

Tarea 5.

Para la realización de esta tarea se ha hecho uso del programa Visual Studio Code.

En primer lugar se ha procedido a crear una carpeta en el escritorio con el nombre de Tarea 5 donde se ha alojado el archivo XML del ejercicio. Seguidamente se ha procedido a crear un archivo XSL en la misma carpeta donde se procederá a introducir todas las instrucciones para realizar la transformación XSLT del anterior archivo bajo las especificaciones del ejercicio.

Antes de nada, se ha procedido a enlazar este archivo XSL con el archivo XML mediante la siguiente línea:

```
<?xml-stylesheet href="artistasej.xsl" type="text/xsl"?>
```

En el encabezado del archivo XLS se han situado las siguientes líneas para declarar el espacio de nombres XSL:

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

Tras esto, lo siguiente que hacemos es indicar el tipo de salida que queremos que nos de, en este caso será un HTML, para ello usaremos el comando output con el method HTML.

```
<xsl:output method="html"/>
```

Seguidamente se ha realizado un template que permite controlar el formato de salida que se aplica a ciertos datos de entrada con el atributo match, que se utiliza para seleccionar los nodos del árbol de entrada.

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

A continuación, se comienza a construir el archivo HTML en el interior con su head y body. Dentro de la cabecera incluimos el título y la codificación de caracteres:

```
<head>  
  <meta charset="UTF-8" />  
  <title>Tabla de Artistas</title>  
</head>
```

Dentro del cuerpo será donde situaremos la tabla con los datos de los artistas atendiendo a los requisitos de la tarea.

Como fila de cabecera de la tabla dejaremos la misma distribución que el ejemplo del ejercicio, con las etiquetas correspondientes a cada columna:

```
<table>
  <tr>
    <th>Código</th>
    <th>Nombre</th>
    <th>Año de nacimiento</th>
    <th>Año de fallecimiento</th>
    <th>País</th>
    <th>Página web</th>
  </tr>
  .
  .
  .
```

Tras esto, se ha utilizado la construcción “xsl:for-each” para iterar a través de todos los elementos “artista” que se encuentran dentro del nodo raíz “artistas”. Para que de esta forma se recorran todos los elementos “artista” con sus nodos hijos:

```
<xsl:for-each select="artistas/artista">
  .
  .
  .
</xsl:for-each>
```

Como describe el ejercicio, la tabla mostrará en cada fila un artista con sus atributos correspondientes. En el caso del atributo fallecimiento, interesa que muestre el valor “Desconocido” en aquellos artistas que no tengan una descripción para este atributo. Esto se ha hecho utilizando el elemento choose para seleccionar el atributo fallecimiento de los artistas mediante el elemento when que contienen una prueba (test) Si la prueba es verdadera, entonces el valor del elemento "xsl:value-of" dentro de ese elemento "xsl:when" se muestra en el resultado de la transformación, en este caso se mostraría el valor del año de fallecimiento. Si no hay ningún elemento "xsl:when" cuya prueba sea verdadera, se usa el elemento "xsl:otherwise" para mostrar el valor “Desconocido”:

```
<td>
  <xsl:choose>
    <xsl:when test="fallecimiento">
      <xsl:value-of select="fallecimiento"/>
    </xsl:when>
    <xsl:otherwise>Desconocido</xsl:otherwise>
  </xsl:choose>
</td>
```

Para mostrar un vínculo a la dirección adecuada, con el texto "Saber más" se utiliza la construcción "xsl:attribute" para agregar un atributo "href" al elemento HTML "a", que contiene la URL de la ficha completa. El valor de este atributo se especifica usando la función XPath "xsl:value-of", que recupera el valor del elemento XML "fichaCompleta". El texto "Saber más" se usa como el texto visible del enlace.

```
<a>
  <xsl:attribute name="href">
    <xsl:value-of select="fichaCompleta" />
  </xsl:attribute>
  Saber más
</a>
```

Para mostrar los artistas que han nacido después del año 1500 se ha utilizado el elemento "xsl:if". La construcción "xsl:if" contiene una prueba (test) Si la prueba es verdadera, los elementos XML que se encuentran dentro del elemento "xsl:if" se procesan y se muestran en el resultado de la transformación. Si la prueba es falsa, se omite el procesamiento de los elementos y no se muestran en el resultado de la transformación. La prueba es "nacimiento > 1500". Esto significa que se evalúa si el valor de "nacimiento" es mayor que 1500. Si es verdadero, los elementos dentro del "xsl:if" se procesan. Si es falso, se omiten.

```
<xsl:if test="nacimiento > 1500">
```

Por último, para ordenar los artistas según su año de nacimiento por orden creciente (del más bajo al más alto), se ha utilizado la construcción "xsl:sort" dentro del bucle "xsl:for-each". El criterio de ordenación es el valor del elemento "nacimiento". La expresión "select" especifica cual es el valor por el que se va a ordenar, y el atributo "order" especifica que el orden debe ser ascendente (de menor a mayor), en este caso como queremos ordenarlo de menor a mayor, utilizamos "ascending" (ascendiente).

```
<xsl:sort select="nacimiento" order="ascending"></xsl:sort>
```

Para generar el HTML resultante de la transformación, abrimos el documento XML resultante con un navegador utilizando una IP local, damos a inspeccionar y copiamos el elemento HTML. Seguidamente lo copiamos en un bloc de notas y lo guardamos con formato HTML.