



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
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

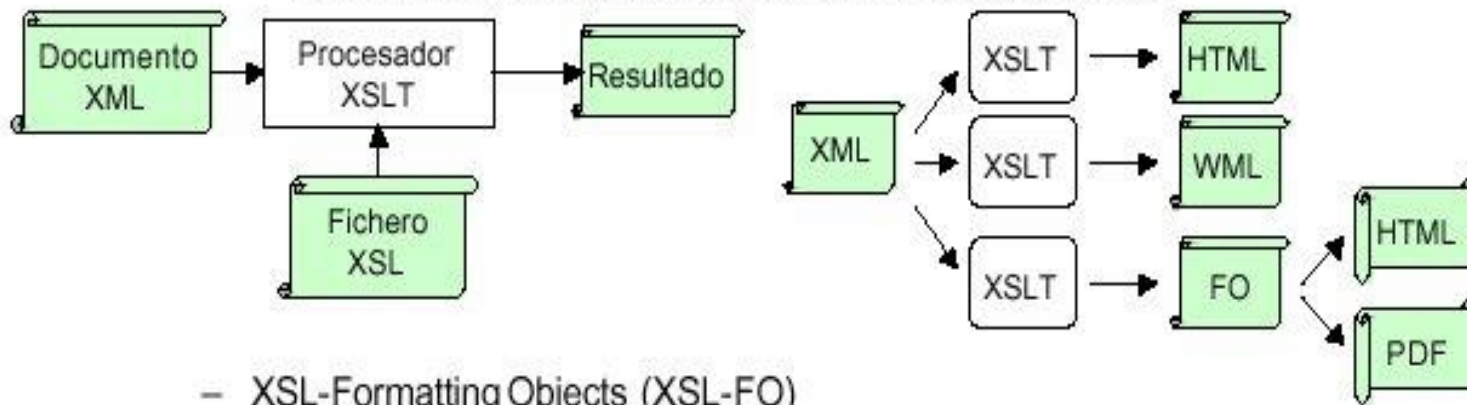
- **Introducción a XSLT**
- Transformaciones XSLT
- Estructura de los documentos XSL
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- Ejercicios propuestos

Software y estándares para la Web

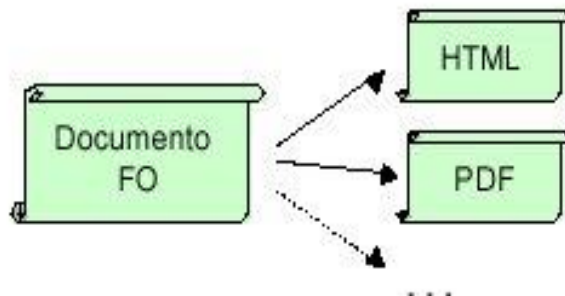
Introducción a XSLT (I): Transformación de documentos XML

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 8-junio-2017
 - Versión 3.0
 - <https://www.w3.org/TR/2017/REC-xslt-30-20170608/>

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

Introducción a XSLT (VI): **AVISO IMPORTANTE**

Grado en
Ingeniería
Informática
del Software

- Las principales herramientas de transformación eran los **navegadores**
- Los navegadores abrían los archivos XML y aplicaban el archivo XSL asociado
- Por **problemas de seguridad** los navegadores han dejado de soportar las transformaciones XSL de los archivos XML
- Actualmente solamente navegadores antiguos y sin soporte de seguridad admiten el uso de transformaciones XSL sobre archivos XML
 - Ejemplo: **Internet Explorer**
- Las herramientas de transformación pueden estar incluidas en los “plugins” de XML de los editores de texto
 - Ejemplo: **NotePad++**

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

- Introducción a XSLT
- **Transformaciones XSLT**
- Estructura de los documentos XSL
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- 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:
 - HTML
 - Textos delimitado por comas
 - PDF
 - 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?

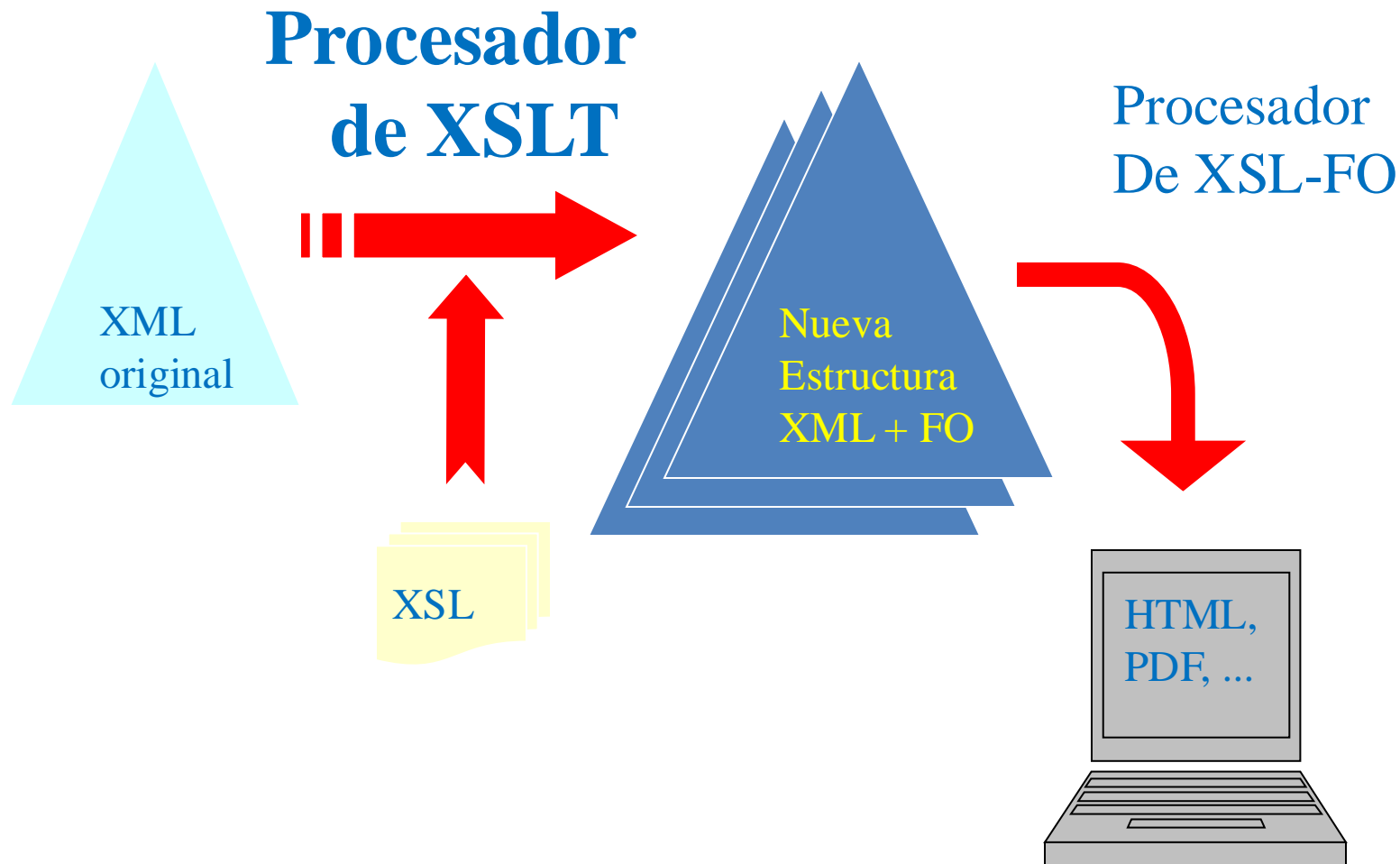
Grado en
Ingeniería
Informática
del Software

- 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>
```

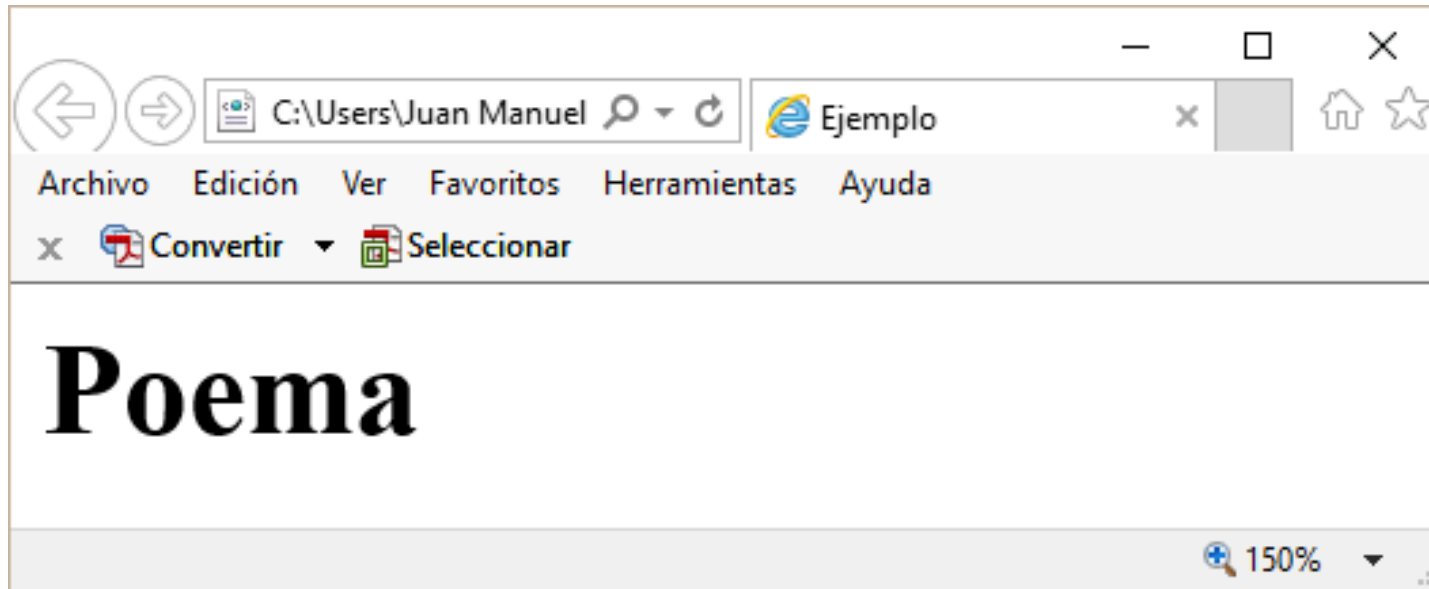
Poema.xsl

```
<?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

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">
    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>
```

Software y estándares para la Web

Transformaciones XSLT (IX): Obtener valores de atributos

Grado en
Ingeniería
Informática
del Software

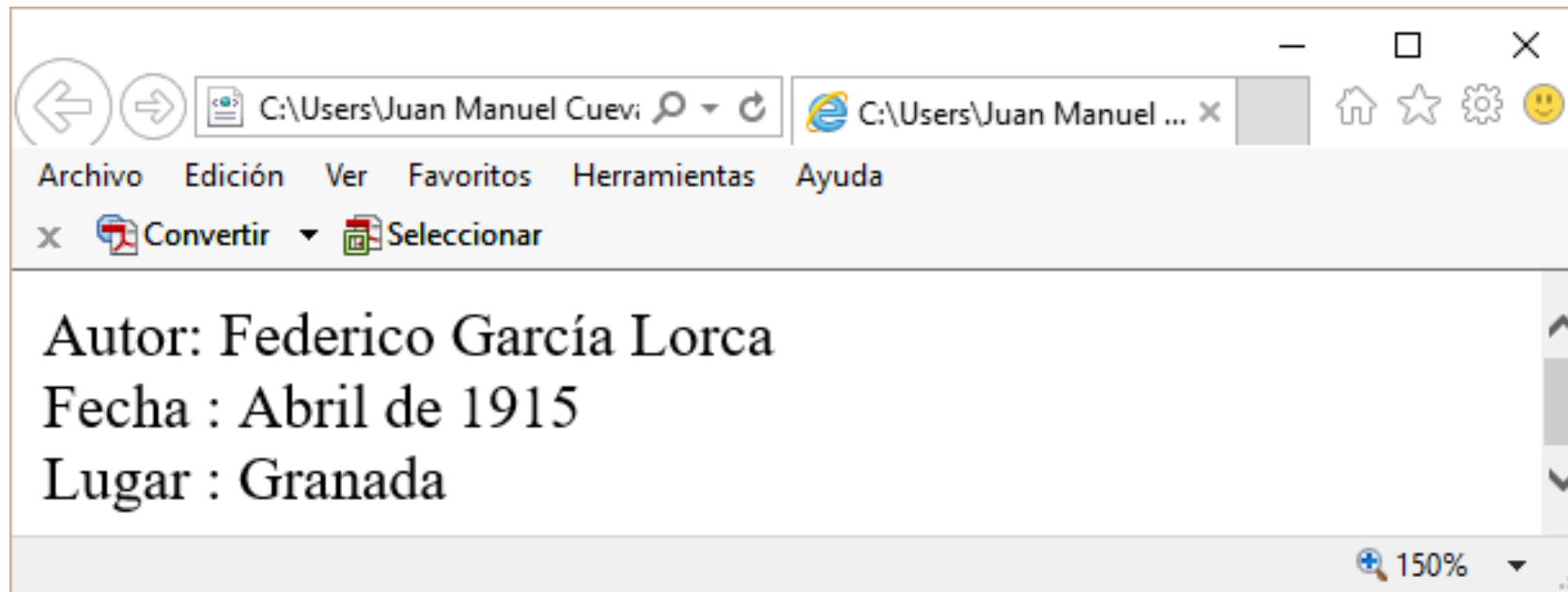
```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="poemaAutorLugarFecha.xsl"?>  
<poema autor="Federico García Lorca" fecha="Abril de 1915"  
lugar="Granada">  
<titulo>Alba</titulo>  
<verso>Mi corazón oprimido</verso>  
<verso>siente junto a la alborada</verso>  
<verso>el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

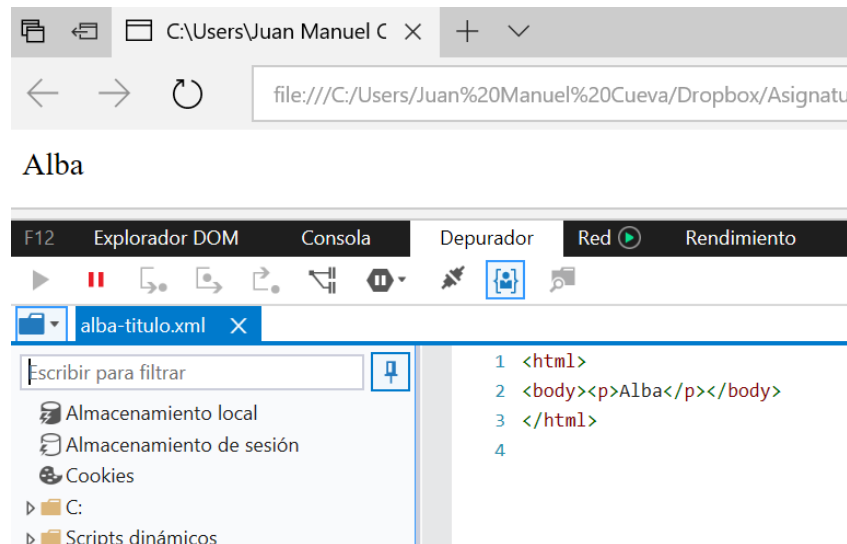
Software y estándares para la Web

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

poemaTitulo.xsl

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



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>
```

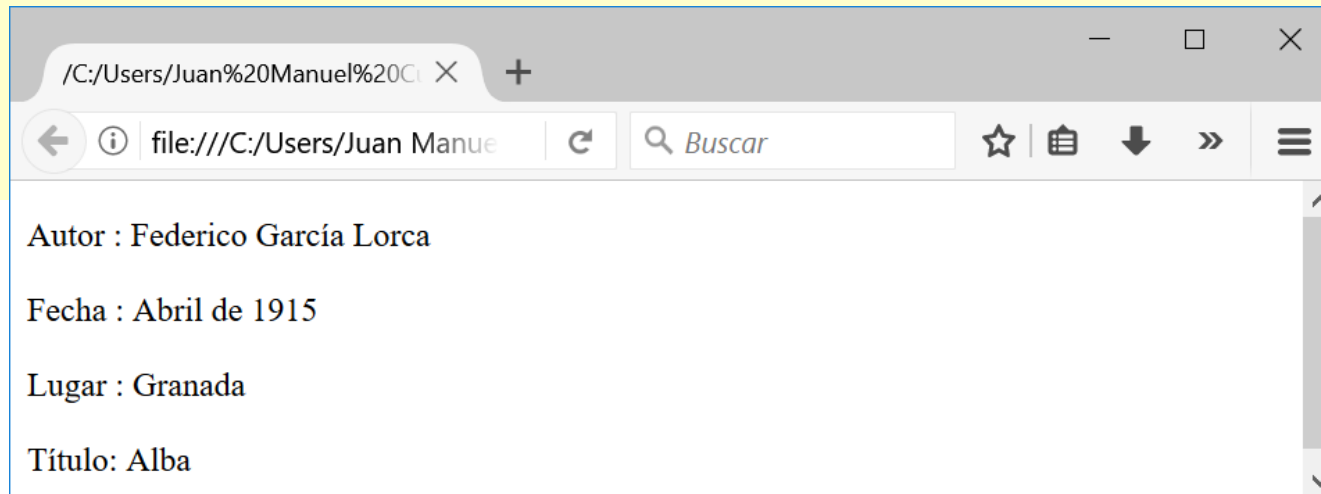
alba-titulo-fecha-lugar.xml

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.xsl

poemaCompleto.xsl

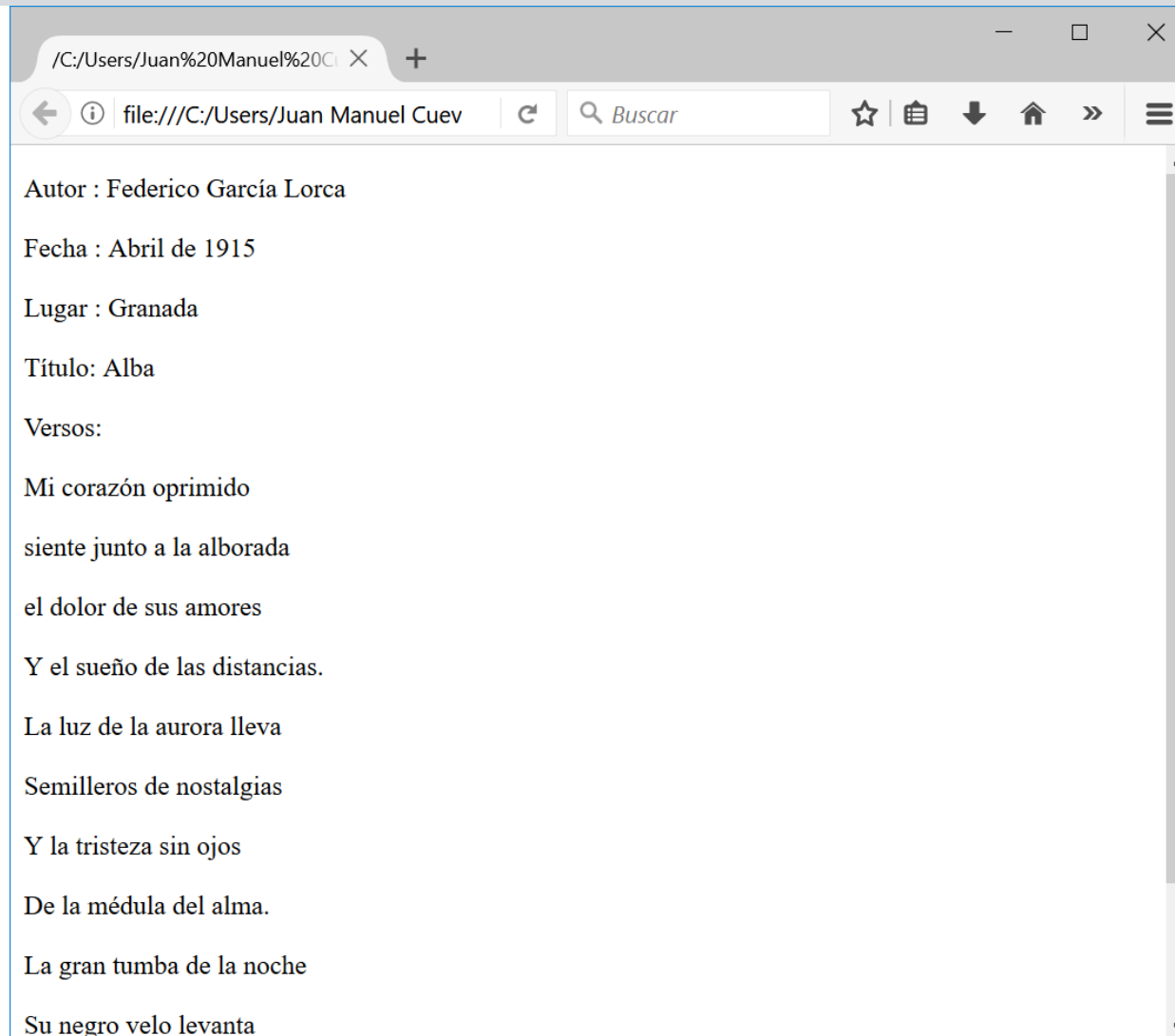
Grado en
Ingeniería
Informática
del Software

```
<?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**
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- 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, 2.0, 3.0
 - **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

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

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

- **<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>
```

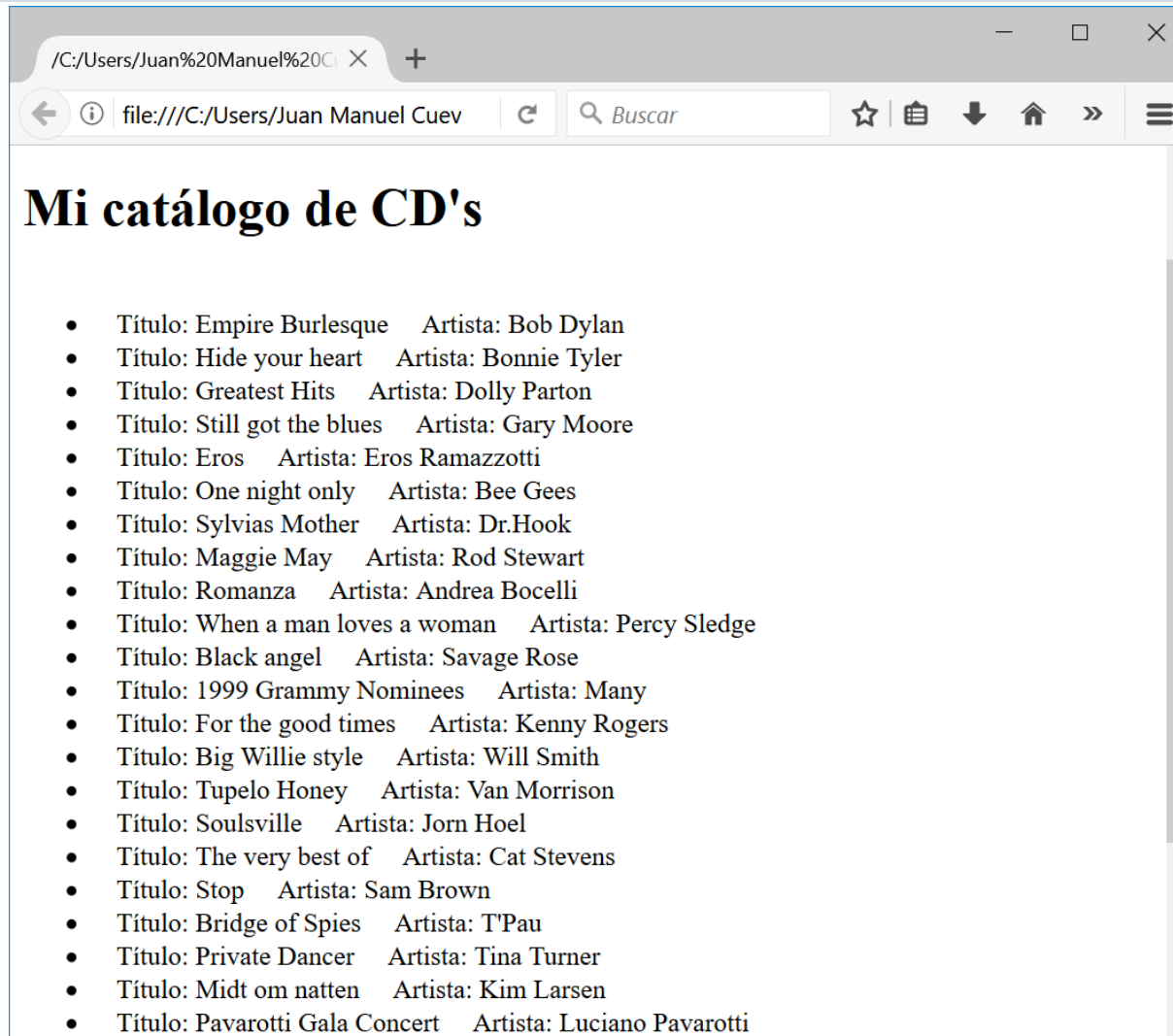
```
<xsl:template match="title">  
  Título: <xsl:value-of select="."/>  
</xsl:template>
```

```
<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 field with 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, there 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

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

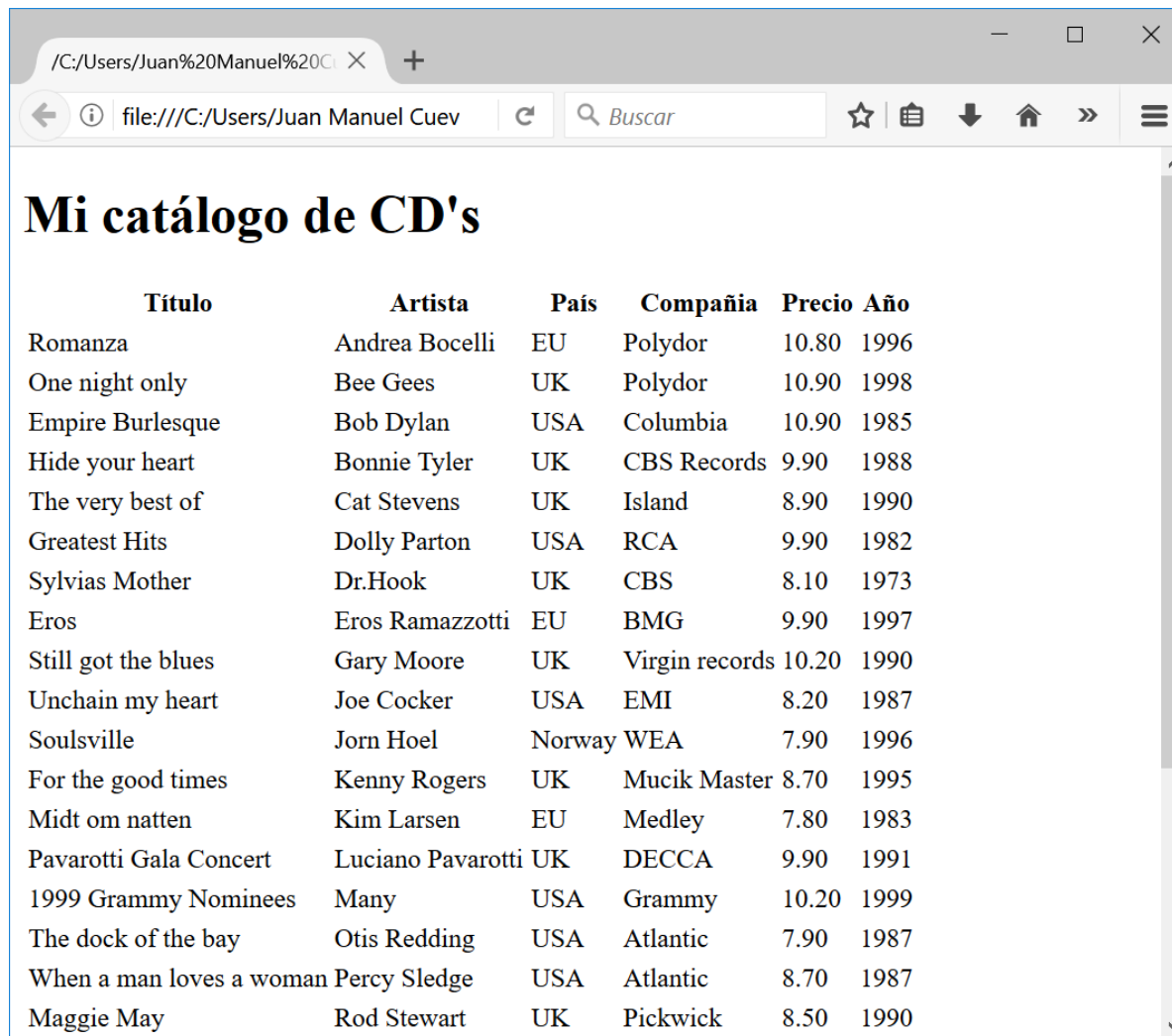
Grado en
Ingeniería
Informática
del Software

```
<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: 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
- **Sentencias iterativas**
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- 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**

```
<?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)

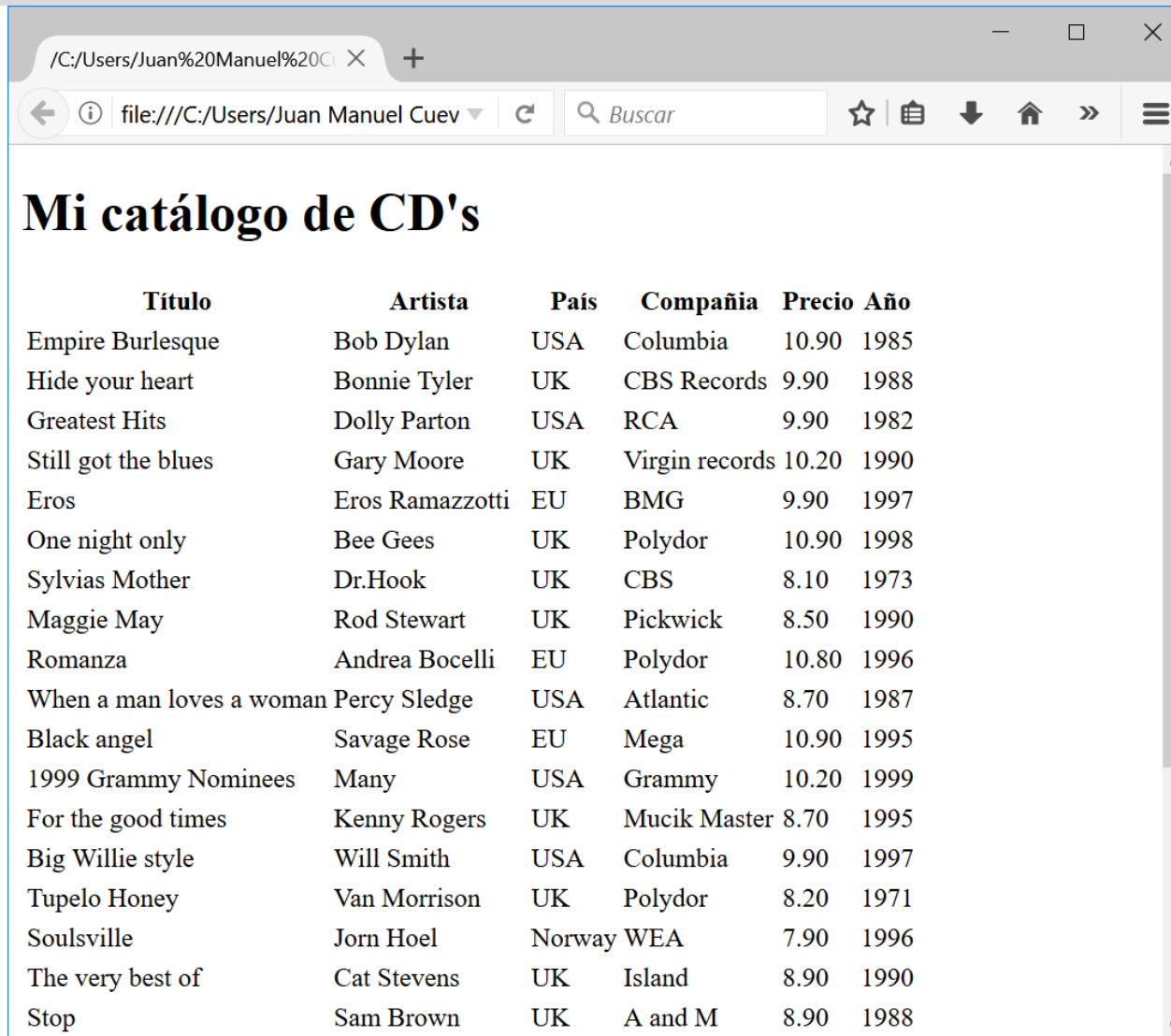
Grado en
Ingeniería
Informática
del Software

```
<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
- Sentencias iterativas
- **Sentencias selectivas**
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- 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

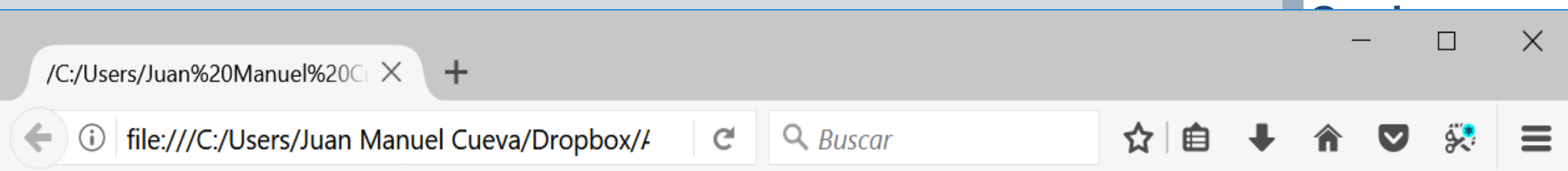
Sentencias selectivas (IV): Ejemplo **musica-if.xsl** (continuación)

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 corresponding artists. Some artists are marked with three asterisks (***) to indicate a price above 10.

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
- Sentencias iterativas
- Sentencias selectivas
- **Construcción de elementos en el árbol resultante**
- Variables y parámetros
- Conclusiones
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- 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`

Grado en
Ingeniería
Informática
del Software

```
<?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**

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

Grado en
Ingeniería
Informática
del Software

Software y estándares para la Web

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

The image shows a web browser window with the address bar displaying the file path: `file:///C:/Users/Juan%20Manuel%20Cueva/Dropbox/Asignaturas/UNIOVI/SoftwareYestandaresWeb/XML/04-Ejemplos/elementos2atributos.xml`. Below the browser, the developer tools are open, showing the XML content in the 'elemento...utos.xml' file:

```
1 <?xml version="1.0" encoding="UTF-16"?>
2   <elemento verso="verso1" />
3   <elemento verso="verso2" />
4   <elemento verso="verso3" />
5   <elemento verso="verso4" />
6   <elemento verso="verso5" />
7
```

The developer tools interface includes a top bar with tabs for F12, Explorador DOM, Consola (1 error), Depurador, Red, Rendimiento, Memoria, Emulación, and Experimentos. The right sidebar contains the 'Inspecciones' panel with a search bar and a button to 'Agregar inspección', and the 'Pila de llamadas' panel with a button to 'Puntos de interrupción'.

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

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

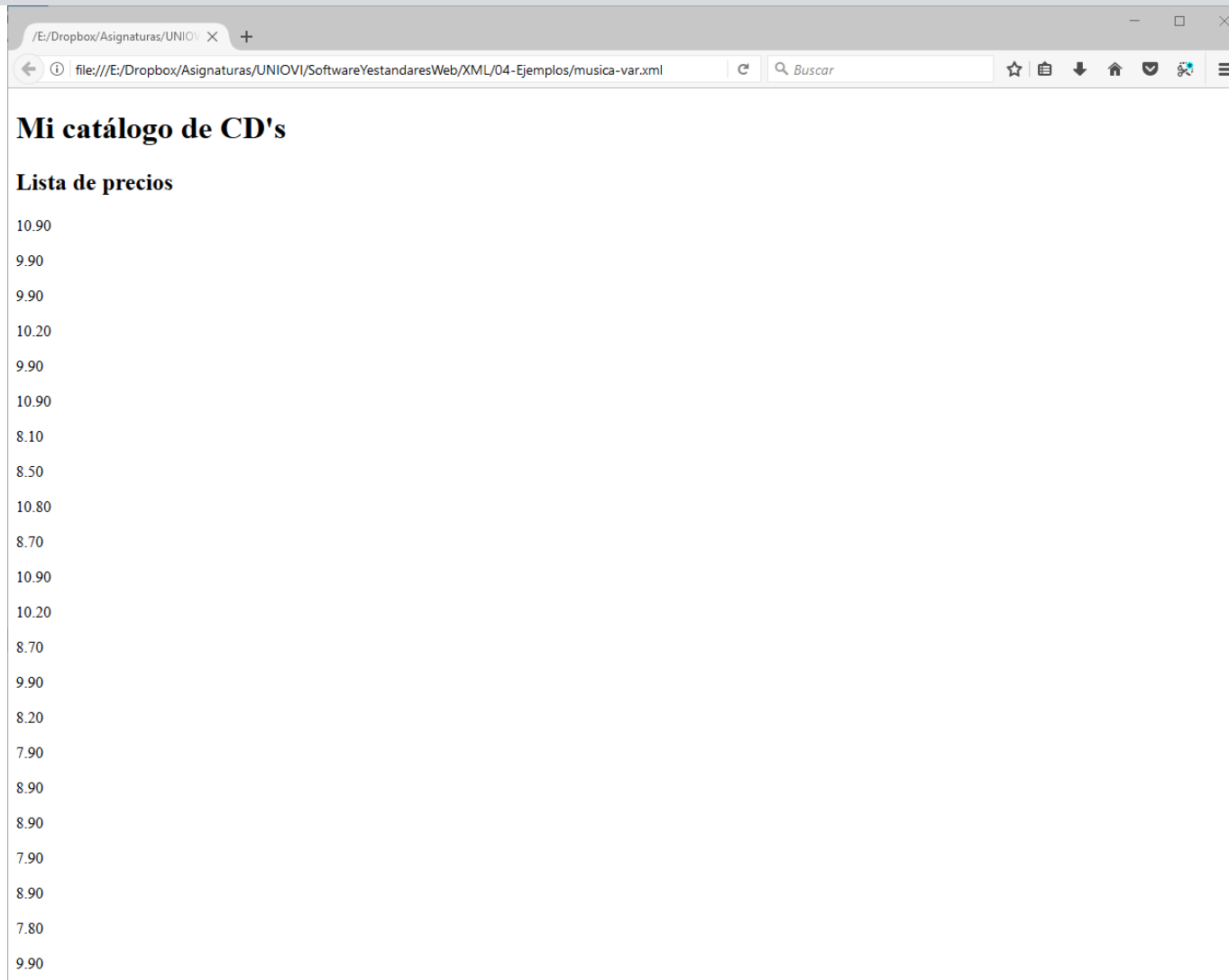
Variables y parámetros (V): Ejemplo musica-var.xsl

```
<?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
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- **Conclusiones**
- Bibliografía
- Referencias Web
- Ejercicios resueltos
- 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 8-junio-2017
 - Versión 3.0
 - <https://www.w3.org/TR/xslt-30/>

```
<?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>

```

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

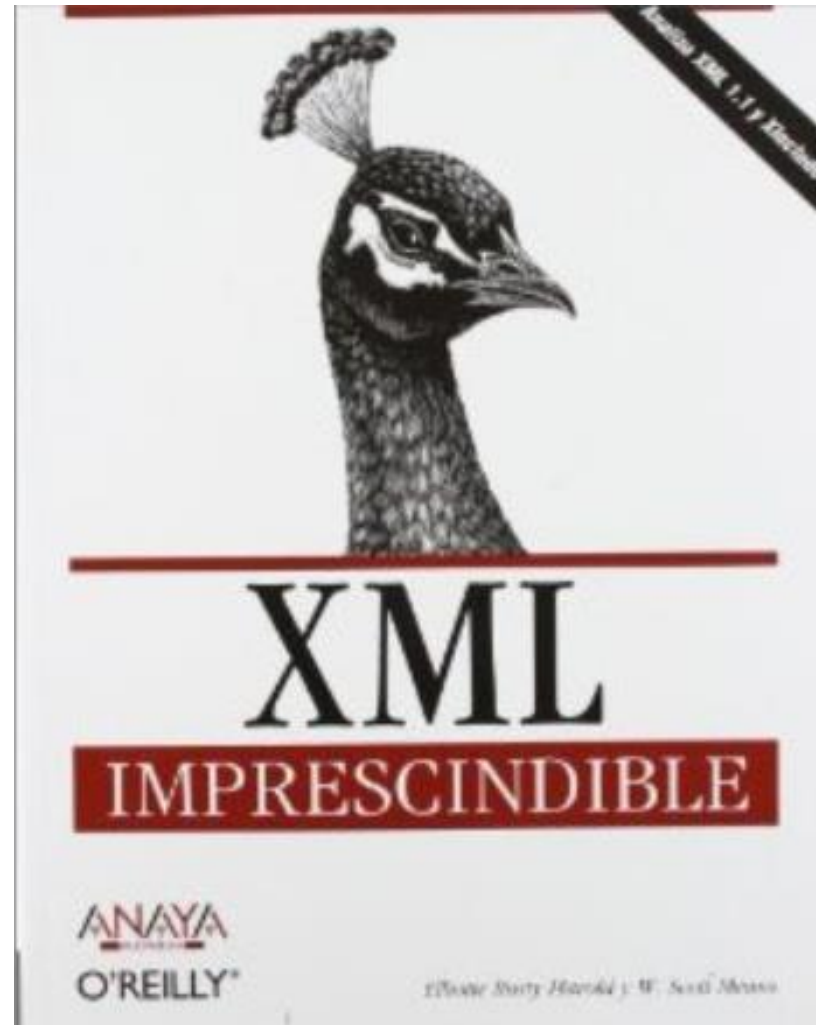
Grado en
Ingeniería
Informática
del Software

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

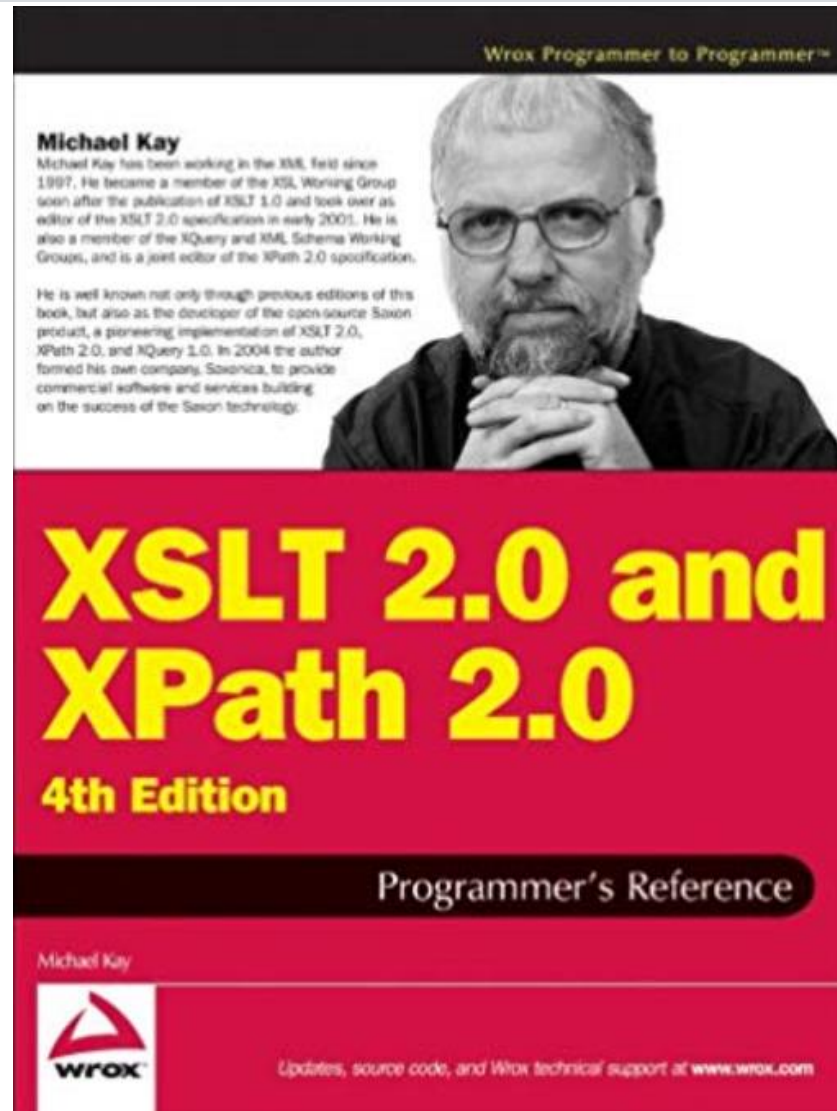


Grado en
Ingeniería
Informática
del Software

Software y estándares para la Web

Bibliografía recomendada

- Libro recomendado de lectura y consulta “**XSLT 2.0 and XPath 2.0. Programmer’s Reference**”. 4th Edition.
- Autor: Michael Kay
- Editorial Wrox (2008)



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
- Sentencias iterativas
- Sentencias selectivas
- Construcción de elementos en el árbol resultante
- Variables y parámetros
- Conclusiones
- Bibliografía
- **Referencias Web**
- Ejercicios resueltos
- 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>
- Introduction to XSLT 3.0
 - <http://www.stylusstudio.com/tutorials/intro-xslt-3.html>
- Mapping Farmers' Markets in Google Earth using XSLT
 - <http://www.stylusstudio.com/tutorials/mapping-google-earth-in-xslt.html>
- Questions tagged [xslt-3.0]
 - <https://stackoverflow.com/questions/tagged/xslt-3.0>

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

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

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

Software y estándares para la Web

Ejercicios resueltos (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 resueltos (I): Solución – artículos.xsl (a)

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="5.0" encoding="utf-8" indent="yes"/>

<xsl:template match="articles">
<xsl:text disable-output-escaping='yes'>&lt;!DOCTYPE html></xsl:text>
<html lang="es">
<head>
<meta name="description" content="Ejercicio resuelto 1"/>
<title>Ejercicio 1</title>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
<header>
<h1>EJERCICIO 1</h1>
<h2>Transformación de artículos de revista</h2>
</header>
<main>
<xsl:for-each select="article">
<section class="article">
<h3>
<xsl:value-of select="title"/>
</h3>
```

Software y estándares para la Web

Ejercicios resueltos (I): Solución – artículos.xsl (b)

Grado en
Ingeniería
Informática
del Software

```
<p>
<span class="section">Año: </span>
<xsl:value-of select="@year"/>
</p>
<p>
<span class="section">Revista: </span>
<xsl:value-of select="journal"/>
</p>
<p>
<span class="section">Volumen: </span>
<xsl:value-of select="volume"/>
</p>
<xsl:if test="count(keywords//keyword) > 0">
<p>
<span class="section">Palabras clave: </span>
<xsl:for-each select="keywords//keyword">
<xsl:value-of select="."/>
<xsl:if test="not(position()=last())">
<xsl:text>, </xsl:text>
</xsl:if>
</xsl:for-each>
</p>
</xsl:if>
```

Software y estándares para la Web

Ejercicios resueltos (I): Solución – artículos.xsl (c)

Grado en
Ingeniería
Informática
del Software

```
<p>
<span class="section">Páginas del artículo: </span>
<xsl:value-of select="@start-page"/>
<xsl:text>-</xsl:text>
<xsl:value-of select="@end-page"/>
</p>
<p class="authors-section">
<span class="section">Autores: </span>
</p>
<ul class="authors">
<xsl:for-each select="authors//author">
<li>
<xsl:value-of select="."/>
<span class="email">
<xsl:text> (</xsl:text>
<xsl:value-of select="@email"/>
<xsl:text>) </xsl:text>
</span>
</li>
</xsl:for-each>
</ul>
<p class="summary-section">
<span class="section">Resumen: </span>
</p>
```

Software y estándares para la Web

Ejercicios resueltos (I): Solución – artículos.xsl (d)

```
<p class="summary">
<xsl:value-of select="summary"/>
</p>
</section>
</xsl:for-each>
<footer>
<div>
<p>Máster en Ingeniería Web</p><p>Universidad de Oviedo</p>
</div>
<div class="validations">
<a href="https://validator.w3.org/check?uri=referer">

</a>
<a href=" http://jigsaw.w3.org/css-validator/check/referer " >

</a>
<a href="http://www.w3.org/WAI/WCAG2AAA-Conformance">

</a>
</div>
</footer>
</main>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Software y estándares para la Web

Ejercicios resueltos (I): Solución – style.css (a)

Grado en
Ingeniería
Informática
del Software

```
body {  
background: #E8CBC0;  
background: linear-gradient(to right, #636FA4, #E8CBC0);  
color: #FFF;  
font-family: Calibri, Arial, sans-serif;  
margin: 0;  
padding: 0;  
}
```

```
body header {  
background: rgba(0, 0, 0, .4);  
color: #FFF;  
margin: 0;  
padding: 1em;  
text-align: center;  
}
```

```
header h2 {  
margin: 0;  
}
```

```
footer {  
background: rgba(0, 0, 0, .4);  
color: #FFF;  
padding: 1em;  
text-align: center;  
}
```

Software y estándares para la Web

Ejercicios resueltos (I): Solución – style.css (b)

Grado en
Ingeniería
Informática
del Software

```
footer>div p {  
margin: 0;  
padding: 0;  
}  
  
.article {  
box-shadow: 5px 5px 5px #888;  
padding: 1.5em;  
margin: 1em 15em 1em 15em;  
background-color: #FFF;  
color: #000;  
}  
  
.article>h3 {  
border-bottom: 1px solid #000;  
font-size: 1.5em;  
margin-top: 0;  
text-align: center;  
}  
  
.section {  
font-weight: bold;  
}
```

Software y estándares para la Web

Ejercicios resueltos (I): Solución – style.css (c)

Grado en
Ingeniería
Informática
del Software

```
.authors {  
margin-top: 0;  
padding-left: 2em;  
}  
  
.authors-section {  
margin-bottom: 0;  
}  
  
.summary-section {  
margin-bottom: 0;  
}  
  
.summary {  
margin: 0;  
padding: 0 0 0 1em;  
}  
  
.email {  
font-style: italic;  
}
```

Software y estándares para la Web

Ejercicios resueltos (I): Solución – style.css (d)

Grado en
Ingeniería
Informática
del Software

```
.validations {  
display: inline-flex;  
padding: 1em;  
text-align: center;  
}  
  
.validations a {  
color: #FFF;  
background-color: rgba(0, 0, 0, 0);  
}
```


Software y estándares para la Web

Ejercicios resueltos (I): Solución – generado.html (fragmento)

Grado en
Ingeniería
Informática
del Software

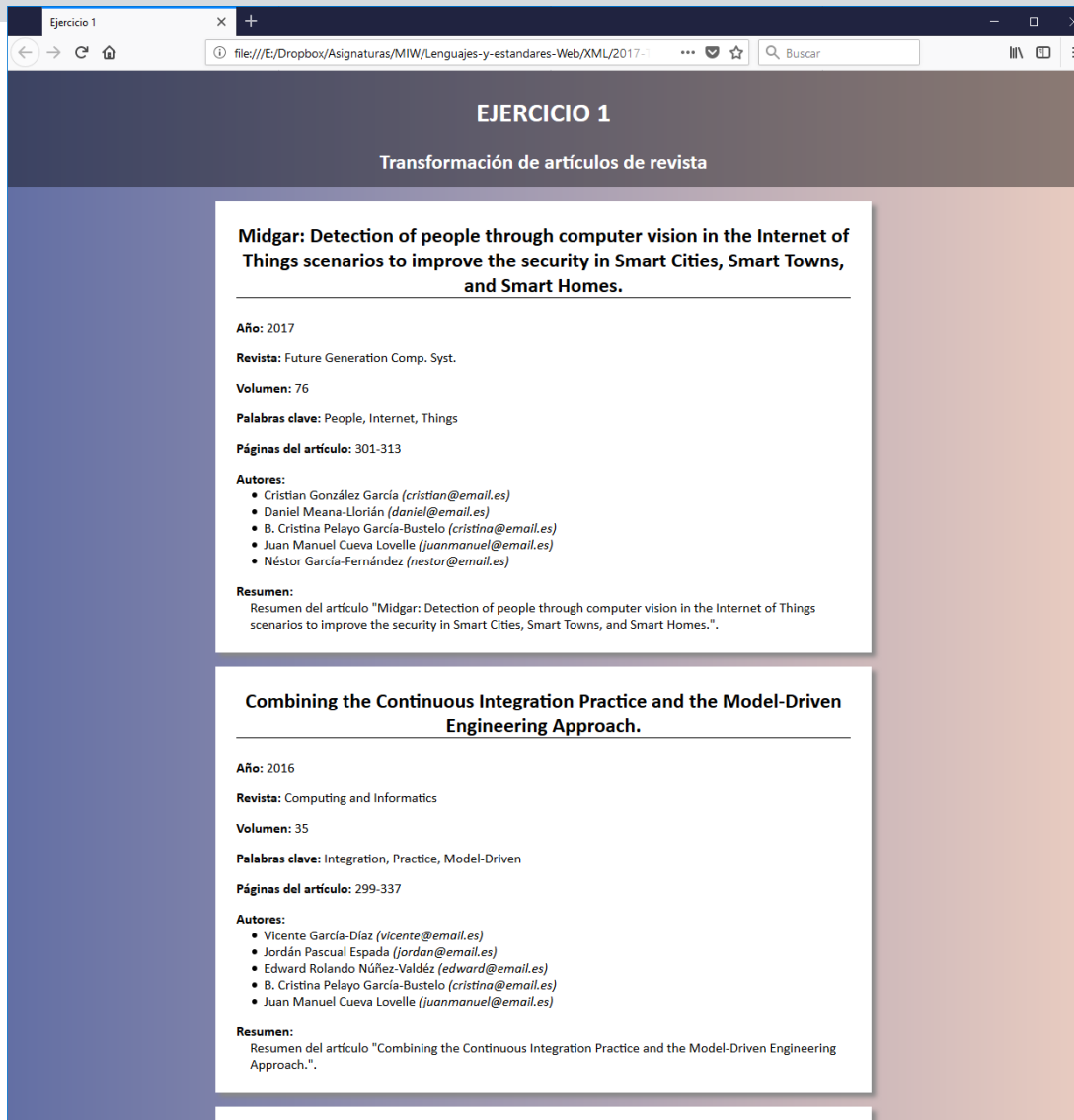
```
<!DOCTYPE html>
<html lang="es">
<head>
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="description" content="Ejercicio resuelto 1">
<title>Ejercicio 1</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body><header><h1>EJERCICIO 1</h1>
<h2>Transformación de artículos de revista</h2></header><main><section class="article"><h3>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.</h3>
<p><span class="section">Año: </span>2017</p>
<p><span class="section">Revista: </span>Future Generation Comp. Syst.</p>
<p><span class="section">Volumen: </span>76</p>
<p><span class="section">Palabras clave: </span>People, Internet, Things</p>
<p><span class="section">Páginas del artículo: </span>301-313</p>
<p class="authors-section"><span class="section">Autores: </span></p>
<ul class="authors">
<li>Cristian González García<span class="email"> (cristian@email.es) </span></li>
<li>Daniel Meana-Llorián<span class="email"> (daniel@email.es) </span></li>
<li>B. Cristina Pelayo García-Bustelo<span class="email"> (cristina@email.es) </span></li>
<li>Juan Manuel Cueva Lovelle<span class="email"> (juanmanuel@email.es) </span></li>
<li>Néstor García-Fernández<span class="email"> (nestor@email.es) </span></li>
</ul>
```

. . .

Software y estándares para la Web

Ejercicios resueltos (I): Solución – Aspecto de la web generada

Grado en
Ingeniería
Informática
del Software



The screenshot shows a web browser window with the address bar displaying a local file path. The page content is titled 'EJERCICIO 1' and 'Transformación de artículos de revista'. It contains two article summaries, each with a title, year, journal, volume, keywords, page numbers, authors, and a brief summary.

EJERCICIO 1
Transformación de artículos de revista

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.

Año: 2017
Revista: Future Generation Comp. Syst.
Volumen: 76
Palabras clave: People, Internet, Things
Páginas del artículo: 301-313

Autores:

- Cristian González García (*cristian@email.es*)
- Daniel Meana-Llorián (*daniel@email.es*)
- B. Cristina Pelayo García-Bustelo (*cristina@email.es*)
- Juan Manuel Cueva Lovelle (*juanmanuel@email.es*)
- Néstor García-Fernández (*nestor@email.es*)

Resumen:
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."

Combining the Continuous Integration Practice and the Model-Driven Engineering Approach.

Año: 2016
Revista: Computing and Informatics
Volumen: 35
Palabras clave: Integration, Practice, Model-Driven
Páginas del artículo: 299-337

Autores:

- Vicente García-Díaz (*vicente@email.es*)
- Jordán Pascual Espada (*jordan@email.es*)
- Edward Rolando Núñez-Valdéz (*edward@email.es*)
- B. Cristina Pelayo García-Bustelo (*cristina@email.es*)
- Juan Manuel Cueva Lovelle (*juanmanuel@email.es*)

Resumen:
Resumen del artículo "Combining the Continuous Integration Practice and the Model-Driven Engineering Approach."

Software y estándares para la Web

Ejercicios resueltos (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 resueltos (II): Solución – articulos2.xsl (a)

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="5.0" encoding="utf-8" indent="yes"/>
<xsl:variable name="years" select="5"/>

<xsl:template match="articles">
<xsl:text disable-output-escaping='yes'>&lt;!DOCTYPE html></xsl:text>
<html lang="es">
<head>
<meta name="description" content="Ejercicio resuelto 2"/>
<title>Ejercicio 2</title>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
<header>
<h1>EJERCICIO 2</h1>
<h2>Artículos de revista de los últimos <xsl:value-of select="$years"/> años</h2>
</header>
<main>
<xsl:for-each select="article">
<xsl:if test="2017-@year &lt; $years">
<section class="article">
<h3>
<xsl:value-of select="title"/>
</h3>
```

Software y estándares para la Web

Ejercicios resueltos (II): Solución – articulos2.xsl (b)

Grado en
Ingeniería
Informática
del Software

```
<p>
<span class="section">Año: </span>
<xsl:value-of select="@year"/>
</p>
<p>
<span class="section">Revista: </span>
<xsl:value-of select="journal"/>
</p>
<p>
<span class="section">Volumen: </span>
<xsl:value-of select="volume"/>
</p>
<xsl:if test="count(keywords//keyword) > 0">
<p>
<span class="section">Palabras clave: </span>
<xsl:for-each select="keywords//keyword">
<xsl:value-of select="."/>
<xsl:if test="not(position()=last())">
<xsl:text>, </xsl:text>
</xsl:if>
</xsl:for-each>
</p>
</xsl:if>
```

Software y estándares para la Web

Ejercicios resueltos (II): Solución – articulos2.xsl (c)

Grado en
Ingeniería
Informática
del Software

```
<p>
<span class="section">Páginas del artículo: </span>
<xsl:value-of select="@start-page"/>
<xsl:text>-</xsl:text>
<xsl:value-of select="@end-page"/>
</p>
<p class="authors-section">
<span class="section">Autores: </span>
</p>
<ul class="authors">
<xsl:for-each select="authors//author">
<li>
<xsl:value-of select="."/>
<span class="email">
<xsl:text> (</xsl:text>
<xsl:value-of select="@email"/>
<xsl:text>) </xsl:text>
</span>
</li>
</xsl:for-each>
</ul>
<p class="summary-section">
<span class="section">Resumen: </span>
</p>
```

Software y estándares para la Web

Ejercicios resueltos (II): Solución – articulos2.xsl (d)

```
<p class="summary">
<xsl:value-of select="summary"/>
</p>
</section>
</xsl:if>
</xsl:for-each>
<footer>
<div>
<p>Máster en Ingeniería Web</p>
<p>Universidad de Oviedo</p>
</div>
<div class="validations">
<a href="https://validator.w3.org/check?uri=referer">

</a>
<a href=" http://jigsaw.w3.org/css-validator/check/referer " >

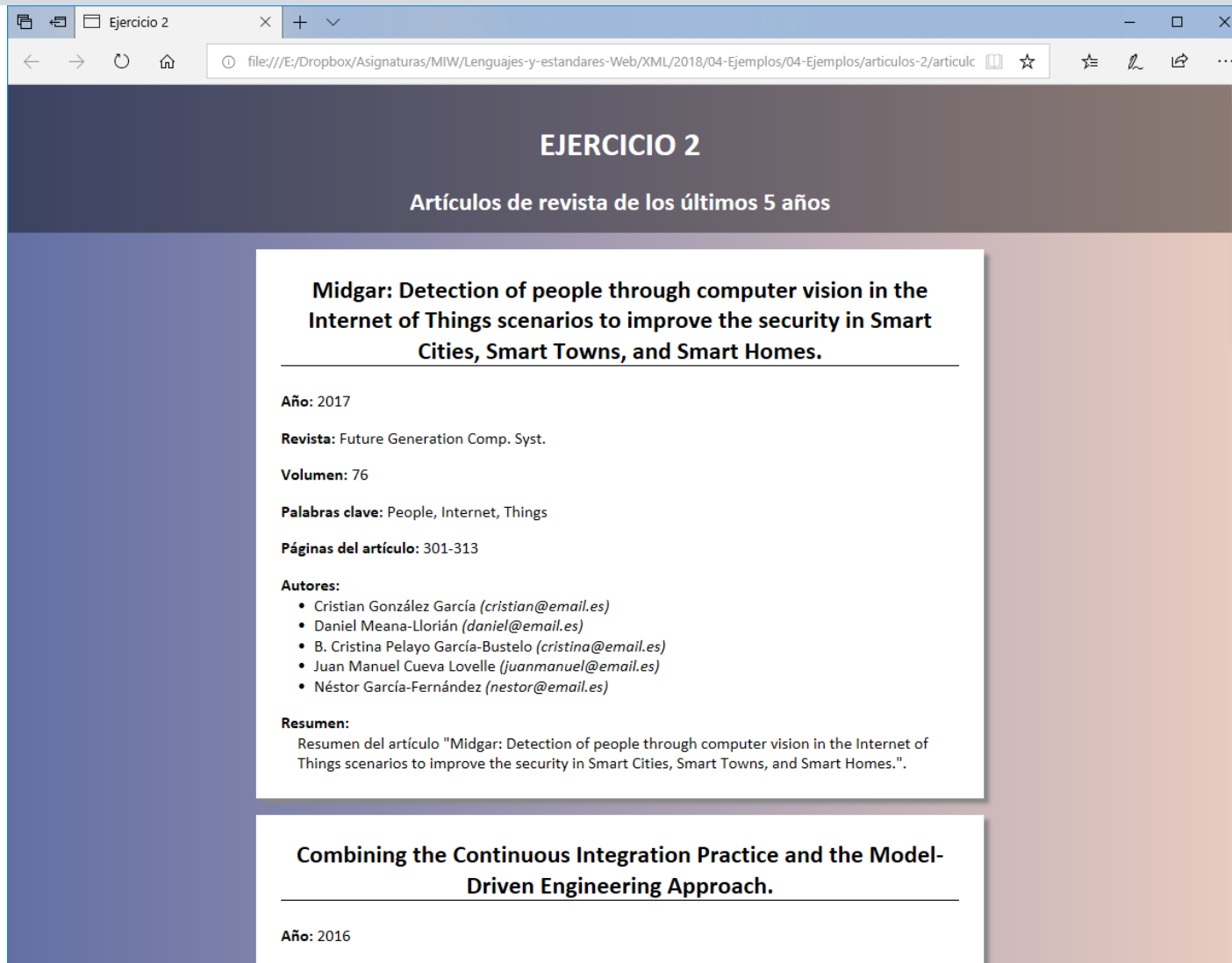
</a>
<a href="http://www.w3.org/WAI/WCAG2AAA-Conformance">

</a>
</div>
</footer>
</main>
</body>
</html>
</xsl:template></xsl:stylesheet>
```

Software y estándares para la Web

Ejercicios resueltos (II): Solución – Aspecto del sitio Web generado

Grado en
Ingeniería
Informática
del Software



Software y estándares para la Web

Ejercicios resueltos (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 resueltos (III): Solución – articulos3.xsl (a)

```
<?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="5.0" encoding="utf-8" indent="yes"/>
<xsl:variable name="since-year" select="2010"/>
<xsl:variable name="until-year" select="2015"/>

<xsl:template match="articles">
<xsl:text disable-output-escaping='yes'>&lt;!DOCTYPE html></xsl:text>
<html lang="es">
<head>
<meta name="description" content="Ejercicio resuelto 3"/>
<title>Ejercicio 3</title>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
<header>
<h1>EJERCICIO 3</h1>
<h2>Artículos de revista entre 2 fechas (<xsl:value-of select="$since-year"/> - <xsl:value-of select="$until-year"/>) y ordenados
</h2>
</header>
<main>
<xsl:for-each select="article">
<xsl:sort order="descending" select="@year"/>
<xsl:if test="$since-year &lt; @year and @year &lt;= $until-year">
<section class="article">
<h3>
<xsl:value-of select="title"/>
</h3>
<p>
<span class="section">Año: </span>
<xsl:value-of select="@year"/>
</p>
</xsl:if>
</xsl:for-each>
</main>
</body>
</html>
```

Software y estándares para la Web

Ejercicios resueltos (III): Solución – articulos3.xsl (b)

```
<p>
<span class="section">Revista: </span>
<xsl:value-of select="journal"/>
</p>
<p>
<span class="section">Volumen: </span>
<xsl:value-of select="volume"/>
</p>
<xsl:if test="count(keywords//keyword) > 0">
<p>
<span class="section">Palabras clave: </span>
<xsl:for-each select="keywords//keyword">
<xsl:value-of select="."/>
<xsl:if test="not(position()=last())">
<xsl:text>, </xsl:text>
</xsl:if>
</xsl:for-each>
</p>
</xsl:if>
<p>
<span class="section">Páginas del artículo: </span>
<xsl:value-of select="@start-page"/>
<xsl:text>-</xsl:text>
<xsl:value-of select="@end-page"/>
</p>
<p class="authors-section">
<span class="section">Autores: </span>
</p>
```

Software y estándares para la Web

Ejercicios resueltos (III): Solución – articulos3.xsl (c)

```
<ul class="authors">
<xsl:for-each select="authors//author">
<li>
<xsl:value-of select="."/>
<span class="email">
<xsl:text> (</xsl:text>
<xsl:value-of select="@email"/>
<xsl:text>) </xsl:text>
</span>
</li>
</xsl:for-each>
</ul>
<p class="summary-section">
<span class="section">Resumen: </span>
</p>
<p class="summary">
<xsl:value-of select="summary"/>
</p>
</section>
</xsl:if>
</xsl:for-each>
```

Software y estándares para la Web

Ejercicios resueltos (III): Solución – articulos3.xsl (d)

```
<footer>
<div>
<p>Máster en Ingeniería Web</p>
<p>Universidad de Oviedo</p>
</div>
<div class="validations">
<a href="https://validator.w3.org/check?uri=referer">

</a>
<a href=" http://jigsaw.w3.org/css-validator/check/referer ">

</a>
<a href="http://www.w3.org/WAI/WCAG2AAA-Conformance">

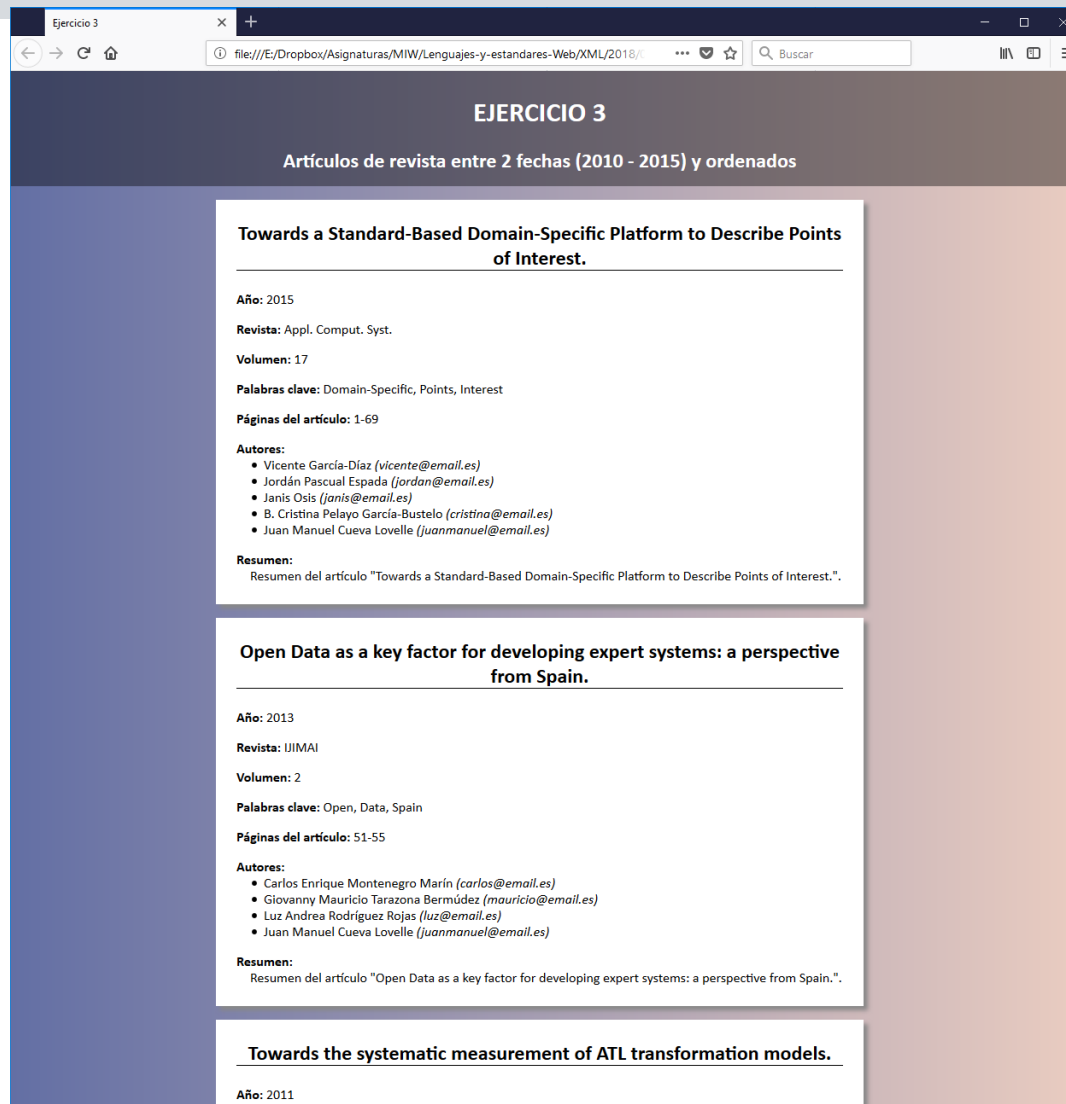
</a>
</div>
</footer>
</main>
</body>
</html>
</xsl:template>

</xsl:stylesheet>
```

Software y estándares para la Web

Ejercicios resueltos (III): Solución – Aspecto del sitio Web generado

Grado en
Ingeniería
Informática
del Software



Software y estándares para la Web

Ejercicios resueltos (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 resueltos (IV): Solución – articulos4.xsl (a)

```
<?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="5.0" encoding="utf-8" indent="yes"/>
<xsl:variable name="author" select="'B. Cristina Pelayo García-Bustelo'"/>

<xsl:template match="articles">
<xsl:text disable-output-escaping='yes'>&lt;!DOCTYPE html></xsl:text>
<html lang="es">
<head>
<meta name="description" content="Ejercicio resuelto 4"/>
<title>Ejercicio 4</title>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
<header>
<h1>EJERCICIO 4</h1>
<h2>
Artículos de revista de un autor (<xsl:value-of select="$author"/>) ordenados por fecha
</h2>
</header>
<main>
<xsl:for-each select="article">
<xsl:sort order="ascending" select="@year"/>
<xsl:if test="contains(authors, $author)">
<section class="article">
<h3>
<xsl:value-of select="title"/>
</h3>
<p>
<span class="section">Año: </span>
<xsl:value-of select="@year"/>
</p>
```

Grado en
Ingeniería
Informática
del Software

Software y estándares para la Web

Ejercicios resueltos (IV): Solución – articulos4.xsl (b)

Grado en
Ingeniería
Informática
del Software

```
<p>
<span class="section">Revista: </span>
<xsl:value-of select="journal"/>
</p>
<p>
<span class="section">Volumen: </span>
<xsl:value-of select="volume"/>
</p>
<xsl:if test="count(keywords//keyword) > 0">
<p>
<span class="section">Palabras clave: </span>
<xsl:for-each select="keywords//keyword">
<xsl:value-of select="."/>
<xsl:if test="not(position()=last())">
<xsl:text>, </xsl:text>
</xsl:if>
</xsl:for-each>
</p>
</xsl:if>
<p>
<span class="section">Páginas del artículo: </span>
<xsl:value-of select="@start-page"/>
<xsl:text>-</xsl:text>
<xsl:value-of select="@end-page"/>
</p>
<p class="authors-section">
<span class="section">Autores: </span>
</p>
```

Software y estándares para la Web

Ejercicios resueltos (IV): Solución – articulos4.xsl (c)

Grado en
Ingeniería
Informática
del Software

```
<ul class="authors">
<xsl:for-each select="authors//author">
<li>
<xsl:value-of select="."/>
<span class="email">
<xsl:text> (</xsl:text>
<xsl:value-of select="@email"/>
<xsl:text>) </xsl:text>
</span>
</li>
</xsl:for-each>
</ul>
<p class="summary-section">
<span class="section">Resumen: </span>
</p>
<p class="summary">
<xsl:value-of select="summary"/>
</p>
</section>
</xsl:if>
</xsl:for-each>
```

Software y estándares para la Web

Ejercicios resueltos (IV): Solución – articulos4.xsl (d)

Grado en
Ingeniería
Informática
del Software

```
<footer>
<div>
<p>Máster en Ingeniería Web</p>
<p>Universidad de Oviedo</p>
</div>
<div class="validations">
<a href="https://validator.w3.org/check?uri=referer">

</a>
<a href=" http://jigsaw.w3.org/css-validator/check/referer ">

</a>
<a href="http://www.w3.org/WAI/WCAG2AAA-Conformance">

</a>
</div>
</footer>
</main>
</body>
</html>
</xsl:template>

</xsl:stylesheet>
```

Software y estándares para la Web

Ejercicios resueltos (IV): Solución – Aspecto del sitio Web generado

Grado en
Ingeniería
Informática
del Software

EJERCICIO 4

Artículos de revista de un autor (B. Cristina Pelayo García-Bustelo) ordenados por fecha

Developing a Business Application with BPM and MDE.

Año: 2009

Revista: IJIMAI

Volumen: 1

Palabras clave: Business, Application, BPM, MDE

Páginas del artículo: 26-32

Autores:

- Héctor Fernández-Fernández (*hector@email.es*)
- Elías Palacios-González (*elias@email.es*)
- Vicente García-Díaz (*vicente@email.es*)
- B. Cristina Pelayo García-Bustelo (*cristina@email.es*)
- Oscar Sanjuán Martínez (*oscar@email.es*)
- Juan Manuel Cueva Lovelle (*juanmanuel@email.es*)

Resumen:

Resumen del artículo "Developing a Business Application with BPM and MDE."

Towards the systematic measurement of ATL transformation models.

Año: 2011

Revista: Softw., Pract. Exper.

Volumen: 41

Palabras clave: ATL Transformation Models

Software y estándares para la Web

Ejercicios resueltos (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^a 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 resueltos (V): Solución – Aspecto del sitio Web generado

Grado en
Ingeniería
Informática
del Software

Véase la
solución en el
archivo
empaquetado
adjunto

The screenshot shows a web browser window with the address bar displaying a file path. The page title is 'EJERCICIO 5' and the subtitle is 'Transformación de recetas de cocina'. The main content is a recipe for 'Arroz integral con pollo'. The recipe details include: Tipo de plato: Primer plato; Plato vegano: No; Ingredientes: Arroz Integral (300 g) / (1032 calorías), Pollo (300 g) / (80 calorías), Cebolleta (1 unidad), Tallos de apio (2 unidades), Dientes de ajo (3 unidades), Tomate frito (2 cucharadas), Curry en pavo (1/2 cucharadita), Pimentón dulce (1/2 cucharadita), Pimiento verde (1/4 unidad), Huevo (1 unidad), Salsa de soja (3 cucharadas), Caldo de pollo (700 ml) / (20 calorías); Elementos necesarios: Cuchillo, Horno, Sartén; Origen de la receta: Las recetas de mi abuela; Dificultad de elaboración: Fácil; Tiempo de elaboración (minutos): 45 minutos; Proceso de elaboración: Paso 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. Paso 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. Paso 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. Paso 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. Paso 5. Una vez reposado el arroz añadimos los trocitos de tortilla y sazonamos con la salsa de soja y un poco de pimienta. Below the recipe is another recipe titled 'Aceituna fritas caseras' with details: Tipo de plato: Entrante; Plato vegano: Si; Ingredientes: Aceitunas rellenas de pimiento (200 g).

Software y estándares para la Web

Ejercicios resueltos (VI): Menú con pollo

- 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

Grado en
Ingeniería
Informática
del Software

Software y estándares para la Web

Ejercicios resueltos (VI): Solución – Aspecto del sitio Web generado

Grado en
Ingeniería
Informática
del Software

Véase la
solución en el
archivo
empaquetado
adjunto

The screenshot shows a web browser window with the address bar displaying a file path. The page has a dark blue header with the text 'EJERCICIO 6' and 'Menú con pollo'. The main content area has a light blue background and features a recipe card for 'Arroz integral con pollo'. The recipe card includes details such as 'Tipo de plato: Primer plato', 'Plato vegano: No', a list of ingredients with their quantities and caloric values, 'Elementos necesarios' (Cuchillo, Horno, Sartén), 'Origen de la receta: Las recetas de mi abuela', 'Dificultad de elaboración: Fácil', 'Tiempo de elaboración (minutos): 45 minutos', and a five-step 'Proceso de elaboración'. Below the first recipe, the start of a second recipe, 'Pechugas de pollo con bechamel', is visible.

EJERCICIO 6
Menú con pollo

Arroz integral con pollo

Tipo de plato: Primer plato
Plato vegano: No

Ingredientes:

- Arroz Integral (300 g) / (1032 calorías)
- Pollo (300 g) / (80 calorías)
- Cebolleta (1 unidad)
- Talos de apio (2 unidades)
- Dientes de ajo (3 unidades)
- Tomate frito (2 cucharadas)
- Curry en pavo (1/2 cucharadita)
- Pimentón dulce (1/2 cucharadita)
- Pimiento verde (1/4 unidad)
- Huevo (1 unidad)
- Salsa de soja (3 cucharadas)
- Caldo de pollo (700 ml) / (20 calorías)

Elementos necesarios:

- Cuchillo
- Horno
- Sartén

Origen de la receta: Las recetas de mi abuela

Dificultad de elaboración: Fácil

Tiempo de elaboración (minutos): 45 minutos

Proceso de elaboración:

Paso 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.

Paso 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.

Paso 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.

Paso 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.

Paso 5. Una vez reposado el arroz añadimos los trocitos de tortilla y sazonamos con la salsa de soja y un poco de pimienta.

Pechugas de pollo con bechamel

Tipo de plato: Primer plato
Plato vegano: No

Ingredientes:

- Pechugas de pollo (4 unidades) / (653 calorías)

Software y estándares para la Web

Ejercicios resueltos (VII): Menú sin leche ni queso

- 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

Grado en
Ingeniería
Informática
del Software

Software y estándares para la Web

Ejercicios resueltos (VII): Solución – Aspecto del sitio Web generado

Grado en
Ingeniería
Informática
del Software

Véase la
solución en el
archivo
empaquetado
adjunto

The screenshot shows a web browser window with the address bar displaying a file path: `file:///E:/Dropbox/Asignaturas/MIW/Lenguajes-y-estandares-Web/XML/2018/04-Ejemplos/recetas-7/recetas.xml`. The page has a dark blue header with the text "EJERCICIO 7" and "Menú sin leche ni queso". The main content area has a light blue background and features a recipe card for "Arroz integral con pollo".

Arroz integral con pollo

Tipo de plato: Primer plato

Plato vegano: No

Ingredientes:

- Arroz Integral (300 g) / (1032 calorías)
- Pollo (300 g) / (80 calorías)
- Cebolleta (1 unidad)
- Talos de apio (2 unidades)
- Dientes de ajo (3 unidades)
- Tomate frito (2 cucharadas)
- Curry en pavo (1/2 cucharadita)
- Pimentón dulce (1/2 cucharadita)
- Pimiento verde (1/4 unidad)
- Huevo (1 unidad)
- Salsa de soja (3 cucharadas)
- Caldo de pollo (700 ml) / (20 calorías)

Elementos necesarios:

- Cuchillo
- Horno
- Sartén

Origen de la receta: Las recetas de mi abuela

Dificultad de elaboración: Fácil

Tiempo de elaboración (minutos): 45 minutos

Proceso de elaboración:

Paso 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.

Paso 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.

Paso 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.

Paso 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.

Paso 5. Una vez reposado el arroz añadimos los trocitos de tortilla y sazonamos con la salsa de soja y un poco de pimienta.

Aceituna fritas caseras

Tipo de plato: Entrante

Plato vegano: Si

Ingredientes:

- Aceitunas rellenas de pimiento (200 g)

Software y estándares para la Web

Ejercicios resueltos (VIII): Menú vegano

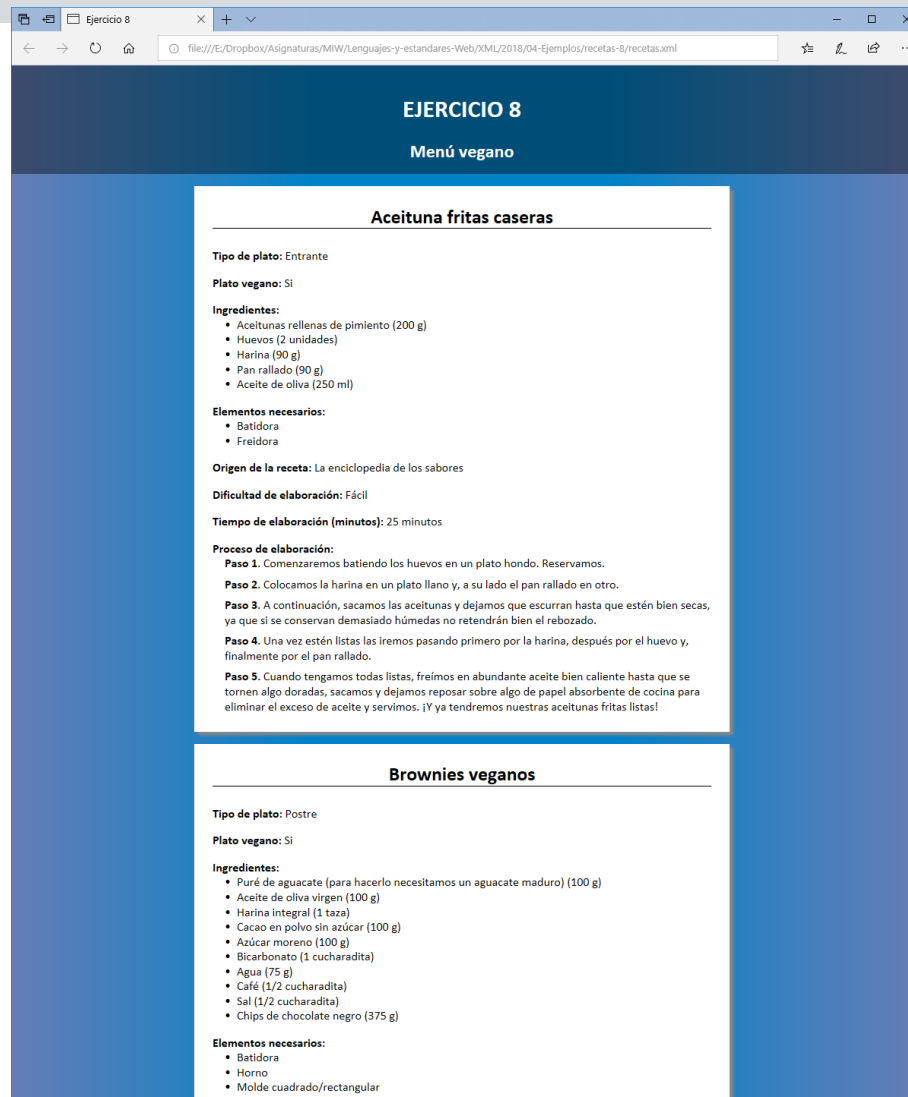
- Supongamos que tenemos una persona vegana invitada
- Se desea generar un menú vegano a partir del documento XML con las recetas de cocina
- El menú vegano debe tener al menos dos entrantes, dos primeros platos, dos segundos platos y postre
- **Transformar el documento XML** por medio de **XSLT** en 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

Grado en
Ingeniería
Informática
del Software

Software y estándares para la Web

Ejercicios resueltos (VIII): Solución – Aspecto del sitio Web generado

Grado en
Ingeniería
Informática
del Software



The screenshot shows a web browser window with the address bar displaying a file path. The page has a dark blue header with the text 'EJERCICIO 8' and 'Menú vegano'. Below the header, there are two recipe cards. The first card is for 'Aceituna fritas caseras' and the second is for 'Brownies veganos'. Each card lists ingredients, necessary elements, and the preparation process.

EJERCICIO 8
Menú vegano

Aceituna fritas caseras

Tipo de plato: Entrante

Plato vegano: Sí

Ingredientes:

- Aceitunas rellenas de pimiento (200 g)
- Huevos (2 unidades)
- Harina (90 g)
- Pan rallado (90 g)
- Aceite de oliva (250 ml)

Elementos necesarios:

- Batidora
- Freidora

Origen de la receta: La enciclopedia de los sabores

Dificultad de elaboración: Fácil

Tiempo de elaboración (minutos): 25 minutos

Proceso de elaboración:

Paso 1. Comenzaremos batiendo los huevos en un plato hondo. Reservamos.

Paso 2. Colocamos la harina en un plato llano y, a su lado el pan rallado en otro.

Paso 3. A continuación, sacamos las aceitunas y dejamos que escurran hasta que estén bien secas, ya que si se conservan demasiado húmedas no retendrán bien el rebozado.

Paso 4. Una vez estén listas las iremos pasando primero por la harina, después por el huevo y, finalmente por el pan rallado.

Paso 5. Cuando tengamos todas listas, freímos en abundante aceite bien caliente hasta que se tornen algo doradas, sacamos y dejamos reposar sobre algo de papel absorbente de cocina para eliminar el exceso de aceite y servimos. ¡Y ya tendremos nuestras aceitunas fritas listas!

Brownies veganos

Tipo de plato: Postre

Plato vegano: Sí

Ingredientes:

- Puré de aguacate (para hacerlo necesitamos un aguacate maduro) (100 g)
- Aceite de oliva virgen (100 g)
- Harina integral (1 taza)
- Cacao en polvo sin azúcar (100 g)
- Azúcar moreno (100 g)
- Bicarbonato (1 cucharadita)
- Agua (75 g)
- Café (1/2 cucharadita)
- Sal (1/2 cucharadita)
- Chips de chocolate negro (375 g)

Elementos necesarios:

- Batidora
- Horno
- Molde cuadrado/rectangular

Véase la
solución en el
archivo
empaquetado
adjunto

Software y estándares para la Web

Esquema

Grado en
Ingeniería
Informática
del Software

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

Software y estándares para la Web

Ejercicios propuestos (I): Transformación de XML a sitio Web (a)

Grado en
Ingeniería
Informática
del Software

- Construir un **documento XML bien formado y válido usando un XML Schema** para contener monumentos del prerrománico asturiano 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): Transformación de XML a sitio Web (b)

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://prerromanicocasturiano.es/>
 - Referencia 3. etc.
- **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): Transformación de XML a sitio Web (a)

- Construir un **documento XML bien formado y válido usando un XML Schema** 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

Grado en
Ingeniería
Informática
del Software

Software y estándares para la Web

Ejercicios propuestos (II): Transformación de XML a sitio Web (b)

Grado en
Ingeniería
Informática
del Software

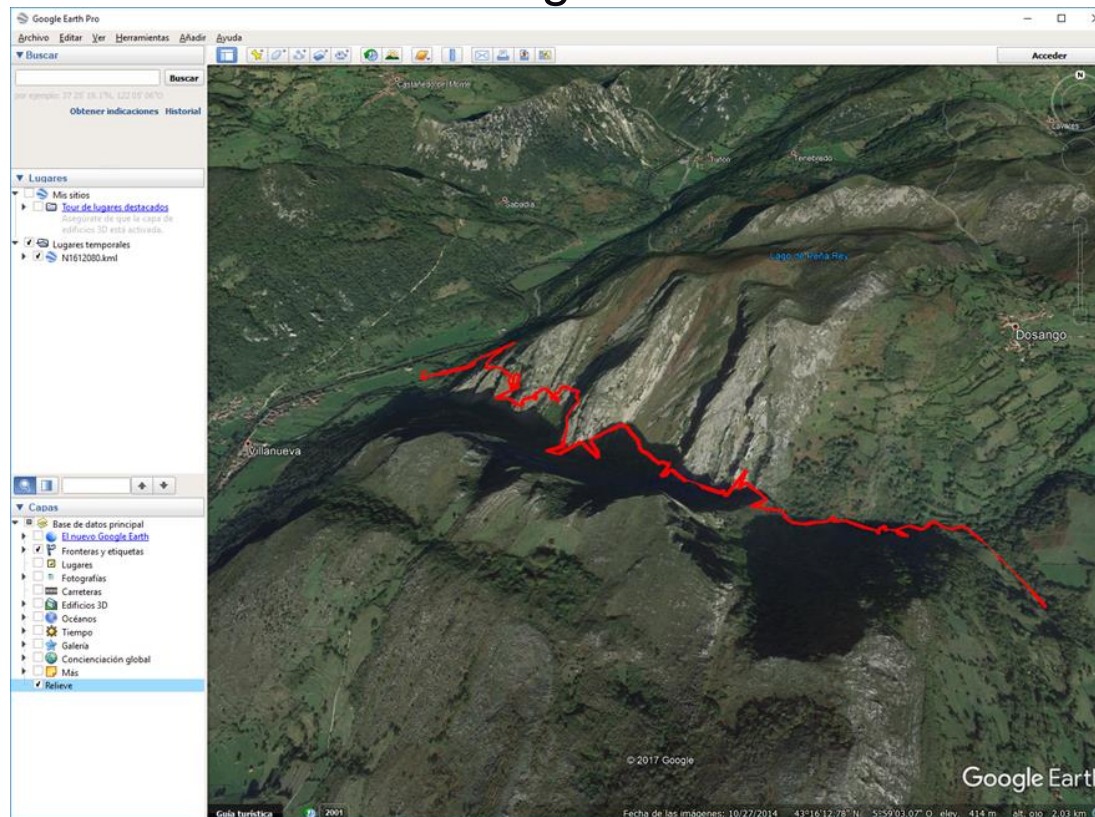
- Referencias y bibliografía con información de la ruta (mínimo 3)
 - Referencia 1: por ejemplo <https://es.wikipedia.org/wiki/Foncalada>
 - Referencia 2: <http://prerromancoasturiano.es/>
 - Referencia 3. etc.
- Recomendación de la ruta de 0 a 10 (por ejemplo “7”)
- Hitos de la ruta (mínimo 3 hitos):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

- **Transformar el documento XML** por medio de **XSLT** en un documento **KML** que muestre el **plano de las rutas turísticas**.
- Debe comprobarse que el archivo generado cumpla estrictamente con el **estándar** de **KML**, comprobando que es posible abrirllos con distintas herramientas como Google Earth.

Grado en
Ingeniería
Informática
del Software

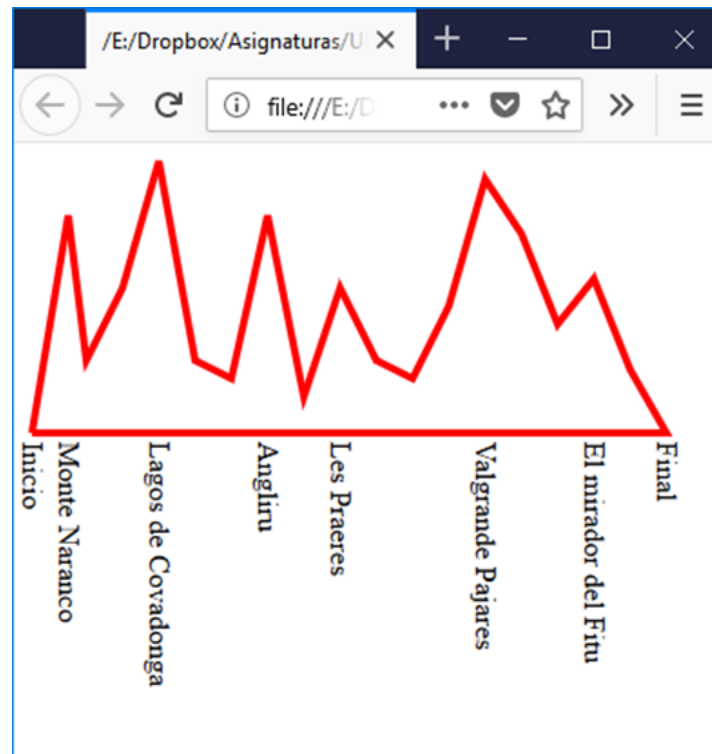


Software y estándares para la Web

Ejercicios propuestos (II): Transformación de XML a SVG (d)

- **Transformar el documento XML** por medio de **XSLT** en un documento **SVG** que muestre el **perfil altimétrico de las rutas turísticas**.
- Debe comprobarse que el archivo generado cumpla estrictamente el **estándar** de **SVG**, comprobando que es posible abrirlos con distintos agentes de usuario (navegadores).

Grado en
Ingeniería
Informática
del Software





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