

# Xquery FLWOR: ejercicios 2

Partiendo del siguiente documento libros.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

- Mostrar el título y el autor de los libros del año 2005, y etiquetar cada uno de ellos con "lib2005".

```
for $libro in /bookstore/book
where $libro/year=2005
return <lib2005>{$libro/title,$libro/author}</lib2005>
```

Recorremos las tuplas de los libros con el for y las guardamos en la variable "libro".

Extraemos los libros del año 2005 con la cláusula where.

Escribimos en el return las etiquetas lib2005 y enmarcamos dentro los nodos de title y author.

- Mostrar los años de publicación. Etiquetar la salida con "publicacion".

```
for $year in /bookstore/book/year
return <publicacion>{$year}</publicacion>
```

Recorremos las tuplas de los años con el for y las guardamos en la variable "year".  
Escribimos en el return las etiquetas "publicacion" y enmarcamos dentro el nodo "year".

- Mostrar los libros ordenados primero por "category" y luego por "title" en una sola consulta.

```
for $libro in /bookstore/book
order by $libro/@category,$libro/title
return $libro
```

Recorremos las tuplas de los libros con el for y las guardamos en la variable "libro".  
Ordenamos la salida a partir del atributo categoría (se indica con la arroba) en primer lugar y el título en segundo lugar  
Escribimos en el return el nodo del libro completo.

- Mostrar cuántos libros hay, y etiquetarlo con "total".

```
<total>{ count (/bookstore/book) }</total>
```

Enmarcamos entre las etiquetas total la función de contar, donde contamos todas las tuplas de libros y por lo tanto, nos devolverá la cantidad de libros que existen.

- Mostrar los títulos de los libros y al final una etiqueta con el número total de libros.

```
let $total := count (/bookstore/book), $titulos := (
    for $libro in /bookstore/book/title
    return <titulo>{$libro/text()}</titulo>
)
return
<resultado>
    {$titulos}
    <total_libros>{$total}</total_libros>
</resultado>
```

Creamos dos variables y asignamos con el let.

La primera de ellas es “total”, donde almacenamos la cantidad de libros que existen (consultar ejercicio anterior), en la segunda variable ejecutamos otra consulta FLWOR donde recuperamos todos los títulos de los libros (podemos ejecutar todas las consultas que queramos de forma anidada como se hace en este ejercicio). Finalmente en el return enmarcamos entre etiquetas “resultado” todos los títulos de los libros y al final, la cantidad de todos ellos entre etiquetas “total\_libros”.

- Mostrar el precio mínimo de los libros.

```
<min>{ max(/bookstore/book/price) }</min>
```

Enmarcamos entre las etiquetas min la función de mínimo, donde nos calcula el mínimo precio de todos los libros existentes.

- Mostrar el precio máximo de los libros.

```
<max>{ max(/bookstore/book/price) }</max>
```

Enmarcamos entre las etiquetas max la función de máximo, donde nos calcula el máximo precio de todos los libros existentes.

- Mostrar el título del libro, su precio y su precio con el IVA incluido, cada uno con su propia etiqueta. Ordenarlos por precio con IVA.

```
for $libro in /bookstore/book
let $precio_iva := ($libro/price * 1.21)
order by $precio_iva
return
<libro>
  <titulo>{$libro/title/text()}</titulo>
  <precio>{$libro/price/text()} €</precio>
  <precio_iva>{$precio_iva} €</precio_iva>
</libro>
```

Recorremos las tuplas de los libros con el for y las guardamos en la variable “libro”.

Creamos una variable llamada “precio\_iva” y calculamos su valor, para ello ejecutamos el cálculo del precio multiplicado por 1.21 (21% de valor añadido).

Ordenamos por el “precio\_iva” que acabamos de calcular.

Escribimos en el return las etiquetas libro (nos devolverá una línea por cada libro del xml origen) y enmarcamos dentro el título del libro, su precio (con el carácter euro por ejemplo en su interior) y su precio con IVA que corresponde a la variable que se ejecuta por cada tupla encontrada y donde nos calcula su valor añadido.