

# Unidad 7 . Almacenamiento de información.

## XQuery - Bookstore Part I

1. Lista el título de los libros ordenados por título.

```
for $x in doc("books")/bookstore/book
order by $x/title
return $x/title/text()
```

### Resultado

2. Lista el título de los libros ordenados por título cuyo precio es superior a 30.

```
for $x in doc("books")/bookstore/book
where $x/price>30
order by $x/title
return $x/title/text()
```

### Resultado

3. Crea una página web (03.html) cuyo título de la web "UD7-01.03 Listado de libros. Nombre y apellidos" y que contenga una lista con viñetas con los títulos de los libros.

```
<html>
  <head>
    <title>Ud07.01.03 Listado de libros. Rebeca Sánchez</title>
    <meta charset="UTF-8"/>
  </head>
  <body>
    <h1>Listado de libros</h1>
    <ul>
      {for $x in doc("books")/bookstore/book
      where $x/price>30
      order by $x/title
      return <li>{$x/title/text()}</li>
      }
    </ul>
  </body>
</html>
```

### Resultado

4. Crea un documento XML(04.html), donde el elemento raíz es libros. Y si la categoría del libro es children el título aparece entre las etiquetas <children> y en caso contrario aparece con la etiqueta

<adult>.

```
<libros>
{
  for $x in doc('books')/bookstore/book
  return if ($x/@category="children")
    then <children>{$x/title/text()}</children>
    else <adult>{$x/title/text()}</adult>
}
</libros>
```

## Resultado

5. Crea una página web (05.html), que contenga todos los libros con su categoría, en la lista aparece distintos colores el título en función de la categoría.

```
<html>
  <head>
    <title>Ud07.01.05 Tabla y listado de libros. Rebeca Sánchez</title>
    <meta charset="UTF-8"/>
    <link href="css/05.css" type="text/css" rel="stylesheet"/>
  </head>
  <body>
    <h1>Bookstore</h1>
    <table>
      <tr>
        <th>Título</th>
        <th>Categoría</th>
      </tr>
      {
        for $x in doc('books')/bookstore/book
        order by $x/title
        return <tr><td>{$x/title/text()}</td><td>{data($x/@category)}</td>
      }
    </table>
    <ul>
      {for $x in doc("books")/bookstore/book
      order by $x/title
      return <li class="{ $x/@category }">{$x/title/text()}</li>
      }
    </ul>
  </body>
</html>
```

## Resultado

6. Lista de libros ordenados por categoría de forma descendiente, y como segundo criterio por título de forma descendiente. Mostrando el título con suprecio precio, donde el precio es mayor que 30 \$. El

resultado (06.txt) es un fichero de texto.

```
for $x in doc('books')/bookstore/book
where $x/price>30
order by $x/@category,$x/title/text()
return concat($x/title/text(),":",$x/price,"$")
```

### Resultado

7. Uso de at, for variable1 at variable2 expresiónXpath, siendo la segunda variable la posición de cada nodo. Genera el siguiente archivo XML (07.xml).

```
<books>
{
  for $x at $y in doc('books')/bookstore/book
  return <book>{concat($y," ",$x/title/text())}</book>
}
</books>
```

### Resultado

8. Uso de variables expresiones en un for \$tema in (lista de valores), \$y in (lista de valores). Para indicar la lista de valor de valorinicial to valorfinal, por ejemplo 1 to 100. Generar automáticamente la siguiente página web (08.html).

```
<html>
  <head>
    <title>Ud07.01.08 Listado de Prácticas. Rebeca Sánchez</title>
    <meta charset="UTF-8"/>
  </head>
  <body>
    <h1>Listado de prácticas</h1>
    <div>
      {
        for $tema in (1 to 7), $ejercicio in (1 to 3)
        return <div>
          <a href="{ $tema }/{ $ejercicio }.html">Tema{ $tema }
          Ejercicio{ $ejercicio}</a>
        </div>
      }
    </div>
  </body>
</html>
```

### Resultado

9. Mostrar el título del libro cuando tenga más de dos autores.

```
for $libro in doc('books')/bookstore/book
let $a := $libro/author
where count($a)>2
return $libro/title/text()
```

### Resultado

10. Trasformar los datos a un archivo XML con el siguiente contenido, donde el atributo autores indica el número de autores de cada libro.

### Resultado