

Realizar las sentencias Xquery del documento libros.xml

1. Listar el título de todos los libros.

```
//libro/titulo
```

2. Listar año y título de todos los libros, ordenados por el año.

```
for $ano in //libro
let $algo:=$ano/@año order by $algo
return
concat($algo,"-", $ano/titulo)
```

3. Listar los libros cuyo precio sea 65.95

```
for $a in //libro[precio=65.95]
return $a
```

4. Listar los libros publicados antes del año 2000

```
for $a in //libro[@año<2000]
return $a
```

5. Listar año y título de los libros publicados por Addison-Wesley después del año 1992.

```
for $a in //libro[editorial='Addison-Wesley']
let $b:=$a[@año>1992]
return
concat($b/@año,"-", $b/titulo)
```

6. Listar año y título de los libros que tienen más de un autor.

```
for $a in //libro
let $b:=$a/autor
where count($b)>1
return
concat($a/@año,"-", $a/titulo)
```

7. Listar año y título de los libros que no tienen autor.

```
for $a in //libro
let $b:=$a/autor
where count($b)<1
return
concat($a/@año,"-", $a/titulo)
```

8. Mostrar los apellidos de los autores que aparecen en el documento, sin repeticiones, ordenados alfabéticamente.

```
for $a in distinct-values(//autor/apellido)
order by $a
return $a
```

9. Por cada libro, listar agrupado en un elemento <result> su título y autores

```
for $a in //libro
return
<result>{$a/autor}, {$a/titulo}</result>
```

10. Por cada libro, obtener su título y el número de autores, agrupados en un elemento <libro>

```
for $a in //libro
return
<result>{count($a/autor)},{$a/titulo}</result>
```

11. Generar un documento html con un encabezado que diga “Listado de libros” seguido de una tabla de 3 columnas, en la que se muestre el título, editorial y precio de cada libro. La tabla deberá tener bordes y la primera fila tendrá un formato especial con los nombres de las columnas. Las columnas de título y editorial irán alineadas a la izquierda, y el precio a la derecha.

```
let $a:=//bib
return
<html>
<head><title>listado libros</title></head>
<body>
<table border="1">
<tr>
<td>titulo</td>
<td>editorial</td>
<td>precio</td></tr>
{for $b in $a/libro
return
<tr>
<td>{$b/titulo} </td>
<td>{$b/editorial} </td>
<td>{$b/precio} </td></tr>}
</table>
</body>
</html>
```

Sentencias DML en eXist en los documentos emp.xml y dept.xml

12. Inserta en el documento emp un nuevo empleado de código 9898 y de nombre Antonio el resto de nodos también serán obligatorios y se incluirán los valores que se deseen.

```
update insert <EMP_ROW>
  <EMP_NO>9898</EMP_NO>
  <APELLIDO>Antonio</APELLIDO>
  <OFICIO>patronsito</OFICIO>
  <DIR>000</DIR>
  <FECHA_ALT>190-12-17</FECHA_ALT>
  <SALARIO>100</SALARIO>
  <DEPT_NO>920</DEPT_NO>
</EMP_ROW> into //EMPLEADOS
```

13. Insertar en el documento dept.xml un nuevo departamento con código 50 nombre costes y localidad Toledo.

```
update insert <DEP_ROW>
  <DEPT_NO>50</DEPT_NO>
  <DNOMBRE>costes</DNOMBRE>
  <LOC>Toledo</LOC>
</DEP_ROW> into //departamentos
```

14. Al departamento 50 cambiar la localidad a Sevilla.

```
update value //DEP_ROW[DEPT_NO=50]/LOC with "SEVILLA"
```

15. Al empleado Antonio incrementar su salario un 10%.

```
update value //EMP_ROW[EMP_NO=9898]/SALARIO with  
(//EMP_ROW[EMP_NO=9898]/SALARIO * 1.1)
```

16. Borrar a Antonio y al departamento costes.

```
update delete //EMP_ROW[EMP_NO=9898]  
update delete //DEP_ROW[DEPT_NO=50]
```

17. Cambiar la etiqueta deptroot por departamento.

```
update rename //DEP_ROW as 'departamento'
```

18. Volver al estado original la tabla dept.

```
update rename //departamento as 'DEP_ROW'
```