

LENGUAJE DE MARCAS

CASO PRACTICO IUD5



ALUMNO CESUR

24/25

Alejandro Muñoz de la Sierra

PROFESOR

José Carlos Villar Herrera

INTRODUCCION

En este caso práctico vamos a modificar una hoja de estilos XSL para cambiar y, por así decirlo, mejorar la forma en que se muestra un archivo XML que contiene un catálogo de discos. La idea es, en esencia, entender mejor cómo la conversión de documentos XML facilita el intercambio de información en sistemas digitales, algo que resulta crucial en la mayoría de las situaciones.

Una tienda de discos nos ha pedido trabajar sobre su listado de álbumes, el cual está almacenado en formato XML, y aunque ya usan una hoja XSL para transformar esos datos en una tabla HTML, la forma actual de presentar la información no permite una consulta ágil; por ello, se vuelve necesario darle un nuevo aire a la presentación.

Para ello, se plantea modificar el archivo XSL de la siguiente manera, de modo algo inesperado:

- Se rediseña la tabla para dividirla en tres columnas básicas: una que muestre el título del disco (con el año de publicación indicado entre paréntesis), otra dedicada al nombre del intérprete, y una tercera que exponga el precio del álbum en dólares.
- Se reordenen los registros, organizando los discos por año de lanzamiento en orden descendente, de forma que, en la mayoría de los casos, los más recientes aparezcan primero.

Con este proceso se ponen en juego conocimientos fundamentales sobre XSLT (Extensible Stylesheet Language Transformations), una tecnología clave para manipular y convertir datos XML a diferentes formatos. Además, se explorarán, de forma casi conversacional, las ventajas de usar hojas de estilo para transformar información, analizando tanto la sintaxis como las herramientas empleadas en este tipo de procesamiento.

Finalmente, se documentará todo el procedimiento de implementación y se evaluarán cuidadosamente los cambios realizados, garantizando que la presentación final resulte clara, precisa y funcional para el usuario final.

MODIFICACIÓN CORRECTA DEL ARCHIVO XSL Y ORDENACIÓN DE DATOS EN LA TABLA

Partimos del siguiente **código XML**:

Ponemos sólo un ejemplo de la estructura que se irá repitiendo para no copiar todos los datos, pero sí para ver claro lo que necesitaremos.

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  .
  .
  .
</catalog>
```

Y ahora del siguiente **código XSLT**:

```
<?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>
<h2>My CD Collection</h2>
<table border="1">
<tr bgcolor="#9acd32">
<th style="text-align:left">Title</th>
<th style="text-align:left">Artist</th>
</tr>
<xsl:for-each select="catalog/cd">
<tr>
<td><xsl:value-of select="title"/></td>
<td><xsl:value-of select="artist"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Ahora teniendo estas dos estructuras procedemos a hacerle los cambios necesarios al XSLT para que el resultado final concuerde con lo exigido en el caso práctico.

Cambios a ejecutar en el XSLT:

Se agregan tres columnas extras que aportan mayor claridad:

La primera mezcla el nombre del disco y su año entre paréntesis. mientras que la segunda te indica quién lo interpreta, y la tercera revela el precio en dólares. Además, se han reordenado los discos para que el lanzamiento más reciente aparezca primero y se vayan mostrando los más antiguos conforme avanzas.

Ahora queda en claro lo que se modificó. Por un lado, se introduce una columna para el precio del disco. Es decir, se ha insertado una nueva sección en el encabezado de la tabla, algo así como:

<th>Price (\$)</th>;

de este modo, puedes visualizar sin esfuerzo el costo de cada disco.

Por otro lado, para ordenar los discos por año se implementó la etiqueta

<xsl:sort> dentro del <xsl:for-each>.

Con este recurso, se dispone que los discos se ordenen en forma descendente, es decir, de los más nuevos a los más viejos, mediante algo similar a:

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

Esto asegura que los lanzamientos más recientes se muestran primero, facilitando la búsqueda.

Finalmente, se decidió combinar en la primera columna el título del disco con su año, usando un sencillo concat(). En el código queda como:

<xsl:value-of select="concat(title, ' (', year, ')')"/>.

Así, en vez de separar el título y el año en columnas diferentes, se presentan juntos – por ejemplo, verás 'Thriller (1982)' o 'Back in Black (1980)'.

Código XSLT con las modificaciones aplicadas:

```
<?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>
<h2>My CD Collection</h2>
<table border="1">
<tr bgcolor="#9acd32">
<th style="text-align:left">Title (Year)</th>
<th style="text-align:left">Artist</th>
<th style="text-align:left">Price ($)</th>
</tr>

<!-- Ordenar por año de manera descendente -->
<xsl:for-each select="catalog/cd">
<xsl:sort select="year" data-type="number" order="descending"/>
<tr>
<td><xsl:value-of select="concat(title, ' (', 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>
```

PRESENTACIÓN DE DATOS EN EL HTML RESULTANTE

Tabla Original:

My CD Collection	
Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli
When a man loves a woman	Percy Sledge
Black angel	Savage Rose
1999 Grammy Nominees	Many
For the good times	Kenny Rogers
Big Willie style	Will Smith
Tupelo Honey	Van Morrison
Soulsville	Jorn Hoel
The very best of	Cat Stevens
Stop	Sam Brown
Bridge of Spies	T`Pau
Private Dancer	Tina Turner
Midt om natten	Kim Larsen
Pavarotti Gala Concert	Luciano Pavarotti
The dock of the bay	Otis Redding
Picture book	Simply Red
Red	The Communards
Unchain my heart	Joe Cocker

Después de aplicar los cambios al XSLT, podemos probar el código en el editor de W3Schools y comprobar cómo se genera la tabla con los discos ordenados y mostrados de forma más integrada, este enfoque mejora la presentación de la información y ofrece una experiencia de usuario más directa y comprensible.

My CD Collection

Title (Year)	Artist	Price (\$)
1999 Grammy Nominees (1999)	Many	10.20
One night only (1998)	Bee Gees	10.90
Eros (1997)	Eros Ramazzotti	9.90
Big Willie style (1997)	Will Smith	9.90
Romanza (1996)	Andrea Bocelli	10.80
Soulsville (1996)	Jorn Hoel	7.90
Black angel (1995)	Savage Rose	10.90
For the good times (1995)	Kenny Rogers	8.70
Pavarotti Gala Concert (1991)	Luciano Pavarotti	9.90
Still got the blues (1990)	Gary Moore	10.20
Maggie May (1990)	Rod Stewart	8.50
The very best of (1990)	Cat Stevens	8.90
Hide your heart (1988)	Bonnie Tyler	9.90
Stop (1988)	Sam Brown	8.90
When a man loves a woman (1987)	Percy Sledge	8.70
Bridge of Spies (1987)	T'Pau	7.90
Red (1987)	The Communards	7.80
Unchain my heart (1987)	Joe Cocker	8.20
Empire Burlesque (1985)	Bob Dylan	10.90
Picture book (1985)	Simply Red	7.20
Private Dancer (1983)	Tina Turner	8.90
Midt om natten (1983)	Kim Larsen	7.80
Greatest Hits (1982)	Dolly Parton	9.90
Sylvias Mother (1973)	Dr.Hook	8.10
Tupelo Honey (1971)	Van Morrison	8.20
The dock of the bay (1968)	Otis Redding	7.90

CONCLUSIONES

Durante este proyecto práctico se evidenció que XSLT resulta, en muchas ocasiones, una herramienta sorprendentemente versátil para transformar y presentar datos que provienen de archivos XML. Al ajustar la hoja de estilos, conseguimos darle un giro inesperado al catálogo de discos, lo que hace que la información se muestre de una forma mucho más limpia y, por cierto, accesible.

Entre las mejoras aplicadas destacan:

Se añadió el año de publicación justo al lado del título del disco, facilitando –en la mayoría de los casos– identificar la cronología de los álbumes.

El intérprete se desplazó a una columna independiente, lo que, sin duda, mejora la legibilidad de los datos.

Los discos se ordenaron en forma descendente según el año de lanzamiento, permitiendo una consulta más práctica y priorizando, de alguna manera, los discos más recientes.

Este ejercicio nos recordó lo importante que es trabajar con estructuras bien definidas y aplicar transformaciones adecuadas según lo que se busca lograr; además, nos permitió explorar un poco más el uso de ciertas etiquetas clave en XSLT, por ejemplo:

`<xsl:sort>` se usa para ordenar dinámicamente la información en tablas.

`<xsl:value-of>` se emplea para extraer datos específicos de un XML y mostrarlos en la salida HTML (algo que, generalmente, es esencial).

Por último, se constató que combinar XSLT con XML es una solución eficiente para estructurar y formatear datos sin tener que recurrir a lenguajes de programación excesivamente complejos. Este enfoque es especialmente útil en escenarios donde se necesita convertir la información a distintos formatos sin tocar la fuente original. La experiencia adquirida en este caso práctico, definitivamente, nos servirá para desarrollar futuras aplicaciones y sistemas que requieran transformar datos de manera flexible y escalable.

REFERENCIAS

https://www.w3schools.com/xml/xsl_intro.asp

https://www.w3schools.com/xml/xsl_transformation.asp

https://www.w3schools.com/xml/xml_xslt.asp

<https://developer.mozilla.org/en-US/docs/Glossary/XSLT>

<https://www.eniun.com/tutorial-xslt/>

<https://www.youtube.com/watch?v=IBCOPz-v6E4>

<https://www.youtube.com/watch?v=WggVR4YI5oI>