



# **XML: eXtensible Markup Languaje**

## **Tecnologías XML** Introducción

**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**

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- **Lenguajes de marcas**
  - Introducción a XML
  - Lenguajes derivados de XML
  - Documentos XML
  - Documentos XML bien formados
  - Documentos XML válidos
  - Ventajas de XML
  - Inconvenientes de XML
  - Bibliografía
  - Referencias Web
  - Ejercicios resueltos
  - Ejercicios propuestos

- Desde el HTML a XHTML
  - HTML
  - XHTML
  - Hojas de Estilo (CSS)
- Desde SGML a XML
  - SGML
  - XML
    - DTD
    - Esquemas XML

# Software y estándares para la Web

## Lenguajes de Marcas (II): HTML en su inicio


**AlCapone.html**

```
<html>
<head>
<title>Pizzeria Al Capone</title>
</head>
<body bgcolor="blue" text="yellow"
      link="red" vlink="white">

<h1>Pizzería Al Capone</h1>
<p>Lista de enlaces</p>
<ul>
<li><a href="Pizzas.html">
      Tipos de Pizzas</a></li>
<li><a href="http://www.mafia.it">
      Patrocinadores</a></li>
<li><a href="#Contacto">Contacto</a></li>
</ul>

<h2><a name="Contacto">Contacto</a></h2>
<p><font color="red">Dirección:</font>
      C/ Génova Nº 3, Oviedo, España</p>
<p><font color="red">Teléfono:</font>
      985203040</p>
</body>
</html>
```

**Página visualizada**



**Problema:**  
Presentación y contenido  
mezclados

**Cabecera**

**Cuerpo**

**Lista**

**Enlaces**

**Detalles de presentación**



# Software y estándares para la Web

## Lenguajes de marcas (III): Hojas de estilo CSS

Enlace a hoja de estilo

Sin aspectos visuales

Identificación elementos

AlCapone.html


```
<html>
<head>
<title>Pizzeria Al Capone</title>
<link rel="stylesheet" href="pizzeria.css">
</head>
<body>
<h1>Pizzería Al Capone</h1>
<p>Lista de enlaces</p>
<ul>
<li><a href="Pizzas.html">
  Tipos de Pizzas</a></li>
<li><a href="http://www.mafia.it">
  Patrocinadores</a></li>
<li><a href="#Contacto">
  Contacto</a></li>
</ul>

<h2><a name="Contacto">Contacto</a></h2>
<p><span class="item">Dirección:</span>
  C/ Génova Nº 3, Oviedo, España</p>
<p><span class="item">Teléfono:</span>
  985203040</p>
</body>
</html>
```

pizzeria.css

```
body { color : yellow;
        background: blue
    }
a:link { color: red }
a:visited { color: white }
span.item { color : red }
```

Página visualizada



# Software y estándares para la Web

## Lenguajes de Marcas (IV): Reutilización de estilos



pizzas.html

```
<html>
<head>
<title>Tipos Pizzas</title>
<link rel="stylesheet" href="pizzeria.css">
</head>
<body>
<h1>Pizzas del Restaurante Al Capone</h1>
<table><caption>Tipos de Pizzas</caption>
<th>
<td>Pizza</td><td>Ingredientes</td><td>Precio</td></th>
<tr><td>Barbacoa</td>
  <td>Salsa barbacoa, Mozzarella, Pollo, Bacon, Ternera</td>
  <td>8 &euro;</td></tr>
  ...
<tr><td>Margarita</td>
  <td>Tomate, Jamón, Queso</td>
  <td>6 &euro;</td>
</tr>
</table>
</body>
</html>
```

Misma  
apariencia



**Problema:**

¿significado de las marcas?  
¿Procesamiento automático?

# Software y estándares para la Web

## Lenguajes de Marcas (V): XHTML

Grado en  
Ingeniería  
Informática  
del Software



- HTML (1991). HyperText Markup Language
- HTML 2.0 (1995)
- HTML 4.0 (1997)
- XHTML (eXtensible HyperText Markup Language)
- XHTML es HTML expresado como XML válido
- XHTML 1.0 fue una recomendación del W3C (26-Enero-2000)
- XHTML 1.1 fue una recomendación del W3C (31-Mayo-2001)
- HTML 5 (2008)
- HTML 5.1 (2016)
- HTML 5.2 (2017)
- Draft HTML 5.3 (9/octubre/2024)  
<https://html.spec.whatwg.org/>
- El estándar conocido como **XHTML5.3** está siendo desarrollado como una adaptación a XML de la especificación **HTML5.3 (suspendido temporalmente)**
- Si HTML se define como un lenguaje derivado de XML, se pueden utilizar fácilmente herramientas creadas para procesamiento de documentos XML genéricos





# Software y estándares para la Web

## Lenguajes de Marcas(VI): de SGML a XML

Grado en  
Ingeniería  
Informática  
del Software

- **SGML** (*Standard Generalized Markup Language*)
  - *Descendiente de IBM GML de la década de 1960*
  - *Utilizado para el intercambio de documentos*
  - *Principio: Separar contenido de la forma de representarlo*
  - *Permite utilizar un conjunto de marcas específico para cada aplicación*
  - *Definido como estándar ISO en 1986*
  - *XML es un subconjunto de SGML*
  - *Problema de SGML: Demasiado complicado para su adopción en la Web*
- **XML** (*eXtended Markup Language*)
  - Desarrollado por el consorcio Web (1995)
  - Versión simplificada de SGML
  - Es un metalenguaje
  - Objetivos:
    - Estándar de intercambio de información a través de la Web
    - Formato abierto, independiente de la plataforma
    - Permite utilizar vocabularios específicos de una aplicación
    - Permite la auto-descripción de dichos vocabularios (documentos auto-descritos)
    - Las aplicaciones pueden descubrir el formato de la información y actuar en consecuencia

# Software y estándares para la Web

## Lenguajes de Marcas (VII): SGML (*Standard Generalized Markup Language*)

Grado en  
Ingeniería  
Informática  
del Software

- Estándar internacional para la definición de la estructura y el contenido de documentos
  - **SGML**: norma ISO 8879:1986(E)
- Es anterior a la Web
- Utiliza un documento tipo DTD (*Document Type Definition*) que definen su estructura o gramática
- La diferencia de SGML con otros lenguajes de marcas es que se pueden utilizar otros caracteres sin ser “<>” para definir las etiquetas, por ejemplo el “-”
- Al ser tan genérico, es difícil de manejar
- Lenguaje de **tipo 2** o libre de contexto en la clasificación de Chomsky
  - XML es un lenguaje de **tipo 3** o regular. Por eso se puede describir con expresiones regulares.

# Software y estándares para la Web

## Lenguajes de Marcas (VIII): XML (eXtensible Markup Language)

Grado en  
Ingeniería  
Informática  
del Software

- **Metalenguaje** es un lenguaje para definir lenguajes
- **Extensible**: Etiquetas no predefinidas
- Diseñado para **describir datos**
- Puede ser estructurado si se usan **DTDs** para validar contenido y sintaxis
  - **Ventajas**: Los DTDs son breves y escuetos
  - **Inconvenientes**: No tienen tipos de datos ni rangos. Abusan del uso de las cadenas de texto para definir datos
- **Schemas** son similares a los DTD's, pero usan un formato XML.
  - Definen detalladamente los documentos XML usando tipos y rangos
  - Son mucho más extensos que los DTDs
- <http://www.w3.org/TR/xml11/>  
(Última versión 29-Septiembre-2006)



# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- **Introducción a XML**
- Lenguajes derivados de XML
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

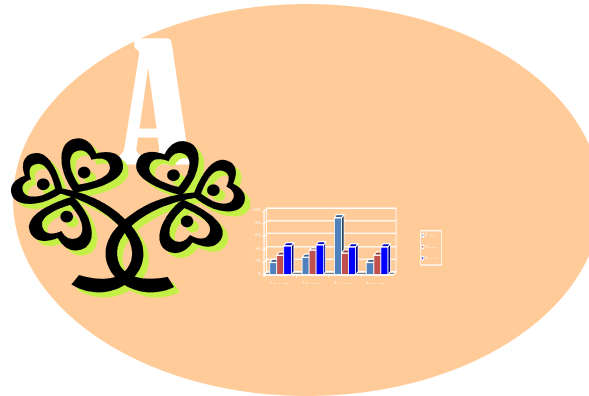
## Introducción a XML: Concepto de documento

Grado en  
Ingeniería  
Informática  
del Software

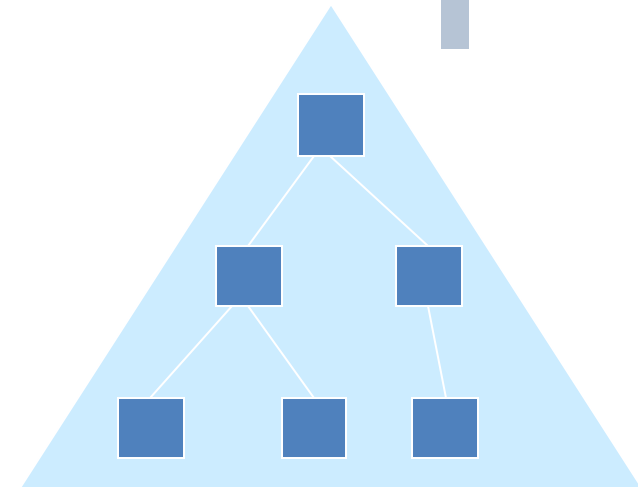
- XML especifica el contenido y la estructura, pero **no la presentación**

Esto es  
el contenido  
del documento

**CONTENIDO**



**PRESENTACIÓN**



**ESTRUCTURA  
DEL ÁRBOL  
DOM (Document  
Object Model)**



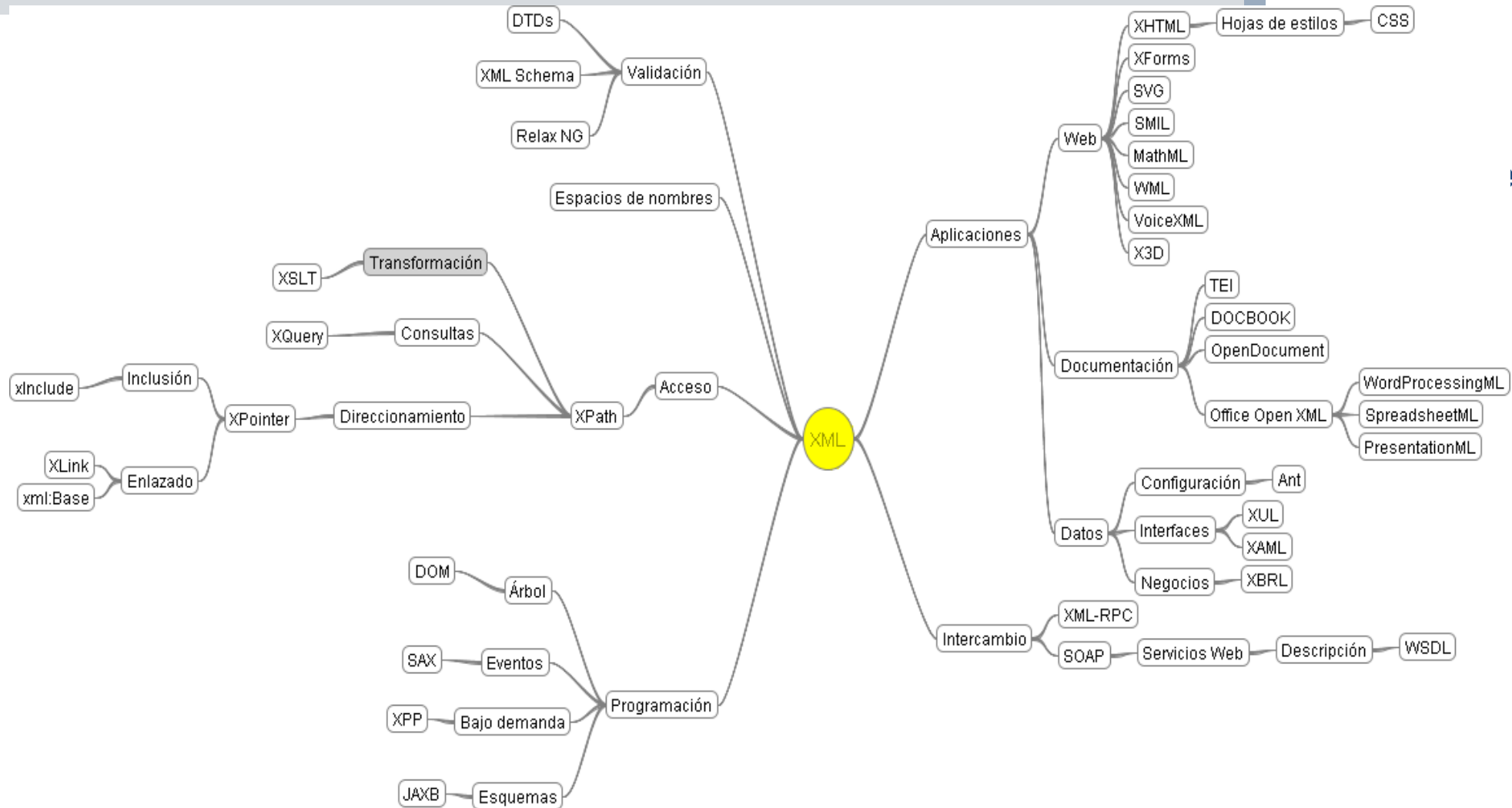
# Software y estándares para la Web

## Introducción a XML: ¿Por qué usar XML?

Grado en  
Ingeniería  
Informática  
del Software

- Un documento XML puede ser fácilmente procesado y sus datos manipulados
- Los documentos XML permiten el **intercambio de información** entre diferentes sistemas
- Los servicios Web usan XML
- XML es la base de muchos estándares en la Web
- Existen APIs para **procesar** esos documentos en JavaScript, Python, Java, C#, C, C++, PHP, etc.
- XML define **datos portables** al igual que Java define código portable

# Introducción a XML: Mapa conceptual XML



## XML es el pilar fundamental de muchas otras tecnologías en la Web

# Software y estándares para la Web

## Introducción a XML: Ejemplo *pizzas.xml* (a)

Grado en  
Ingeniería  
Informática  
del Software

pizzas.xml

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

<pizzas>

<pizza nombre="Barbacoa" precio="8,0€">
<ingrediente nombre="Salsa Barbacoa"/>
<ingrediente nombre="Mozzarella"/>
<ingrediente nombre="Tocineta"/>
<ingrediente nombre="Tenera"/>
</pizza>

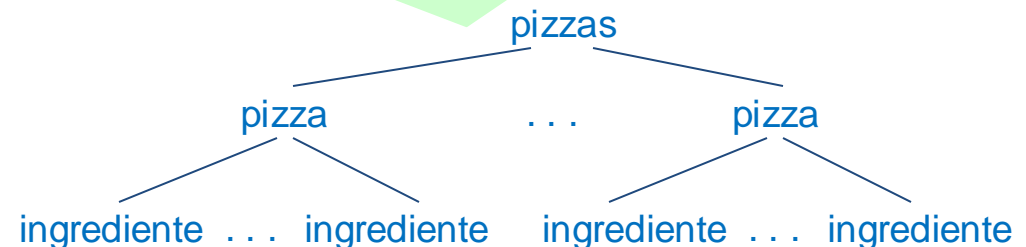
<pizza nombre="Margarita" precio="6,5€">
<ingrediente nombre="Tomate"/>
<ingrediente nombre="Jamón"/>
<ingrediente nombre="Queso"/>
</pizza>
...
</pizzas>
```

DTD = Declaración de Tipo de Documento

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>
```

Estructura del árbol DOM



Las marcas tienen un significado propio de la aplicación

# Software y estándares para la Web

## Introducción a XML: Ejemplo *pizzas.xml* (b)

pizzas.xml

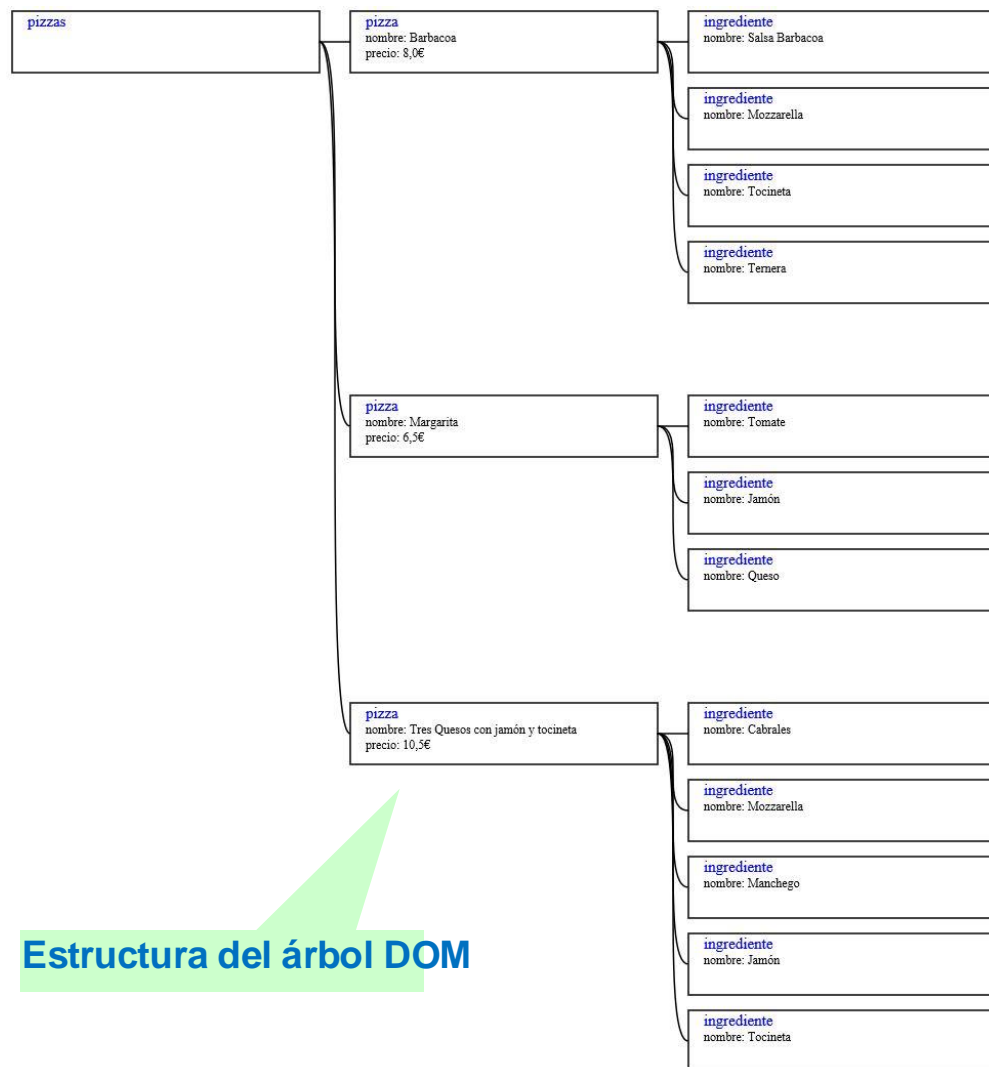
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pizzas SYSTEM "pizzas.dtd">
<pizzas>
  <pizza nombre="Barbacoa" precio="8,0€" >
    <ingrediente nombre="Salsa Barbacoa"/>
    <ingrediente nombre="Mozzarella"/>
    <ingrediente nombre="Tocineta"/>
    <ingrediente nombre="Tenera"/>
  </pizza>
  <pizza nombre="Margarita" precio="6,5€" >
    <ingrediente nombre="Tomate"/>
    <ingrediente nombre="Jamón"/>
    <ingrediente nombre="Queso"/>
  </pizza>
  <pizza nombre="Tres Quesos con jamón y tocineta" precio="10,5€" >
    <ingrediente nombre="Cabrales"/>
    <ingrediente nombre="Mozzarella"/>
    <ingrediente nombre="Manchego"/>
    <ingrediente nombre="Jamón"/>
    <ingrediente nombre="Tocineta"/>
  </pizza>
</pizzas>
```

Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

## Introducción a XML: Ejemplo *pizzas.xml* (c)

Grado en  
Ingeniería  
Informática  
del Software



Estructura del árbol DOM

**xml2svg.exe** trabaja  
en línea de comandos  
del sistema operativo

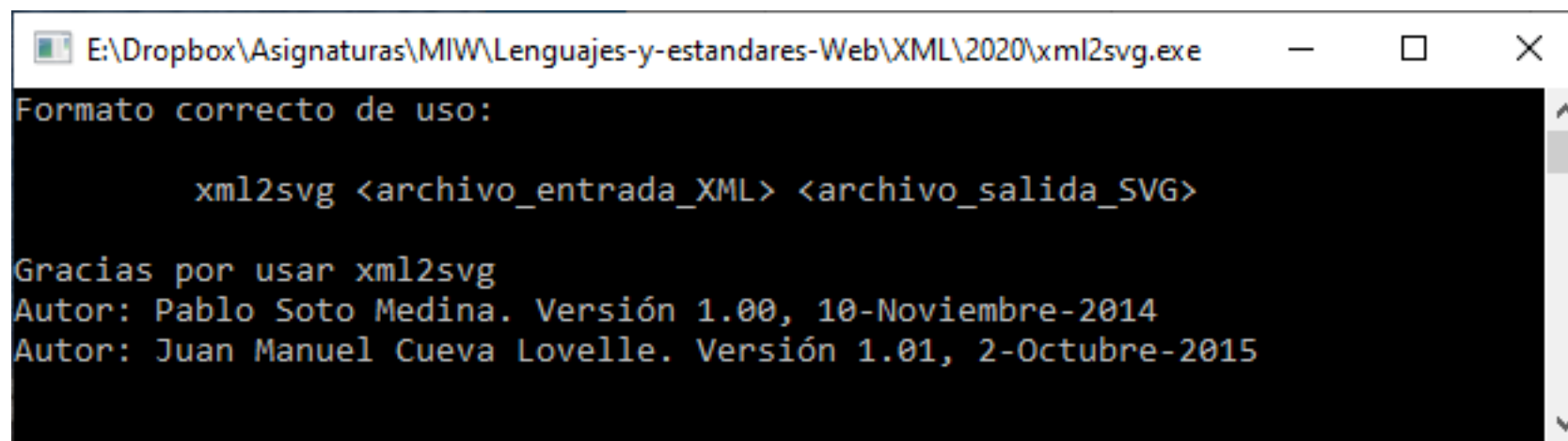
Creado con **xml2svg.exe**



# Software y estándares para la Web

## Introducción a XML: Conversión de *pizzas.xml* a *pizzas.svg* (d)

Grado en  
Ingeniería  
Informática  
del Software



```
E:\Dropbox\Asignaturas\MIW\Lenguajes-y-estandares-Web\XML\2020\xml2svg.exe
Formato correcto de uso:

    xml2svg <archivo_entrada_XML> <archivo_salida_SVG>

Gracias por usar xml2svg
Autor: Pablo Soto Medina. Versión 1.00, 10-Noviembre-2014
Autor: Juan Manuel Cueva Lovelle. Versión 1.01, 2-Octubre-2015
```

**xml2svg.exe** trabaja en línea de comandos del sistema operativo (\*)



```
Símbolo del sistema - xml2svg pizzas.xml pizzas.svg
E:\Dropbox\Asignaturas\UNIOVI\SoftwareYestandaresWeb\2020\XML\01-Ejemplos>xml2svg pizzas.xml pizzas.svg
Done.

Gracias por usar xml2svg
Autor: Pablo Soto Medina. Versión 1.00, 10-Noviembre-2014
Autor: Juan Manuel Cueva Lovelle. Versión 1.01, 2-Octubre-2015
```

(\*) Si el archivo XML está enlazado a un DTD o Scheme, este debe estar en la misma carpeta

# Software y estándares para la Web

## Introducción a XML: xml2svg.exe en Linux (e)

```
cueva@Dron: /mnt/e/Dropbox/Python-Win-02000-Servicios-Web
(base) cueva@Dron:/mnt/e/Dropbox/Python-Win-02000-Servicios-Web$ mono xml2svg.exe
Formato correcto de uso:

    xml2svg <archivo_entrada_XML> <archivo_salida_SVG>

Gracias por usar xml2svg
Autor: Pablo Soto Medina. Versión 1.00, 10-Noviembre-2014
Autor: Juan Manuel Cueva Lovelle. Versión 1.01, 2-Octubre-2015

(base) cueva@Dron:/mnt/e/Dropbox/Python-Win-02000-Servicios-Web$
```

(\*) Si el archivo XML está enlazado a un DTD o Scheme, este debe estar en la misma carpeta

# Software y estándares para la Web

## Introducción a XML: xml2svg.exe en Mac OSX (f)

```
Python-Win-02000-Servicios-Web — -bash — 90x11
(base) MacBook-Air-de-Usuario:Python-Win-02000-Servicios-Web usuario$ mono xml2svg.exe
Formato correcto de uso:

    xml2svg <archivo_entrada_XML> <archivo_salida_SVG>

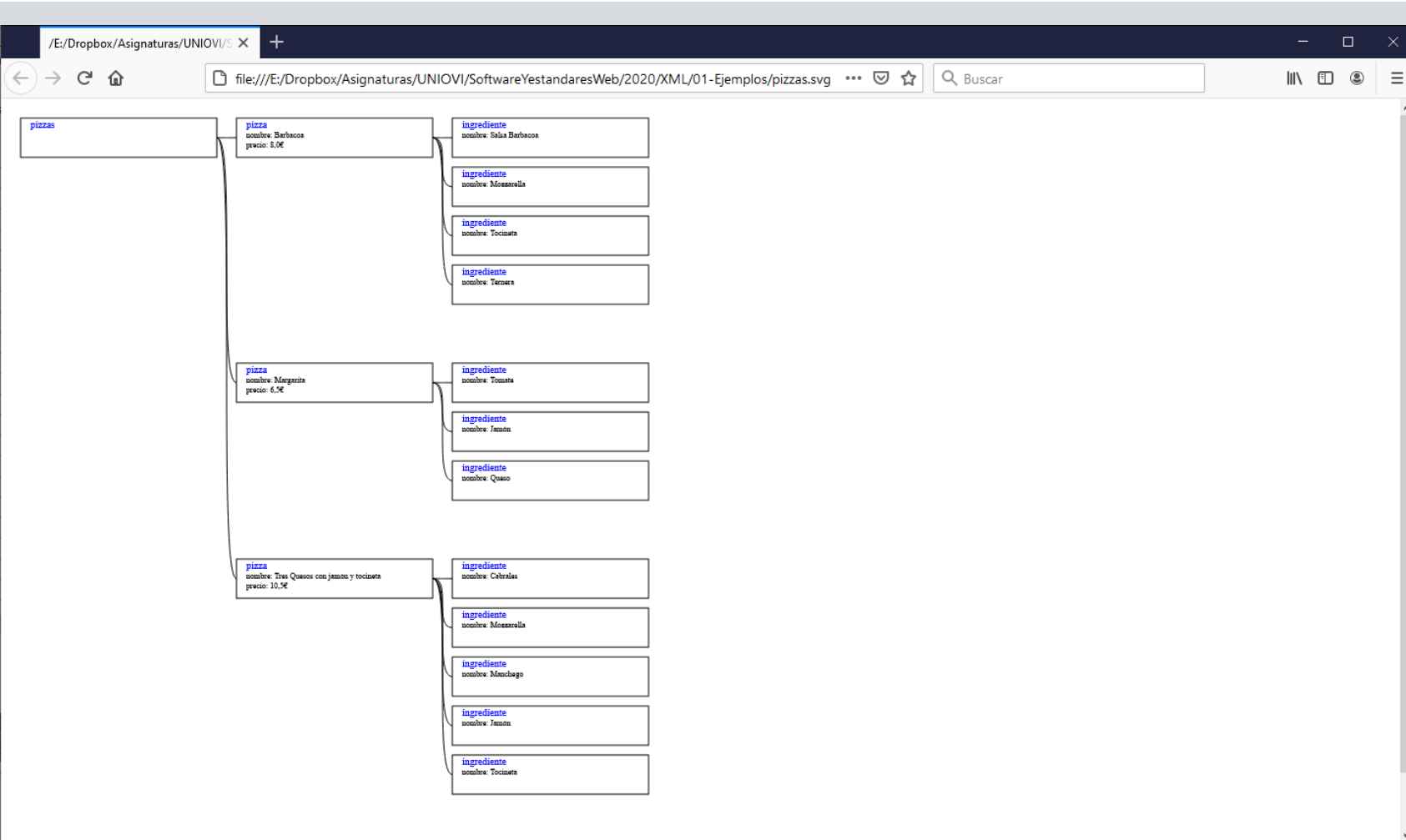
Gracias por usar xml2svg
Autor: Pablo Soto Medina. Versión 1.00, 10-Noviembre-2014
Autor: Juan Manuel Cueva Lovelle. Versión 1.01, 2-Octubre-2015

(base) MacBook-Air-de-Usuario:Python-Win-02000-Servicios-Web usuario$
```

(\*) Si el archivo XML está enlazado a un DTD o Scheme, este debe estar en la misma carpeta

# Software y estándares para la Web

## Introducción a XML: Visualización con un navegador de *pizzas.svg* (g)



Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

## Introducción a XML: Ejemplo *pizzas.dtd* (h)

Grado en  
Ingeniería  
Informática  
del Software

- Una **declaración de tipo de documento** (*Document Type Definition, DTD*) define la estructura de los elementos y atributos permitidos en un documento XML
- Un DTD puede estar incrustado en el documento XML o en un archivo aparte
- DTD externo:

**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>
```



- En el ejemplo anterior se modelaron las pizzas utilizando **solamente atributos**
- En el ejemplo siguiente además se utilizarán **elementos** con versos como contenido
- La diferencia de usar elementos o atributos para incluir la información es que se forman árboles DOM diferentes

# Software y estándares para la Web

## Introducción a XML: Ejemplo *alba.xml* (a)

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE poema SYSTEM "poema.dtd">
<poema autor="Federico García Lorca" fecha="Abril 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>
<verso>La luz de la aurora lleva</verso>
<verso>Semilleros de nostalgias</verso>
<verso>Y la tristeza sin ojos</verso>
<verso>De la médula del alma.</verso>
<verso>La gran tumba de la noche</verso>
<verso>Su negro velo levanta</verso>
<verso>Para ocultar con el día</verso>
<verso>La inmensa cumbre estrellada.</verso>
</poema>
```

alba.xml

# Software y estándares para la Web

## Introducción a XML: Ejemplo *poema.dtd* (b)

poema.dtd

Grado en  
Ingeniería  
Informática  
del Software

```
<!ELEMENT poema (título, verso+)>
<!ELEMENT título (#PCDATA)>
<!ELEMENT verso (#PCDATA)>
<!ATTLIST poema
    autor CDATA #REQUIRED
    fecha CDATA #REQUIRED
    lugar CDATA #IMPLIED>
```

Opcional

# Software y estándares para la Web

## Introducción a XML: Árbol *alba.xml* (c)

Grado en  
Ingeniería  
Informática  
del Software



Creado con **xml2svg.exe**

# Software y estándares para la Web

## Introducción a XML: Ejemplo *libros.xml* (a)

libros.xml

Grado en  
Ingeniería  
Informática  
del Software

```
<?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>
```



# Software y estándares para la Web

## Introducción a XML: Ejemplo *libros.xml* (b)

libros.xml (continuación)

```
<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>
```

Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

## Introducción a XML: Ejemplo *libros.dtd* (c)

libros.dtd

Grado en  
Ingeniería  
Informática  
del Software

```
<!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

## Introducción a XML: Árbol DOM *libros.xml* (d)

Grado en  
Ingeniería  
Informática  
del Software



Creado con **xml2svg.exe**

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- **Lenguajes derivados de XML**
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

## Algunos lenguajes derivados de XML

### Grado en Ingeniería Informática del Software

- MathML, OpenMath
  - Visualización de ecuaciones matemáticas
- SVG, X3D, VRML, 3DXML
  - Gráficos vectoriales
- SMIL
  - Presentaciones multimedia
- P3P
  - Descripción de características de privacidad
- WML
  - Similar a HTML para teléfonos móviles (obsoleto)
- VoiceXML
  - Información de voz y audio
- TTML
  - Subtítulos de películas y series
  - Promovido por Netflix
- XML Signature
  - Firma de recursos y documentos
- ePub
  - Libros electrónicos
  - <https://www.w3.org/publishing/epub3/epub-spec.html>
- XAML
  - Interfaces y tecnología .NET
- XMI (Metadata Interchange)
  - Intercambio de metadatos de diseños de software en UML basado en XML
- XKMS
  - Firmas y criptografía
- XML Query
  - Consultas de documentos y bases de datos
- XBRL
  - Contabilidad y auditoría
- ebXML
  - Negocios electrónicos (e-business)
- SyncXML
  - Sincronización de dispositivos
- UPnP
  - Plug and Play universal
- KML, KMZ, GPX
  - Información geográfica (Google Earth, Google Maps, Garmin,...)
- Office Open XML
  - Documentos de Office
  - <http://www.ecma-international.org/publications/standards/Ecma-376.htm>

[List of XML markup languages - Wikipedia](#)

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- **Lenguajes derivados de XML**
  - KML
  - SVG
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

## Introducción a KML – Estándar del Open Geospatial Consortium (OGC)

Grado en  
Ingeniería  
Informática  
del Software

- KML es un formato de archivo que se puede utilizar para mostrar **datos geográficos** usando software, como Google Earth, Google Maps y Google Maps para móviles.
- KML utiliza una estructura basada en etiquetas con atributos y elementos anidados y está **basado en el estándar XML**
- **KMZ** es un archivo KML empaquetado en usando ZIP, el resultado es un archivo en binario
- <http://www.opengeospatial.org/standards/kml>
  - Especificación del estándar KML en *Open Geospatial Consortium (OGC)*



# Software y estándares para la Web

## Ejemplo KML: Oviedo.kml

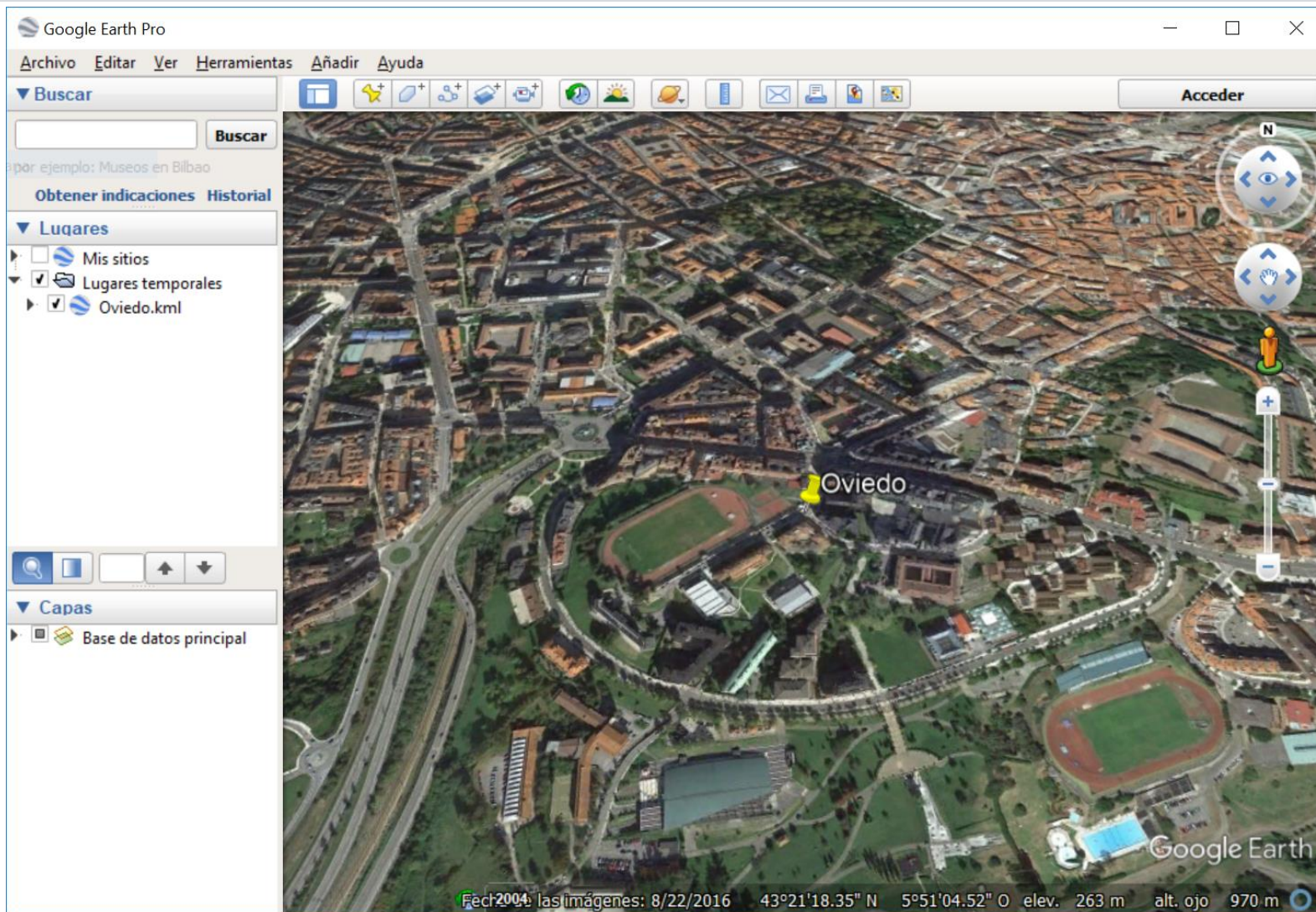
```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document id='Universidad de Oviedo'>
<Placemark>
<name>Oviedo</name>
<description>Escuela de Ingeniería Informática</description>
<Point>
<coordinates>
-5.851256792487727,43.355098167157855
</coordinates>
<altitudeMode>relativeToGround</altitudeMode>
</Point>
</Placemark>
</Document>
</kml>
```

Schema = Validación con una URI

# Software y estándares para la Web

## Oviedo.kml en Google Earth

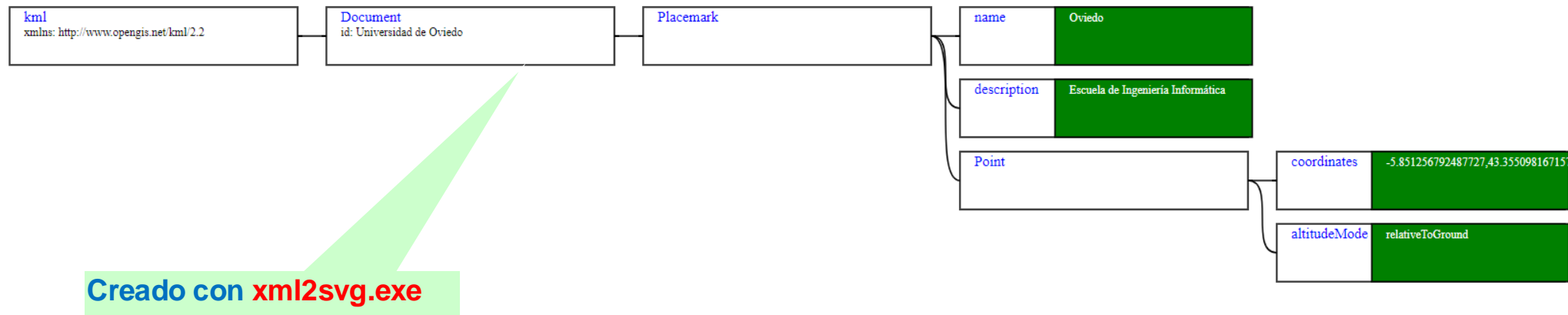
Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Árbol de Oviedo.kml

Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- **Lenguajes derivados de XML**
  - KML
  - **SVG**
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

- **Scalable Vector Graphics (SVG) 2.0**

- SVG 1.0 se convirtió en una recomendación de W3C en 04 de septiembre de 2001.
- SVG 1.1 se convirtió en una recomendación de W3C en 14 de enero de 2003.
- SVG 1.1 (segunda edición) se convirtió en una recomendación de W3C en 16 de agosto de 2011.
- SVG 2.0 W3C *Candidate Recommendation* (4 de octubre de 2018)
- <https://www.w3.org/TR/SVG2/>

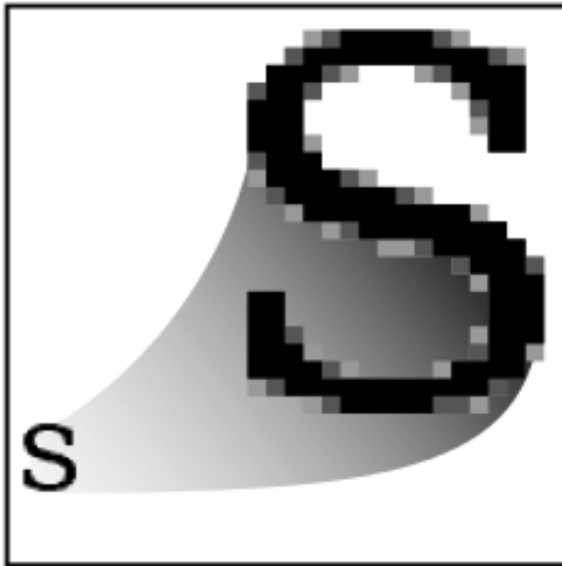


# Software y estándares para la Web

## SVG : Lenguaje derivado de XML

Grado en  
Ingeniería  
Informática  
del Software

- SVG es un lenguaje para describir **gráficos vectoriales bidimensionales**
- Utiliza la sintaxis XML



**Raster**  
.jpeg .gif .png



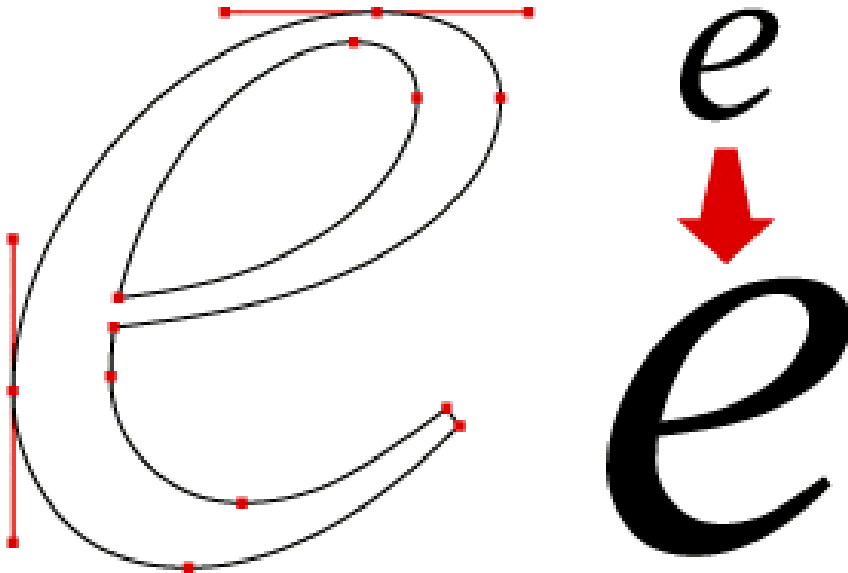
**Vector**  
.svg

# Software y estándares para la Web

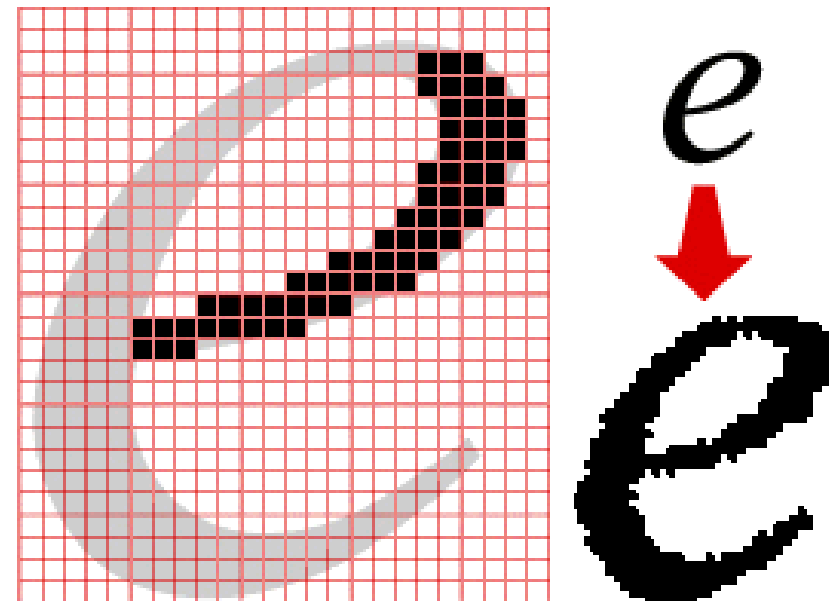
## Gráficos vectoriales versus gráficos raster

Grado en  
Ingeniería  
Informática  
del Software

VECTOR GRAPHICS



BITMAPMED (RASTER) GRAPHICS



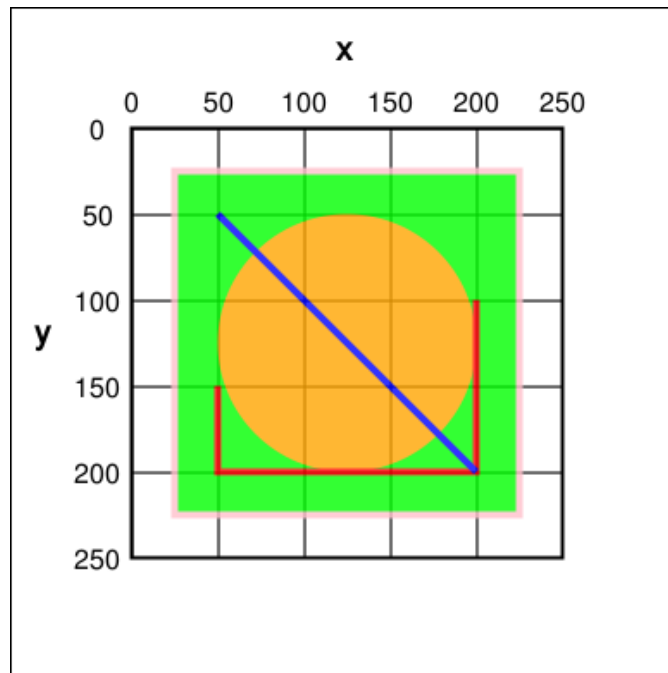


# Software y estándares para la Web

## Introducción a SVG (I)

Grado en  
Ingeniería  
Informática  
del Software

- SVG permite tres tipos de objetos gráficos:
  - formas gráficas
    - por ejemplo, rutas que consisten en líneas rectas y curvas
  - Imágenes
  - Texto en formato vectorial
- Sistema de coordenadas en SVG





# Software y estándares para la Web

## Introducción a SVG (II)

Grado en  
Ingeniería  
Informática  
del Software

- Los **objetos gráficos** pueden ser agrupados, cambiados de estilo, transformados y compuestos.
- El conjunto de características incluye
  - Transformaciones anidadas
  - Trazados
  - Máscaras
  - Efectos de filtro
  - Objetos con plantilla de recorte.
- Los gráficos de SVG pueden ser **interactivos y dinámicos**, incluyendo animaciones
- La especificación de un gráfico sVG se puede
  - **Incrustar en HTML**
  - Ser un **archivo independiente** con la extensión
    - **.svg** (archivo de texto)
    - **.sgvz** (archivo binario, obtenido comprimiendo en formato ZIP)

# Software y estándares para la Web

## Ventajas de SVG

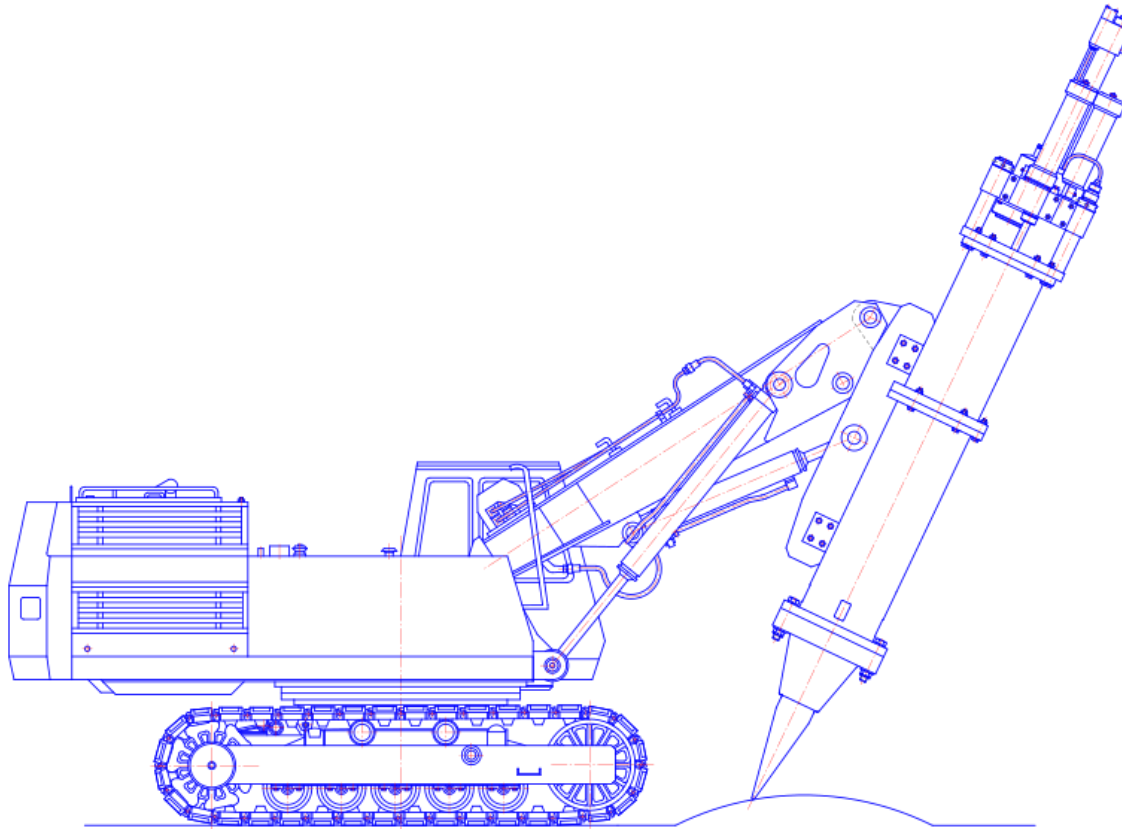
- Las imágenes SVG pueden ser creadas y editadas con cualquier **editor de texto**
- Las imágenes SVG pueden ser **buscadas e indexadas**
- Las imágenes SVG son **escalables**
- Las imágenes SVG pueden ser **impresas con alta calidad en cualquier resolución**
- Las imágenes SVG **se pueden ampliar**
- Las imágenes SVG **no pierden calidad al cambiar de tamaño** debido a que son vectoriales
- SVG es un **estándar abierto**
- Los archivos SVG son **XML**
- Existen **herramientas** que generan gráficos en formato SVG

Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

Imagen generada en SVG

Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Herramientas que generan SVG

Grado en  
Ingeniería  
Informática  
del Software

- Inkscape
  - es un editor de gráficos vectoriales libre y de código abierto
  - <https://gitlab.com/inkscape/inkscape>
- LibreOffice Draw
- Adobe Illustrator
- CorelDRAW
- Gnuplot
- Scribus
- Etc.

# Software y estándares para la Web

## Ejemplo SVG incrustado en HTML - **circulo.html**

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Círculo SVG</title>
</head>
<body>

<h1>Círculo con SVG</h1>

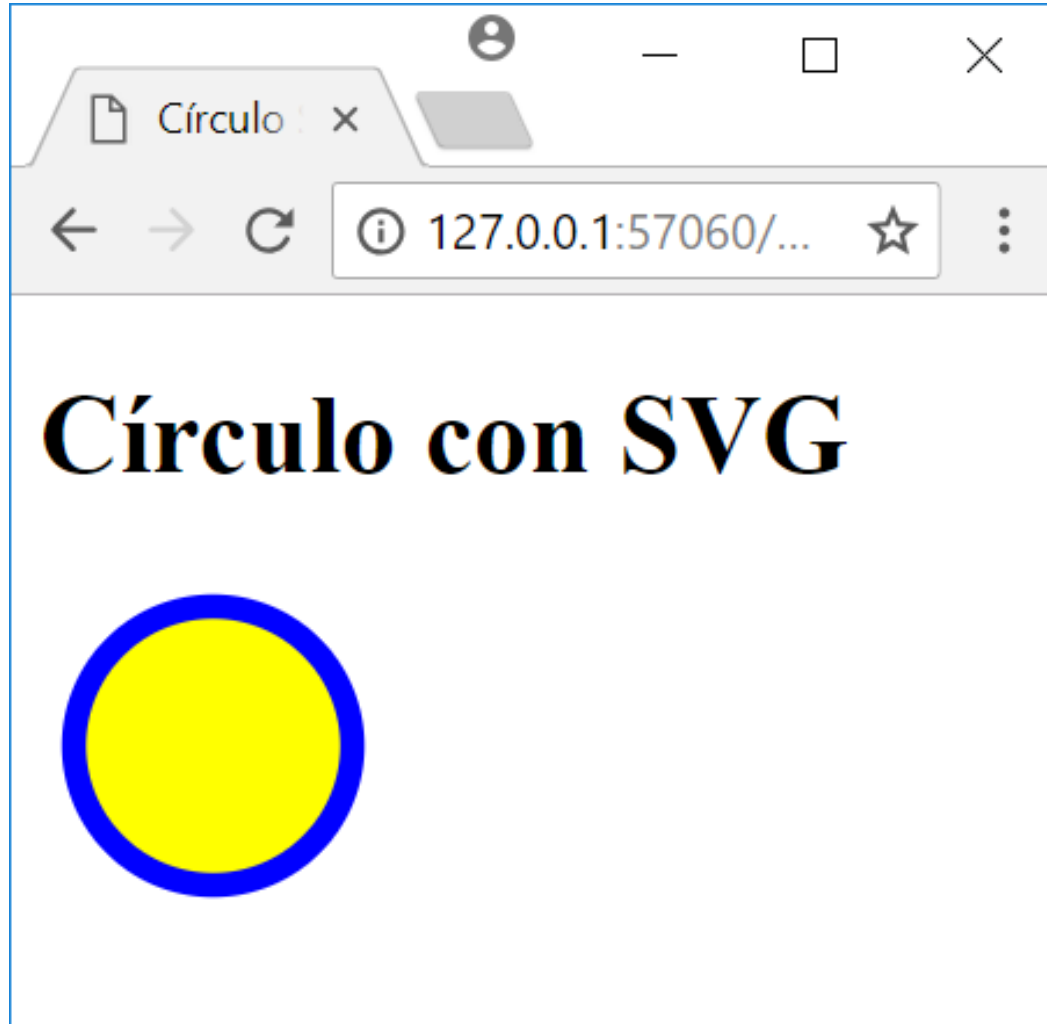
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="blue" stroke-width="7" fill="yellow" />
  Su agente de usuario no soporta SVG
</svg>

</body>
</html>
```

# Software y estándares para la Web

## Visualización - SVG - **circulo.html**

Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Ejemplo SVG incrustado en HTML - polilinea.html

Grado en  
Ingeniería  
Informática  
del Software

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Polilínea SVG</title>
</head>
<body> <h1>Polilínea cerrada con SVG</h1>
<svg height="180" width="500">
  <polyline points=
    "0,160
     20,40
     40,120
     60,80
     80,0
     100,120
     120,130
     140,40
     160,140
     180,80
     200,120
     220,130
     240,90
     260,20
     280,50
     300,100
     320,75
     340,125
     360,160
     0,160"
    style="fill:white;stroke:red;stroke-width:4" />
  Su agente de usuario no soporta SVG
</svg>
</body></html>
```

# Software y estándares para la Web

## Visualización - SVG - **polilinea.html**

Grado en  
Ingeniería  
Informática  
del Software

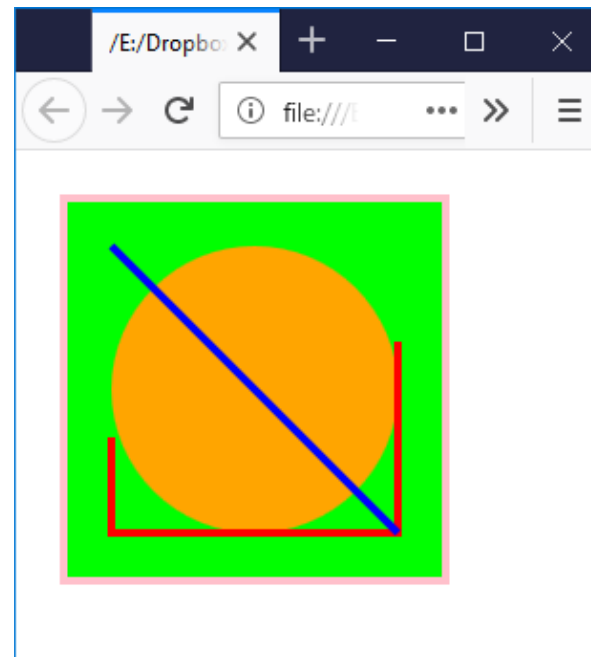
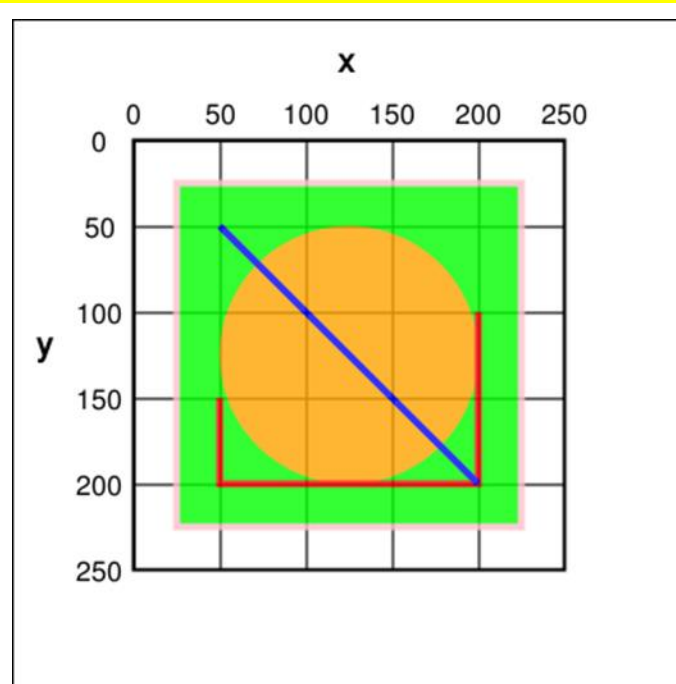




# Software y estándares para la Web

## Ejemplo SVG – Rectangulo-circulo.svg

```
<?xml version="1.0" encoding="UTF-8" ?>
<svg xmlns="http://www.w3.org/2000/svg" version="2.0">
  <rect x="25" y="25" width="200" height="200" fill="lime" stroke-width="4" stroke="pink" />
  <circle cx="125" cy="125" r="75" fill="orange" />
  <polyline points="50,150 50,200 200,200 200,100" stroke="red" stroke-width="4" fill="none" />
  <line x1="50" y1="50" x2="200" y2="200" stroke="blue" stroke-width="4" />
</svg>
```



# Software y estándares para la Web

## Ejemplo SVG – polilinea-texto.svg (a)

```
<?xml version="1.0" encoding="UTF-8" ?>
<svg xmlns="http://www.w3.org/2000/svg" version="2.0">

<polyline points=
    "10,160
     30,40
     40,120
     60,80
     80,10
     100,120
     120,130
     140,40
     160,140
     180,80
     200,120
     220,130
     240,90
     260,20
     280,50
     300,100
     320,75
     340,125
     360,160
     10,160"
    style="fill:white;stroke:red;stroke-width:4" />
```

Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

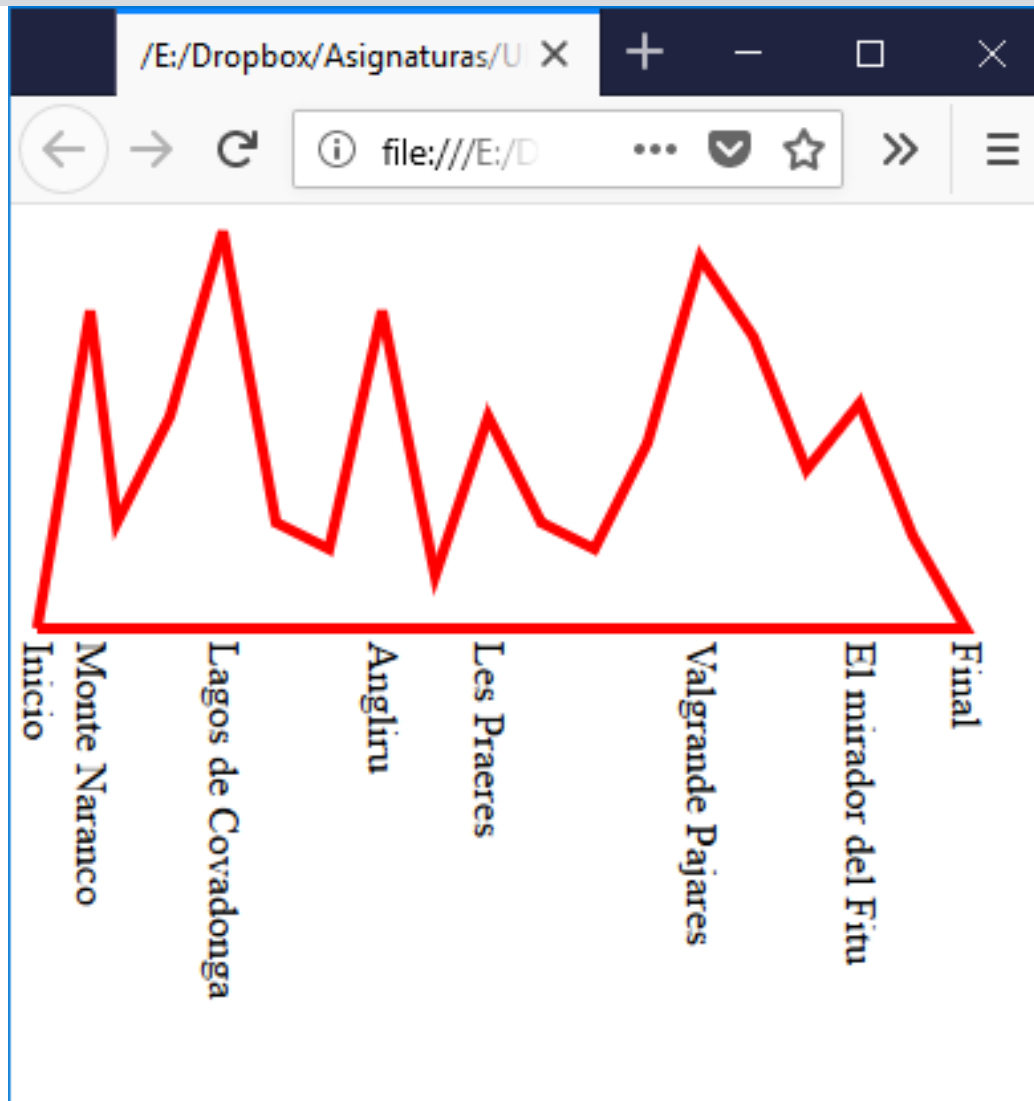
## Ejemplo SVG – polilinea-texto.svg (b)

Grado en  
Ingeniería  
Informática  
del Software

```
<text x="10" y="165" style="writing-mode: tb; glyph-orientation-vertical: 0;">
    Inicio
</text>
<text x="30" y="165" style="writing-mode: tb; glyph-orientation-vertical: 0;">
    Monte Naranco
</text>
<text x="80" y="165" style="writing-mode: tb; glyph-orientation-vertical: 0;">
    Lagos de Covadonga
</text>
<text x="140" y="165" style="writing-mode: tb; glyph-orientation-vertical: 0;">
    Angliru
</text>
<text x="180" y="165" style="writing-mode: tb; glyph-orientation-vertical: 0;">
    Les Praeres
</text>
<text x="260" y="165" style="writing-mode: tb; glyph-orientation-vertical: 0;">
    Valgrande Pajares
</text>
<text x="320" y="165" style="writing-mode: tb; glyph-orientation-vertical: 0;">
    El mirador del Fitu
</text>
<text x="360" y="165" style="writing-mode: tb; glyph-orientation-vertical: 0;">
    Final
</text>
</svg>
```

# Software y estándares para la Web

## Visualización SVG – polilinea-texto.svg



Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- **Documentos XML**
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

## Documentos XML (I): Definición y editores

Grado en  
Ingeniería  
Informática  
del Software

- *“Información jerarquizada, en forma de texto, que constituye un objeto de datos que puede ser presentado mediante una estructura de árbol, que puede estar almacenado en un único archivo o dividido en varios”*
- Editores para crear archivos XML:
  - Editores de texto: **Visual Studio Code (recomendado)**, Notepad++, Brackets, VisualStudio, Eclipse, Sublime,...
  - Editor especializado: **XMLSpy**

Documento XML = datos + marcado

# Software y estándares para la Web

## Documentos XML (II): Partes de un documento XML

Grado en  
Ingeniería  
Informática  
del Software

- **Prólogo**
  - Declaración XML
  - Declaración del Tipo de Documento (DTD) o Scheme
- **Cuerpo**
  - Elementos
  - Atributos
  - Entidades predefinidas
  - Instrucciones de proceso
  - Secciones CDATA
- **Comentarios:**
  - Se pueden introducir en cualquier lugar del cuerpo o del prólogo, pero nunca dentro de las declaraciones, etiquetas u otros comentarios.
  - Dentro de los comentarios no pueden aparecer los caracteres --  
`<!-- Texto de comentario-->`

```
<?xml version= "1.0" encoding = "UTF-8" standalone="yes" ?>
```

- **version (obligatoria)**: versión de XML usada en el documento
  - 1.0 = versión más habitual
  - 1.1 = aumenta capacidad de soporte de Unicode
- **encoding (opcional)**: la forma en que se ha codificado el documento.
  - Por defecto es UTF-8, aunque podrían ponerse otras, como UTF-16, US-ASCII, ISO-8859-1, etc.
  - **UTF-8** = caracteres Unicode
  - iso-8859-1 = caracteres latinos
- **standalone (opcional)**
  - Indica al procesador XML si un documento es independiente, es decir no usa fuentes externas (standalone="yes")
  - Indica que se basa en información de fuentes externas, es decir, si depende de declaraciones de marca externas como una DTD externa (standalone="no").



# Software y estándares para la Web

## Documentos XML (IV): Prólogo. Declaración del Tipo de Documento

Grado en  
Ingeniería  
Informática  
del Software

```
<!DOCTYPE nombre SYSTEM|PUBLIC uri>
```

- Declara una DTD que se quiera incorporar al documento
- **DOCTYPE**: indica que la etiqueta contiene una DTD
- **nombre**: declara el nombre de la DTD
- **PUBLIC**: indica que la DTD es pública y está disponible
- **SYSTEM**: La DTD no es pública
- **URI** (*Identificadores únicos de recursos*)
  - URI = URL + URN
    - URL (*Locator*) tiene doble funcionalidad:
      - Identificar recursos
      - Protocolo de acceso
      - Ejemplo: `http://www.uniovi.es`
    - URN: Nombre único de recursos
      - Ejemplo: `urn:isbn:0-395-36341-1`

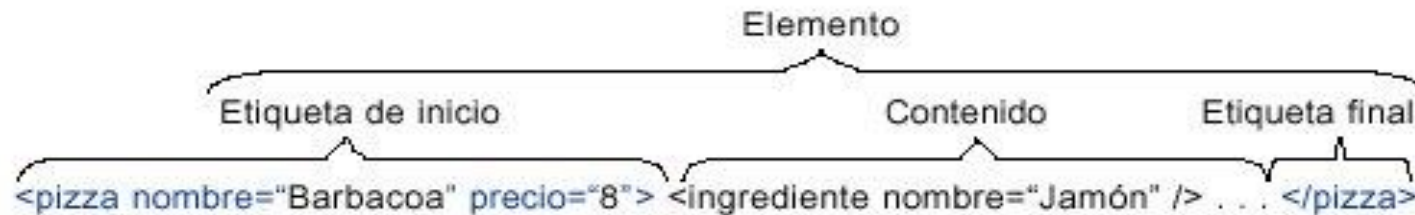
```
<!DOCTYPE pizzas SYSTEM "pizzas.dtd">
```

# Software y estándares para la Web

## Documentos XML (V): Cuerpo. Elementos (a)

Grado en  
Ingeniería  
Informática  
del Software

- El **cuerpo** del documento está formado por un **elemento**
- Un elemento está formado por:
  - Una **etiqueta inicial** (nombre entre signos < y > ): <etiqueta>
  - La etiqueta inicial puede contener una lista **de atributos y valores**:  
<etiqueta atributo="valor">
  - Los contenidos del elemento (puede estar vacío)
  - El elemento debe acabar con **una etiqueta final** con el mismo nombre
  - El **contenido** del elemento es todo lo que hay entre la etiqueta inicial y la final
    - El contenido pueden ser otros elementos



- En caso de un elemento vacío puede usarse la sintaxis: <etiqueta />

**<ingrediente nombre="Jamón" calorías="8"></ingrediente>**

|| (es equivalente a)

**<ingrediente nombre="Jamón" calorías="8" />**

- Es necesario cerrar todas las etiquetas
- **XML es sensible a mayúsculas/minúsculas**

# Software y estándares para la Web

## Documentos XML (VI): Cuerpo. Elementos (b)

Grado en  
Ingeniería  
Informática  
del Software

- Sintaxis de los elementos:  
`<NombreElemento>Contenido</NombreElemento>`
- Representación de elementos (llamados “*singletons*”) vacíos:  
`<NombreElemento/>`
- Anidamiento de elementos jerárquico

```
<Estudiante>  
  <Nombre>Juan Manuel</Nombre>  
  <Edad>18</Edad>  
  <Créditos>60</Créditos>  
</Estudiante>
```

### Convención para nombres de elementos

- Consisten de uno o más caracteres sin espacios en el medio.
  - Solo puede comenzar con una letra o con un “\_”
  - Los siguientes caracteres pueden ser cualquiera del estándar “Unicode”.
- Sensibles a mayúsculas/minúsculas
- **Ejemplos:**
  - *Válidos*
    - `“Estudiante”`
    - `“_Estudiante”`
    - `“Estudiante1234”`
    - `“Estudianteß”`
  - *No válidos*
    - `“-Estudiante”`
    - `“1234”`
    - `“Estudiante 1234”`

# Software y estándares para la Web

## Documentos XML (VIII): Cuerpo. Anidamiento

Grado en  
Ingeniería  
Informática  
del Software

- Las etiquetas que se abran deben cerrarse sin que se produzcan anidamientos

- **Correcto**

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

- **Incorrecto**

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

# Software y estándares para la Web

## Documentos XML (IX): Cuerpo. Atributos

Grado en  
Ingeniería  
Informática  
del Software

- Los valores de los atributos se escriben entre comillas dobles o simples.
- El orden de atributos no es significativo
- No puede haber nombres de atributos repetidos
- Pueden incluirse valores entrecomillados siempre que las comillas sean diferentes de las externas.
  - Ejemplo: frase "Diego dijo 'digo' "

```
<Asignaturas unidad="créditos" >60</Asignaturas>
```

# Software y estándares para la Web

## Documentos XML (X): Cuerpo. Atributos predefinidos

Grado en  
Ingeniería  
Informática  
del Software

- Hay varios atributos predefinidos.
- Por ejemplo:
  - **xml:lang** especifica el código del idioma: **en** (inglés), **sp** (español), etc.
  - **xml:space** especifica cómo tratar el espacio en blanco:
    - **preserve** = mantenerlo
    - **default** = dejar libertar a la aplicación para tratarlo como quiera
  - **xml:base** define la URL que identifica las direcciones relativas

# Software y estándares para la Web

## Documentos XML (XI): Ejemplo de atributos predefinidos

Grado en  
Ingeniería  
Informática  
del Software

```
<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>
<p xml:lang="sp">¿Qué color es?</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>
```



# Software y estándares para la Web

## Documentos XML (XII): Cuerpo. Entidades predefinidas

Grado en  
Ingeniería  
Informática  
del Software

Entidad	Carácter
&amp;	&
&lt;	<
&gt;	>
&apos;	'
&quot;	"

- Cualquier carácter Unicode puede indicarse mediante & seguido del número y acabado por ;

- Mecanismo que permite a los documentos XML contener instrucciones específicas para las aplicaciones que los van a usar, sin que éstas formen parte de los datos del propio documento (**actualmente en desuso por problemas de seguridad**)
- Ejemplos:

- `<?xml version='1.0' ?>`
- `<?xml-stylesheet type= "text/xsl" href= "MiHojaDeEstilo.xsl"?>`
- `<?php ... ?>`

- Construcción en XML que permite **especificar datos, utilizando cualquier carácter, especial o no, sin que se interprete como marcado XML**
- La razón de esta construcción es que a veces es necesario para los autores de documentos XML, **poder leerlo fácilmente** sin tener que descifrar los códigos de entidades.
- Por ejemplo, con **código fuente** de lenguajes de programación.

### Ejemplo **sin** CDATA

```
<codigo>
if ( x &lt; 3 &amp;&amp; x &gt; 4)
    printf (&quot; Hola&quot; ) ;
</codigo>
```

### Ejemplo **con** CDATA

```
<codigo>
<![CDATA[
if ( x < 3 && x > 4)
    printf ( " Hola " ) ;
] ]>
</codigo>
```

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- Documentos XML
- **Documentos XML bien formados**
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

## Documentos XML bien formados (I)

- **Documento bien formado**

- Sigue las reglas sintácticas de XML
- No se comprueba su validez con un DTD o Schema
- Importante:
  - Contiene un único elemento raíz que envuelve todo
  - 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>
```

- El documento puede contener varias **instrucciones de procesamiento**
  - Indican cómo debe procesarse el documento

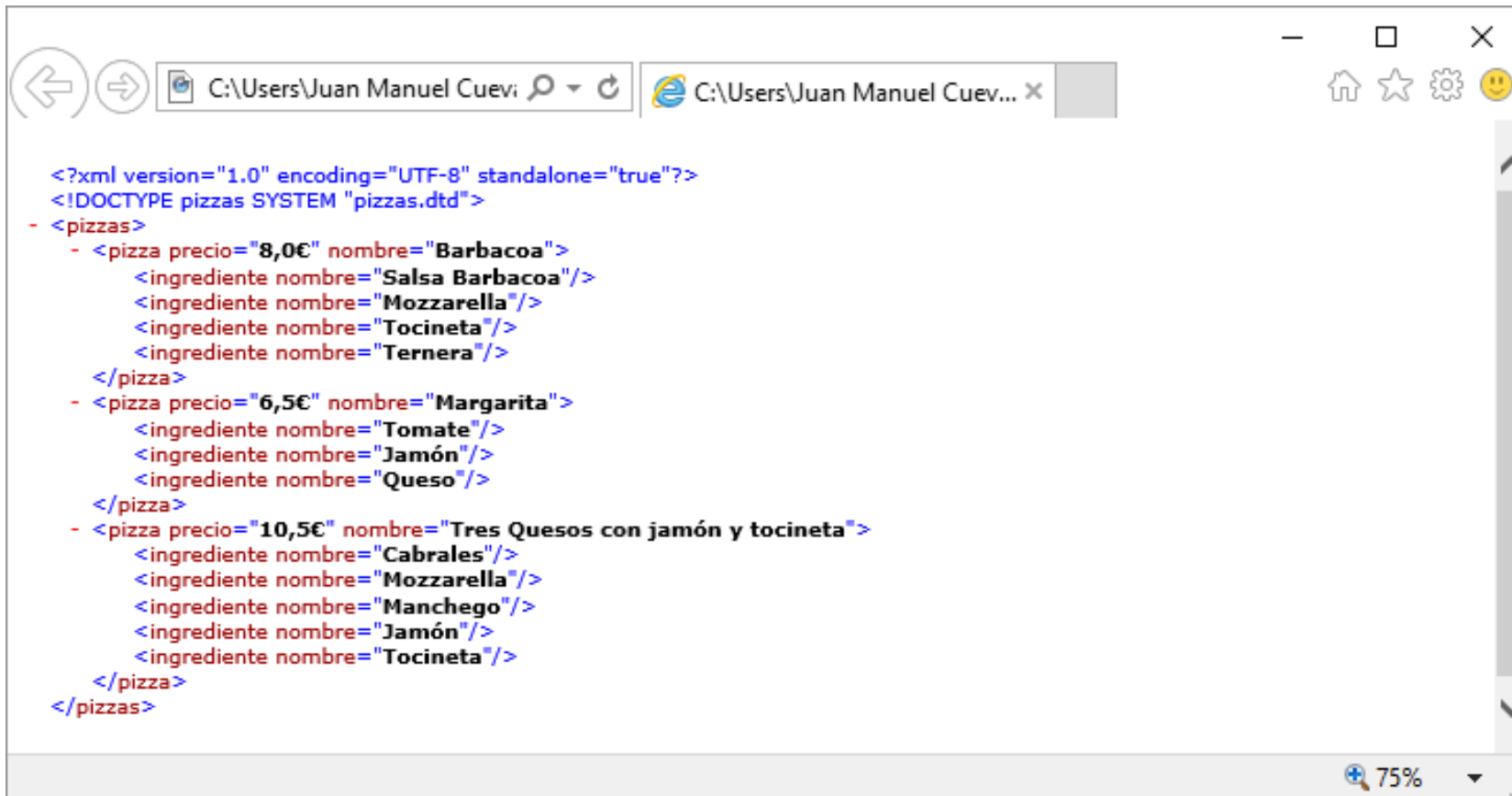
**<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>**

# Software y estándares para la Web

## Documentos XML bien formados (II)

- Los navegadores comprueban si un documento XML está bien formado. **Bien formado.**

Grado en  
Ingeniería  
Informática  
del Software



```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<!DOCTYPE pizzas SYSTEM "pizzas.dtd">
- <pizzas>
  - <pizza precio="8,0€" nombre="Barbacoa">
    <ingrediente nombre="Salsa Barbacoa"/>
    <ingrediente nombre="Mozzarella"/>
    <ingrediente nombre="Tocineta"/>
    <ingrediente nombre="Ternera"/>
  </pizza>
  - <pizza precio="6,5€" nombre="Margarita">
    <ingrediente nombre="Tomate"/>
    <ingrediente nombre="Jamón"/>
    <ingrediente nombre="Queso"/>
  </pizza>
  - <pizza precio="10,5€" nombre="Tres Quesos con jamón y tocineta">
    <ingrediente nombre="Cabrales"/>
    <ingrediente nombre="Mozzarella"/>
    <ingrediente nombre="Manchego"/>
    <ingrediente nombre="Jamón"/>
    <ingrediente nombre="Tocineta"/>
  </pizza>
</pizzas>
```

- Ejemplo
  - **Documento XML mal formado.**
  - No se ha cerrado la etiqueta <pizza>

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE pizzas SYSTEM "pizzas.dtd">
<pizzas>
<pizza precio="8" nombre="Barbacoa">
<ingrediente nombre="Salsa Barbacoa"/>
<ingrediente nombre="Mozzarella"/>
<ingrediente nombre="Tocineta"/>
<ingrediente nombre="Ternera"/>
</pizzas>
```



# Software y estándares para la Web

## Documentos XML mal formados (II)

Grado en  
Ingeniería  
Informática  
del Software

- Los navegadores detectan que un documento XML está **mal formado**.



- Ejemplo
  - **Documento XML mal formado.**
  - No hay un único elemento raíz que envuelve todo <pizzas> y </pizzas>

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

<pizza precio="8" nombre="Barbacoa">
  <ingrediente nombre="Salsa Barbacoa"/>
  <ingrediente nombre="Mozzarella"/>
  <ingrediente nombre="Tocineta"/>
  <ingrediente nombre="Ternera"/>
</pizza>

<pizza precio="6,5" nombre="Margarita">
  <ingrediente nombre="Tomate"/>
  <ingrediente nombre="Jamón"/>
  <ingrediente nombre="Queso"/>
</pizza>
```

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- Documentos XML
- Documentos XML bien formados
- **Documentos XML válidos**
- Ventajas de XML
- Inconvenientes de XML
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

## Documentos XML válidos (I)

Grado en  
Ingeniería  
Informática  
del Software

- Se puede incluir una declaración del tipo de documento (DTD)

pizzas.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE pizzas SYSTEM "pizzas.dtd">
<pizzas>
  <pizza nombre="Margarita" precio="6">
    <ingrediente nombre="Tomate" />
  </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>
```

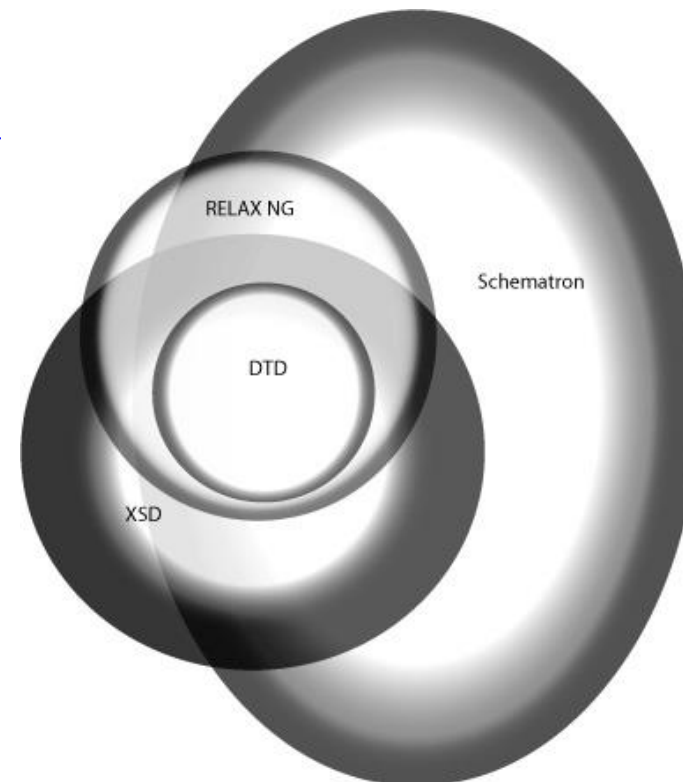
- Documento XML válido**
  - Está bien formado y
  - La estructura encaja con la declaración del tipo de documento (DTD)

# Software y estándares para la Web

## Documentos XML válidos (II)

Grado en  
Ingeniería  
Informática  
del Software

- Es posible validar la estructura de los documentos utilizando diversas alternativas:
  - DTDs
  - XML Schema
    - <http://www.w3.org/XML/Schema>
  - Relax NG
    - <http://relaxng.org/>
  - Schematron
    - <http://www.schematron.com/>



# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- **Ventajas de XML**
- Inconvenientes de XML
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

## Ventajas de XML

**Grado en  
Ingeniería  
Informática  
del Software**

- Es un formato estructurado (definido por una gramática)
- Contiene información y meta-información
  - Ha sido diseñado específicamente para Internet
- Soportado por visualizadores y servidores
  - Numerosas herramientas de procesamiento
  - Legible por personas humanas (archivo de texto)
  - 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 o un esquema XML común, facilita intercambio de información

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- **Inconvenientes de XML**
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos



# Software y estándares para la Web

## Inconvenientes de XML(I)

- 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;  
}
```

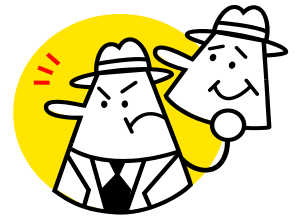
```
<function name="main" type="int">  
  <arg type="void" />  
  <call function="printf">  
    <param>Hola</param>  
  </call>  
  <return value="0"/>  
</function>
```

- Es posible crear formatos y vocabularios propietarios
- Puede fomentar la proliferación de vocabularios específicos
- Bueno para texto, malo para datos binarios

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

- Poco eficiente como lenguaje de almacenamiento de bases de datos

Grado en  
Ingeniería  
Informática  
del Software

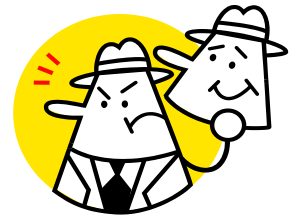
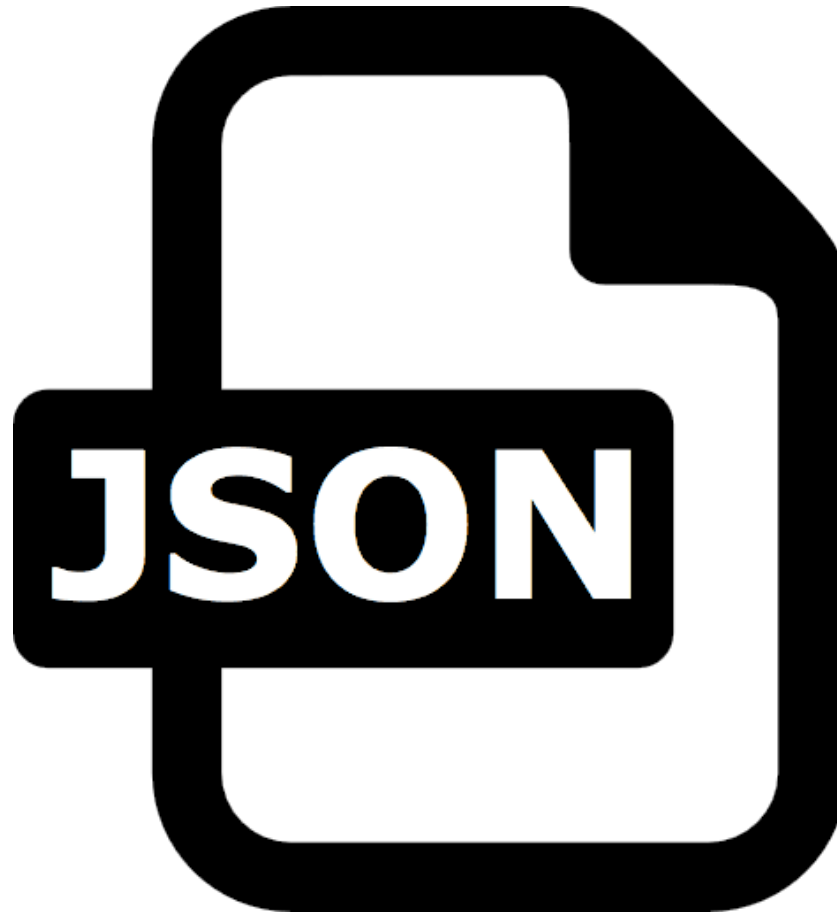


# Software y estándares para la Web

## Inconvenientes de XML(II)

- Está siendo sustituido en muchas aplicaciones por JSON (JavaScript Object Notation)

Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Esquema

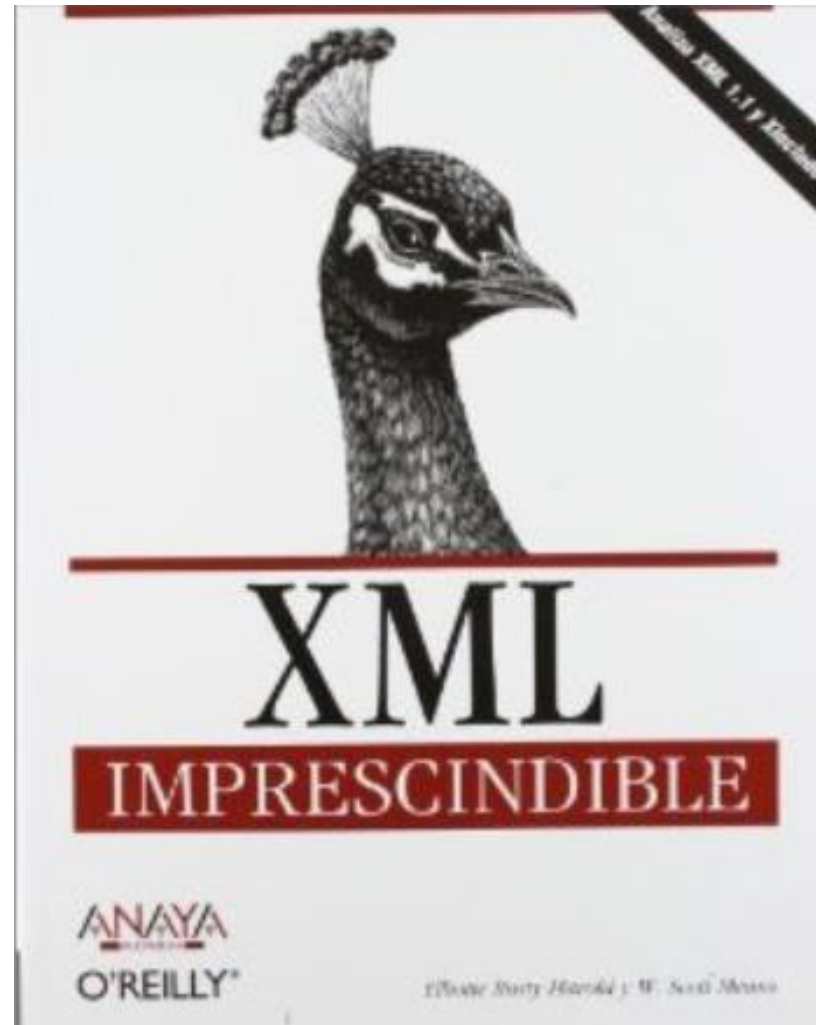
Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- **Bibliografía**
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

## Bibliografía (I)

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



Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

## Bibliografía (II)

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



Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- Bibliografía
- **Referencias Web**
- Ejercicios resueltos
- Ejercicios propuestos

# Software y estándares para la Web

## Referencias Web (I)

Grado en  
Ingeniería  
Informática  
del Software

- <http://www.w3.org/TR/xml11/>
  - Especificación del estándar XML en W3C
  - Última versión 29-Septiembre-2006
- <https://www.w3.org/TR/SVG2/>
  - Especificación del estándar SVG
- <http://www.w3schools.com/xml>
  - Tutoriales on-line de XML
- [https://www.w3schools.com/graphics/svg\\_intro.asp](https://www.w3schools.com/graphics/svg_intro.asp)
  - Tutoriales on-line SVG
- <http://www.opengeospatial.org/standards/kml>
  - Especificación del estándar KML
- <https://developers.google.com/kml/documentation/>
  - Documentación de KML
- <http://json.org/>
  - Especificación del estándar JSON

# Software y estándares para la Web

## Referencias Web (II)

- [https://en.wikipedia.org/wiki/Scalable\\_Vector\\_Graphics](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics)

Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- 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** para contener artículos de revista 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) – Solución

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<articles>

<article year="2017" start-page="301" end-page="313">
<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>
<authors>
<author email="cristian@email.es">Cristian González García</author>
<author email="daniel@email.es">Daniel Meana-Llorián</author>
<author email="cristina@email.es">B. Cristina Pelayo García-Bustelo</author>
<author email="juanmanuel@email.es">Juan Manuel Cueva Lovelle</author>
<author email="nestor@email.es">Néstor García-Fernández</author>
</authors>
<summary>
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.".
</summary>
<keywords>
<keyword>IoT</keyword>
<keyword>security</keyword>
<keyword>Computer Vision</keyword>
</keywords>
<journal>Future Generation Comp. Syst.</journal>
<volume>76</volume>
</article>
```

# Software y estándares para la Web

## Ejercicios resueltos (I) – Solución (continuación)

Grado en  
Ingeniería  
Informática  
del Software

```
<article year="2016" start-page="299" end-page="337">
<title>Combining the Continuous Integration Practice and the Model-Driven Engineering Approach.</title>
<authors>
<author email="vicente@email.es">Vicente García-Díaz</author>
<author email="jordan@email.es">Jordán Pascual Espada</author>
<author email="edward@email.es">Edward Rolando Núñez-Valdéz</author>
<author email="cristina@email.es">B. Cristina Pelayo García-Bustelo</author>
<author email="juanmanuel@email.es">Juan Manuel Cueva Lovelle</author>
</authors>
<summary>
Resumen del artículo "Combining the Continuous Integration Practice and the Model-Driven Engineering
Approach.".
</summary>
<keywords>
<keyword>Integration</keyword>
<keyword>Practice</keyword>
<keyword>Model-Driven</keyword>
</keywords>
<journal>Computing and Informatics</journal>
<volume>35</volume>
</article>

. . .

</articles>
```

# Software y estándares para la Web

## Ejercicios resueltos (I) – Comprobación de bien formado

Grado en  
Ingeniería  
Informática  
del Software

```
file:///E:/Dropbox/Asignaturas/MIW/Len...
Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

<articles>
  <article year="2017" start-page="301" end-page="313">
    <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>
    <authors>
      <author email="cristian@email.es">Cristian González García</author>
      <author email="daniel@email.es">Daniel Meana-Llorián</author>
      <author email="cristina@email.es">B. Cristina Pelayo García-Bustelo</author>
      <author email="juanmanuel@email.es">Juan Manuel Cueva Lovelle</author>
      <author email="nestor@email.es">Néstor García-Fernández</author>
    </authors>
    <summary>
      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.".
    </summary>
    <keywords>
      <keyword>IoT</keyword>
      <keyword>Security</keyword>
      <keyword>Computer Vision</keyword>
    </keywords>
    <journal>Future Generation Comp. Syst.</journal>
    <volume>76</volume>
  </article>
  <article year="2016" start-page="299" end-page="337">
    <title>
      Combining the Continuous Integration Practice and the Model-Driven Engineering Approach.
    </title>
    <authors>
      <author email="vicente@email.es">Vicente García-Díaz</author>
      <author email="jordan@email.es">Jordán Pascual Espada</author>
      <author email="edward@email.es">Edward Rolando Núñez-Valdéz</author>
      <author email="cristina@email.es">B. Cristina Pelayo García-Bustelo</author>
      <author email="juanmanuel@email.es">Juan Manuel Cueva Lovelle</author>
    </authors>
    <summary>
      Resumen del artículo "Combining the Continuous Integration Practice and the Model-Driven Engineering Approach.".
    </summary>
    <keywords>
      <keyword>Integration</keyword>
      <keyword>Practice</keyword>
      <keyword>Model-Driven</keyword>
    </keywords>
    <journal>Computing and Informatics</journal>
    <volume>35</volume>
  </article>
```

# Software y estándares para la Web

## Ejercicios resueltos (II)

Grado en  
Ingeniería  
Informática  
del Software

- Construir un **documento XML bien formado** 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 “Fabas 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<sup>a</sup> Luisa”, “Libro de las 1001 recetas”, “www.recetasMUYricas.com”)

# Software y estándares para la Web

## Ejercicios resueltos (II) - Solución

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cooking-recipes>
<cooking-recipe type="Primer plato" vegan="No">
<name>Arroz integral con pollo</name>
<ingredients>
<ingredient quantity="300" measure="g" calories="1032">Arroz Integral</ingredient>
<ingredient quantity="300" measure="g" calories="80">Pollo</ingredient>
<ingredient quantity="1" measure="unidad">Cebolleta</ingredient>
<ingredient quantity="2" measure="unidades">Tallos de apio</ingredient>
<ingredient quantity="3" measure="unidades">Dientes de ajo</ingredient>
<ingredient quantity="2" measure="cucharadas">Tomate frito</ingredient>
<ingredient quantity="1/2" measure="cucharadita">Curry en pavo</ingredient>
<ingredient quantity="1/2" measure="cucharadita">Pimentón dulce</ingredient>
<ingredient quantity="1/4" measure="unidad">Pimiento verde</ingredient>
<ingredient quantity="1" measure="unidad">Huevo</ingredient>
<ingredient quantity="3" measure="cucharadas">Salsa de soja</ingredient>
<ingredient quantity="700" measure="ml" calories="20">Caldo de pollo</ingredient>
</ingredients>
<elaboration-process difficulty="Fácil" time-minutes="45">
<steps>
<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 sofreímos primero los ajos durante un minuto y medio, luego los retiramos de la sartén y reservamos.</step>
<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. Sofreír durante 5 minutos. Apagar el fuego y añadir el arroz integral junto al curry y el pimentón. Dejar que el arroz repose en la sartén caliente dos o tres minutos.</step>
```

# Software y estándares para la Web

## Ejercicios resueltos (II) – Solución (continuación)

Grado en  
Ingeniería  
Informática  
del Software

```
<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 cocción hasta los 20 minutos. Luego apagar el fuego.</step>
<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 gusto la cortamos en trocitos.</step>
<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>
</steps>
<elements>
<element>Cuchillo</element>
<element>Horno</element>
<element>Sartén</element>
</elements>
</elaboration-process>
<origin>Recetas de mi abuela</origin>
</cooking-recipe>

. . .

</cooking-recipes>
```



# Software y estándares para la Web

## Ejercicios resueltos (II) – Comprobación de bien formado

Grado en  
Ingeniería  
Informática  
del Software

```
/E:/Dropbox/Asignaturas/MIW/Len X +
file:///E:/Dropbox/Asignaturas/MIW/Lenguajes-y-estandares-Web/XML/2018/01
Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

-<cooking-recipes>
-<cooking-recipe type="Primer plato" vegan="No">
  <name>Arroz integral con pollo</name>
  <ingredients>
    <ingredient quantity="300" measure="g" calories="1032">Arroz Integral</ingredient>
    <ingredient quantity="300" measure="g" calories="80">Pollo</ingredient>
    <ingredient quantity="1" measure="unidad">Cebolleta</ingredient>
    <ingredient quantity="2" measure="unidades">Tallos de apio</ingredient>
    <ingredient quantity="3" measure="unidades">Dientes de ajo</ingredient>
    <ingredient quantity="2" measure="cucharadas">Tomate frito</ingredient>
    <ingredient quantity="1/2" measure="cucharadita">Curry en pavo</ingredient>
    <ingredient quantity="1/2" measure="cucharadita">Pimentón dulce</ingredient>
    <ingredient quantity="1/4" measure="unidad">Pimiento verde</ingredient>
    <ingredient quantity="1" measure="unidad">Huevo</ingredient>
    <ingredient quantity="3" measure="cucharadas">Salsa de soja</ingredient>
    <ingredient quantity="700" measure="ml" calories="20">Caldo de pollo</ingredient>
  </ingredients>
  <elaboration-process difficulty="Fácil" time-minutes="45">
    <steps>
      <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 primero los ajos durante un minuto y medio, luego los retiramos de la sartén y reservamos.
      </step>
      <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 5 minutos. Apagar el fuego y añadir el arroz integral junto al curry y el pimentón. Dejar que el arroz repose en la sartén caliente dos o tres minutos.
      </step>
      <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 cocción hasta los 20 minutos. Luego apagar el fuego.
      </step>
      <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 gusto la cortamos en trocitos.
      </step>
      <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>
    </steps>
  </elaboration-process>
  <elements>
    <element>Cuchillo</element>
    <element>Horno</element>
    <element>Sartén</element>
  </elements>
  <elaboration-process>
    <origin>Recetas de mi abuela</origin>
  </elaboration-process>
</cooking-recipe>
```

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Lenguajes de marcas
- Introducción a XML
- Lenguajes derivados de XML
- Documentos XML
- Documentos XML bien formados
- Documentos XML válidos
- Ventajas de XML
- Inconvenientes de XML
- 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** 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** 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://prerromanicoasturiano.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** Introducción

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