Tecnologías XML y Web Semántica

Departamento de Informática
Universidad de Oviedo

Sesión 1
Lenguaje XML

Departamento de Informática
Universidad de Oviedo

Esquema de la Sesión

- (9-11h) Primera parte: Introducción a XML
  - Definición de XML
  - DTDs
  - Ventajas/Inconvenientes

  (11-11:30h) Descanso
  - Creación y validación documentos XML con DTDs
- (12-13h) Segunda Parte: Vocabularios XML
  - Espacios de Nombres
  - XML Schema
  - Algunos vocabularios XML

  (13-13:30h) Descanso
  - Creación/Validación documentos XML con Schema

Marcado de documentos

Orígenes: Industria de la publicación

Se usaban marcas para indicar cómo componer el documento

Sólo texto

```
ALBA Abril de 1915 Granada Mi corazón oprimido
siente junto a la alborada el dolor de sus
amores y el sueño de las distancias.
```

Texto marcado

```
]ALBA[  ← Título, negrita, centrado, 14pt
]Abril de 1915[← SubTítulo, negrita, centrado
]Granada[← SubTítulo cursiva, centrado
]Mi corazón oprimido [← Verso, 10pt
]siente junto a la alborada [← Verso
]el dolor de sus amores [← Verso
]y el sueño de las distancias. [← Verso
```

Resultado

ALBA

Abril de 1915

Granada

Mi corazón oprimido
siente junto a la alborada
el dolor de sus amores
y el sueño de las distancias.

Marcado Descriptivo

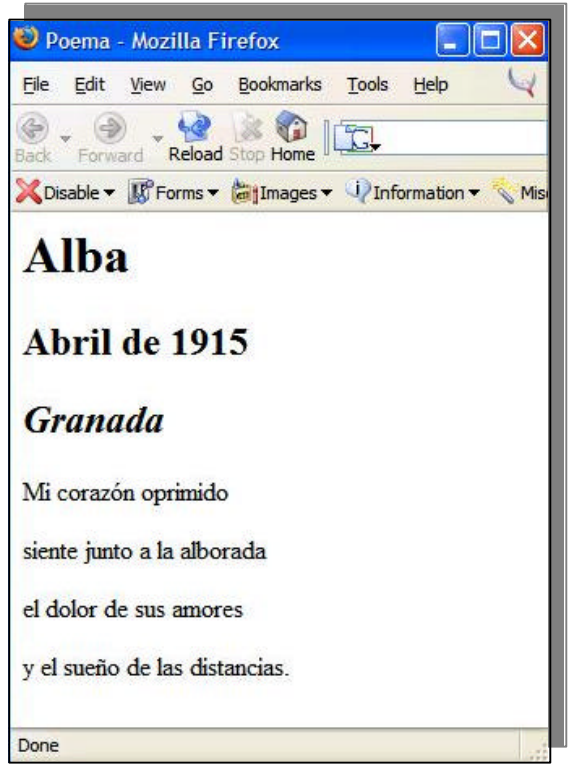
El marcado **no** es la información que contiene el documento

Marcado = **información acerca del documento** = **meta-información**

Lenguajes de Marcado descriptivo: Incluyen marcas que describen cómo procesar el documento

Ejemplo: HTML

```
<html>
<head><title>Poema</title></head>
<body>
<h1>Alba</h1>
<h2>Abril de 1915 </h2>
<h2><i>Granada</i></h2>
<p>Mi corazón oprimido</p>
<p>siente junto a la alborada</p>
<p>el dolor de sus amores</p>
<p>y el sueño de las distancias. </p>
</body>
</html>
```



Marcado Generalizado

Sintaxis común que facilita la creación de lenguajes descriptivos

También pueden incluir reglas de aplicación y otras facilidades

HTML

```
<html>
<head><title>Poema</title></head>
<body>
<h1>Alba</h1>
<h2>Abril de 1915 </h2>
<h2><i>Granada</i></h2>
<p>Mi corazón oprimido</p>
<p>siente junto a la alborada</p>
<p>el dolor de sus amores</p>
<p>y el sueño de las distancias.</p>
</body>
</html>
```

Otras marcas...(misma sintaxis)

```
<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>
```

Evolución del Marcado Generalizado

(70-) GML (*Generalizared Markup Language)*

- Desarrollado en IBM *(Goldfrab, Mosher, Lorie)*

(86) SGML (*Standard Generalized Markup Language*)

- Estándar ISO
- Utilizado en intercambio de documentos
- Permite separar presentación de contenido
- Muy flexible (vocabularios específicos para cada aplicación)
- HTML nace como vocabulario de SGML

(96) XML (*eXtensible Markup Language*)

- Simplificación de SGML
- Objetivo = Aplicación en Internet

Problemas de SGML

SGML era muy complejo

Para identificar la estructura era necesario el DTD...

...pero en la Web no siempre está disponible

Ejemplo (especificación listas HTML)

La etiqueta final es obligatoria

```
<!ELEMENT OL  - -  (LI)+  >
<!ELEMENT LI  - O  (%flow;)* >
<!ELEMENT P   - O  (%inline;)* >
```

La etiqueta final es opcional

<P> párrafo <UL> lista </UL>

?

<P> párrafo
</P>
<OL> lista
</OL>

<P> párrafo
<OL> lista
</OL>
</P>

a ambigüedad se deshace mediante el DTD

En Internet no siempre podremos acceder al DTD

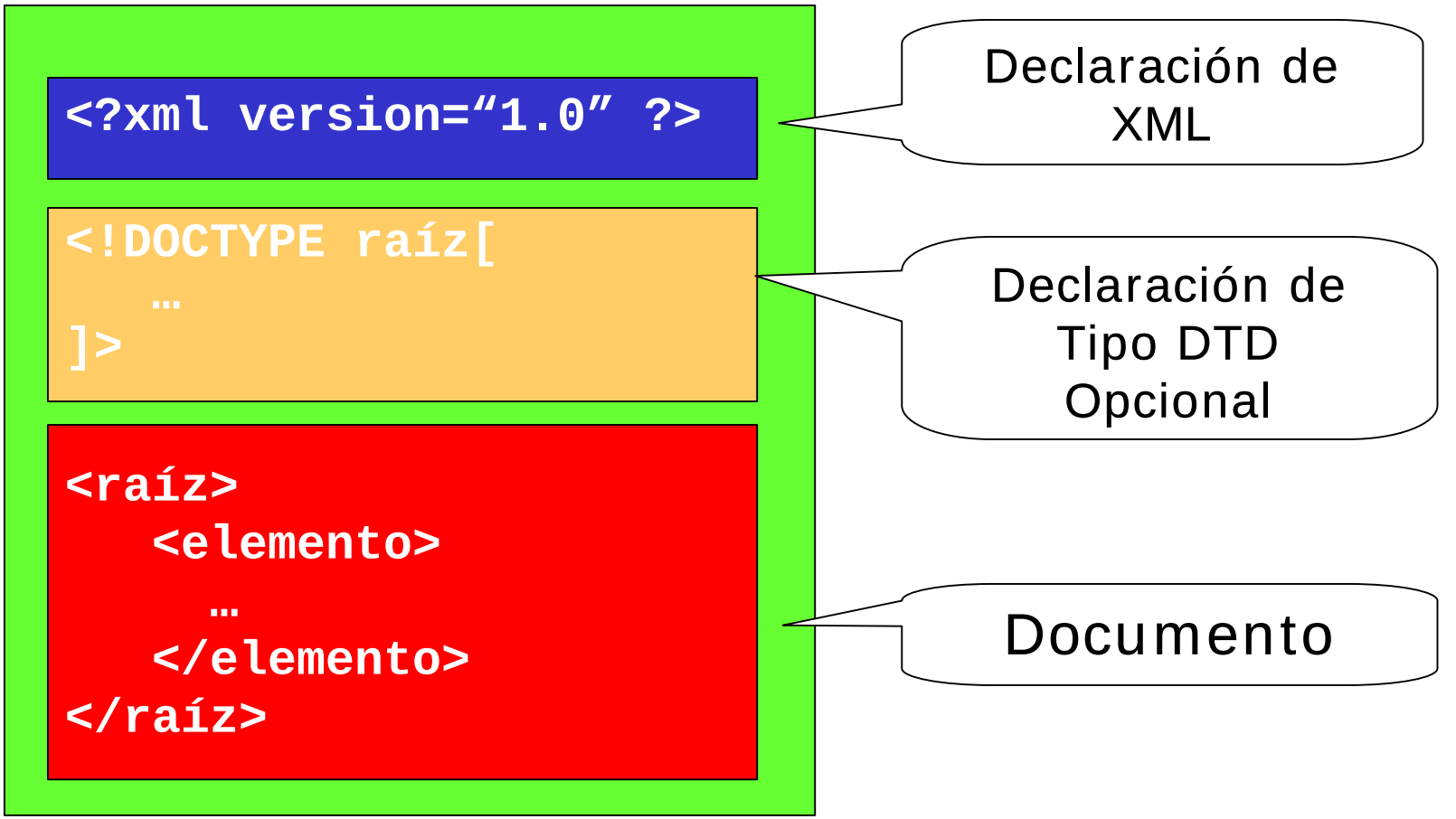
Desarrollado por T. Bray, J. Paoli, C. M. Sperberg-McQueen (1995)

Objetivos de diseño (según la especificación)

1. Utilizable en Internet
2. Soporte a gran variedad de aplicaciones
3. Compatible con SGML
4. Debe ser fácil escribir programas que procesen XML
5. Número de características opcionales = Mínimo
6. Documentos legibles por personas
7. El diseño de XML debe poder hacerse rápidamente
8. El diseño de XML debe ser formal y conciso
9. La creación de documentos XML debe ser fácil
10. La concisión de las marcas XML no tiene importancia (es preferible la claridad a la brevedad)

20% de características de SGML $\Leftrightarrow$ 80% de funcionalidad de SGML

Curiosidad: Especificación de XML = 26 páginas, de SGML > 500


<?xml version="1.0" ?>
<!DOCTYPE raíz[
…
]>
<raíz>
<elemento>
…
</elemento>
</raíz>
Declaración de XML
Declaración de Tipo DTD Opcional
Documento

Declaración de XML

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

version: Actual = 1.0

Borrador de versión 1.1

Mayor compatibilidad con Unicode

Identificadores: Permite cualquier carácter Unicode

encoding: UTF-8, UTF-16, iso-8859-1, etc.

standalone: Indica si el documento no hace referencias a entidades externas

Los documentos consisten en una serie de datos marcados mediante etiquetas

Las etiquetas describen la estructura del documento

Un elemento = grupo formado por etiqueta inicial, etiqueta final y contenido entre ambas. La etiqueta inicial puede incluir atributos.

```
<etiqueta attributo="valor">……</etiqueta>
```

Distinción minúsculas/mayúsculas

Elemento vacío: Entre la etiqueta inicial y final no hay información:

```
<etiqueta attributo="valor"></etiqueta>
```

ß

```
<etiqueta atributo="valor"/>
```

Se pueden anidar elementos

```
<externo>
  <interno>texto</interno>
</externo>
```

...pero no se pueden entrelazar:

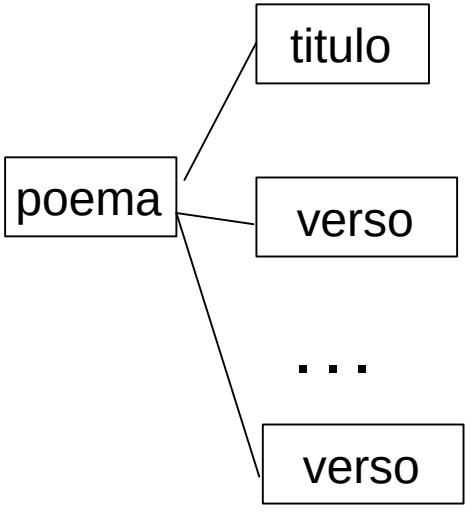
```
<externo>
  <interno>texto</externo>
</interno>
```

Documentos XML
Estructura General

Sólo puede haber un único elemento raíz

Cada documento XML equivale a un árbol

```
<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>
```



Atributos

Cada elemento puede contener atributos en la etiqueta inicial

```
<pizza nombre="Margarita" precio="6">
  . . .
</pizza>
```

El orden de los atributos no es significativo

No puede haber 2 atributos con el mismo nombre

Atributos predefinidos:

xml:lang: Especifica el idioma.

Por ejemplo: en (inglés), sp (español)

xml:space: Especifica cómo tratar el espacio en blanco.

Valores: preserve = Mantenerlo

default = Permitir a la aplicación que lo trate como quiera.

Documentos XML
Otras características

Comentarios

```
<!-- el texto de un comentario
        no es analizado -->
```

Caracteres especiales: No pueden incluirse directamente

```
<código>
  if x &lt; 4 then x:=x + 1;
</código>
```

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

Secciones CDATA

Si se desea introducir código sin analizar

```
<código>
  if x < 3 && x > 4 then
     print “Hola”
</código>
```

```
<código>
  if x &lt; 3
   &amp;amp;&amp;amp; x &gt; 4 then
     print &amp;quot;Hola&amp;quot;
</código>
```

```
<código>
<![CDATA[
if x < 3 && x > 4 then
      print “Hola”
]]>
</código>
```

Documentos XML
Instrucciones de Procesamiento

Es posible incluir instrucciones que indican al procesador alguna acción a realizar

Sintaxis: `<?aplicación datos ?>`

Pueden utilizarse para asociar una hoja de estilos al documento:

```
<?xml-stylesheet type="text/xsl" href="hoja.xsl"?>
```

...o para otros propósitos especiales

En realidad la declaración de documento es una instrucción de procesamiento

```
<?xml version="1.0" ?>
```

Documento bien formado

Sigue las reglas sintácticas

Importante:

- Contiene un único elemento raíz
- Todas las etiquetas están correctamente anidadas

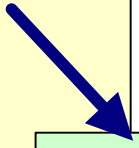
```
<pizzas>
 <pizza nombre="Margarita" precio="6">
  <ingrediente nombre="Tomate" />
  <ingrediente nombre="Queso" />
 </pizza>
</pizzas>
```

```
<pizzas>
 <pizza nombre="Margarita" precio="6">
   <ingrediente nombre="Tomate" >
</pizzas>
```

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</vers
<verso>el dolor de sus amores</verso>
<verso>y el sueño de las distancias. </
</poema>
```



pizzas.dtd

```
<!ELEMENT poema (titulo,autor?,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 de tipo de documento (DTD)

DTD interno

```
<?xml version="1.0"?>
<!DOCTYPE poema [
 <!ELEMENT poema (titulo, autor?, verso+)>
. . .
]>
<poema>. . . </poema>
```

También es posible especificar un DTD externo y añadir definiciones locales

DTD externo

SYSTEM (DTDs de ámbito local)

```
<?xml version="1.0"?>
<!DOCTYPE poemaSYSTEM "http://www.poemas.org/poema.dtd" >
<poema>
. . .
</poema>
```

PUBLIC (DTDs compartidos por diversas organizaciones)

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

Tipos de declaraciones

ELEMENT

- Elementos del documento XML

ATTLIST

- Lista de atributos de un elemento

ENTITY

- Entidades (≈variables o macros)

NOTATION

- Definen tipos de contenidos
- Facilitan la inclusión de formatos binarios (imágenes, vídeos, sonidos, ...)

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

```
<!ELEMENTpoema (titulo,autor?,verso+)>
<!ELEMENT publicacion (poema | novela | ensayo) >
<!ELEMENT autor EMPTY>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT sección (título, (contenido | sección+))>
<!ELEMENT p (#PCDATA | a | ul | em )* >
```

Recursividad

PCDATA = Parsed Character Data
Indica que los datos son analizados buscando etiquetas

DTD
Atributos

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

Valor de los Atributos

#REQUIRED	Obligatorio
#IMPLIED	Opcional
#FIXED	Constante
Valor	Valor por defecto

```
<!ATTLIST poema fecha CDATA #REQUIRED
                  lugar CDATA #IMPLIED>

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

<!ATTLIST persona código   ID   #REQUIRED>

<!ATTLIST autor código IDREF #REQUIRED>

<!ATTLIST enEstantería (sí|no) “sí” >

<!ATTLIST impuesto tipo CDATA #FIXED “IVA”>
```

```
<poema lugar=“Oviedo” fecha="2004">
  <precio moneda=“euros” valor=“20” />
  <autor código="35" />
</poema>

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

<impuesto tipo=“IVA” />
```

DTD
Entidades Generales

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

Se denotan por **`&entidad;`**

No se admite recursividad

```
<!ENTITY pepe “Jose Luis López López”>
<!ENTITY humanidades “<lugar>Biblioteca de Humanidades</lugar>” >
```

```
<poema autor=”&pepe;" >
  &humanidades;
</poema>
```

```
<pizza nombre=” Jose Luis López López" >
  <lugar>Biblioteca de Humanidades</lugar>
</pizza>
```

Entidades numéricas: Código numérico del carácter

∀ ∀ ∃ ∃

Entidades predefinidas: Permiten incluir etiquetas sin analizar

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

DTD

Entidades externas

Permiten usar archivos externos

Formar un documento XML a partir de varios

poema1.xml

```
<poema fecha="1915" lugar="Granada">
<titulo>Alba</titulo>
<verso>Mi corazón oprimido</verso>
...
<verso>de las distancias.</verso>
</poema>
```

poema2.xml

```
<poema fecha="l920" lugar="Zujaira">
<titulo>CANTOS NUEVOS</titulo>
<verso>Dice la tarde</verso>
...
<verso>y suspira el viento.</verso>
</poema>
```

poemas.xml

```
<!DOCTYPE poemas [
 <!ENTITY p1 SYSTEM "p1.xml">
 <!ENTITY p2 SYSTEM "p2.xml">]>
<poemas>
&p1;
&p2;
</poemas>
```

poemas.xml

```
<poemas>
<poema fecha="1915" lugar="Granada">
<titulo>Alba</titulo>
<verso>Mi corazón oprimido</verso>
...
<verso>de las distancias.</verso>
</poema>
<poema fecha="l920" lugar="Zujaira">
<titulo>CANTOS NUEVOS</titulo>
<verso>Dice la tarde</verso>
...
<verso>y suspira el viento.</verso>
</poema>
</poemas>
```

Entidades externas no analizadas

El contenido de los ficheros es analizado (deben seguir sintaxis XML)

Para incluir ficheros externos sin analizar se utiliza NDATA (Notation Data)

Aplicaciones: Incluir formatos binarios

Mediante NOTATION se puede indicar qué aplicación se hará cargo de dichas notaciones

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

DTD

Entidades Parámetro

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;
```

Ejercicios

Creación de ficheros XML y validación

Procesadores de XML

- Chequean que está bien formado
- Validan

Productos

- Visuales: XML Writer, XML Spy, ...
- Modo texto: xmllint, msxml, ...

xmllint forma parte de la librería libxml de GNOME

```
xmllint --valid --noout fichero.xml
```

Validar
Si no se pone nada,
Chequea que está bien formado

No muestra resultado
Si no hay mensajes ⇒ OK

Discusión sobre XML: Ventajas

Es un formato estructurado

- Contiene información y meta-información

Ha sido diseñado específicamente para Internet

- Soportado por visualizadores y servidores

Numerosas herramientas de procesamiento

Legible por personas humanas

Admite la definición de vocabularios específicos

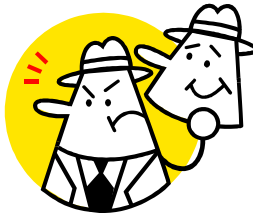
Separa contenido del procesamiento y visualización

Aumenta la seguridad mediante la validación de documentos

Formato abierto, respaldado por numerosas organizaciones

Una vez definido un DTD común, facilita intercambio de información

Discusión sobre XML: Inconvenientes



Puede requerir demasiado espacio, ancho de banda y tiempo de procesamiento

- Documentos largos con mucha información redundante

Es una sintaxis de documentos, no un lenguaje de programación

```
int main(void) {
 printf("Hola");
 return 0;
}
```

```
<function name="main" type="int">
<arg type="void" />
<call function="printf">
  <param>Hola</param>
</call>
<return value="0"/>
</function>
```

Es posible crear formatos y vocabularios propietarios

Puede fomentar la proliferación de vocabularios específicos

Poco eficiente como lenguaje de almacenamiento de bases de datos

Bueno para texto,

- malo para datos binarios

```
<?xml version="1.0">
<imagen formato="base64">
DS34JSCDF029876D76523981DFNDF3F2134F5FD019A
FGF23DAND345CD2135911943DCBKAPFGDAJJK32A10
....
</imagen>
```

Vocabularios de XML

Existe una gran cantidad de vocabularios (o *aplicaciones*) XML

Conviene utilizar vocabularios estándar

- XHTML: Versión de HTML adaptada a XML
- WML (Wireless markup language): Teléfonos móviles
- SVG (Scalable Vector Graphics): Gráficos Vectoriales
- XSL: Hojas de estilo para documentos XML
  - Formado por XPath, XSLT y XSL-FO
- SMIL: Multimedia
- VoiceXML: Portales de Voz
- MathML: Fórmulas Matemáticas
- X3D: Representaciones tridimensionales
- ...etc


Dispositivo del Cliente
HTML
WML
XSL-FO
PDF
VoiceXML
XML
XSLT
XSLT
XSLT
XSLT
XSLT
. . .

Ejemplo: Fichero de Alumnos

Partiendo de un fichero de Alumnos
Se obtienen ficheros en XHTML, SVG, PDF, WML y X3D

Vocabularios XML

Es conveniente *no reinventar la rueda*

Antes de crear un vocabulario, chequear existentes

- Los vocabularios también se llaman *aplicaciones*

Preferiblemente, chequear los desarrollados por organismos de estandarización

Algunos almacenes:

- http://xml.coverpages.org/xmlApplications.html
- http://www.xml.org/xml/registry.jsp

Selección de Enlaces

Página del consorcio: http://www.w3c.org

En español: http://www.it.uc3m.es/~xml/enlaces.html

Especificación anotada: http://www.xml.com/axml/testaxml.htm

XML en industria: http://www.xml.org

Diseño de vocabularios XML: http://www.xmlpatterns.com

Tutoriales: http://www.w3schools.com

Artículos de XML:

- http://www.topxml.com
- http://www.xmlpatterns.com

Software de XML

- http://www.xmlsoftware.com
- http://www.xmlhack.com
- http://www.garshol.priv.no/download/xmltools/

Fin de la Presentación