

UNIDAD 6 Almacenamiento de información

<

Actividad 7 Tenemos almacenados en el documento XML libros.xml los datos relativos libros.

Se pide:

- A) Convertir el archivo XML en una BD Nativas usando XQuery, nombre de la base de datos libros.
- B) Usando XQuery realiza las siguientes consultas:
 1. Consulta: Lista el título de los libros ordenados por título.
for \$x in doc('libros')/bookstore/book
return \$x/title/text()
 2. Consulta: Lista el título de los libros ordenados por título cuyo precio es superior a 30.
for \$x in doc('libros')/bookstore/book
order by \$x/title
where \$x/price>30
return data(\$x/title)
 3. Crea una página web (03.html) cuyo título de la web “UD6-3.3 Listado de libros. Nombre y apellidos” y que contenga una lista con viñetas con los títulos de los libros.



```
<html lang="es">
<head>
  <meta charset="UTF-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title> Ud6.7 Listado de Libros Juan Miguel Dominguez</title>
</head>
<body>
  <header> Listado de libros</header>
  <main>
    <ul>
      {
        for $x in doc('libros')/bookstore/book
        return <li>{$x/title/text()}</li>
      }
    </ul>
  </main>
</body>
</html>
```

```

    }
  </ul>
</main>

</body>
</html>

```

4. Crea un documento XML(04.xml) clasificación de los libros según categoría, 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>
  <adult>Everyday Italian</adult>
  <children>Harry Potter</children>
  <adult>XQuery Kick Start</adult>
  <adult>Learning XML</adult>
  <adult>Clean Code</adult>
  <adult>A Brief History of Time</adult>
  <adult>1984</adult>
  <adult>The Power of Habit</adult>
  <adult>Sapiens: A Brief History of Humankind</adult>
  <adult>The Pragmatic Programmer</adult>
  <adult>Steve Jobs</adult>
  <children>The Gruffalo</children>
  <adult>Rich Dad Poor Dad</adult>
  <adult>The Name of the Wind</adult>
</libros>

```

```
<libro>
```

```
{
```

```
for $x in doc('libros')/bookstore/book
```

```
return if ($x/@category="children")
```

```
then (<children> {data($x/title)} </children>,"&#10;")
```

```
else (<adult> {data($x/title)} </adult>,"&#10;")
```

```
}
```

```
</libro>
```

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.

Bookstore

Titulo	Categoría
Everyday Italian	cooking
Harry Potter	children
Learning XML	web
XQuery Kick Start	web

- Everyday Italian
- Harry Potter
- Learning XML
- XQuery Kick Start

```
<html lang="es">
<head>
  <meta charset="UTF-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title> Ud6.7 Lisatdo de Libors Juan Miguel Dominguez</title>
</head>
<body>
  <header>
    El Tiempo Madrid
  </header>
  <main>
    <table>
      <thead>
        <tr>
          <th>Titulo </th>
          <th>Categoría</th>
        </tr>
      </thead>
      <tbody>
        {for $x in doc('libros')/bookstore/book
        return

          <tr>
            <td>{data($x/title)}</td>
            <td>{data($x/@category)}</td>
          </tr>
```

```

    }
  </tbody>
</table>
</main>
</body>
</html>

```

6. Lista de libros ordenados por categoría de forma descendiente, y como segundo criterio por título de forma descendiente. Mostrando el título con precio, donde el precio es mayor que 30 \$. El resultado (06.txt) es un fichero de texto:

```

Learning XML:39.95$
XQuery Kick Start:49.99$

```

```
for $x in doc("libros")/bookstore/book
```

```
where $x/price >30
```

```
order by $x/@category descending, $x/title descending
```

```
return concat($x/title/text()," ", data($x/Price))
```

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>
  <book>1. Everyday Italian</book>
  <book>2. Harry Potter</book>
  <book>3. XQuery Kick Start</book>
  <book>4. Learning XML</book>
</books>

```

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 valor inicial to valor final, por ejemplo 1 to 100
Generar automáticamente la siguiente página web (08.html)

Listado de prácticas

[Tema1 Ejercicio1](#)
[Tema1 Ejercicio2](#)
[Tema1 Ejercicio3](#)
[Tema2 Ejercicio1](#)
[Tema2 Ejercicio2](#)
[Tema2 Ejercicio3](#)
[Tema3 Ejercicio1](#)
[Tema3 Ejercicio2](#)
[Tema3 Ejercicio3](#)
[Tema4 Ejercicio1](#)
[Tema4 Ejercicio2](#)
[Tema4 Ejercicio3](#)
[Tema5 Ejercicio1](#)
[Tema5 Ejercicio2](#)
[Tema5 Ejercicio3](#)
[Tema6 Ejercicio1](#)
[Tema6 Ejercicio2](#)
[Tema6 Ejercicio3](#)
[Tema7 Ejercicio1](#)
[Tema7 Ejercicio2](#)
[Tema7 Ejercicio3](#)

```
<html>
<head>
<title>Listado de prácticas</title>
</head>
```

```
<body>
```

```
<h1>Listado de prácticas</h1>
```

```
{
for $t in /practicas/tema
for $e at $pos in $t/ejercicio
```

```
let $tema := $t/@id
```

```
return
```

```
<div>
<a href="tema{$tema}_ej{$pos}.html">
  Tema{$tema} {data($e)}
</a>
</div>

}
```

```
</body>
</html>
```

Donde cada ejercicio es un enlace

```
<div>
  <a href="1/2.html">Tema1 Ejercicio2</a>
</div>
```

9. Mostrar el título del libro cuando tenga más de dos autores.
for \$x in doc('libros')/bookstore/book

```
let $c := $x/author
```

```
where count($c) > 2
return data($x/title)
```

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

```
<libros>
  <libro autores="1">Everyday Italian</libro>
  <libro autores="1">Harry Potter</libro>
  <libro autores="5">XQuery Kick Start</libro>
  <libro autores="1">Learning XML</libro>
</libros>
<libros>
```

```
{
for $x in doc('libros')/bookstore/book
```

```
let $c := $x/author
```

```
return <libro autores="{count($c)}">{data($x/title)}</libro>}
```

```
</libros>
```

11. Mostrar el título y el autor de los libros del año 2005, y etiquetar cada uno de ellos con “lib2005”.

```
<libros>
```

```
{  
for $x in doc('libros')/bookstore/book[year=2005]
```

```
let $c := $x/author
```

```
(:where $x/year = 2005:)
```

```
return <lib2005> {concat(data($x/title)," ",data($c))} </lib2005>}
```

```
</libros>
```

12. Mostrar los años de publicación, primero con for y luego con let. Etiquetar la salida con publicación
<publicaciones>

```
{  
for $x in doc('libros')/bookstore/book/year  
  
return <publicacion> {data($x)} </publicacion>  
}
```

```
{  
let $c := doc('libros')/bookstore/book/year  
  
return <publicacionconlet> {data($c)} </publicacionconlet>  
}
```

```
</publicaciones>
```

13. Mostrar los libros ordenados primero por “category” y luego por “title” en una sola consulta.
for \$x in doc('libros')/bookstore/book

```
order by $x/@category, $x/title
```

```
return $x
```

14. Mostrar cuántos libros hay, y etiquetarlo con “total”
for \$x in doc('libros')/bookstore

```
return concat("total": , count($x/book))
```

15. Mostrar los títulos de los libros y al final una etiqueta con el número total de libros
<libro>

```
{  
for $b in doc("libros.xml")/bookstore  
return <titulo>{data($b/book/title)}</ titulo >
```

```

}
<total>
{
  count(doc("libros.xml")/bookstore/book)
}
</total>
</libro>

```

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

```

<libros>
{
  for $x in doc('libros')/bookstore

    return <librocaro> {max($x/book/price)} </librocaro> <librobarato> {min($x/book/price)} </librobarato>
}
</libros>

```

17. Mostrar el título del libro, su precio y su precio con el IVA incluido, cada uno con su propia etiqueta. Ordenado de precio con IVA (4% en España)

```

<libros>
{
  for $x in doc('libros')/bookstore

    let $i := $x/book/Price *1.04

    <libro>
    return <titulo> {data($x/book/title)} </titulo> <precio> {data($x/book/price)} </precio> <precioiva>
    {data($i)} </precioiva>
    </libro>
  }
</libros>

```

18. Mostrar la suma total de los precios con la etiqueta total

```

for $x in doc('libros')/bookstore

return <total> sum($x/book/price) </total>

```

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

```

<precios>
{
  for $b in doc("libros.xml")/bookstore/book
    return <precio>{data($b/price)}</precio>
}
<total>
{
  sum(doc("libros.xml")/bookstore/book/price)
}
</total>
</precios>

```

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

```

for $b in doc("libros.xml")/bookstore/book
return

```

```

<libro>
  <titulo>{data($b/title)}</titulo>
  <numAutores>{count($b/author)}</numAutores>
</libro>

```

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

```

for $b in doc("libros.xml")/bookstore/book
return
<libro>
{
  concat(
    data($b/title),
    " (",
    count($b/author),
    ")"
  )
}
</libro>

```

22. Mostrar los libros escritos en años que terminan en "3"

```

for $b in doc("libros.xml")/bookstore/book
where ends-with(data($b/year), "3")
return $b/title

```

23. Mostrar los libros cuya categoría empiece por C

```

for $b in doc("libros.xml")/bookstore/book
where starts-with(lower-case(data($b/@category)), "c")
return $b/title

```

24. Mostrar los libros que contenga una X mayúscula o minúscula en el título ordenados de manera descendente

```

for $b in doc("libros.xml")/bookstore/book
where contains(lower-case(data($b/title)), "x")
order by $b/title descending
return $b/title

```

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

```

for $b in doc("libros.xml")/bookstore/book
return
<libro>
  <titulo>{data($b/title)}</titulo>
  <caracteres>{string-length(data($b/title))}</caracteres>
</libro>

```

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

```

for $y in distinct-values(doc("libros.xml")/bookstore/book/year)
order by $y

```

```
return  
<año>{$y}</año>
```

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

```
for $a in distinct-values(doc("libros.xml")/bookstore/book/author)  
order by string-length($a)  
return  
<autor>{$a}</autor>
```

28. Mostrar los títulos en una tabla de HTML.

```
<html>  
<body>  
  
<table border="1">  
  <tr>  
    <th>Título</th>  
  </tr>  
  
  {  
    for $b in doc("libros.xml")/bookstore/book  
    return  
    <tr>  
      <td>{data($b/title)}</td>  
    </tr>  
  }  
  
</table>  
  
</body>  
</html>
```

FORMA DE ENTREGA:

Crea un web que contenga todas las consultas(captura) y sus resultados (captura), y publicado en la web de lenguaje de marcas.

