



# **XML: eXtensible Markup Languaje**

## **Tecnologías XML**

Transformación de documentos XML (XSLT)

**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**

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios propuestos

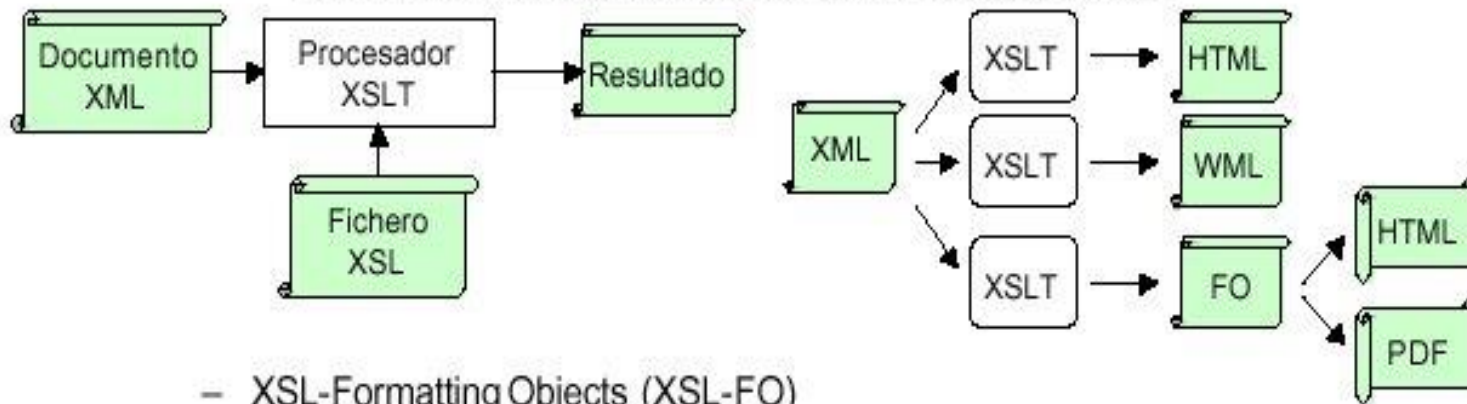
- **Introducción a XSLT**
- Transformaciones XSLT
- Estructura de los documentos XSL
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios propuestos

# Software y estándares para la Web

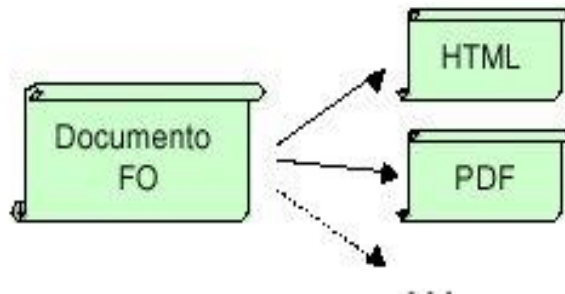
## Introducción a XSLT (I): Transformación de documentos XML

Grado en  
Ingeniería  
Informática  
del Software

- XSL (eXtensible Stylesheet Language)
  - XSL Transformations (XSLT)
    - Lenguaje de transformación de documentos
    - Utiliza XPath para hacer referencias a partes de un documento



- XSL-Formatting Objects (XSL-FO)
  - Lenguaje que incluye instrucciones de formato independientes del dispositivo



# Software y estándares para la Web

## Introducción a XSLT (II): Presentación de un documento XML

Grado en  
Ingeniería  
Informática  
del Software

- **XML no incorpora ninguna semántica intrínseca de presentación**
- Soluciones
  - **XML+CSS**
  - **XML+XSL**\_(Extensible Stylesheet Language, "lenguaje extensible de hojas de estilo").
    - Familia de lenguajes basados en XML
    - Permite describir cómo la información contenida en un documento XML debe ser transformada o formateada para su presentación en un medio específico

# Software y estándares para la Web

## Introducción a XSLT (III): ¿Por qué XSL?

Grado en  
Ingeniería  
Informática  
del Software

- Está diseñado para integrarse en la arquitectura XML.
- Es mucho más potente que CSS.
  - CSS no tiene capacidades de transformación.
- Cada vez hay más herramientas para hacer transformaciones XSLT.
- Es un **estándar del W3C**
  - Recomendación del 23-enero-2007
  - Versión 2.0
  - <http://www.w3.org/TR/xslt20/>

# Software y estándares para la Web

## Introducción a XSLT (IV): CSS versus XSL

Grado en  
Ingeniería  
Informática  
del Software

Cuestiones	CSS	XSL
<i>¿Puede ser usada con HTML?</i>	SI	NO
<i>¿Puede ser usado con XML?</i>	SI	SI
<i>¿Lenguajes de transformación?</i>	NO	SI
<i>¿Sintaxis?</i>	CSS	XML

# Software y estándares para la Web

## Introducción a XSLT (V): XSL

Grado en  
Ingeniería  
Informática  
del Software

- Formado por:
  - **XSLT** (*Extensible Stylesheet Language Transformations*): permite convertir documentos XML de una sintaxis a otra (por ejemplo, de un XML a otro o a un documento **HTML**)
  - **XSL-FO** (*lenguaje de hojas extensibles de formateo de objetos*): permite especificar el formato visual con el cual se quiere presentar un documento XML, es usado principalmente para generar documentos PDF
  - **XPath**, o XML Path Language: *sintaxis (no basada en XML) para acceder o referirse a porciones de un documento XML*

Estas tres especificaciones son recomendaciones oficiales del W3C  
<http://www.w3.org/standards/xml/>



# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- **Transformaciones XSLT**
- Estructura de los documentos XSL
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios propuestos

- **Transformación estructural del XML**
  - Los datos son convertidos de la estructura de entrada (XML) a una estructura que refleje la salida deseada
- **Formato de salida**
  - La salida generada por la transformación del documento se entrega en el formato requerido
  - Formatos:
    - PDF
    - HTML
    - Textos delimitado por comas
    - Etc.

# Software y estándares para la Web

## Transformaciones XSLT (II): ¿Por qué transformar XML?

Grado en  
Ingeniería  
Informática  
del Software

- **Conversión entre modelos de datos**
  - Aunque legible, XML está pensado para el intercambio de información entre aplicaciones.
- Es necesaria una **capa de presentación** para hacer “amigable” el acceso de los humanos a la información XML.
  - XML produce separación entre el modelo de datos y presentación visual.

# Software y estándares para la Web

## Transformaciones XSLT (III): ¿Cómo funciona?

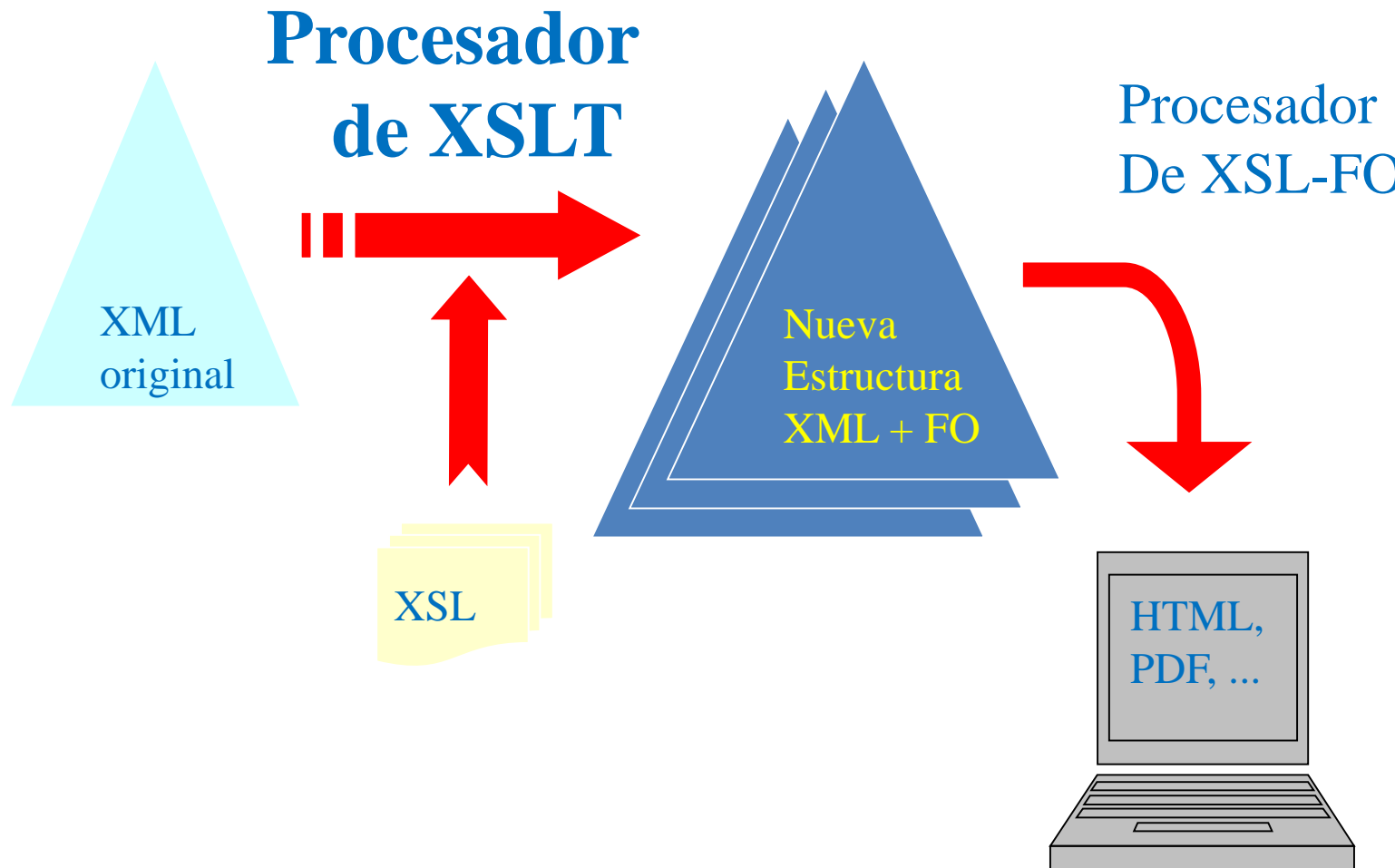
Grado en  
Ingeniería  
Informática  
del Software

- Durante el **proceso de transformación**, XSLT utiliza **XPath** para definir partes del documento fuente que encajan dentro de una o mas plantillas predefinidas
- Por **cada coincidencia**, XSLT transformará esta parte del documento fuente para generar el documento resultante
- La porción del documento fuente que **no encaja** con la plantilla permanecerá **sin modificación** alguna

# Software y estándares para la Web

## Transformaciones XSLT (IV): Proceso de publicación XML-XSL

Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Transformaciones XSLT (V): Ejemplo transformando a HTML (a)

poema.xsl

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="html"/>

  <xsl:template match="/">
  <html>
    <head>
      <title>Ejemplo</title>
    </head>
    <body>
      <h1>Poema</h1>
    </body>
  </html>
  </xsl:template>

</xsl:stylesheet>
```

Espacio de  
nombres

Plantilla

# Software y estándares para la Web

## Transformaciones XSLT (VI): Reglas de plantillas

Grado en  
Ingeniería  
Informática  
del Software

- Las reglas de plantillas (*templates*) se identifican mediante:

```
<xsl:template match="expresión Xpath">  
... resultado generado...  
</xsl:template>
```

- El valor del atributo match es una expresión XPath
- El procesador XSLT recorre el árbol del documento XML y encuentra un nodo determinado que coincide con la expresión dada, entonces genera el resultado de la plantilla

# Software y estándares para la Web

## Transformaciones XSLT (VII): ¿Cómo asociar una hoja de estilos?

Grado en  
Ingeniería  
Informática  
del Software

- Es posible indicar en el documento XML qué hoja de estilos XSL lleva asociada o también una CSS:
  - Type: “text/css” (CSS) o “text/xsl” (XSL)”
- Un **navegador** puede reconocer dicha información y mostrar el resultado de la transformación
- Los navegadores pueden abrir y mostrar el contenido de un archivo XML asociado a una hoja de estilos o a una CSS

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



## Transformaciones XSLT (VI): Ejemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="poema.xsl"?>
<poema autor="Federico García Lorca" fecha="Abril de 1915" lugar="Granada">
<titulo>Alba</titulo>
<verso>Mi corazón oprimido</verso>
<verso>late junto a la alborada</verso>
...
</poema>
```

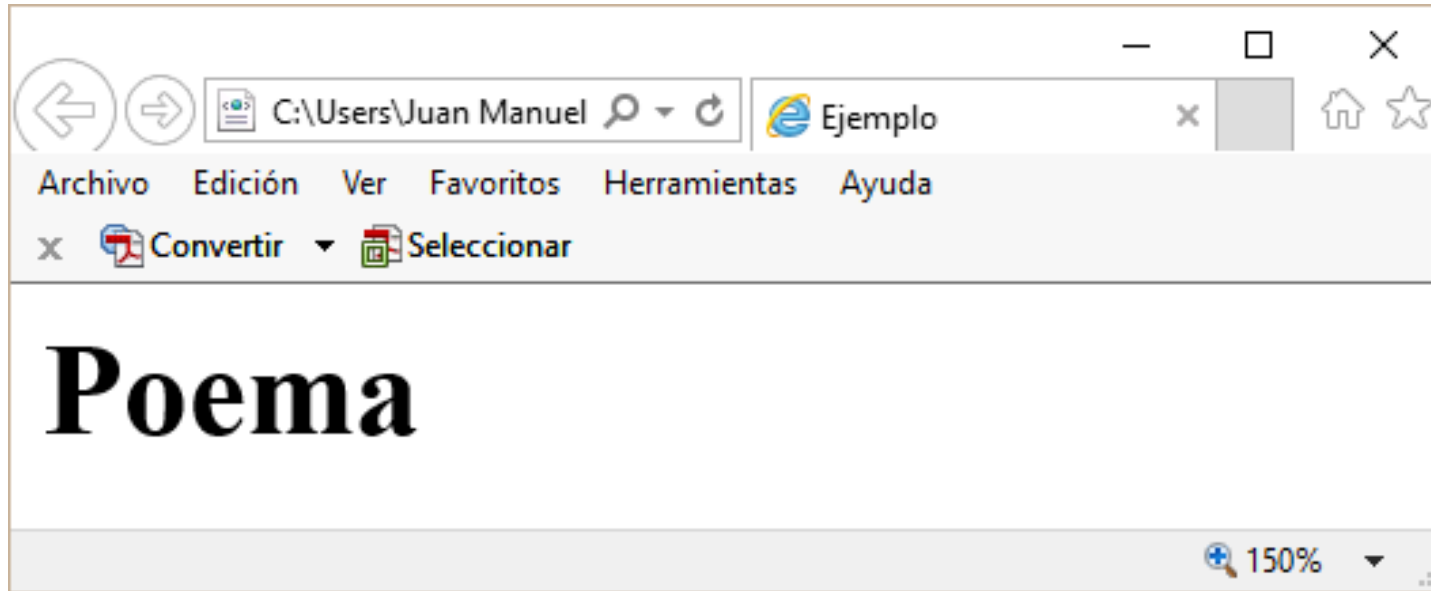
Poema.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="/">
<html><head><title>Ejemplo</title></head>
  <body><h1>Poema</h1></body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# Software y estándares para la Web

## Transformaciones XSLT (VII): Visualización en el navegador

Grado en  
Ingeniería  
Informática  
del Software



```
alba.xml
1 <html>
2 <head>
3 <META http-equiv="Content-Type" content="text/html; charset=UTF-16">
4 <title>Ejemplo</title>
5 </head>
6 <body>
7 <h1>Poema</h1>
8 </body>
9 </html>
```

# Software y estándares para la Web

## Transformaciones XSLT (VIII): Obtener valores de atributos

poemaAutorLugarFecha.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="html"/>

  <xsl:template match="poema">
    Autor : <xsl:value-of select="@autor"/>
    <br/>
    Fecha : <xsl:value-of select="@fecha"/>
    <br/>
    Lugar : <xsl:value-of select="@lugar"/>
    <br/>
  </xsl:template>

</xsl:stylesheet>
```

Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

## Transformaciones XSLT (IX): Obtener valores de atributos

Grado en  
Ingeniería  
Informática  
del Software

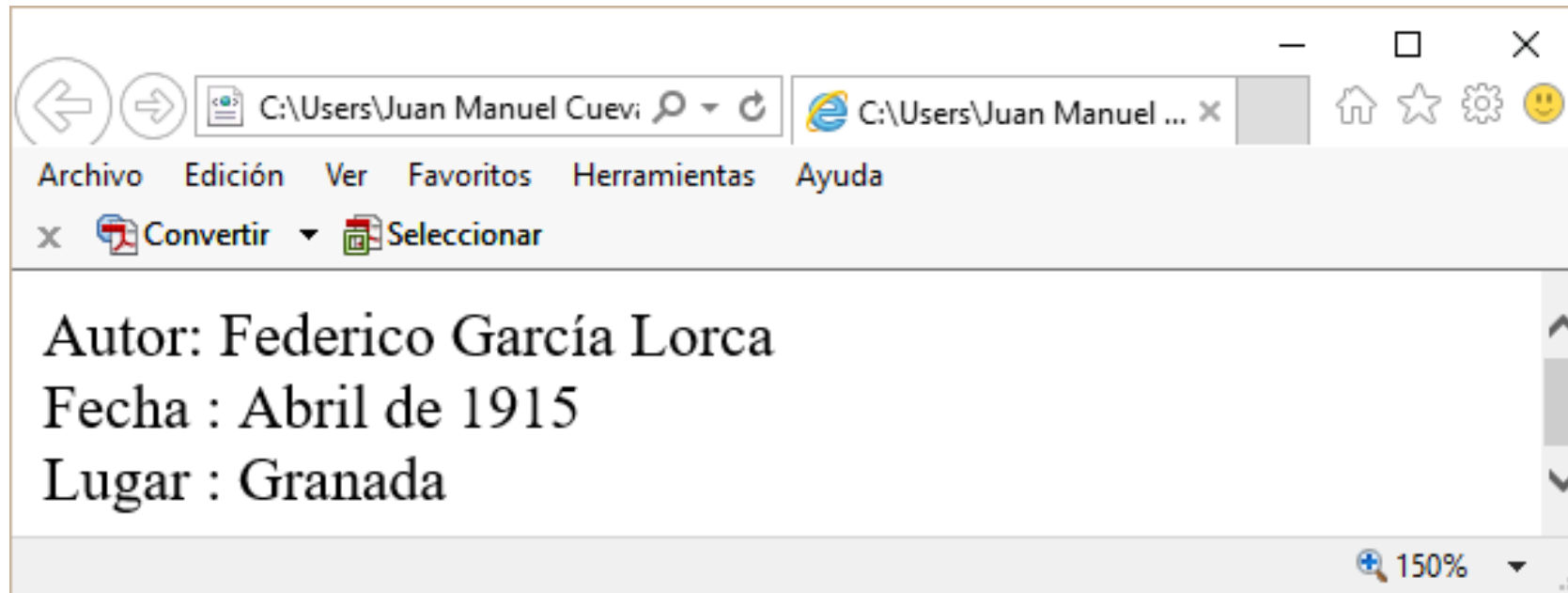
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="poemaAutorLugarFecha.xsl"?>
<poema autor="Federico García Lorca" fecha="Abril de 1915"
lugar="Granada">
<titulo>Alba</titulo>
<verso>Mi corazón oprimido</verso>
<verso>siente junto a la alborada</verso>
<verso>el dolor de sus amores</verso>
<verso>Y el sueño de las distancias.</verso>
<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-atributos.xml

# Software y estándares para la Web

## Transformaciones XSLT (X): Visualización en el navegador

Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Transformaciones XSLT (XI): Ejemplo para obtener el texto de un elemento

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version= "1.0" encoding="UTF-8"?>
  <?xml-stylesheet type= "text/xsl" href ="intro.xsl"?>
    <miCurso>
      <curso>Máster en Ingeniería Web</curso>
    </miCurso>
```

```
<?xml version= "1.0" encoding="UTF-8"?>
<xsl:stylesheet version ="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match = "miCurso">
    <html>
      <body><p><xsl:value-of select="curso"/></p></body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

```
<html>
<body><p>Máster en Ingeniería Web</p></body>
</html>
```

# Software y estándares para la Web

## Transformaciones XSLT (XII): Ejemplo para obtener el texto de un elemento

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="poemaTitulo.xsl"?>
<poema autor="Federico García Lorca" 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>
<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-titulo.xml

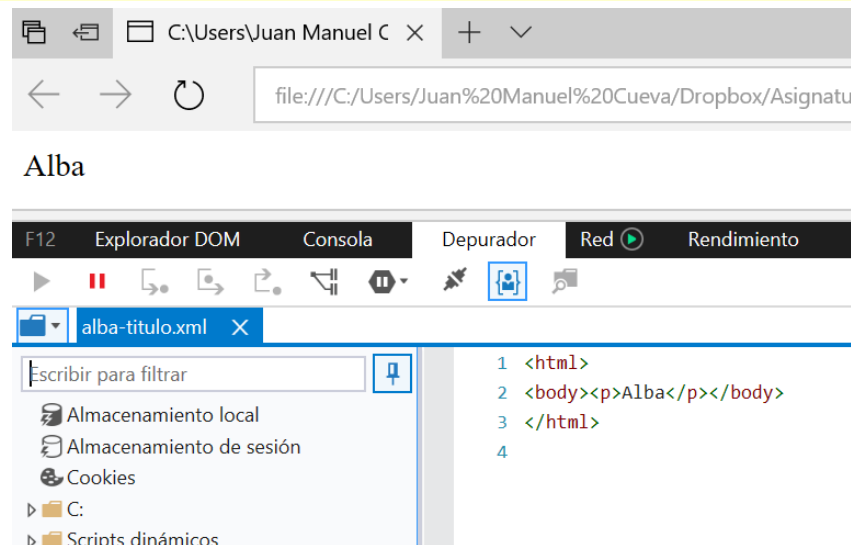
# Software y estándares para la Web

## Transformaciones XSLT (XIII): Ejemplo para obtener el texto de un elemento

poemaTitulo.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="poema">
<html>
    <body><p><xsl:value-of select="titulo"/></p></body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Grado en  
Ingeniería  
Informática  
del Software





# Software y estándares para la Web

## Transformaciones XSLT (XIV): Ejemplo de atributos y elemento

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="poemaTituloFechaLugar.xsl"?>
<poema autor="Federico García Lorca" 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>
<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>
```

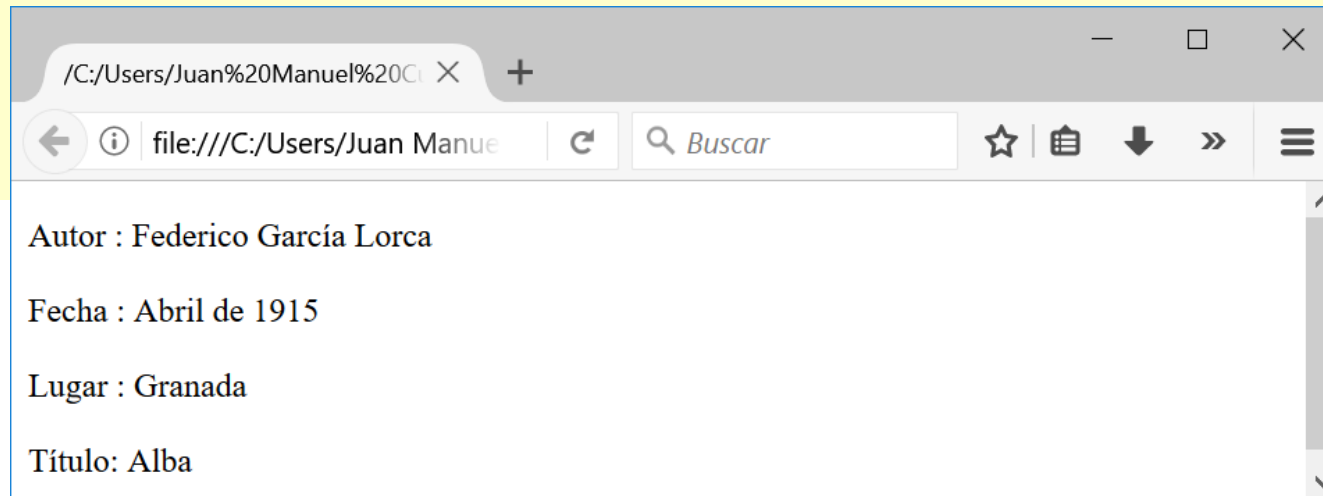
alba-titulo-fecha-lugar.xml

# Software y estándares para la Web

## Transformaciones XSLT (XV): Ejemplo de elemento y atributos

poemaTituloFechaLugar.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="poema">
<html>
<body>
  <p>Autor : <xsl:value-of select="@autor"/></p>
  <p>Fecha : <xsl:value-of select="@fecha"/></p>
  <p>Lugar : <xsl:value-of select="@lugar"/></p>
  <p>Título :<xsl:value-of select="titulo"/></p>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```



Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

## Transformaciones XSLT (XVI): Ejemplo **alba-completo.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="poemaCompleto.xsl"?>
<poema autor="Federico García Lorca" 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>
<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-completo.xml

# Software y estándares para la Web

## Transformaciones XSLT (XVII): Ejemplo poemaCompleto.xls

poemaCompleto.xsl

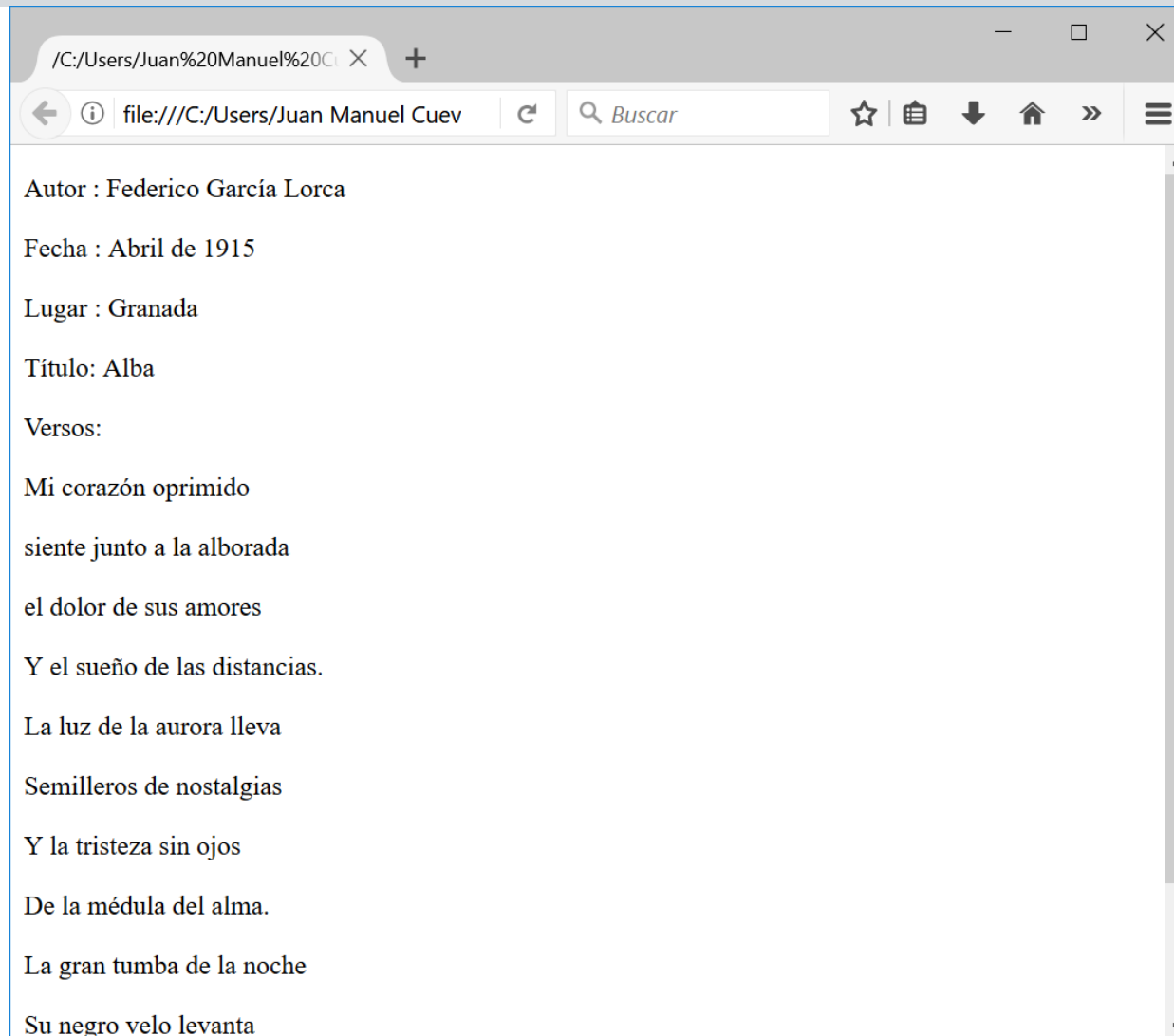
Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="poema">
<html>
<body>
  <p>Autor : <xsl:value-of select="@autor"/></p>
  <p>Fecha : <xsl:value-of select="@fecha"/></p>
  <p>Lugar : <xsl:value-of select="@lugar"/></p>
  <p>Título :<xsl:value-of select="titulo"/></p>
  <p>Versos:</p>
  <xsl:for-each select="verso">
    <p><xsl:value-of select="."/></p>
  </xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# Software y estándares para la Web

## Transformaciones XSLT (XVIII): Ejemplo poemaCompleto.xls

Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- **Estructura de los documentos XSL**
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios propuestos

# Software y estándares para la Web

## Estructura de los documentos XSL (I)

Grado en  
Ingeniería  
Informática  
del Software

- **Elementos de XSLT**

- Pertenecen al *namespace xsl*
- Sus etiquetas llevan el *prefijo xsl*
  - Son el equivalente a las palabras clave del lenguaje de programación
  - Definidos por el estándar
  - Interpretados por cualquier procesador de XSLT

- **Elementos LRE (Literal Result Elements)**

- Son elementos que no pertenecen a XSLT, sino que se repiten en la salida sin más.
- No son parte del estándar

- **Elementos de extensión**

- Son elementos no-estándar (al igual que los LRE), que son manejados por implementaciones concretas del procesador.
- No son parte del estándar
- No los utilizaremos

# Software y estándares para la Web

## Estructura de los documentos XSL (II): <xsl:stylesheet>

Grado en  
Ingeniería  
Informática  
del Software

- Es el elemento raíz de una hoja XSL
  - **version**: Suele ser 1.0
  - **xmlns:xsl**: Asigna el *namespace xsl*
    - las etiquetas de XSL empiezan por el prefijo xsl:
  - El valor para XSLT suele ser  
**<http://www.w3.org/1999/XSL/Transform>**

```
<xsl:stylesheet version ="1.0"  
xmlns:xsl= "http://www.w3.org/1999/XSL/Transform">  
...  
</xsl:stylesheet>
```



# Software y estándares para la Web

## Estructura de los documentos XSL (III): Elementos de nivel superior

Grado en  
Ingeniería  
Informática  
del Software

- Son elementos hijos de `xsl:stylesheet`
- Son hijos directos (tampoco pueden anidarse)
- Dos excepciones: `xsl:variable` y `xsl:param`
- No son instrucciones sobre cómo procesar elementos, sino estructuras contenedoras para instrucciones

<code>xsl:output</code>	<code>xsl:strip-space</code>	<code>xsl:namespace-alias</code>
<code>xsl:template</code>	<code>xsl:preserve-space</code>	<code>xsl:attribute-set</code>
<code>xsl:include</code>	<code>xsl:key</code>	<code>xsl:variable</code>
<code>xsl:import</code>	<code>xsl:decimal-format</code>	<code>xsl:param</code>

# Software y estándares para la Web

## Estructura de los documentos XSL (IV): `<xsl:output>`

Grado en  
Ingeniería  
Informática  
del Software

- Define qué tipo de salida se va a generar como resultado
  - **method**: xml, html o text
  - **encoding**: define la forma de representar caracteres que se adoptará en la salida (iso-8859-1, **UTF-8**, UTF-16, windows-1252)
  - **omit-xml-declaration**: yes o no. Indica si se genera o no la declaración `<?xml...?>`
  - **indent**: yes o no. Si es yes, el procesador (para salidas xml o html) indentará el resultado

- **<xsl:include>** que permite referenciar plantillas procedentes de una fuente externa
- **<xsl:strip-space>** que elimina antes del procesamiento todos los modos consistentes en espacios en blanco
- **<xsl:preserve-space>** mantiene los espacios en blanco

# Software y estándares para la Web

## Estructura de los documentos XSL (VI): Reglas de transformación

Grado en  
Ingeniería  
Informática  
del Software

- Es el elemento básico y fundamental del lenguaje de transformación XSLT
- Una **regla** (o “**template**”) consta de dos partes:
  - Una “**etiqueta**” formada por un patrón de localización que selecciona nodos en el árbol XML origen sobre los que se aplica la transformación
    - **Expresión XPath.**
  - Una “**acción**” que indica la transformación a realizar sobre los nodos seleccionados.
- Cada hoja de estilo XSL debe contener al menos una regla si se quiere que ejecute algo

# Software y estándares para la Web

## Estructura de los documentos XSL (VII): Obtención de patrones

Grado en  
Ingeniería  
Informática  
del Software

- **<xsl:template match= “...”>**
- **<xsl:apply-templates select=“...”>**

# Software y estándares para la Web

## Estructura de los documentos XSL (VII): `<xsl:template match = . . .>`

Grado en  
Ingeniería  
Informática  
del Software

- Cada etiqueta `<xsl:template>` contiene reglas por aplicar a ciertos elementos de un nodo
  - `match` es utilizado para asociar la plantilla con un elemento XML. El atributo `match` también puede utilizarse para aplicarse a una rama completa del documento XML (Por ejemplo `match="/"` define a todo el documento)
  - `name` Además de cuando encaja, un *template* puede invocarse explícitamente (en ese caso se necesita que tenga un nombre)
- Ejemplo: `<xsl:template match="/">`

# Software y estándares para la Web

## Estructura de los documentos XSL (VIII): Ejemplo

Grado en  
Ingeniería  
Informática  
del Software

```
<xslTutorial >  
<bold>Hola a todos</bold>  
<red>I am </red>  
<italic>fine</italic>  
</xslTutorial>
```

```
<p> <b>Hola a todos</b></p>  
<p style="color:red">I am </p>  
<p> <i>fine</i></p>
```

```
<xsl:stylesheet  
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>  
  <xsl:template match="bold">  
    <p><b><xsl:value-of select="."/></b></p>  
  </xsl:template>  
  <xsl:template match="red">  
    <p style="color:red"><xsl:value-of select="."/></p>  
  </xsl:template>  
  <xsl:template match="italic">  
    <p><i><xsl:value-of select="."/></i></p>  
  </xsl:template>  
</xsl:stylesheet>
```

Hola a todos  
I am  
fine.

# Software y estándares para la Web

## Estructura de los documentos XSL (IX): `<xsl:apply-templates>`

Grado en  
Ingeniería  
Informática  
del Software

- Se utiliza para indicar al procesador que intente emparejar *templates* con cierto nodo o conjunto de nodos (*nodeset*)
  - **select**: Su valor es una expresión **XPath** del conjunto de nodos. El procesador intentará emparejar ese conjunto de nodos con sus *templates* respectivos
- Permite realizar un **tratamiento recursivo** de todos los elementos del árbol fuente



# Software y estándares para la Web

## Estructura de los documentos XSL (X): Ejemplo **musica-apply.xml**

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="musica-apply.xsl"?>
<catalogo>
<cd>
  <title>Empire Burlesque</title>
  <artist>Bob Dylan</artist>
  <country>USA</country>
  <company>Columbia</company>
  <price>10.90</price>
  <year>1985</year>
</cd>
...
</catalogo>
```

# Software y estándares para la Web

## Estructura de los documentos XSL (XII): Ejemplo **musica-apply.xsl**

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes" />
  <xsl:template match="/">
    <html>
      <body>
        <h1>Mi catálogo de CD's</h1>
        <ul>
          <xsl:apply-templates/>
        </ul>
      </body>
    </html>
  </xsl:template>
```

# Software y estándares para la Web

## Estructura de los documentos XSL (XIII): Ejemplo **musica-apply.xsl**

Grado en  
Ingeniería  
Informática  
del Software

```
<xsl:template match="cd">  
  <li> <xsl:apply-templates select="title"/>  
  <xsl:apply-templates select="artist"/> </li>  
</xsl:template>
```

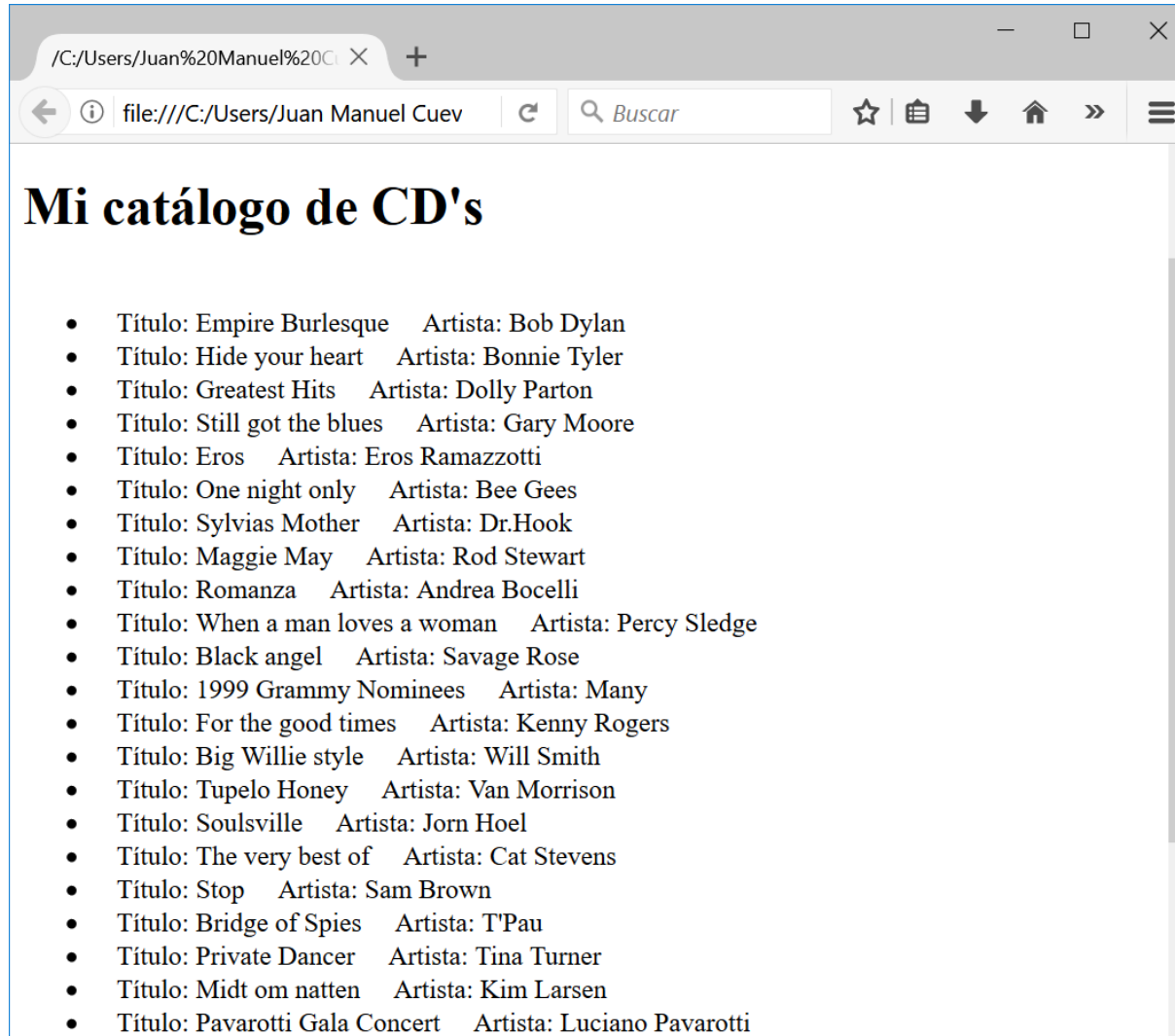
```
<xsl:template match="title">  
  Título: <xsl:value-of select="."/>  
</xsl:template>
```

```
<xsl:template match="artist">  
  Artista: <xsl:value-of select="."/>  
</xsl:template>  
</xsl:stylesheet>
```

# Software y estándares para la Web

## Estructura de los documentos XSL (XIV): Ejemplo musica-apply.xsl

Grado en  
Ingeniería  
Informática  
del Software



The screenshot shows a web browser window with the address bar displaying 'file:///C:/Users/Juan Manuel Cueva...'. The browser's address bar also contains a search icon and the text 'Buscar'. The main content area of the browser displays the title 'Mi catálogo de CD's' in a large, bold, black font. Below the title is a list of 20 items, each consisting of a bullet point followed by the title and artist of a CD. The list is as follows:

- Título: Empire Burlesque    Artista: Bob Dylan
- Título: Hide your heart    Artista: Bonnie Tyler
- Título: Greatest Hits    Artista: Dolly Parton
- Título: Still got the blues    Artista: Gary Moore
- Título: Eros    Artista: Eros Ramazzotti
- Título: One night only    Artista: Bee Gees
- Título: Sylvias Mother    Artista: Dr.Hook
- Título: Maggie May    Artista: Rod Stewart
- Título: Romanza    Artista: Andrea Bocelli
- Título: When a man loves a woman    Artista: Percy Sledge
- Título: Black angel    Artista: Savage Rose
- Título: 1999 Grammy Nominees    Artista: Many
- Título: For the good times    Artista: Kenny Rogers
- Título: Big Willie style    Artista: Will Smith
- Título: Tupelo Honey    Artista: Van Morrison
- Título: Soulsville    Artista: Jorn Hoel
- Título: The very best of    Artista: Cat Stevens
- Título: Stop    Artista: Sam Brown
- Título: Bridge of Spies    Artista: T'Pau
- Título: Private Dancer    Artista: Tina Turner
- Título: Midt om natten    Artista: Kim Larsen
- Título: Pavarotti Gala Concert    Artista: Luciano Pavarotti

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- **XPath**
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios propuestos

- **Objetivo:** Identificar a los elementos del árbol XML
- Tienen la forma
  - `nodo1/nodo2/.../nodoN`
- Ejemplo: `persona/apellido/materno`
- Describen un camino (*path*)
- Resultado:
  - Un conjunto de nodos
  - String, número o boolean
- Es un estándar del W3C
  - <http://www.w3.org/TR/xpath20/>

# Software y estándares para la Web

## XPath(II): XPath versus Sistema de Archivos

**Grado en  
Ingeniería  
Informática  
del Software**

Archivos y directorios	Nodos dentro de nodos
Respecto a directorio actual	Respecto a nodo actual o de contexto
* = cualquier cosa	* = cualquier nodo
Un archivo por ruta	Uno o varios nodos por ruta

# Software y estándares para la Web

## XPath (III): Resumen de sintaxis abreviada XPath

Grado en  
Ingeniería  
Informática  
del Software

- X – hijo elemento “X”
- \* - todos los hijos elemento
- text() - todos los hijos texto
- @Y – atributo “Y”
- X[1] – primer hijo “X”
- X[last()] - último hijo “X”
- \*/X – nietos “X”
- X//Y – descendientes “Y” de hijo “X”
- //Y – descendientes “Y” de la RAIZ
- //Y/X - descendientes “Y” de HIJO “X”
- .. – padre
- . – todo lo de este nodo
- //X[1][@Y=”Z”] - primeros hijos X con atributo Y=”Z”



# Software y estándares para la Web

## XPath (IV): <xsl:value-of>

Grado en  
Ingeniería  
Informática  
del Software

- La etiqueta `<xsl:value-of>` se utiliza para seleccionar el valor de uno o varios nodos del árbol XML y agregarlo al archivo de salida de la transformación
- El valor del atributo **select** contiene una expresión **XPath**.
- La expresión Xpath se puede ver como la navegación de un sistema de archivos donde una diagonal (/) separa los nodos hijos
- Sintaxis:

```
<xsl:value-of select = "expresión XPath" >
```

- Ejemplo:

```
<xsl:value-of select="catalogo/cd/titulo"/ >
```

# Software y estándares para la Web

## XPath (V): Ordenar nodos `<xsl:sort>`

Grado en  
Ingeniería  
Informática  
del Software

- Permite aplicar un *template* a un conjunto de nodos ordenándolos alfabética o numéricamente
- Sintaxis:

```
<xsl:apply-templates select = “expresión Xpath” >
```

```
  <xsl:sort select= “expresión Xpath”  
            data-Type = “text | number”  
            order = “ascending | descending” />
```

```
</xsl:apply-templates>
```

# Software y estándares para la Web

## XPath (VI): Ejemplo amigos.xml

```
<amigos>
<nombre>Juan</nombre>
<nombre>Jaime</nombre>
<nombre>Carlos</nombre>
<nombre>Alicia</nombre>
<nombre>Marta</nombre>
</amigos>
```

```
<xsl:stylesheet version = '1.0'
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match="/">
  <table>
    <xsl:for-each select="//nombre">
      <xsl:sort order="ascending" select="."/>
      <tr>
        <th><xsl:value-of select="."/></th>
      </tr>
    </xsl:for-each>
  </table>
</xsl:template>
</xsl:stylesheet>
```

```
<table>
  <tr>
    <th>Alicia</th>
  </tr>
  <tr>
    <th>Carlos</th>
  </tr>
  <tr>
    <th>Jaime</th>
  </tr>
  <tr>
    <th>Juan</th>
  </tr>
  <tr>
    <th>Marta</th>
  </tr>
</table>
```

↓

**Alicia**  
**Carlos**  
**Jaime**  
**Juan**  
**Marta**

# Software y estándares para la Web

## XPath (VII): Ejemplo `<xsl:sort>` musica-sort.xml

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="musica-sort.xsl"?>
<catalogo>
<cd>
  <title>Empire Burlesque</title>
  <artist>Bob Dylan</artist>
  <country>USA</country>
  <company>Columbia</company>
  <price>10.90</price>
  <year>1985</year>
</cd>
...
</catalogo>
```

# Software y estándares para la Web

## XPath (VIII): Ejemplo <xsl:sort> musica-sort.xsl

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes" />
  <xsl:template match="/">
    <html>
    <body>
      <h1>Mi catálogo de CD's</h1>
      <table>
        <tr>
          <th>Título</th>
          <th>Artista</th>
          <th>País</th>
          <th>Compañía</th>
          <th>Precio</th>
          <th>Año</th>
        </tr>
```

# Software y estándares para la Web

## XPath (IX): Ejemplo `<xsl:sort>` musica-sort.xsl (continuación)

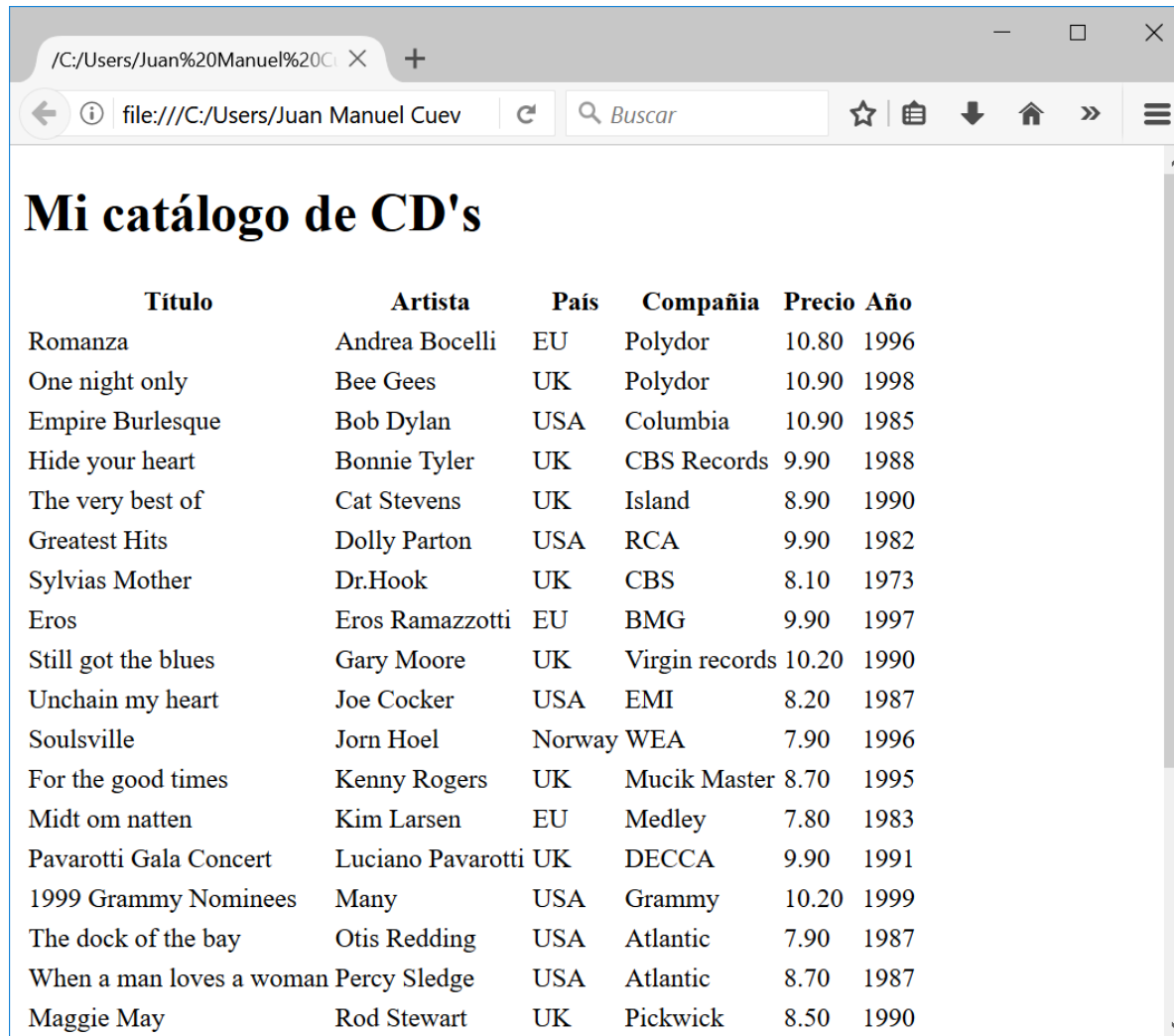
Grado en  
Ingeniería  
Informática  
del Software

```
<xsl:for-each select="catalogo/cd">
  <xsl:sort order="ascending" select="artist"/>
  <tr>
    <td><xsl:value-of select="title" /> </td>
    <td><xsl:value-of select="artist" /> </td>
    <td><xsl:value-of select="country" /> </td>
    <td><xsl:value-of select="company" /> </td>
    <td><xsl:value-of select="price" /> </td>
    <td><xsl:value-of select="year" /> </td>
  </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# Software y estándares para la Web

## XPath (X): Ejemplo <xsl:sort> musica-sort.xml

Grado en  
Ingeniería  
Informática  
del Software



**Mi catálogo de CD's**

Título	Artista	País	Compañía	Precio	Año
Romanza	Andrea Bocelli	EU	Polydor	10.80	1996
One night only	Bee Gees	UK	Polydor	10.90	1998
Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988
The very best of	Cat Stevens	UK	Island	8.90	1990
Greatest Hits	Dolly Parton	USA	RCA	9.90	1982
Sylvias Mother	Dr.Hook	UK	CBS	8.10	1973
Eros	Eros Ramazzotti	EU	BMG	9.90	1997
Still got the blues	Gary Moore	UK	Virgin records	10.20	1990
Unchain my heart	Joe Cocker	USA	EMI	8.20	1987
Soulsville	Jorn Hoel	Norway	WEA	7.90	1996
For the good times	Kenny Rogers	UK	Mucik Master	8.70	1995
Midt om natten	Kim Larsen	EU	Medley	7.80	1983
Pavarotti Gala Concert	Luciano Pavarotti	UK	DECCA	9.90	1991
1999 Grammy Nominees	Many	USA	Grammy	10.20	1999
The dock of the bay	Otis Redding	USA	Atlantic	7.90	1987
When a man loves a woman	Percy Sledge	USA	Atlantic	8.70	1987
Maggie May	Rod Stewart	UK	Pickwick	8.50	1990

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- XPath
- **Sentencias iterativas**
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios propuestos



# Software y estándares para la Web

## Sentencias iterativas (I): <xsl:for-each>

Grado en  
Ingeniería  
Informática  
del Software

- La etiqueta XSL **<xsl:for-each>** se utiliza para seleccionar todos los elementos XML del nodo especificado.
- El valor del atributo **select** contiene una expresión **XPath**. Esta trabaja como la navegación de un sistema de ficheros donde una diagonal vertical delantera (/) selecciona subdirectorios.
- **Sintaxis:**

```
<xsl:for-each select="XPATH">
```

```
  ...  
</xsl:for-each>
```

# Software y estándares para la Web

## Sentencias iterativas (II): Ejemplo

Grado en  
Ingeniería  
Informática  
del Software

```
<xslTutorial >
<list> <entry name="A"/>
      <entry name="B"/>
      <entry name="C"/>
      <entry name="D"/>
</list>
</xslTutorial>
```

```
<HTML>
<HEAD> </HEAD>
<BODY> A, B, C, D, </BODY>
</HTML>
```

```
<xsl:stylesheet
xmlns:xsl='http://www.w3.org/1999/XSL/Transform' >
<xsl:template match="list">
  <xsl:for-each select="entry">
    <xsl:value-of select="@name"/>
    <xsl:text> , </xsl:text>
  </xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

A, B, C, D,

# Software y estándares para la Web

## Sentencias iterativas (III): Ejemplo musica.xml

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="musica.xsl"?>
<catalogo>
<cd>
  <title>Empire Burlesque</title>
  <artist>Bob Dylan</artist>
  <country>USA</country>
  <company>Columbia</company>
  <price>10.90</price>
  <year>1985</year>
</cd>
...
</catalogo>
```

# Software y estándares para la Web

## Sentencias iterativas (IV): Ejemplo **musica.xsl**

Grade on

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes" />
<xsl:template match="/">
<html><body> <h1>Mi catálogo de CD's</h1>
  <table>
<tr>
  <th>Título</th>
  <th>Artista</th>
  <th>País</th>
  <th>Compañía</th>
  <th>Precio</th>
  <th>Año</th>
</tr>
```

# Software y estándares para la Web

## Sentencias iterativas (V): Ejemplo **musica.xsl** (continuación)

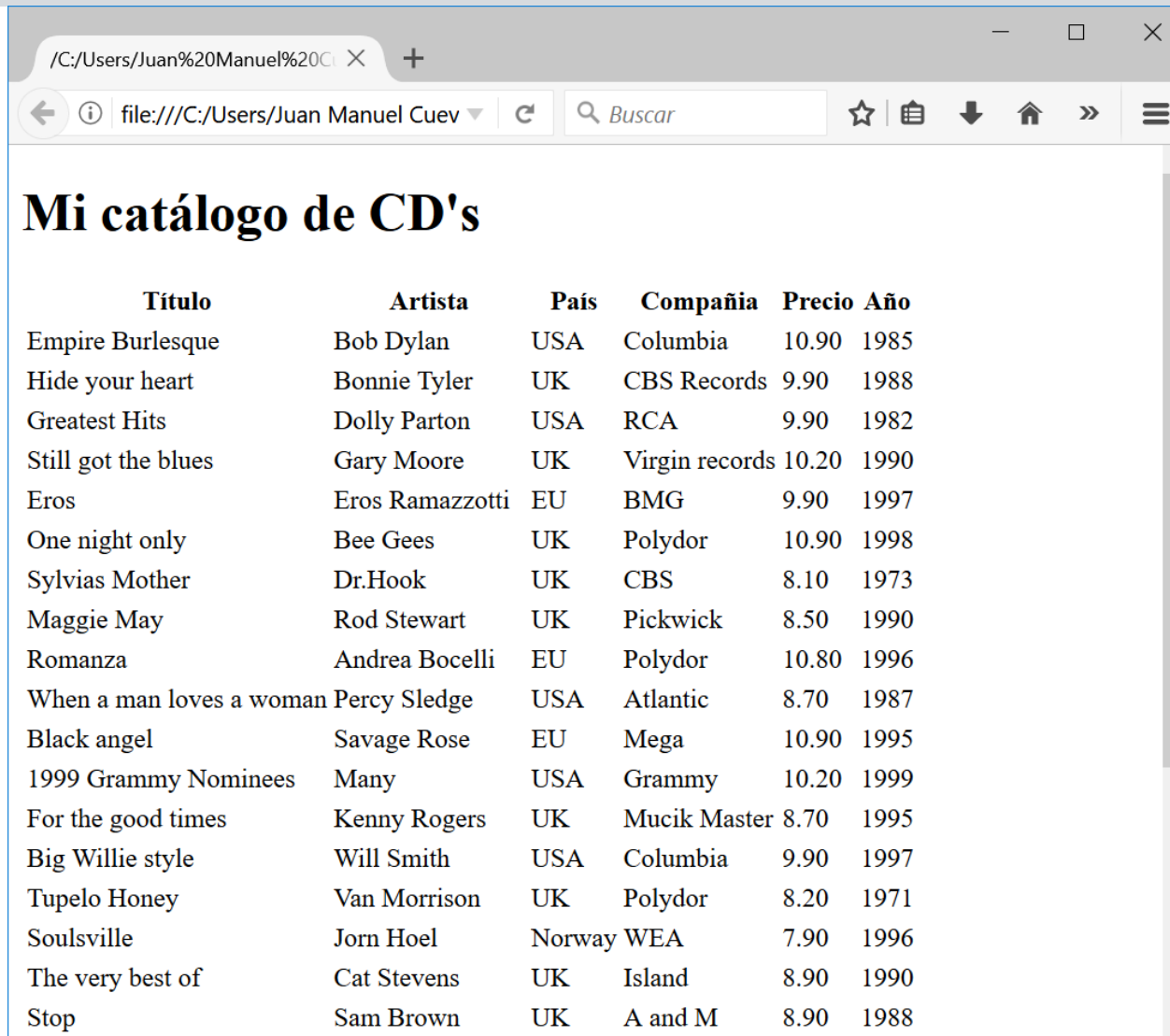
Grado en  
Ingeniería  
Informática  
del Software

```
<xsl:for-each select="catalogo/cd">
  <tr>
    <td><xsl:value-of select="title" /> </td>
    <td><xsl:value-of select="artist" /> </td>
    <td><xsl:value-of select="country" /> </td>
    <td><xsl:value-of select="company" /> </td>
    <td><xsl:value-of select="price" /> </td>
    <td><xsl:value-of select="year" /> </td>
  </tr>
</xsl:for-each>
</table>
</body></html>
</xsl:template>
</xsl:stylesheet>
```

# Software y estándares para la Web

## Sentencias iterativas (VI): Ejemplo musica.xsl

Grado en  
Ingeniería  
Informática  
del Software



**Mi catálogo de CD's**

Título	Artista	País	Compañía	Precio	Año
Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988
Greatest Hits	Dolly Parton	USA	RCA	9.90	1982
Still got the blues	Gary Moore	UK	Virgin records	10.20	1990
Eros	Eros Ramazzotti	EU	BMG	9.90	1997
One night only	Bee Gees	UK	Polydor	10.90	1998
Sylvias Mother	Dr.Hook	UK	CBS	8.10	1973
Maggie May	Rod Stewart	UK	Pickwick	8.50	1990
Romanza	Andrea Bocelli	EU	Polydor	10.80	1996
When a man loves a woman	Percy Sledge	USA	Atlantic	8.70	1987
Black angel	Savage Rose	EU	Mega	10.90	1995
1999 Grammy Nominees	Many	USA	Grammy	10.20	1999
For the good times	Kenny Rogers	UK	Mucik Master	8.70	1995
Big Willie style	Will Smith	USA	Columbia	9.90	1997
Tupelo Honey	Van Morrison	UK	Polydor	8.20	1971
Soulsville	Jorn Hoel	Norway	WEA	7.90	1996
The very best of	Cat Stevens	UK	Island	8.90	1990
Stop	Sam Brown	UK	A and M	8.90	1988

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- XPath
- Sentencias iterativas
- **Sentencias selectivas**
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios propuestos

# Software y estándares para la Web

## Sentencias selectivas (I): <xsl:if>

Grado en  
Ingeniería  
Informática  
del Software

- Permite decidir si se va a procesar o no una parte de la hoja XSL en función de una condición.
- Sintaxis:  

```
<xsl:if test='condición'>  
...  
</xsl:if>
```
- No contiene parte *else*.
- Podemos usar `=`, `!=`, `>=`, `>` y `not( )`



# Software y estándares para la Web

## Sentencias selectivas (II): Ejemplo

Grado en  
Ingeniería  
Informática  
del Software

```
<xslTutorial >  
<list> <entry name="A"/>  
      <entry name="B"/>  
      <entry name="C"/>  
      <entry name="D"/>  
</list>  
</xslTutorial>
```

```
<xsl:stylesheet  
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform' >  
  <xsl:template match="list">  
    <xsl:for-each select="entry">  
      <xsl:value-of select="@name"/>  
      <xsl:if test="not(position()=last())">  
        <xsl:text> , </xsl:text>  
      </xsl:if>  
    </xsl:for-each>  
  </xsl:template>  
</xsl:stylesheet>
```

A, B, C, D

# Software y estándares para la Web

## Sentencias selectivas (III): Ejemplo musica-if.xml

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="música-if.xsl"?>
<catalogo>
<cd>
  <title>Empire Burlesque</title>
  <artist>Bob Dylan</artist>
  <country>USA</country>
  <company>Columbia</company>
  <price>10.90</price>
  <year>1985</year>
</cd>
...
</catalogo>
```

# Software y estándares para la Web

## Sentencias selectivas (IV): Ejemplo **musica-if.xsl**

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes" />
  <xsl:template match="/">
    <html>
    <body>
      <h1>Mi catálogo de CD's ordenados por años posteriores a 1990</h1>
      <table>
        <tr>
          <th>Título</th>
          <th>Artista</th>
          <th>País</th>
          <th>Compañía</th>
          <th>Precio</th>
          <th>Año</th>
        </tr>
```

# Software y estándares para la Web

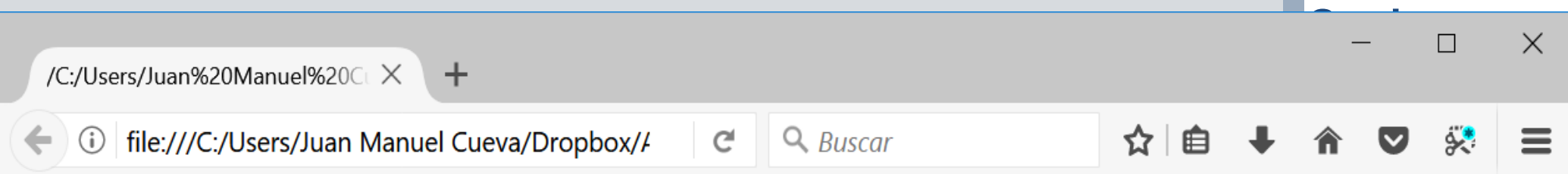
## Sentencias selectivas (IV): Ejemplo **musica-if.xsl** (continuación)

Grado en  
Ingeniería  
Informática  
del Software

```
<xsl:for-each select="catalogo/cd">
  <xsl:sort order="ascending" select="year"/>
  <xsl:if test="year > 1990">
    <tr>
      <td><xsl:value-of select="title" /> </td>
      <td><xsl:value-of select="artist" /> </td>
      <td><xsl:value-of select="country" /> </td>
      <td><xsl:value-of select="company" /> </td>
      <td><xsl:value-of select="price" /> </td>
      <td><xsl:value-of select="year" /> </td>
    </tr>
  </xsl:if>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# Software y estándares para la Web

## Sentencias selectivas (V): Ejemplo musica-if.xml



## Mi catálogo de CD's ordenados por años posteriores a 1990

Título	Artista	País	Compañía	Precio	Año
Pavarotti Gala Concert	Luciano Pavarotti	UK	DECCA	9.90	1991
Black angel	Savage Rose	EU	Mega	10.90	1995
For the good times	Kenny Rogers	UK	Mucik Master	8.70	1995
Romanza	Andrea Bocelli	EU	Polydor	10.80	1996
Soulsville	Jorn Hoel	Norway	WEA	7.90	1996
Eros	Eros Ramazzotti	EU	BMG	9.90	1997
Big Willie style	Will Smith	USA	Columbia	9.90	1997
One night only	Bee Gees	UK	Polydor	10.90	1998
1999 Grammy Nominees	Many	USA	Grammy	10.20	1999

# Software y estándares para la Web

## Sentencias selectivas (VI): <xsl:choose>

- Permite decidir qué parte de una hoja XSL se debe procesar en función de varias condiciones
- Contiene elementos xsl:when
  - Atributo: test (similar al de xsl:if)
  - Son los diferentes “casos” de una sentencia CASE
- Caso por defecto: xsl:otherwise (sin atributos)
- Sintaxis:

```
<xsl:choose>
  <xsl:when test="condición">
    ...
  </xsl:when>
  <xsl:when test="condición">
    ...
  </xsl:when>
  <xsl:otherwise>
    ...
  </xsl:otherwise>
</xsl:choose>
```

```

<xslTutorial >
<SECTION>
  <DATA>I need a pen.</DATA>
  <DATA>I need some paper.</DATA>
  <SUMMARY>
    I need a pen and some paper
  </SUMMARY>
</SECTION>
<SECTION>
  <DATA>I need bread.</DATA>
  <DATA>I need butter.</DATA>
</SECTION>
</xslTutorial>

```

la Web

```

<HTML> <HEAD> </HEAD>
<BODY>
  <P>SUMMARY:
  I need a pen
  and some paper</P>
  <P>DATA: I need bread.
  </P>
  <P>DATA: I need butter.
  </P>
</BODY> </HTML>

```

Grado en  
Ingeniería  
Informática  
del Software

SUMMARY: I need a pen  
and some paper  
DATA: I need bread.  
DATA: I need butter.

```

<xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform' >
  <xsl:template match="//SECTION">
    <xsl:choose>
      <xsl:when test='SUMMARY'>
        <P><xsl:text>SUMMARY:</xsl:text><xsl:value-of select="SUMMARY"/></P>
      </xsl:when>
      <xsl:otherwise>
        <xsl:for-each select="DATA">
          <P><xsl:text> DATA: </xsl:text> <xsl:value-of select="."/></P>
        </xsl:for-each>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>

```

# Software y estándares para la Web

## Sentencias selectivas (VIII): Ejemplo **musica-choose.xml**

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="musica-choose.xsl"?>
<catalogo>
<cd>
  <title>Empire Burlesque</title>
  <artist>Bob Dylan</artist>
  <country>USA</country>
  <company>Columbia</company>
  <price>10.90</price>
  <year>1985</year>
</cd>
...
</catalogo>
```



# Software y estándares para la Web

## Sentencias selectivas (IX): Ejemplo musica-choose.xsl

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
  <h1>Mi catálogo de CD's</h1>
  <h2>Marcados con *** los de precio superior a 10</h2>
  <table>
    <tr>
      <th>Título</th>
      <th>Artista</th>
    </tr>
```

# Software y estándares para la Web

## Sentencias selectivas (X): Ejemplo **musica-choose.xsl** (continuación)

Grado en  
Ingeniería  
Informática  
del Software

```
<xsl:for-each select="catalogo/cd">
  <xsl:sort order="ascending" select="artist"/>
  <tr>
    <td><xsl:value-of select="title"/></td>
    <xsl:choose>
      <xsl:when test="price > 10">
        <td>
          ***
          <xsl:value-of select="artist"/>
          ***
        </td>
      </xsl:when>
      <xsl:otherwise>
        <td><xsl:value-of select="artist"/></td>
      </xsl:otherwise>
    </xsl:choose>
  </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# Software y estándares para la Web

## Sentencias selectivas (X): Ejemplo musica-choose.xml

Grado en  
Ingeniería  
Informática  
del Software



The screenshot shows a web browser window with the address bar displaying a file path. The page title is "Mi catálogo de CD's". Below the title, there is a section header "Marcados con \*\*\* los de precio superior a 10". The main content is a table with two columns: "Título" and "Artista". The table lists 15 CD titles and their respective artists. The first three entries are marked with "\*\*\*" on both sides, indicating a price above 10. The browser's address bar shows the file path: file:///C:/Users/Juan Manuel Cueva/Dropbox/... The search bar contains the text "Buscar". The browser's toolbar includes icons for back, forward, home, and other standard functions.

Título	Artista
Romanza	*** Andrea Bocelli ***
One night only	*** Bee Gees ***
Empire Burlesque	*** Bob Dylan ***
Hide your heart	Bonnie Tyler
The very best of	Cat Stevens
Greatest Hits	Dolly Parton
Sylvias Mother	Dr.Hook
Eros	Eros Ramazzotti
Still got the blues	*** Gary Moore ***
Unchain my heart	Joe Cocker
Soulsville	Jorn Hoel
For the good times	Kenny Rogers
Midt om natten	Kim Larsen
Pavarotti Gala Concert	Luciano Pavarotti

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- **Construcción de elementos en el árbol resultante**
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios propuestos

# Software y estándares para la Web

## Construcción de elementos en el árbol resultante (I)

Grado en  
Ingeniería  
Informática  
del Software

- ¿Cómo generar un elemento con cierta etiqueta y “construir” sus atributos?
- Se pueden utilizar los llamados **AVT (Attribute Value Template)**:
  - las expresiones entre llaves se evalúan como si hubiera un value-of
  - Para poner llaves "de verdad", poner cada una dos veces
- Se pueden necesitar instrucciones para “construir” dichos elementos
- **<xsl:element>**: Construye un elemento en el árbol resultado
  - Atributos: name
- **<xsl:attribute>**: añade un atributo al elemento
  - Atributos: name
  - El valor está encerrado como texto libre dentro de xsl:attribute
- Se puede generar XML a partir de XML

# Software y estándares para la Web

Construcción de elementos en el árbol resultante (II): Ejemplo `elementos2atributos.xml`

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="elementos2atributos.xsl"?>
<poema>
  <verso>verso1</verso>
  <verso>verso2</verso>
  <verso>verso3</verso>
  <verso>verso4</verso>
  <verso>verso5</verso>
</poema>
```

# Software y estándares para la Web

Construcción de elementos en el árbol resultante (III): Ejemplo `elementos2atributos.xsl`

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >

<xsl:output method="xml" indent="yes"/>

<xsl:template match="//verso">
  <xsl:element name="elemento">
    <xsl:attribute name="{name()}">
      <xsl:value-of select="."/>
    </xsl:attribute>
  </xsl:element>
</xsl:template>

</xsl:stylesheet>
```

# Software y estándares para la Web

Construcción de elementos en el árbol resultante (IV): Ejemplo. **resultado**

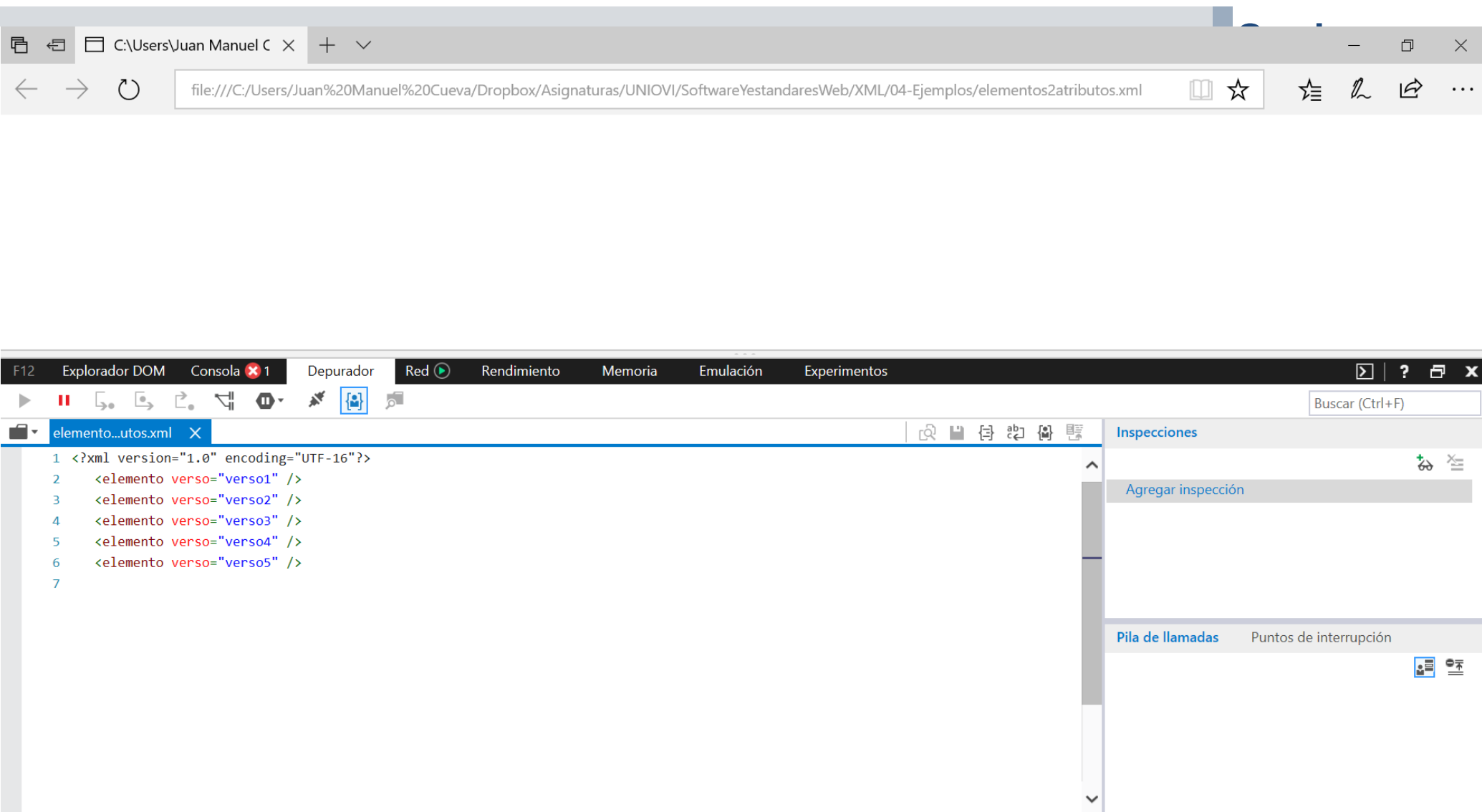
Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-16"?>  
  <elemento verso="verso1" />  
  <elemento verso="verso2" />  
  <elemento verso="verso3" />  
  <elemento verso="verso4" />  
  <elemento verso="verso5" />
```



# Software y estándares para la Web

Construcción de elementos en el árbol resultante (V): Ejemplo. **resultado**



# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- **Variables y parámetros**
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios propuestos

# Software y estándares para la Web

## Variables y parámetros (I): `<xsl:variable>`

Grado en  
Ingeniería  
Informática  
del Software

- Las variables en XSLT **no son verdaderas variables**, debido a que sus valores no se pueden cambiar
- Son hijas directas del elemento `<xsl:stylesheet>`
- Se podría decir que se parecen más a las constantes de los lenguajes de programación convencionales
- Solamente se puede cambiar una variable si está declarada dentro de un bucle `<xsl:for-each >`, en este caso el valor cambia en cada iteración
- Mediante el elemento `<xsl:variable>` podemos definir variables
- Si se usa como **elemento de alto nivel** declaramos una variable con validez **global**

# Software y estándares para la Web

## Variables y parámetros (II): `<xsl:variable>`

Grado en  
Ingeniería  
Informática  
del Software

- Si el elemento `<xsl:variable>` se encuentra dentro de una plantilla, declaramos una variable **local**.
- Las variables globales son también válidas cuando se importan de otras hojas de estilo.
- Es posible **asignar un valor** a las variables a través del atributo **select**, o bien a través del contenido del elemento, que tendrá la forma de una cadena o secuencia de caracteres.
- Para realizar una **llamada del valor de la variable** debe anteponerse el carácter `$` al nombre de la variable.
- Si se definen dentro de `<xsl:template>` son locales y NO conservan el valor de una llamada a otra llamada de la plantilla o *template*

- Sintaxis:

```
<xsl:variable name="nombre" select="expresión XPath"/>
```

- Ejemplo:

```
<xsl:for-each select = "catalogo/cd/price" />  
  <xsl:variable name = "precio" select = "." />  
  <xsl:value-of select = "$precio" />  
</xsl:for-each>
```

# Software y estándares para la Web

## Variables y parámetros (IV): Ejemplo **musica-var.xml**

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="musica-var.xsl"?>
<catalogo>
<cd>
  <title>Empire Burlesque</title>
  <artist>Bob Dylan</artist>
  <country>USA</country>
  <company>Columbia</company>
  <price>10.90</price>
  <year>1985</year>
</cd>
...
</catalogo>
```

# Software y estándares para la Web

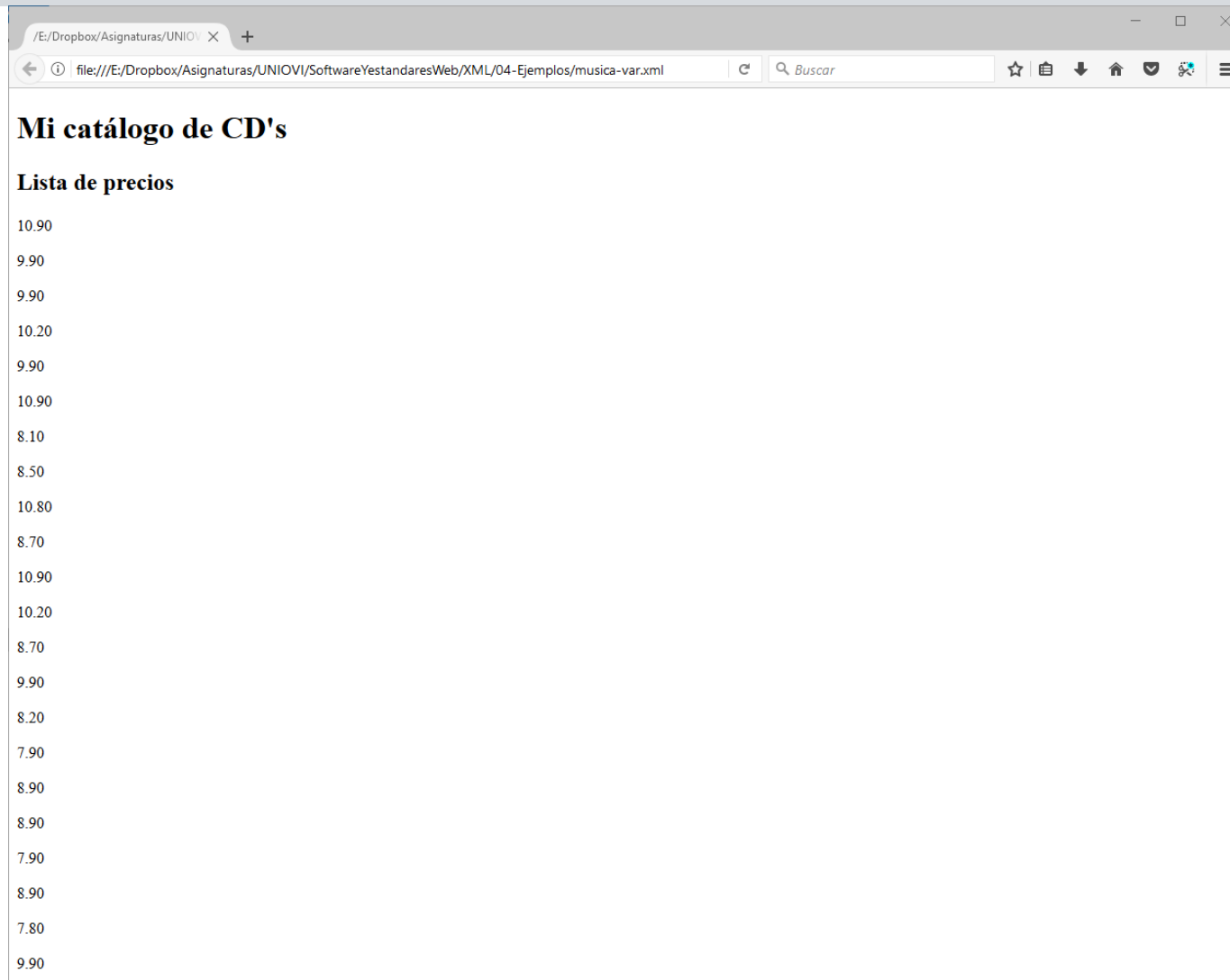
## Variables y parámetros (V): Ejemplo musica-var.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes" />
<xsl:template match="/">
<html>
<body>
<h1>Mi catálogo de CD's</h1>
<h2>Lista de precios</h2>
  <xsl:for-each select="catalogo/cd/price">
    <xsl:variable name = "precio" select = "." />
    <p><xsl:value-of select = "$precio" /></p>
  </xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# Software y estándares para la Web

## Variables y parámetros (VI): Ejemplo musica-var.xsl

Grado en  
Ingeniería  
Informática  
del Software



**Mi catálogo de CD's**

**Lista de precios**

10.90
9.90
9.90
10.20
9.90
10.90
8.10
8.50
10.80
8.70
10.90
10.20
8.70
9.90
8.20
7.90
8.90
8.90
7.90
8.90
7.80
9.90



# Software y estándares para la Web

## Variables y parámetros (VII): `<xsl:param>`

Grado en  
Ingeniería  
Informática  
del Software

- Mediante el elemento `<xsl:param>` podemos definir parámetros para la llamada a las plantillas
- Se utiliza para definir parámetros en las plantillas
- Si se usa como elemento de alto nivel declaramos un parámetro con validez **global**
- Si el elemento `<xsl:param>` se encuentra dentro de una plantilla, es un parámetro **local**
- Para realizar una **llamada del parámetro** debe anteponerse el carácter `$` al nombre del parámetro.

# Software y estándares para la Web

## Variables y parámetros (VIII): Sintaxis <xsl:param>

Grado en  
Ingeniería  
Informática  
del Software

- Sintaxis:

```
<xsl:param name="nombre" select="expresión XPath"/>
```

- Ejemplo:

```
<xsl:param name = "titulo" select ="catalogo/cd/titulo" />
```

# Software y estándares para la Web

## Variables y parámetros (IX): `<xsl:with-param>`

Grado en  
Ingeniería  
Informática  
del Software

- Este elemento define que un parámetro va a ser pasado a una plantilla como argumento
- El valor de `<xsl:with-param>` debe coincidir con el nombre de un elemento `<xsl:param>`
- El elemento `<xsl:with-param>` se permite dentro de `<xsl:apply-templates>` y `<xsl:call-template>`

# Software y estándares para la Web

## Variables y parámetros (X): Ejemplo **musica-par.xml**

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="musica-par.xsl"?>
<catalogo>
<cd>
  <title>Empire Burlesque</title>
  <artist>Bob Dylan</artist>
  <country>USA</country>
  <company>Columbia</company>
  <price>10.90</price>
  <year>1985</year>
</cd>
...
</catalogo>
```

# Software y estándares para la Web

## Variables y parámetros (XI): Ejemplo musica-par.xsl

Grado en  
Ingeniería  
Informática  
del Software

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes" />

  <xsl:param name="ordenadoPor" select="'artist'"/>
  <xsl:param name="orden" select="'ascending'"/>

  <xsl:template match="/">
    <html>
    <body>
    <h1>Mi catálogo de CD's</h1>
    <h2>Listado de artistas ordenado</h2>
    <xsl:for-each select="catalogo/cd">
      <xsl:sort select = "[name()=$ordenadoPor]" order="{ $orden }"/>
    <p><xsl:value-of select="[name()=$ordenadoPor]"/></p>
    </xsl:for-each>
    </body>
    </html>
  </xsl:template>

</xsl:stylesheet>
```

# Software y estándares para la Web

## Variables y parámetros (XII): Ejemplo musica-par.xsl

Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- **Conclusiones**
- Bibliografía
- Referencias Web
- Ejercicios propuestos

- **XSLT o Transformaciones XSL** es un estándar de la organización [W3C](#) que presenta una forma de transformar documentos [XML](#) en otros documentos
- **Estándar del W3C**
  - Recomendación del 23-enero-2007
  - Versión 2.0
  - <http://www.w3.org/TR/xslt20/>



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

  <ITEM>
    <TITLE>Vase and Stones</TITLE>
    <ARTIST>Linda Mann</ARTIST>
    <DIMENSIONS>20x30 inches</DIMENSIONS>
    <MATERIALS>oil</MATERIALS>
    <YEAR>1996</YEAR>
    <DESCRIPTION>Still Life</DESCRIPTION>
    <PREVIEW-SMALL src="burl-s.jpg" width="300"
height="194" alt="Vase and Stones"/>
    <BIDS>
      <BID>
        <PRICE>6000</PRICE>
        <TIME>3:02:22 PM</TIME>
        <BIDDER>Chris</BIDDER>
        <TIMESTAMP>1307</TIMESTAMP>
      </BID>
      <BID>
        <PRICE>5700</PRICE>
        <TIME>2:58:42 PM</TIME>
        <BIDDER>John</BIDDER>
        <TIMESTAMP>1315</TIMESTAMP>
      </BID>
      <BID>
        <PRICE>5600</PRICE>
        <TIME>2:54:32 PM</TIME>
        <BIDDER>Andrew</BIDDER>
        <TIMESTAMP>1308</TIMESTAMP>
      </BID>
      <BID>
        <PRICE>5500</PRICE>
        <TIME>2:48:08 PM</TIME>
        <BIDDER>Chris</BIDDER>
        <TIMESTAMP>1307</TIMESTAMP>
      </BID>
      <BID>
        <PRICE>5000</PRICE>
        <TIME>2:47:58 PM</TIME>
        <BIDDER>opening price</BIDDER>
        <TIMESTAMP>1298</TIMESTAMP>
      </BID>
    </BIDS>
  </ITEM>
</AUCTIONBLOCK>
```

tabla.xml

barra.xml

arte.xml

Price	Time	Bidder
\$6000	3:02:22 PM	Chris
\$5700	2:58:42 PM	John
\$5600	2:54:32 PM	Andrew
\$5500	2:48:08 PM	Chris
\$5000	2:47:58 PM	opening price

del Software

Vase and Stones by Linda Mann

Chris (3:02 PM)	\$6000
John (2:58 PM)	\$5700
Andrew (2:54 PM)	\$5600
Chris (2:48 PM)	\$5500
\$5000 opening price (2:47 PM)	



**Vase and  
Stones**  
**Linda Mann**

Size: 20x30 inches

Oil, 1996

High bid: \$5700 (John)

Opening bid: \$5000

Copyright © 1997 Linda Mann, all  
rights reserved.  
Linda Mann Art Gallery

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

```
<AUCTIONBLOCK>
```

```
<ITEM>
```

```
<TITLE>Vase and Stones</TITLE>
```

```
<ARTIST>Linda Mann</ARTIST>
```

```
<DIMENSIONS>20x30 inches</DIMENSIONS>
```

```
<MATERIALS>oil</MATERIALS>
```

```
<YEAR>1996</YEAR>
```

```
<DESCRIPTION>Still Life</DESCRIPTION>
```

```
<PREVIEW-SMALL src="burl-s.jpg" width="300"
```

```
height="194" alt="Vase and Stones"/>
```

```
<BIDS>
```

```
<BID>
```

```
<PRICE>6000</PRICE>
```

```
<TIME>3:02:22 PM</TIME>
```

```
<BIDDER>Chris</BIDDER>
```

```
<TIMESTAMP>1307</TIMESTAMP>
```

```
</BID>
```

```
<BID>
```

```
<PRICE>5700</PRICE>
```

```
<TIME>2:58:42 PM</TIME>
```

```
<BIDDER>John</BIDDER>
```

```
<TIMESTAMP>1315</TIMESTAMP>
```

```
</BID>
```

```
<BID>
```

```
<PRICE>5600</PRICE>
```

```
<TIME>2:54:32 PM</TIME>
```

```
<BIDDER>Andrew</BIDDER>
```

```
<TIMESTAMP>1308</TIMESTAMP>
```

```
</BID>
```

```
<BID>
```

```
<PRICE>5500</PRICE>
```

```
<TIME>2:48:08 PM</TIME>
```

```
<BIDDER>Chris</BIDDER>
```

```
<TIMESTAMP>1307</TIMESTAMP>
```

```
</BID>
```

```
<BID>
```

```
<PRICE>5000</PRICE>
```

```
<TIME>2:47:58 PM</TIME>
```

```
<BIDDER>opening price</BIDDER>
```

```
<TIMESTAMP>1298</TIMESTAMP>
```

ie5.xsl

ie4.xsl

nav3.xsl

nokia.xsl

sony.xsl

edi\_x.xsl

sap\_y.xsl

flat\_z.xsl



en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Esquema

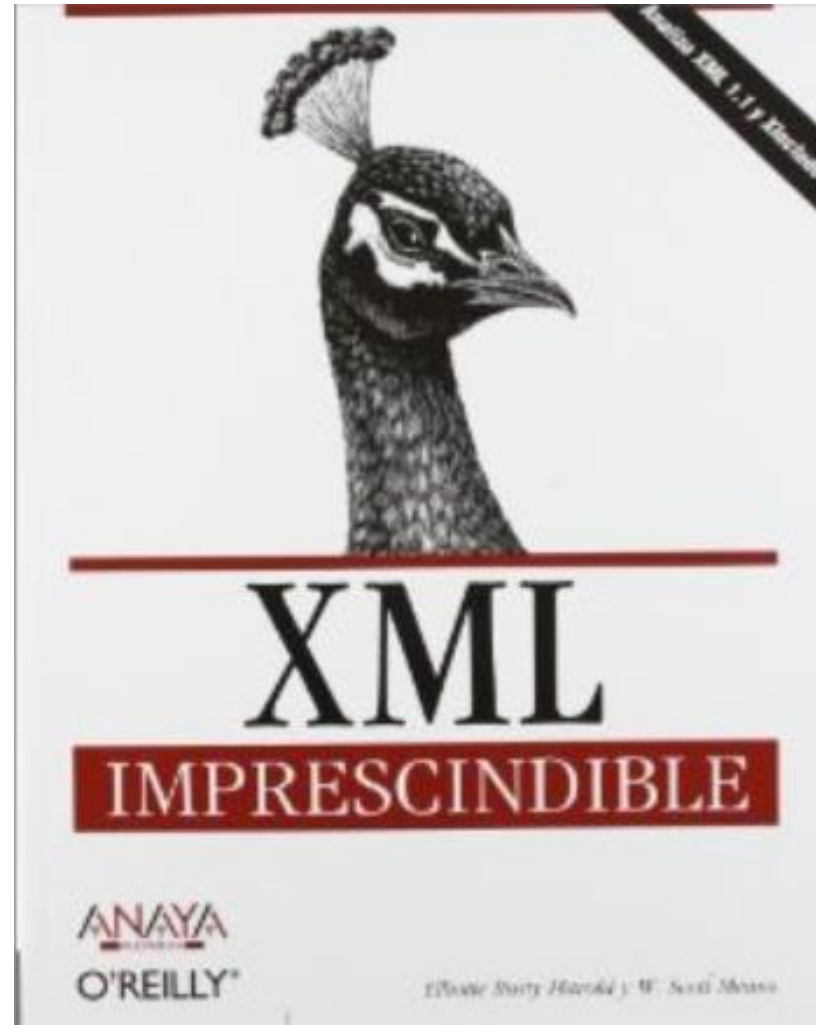
Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- **Bibliografía**
- Referencias Web
- 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



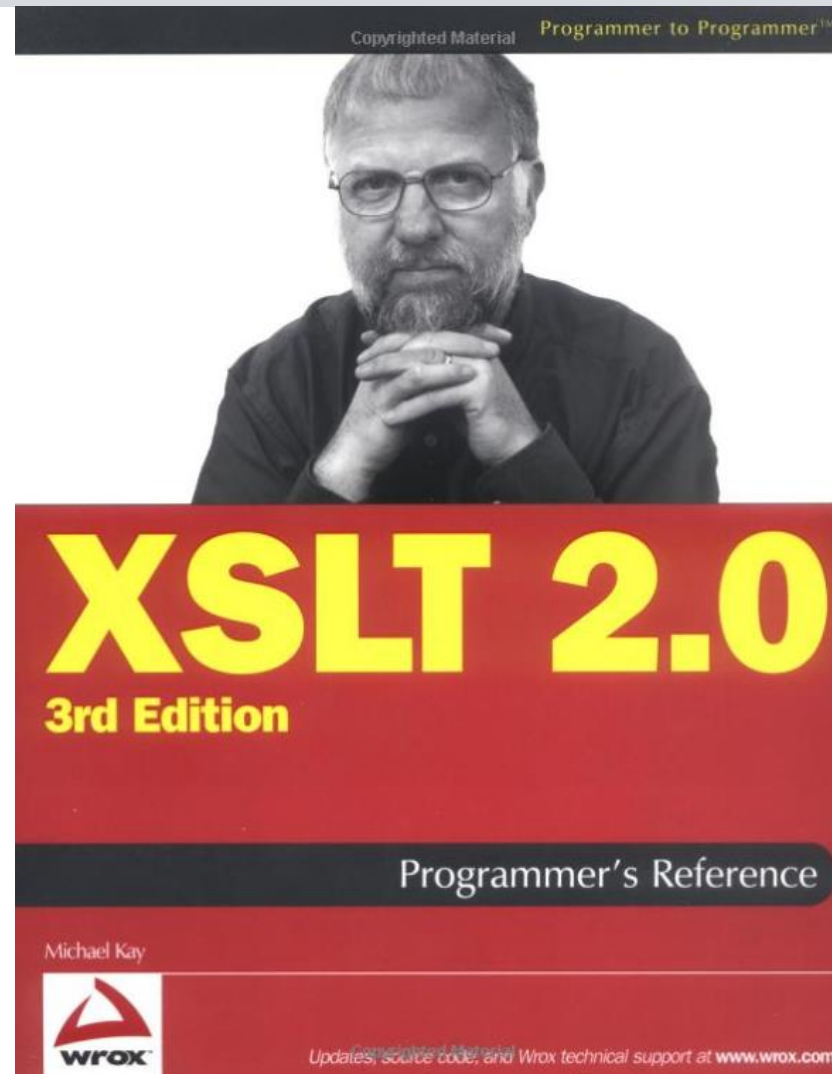
Grado en  
Ingeniería  
Informática  
del Software



# Software y estándares para la Web

## Bibliografía recomendada

- Libro recomendado de lectura y consulta “**XSLT 2.0. Programmer’s Reference**”. 3rd Edition.
- Autor: Michael Kay
- Editorial Wrox (2004)



Grado en  
Ingeniería  
Informática  
del Software

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- **Referencias Web**
- Ejercicios propuestos

# Software y estándares para la Web

## Referencias Web (I)

Grado en  
Ingeniería  
Informática  
del Software

- XSLT en el W3C
  - Panorama general
    - <http://www.w3.org/standards/xml/transformation>
  - Situación actual del XSLT
    - [http://www.w3.org/standards/techs/xslt#w3c\\_all](http://www.w3.org/standards/techs/xslt#w3c_all)
  - Recomendación XSLT 2.0 (23-enero-2007)
    - <http://www.w3.org/TR/xslt20/>
  - Asociando hojas de estilo con documentos XML (28-octubre-2010)
    - <http://www.w3.org/TR/xml-styleSheet/>
  - Recomendación XML (26-Noviembre-2008)
    - <http://www.w3.org/TR/xml/>



# Software y estándares para la Web

## Referencias Web (II)

**Grado en  
Ingeniería  
Informática  
del Software**

- Tutorial on-line
  - <http://www.w3schools.com/xsl/>
- Ejemplos de transformaciones XSLT
  - [http://www.aviation-ia.com/aeec/projects/aoc/XSLT\\_Examples.pdf](http://www.aviation-ia.com/aeec/projects/aoc/XSLT_Examples.pdf)

# Software y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Introducción a XSLT
- Transformaciones XSLT
- Estructura de los documentos XSL
- Xpath
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- **Ejercicios propuestos**

# Software y estándares para la Web

## Ejercicios propuestos (I): Transformación de artículos de revista

Grado en  
Ingeniería  
Informática  
del Software

- Construir un **documento XML bien formado y válido usando un XML Schema** para contener artículos de revistas con los siguientes requisitos mínimos:
  - Título del artículo
  - Autores y su correo electrónico
  - Resumen
  - Palabras clave
  - Nombre de la revista
  - Número o volumen de la revista
  - Página de inicio del artículo
  - Página final del artículo
  - Año
- **Transformar el documento XML** por medio de **XSLT** en un documento HTML5 y CSS usando diferentes navegadores
- Debe comprobarse que los archivos generados cumplan estrictamente los **estándares** correspondientes de HTML5 y CSS
- El formato de presentación es libre, pero debe mostrar **toda** la información del archivo XML

# Software y estándares para la Web

## Ejercicios propuestos (II): Artículos de revista de los últimos 5 años

Grado en  
Ingeniería  
Informática  
del Software

- Usar el **documento XML bien formado y válido usando un XML Schema** que contiene artículos de revistas para generar un informe con los artículos de revista de los últimos 5 años ordenados por fechas
- **Transformar el documento XML** por medio de **XSLT** en un documento HTML5 y CSS usando diferentes navegadores y que genere el informe con los artículos de los últimos 5 años ordenados por fechas
- Debe comprobarse que los archivos generados cumplan estrictamente los **estándares** correspondientes de HTML5 y CSS
- El formato de presentación es libre, pero debe mostrar **toda** la información del archivo XML con los artículos de los últimos 5 años

# Software y estándares para la Web

## Ejercicios propuestos (III): Artículos de revista entre dos fechas

Grado en  
Ingeniería  
Informática  
del Software

- Usar el **documento XML bien formado y válido usando un XML Schema** que contiene artículos de revistas para generar un informe con los artículos de revista entre dos fechas
- El informe debe estar ordenado por fechas de los artículos
- **Transformar el documento XML** por medio de **XSLT** en un documento HTML5 y CSS usando diferentes navegadores y que genere el informe con los artículos entre dos fechas. El informe está ordenado por fechas
- Debe comprobarse que los archivos generados cumplan estrictamente los **estándares** correspondientes de HTML5 y CSS
- El formato de presentación es libre, pero debe mostrar **toda** la información del archivo XML con los artículos entre dos fechas

# Software y estándares para la Web

## Ejercicios propuestos (IV): Artículos de revista de un autor

Grado en  
Ingeniería  
Informática  
del Software

- Usar el **documento XML bien formado y válido usando un XML Schema** que contiene artículos de revistas para generar un informe con los artículos de revista de un determinado autor
- El informe debe estar ordenado por fechas de los artículos
- **Transformar el documento XML** por medio de **XSLT** en un documento HTML5 y CSS usando diferentes navegadores y que genere el informe con los artículos de un autor. El informe está ordenado por fechas
- Debe comprobarse que los archivos generados cumplan estrictamente los **estándares** correspondientes de HTML5 y CSS
- El formato de presentación es libre, pero debe mostrar **toda** la información del archivo XML con los artículos de un autor

# Software y estándares para la Web

## Ejercicios propuestos (V): Transformación de recetas de cocina

Grado en  
Ingeniería  
Informática  
del Software

- Construir un **documento XML bien formado y válido usando un XML Schema** para contener recetas de cocina con los siguientes requisitos mínimos:
  - Nombre de la receta (por ejemplo “Fabada Asturiana”)
  - Tipo de plato (postre, primer plato, entrante, . . .)
  - Ingredientes con cantidades (por ejemplo “Fabes 500 gramos”)
  - Calorías del plato (opcional)
  - Proceso de elaboración, especificado en pasos, por ejemplo:
    - Paso 1: Poner *les fabes* a remojo la noche anterior
    - Paso 2: Poner *les fabes* a cocer con agua y laurel
    - Paso 3: etc...
  - Dificultad del proceso de elaboración (por ejemplo “Fácil”, “Medio”, “Difícil”,...)
  - Tiempo de elaboración (por ejemplo “45 minutos”)
  - Elementos utilizados para la elaboración (microondas, wok, horno, freidora, . . .)
  - Origen de la receta (por ejemplo “Receta de mi abuela”, “Libro de M<sup>a</sup> Luisa”, “Libro de las 1001 recetas”, “www.recetasMUYricas.com”)
- **Transformar el documento XML** por medio de **XSLT** en un documento HTML5 y CSS usando diferentes navegadores
- Debe comprobarse que los archivos generados cumplan estrictamente los **estándares** correspondientes de HTML5 y CSS
- El formato de presentación es libre, pero debe mostrar **toda** la información del archivo XML.

# Software y estándares para la Web

## Ejercicios propuestos (VI): Menú con pollo

Grado en  
Ingeniería  
Informática  
del Software

- Supongamos que solamente tenemos pollo en nuestra nevera
- Se desea obtener todas las recetas que usen pollo a partir del documento XML con las recetas de cocina
- **Transformar el documento XML** por medio de **XSLT** en un documento HTML5 y CSS usando diferentes navegadores para obtener las recetas que usen pollo
- Debe comprobarse que los archivos generados cumplan estrictamente los **estándares** correspondientes de HTML5 y CSS
- El formato de presentación es libre, pero debe mostrar **toda** la información del archivo XML que contenga recetas con pollo



# Software y estándares para la Web

## Ejercicios propuestos (VII): Menú sin leche ni queso

Grado en  
Ingeniería  
Informática  
del Software

- Supongamos que tenemos una persona invitada que es intolerante a la leche y al queso
- Se desea obtener todas las recetas que no usen ni leche ni queso a partir del documento XML con las recetas de cocina
- El menú debe tener al menos dos entrantes, dos primeros platos, dos segundos platos y postre
- **Transformar el documento XML** por medio de **XSLT** en un documento HTML5 y CSS usando diferentes navegadores para obtener las recetas que no usen ni leche ni queso
- Debe comprobarse que los archivos generados cumplan estrictamente los **estándares** correspondientes de HTML5 y CSS
- El formato de presentación es libre, pero debe mostrar **toda** la información del archivo XML que contenga recetas que no usen leche ni queso

# Software y estándares para la Web

## Ejercicios propuestos (VIII): Menú vegano

Grado en  
Ingeniería  
Informática  
del Software

- Supongamos que tenemos una persona vegana invitada
- Se desea generar un menú vegano a partir del documento XML con las recetas de cocina
- El menú vegano debe tener al menos dos entrantes, dos primeros platos, dos segundos platos y postre
- **Transformar el documento XML** por medio de **XSLT** en un documento HTML5 y CSS usando diferentes navegadores para obtener el menú vegano
- Debe comprobarse que los archivos generados cumplan estrictamente los **estándares** correspondientes de HTML5 y CSS
- El formato de presentación es libre, pero debe mostrar **toda** la información del archivo XML que contenga recetas veganas