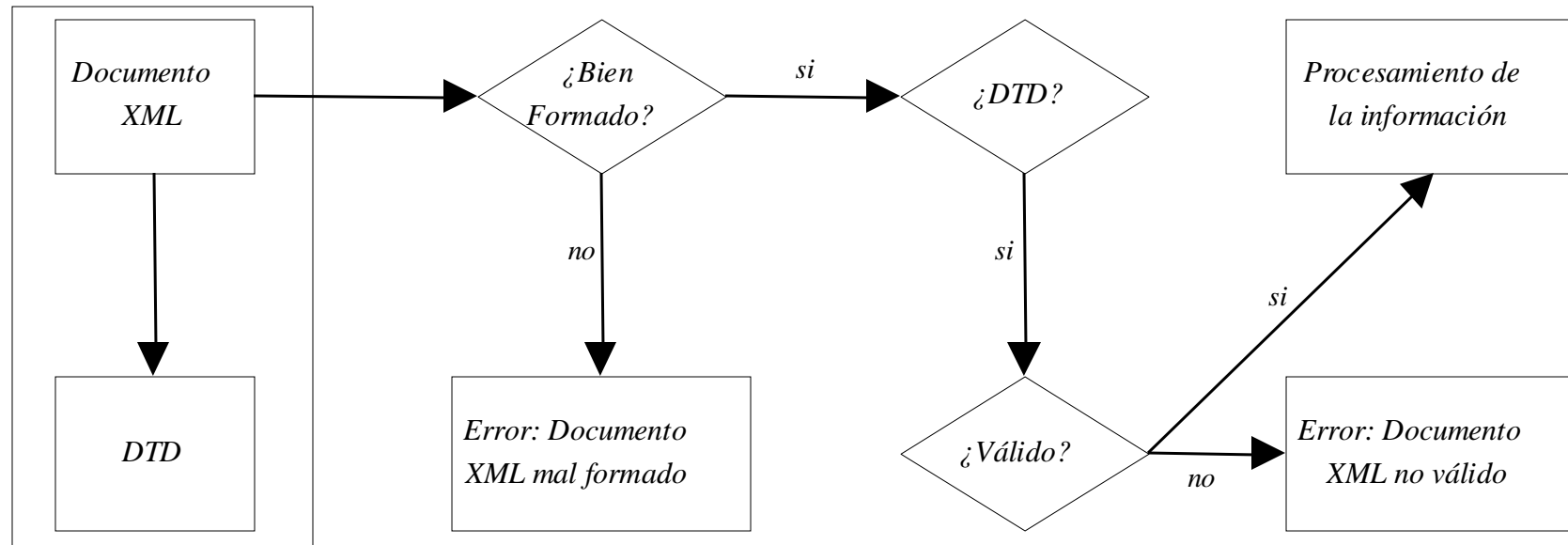
Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- Declaración de listas de atributos
- Declaración de entidades
- Declaración de notaciones
- Espacios de nombres
- **Proceso de validación de documentos XML con DTD**
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Proceso de validación de documentos XML con DTD (I)

Grado en
Ingeniería
Informática
del Software

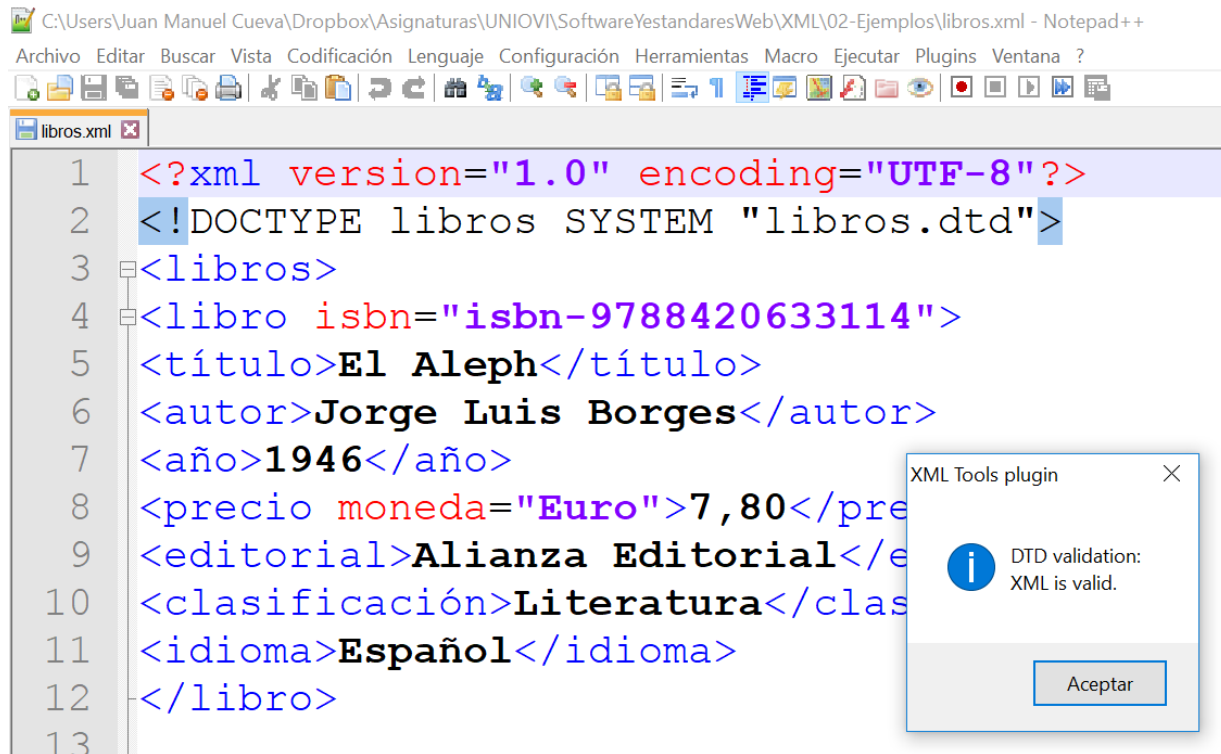


Software y estándares para la Web

Proceso de validación de documentos XML con DTD (II)

Grado en
Ingeniería
Informática
del Software

- Usando **Notepad++**
- Se debe instalar un *plugin*
- Menú: Plugins – Plugin Manager – XML Tools



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE libros SYSTEM "libros.dtd">
3 <libros>
4 <libro isbn="isbn-9788420633114">
5   <título>El Aleph</título>
6   <autor>Jorge Luis Borges</autor>
7   <año>1946</año>
8   <precio moneda="Euro">7,80</precio>
9   <editorial>Alianza Editorial</editorial>
10  <clasificación>Literatura</clasificación>
11  <idioma>Español</idioma>
12 </libro>
13
```


Software y estándares para la Web

Proceso de validación de documentos XML con DTD (III)

Grado en
Ingeniería
Informática
del Software

- Validación en línea:
 - <https://codebeautify.org/xmlvalidator>

The screenshot shows the Code Beautify XML Validator interface. The browser address bar displays `codebeautify.org/xmlvalidator`. The page title is "XML Validator". Below the title, there are tabs for "XML Validator", "XML Stylesheet Editor", and "XML Viewer". The "XML Validator" tab is active. The XML document being validated is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE libros [
3
4 <!ELEMENT libros (libro+)>
5 <!ELEMENT libro (titulo, autor+, año?, precio, editorial, clasificación, idioma)>
6 <!ELEMENT titulo (#PCDATA)>
7 <!ELEMENT autor (#PCDATA)>
8 <!ELEMENT año (#PCDATA)>
9 <!ELEMENT precio (#PCDATA)>
10 <!ELEMENT editorial (#PCDATA)>
11 <!ELEMENT clasificación (#PCDATA)>
12 <!ELEMENT idioma (#PCDATA)>
13 <!ATTLIST libro isbn ID #REQUIRED>
14 <!ATTLIST precio moneda CDATA #REQUIRED>
15 ]>
16 <libros>
17 <libro isbn="isbn-9788420633114">
18 <titulo>El Aleph</titulo>
19 <autor>Jorge Luis Borges</autor>
```

Below the XML document, there are buttons for "Validate", "Load URL", "Browse", and "Download". The "Validate" button has been clicked, and a green bar at the bottom of the editor area displays the message "Valid XML". At the bottom of the page, there is a Google AdWords banner that reads "Destaca con Google AdWords. Empieza ahora con 75€ de publicidad gratuita en AdWords*".

Software y estándares para la Web

Proceso de validación de documentos XML con DTD (IV)

Grado en
Ingeniería
Informática
del Software

- Usando Eclipse con plugin XML
- Usando Visual Studio 2019 o siguientes
- En Linux con el comando **xmllint**
 - `$ xmllint --noout --dtdvalid poema.dtd alba.xml`



The image displays two terminal windows. The top window, titled 'cueva@Dron: /mnt/e/Dropbox/Python-Win-02000-Servicios-Web', shows a red background with white text. It contains the command `(base) cueva@Dron:/mnt/e/Dropbox/Python-Win-02000-Servicios-Web$ xmllint --noout --dtdvalid poema.dtd alba.xml` and the prompt `(base) cueva@Dron:/mnt/e/Dropbox/Python-Win-02000-Servicios-Web$`. The bottom window, titled 'Python-Win-02000-Servicios-Web — -bash — 117x5', has a blue background with white text. It shows the command `(base) MacBook-Air-de-Usuario:Python-Win-02000-Servicios-Web usuario$ xmllint --noout --dtdvalid poema.dtd alba.xml` and the prompt `(base) MacBook-Air-de-Usuario:Python-Win-02000-Servicios-Web usuario$`.

Software y estándares para la Web

Proceso de validación de documentos XML con DTD (V)

Grado en
Ingeniería
Informática
del Software

- **Visual Studio Code**
 - Tiene varios plug-ins
 - Se recomienda “**XML Language Support by Red Hat**”
 - <https://marketplace.visualstudio.com/items?itemName=redhat.vscode-xml>
 - Soporta validación de XML con DTD

Software y estándares para la Web

Esquema

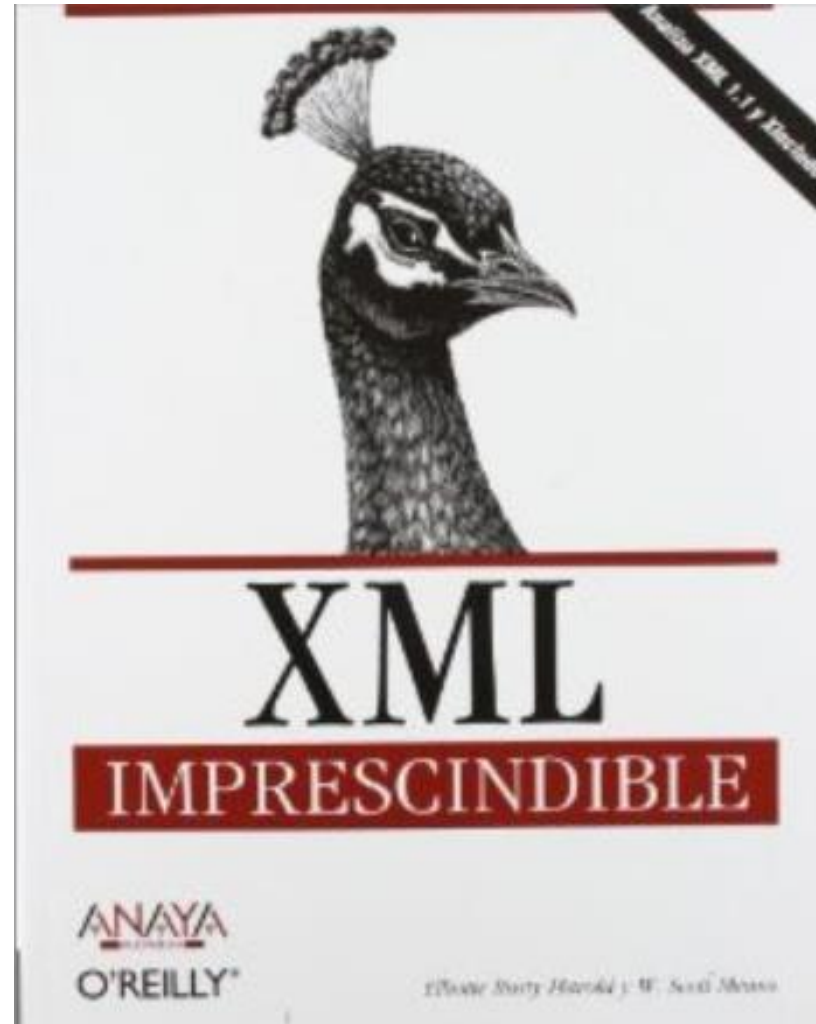
Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- Declaración de listas de atributos
- Declaración de entidades
- Declaración de notaciones
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- **Bibliografía**
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Tecnologías XML

Bibliografía (I)

- Libro recomendado de lectura y consulta:
 - “**XML imprescindible**”
 - ANAYA/O'Reilly (2005)
 - E. Rusty Harold y W. Scott Means



**Máster en
Ingeniería
Web**

Tecnologías XML

Bibliografía (II)

- Libro recomendado de consulta:
 - “**Beginning XML**”
 - John Wiley & Sons (2012)
 - Joe Fawcett, Liam R.E. Quin, and Danny Ayers



**Máster en
Ingeniería
Web**

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- Declaración de listas de atributos
- Declaración de entidades
- Declaración de notaciones
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- Bibliografía
- **Referencias Web**
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Referencias Web

Grado en
Ingeniería
Informática
del Software

- <http://www.w3.org/TR/xml11/>
 - Especificación del estándar XML en W3C
 - Última versión 29-Septiembre-2006
- <http://www.w3schools.com/xml>
 - Tutoriales on-line de XML
- Libros on-line sobre XML
 - <http://www.onlineprogrammingbooks.com/xml/>
- Tutorial de DTD
 - <https://www.abrirllave.com/dtd/guion-del-tutorial.php>
- Validadores de XML con DTD
 - <https://codebeautify.org/xmlvalidator>
 - <http://www.xmlvalidation.com/documentation.html>

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- Declaración de listas de atributos
- Declaración de entidades
- Declaración de notaciones
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- Bibliografía
- Referencias Web
- **Ejercicios resueltos**
- Ejercicios propuestos

Software y estándares para la Web

Ejercicios resueltos (I)

Grado en
Ingeniería
Informática
del Software

- Construir un **documento XML bien formado y válido usando un DTD** para contener artículos de revistas con los siguientes requisitos mínimos:
 - Título del artículo
 - Autores y su correo electrónico
 - Resumen
 - Palabras clave
 - Nombre de la revista
 - Número o volumen de la revista
 - Página de inicio del artículo
 - Página final del artículo
 - Año

Software y estándares para la Web

Ejercicios resueltos (I) – **articulos.dtd**

Grado en
Ingeniería
Informática
del Software

```
<!ELEMENT articles (article+)>
<!ELEMENT article (title, authors, summary, keywords, journal, volume)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT authors (author+)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT summary (#PCDATA)>
<!ELEMENT keywords (keyword+)>
<!ELEMENT keyword (#PCDATA)>
<!ELEMENT journal (#PCDATA)>
<!ELEMENT volume (#PCDATA)>
<!ATTLIST article
    year CDATA #REQUIRED
    start-page CDATA #REQUIRED
    end-page CDATA #REQUIRED>
<!ATTLIST author
    email CDATA #REQUIRED>
```

Software y estándares para la Web

Ejercicios resueltos (I) – **articulos.xml** validado con DTD

E:\Dropbox\Asignaturas\MIW\Lenguajes-y-estandares-Web\XML\2018\02-Ejemplos\articulos.xml - Notepad++

Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?

articulos.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE articles SYSTEM "articulos.dtd">
3 <articles>
4   <article year="2017" start-page="301" end-page="313">
5     <title>Midgar: Detection of people through computer vision in the Internet of Things scenarios to improve the security in Smart Cities, Smart Towns, and Smart Homes.</title>
6     <authors>
7       <author email="cristian@email.es">Cristian González García</author>
8       <author email="daniel@email.es">Daniel Meana-Llorián</author>
9       <author email="cristina@email.es">B. Cristina Pelayo García-Bustelo</author>
10      <author email="juanmanuel@email.es">Juan Manuel Cueva Lovelle</author>
11      <author email="nestor@email.es">Néstor García-Fernández</author>
12    </authors>
13    <summary>
14      Resumen del artículo "Midgar: Detection of people through computer vision in the Internet of Things scenarios to improve the security in Smart Cities, Smart Towns, and Smart Homes.".
15    </summary>
16    <keywords>
17      <keyword>People</keyword>
18      <keyword>Internet</keyword>
19      <keyword>Things</keyword>
20    </keywords>
21    <journal>Future Generation Comp. Syst.</journal>
22    <volume>76</volume>
23  </article>
24
25  <article year="2016" start-page="299" end-page="337">
26    <title>Combining the Continuous Integration Practice and the Model-Driven Engineering Approach.</title>
27    <authors>
28      <author email="vicente@email.es">Vicente García-Díaz</author>
29      <author email="jordan@email.es">Jordán Pascual Espada</author>
30      <author email="edward@email.es">Edward Rolando Núñez-Valdéz</author>
31      <author email="cristina@email.es">B. Cristina Pelayo García-Bustelo</author>
32      <author email="juanmanuel@email.es">Juan Manuel Cueva Lovelle</author>
33    </authors>
34    <summary>
35      Resumen del artículo "Combining the Continuous Integration Practice and the Model-Driven Engineering Approach.".
36    </summary>
37    <keywords>
38      <keyword>Integration</keyword>
39      <keyword>Practice</keyword>
40      <keyword>Model-Driven</keyword>
41    </keywords>
42    <journal>Computing and Informatics</journal>
43    <volume>35</volume>
44  </article>
45
46  <article year="2013" start-page="51" end-page="55">
```

XML Tools plugin

DTD validation:
XML is valid.

Aceptar

length: 9.379 lines: 223 Ln: 1 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

Software y estándares para la Web

Ejercicios resueltos (II)

Grado en
Ingeniería
Informática
del Software

- Construir un **documento XML bien formado y válido usando un DTD** para contener recetas de cocina con los siguientes requisitos mínimos:
 - Nombre de la receta (por ejemplo “Fabada Asturiana”)
 - Tipo de plato (postre, primer plato, entrante,. . .)
 - Ingredientes con cantidades (por ejemplo “Fabes 500 gramos”)
 - Calorías del plato (opcional)
 - Proceso de elaboración, especificado en pasos, por ejemplo:
 - Paso 1: Poner *les fabes* a remojo la noche anterior
 - Paso 2: Poner *les fabes* a cocer con agua y laurel
 - Paso 3: etc...
 - Dificultad del proceso de elaboración (por ejemplo “Fácil”, “Medio”, “Difícil”,...)
 - Tiempo de elaboración (por ejemplo “45 minutos”)
 - Elementos utilizados para la elaboración (microondas, wok, horno, freidora,. . .)
 - Origen de la receta (por ejemplo “Receta de mi abuela”, “Libro de M^a Luisa”, “Libro de las 1001 recetas”, “www.recetasMUYricas.com”)

Software y estándares para la Web

Ejercicios resueltos (II) – **recetas.dtd**

Grado en
Ingeniería
Informática
del Software

```
<!ELEMENT cooking-recipes (cooking-recipe+)>
<!ELEMENT cooking-recipe (name, ingredients, elaboration-process, origin)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT ingredients (ingredient+)>
<!ELEMENT ingredient (#PCDATA)>
<!ELEMENT elaboration-process (steps, elements)>
<!ELEMENT steps (step+)>
<!ELEMENT step (#PCDATA)>
<!ELEMENT elements (element*)>
<!ELEMENT element (#PCDATA)>
<!ELEMENT origin (#PCDATA)>
<!ATTLIST cooking-recipe
    type CDATA #REQUIRED
    vegan CDATA #REQUIRED>
<!ATTLIST ingredient
    quantity CDATA #REQUIRED
    measure CDATA #REQUIRED
    calories CDATA #IMPLIED>
<!ATTLIST elaboration-process
    difficulty CDATA #REQUIRED
    time-minutes CDATA #REQUIRED>
<!ATTLIST step
    number CDATA #REQUIRED>
```

Software y estándares para la Web

Ejercicios resueltos (II) – recetas.xml validado con DTD

E:\Dropbox\Asignaturas\MIW\Lenguajes-y-estándares-Web\XML\2018\02-Ejemplos\recetas.xml - Notepad++

Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?

recetas.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE recetas SYSTEM "recetas.dtd">
3 <cooking-recipes>
4   <cooking-recipe type="Primer plato" vegan="No">
5     <name>Arroz integral con pollo</name>
6     <ingredients>
7       <ingredient quantity="300" measure="g" calories="1032">Arroz Integral</ingredient>
8       <ingredient quantity="300" measure="g" calories="80">Pollo</ingredient>
9       <ingredient quantity="1" measure="unidad">Cebolleta</ingredient>
10      <ingredient quantity="2" measure="unidades">Tallos de apio</ingredient>
11      <ingredient quantity="3" measure="unidades">Dientes de ajo</ingredient>
12      <ingredient quantity="2" measure="cucharadas">Tomate frito</ingredient>
13      <ingredient quantity="1/2" measure="cucharadita">Curry en pavo</ingredient>
14      <ingredient quantity="1/2" measure="cucharadita">Pimentón dulce</ingredient>
15      <ingredient quantity="1/4" measure="unidad">Pimiento verde</ingredient>
16      <ingredient quantity="1" measure="unidad">Huevo</ingredient>
17      <ingredient quantity="3" measure="cucharadas">Salsa de soja</ingredient>
18      <ingredient quantity="700" measure="ml" calories="20">Caldo de pollo</ingredient>
19    </ingredients>
20    <elaboration-process difficulty="Fácil" time-minutes="45">
21      <steps>
22        <step number="1">Comenzamos preparando un sofrito: cortar la cebolleta, ajos, pimiento verde, y los tallos de apio en trozos. Añadimos aceite de oliva a una sartén y sofreimos prime
23        <step number="2">En este mismo aceite dorar el pollo durante dos minutos, y retirar. Agregar la cebolleta, el pimiento verde y los tallos de apio cortados en trozos. Sofreir durante
24        <step number="3">A continuación añadir el caldo de pollo y cocinar el arroz 20 minutos. Cuando falten 10 minutos, añadiremos el pollo que habíamos retirado y los ajos. Continuar la
25        <step number="4">Tapar el arroz con un paño y dejar reposar 8 minutos. Mientras reposa batimos el huevo y hacemos una tortilla francesa en una sartén antiadherente. Una vez a nuestro
26        <step number="5">Una vez reposado el arroz añadimos los trocitos de tortilla y sazonamos con la salsa de soja y un poco de pimienta.</step>
27      </steps>
28      <elements>
29        <element>Cuchillo</element>
30        <element>Horno</element>
31        <element>Sartén</element>
32      </elements>
33    </elaboration-process>
34    <origin>Recetas de mi abuela</origin>
35  </cooking-recipe>
36
37  <cooking-recipe type="Entrante" vegan="Si">
38    <name>Aceituna fritas caseras</name>
39    <ingredients>
40      <ingredient quantity="200" measure="g">Aceitunas rellenas de pimiento</ingredient>
41      <ingredient quantity="2" measure="unidades">Huevos</ingredient>
42      <ingredient quantity="90" measure="g">Harina</ingredient>
43      <ingredient quantity="90" measure="g">Pan rallado</ingredient>
44      <ingredient quantity="250" measure="ml">Aceite de oliva</ingredient>
45    </ingredients>
```

XML Tools plugin

DTD validation:
XML is valid.

Aceptar

length: 31.021 lines: 384 Ln: 2 Col: 40 Sel: 0|0 Windows (CR LF) UTF-8 INS

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Declaración de Tipo de Documento
- Declaración de elementos
- Declaración de listas de atributos
- Declaración de entidades
- Declaración de notaciones
- Espacios de nombres
- Proceso de validación de documentos XML con DTD
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- **Ejercicios propuestos**

Software y estándares para la Web

Ejercicios propuestos (I): Monumentos prerrománicos

Grado en
Ingeniería
Informática
del Software

- Construir un **documento XML bien formado y válido usando un DTD** para contener monumentos del prerrománico asturiano (5 monumentos) con los siguientes requisitos mínimos:
 - Nombre del monumento (por ejemplo “Fuente de Foncalada”)
 - Tipo de monumento (por ejemplo “Arquitectura hidráulica”, “Iglesia”, “Palacio”)
 - Año de construcción aproximado (por ejemplo “1096”)
 - Constructor (por ejemplo “Desconocido”, “Ramiro I”)
 - Descripción del monumento
 - Estado del monumento (por ejemplo “bien conservado”, “mal conservado”, etc)
 - Municipio (por ejemplo “Oviedo”)
 - Dirección opcional (por ejemplo “calle Foncalada”)
 - Coordenadas geográficas: longitud, latitud y altitud
 - Galería de fotografías:
 - Fotografía 1: Por ejemplo Foncalada-01.jpg
 - Fotografía 2: Por ejemplo Foncalada-02.jpg
 - Fotografía 3: etc...

Software y estándares para la Web

Ejercicios propuestos (I): Monumentos prerrománicos - continuación

**Grado en
Ingeniería
Informática
del Software**

- Galería de vídeos:
 - Video 1: Por ejemplo Foncalada-01.mpeg
 - Vídeo 2: Por ejemplo Foncalada-02.mpeg
 - Video 3: etc...
- Recomendación de visita de 0 a 10 (por ejemplo “7”)
- Horario de visita
- Días de visita
- ¿Quién lo enseña?
- Referencias y bibliografía con información del monumento
 - Referencia 1: por ejemplo <https://es.wikipedia.org/wiki/Foncalada>
 - Referencia 2: <http://prerromancoasturiano.es/>
 - Referencia 3. etc.

Software y estándares para la Web

Ejercicios propuestos (II): Rutas turísticas

Grado en
Ingeniería
Informática
del Software

- Construir un **documento XML bien formado y válido usando un DTD** para contener rutas turísticas (5 rutas mínimo) con los siguientes requisitos mínimos:
 - Nombre de la ruta turística (por ejemplo “Ruta por Oviedo”)
 - Tipo de ruta (por ejemplo “Arquitectura y monumentos”, “Gastronómica”, “Paisajística”, “Mixta tapas y monumentos”, “Escalada”, “Senderismo”, etc.)
 - Medio de transporte (por ejemplo “A pie”, “Automóvil”, “Bicicleta”, “Canoa”, “Mixta a pie y tren”, etc.)
 - Fecha de inicio de la ruta (opcional)
 - Hora de inicio de la ruta (opcional)
 - Tiempo de duración de la ruta (por ejemplo “2 horas”, “3 días”, “2 semanas”, “3 meses”)
 - Agencia que gestiona la ruta (por ejemplo “Sin agencia”, “NaturAller”)
 - Descripción de la ruta
 - Personas adecuadas para la ruta (por ejemplo “Se puede ir con niños”, “Personas en buena forma física”, “tercera edad”, etc.)
 - Lugar de inicio de la ruta (por ejemplo “Oviedo”)
 - Dirección de inicio de la ruta (por ejemplo “calle Foncalada”)
 - Coordenadas geográficas de inicio de la ruta: longitud, latitud y altitud

Software y estándares para la Web

Ejercicios propuestos (II): Rutas turísticas - continuación

**Grado en
Ingeniería
Informática
del Software**

- Referencias y bibliografía con información de la ruta (mínimo 3)
 - Referencia 1: por ejemplo <https://es.wikipedia.org/wiki/Foncalada>
 - Referencia 2: <http://prerromancoasturiano.es/>
 - Referencia 3. etc.
- Recomendación de la ruta de 0 a 10 (por ejemplo “7”)
- Hitos de la ruta (mínimo 3 hitos):
 - Nombre del sitio
 - Descripción del sitio
 - Coordenadas geográficas del sitio: longitud, latitud, altitud
 - Distancia desde el hito anterior (las unidades se expresarán como atributos)
 - Galería de fotografías del hito (mínimo 1, máximo 5)
 - Fotografía 1: Por ejemplo Monumento.jpg
 - Fotografía 2: Por ejemplo Panorama.jpg
 - Fotografía 3: etc...
 - Galería de vídeos del hito (opcional). Mínimo 0 y máximo 3.
 - Video 1: Por ejemplo Paisaje360.mpeg o enlace a YouTube, Vimeo, etc.
 - Video 2: Por ejemplo Modelo3D.mpeg
 - Video 3: etc...



XML: eXtensible Markup Languaje

Tecnologías XML

DTD (*Document Type Definition*)

Dr. Juan Manuel Cueva Lovelle
Departamento de Informática
Universidad de Oviedo
cueva@uniovi.es