

## Boeltín 1. Xquery – Archivo bookstore.xml

Crea etiquetas nuevas cuando sea necesario.

1. Mostrar los títulos de los libros con la etiqueta "titulo".  

```
for $x in /bookstore/book
return <titulo>{$x/title/string()} </titulo>
```
2. Mostrar los libros cuyo precio sea menor o igual a 30. Primero incluyendo la condición en la cláusula "where" y luego en la ruta del XPath.  

```
for $x in //book
where $x//price <= 30
return $x
```
3. Mostrar sólo el título de los libros cuyo precio sea menor o igual a 30.  

```
for $x in //book
where $x/price <= 30
return $x/title
```
4. Mostrar sólo el título sin atributos de los libros cuyo precio sea menor o igual a 30.  

```
for $x in //book
where $x/price <= 30
return <title>{$x/title/text()}</title>
```
5. –
6. Mostrar los años de publicación, primero con "for" y luego con "let" para comprobar la diferencia entre ellos. Etiquetar la salida con "publicacion".  

```
for $ano in distinct-values(//year)
return <year>{$ano}</year>
```

  

```
let $ano := distinct-values(//year)
return <year>{$ano}</year>
```
7. Mostrar los libros ordenados primero por "category" y luego por "title".  

```
for $book in //book
order by /@category,/title
return $book
```
8. Mostrar cuántos libros hay, y etiquetarlo con "total".  

```
let $num := count(//book)
return <total>{$num}</total>
```

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

```
let $num := count(//book)
return
<booksotre>
{
  for $book in //book
  return $book/title
}
<total>{$num}</total>
</booksotre>
```

10. Mostrar el precio mínimo y máximo de los libros.

```
let $max := max(//price/number())
let $min := min(//price/number())
return
<precio>
  <maximo>{$max}</maximo>
  <minimo>{$min}</minimo>
</precio>
```

11. --

12. Mostrar la suma total de los precios de los libros con la etiqueta "total".

```
let $sum := sum(//price/number())
return <total>{$sum}</total>
```

13. Mostrar cada uno de los precios de los libros, y al final una nueva etiqueta con la suma de los precios.

```
let $sum := sum(//price/number())
return
<books>
{
  for $book in //book
  return
    <book>
      <title>{$book/title/string()}</title>
      <price>{$book/price/string()}</price>
    </book>
}
<total>{$sum}€</total>
</books>
```

14. Mostrar el título y el número de autores que tiene cada título en etiquetas diferentes.

```
for $book in //book
return
<book>
  <title>{$book/title/string()}</title>
  <numeroAutor>
    {
      let $num := count($book/author)
      return $num
    }
  </numeroAutor>
</book>
```

15. Mostrar en la misma etiqueta el título y entre paréntesis el número de autores que tiene ese título.

```
for $book in //book
return
<books>
  {concat($book/title/text(),"(",count($book/author),")")}
</books>
```

16. Mostrar los libros escritos en años que terminen en "3".

```
for $book in //book
where ends-with($book/year,'3')
return $book
```

17. Mostrar los libros cuya categoría empiece por "C".

```
for $book in //book
where starts-with($book/@category,'C')
return $book
```

18. --

19. Mostrar el título y el número de caracteres que tiene cada título, cada uno con su propia etiqueta.

```
for $book in //book
let $title := $book/title
return
<book>
  {concat($book/title/text(),"(",string-length($book/title/text()),")")}
</book>
```

20. Mostrar todos los años en los que se ha publicado un libro eliminando los repetidos. Etiquétalos con "año".

```
for $ano in distinct-values(//year)
return <year>{$ano}</year>
```

21. Mostrar todos los autores eliminando los que se repiten y ordenados por el número de caracteres que tiene cada autor.

```
for $autor in distinct-values(//author)
return <year>{$autor}</year>
```

22. Mostrar los títulos en una tabla de HTML.

```
let $store := /bookstore
return
<table>
  <tbody>
    <tr>
      <td>Titulo</td>
    </tr>
    {
      for $title in $store//title/text()
      return <tr><td>{$title}</td></tr>
    }
  </tbody>
</table>
```