

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML (XSLT)

OBJETIVOS

- Conocer XSL como herramienta de transformación de documentos XML.
- Saber qué elementos básicos pueden utilizarse en la definición de estilos XSL.
- Utilización del plantillas para facilitar el proceso de conversión y transformación de documentos XML.
- Conocer qué operadores permiten discriminar los datos para aplicar estilos diferentes.
- Realizar ejercicios prácticos que afiancen los conceptos de la transformación a través de XSL.

1. INTRODUCCIÓN

- Tal como hemos visto, la tecnología XML permite separa de manera efectiva los datos a almacenar, la estructura en la que se organiza y la presentación de los mismos.
- Vamos a estudiar XSL(Extensible Stylesheet Language) que es una herramienta que nos va a permitir convertir y transformar los datos en el formato que deseemos.
- XSL es a XML, lo que a las hojas de estilo en cascada (CSS) a HTML. Nos permite decidir como qué datos ver, en qué orden visualizarlos, etc.

Veamos un ejemplo:

Partimos de un fichero XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE libreria SYSTEM "Ejemplo1.dtd">
<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>
```

El fichero XML validado a través de un DTD de nombre:

EJEMPLO1.DTD

```
<!ELEMENT libreria (libro+)>
<!ELEMENT libro (titulo,autor, editor, isbn, precio)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT editor (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
```

- **Vamos a crear una XSL para tener una visualización de los datos “más amigable”**

- **EJEMPLO1.XSL**

```
<?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>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>
```

- **Esta hoja de estilo define que por cada libro almacenado en la librería, se añade una entrada en una tabla HTML con los valores del título y el nombre del autor. El resultado es el siguiente**

➤ EJEMPLO1_V1.XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="Ejemplo1.xsl"?>
<!DOCTYPE libreria SYSTEM "Ejemplo1.dtd">
<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>
```

Si ejecutamos este fichero nos resulta la siguiente imagen:

Mi biblioteca

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

SI SE QUIERE UTILIZAR CORRECTAMENTE ESTA TECNOLOGÍA DE REPRESENTACIÓN DE DOCUMENTOS XML, SE DEBEN SEGUIR LOS SIGUIENTES PASOS:

- Tener bien definido el documento XML.
- Crear una hoja de estilo XSL bien formada.
- Vincular al documento XML la hoja de estilo XSL
- Los dos primeros puntos se asume al estar determinados por los DTDs o esquemas . Para vincular un XSL a un documento XML se añade al fichero XML lo siguiente (:
- `<?xml version="1.0" encoding="ISO-8859-1"?>`
- `<?xml-stylesheet type="text/xsl" href="Ejemplo1.xsl"?>`
- `<libreria>` (ejemplo nodo raíz) (ejemplo)
- ...
- `</libreria>`

2. TRANSFORMACIÓN DE DOCUMENTOS (XSL)

- XSL es un lenguaje que define un conjunto de reglas para transformar un XML en un resultado formateado, para ello, se necesita almacenar esas reglas dentro de un fichero llamado hoja de estilo XSL.

- El fichero XSL tiene que estar bien formado. Tiene que tener una declaración inicial que determina su contenido como XML y su codificación:

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

- Luego se indica la etiqueta que identifica al fichero XSL y el espacio de nombres en el que se basa. Tenemos dos opciones:

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

- La siguiente etiqueta es la que nos permite definir un elemento plantilla dentro de XSL

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

`...`

`</xsl:template>`

- Todo lo que vaya entre esas dos etiquetas será lo que permite generar la salida formateada. El atributo match indica sobre qué parte del documento XML se quiere actuar. En esta caso sobre la raíz del documento XML.

3. ELEMENTOS BÁSICOS

1. XSL:FOR-EACH.
2. XSL:VALUE-OF
3. XSL:SORT

3.1 XSL: FOR-EACH

- Se utiliza esta sentencia, para recorrer todos y cada uno de los elementos del fichero XML. La siguiente etiqueta indica que se recorran todo el conjunto de elementos XML que sean libros:

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

3.2 XSL: VALUE-OF

- La etiqueta que extrae la información de ese elemento XML seleccionado es la siguiente:

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

- Esta es la manera más sencilla de realizar un listado completo, formateado y con estilos adecuados, de todos los libros almacenados en la biblioteca.
- Si queremos mostrar el valor de un atributo, por ejemplo el código del título, pondremos:

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

- Si lo que queremos es que aparezca una imagen tendremos que hacer lo siguiente:

- En el xml tendremos una etiqueta llamada foto donde indicamos el nombre del fichero. El fichero de la imagen se guardará en la misma carpeta que el xml. La extensión del fichero será jpg ó png

- <foto>leon.png </foto>

➤ En el xsl haremos lo siguiente:

- <td>
 -
 - <xsl:attribute name="src">
 - <xsl:value-of select="foto"/>
 - </xsl:attribute>
 -
- </td>

3.3 XSL: SORT

➤ Para poder mostrar los datos de manera ordenada. Si por ejemplo nos interesa ordenar los libros por título, tras la etiqueta "for-each" incluimos:

<xsl:sort select="titulo"/>

EJEMPLO2.XLS

```
<?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>Mi biblioteca </h1>
      <table>
        <tr bgcolor="#887788">
          <th>Título</th>
          <th>Autor</th>
        </tr>
        <xsl:for-each select="libreria/libro">
          <b><xsl:sort select="titulo"/></b>
          <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>
```

EJEMPLO1_V2.XLM

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="Ejemplo2.xsl"?>
<!DOCTYPE libreria SYSTEM "Ejemplo1.dtd">
<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>
```

Si ejecutamos este fichero nos resulta la siguiente imagen:

Mi biblioteca

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

- Si quisiéramos ordenar por autor:

<xsl:sort select="autor"/>

EJEMPLO3.XLS

```
<?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>Mi biblioteca </h1>
      <table>
        <tr bgcolor="#887788">
          <th>Título</th>
          <th>Autor</th>
        </tr>
        <xsl:for-each select="libreria/libro">
          <b><xsl:sort select="autor"/></b>
          <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>
```

EJEMPLO1_V3.XLM

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="Ejemplo3.xsl"?>
<!DOCTYPE libreria SYSTEM "Ejemplo1.dtd">
<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>
```

Si ejecutamos este fichero nos resulta la siguiente imagen:

Mi biblioteca

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

➤ Si queremos filtrar, y que solo se muestre los libros del autor Carlos Garre, la sentencia sería:

```
<xsl:for-each select="libreria/libro[autor='Carlos Garre']">
```

EJEMPLO4.XLS

```
<?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>Mi biblioteca </h1>
<table>
  <tr bgcolor="#887788">
    <th>Título</th>
    <th>Autor</th>
  </tr>
  <xsl:for-each select="libreria/libro[autor='Carlos Garre']">
    <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>
```

EJEMPLO1_V4.XLM

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="Ejemplo4.xsl"?>
<!DOCTYPE libreria SYSTEM "Ejemplo1.dtd">
<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>
```

Si ejecutamos este fichero nos resulta la siguiente imagen:

Mi biblioteca

| Título | Autor |
|---|--------------|
| Principios de seguridad informática para usuarios | Carlos Garre |

➤ Si queremos filtrar, y que muestre todos los libros en los que el autor no es Carlos Garre, la sentencia sería:

```
<xsl:for-each select="libreria/libro[autor!='Carlos Garre']">
```

EJEMPLO5.XLS

```
<?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>Mi biblioteca </h1>
            <table>
                <tr bgcolor="#887788">
                    <th>Título</th>
                    <th>Autor</th>
                </tr>
                <xsl:for-each
select="libreria/libro[autor!='Carlos Garre']">
                    <xsl:sort select="autor"/>
                    <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>
EJEMPLO1_V5.XLM
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="Ejemplo5.xsl"?>
<!DOCTYPE libreria SYSTEM "Ejemplo1.dtd">
<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>
```

Si ejecutamos este fichero nos resulta la siguiente imagen:

Mi biblioteca

| Título | Autor |
|--|-----------------|
| Ejercicios complementarios de lógica digital | Alberto Sánchez |
| Physics-based animation | Kenny Erleben |

4. OPERADORES EN XSL

1. Elemento XSL:IF
2. Elemento XSL:CHOOSE

4.1 XSL:IF

- Los operadores lógicos que se pueden utilizar para cambiar el patrón de búsqueda o filtro son los siguientes:

Operador de igualdad (=): “=”
Operador de desigualdad (≠): “!=”
Operador menor que (<): “<”
Operador mayor que (>): “>”

- Con `<xsl:if>` podemos realizar condiciones más complejas. La sintaxis de este nuevo elemento es:

```
<xsl:if test="expresion">  
....  
</xsl:if>
```

- Veamos un ejemplo: queremos mostrar todos aquellos libros, cuyo coste sea superior a 12 euros. La sentencia sería:

```
<xsl:for-each select="libreria/libro">  
  <xsl:if test="precio&gt; 12">  
    ...  
  </xsl:if>  
</xsl:for-each>
```

EJEMPLO6.XSL

```
<?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>Mi biblioteca </h1>
      <table>
        <tr bgcolor="#887788">
          <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="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>
```

EJEMPLO1_V6.XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="Ejemplo6.xsl"?>
<!DOCTYPE libreria SYSTEM "Ejemplo1.dtd">
<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>
```

Si ejecutamos este fichero nos resulta la siguiente imagen:

Mi biblioteca

| Título | Autor | Precio |
|---|---------------|--------|
| Physics-based animation | Kenny Erleben | 50.06 |
| Principios de seguridad informática para usuarios | Carlos Garre | 13.00 |

4.2 XSL:CHOOSE

➤ El elemento `<xsl:choose>` es muy similar a la instrucción `switch` de C/C++. Se puede establecer tantas expresiones condicionales como se quieran mediante los elementos `<xsl:when>`. Si se quiere establecer una condición por defecto el elemento a utilizar es `<xsl:otherwise>`.

La sintaxis sería:

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

➤ En el siguiente ejemplo, si el libro cuesta menos de 12,5€ se va a mostrar en color rojo todos sus datos, si es mayor de 25,50€ será en color verde, y si no cumple ninguna de las anteriores, en color azul.

EJEMPLO7.XSL

```
<?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>Mi bibliteca </h1>
      <table>
        <tr bgcolor="#887788">
          <th>Título</th>
          <th>Autor</th>
```

```

        <th>Precio</th>
    </tr>
    <xsl:for-each select="libreria/libro">
    <tr>
        <xsl:choose>
            <xsl:when test="precio< 12.50">
    <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> 25.50">
    <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:otherwise>
    <td bgcolor="#0000ff"> <xsl:value-of select="titulo"/> </td>
    <td bgcolor="#0000ff"> <xsl:value-of select="autor"/> </td>
    <td bgcolor="#0000ff"> <xsl:value-of select="precio"/> </td>
            </xsl:otherwise>
        </xsl:choose>
    </tr>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

EJEMPLO1_V7.XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="Ejemplo7.xsl"?>
<!DOCTYPE libreria SYSTEM "Ejemplo1.dtd">
<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>
```

Si ejecutamos este fichero nos resulta la siguiente imagen:

Mi biblioteca

| Título | Autor | Precio |
|---|-----------------|--------|
| Physics-based animation | Kenny Erleben | 50.06 |
| Principios de seguridad informática para usuarios | Carlos Garre | 13.00 |
| Ejercicios complementarios de lógica digital | Alberto Sánchez | 12.00 |

5. PLANTILLAS

5.1 Elemento XSL:TEMPLATE

➤ Las plantillas, permiten definir sobre qué parte del documento XML se quiere actuar. Hasta ahora solo hemos utilizado una plantilla.

En todos nuestros ejemplo, el atributo match referenciaba siempre la raíz del documento XML, no estableciendo distinción entre otras ramas o elementos que lo componen.

Vamos a hacer un cambio en el fichero XML para que, usando distintas plantillas, podamos ver el nombre del autor en un gris claro y el título en otro más oscuro.

EJEMPLO8.XSL

```
<?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="#887788">
        <th>Título</th>
        <th>Autor</th>
      </tr>
      <xsl:apply-templates select="libro"/>
    </table>
  </xsl:template>
  <xsl:template match="libro">
    <tr>
      <xsl:apply-templates select="titulo"/>
      <xsl:apply-templates select="autor"/>
    </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>
```

EJEMPLO1_V8.XLM

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="Ejemplo8.xsl"?>
<!DOCTYPE libreria SYSTEM "Ejemplo1.dtd">
<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>
```

- En este ejemplo se ha creado cinco plantillas personalizadas para los elementos “/”, “librería”, “libro”, “titulo”, y “autor”.
- En este ejemplo se ha utilizado unos elementos que no habíamos visto anteriormente y que son:

<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=“XX”/>: Indica que se aplique en ese momento, el contenido de la plantilla “XX” 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.

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 |