
 Departamento de Lenguajes y  
Sistemas Informáticos

 Universitat d'Alacant  
Universidad de Alicante

# XML

Programación en Internet  
Curso 2009-2010

Programación en Internet – Curso 2009-2010

## Contenidos

- Introducción
- Versiones
- Estructura de un documento
- Definición de un DTD
- Validación de un documento

## Introducción

- XML
- Versiones
- Ventajas
- Objetivos de diseño
- Qué necesito para usar XML

## XML

- *Extensible Markup Language*
- *World Wide Web Consortium (W3C)*
- Problema:
  - HTML inflexible, mezcla contenido y presentación
  - SGML demasiado complejo
- Subconjunto de SGML

GML (IBM, 1969)



SGML (ISO 8879, 1986)



XML (W3C, 1998)

Programación en Internet – Curso 2009-2010

## XML

- No es un lenguaje (no etiquetas predefinidas), es un **metalenguaje**:
  - Definir etiquetas y sus atributos
  - Definir relaciones estructurales
- XHTML: híbrido HTML + XML
  - HTML escrito según XML (aplicación de XML)
  - Sustituto de HTML

Programación en Internet – Curso 2009-2010

## Versiones

- 10/2/1998: XML 1.0
- 6/10/2000: XML 1.0 Second Edition
- 4/2/2004: XML 1.0 Third Edition
- 16/8/2006: XML 1.0 Fourth Edition
  - Las tres ediciones corrigen erratas y clarifican/detallan el estándar, pero no son una nueva versión

## Versiones

- 4/2/2004: XML 1.1
  - Actualiza 1.0 respecto al juego de caracteres (no depende de una versión específica de Unicode, permite utilizar siempre la última), añade verificación de normalización de caracteres y las normas de final de línea se ajustan más a Unicode
  - W3C: *“You are encouraged to create or generate XML 1.0 documents if you do not need the new features in XML 1.1; XML Parsers are expected to understand both XML 1.0 and XML 1.1”*
- 16/8/2006: XML 1.1 Second Edition
  - Corrige erratas y clarifica/detalla el estándar, pero no es una nueva versión

## Versiones

- **Extensible Markup Language (XML) 1.0 (Fourth Edition)**
  - W3C Recommendation 16 August 2006
  - <http://www.w3.org/TR/2006/REC-xml-20060816/>
  - Para obtener siempre la última versión:
    - <http://www.w3.org/TR/xml/>

Programación en Internet – Curso 2009-2010

## Ventajas

- Mejora búsquedas → Metadatos
- Facilita intercambio información → Estándar
- Visión estructurada → Tratamiento local
- Integración diferentes fuentes
- Actualizaciones granulares → División de la información
- **Separación contenido (datos) / presentación**
  - Facilita mantenimiento
  - Ofrece múltiples presentaciones

Programación en Internet – Curso 2009-2010

## Ventajas

- Permite crear lenguajes de marcado para dominios específicos:
  - Química: *Chemical Markup Language* (CML)
  - Matemáticas: *Mathematical Markup Language* (MathML)
  - Música: MusicXML
  - Información monetaria: *Open Financial Exchange* (OFX)
  - Recursos humanos (ofertas de trabajo, currículos, etc.): HR-XML

Programación en Internet – Curso 2009-2010

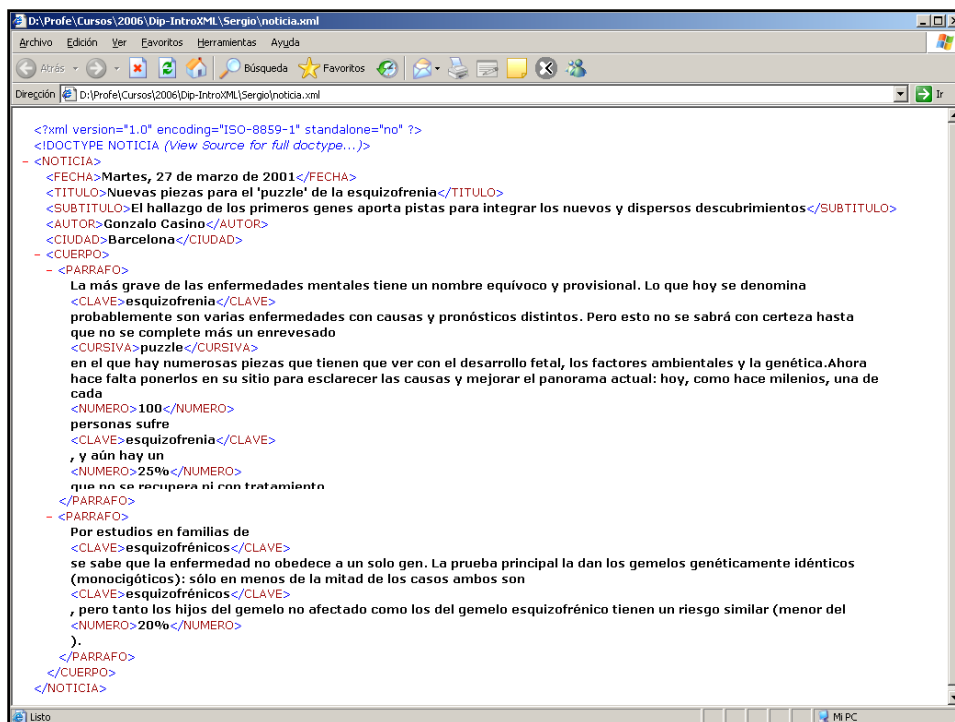
## Ventajas

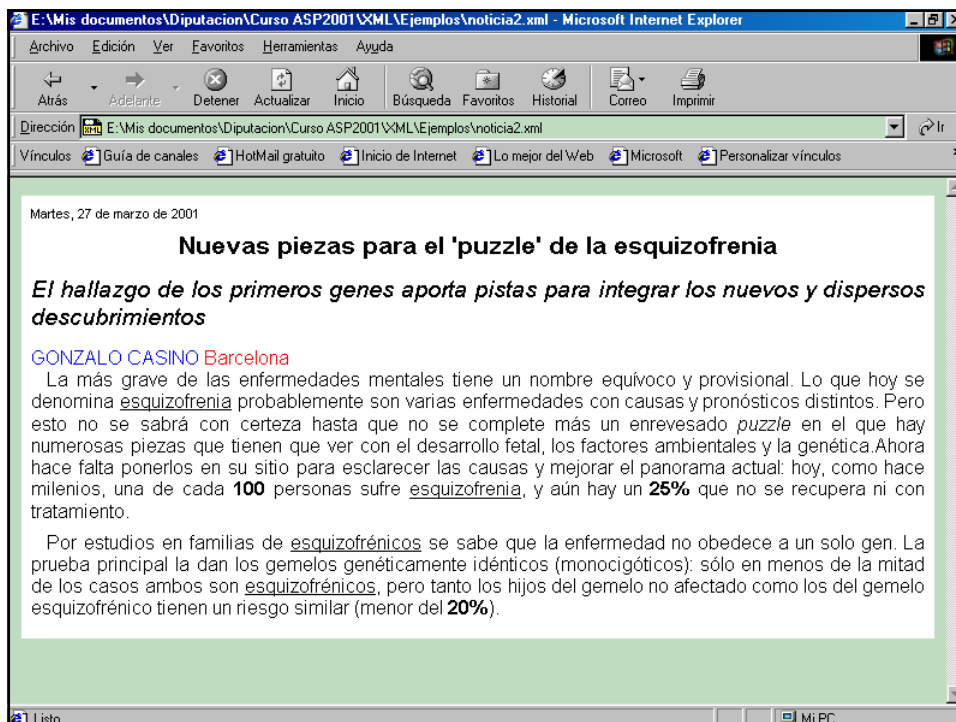
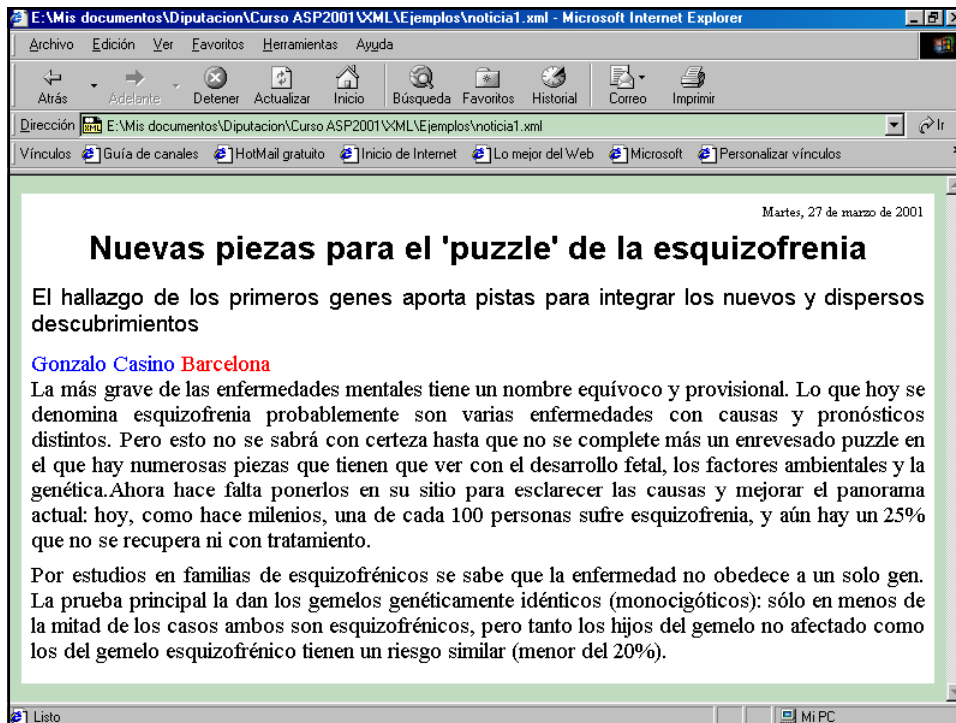
- El contenido se describe a sí mismo:

```
<persona id="p110" sexo="m">
<nombre>Pedro López</nombre>
<direccion>de los Palotes, 120</direccion>
<fnacimiento>30/06/1970</fnacimiento>
</persona>
```

```
-----
p110;m;Pedro López;de los Palotes, 120; 30/06/1970
```

```
-----
0A 11 3C 2E 52 78 90 AA BC EA ...
```





Programación en Internet – Curso 2009-2010

## Objetivos de diseño

1. Se debe de poder usar en Internet
  - Parte de HTML
  - Tan fácil de ver como HTML
2. Amplia variedad de aplicaciones (estructurada, textual, multimedia, etc.)
3. Compatible con SGML (reutilización de herramientas)
4. Sencillo de emplear (más fácil que SGML)
5. Características opcionales: mínimas (no dos formas de hacer lo mismo)

Programación en Internet – Curso 2009-2010

## Objetivos de diseño

6. Lectura y estructura clara (lectura por un humano)
7. Diseño rápido (1996-1998)
8. Diseño formal y conciso (reglas)
9. Documentos fáciles de crear (simple editor ASCII)
10. Concisión etiquetas: mínima importancia (no ahorro etiquetas)



## Qué necesito para usar XML

- Editor ASCII estándar
- Visualización:
  - Microsoft Internet Explorer 5
  - Netscape 6
- Ficheros: .xml
  - Nombres cortos y sencillos
  - No caracteres especiales ni espacios en blanco
- Qué vamos a usar:
  - Microsoft Internet Explorer con XML/XSL Viewer Tools
  - Microsoft XML Notepad 2007
  - ezDTD 1.5
  - Altova XML Spy 2004

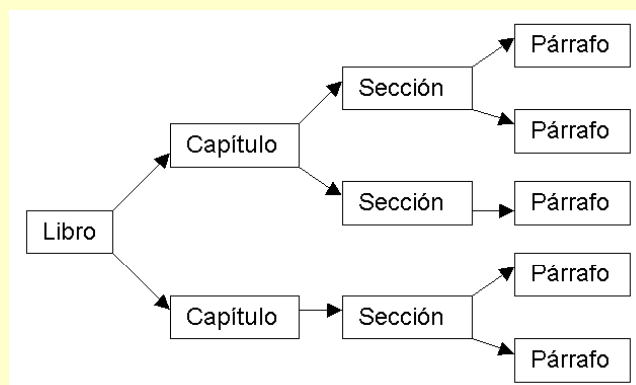
## Estructura de un documento

- Estructura lógica
- Estructura de un documento
- Estructura de un DTD

## Estructura lógica

- Plantilla:
  - Etiquetas
  - Atributos
  - Posición
- Estructura definida por un DTD (*Document Type Definition*)
  - Opcional
  - Define jerarquía (organización) y granularidad (grado de división en partes más pequeñas)

## Estructura lógica



## Estructura lógica

- Documento “bien formado”:  
respetar sintaxis de XML
- Documento válido: respeta DTD

## Estructura de un documento

- Comentarios:  
`<!-- Comentario -->`
- Cabecera (declaración XML) → Instrucción de procesamiento:  
`<?xml  
  version="1.0"  
  encoding="ISO-8859-1"  
  standalone="yes" ?>`
- Si no se incluye:
  - Versión: 1.0
  - Codificación de caracteres: UTF-8
  - Entidades externas: sí

## Estructura de un documento

- Etiquetas → Definen un elemento:  
`<LIBRO></LIBRO>`
- Etiquetas vacías (inicial y final):  
`<LIBRO/>`
- Distinción mayúsculas / minúsculas

## Estructura de un documento

- Estructura jerárquica
  - Organización fija
  - No solapamiento
- Elemento raíz único
- Atributos: siempre `"` o `'`
  - `&quot;` y `&apos;`
- Atributos reservados:
  - `xml:lang`
  - `xml:space: default | preserve`

## Estructura de un documento

- Ejemplo de xml:lang:

```
<p xml:lang="en">The quick brown fox jumps over the  
lazy dog.</p>  
<p xml:lang="en-GB">What colour is it?</p>  
<p xml:lang="en-US">What color is it?</p>  
<sp who="Faust" desc='leise' xml:lang="de">  
  <l>Habe nun, ach! Philosophie,</l>  
  <l>Juristerei, und Medizin</l>  
  <l>und leider auch Theologie</l>  
  <l>durchaus studiert mit heißem Bemüh'n.</l>  
</sp>
```

## Estructura de un documento

- Espacios en blanco:
  - Tabulador
  - Avance de línea
  - Retorno de carro
  - Espacio en blanco
- Normalización caracteres final de línea:
  - Macintosh CR → LF
  - MS-DOS / Windows CR+LF → LF
  - Unix LF

## Estructura de un documento

- Caracteres especiales:

Carácter	Codificación
<	&lt;
>	&gt;
&	&amp;
“	&quot;
‘	&apos;

## Estructura de un documento

- Secciones CDATA: permite que el analizador ignore ciertas secciones del documento
- Sintaxis:  

```
<![CDATA[  
...  
]]>
```
- Solo la cadena final (]]>) se reconoce como marcado y se pueden emplear los caracteres especiales
- Las secciones CDATA no se pueden anidar
- Utilidad: incluir código de script que suele contener los caracteres &, <, >, ‘ y “

## Estructura de un documento

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>Una prueba del CDATA</title>
<script type="text/javascript">
if(a < 5 && a > 1)
  alert("El valor de a no es correcto");
</script>
</head>
<body>
<p>Una prueba del CDATA</p>
</body>
</html>
```

## Estructura de un documento

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>Una prueba del CDATA</title>
<script type="text/javascript">
<![CDATA[
if(a < 5 && a > 1)
  alert("El valor de a no es correcto");
]]>
</script>
</head>
<body>
<p>Una prueba del CDATA</p>
</body>
</html>
```

## Estructura de un DTD

- Representa una gramática
- Declaración:  

```
<! ... >
```
- Cuatro tipos:
  - Elemento (ELEMENT)
  - Atributo (ATTLIST)
  - Entidad (ENTITY)
  - Notación (NOTATION)

## Declaración de elemento

- Define un elemento (etiqueta) nuevo:  

```
<!ELEMENT nombre  
contenido >
```
- Nombre:
  - Primer carácter: letra, '\_', ':'
  - Resto: letra, '\_', ':', dígitos, '.' o '-'



Programación en Internet – Curso 2009-2010

## Declaración de elemento

- Contenido:
  - Nada: `EMPTY`
  - Elementos hijo: `ANY` o modelo de grupo
  - Texto
  - Mezcla

Programación en Internet – Curso 2009-2010

## Declaración de elemento

- Modelo de grupo:
  - Secuencia de aparición: `,` `|`
  - Secuencia de cantidad: `?` `+` `*`
- Texto: `#PCDATA`

## Ejemplos

```
<!ELEMENT algo (a, b?, c+, d*)>
```

```
<!ELEMENT algo ((a | b), c?)>
```

```
<!ELEMENT algo ((a | b)+, c?)>
```

```
<!ELEMENT algo ((a, b)+, (c | d)+)>
```

## Declaración de elemento

- Modelo de contenido mixto:
  - #PCDATA primer elemento
  - Conectores de elección (|)
  - Grupo opcional y repetible (\*)

## Declaración de atributo

- Define atributos de un elemento (etiqueta):

```
<!ATTLIST elemento
  atributo1 contenido1
  defecto1
  atributo2 contenido2
  defecto2>
```

## Declaración de atributo

- Contenido:
  - CDATA
  - NMTOKEN
  - NMTOKENS
  - ENTITY
  - ENTITIES
  - ID
  - IDREF
  - NOTATION
  - Grupo de tokens

## Declaración de atributo

- `NMTOKEN` (name token) parecidos a `CDATA`, pero sólo aceptan los caracteres válidos para nombrar cosas (letras, números, puntos, guiones, subrayados y los dos puntos)
- `NMTOKENS` lista de `NMTOKEN` separados por espacios en blanco

## Declaración de atributo

- `ENTITY` referencia un fichero externo an external file. Each entity name must match the name of an unparsed entity declared in the DTD.
- `ENTITIES` references a list of entity references, employing syntax separated by blank spaces.

## Declaración de atributo

- NOTATION: su valor se ajusta a una notación declarada

```
<!ATTLIST mensaje fecha NOTATION (ISO-  
DATE | EUROPEAN-DATE) #REQUIRED>
```

## Declaración de atributo

- Valor por defecto (opcional):
  - #REQUIRED → El atributo es obligatorio
  - #IMPLIED → El atributo es opcional
  - #FIXED "valor" → El atributo tiene un valor fijo
  - "valor" → Valor por defecto

## Ejemplos

```
<!ATTLIST img
  src CDATA #IMPLIED
  align (left | center | right)
  size NMTOKEN #REQUIRED>

<!ATTLIST alumno
  nombre CDATA #REQUIRED
  matricula ID #REQUIRED>
```

## Declaración de entidad

- Asocia un nombre con un fragmento de contenido
- Ventajas:
  - Facilita la escritura
  - Disminuye los posibles errores
  - Clarifica la estructura
  - Facilita el mantenimiento
- Tres tipos:
  - Internas
  - Externas
  - Paramétricas

## Declaración de entidad

- Internas:
  - `<!ENTITY entidad "valor">`
  - `&entidad;`
- Externas:
  - `<!ENTITY entidad SYSTEM "fichero.xml">`
  - `&entidad;`

## Declaración de entidad

- Entidades predefinidas:
    - `<!ENTITY lt "&#38;#60;">`
    - `<!ENTITY gt "&#62;">`
    - `<!ENTITY amp "&#38;#38;">`
    - `<!ENTITY apos "&#39;">`
    - `<!ENTITY quot "&#34;">`
- Escapar

## Declaración de entidad

- Paramétrica: asocia un nombre con un fragmento de DTD que se repite varias veces:

```
<!ENTITY % nombre contenido>
```

- Cuando se usa  $\Rightarrow$  sustitución:

```
%nombre ;
```

## Ejemplos

```
<!ENTITY % TextAlign "align (left|center|right|justify)
#IMPLIED">
```

```
<!ATTLIST div
  %attrs;
  %TextAlign;
>
```

```
<!ATTLIST p
  %attrs;
  %TextAlign;
>
```

```
<!ATTLIST h1
  %attrs;
  %TextAlign;
>
```



## Declaración de notación

- Especifica tipo ficheros binarios referenciados:

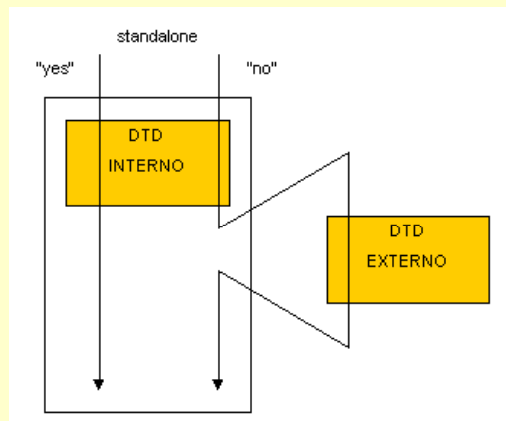
```
<!NOTATION GIF87A SYSTEM  
"GIF" >
```

## Colocación en un documento

- Al principio:
    - Incluir el DTD
- ```
<!DOCTYPE NOTICIA [...] >
```
- Referenciar el DTD
    - PUBLIC: global, un estándar
    - SYSTEM: local, definido por el usuario
- ```
<!DOCTYPE NOTICIA SYSTEM  
"noticia.dtd" >
```
- Una combinación

## Orden de procesamiento

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
```



## DTDs ya definidos

- Book XML DTD
- Channel Definition Format (CDF)
- Chemical Markup Language (CML)
- Genealogical Data in XML (GedML)
- Mathematical Markup Language (MathML)
- Open Software Description (OSD)
- Resource Description Framework (RDF)
- Web Distributed Data Exchange (WDDX)

Programación en Internet – Curso 2009-2010

## Validación de un documento

- Documento bien formado → XML
- Documento válido → DTD

