# Caso Práctico 1 – Tabla de Discos

## 1. Introducción

En este caso práctico se trabaja la conversión y adaptación de un documento XML mediante el uso del lenguaje XSLT (Extensible Stylesheet Language for Transformations), que permite transformar datos estructurados en un formato XML hacia otros formatos, como HTML, de forma automática. Esto es muy útil para mostrar información de manera visual y comprensible, directamente en un navegador web.

Este tipo de transformación separa los **datos** del **formato de presentación**, lo que permite mantener la información central en XML y mostrarla de distintas formas según las necesidades (tablas, listas, informes, etc.).

Se utilizan las siguientes tecnologías:

- XML: para almacenar los datos estructurados.
- XSLT: para transformar los datos XML en un documento HTML.
- XPath: para seleccionar y ordenar los elementos del documento XML.

## 2. Objetivo del ejercicio y requisitos funcionales

Modificar una hoja de estilos XSL para que:

- Se muestre en una tabla HTML con tres columnas:
  - Nombre del disco y año (entre paréntesis).
  - Intérprete.
  - Precio.
- Se ordene por año descendente (del más reciente al más antiguo).

## 3. Código XML de entrada (cdcatalog.xml)

El documento XML contiene un listado de discos. Cada disco está representado por un nodo <cd> que incluye título, artista, precio y año. Este será el documento de origen que transformaremos mediante XSLT.

```
<catalog>
   <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <price>10.90</price>
    <year>1985</year>
   </cd>
   <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <price>9.90</price>
```

```
    <year>1988</year>
  </cd>
</catalog>
```

# 4. Código XSL modificado (cdcatalog.xsl)

A continuación se muestra la hoja de estilo XSL modificada que transforma el XML anterior en una tabla HTML. Se ha adaptado para cumplir con los requisitos especificados: mostrar el título junto al año, el artista, y el precio, y ordenar los discos por año de forma descendente.

```
<?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>Catálogo de Discos</title>
    </head>
    <body>
         <h2>Catálogo de Discos</h2>
         <table border="1">
         <tr bgcolor="#9acd32">
              <th>Disco (Año)</th>
             <th>Intérprete</th>
              <th>Precio</th>
         </tr>
         <xsl:for-each select="catalog/cd">
              <xsl:sort select="year"
data-type="number" order="descending"/>
              <tr>
              <td>
              <xsl:value-of select="title"/>
(<xsl:value-of select="year"/>)
              </td>
              <td>
              <xsl:value-of select="artist"/>
              </td>
              <td>
              <xsl:value-of select="price"/>
              </td>
              </tr>
         </xsl:for-each>
         </table>
    </body>
    </html>
   </xsl:template>
 </xsl:stylesheet>
```

## 4.1 Explicación detallada del XSL modificado

A continuación se explican paso a paso los cambios realizados en la hoja de estilos cdcatalog.xsl para cumplir con los requisitos del caso práctico:

### a) Añadir una tabla con tres columnas

Se crea una tabla HTML con la etiqueta <table> y se definen tres columnas mediante <th>. Las cabeceras indican el contenido esperado: el título del disco (incluyendo el año entre paréntesis), el intérprete y el precio. Se incluye además un color de fondo con el atributo bgcolor="#9acd32" para mejorar la visualización de la tabla.

Cada fila de la tabla corresponde a un disco. Las filas se generan dentro del bucle xsl:for-each, que recorre cada nodo <cd> dentro del nodo raíz <catalog>.

```
<table border="1">
  <tr bgcolor="#9acd32">
    <th>Disco (Año)</th>
    <th>Intérprete</th>
    <th>Precio</th>
  </tr>
```

Esto crea una tabla de tres columnas: una para el título y año, otra para el artista y otra para el precio. El color de fondo se aplica para mejorar la presentación visual.

### b) Ordenación por año descendente

Uno de los requisitos era que los discos se ordenaran por año, del más reciente al más antiguo. Para ello se usa la instrucción:

```
<xsl:sort select="year" data-type="number"
order="descending"/>
```

La instrucción xsl:sort permite ordenar los resultados del bucle. En este caso:

- select="year" especifica que se utilizará el nodo <year> como base para la ordenación.
- data-type="number" indica que se deben tratar como valores numéricos y no como texto (para evitar que 1988 aparezca antes que 2002 por error).
- order="descending" organiza los resultados del más reciente al más antiguo.

Esto asegura que los discos se muestren en la tabla en orden cronológico inverso.

### c) Formato de la primera columna: "Título (Año)"

En lugar de mostrar solo el título, se requiere mostrar también el año entre paréntesis. Para ello, se combinan dos expresiones xsl:value-of:

```
<td>
  <xsl:value-of select="title"/> (<xsl:value-of
select="year"/>)
</td>
```

Para mostrar el nombre del disco seguido del año, se usan dos llamadas a xsl:value-of, una para extraer el texto del nodo <title> y otra para el nodo <year>. Estas instrucciones se ejecutan en el contexto del nodo <cd>, que es el elemento actual durante la iteración.

## d) Segunda y tercera columna

Se utilizan directamente los nodos <artist> y <price> para mostrar el intérprete y el precio, respectivamente:

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

Aquí no es necesario hacer transformaciones complejas, simplemente se muestra el contenido de los nodos <artist> y <price> mediante xsl:value-of, tal como están en el XML. Estas instrucciones también se ejecutan dentro del bucle y en el contexto del nodo actual <cd>.

## e) Generación de la estructura HTML completa

Toda la estructura de salida se encierra dentro de una plantilla que se activa al hacer match="/", lo cual significa que se aplica cuando se encuentra el nodo raíz del documento XML. Este es el punto de partida habitual en transformaciones XSLT. Desde ahí, se aplican las transformaciones necesarias al resto de nodos.

```
<xsl:template match="/">
  <html>
    <head><title>Catálogo de Discos</title></head>
    <body>
      <!-- contenido de la tabla aquí -->
    </body>
  </html>
</xsl:template>
```


# 5. Conclusión

Este ejercicio ha servido para practicar el uso de hojas de estilo XSLT, la transformación de documentos XML en HTML, y la aplicación de ordenación con XPath. Se ha demostrado la utilidad de separar la lógica de presentación de los datos estructurados, así como la potencia que ofrece XSLT para transformar documentos en diferentes formatos sin necesidad de reescribir el contenido original. Este tipo de transformación es muy útil en entornos web y empresariales donde la información debe presentarse de formas distintas según el usuario o el dispositivo.

# 6. Recursos utilizados

- Contenidos de la Unidad Didáctica 5.
- Editor online de W3Schools para pruebas.
- Udemy
- Para ver la imágen de captura bien: https://drive.google.com/file/d/1BOMaWPzAEJWsxnI3bTz_RMpk4IVRAj3E/view?usp=drive_link