

# CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

Unidad 5

# CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- ◉ XML separa los datos que se deben almacenar, la estructura y su presentación.
- ◉ Deberemos utilizar herramientas para convertir estos documentos en otros formatos más amigables y útiles→ XSL (Extensible Stylesheet Language).
- ◉ Con XSL podemos:
  - Establecer criterios sobre qué datos ver.
  - Establecer el orden de visualización.
  - Establecer filtros.
  - Definir formatos de salida.
- ◉ Vamos a partir de un ejemplo y poco a poco iremos evolucionando los conceptos y complementándolos.

# EJEMPLO INICIAL

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

```
<libreria>
```

```
  <libro>
```

```
    <titulo>Physics-based animation</titulo>
```

```
    <autor>Kenny Erleben</autor>
```

```
    <editor>Charles River Media</editor>
```

```
    <isbn>978-1584503804</isbn>
```

```
    <precio>50.06</precio>
```

```
  </libro>
```

```
  <libro>
```

```
    <titulo>Principios de seguridad informática para usuarios</titulo>
```

```
    <autor>Carlos Garre</autor>
```

```
    <editor>Dykinson</editor>
```

```
    <isbn>978-84-9849-998-8</isbn>
```

```
    <precio>13.00</precio>
```

```
  </libro>
```

```
  <libro>
```

```
    <titulo>Ejercicios complementarios de lógica digital</titulo>
```

```
    <autor>Alberto Sánchez</autor>
```

```
    <editor>Dykinson</editor>
```

```
    <isbn>978-84-9849-703-8</isbn>
```

```
    <precio>12.00</precio>
```

```
  </libro>
```

```
</libreria>
```

# EJEMPLO INICIAL

- ◉ El archivo XML contiene la información de una biblioteca con la información de los libros.
- ◉ A continuación vamos a diseñar el XSL para que muestre formateadamente en HTML por ejemplo.

# EJEMPLO INICIAL

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca</h1>
    <table>
      <tr bgcolor="#887788">
        <th>Título</th>
        <th>Autor</th>
      </tr>
      <xsl:for-each select="libreria/libro">
        <tr>
          <td><xsl:value-of select="titulo"/></td>
          <td><xsl:value-of select="autor"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

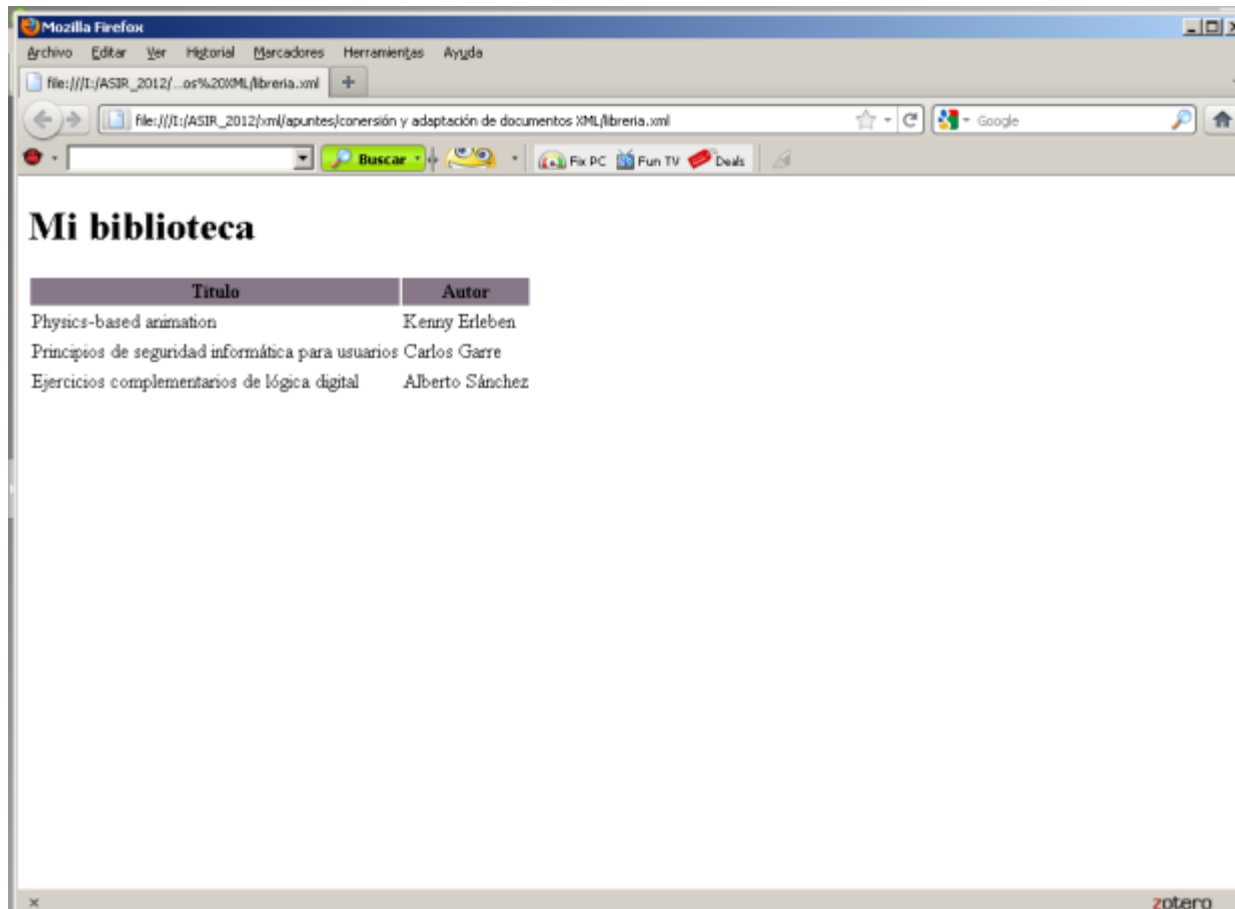
# EJEMPLO INICIAL

- ⦿ Este XSL define que para cada libro almacenado en el XML de la librería se añade una entrada en una tabla HTML que muestra el título y autor de cada libro.
- ⦿ Si además lo vinculamos...

# EJEMPLO INICIAL

```
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet version="1.0" type="text/xsl" href="librería_xsl.xsl"?>
<libreria>
  <libro>
    <titulo>Physics-based animation</titulo>
    <autor>Kenny Erleben</autor>
    <editor>Charles River Media</editor>
    <isbn>978-1584503804</isbn>
    <precio>50.06</precio>
  </libro>
  <libro>
    <titulo>Principios de seguridad informática para usuarios</titulo>
    <autor>Carlos Garre</autor>
    <editor>Dykinson</editor>
    <isbn>978-84-9849-998-8</isbn>
    <precio>13.00</precio>
  </libro>
  <libro>
    <titulo>Ejercicios complementarios de lógica digital</titulo>
    <autor>Alberto Sánchez</autor>
    <editor>Dykinson</editor>
    <isbn>978-84-9849-703-8</isbn>
    <precio>12.00</precio>
  </libro>
</libreria>.
```

# EJEMPLO INICIAL





# ACLARACIONES

- ◉ Podemos realizar transformaciones de cualquier XML a cualquier formato que pueda ser reconocido por un navegador web.
- ◉ Pasos a seguir para realizar la conversión adecuadamente:
  - Tener bien definido el documento XML.
  - Crear una hoja de estilo XSL bien formada.
  - Vincular al documento XML la hoja de estilo XSL.
- ◉ Cumplidos los pasos anteriores el procesador de documentos XSL recorre todo el árbol del documento XML.

# ACLARACIONES

- ◉ Mediante la plantilla en la hoja de estilos podremos definir qué formatos tendrán cada una de las partes del documento.
- ◉ Una plantilla va a ser el patrón que puede generar resultados de formateo en el árbol del documento final.

# EJERCICIOS

1. Insertar varios libros más en el documento XML anterior y ver cómo se modifica el resultado, insertar alguno cuyo precio sea inferior a 10.50 euros.
2. Insertar el número de páginas en cada libro en el fichero XML.
3. Cambiar el título de la tabla a “Mi biblioteca personal”.
4. Cambiar el color de la tabla en el fichero XSL.
5. Añadir una columna más al final en la que se muestre el precio del libro.
6. Añadir una columna más al principio en la que se muestre el ISBN.

# EJERCICIOS

2. Insertar el número de páginas en cada libro en el fichero XML.

Ejemplo:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet type="text/xsl" href="libreria_xsl.xsl"?>
<libreria>
  <libro>
    <titulo>Physics-based animation</titulo>
    <autor>Kenny Erleben</autor>
    <editor>Charles River Media</editor>
    <isbn>978-1584503804</isbn>
    <precio>50.06</precio>
    <numpag>200</numpag>
  </libro>
</libreria>
```

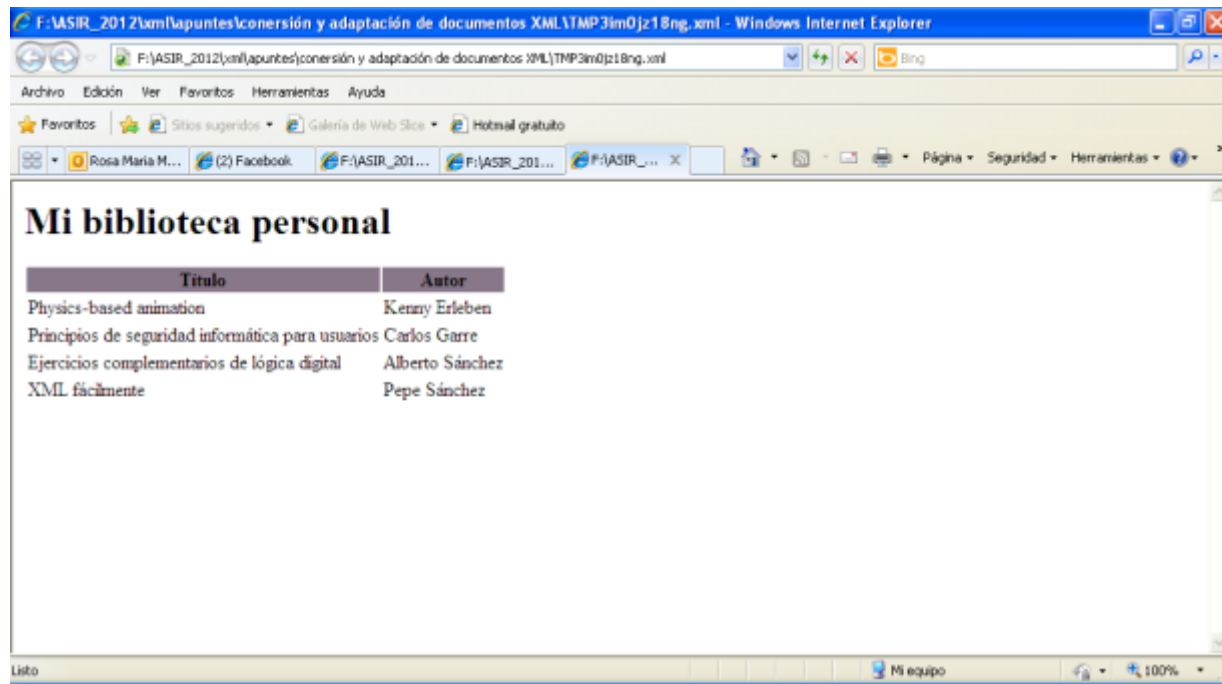
# EJERCICIOS

## 3. Cambiar el título de la tabla a “Mi biblioteca personal”.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
      <table>
        <tr bgcolor="#887788">
          <th>Título</th>
          <th>Autor</th>
        </tr>
        <xsl:for-each select="libreria/libro">
          <tr>
            <td><xsl:value-of select="titulo"/></td>
            <td><xsl:value-of select="autor"/></td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

# EJERCICIOS

## 3. Cambiar el título de la tabla a “Mi biblioteca personal”.



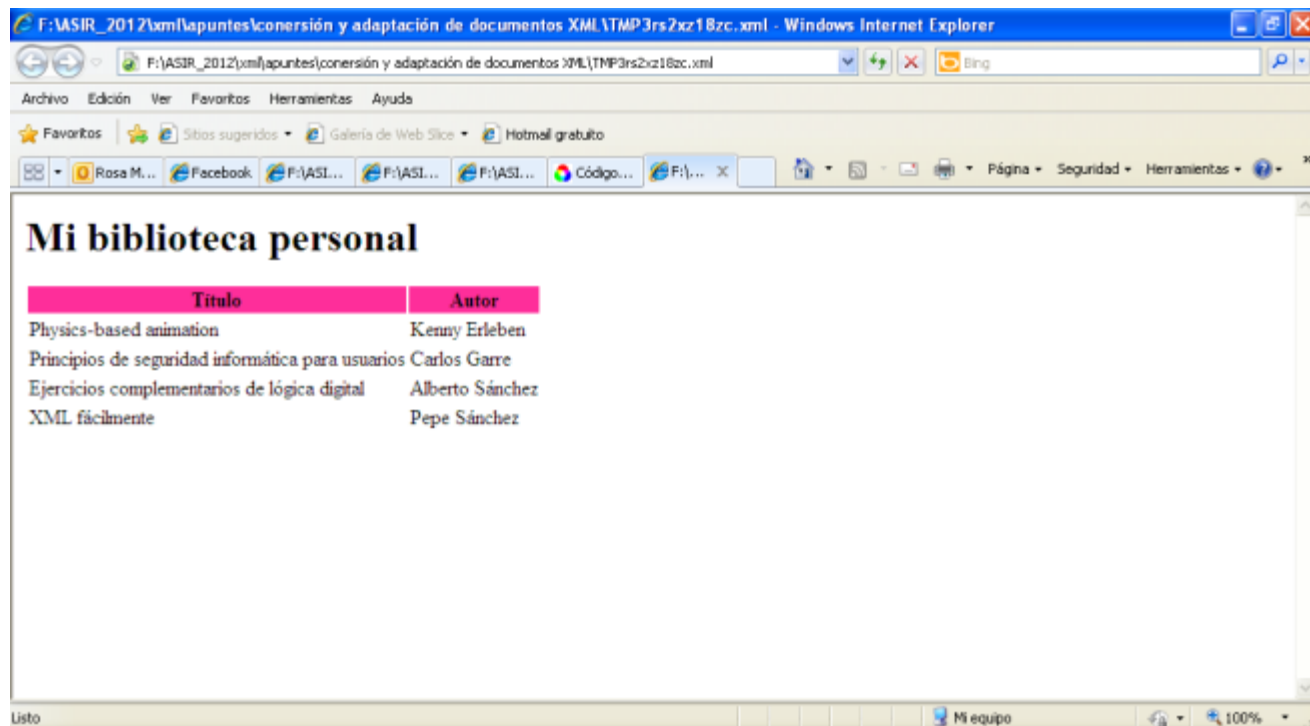
# EJERCICIOS

## 4. Cambiar el color de la tabla en el fichero XSL.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
      <table>
        <tr bgcolor="#FE2E9A">
          <th>Título</th>
          <th>Autor</th>
        </tr>
        <xsl:for-each select="libreria/libro">
          <tr>
            <td><xsl:value-of select="titulo"/></td>
            <td><xsl:value-of select="autor"/></td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

# EJERCICIOS

## 4. Cambiar el color de la tabla en el fichero XSL.



Códigos de colores: <http://html-color-codes.info/codigos-de-colores-hexadecimales/>



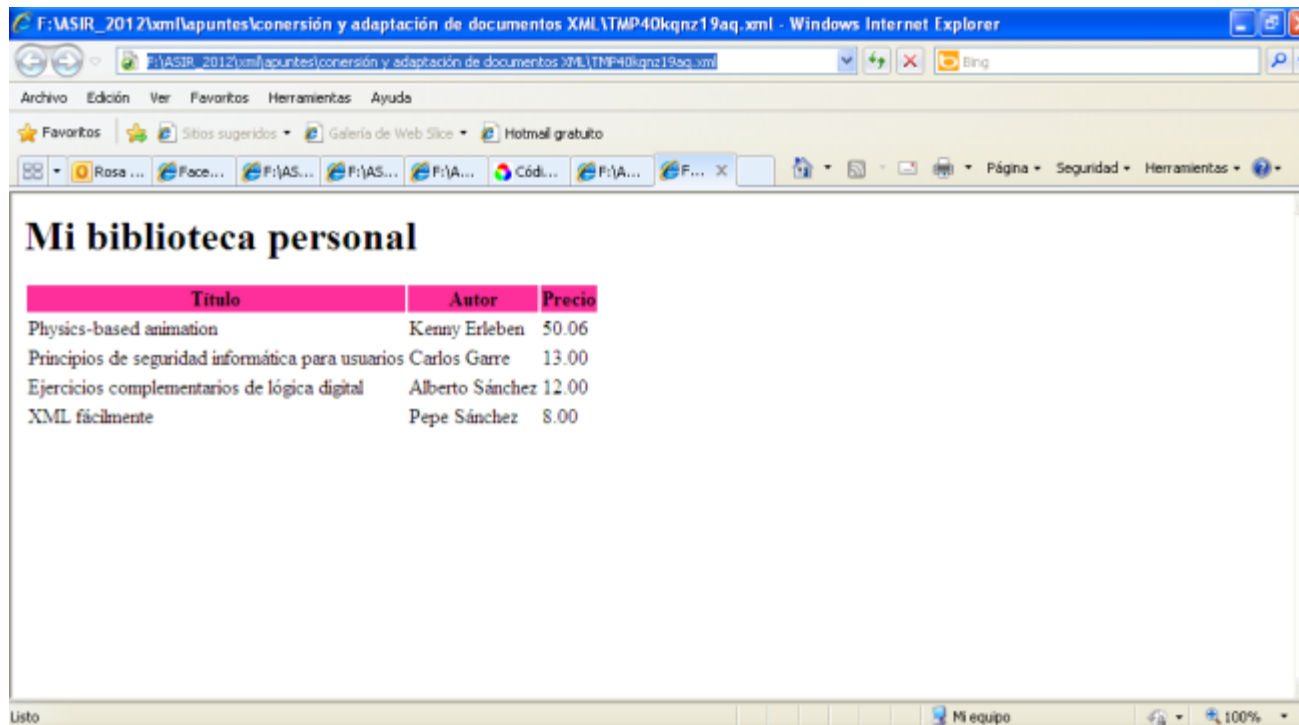
# EJERCICIOS

## 5. Añadir una columna más al final en la que se muestre el precio del libro.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
    <table>
      <tr bgcolor="#FE2E9A">
        <th>Título</th>
        <th>Autor</th>
        <th>Precio</th>
      </tr>
      <xsl:for-each select="libreria/libro">
        <tr>
          <td><xsl:value-of select="titulo"/></td>
          <td><xsl:value-of select="autor"/></td>
          <td><xsl:value-of select="precio"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# EJERCICIOS

5. Añadir una columna más al final en la que se muestre el precio del libro.



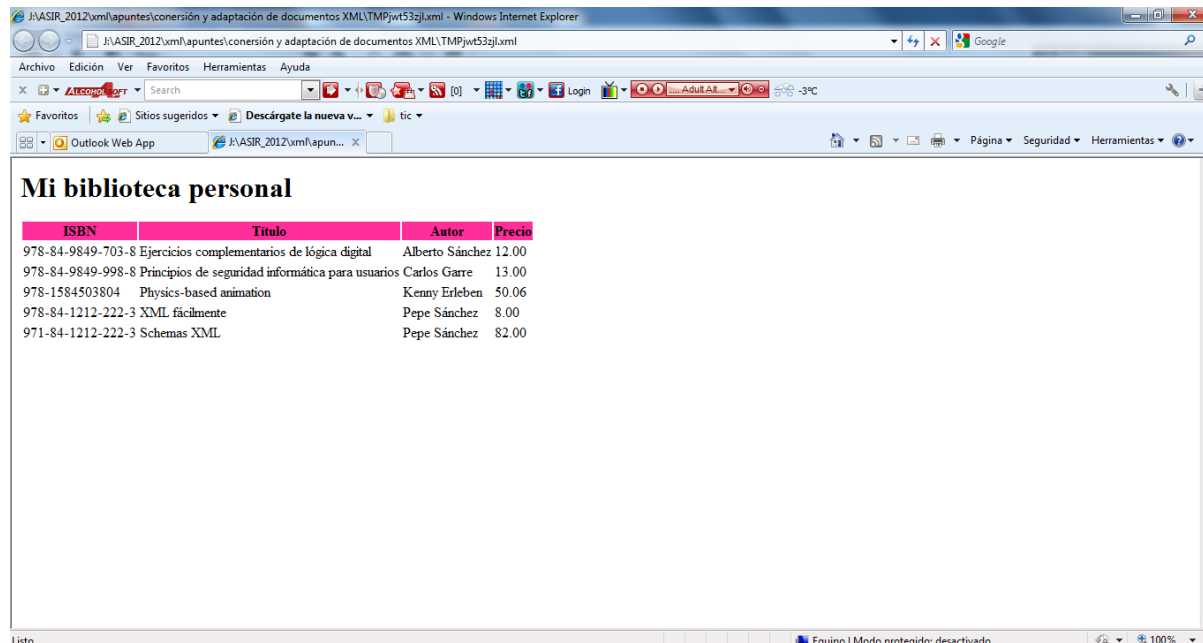
# EJERCICIOS

## 6. Añadir una columna más al principio en la que se muestre el ISBN.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
      <table>
        <tr bgcolor="#FE2E9A">
          <th>ISBN</th>
          <th>Título</th>
          <th>Autor</th>
          <th>Precio</th>
        </tr>
        <xsl:for-each select="libreria/libro">
          <tr>
            <td><xsl:value-of select="isbn"/></td>
            <td><xsl:value-of select="titulo"/></td>
            <td><xsl:value-of select="autor"/></td>
            <td><xsl:value-of select="precio"/></td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

# EJERCICIOS

6. Añadir una columna más al principio en la que se muestre el ISBN.



The screenshot shows a web browser window with the title "Mi biblioteca personal". The browser's address bar shows a local file path. The table displayed has four columns: ISBN, Título, Autor, and Precio. The data rows are as follows:

ISBN	Título	Autor	Precio
978-84-9849-703-8	Ejercicios complementarios de lógica digital	Alberto Sánchez	12.00
978-84-9849-998-8	Principios de seguridad informática para usuarios	Carlos Garre	13.00
978-1584503804	Physics-based animation	Kenny Erleben	50.06
978-84-1212-222-3	XML fácilmente	Pepe Sánchez	8.00
971-84-1212-222-3	Schemas XML	Pepe Sánchez	82.00

# TRANSFORMACIÓN DE DOCUMENTOS XSL

- ⦿ XSL es un lenguaje que me va a permitir definir un conjunto de reglas que, aplicadas sobre un documento XML, nos posibilitará transformarlo en un resultado formateado más adecuado a lo que deseamos.
- ⦿ Las reglas las almacenaremos en un fichero XSL (hoja de estilo).

# ESTRUCTURA ARCHIVO XSL

- ◉ El fichero XSL debe estar bien formado.
- ◉ Debe contener:
  - Declaración inicial XML:  
`<?xml version="1.0" encoding="ISO-8859-1"?>`
  - Etiqueta que identifica al fichero XSL y el espacio de nombres en el que se basa:  
`<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">`  
o  
`<xsl:transform version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">`
  - Se pueden utilizar cualquiera de las dos... soportadas por el estándar W3C.

# ESTRUCTURA ARCHIVO XSL

## ◉ Debe contener también:

- Definición de un elemento plantilla dentro del XSL (después de las otras dos instrucciones vistas):

```
<xsl:template match="/">
```

```
...
```

```
</xsl:template>
```

Lo que esté entre esas etiquetas será lo que permitirá generar la salida formateada.

Usando el atributo match indicaremos sobre qué parte del documento XML se quiere incidir: en el ejemplo se incidía sobre la raíz ("/").

# ELEMENTOS BÁSICOS

## ◉ XSL:FOR EACH

- Si queremos recorrer todos los elementos hijos y extraer datos para volver a formatearlos.

- Ejemplo:

```
<xsl:for-each select="librería/libro">
```

...

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

En este caso estamos indicando que se recorran todos los elementos que sean libros.



# ELEMENTOS BÁSICOS

## ○ XSL:VALUE-OF

- Para extraer el contenido del libro.
- Ejemplo:

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

# ELEMENTOS BÁSICOS

## ◉ XSL:SORT

- Se utiliza para mostrar ordenadamente por uno de los elementos.

- Ejemplo:

```
<xsl:sort select="título"/>
```

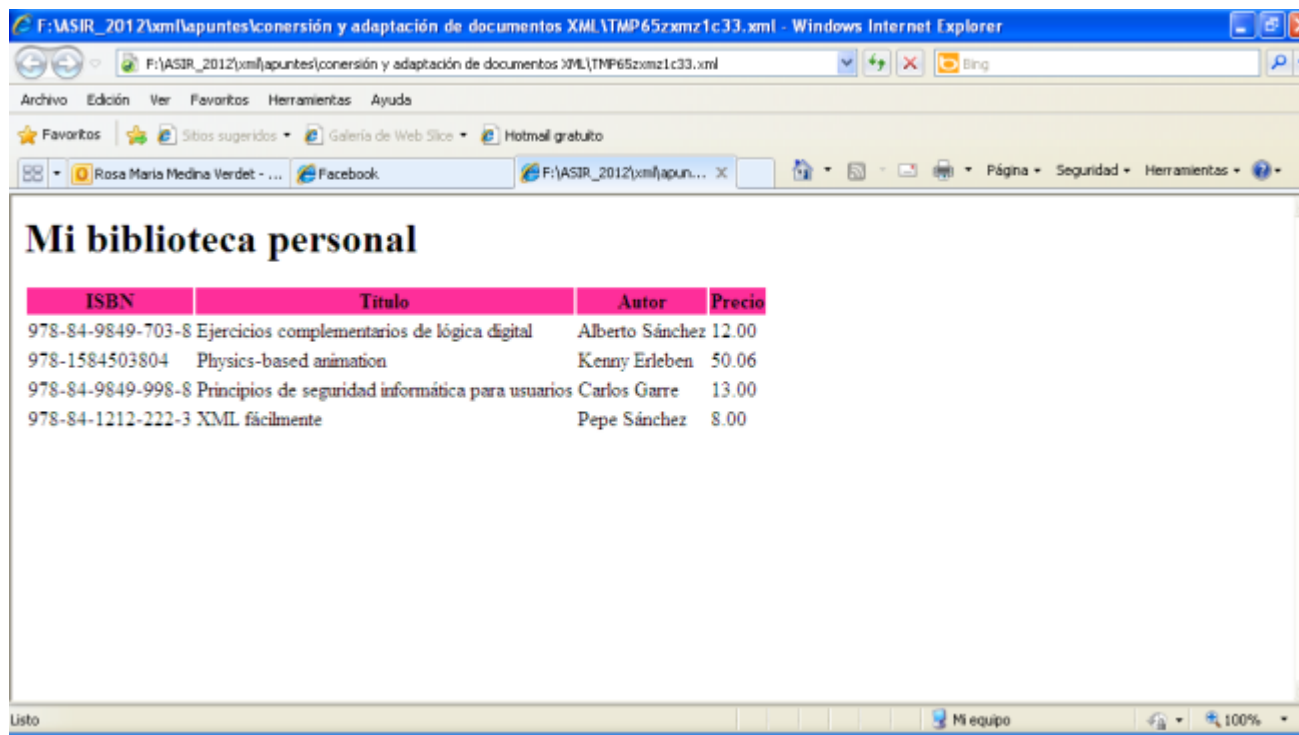
# ELEMENTOS BÁSICOS

## ◉ XSL:SORT-ejemplo7:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
    <table>
      <tr bgcolor="#FE2E9A">
        <th>ISBN</th>
        <th>Título</th>
        <th>Autor</th>
        <th>Precio</th>
      </tr>
      <xsl:for-each select="libreria/libro">
        <xsl:sort select="titulo"/>
        <tr>
          <td><xsl:value-of select="isbn"/></td>
          <td><xsl:value-of select="titulo"/></td>
          <td><xsl:value-of select="autor"/></td>
          <td><xsl:value-of select="precio"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# ELEMENTOS BÁSICOS

## ◉ XSL: SORT-ejemplo:



# ELEMENTOS BÁSICOS

## 🕸 XSL:SORT-ejemplo 8:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
    <table>
      <tr bgcolor="#FE2E9A">
        <th>ISBN</th>
        <th>Título</th>
        <th>Autor</th>
        <th>Precio</th>
      </tr>
      <xsl:for-each select="libreria/libro">
        <xsl:sort select="autor"/>
        <tr>
          <td><xsl:value-of select="isbn"/></td>
          <td><xsl:value-of select="titulo"/></td>
          <td><xsl:value-of select="autor"/></td>
          <td><xsl:value-of select="precio"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# ELEMENTOS BÁSICOS

## ○ XSL: SORT-ejemplo:



# ELEMENTOS BÁSICOS

## ◉ Búsquedas:

- Se pueden realizar búsquedas con patrones preestablecidos (filtrado por un patrón).
- Ejemplo: Obtener todos los libros del autor: “Kenny Erleben”.

```
<xsl:for-each select=“librería/libro[autor=‘Kenny Erleben’]”>
```

...

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

# ELEMENTOS BÁSICOS

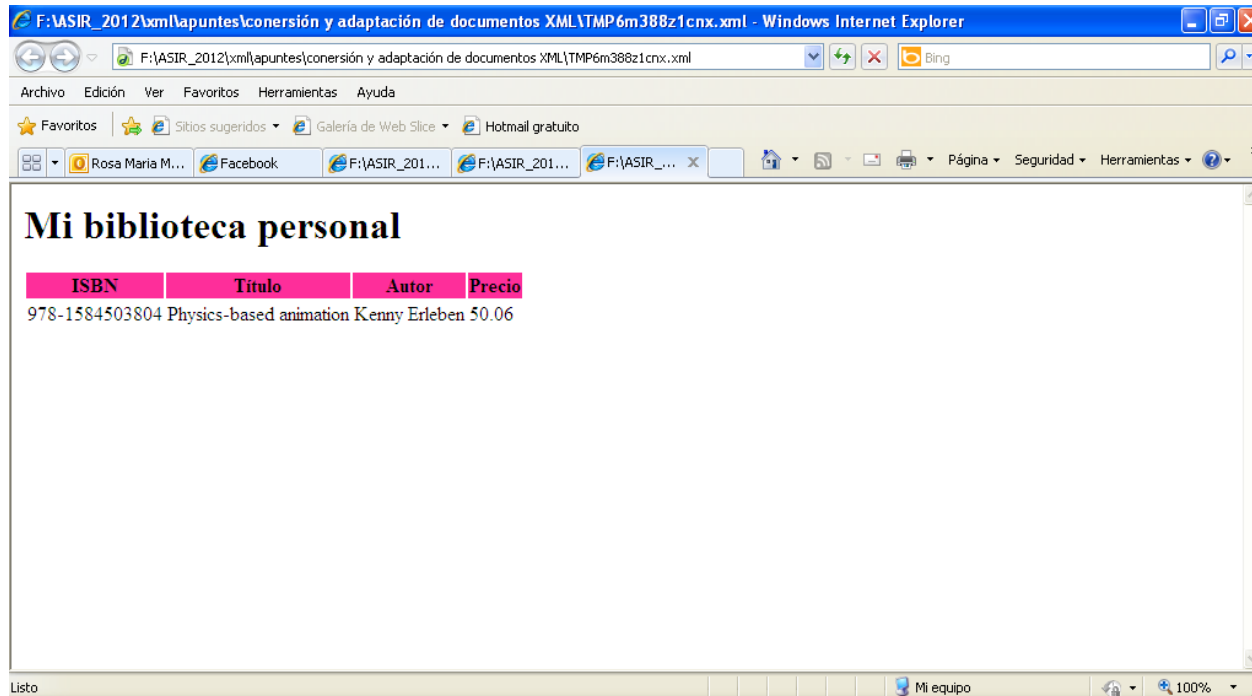
## 🕯 XSL:BÚSQUEDAS-ejemplo9:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
    <table>
      <tr bgcolor="#FE2E9A">
        <th>ISBN</th>
        <th>Título</th>
        <th>Autor</th>
        <th>Precio</th>
      </tr>
      <xsl:for-each select="libreria/libro[autor='Kenny Erleben']">
        <xsl:sort select="autor"/>
        <tr>
          <td><xsl:value-of select="isbn"/></td>
          <td><xsl:value-of select="titulo"/></td>
          <td><xsl:value-of select="autor"/></td>
          <td><xsl:value-of select="precio"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```



# ELEMENTOS BÁSICOS

## ○ XSL: BÚSQUEDAS-ejemplo:



# ELEMENTOS BÁSICOS

## ◉ Búsquedas 2:

- Ejemplo: Obtener todos los libros que no sean del autor: “Carlos Garre”.

```
<xsl:for-each select=“librería/libro[autor!=‘Carlos Garre’]”>
```

...

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

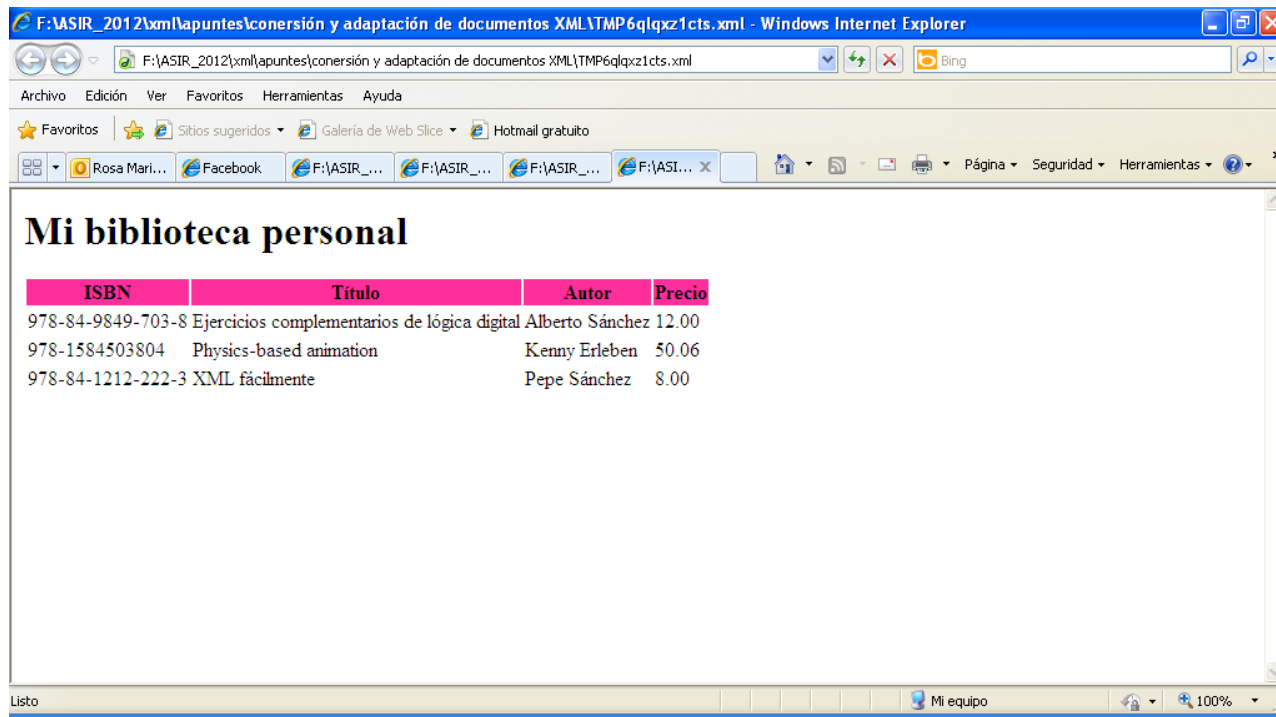
# ELEMENTOS BÁSICOS

## 🕸 XSL:BÚSQUEDAS-ejemplo 10:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
    <table>
      <tr bgcolor="#FE2E9A">
        <th>ISBN</th>
        <th>Título</th>
        <th>Autor</th>
        <th>Precio</th>
      </tr>
      <xsl:for-each select="libreria/libro[autor!='Carlos Garre']">
        <xsl:sort select="autor"/>
        <tr>
          <td><xsl:value-of select="isbn"/></td>
          <td><xsl:value-of select="titulo"/></td>
          <td><xsl:value-of select="autor"/></td>
          <td><xsl:value-of select="precio"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# ELEMENTOS BÁSICOS

## ○ XSL: BÚSQUEDAS-ejemplo 2:



## EJERCICIOS 2- PROPUESTOS

1. Insertar en el fichero XML que se indica inicialmente, un nuevo libro cuyo autor sea el mismo a uno ya incluido anteriormente (ej11).
2. Buscar los libros del autor anterior y comprobar que salen todos sus libros (ordenados por autor) (ej12).
3. Buscar todos los libros que no son del autor anterior. Comprobar que en los resultados faltan sus libros (ej13).

## EJERCICIOS 2- PROPUESTOS

1. Insertar en el fichero XML que se indica inicialmente, un nuevo libro cuyo autor sea el mismo a uno ya incluido anteriormente.

<libro>

<titulo>Schemas XML</titulo>

<autor>Pepe Sánchez</autor>

<editor>BM</editor>

<isbn>971-84-1212-222-3</isbn>

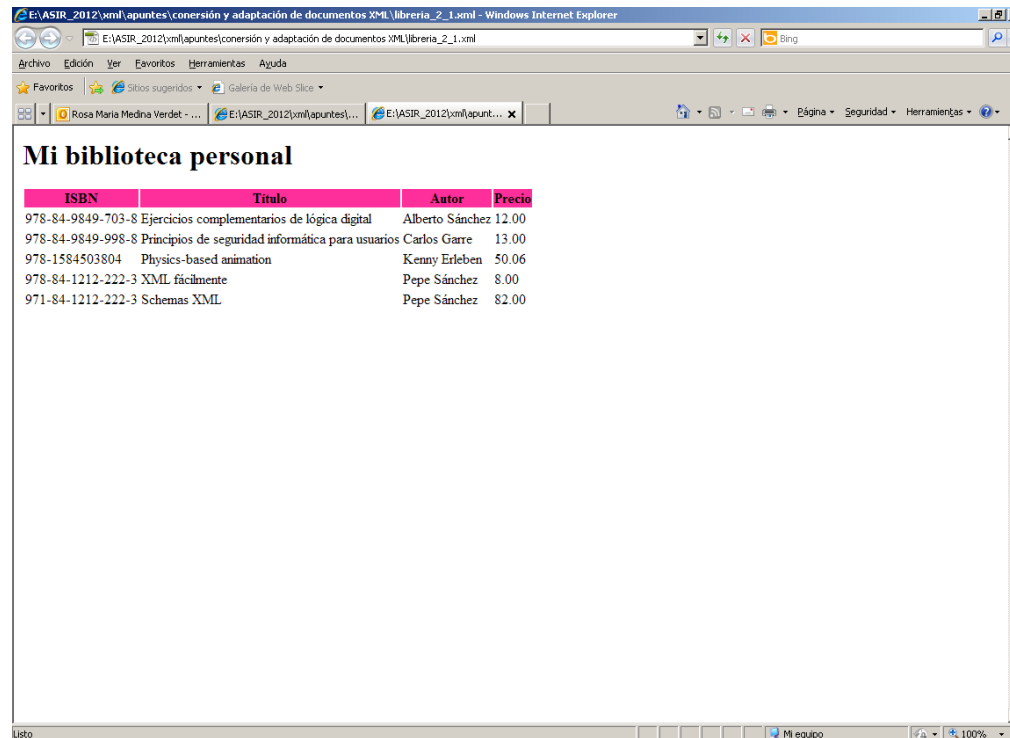
<precio>82.00</precio>

<numpag>150</numpag>

</libro>

# EJERCICIOS 2- PROPUESTOS

1. Insertar en el fichero XML que se indica inicialmente, un nuevo libro cuyo autor sea el mismo a uno ya incluido anteriormente.



The screenshot shows a Windows Internet Explorer browser window with the title bar 'E:\ASIR\_2012\xml\apuntes\conversión y adaptación de documentos XML\libreria\_2\_1.xml - Windows Internet Explorer'. The address bar shows the file path 'E:\ASIR\_2012\xml\apuntes\conversión y adaptación de documentos XML\libreria\_2\_1.xml'. The browser displays a web page titled 'Mi biblioteca personal'. The page contains a table with four columns: ISBN, Título, Autor, and Precio. The table lists five books with their respective details.

ISBN	Título	Autor	Precio
978-84-9849-703-8	Ejercicios complementarios de lógica digital	Alberto Sánchez	12.00
978-84-9849-998-8	Principios de seguridad informática para usuarios	Carlos Garre	13.00
978-1584503804	Physics-based animation	Kenny Erleben	50.06
978-84-1212-222-3	XML fácilmente	Pepe Sánchez	8.00
971-84-1212-222-3	Schemas XML	Pepe Sánchez	82.00

# EJERCICIOS 2- PROPUESTOS

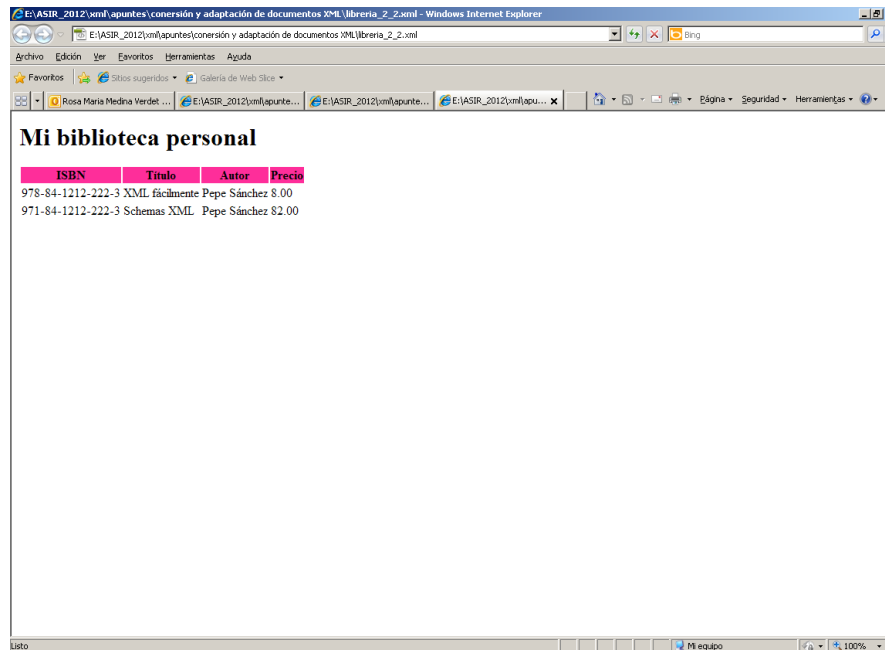
2. Buscar los libros del autor anterior y comprobar que salen todos sus libros (ordenados por autor).

```
<?xml version="1.0" encoding="iso-8859-1"
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
    <table>
      <tr bgcolor="#FE2E9A">
        <th>ISBN</th>
        <th>Título</th>
        <th>Autor</th>
        <th>Precio</th>
      </tr>
      <xsl:for-each select="libreria/libro[autor='Pepe Sánchez']">
        <xsl:sort select="autor"/>
        <tr>
          <td><xsl:value-of select="isbn"/></td>
          <td><xsl:value-of select="titulo"/></td>
          <td><xsl:value-of select="autor"/></td>
          <td><xsl:value-of select="precio"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```



# EJERCICIOS 2- PROPUESTOS

2. Buscar los libros del autor anterior y comprobar que salen todos sus libros (ordenados por autor).



# EJERCICIOS 2- PROPUESTOS

3. Buscar todos los libros que no son del autor anterior. Comprobar que en los resultados faltan sus libros.

```
<?xml version="1.0" encoding="iso-8859-1"?><!-- DWXMLSource="http://" -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>

    <body>

        <h1>Mi biblioteca personal</h1>

        <table>

            <tr bgcolor="#FE2E9A">

                <th>ISBN</th>
                <th>Título</th>
                <th>Autor</th>
                <th>Precio</th>

            </tr>
            <xsl:for-each select="libreria/libro[autor!='Pepe Sánchez']">
            <xsl:sort select="autor"/>
            <tr>

                <td><xsl:value-of select="isbn"/></td>
                <td><xsl:value-of select="titulo"/></td>
                <td><xsl:value-of select="autor"/></td>
                <td><xsl:value-of select="precio"/></td>

            </tr>
            </xsl:for-each>

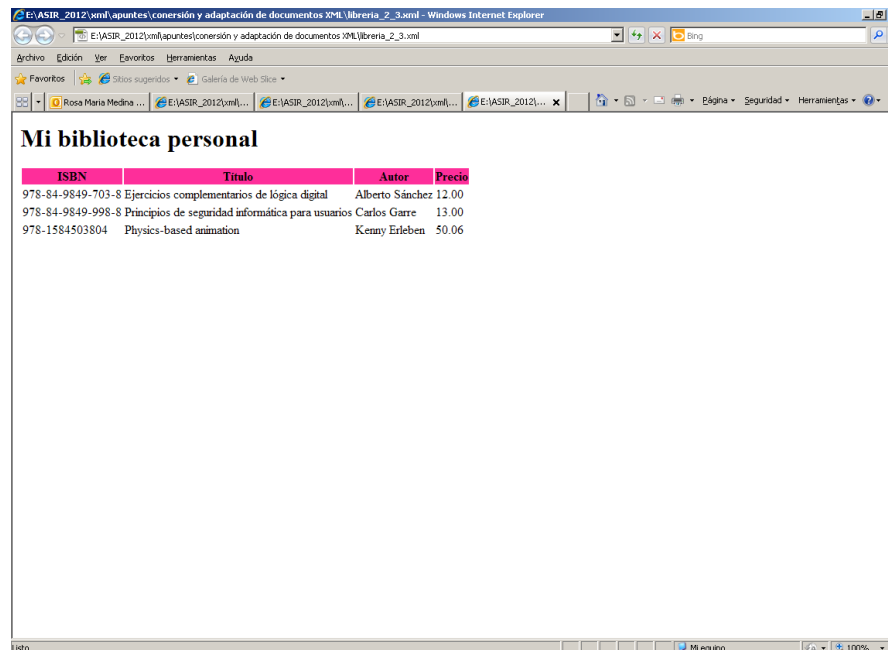
        </table>

    </body>

</html>
</xsl:template>
</xsl:stylesheet>
```

# EJERCICIOS 2- PROPUESTOS

2. Buscar los libros del autor anterior y comprobar que salen todos sus libros (ordenados por autor).



The screenshot shows a Windows Internet Explorer browser window. The address bar displays a local file path: E:\ASIR\_2012\xml\apuntes\conversión y adaptación de documentos XML\libreria\_2\_3.xml. The browser's menu bar includes Archivo, Edición, Ver, Favoritos, Herramientas, and Ayuda. The toolbar shows icons for Favorites, Suggested Sites, Web Slice Gallery, and a search bar with the Bing logo. The address bar also shows several open tabs, all with the same local file path. The main content area is titled "Mi biblioteca personal" and contains a table with four columns: ISBN, Título, Autor, and Precio. The table lists three books.

ISBN	Título	Autor	Precio
978-84-9849-703-8	Ejercicios complementarios de lógica digital	Alberto Sánchez	12.00
978-84-9849-998-8	Principios de seguridad informática para usuarios	Carlos Garre	13.00
978-1584503804	Physics-based animation	Kenny Erleben	50.06

# OPERADORES EN XSL

## ◉ Operadores comunes:

- Hasta el momento se ha trabajado con operadores de igualdad y desigualdad pero existen más operadores:

- ◉ Operadores lógicos:

- Igualdad (=): “=”.
- Desigualdad ( $\neq$ ): “!=”.
- Menor que (<): “&lt;”.
- Mayor que (>): “&gt;”.

# OPERADORES EN XSL

## ◉ Elemento XSL:IF:

- A veces no es suficiente con esos operadores, usaremos: “<xsl:if>” para condiciones más complejas.
- Sintaxis:  
    <xsl:if test=“expresion”>  
    ...  
    </xsl:if>

# OPERADORES EN XSL

## ◉ Elemento XSL:IF:

### ■ Ejemplo14:

Mostrar aquellos libros cuyo coste sea superior a 12 euros.

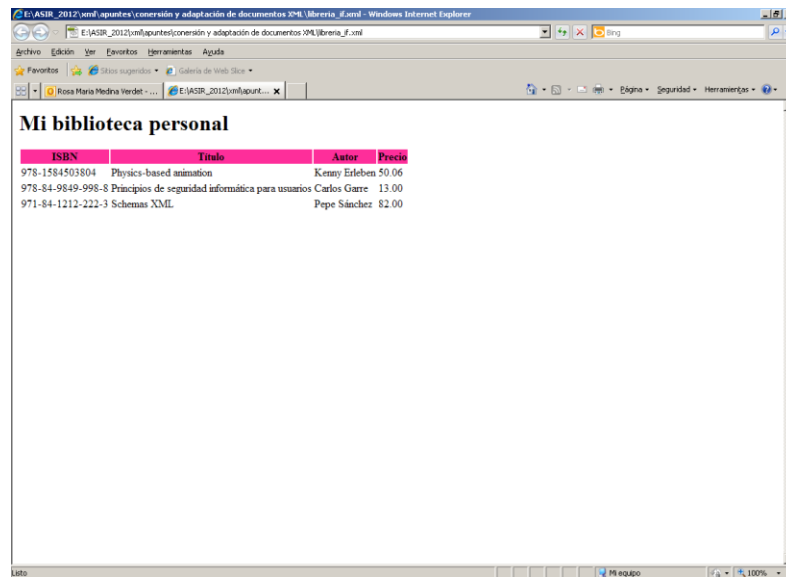
```
<?xml version="1.0" encoding="iso-8859-1"?><!-- DWXMLSource="http://" -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Mi biblioteca personal</h1>
    <table>
      <tr bgcolor="#FE2E9A">
        <th>ISBN</th>
        <th>Título</th>
        <th>Autor</th>
        <th>Precio</th>
      </tr>
      <xsl:for-each select="libreria/libro">
        <xsl:if test="precio > 12">
          <tr>
            <td><xsl:value-of select="isbn"/></td>
            <td><xsl:value-of select="titulo"/></td>
            <td><xsl:value-of select="autor"/></td>
            <td><xsl:value-of select="precio"/></td>
          </tr>
        </xsl:if>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# OPERADORES EN XSL

## ◉ Elemento XSL:IF:

### ■ Ejemplo:

- Mostrar aquellos libros cuyo coste sea superior a 12 euros.



Windows Internet Explorer

E:\ASIR\_2012\unl\apuntes\conversión y adaptación de documentos XML\libreria\_f.xml

Archivo Edición Ver Favoritos Herramientas Ayuda

Favoritos Sitios sugeridos Galería de Web Sites

Rosa María Medina Verdell E:\ASIR\_2012\unl\apuntes...

**Mi biblioteca personal**

ISBN	Título	Autor	Precio
978-1584503804	Physics-based animation	Kenny Erleben	50.06
978-84-9849-998-8	Principios de seguridad informática para usuarios	Carlos Garre	13.00
971-84-1212-222-3	Schemas XML	Pepe Sánchez	82.00

Estado

MS equipo

100%

## EJERCICIOS 3- PROPUESTOS

1. Insertar en el fichero XML que se indica inicialmente, dos nuevos libros cuyo autor sea nuevo. Sus previos serán: 10.50 y 12.50 euros (ej15).
2. Buscar los libros cuyo coste sea superior a 10 euros y comprobar que salen todos sus libros(los nuevos también)(ej16).
3. En el mismo XSL anterior añade una condición en la que se muestren aquellos cuyo autor sea el último definido (ej17).
4. Cambiar la condición del coste del libro. Ahora solo se quiere aquellos libros cuyo coste sea superior a 12 euros. Comprobar que solamente aparece un libro de los introducidos anteriormente(ej18).



# EJERCICIOS 3- PROPUESTOS

1. Insertar en el fichero XML que se indica inicialmente, dos nuevos libros cuyo autor sea nuevo. Sus previos serán: 10.50 y 12.50 euros.

<libro>

<titulo>Ejercicios XSL</titulo>

<autor>Romerales</autor>

<editor>Anaya</editor>

<isbn>978-88-1212-222-3</isbn>

<precio>10.50</precio>

<numpag>120</numpag>

</libro>

<libro>

<titulo>La biblia de XML</titulo>

<autor>Romerales</autor>

<editor>Anaya</editor>

<isbn>979-89-1212-222-3</isbn>

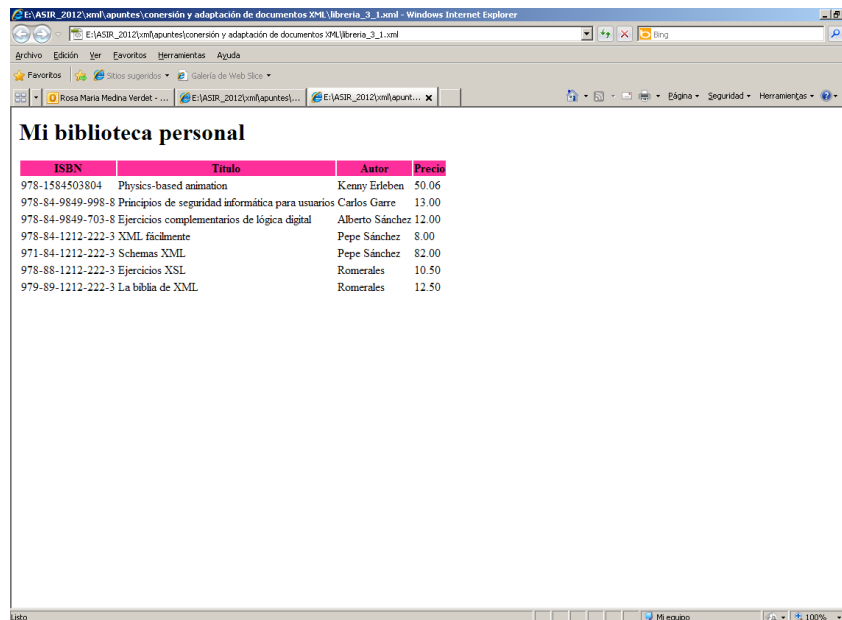
<precio>12.50</precio>

<numpag>189</numpag>

</libro>

# EJERCICIOS 3- PROPUESTOS

1. Insertar en el fichero XML que se indica inicialmente, dos nuevos libros cuyo autor sea nuevo. Sus previos serán: 10.50 y 12.50 euros.



The screenshot shows a web browser window with the title "Mi biblioteca personal". The browser's address bar shows a file path: "E:\ASIR\_2012\xml\apuntes\conversión y adaptación de documentos XML\libreria\_3\_1.xml". The table below is displayed on the page.

ISBN	Título	Autor	Precio
978-1584503804	Physics-based animation	Kenny Erleben	50.06
978-84-9849-998-8	Principios de seguridad informática para usuarios	Carlos Garre	13.00
978-84-9849-703-8	Ejercicios complementarios de lógica digital	Alberto Sánchez	12.00
978-84-1212-222-3	XML fácilmente	Pepe Sánchez	8.00
971-84-1212-222-3	Schemas XML	Pepe Sánchez	82.00
978-88-1212-222-3	Ejercicios XSL	Romerales	10.50
979-89-1212-222-3	La biblia de XML	Romerales	12.50

# EJERCICIOS 3- PROPUESTOS

2. Buscar los libros cuyo coste sea superior a 10 euros y comprobar que salen todos sus libros(los nuevos también).

```
<?xml version="1.0" encoding="iso-8859-1"?><!-- DWXMLSource="http://" -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>

    <body>

        <h1>Mi biblioteca personal</h1>

        <table>

            <tr bgcolor="#FE2E9A">
                <th>ISBN</th>
                <th>Título</th>
                <th>Autor</th>
                <th>Precio</th>

            </tr>
            <xsl:for-each select="libreria/libro">
                <xsl:if test="precio &gt; 10">

                    <tr>

                        <td><xsl:value-of select="isbn"/></td>
                        <td><xsl:value-of select="titulo"/></td>
                        <td><xsl:value-of select="autor"/></td>
                        <td><xsl:value-of select="precio"/></td>

                    </tr>
                </xsl:if>
            </xsl:for-each>

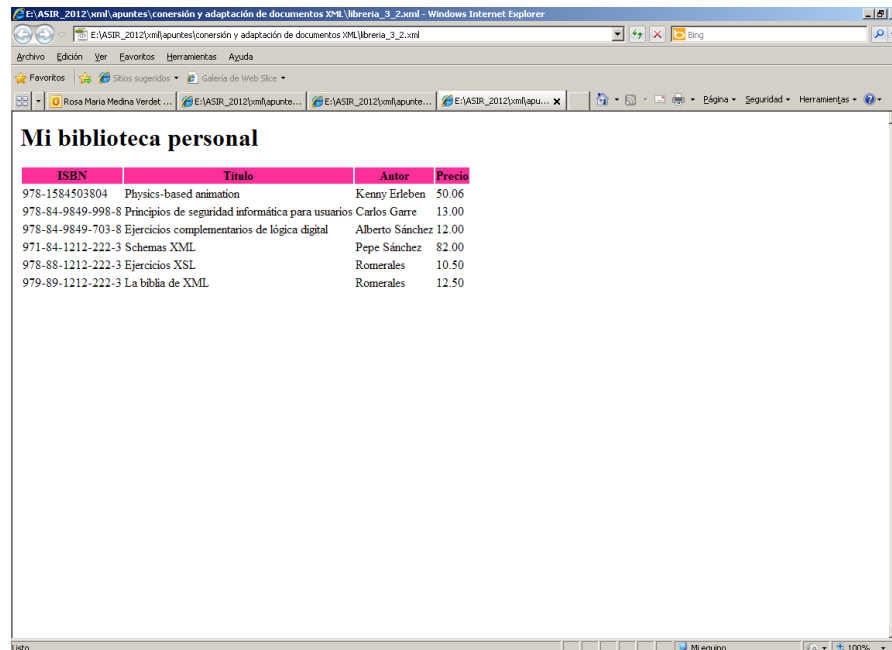
        </table>

    </body>

</html>
</xsl:template>
</xsl:stylesheet>
```

# EJERCICIOS 3- PROPUESTOS

2. Buscar los libros cuyo coste sea superior a 10 euros y comprobar que salen todos sus libros(los nuevos también).



The screenshot shows a web browser window with the title "Mi biblioteca personal". The browser's address bar shows a file path: "E:\ASIR\_2012\xml\apuntes\conversión y adaptación de documentos XML\libreria\_3\_2.xml". The table below lists books with columns for ISBN, Title, Author, and Price. The prices are in euros.

ISBN	Título	Autor	Precio
978-1584503804	Physics-based animation	Kenny Erleben	50.06
978-84-9849-998-8	Principios de seguridad informática para usuarios	Carlos Garre	13.00
978-84-9849-703-8	Ejercicios complementarios de lógica digital	Alberto Sánchez	12.00
971-84-1212-222-3	Schemas XML	Pepe Sánchez	82.00
978-88-1212-222-3	Ejercicios XSL	Romerales	10.50
979-89-1212-222-3	La biblia de XML	Romerales	12.50

# EJERCICIOS 3- PROPUESTOS

3. En el mismo XSL anterior añade una condición en la que se muestren aquellos cuyo autor sea el último definido.

```
<?xml version="1.0" encoding="iso-8859-1"?><!-- DWXMLSource="http://" -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>

    <body>

        <h1>Mi biblioteca personal</h1>

        <table>

            <tr bgcolor="#FEZE9A">

                <th>ISBN</th>
                <th>Titulo</th>
                <th>Autor</th>
                <th>Precio</th>

            </tr>
            <xsl:for-each select="libreria/libro[autor='Romerales']">
            <xsl:if test="precio &gt; 10">

                <tr>

                    <td><xsl:value-of select="isbn"/></td>
                    <td><xsl:value-of select="titulo"/></td>
                    <td><xsl:value-of select="autor"/></td>
                    <td><xsl:value-of select="precio"/></td>

                </tr>

            </xsl:for-each>

        </table>

    </body>

</html>
</xsl:template>
</xsl:stylesheet>
```

# EJERCICIOS 3- PROPUESTOS

3. En el mismo XSL anterior añade una condición en la que se muestren aquellos cuyo autor sea el último definido.

Otra opción:

```
<xsl:for-each select="libreria/libro">
```

```
  <xsl:if test="precio > 10">
```

```
    <xsl:if test="autor = 'Romerales'">
```

```
      <tr>
```

```
        <td><xsl:value-of select="isbn"/></td>
```

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

```
        <td><xsl:value-of select="autor"/></td>
```

```
        <td><xsl:value-of select="precio"/></td>
```

```
      </tr>
```

```
    </xsl:if>
```

```
  </xsl:if>
```

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

# EJERCICIOS 3- PROPUESTOS

3. En el mismo XSL anterior añade una condición en la que se muestren aquellos cuyo autor sea el último definido.



## EJERCICIOS 3- PROPUESTOS

4. Cambiar la condición del coste del libro. Ahora solo se quiere aquellos libros cuyo coste sea superior a 12 euros. Comprobar que solamente aparece un libro de los introducidos anteriormente.

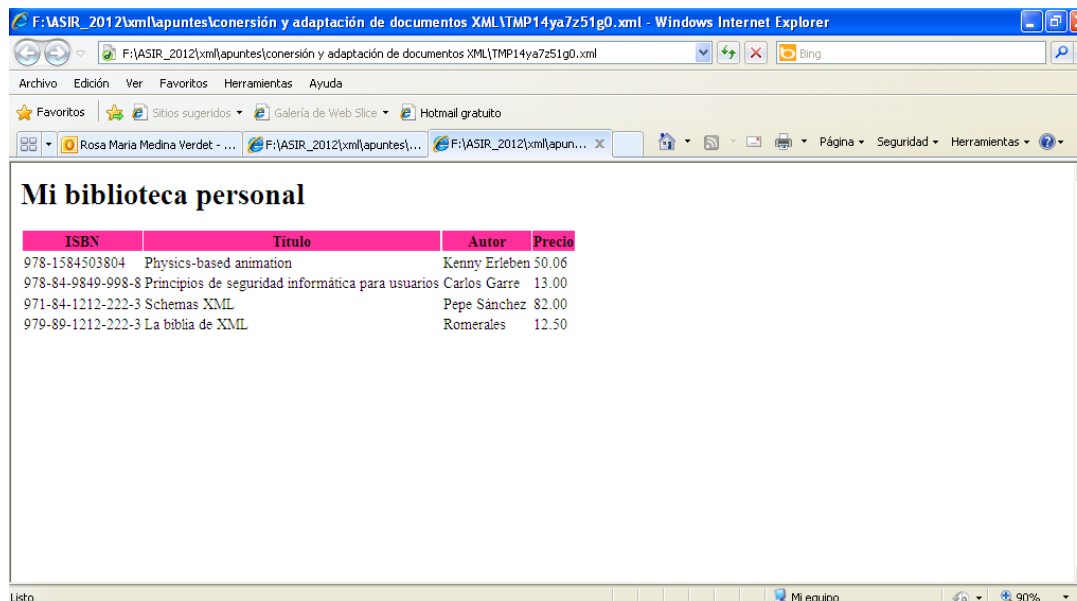
Otra opción:

```
<xsl:for-each select="libreria/libro">
  <xsl:if test="precio > 12">
    <tr>
      <td><xsl:value-of select="isbn"/></td>
      <td><xsl:value-of select="titulo"/></td>
      <td><xsl:value-of select="autor"/></td>
      <td><xsl:value-of select="precio"/></td>
    </tr>
  </xsl:if>
</xsl:for-each>
```



# EJERCICIOS 3- PROPUESTOS

4. Cambiar la condición del coste del libro. Ahora solo se quiere aquellos libros cuyo coste sea superior a 12 euros. Comprobar que solamente aparece un libro de los introducidos anteriormente.



ISBN	Título	Autor	Precio
978-1584503804	Physics-based animation	Kenny Erleben	50.06
978-84-9849-998-8	Principios de seguridad informática para usuarios	Carlos Garre	13.00
971-84-1212-222-3	Schemas XML	Pepe Sánchez	82.00
979-89-1212-222-3	La biblia de XML	Romerales	12.50

# OPERADORES EN XSL\*\*\*

## ◉ Elemento XSL:CHOOSE (ej19):

- Para poder establecer múltiples condiciones (similar a switch de C o CASE de Pascal).
- Con “<xsl:when>” se establecen las condiciones.
- Para establecer una condición por defecto la cláusula sería: “<xsl:otherwise>”.

### ■ Sintaxis:

```
<xsl:choose>
<xsl:when test=“expresion”>
...
</xsl:when>
<xsl:when test =“expresion”>
...
</xsl:when>
<xsl:otherwise>
...
</xsl:otherwise>
</xsl:choose>
```

# OPERADORES EN XSL

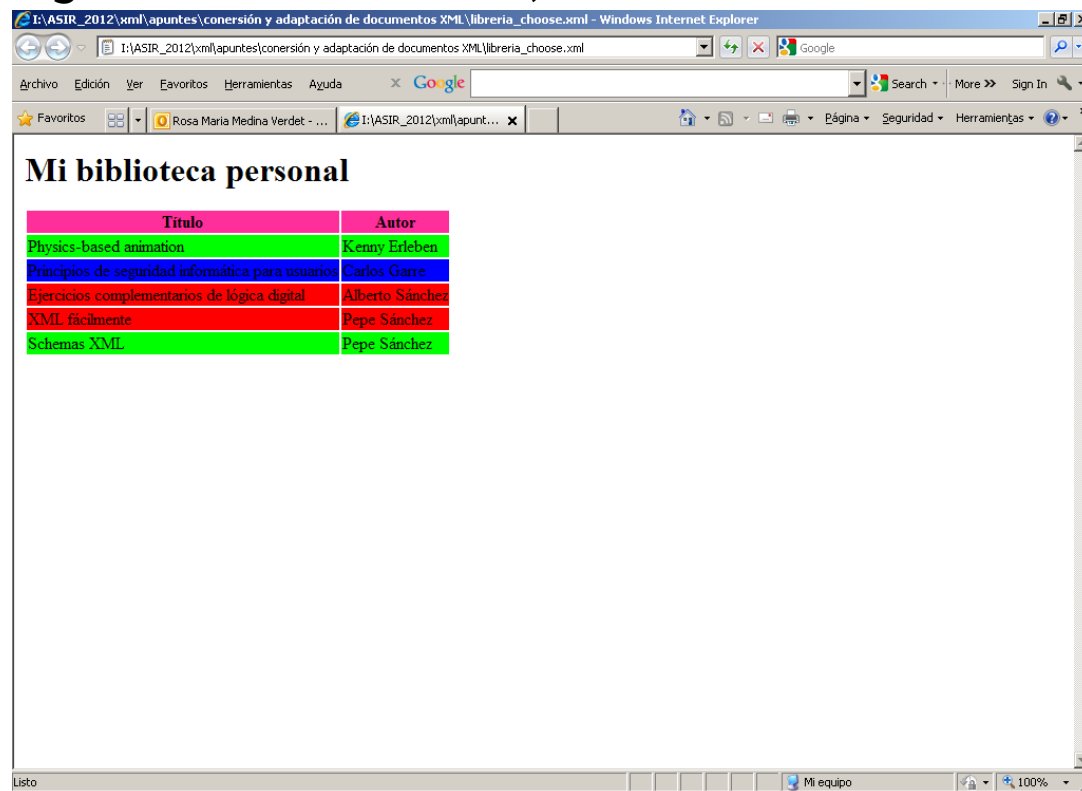
- ◉ Elemento XSL:CHOOSE- ejemplo: vamos a realizar un xsl en el cual si el libro es menor de 12.50 euros muestre en color rojo todos sus datos. Si es mayor de 25.50 euros en color verde. Si no cumple ninguna de las anteriores, en color azul.(ej20)

```
<xsl:for-each select="libreria/libro">

  <tr>
    <xsl:choose>
      <xsl:when test="precio &lt; 12.50">
        <td bgcolor="#ff0000">
          <xsl:value-of select="titulo"/></td>
          <td bgcolor="#ff0000">
            <xsl:value-of select="autor"/></td>
        </xsl:when>
        <xsl:when test="precio &gt; 25.50">
          <td bgcolor="#00ff00">
            <xsl:value-of select="titulo"/></td>
            <td bgcolor="#00ff00">
              <xsl:value-of select="autor"/></td>
          </xsl:when>
          <xsl:otherwise>
            <td bgcolor="#0000ff">
              <xsl:value-of select="titulo"/></td>
              <td bgcolor="#0000ff">
                <xsl:value-of select="autor"/></td>
            </xsl:otherwise>
          </xsl:choose>
        </tr>
      </xsl:for-each>
```

# OPERADORES EN XSL

- ◉ Elemento XSL:CHOOSE- ejemplo: vamos a realizar un xsl en el cual si el libro es menor de 12.50 euros muestre en color rojo todos sus datos. Si es mayor de 25.50 euros en color verde. Si no cumple ninguna de las anteriores, en color azul.



Windows Internet Explorer

I:\ASIR\_2012\xml\apuntes\conversión y adaptación de documentos XML\libreria\_choose.xml

Google

Archivo Edición Ver Favoritos Herramientas Ayuda

Google

Search More >> Sign In

Favoritos Rosa Maria Medina Verdet - ... I:\ASIR\_2012\xml\apuntes...

Página Seguridad Herramientas

### Mi biblioteca personal

Título	Autor
Physics-based animation	Kenny Erleben
Principios de seguridad informática para usuarios	Carlos Garre
Ejercicios complementarios de lógica digital	Alberto Sánchez
XML fácilmente	Pepe Sánchez
Schemas XML	Pepe Sánchez

Listo

Mi equipo

100%

## EJERCICIOS 4- PROPUESTOS

1. Insertar en el fichero XML que se indica inicialmente, dos nuevos libros de un nuevo autor. Sus precios serán: 10.50 y 60.50 euros respectivamente (ej21).
2. Añadir las columnas precio al final e ISBN al principio de la tabla de resultados (ej21).
3. Cambiar los colores para que si el coste es mayor que 25 euros, las filas sean de color rojo y verde si es menor que 25 euros (ej22).

# EJERCICIOS 4- PROPUESTOS

1. Insertar en el fichero XML que se indica inicialmente, dos nuevos libros de un nuevo autor. Sus precios serán: 10.50 y 60.50 euros respectivamente.

<libro>

<titulo>DTD's</titulo>

<autor>Pepe Ríos</autor>

<editor>BM</editor>

<isbn>971-84-1212-111-3</isbn>

<precio>60.50</precio>

<numpag>125</numpag>

</libro>

<libro>

<titulo>RSS</titulo>

<autor>JOsé Sánchez</autor>

<editor>Anaya</editor>

<isbn>971-84-1111-222-3</isbn>

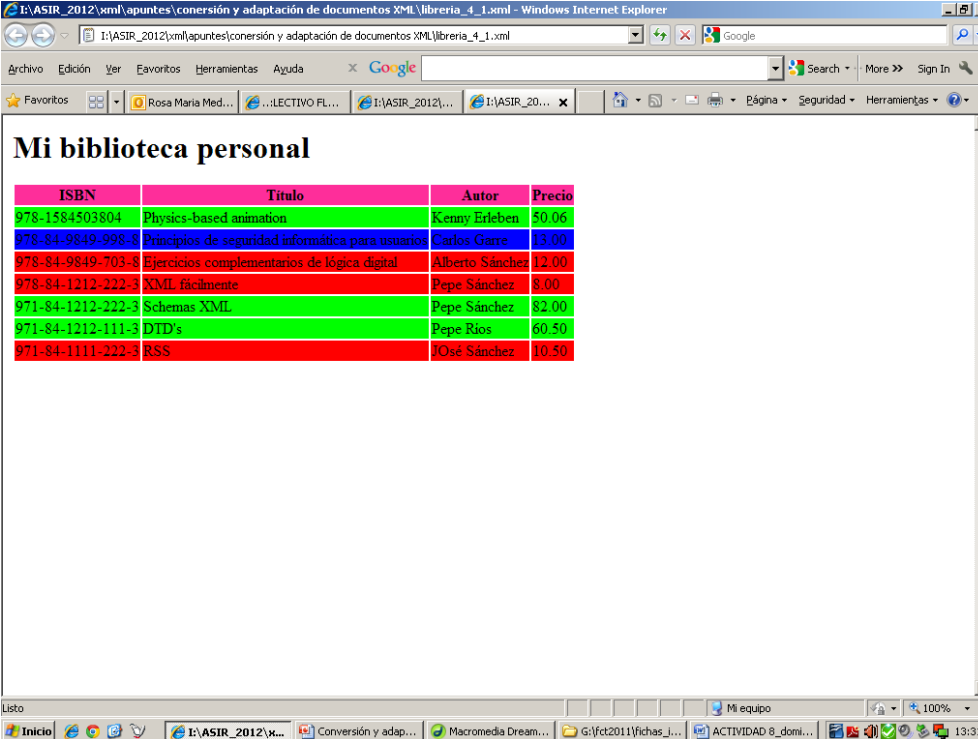
<precio>10.50</precio>

<numpag>220</numpag>

</libro>

# EJERCICIOS 4- PROPUESTOS

1. Añadir las columnas precio al final e ISBN al principio de la tabla de resultados.



The screenshot shows a web browser window with the title "Mi biblioteca personal". The browser's address bar shows a local file path: "I:\ASIR\_2012\xml\apuntes\conersión y adaptación de documentos XML\libreria\_4\_1.xml". The table displayed has four columns: ISBN, Título, Autor, and Precio. The data rows are as follows:

ISBN	Título	Autor	Precio
978-1584503804	Physics-based animation	Kenny Erleben	50.06
978-84-9849-998-8	Principios de seguridad informática para usuarios	Carlos Garre	13.00
978-84-9849-703-8	Ejercicios complementarios de lógica digital	Alberto Sánchez	12.00
978-84-1212-222-3	XML fácilmente	Pepe Sánchez	8.00
971-84-1212-222-3	Schemas XML	Pepe Sánchez	82.00
971-84-1212-111-3	DTD's	Pepe Ríos	60.50
971-84-1111-222-3	RSS	JOSé Sánchez	10.50

# EJERCICIOS 4- PROPUESTOS

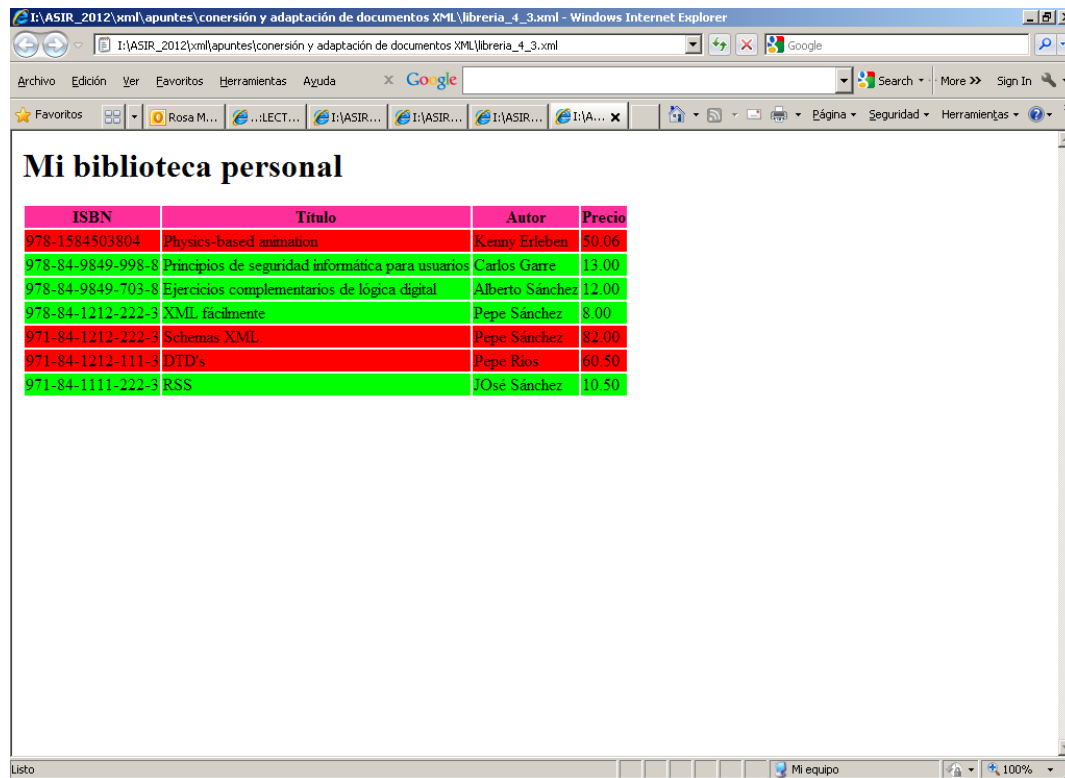
3. Cambiar los colores para que si el coste es mayor que 25 euros, las filas sean de color rojo y verde si es menor que 25 euros.

```
<xsl:for-each select="libreria/libro">
  <tr>
    <xsl:choose>
      <xsl:when test="precio > 25.00">
        <td bgcolor="#ff0000">
          <xsl:value-of select="isbn"/></td>
          <td bgcolor="#ff0000">
            <xsl:value-of select="titulo"/></td>
        <td bgcolor="#ff0000">
          <xsl:value-of select="autor"/></td>
          <td bgcolor="#ff0000">
            <xsl:value-of select="precio"/></td>
        </xsl:when>
        <xsl:when test="precio < 25.00">
          <td bgcolor="#00ff00">
            <xsl:value-of select="isbn"/></td>
            <td bgcolor="#00ff00">
              <xsl:value-of select="titulo"/></td>
          <td bgcolor="#00ff00">
            <xsl:value-of select="autor"/></td>
            <td bgcolor="#00ff00">
              <xsl:value-of select="precio"/></td>
          </xsl:when>
        </xsl:choose>
      </tr>
    </xsl:for-each>
```



# EJERCICIOS 4- PROPUESTOS

3. Cambiar los colores para que si el coste es mayor que 25 euros, las filas sean de color rojo y verde si es menor que 25 euros.



The screenshot shows a web browser window titled "Mi biblioteca personal". It displays a table with four columns: ISBN, Título, Autor, and Precio. The rows are color-coded: red for prices greater than 25 euros and green for prices less than or equal to 25 euros.

ISBN	Título	Autor	Precio
978-1584503804	Physics-based animation	Kenny Erleben	50.06
978-84-9849-998-8	Principios de seguridad informática para usuarios	Carlos Garre	13.00
978-84-9849-703-8	Ejercicios complementarios de lógica digital	Alberto Sánchez	12.00
978-84-1212-222-3	XML fácilmente	Pepe Sánchez	8.00
971-84-1212-222-3	Schemas XML	Pepe Sánchez	82.00
971-84-1212-111-3	DTD's	Pepe Ríos	60.50
971-84-1111-222-3	RSS	IOsé Sánchez	10.50

# LAS PLANTILLAS EN XSL

## ◉ Elemento XSL:TEMPLATE:

- Hasta ahora hemos utilizado una sola plantilla: `<xsl:template>...`
- Así, con esa etiqueta, definíamos sobre qué parte del documento XML íbamos a actuar.
- En todos los ejemplos anteriores el atributo “match” referenciaba la raíz del documento XML.
- Vamos a usar varias plantillas para ver distintos formatos en distintas partes del documento.

# LAS PLANTILLAS EN XSL

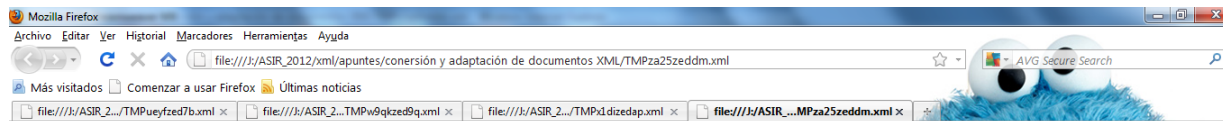
- Elemento XSL:TEMPLATE (ej23):

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0"
  <xsl:template match="/">
<html>
  <body>
    <h1>Ejemplo plantillas</h1>
    <xsl:apply-templates/>
  </body>
</html>
</xsl:template>
<xsl:template match="libreria">
  <h2> Mi biblioteca </h2>
  <table>
    <tr bgcolor="#007788">
      <th> Título</th> <th>Autor</th>
    </tr>
    <xsl:apply-templates select="libro"/>
  </table>
</xsl:template>
```

```
<xsl:template match="libro">
  <tr>
    <td><xsl:apply-templates
      select="titulo"/></td>
    <td><xsl:apply-templates
      select="autor"/></td>
  </tr>
</xsl:template>
<xsl:template match="titulo">
  <td
    bgcolor="#DDEEDD"><xsl:value-of
      select="."/></td>
</xsl:template>
<xsl:template match="autor">
  <td
    bgcolor="#AABBAA"><xsl:value-of
      select="."/></td>
</xsl:template>
</xsl:stylesheet>
```

# LAS PLANTILLAS EN XSL

## ◉ Elemento XSL:TEMPLATE:



### Ejemplo plantillas

#### Mi biblioteca

Título	Autor
Physics-based animation	Kenny Erleben
Principios de seguridad informática para usuarios	Carlos Garre
Ejercicios complementarios de lógica digital	Alberto Sánchez
XML fácilmente	Pepe Sánchez

Terminado **OJO:** probado con Firefox pues IE da problemas con plantillas

# LAS PLANTILLAS EN XSL

## ◉ Elemento XSL:TEMPLATE:

- En el ejemplo hemos visto el uso de 5 plantillas personalizadas para los elementos: “/”, “librería”, “libro”, “titulo” y “autor”.
- Cuando se llama desde el documento XML a esta hoja de estilos XSL, se intenta reconocer los patrones de las plantillas, sustituyendo en la salida los contenidos que se indiquen por cada plantilla.
- `<xsl:apply-templates/>`: Indica que se apliquen el resto de las plantillas definidas en cuanto se cumplan el patrón “match” de cada una.
- `<xsl:apply-templates select=“XXX”/>`: Indica que se aplique en ese momento, el contenido de la plantilla “XXX” definida en el documento XSL.
- `<xsl:value-of select “.”/>`: En el nodo XML actual de la plantilla actual, indica que se copie el valor que contenga ese nodo. Por ejemplo, si el nodo actual es de tipo “titulo”, insertará como resultado el título del libro actual.

# LAS PLANTILLAS EN XSL

## ◉ Elemento XSL:TEMPLATE:

- El proceso seguido cuando se recorre un documento XML desde una transformación XSL, es idéntico siempre:
  - Empezamos por el nodo raíz aplicando la plantilla que tengamos asignada. En el ejemplo anterior: “/” que genera un esqueleto de documento HTML válido.
  - `<xsl:apply-templates/>` de la plantilla “/” indica que se recorra todos y cada uno de los nodos definidos en el fichero XML aplicando las plantillas definidas.
  - Si en ese recorrido se encuentra el elemento “libreria”, se aplicará la plantilla del mismo nombre definida en el fichero XSL.
  - Es importante decir que en la plantilla librería se establecen las propiedades y etiquetas necesarias HTML para crear una tabla de datos.
  - `<xsl:apply-templates select=“libro”/>` indica que cuando se reconozca un nodo XML “libro” se aplique la plantilla definida para ello. Ídem para título y autor.

# LAS PLANTILLAS EN XSL

## ◉ Elemento XSL:TEMPLATE:

- Si en el recorrido del documento XML se encuentra sobre un nodo “titulo” o “autor”, se aplicará su plantilla correspondiente, con la salvedad que en esos nodos se requiere la extracción de la información para el documento final. Esa funcionalidad vendrá dada por el elemento “<xsl:value-of select=“.”/>”, copiando el valor de ese nodo y enviándolo al documento HTML final.
- Por tanto, el recorrido del documento XML va a ser un conjunto de llamadas a “procedimientos” (plantillas) que van integrando datos de salida en el documento final.

## EJERCICIOS 5- PROPUESTOS(FALTA)

1. Insertar el dato número de páginas en cada libro.
2. Añadir las columnas precio y número de páginas al final de la tabla de resultados.
3. Añadir ISBN al principio de la tabla.
4. Añadir al fichero XSL una plantilla nueva para cambiar el estilo del elemento número de páginas para que se muestre en rojo si el libro tiene más de 150 páginas solamente.