

**Lenguajes de marcas
y sistemas de gestión de información.**

TAREA 05



JESÚS ROMERO PÉREZ

Ejercicio 1

La tarea consiste en escribir una transformación XSLT que a partir de un fichero XML con datos de artistas genere una página web con una tabla con los datos de los artistas.

El elemento raíz del fichero es artistas. Dentro de este elemento hay uno o más elementos artista, como se puede ver en el ejemplo. Todos los elementos y atributos son obligatorios, salvo el elemento fallecimiento.

Ejemplo
artistasej.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<artistas>
  <artista cod="a101">
    <nombreCompleto>Diego Velázquez</nombreCompleto>
    <nacimiento>1599</nacimiento>
    <fallecimiento>1660</fallecimiento>
    <pais>España</pais>
    <fichaCompleta>https://es.wikipedia.org/wiki/Diego_Vel%C3%A1zquez</fichaCompleta>
  </artista>
  <artista cod="a102">
    <nombreCompleto>Michelangelo Caravaggio</nombreCompleto>
    <nacimiento>1571</nacimiento>
    <pais>Italia</pais>
    <fichaCompleta>https://es.wikipedia.org/wiki/Caravaggio</fichaCompleta>
  </artista>
  <artista cod="a103">
    <nombreCompleto>Herrada de Landsberg</nombreCompleto>
    <nacimiento>1125</nacimiento>
    <fallecimiento>1195</fallecimiento>
    <pais>Alsacia</pais>
    <fichaCompleta> https://es.wikipedia.org/wiki/Herrada_de_Landsberg</fichaCompleta>
  </artista>
  <artista cod="a104">
    <nombreCompleto>Francisco de Goya</nombreCompleto>
    <nacimiento>1746</nacimiento>
    <fallecimiento>1828</fallecimiento>
    <pais>España</pais>
    <fichaCompleta>https://es.wikipedia.org/wiki/Francisco_de_Goya</fichaCompleta>
  </artista>
</artistas>
```

Al aplicar la transformación, el resultado debe ser una página web válida. Es decir, tiene que tener todos los elementos básicos de una página web: título, codificación de caracteres, elementos, html, body y head.

Lenguajes de marcas y sistemas de gestión de información.

Dentro del cuerpo de la página habrá una tabla, obtenida a partir de los datos del fichero artistas.xml.

La tabla tendrá una fila por cada artista mostrado, y una primera fila como cabecera, con los nombres: Código, Nombre, Año de nacimiento, Año de fallecimiento, País y Página web.

Como se puede ver en el fichero, el año de fallecimiento no está disponible para todos los artistas. En ese caso, en la columna correspondiente se mostrará el texto “Desconocido”.

En la última columna, el valor se obtiene del elemento fichaCompleta. Debe mostrarse un vínculo a la dirección adecuada, con el texto “Saber más”.

Sólo se mostrarán los artistas nacidos después de 1500.

Los artistas se mostrarán ordenados por año de nacimiento creciente. Es decir, en la primera fila, el que tenga un año de nacimiento anterior.

A continuación de la tabla, se mostrarán los siguientes recuentos (deben calcularse dentro del XSLT):

Número total de artistas nacidos después de 1500.

Número de artistas nacidos en España.

Número de artistas nacidos en el siglo XVI.

Ejemplo

Para el fichero artistasej.xml se obtendrá una página web con este aspecto:

Código	Nombre	Año de nacimiento	Año de fallecimiento	País	Página web
a102	Michelangelo Caravaggio	1571	Desconocido	Italia	Saber más
a101	Diego Velázquez	1599	1660	España	Saber más
a104	Francisco de Goya	1746	1828	España	Saber más

Número total de artistas nacidos después de 1500: 3

Número de artistas nacidos en España: 2

Número de artistas nacidos en el siglo XVI: 2

SOLUCIÓN:

Este ejercicio consistió en crear una transformación de un archivo XML que contiene información sobre artistas utilizando un archivo XSLT. Mi objetivo fue procesar y mostrar los datos del XML en un formato HTML de forma estructurada y ordenada.

Primero, el archivo XML contiene datos de artistas como el código, nombre completo, año de nacimiento, año de fallecimiento, país de origen y un enlace a su ficha completa. A partir de estos datos, utilicé XSLT para generar una página HTML.

En el archivo XSLT, usé `xsl:for-each` para recorrer los elementos `<artista>` y filtrar aquellos que nacieron después de 1500. Luego, ordené los artistas por su año de nacimiento usando `xsl:sort` para mostrarlos de manera ascendente. Además, creé una tabla HTML en la que se presentan estos artistas con la información correspondiente y un enlace para obtener más detalles.

Lenguajes de marcas y sistemas de gestión de información.

También incluí unos recuentos, como la cantidad de artistas nacidos después de 1500, los nacidos en España y los que nacieron en el siglo XVI. Esto se logró utilizando la función count() de XSLT.

Solución en Firefox

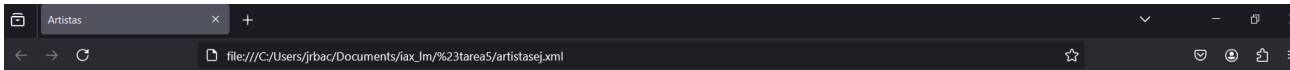


Tabla de Artistas

Código	Nombre	Año de nacimiento	Año de fallecimiento	País	Página web
a102	Michelangelo Caravaggio	1571	Desconocido	Italia	Saber más
a101	Diego Velázquez	1599	1660	España	Saber más
a104	Francisco de Goya	1746	1828	España	Saber más

Recuentos

Número total de artistas nacidos después de 1500: 3

Número de artistas nacidos en España: 2

Número de artistas nacidos en el siglo XVI: 2

Ejercicio 2

Partiendo del mismo documento XML del ejercicio 1, escribir en XPATH las siguientes consultas (se debe conseguir que la consulta, además de ser correcta, consiga la salida indicada en cada caso):

1. Código y nombre de los artistas del siglo XVI:

a101 - Diego Velázquez

a102 - Michelangelo Caravaggio

2. Año medio de la fecha de nacimiento de todos los artistas:

1510.25

3. Nombre y edad con la que murieron todos los artistas excepto el artista con código "a102" (Michelangelo Caravaggio):

Diego Velázquez murió a los 61 años.

Herrada de Landsberg murió a los 70 años.

Francisco de Goya murió a los 82 años.

Lenguajes de marcas y sistemas de gestión de información.

4. Nombre, país de nacimiento y edad de fallecimiento de los artistas con código distinto a "a102" que murieron con menos de 75 años:

Diego Velázquez nació en España y falleció con 61 años

Herrada de Landsberg nació en Alsacia y falleció con 70 años

SOLUCIÓN:

Consulta 1:

/artistas/artista[nacimiento >= 1501 and nacimiento <= 1600]/concat(@cod, ' - ', nombreCompleto)

Consulta 2:

sum(/artistas/artista/nacimiento) div count(/artistas/artista/nacimiento)

Consulta 3:

/artistas/artista[@cod != 'a102' and fallecimiento]/concat(nombreCompleto, ' murió a los ', fallecimiento - nacimiento, ' años.')

Consulta 4:

/artistas/artista[@cod != 'a102' and fallecimiento - nacimiento < 75]/concat(nombreCompleto, ' nació en ', pais, ' y falleció con ', fallecimiento - nacimiento, ' años.')

Lenguajes de marcas y sistemas de gestión de información.

Solución en BASEX:

The screenshot shows the BaseX 11.6 interface. The left pane displays the file system with 'BaseX' selected. The middle pane shows the query editor with the following XQuery:

```
1 /artistas/artista[nacimiento >= 1501 and nacimiento <= 1600]/concat(@cod, ' - ',  
2 nombreCompleto)
```

The right pane shows a tree view of the 'artistasej.xml' file. The bottom pane displays the results of the query:

```
2 Results, 55 b  
a101 - Diego Velázquez  
a102 - Michelangelo Caravaggio
```

The bottom right pane shows the query plan and compilation details:

Query: /artistas/artista[nacimiento >= 1501 and nacimiento <= 1600]/concat(@cod, ' - ', nombreCompleto)

Query Plan: <QueryPlan compiled="true" updating="false">
<DualMap type="xs:string">

The screenshot shows the BaseX 11.6 interface. The left pane displays the file system with 'BaseX' selected. The middle pane shows the query editor with the following XQuery:

```
1 sum(/artistas/artista/nacimiento) div count(/artistas/artista/nacimiento)  
2
```

The right pane shows a tree view of the 'artistasej.xml' file. The bottom pane displays the results of the query:

```
1 Result, 7 b  
1510.25
```

The bottom right pane shows the query plan and compilation details:

Query: sum(/artistas/artista/nacimiento) div count(/artistas/artista/nacimiento)

Query Plan: <QueryPlan compiled="true" updating="false">
<DbI type="xs:double" size="1">1510.25</DbI>
</QueryPlan>

Lenguajes de marcas y sistemas de gestión de información.

The screenshot shows the BaseX 11.6 interface with the following components:

- File Explorer:** Displays the file structure of the 'BaseX' directory, including files like 'BaseX.jar', 'CHANGELOG', 'LICENSE', 'readme.txt', and 'uninst.exe'.
- Editor:** Contains the XQuery code:

```
1 /artistas/artista[@cod != 'a102' and fallecimiento]/concat(nombreCompleto, '
2 murió a los ', fallecimiento - nacimiento, ' años.')
```
- Result:** Shows 3 results in a table format:

Nombre Completo	País	Nacimiento	Fallecimiento
Diego Velázquez	España	1599	1660
Herrada de Landsberg	Alsacia	1571	1660
Francisco de Goya	España	1746	1828
- Query Plan:** Displays the compiled query plan, including the predicate and the path expression.

The results section shows 3 Results, 126 b. The query plan is compiled with the following details:

- Query Plan compiled="true" updating="false"
- Query Plan type="xs:string"
- Query Plan path type="element(artista)" database="artistasej"
- Query Plan DBNode pre="0" type="document-node()" size="1" database="artistasej"
- Query Plan iterStep axis="child" test="artistas" type="element(artistas)" database="artistasej"