

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 $b in doc("libros.xml")//book
order by $b/title
return $b/title
```

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

```
for $b in doc("libros.xml")//book
where $b/price > 30
order by $b/title
return $b/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>
  <head>
    <title>UD6-3.3 Listado de libros. TuNombre TusApellidos</title>
  </head>
  <body>
    <ul>
      {
        for $b in doc("libros.xml")//book
        return <li>{data($b/title)}</li>
      }
    </ul>
  </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>
{
  for $b in doc("libros.xml")//book
  return
    if ($b/@category = "children") then
      <children>{data($b/title)}</children>
    else
      <adult>{data($b/title)}</adult>
}
```

</libros>

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>
<body>
<ul>
{
  for $b in doc("libros.xml")//book
  let $color :=
    if ($b/@category = 'children') then 'blue'
    else if ($b/@category = 'cooking') then 'green'
    else 'red'
  return <li style="color: {$color};">{data($b/title)} ({data($b/@category)})</li>
}
</ul>
</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:

```
for $b in doc("libros.xml")//book
where $b/price > 30
order by $b/@category descending, $b/title descending
return concat(data($b/title), ":", data($b/price), "$&#10;")
```

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 $b at $pos in doc("libros.xml")//book
  return <book>{concat($pos, ". ", data($b/title))}</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 valorinicial to valorfinal, por ejemplo 1 to 100

```
<html>
<body>
<h1>Listado de prácticas</h1>
{
  for $tema in (1 to 7)
  for $ejercicio in (1 to 3)
  return
    <div>
      <a href="{concat($tema, '/', $ejercicio, '.html')}">
        Tema{$tema} Ejercicio{$ejercicio}
    </div>
}
```

```

    </a>
  </div>
}
</body>
</html>

```

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

```

for $b in doc("libros.xml")//book
where count($b/author) > 2
return $b/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>
{
  for $b in doc("libros.xml")//book
  return <libro autores="{count($b/author)}">{data($b/title)}</libro>
}
</libros>

```

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

```

for $b in doc("libros.xml")//book
where $b/year = 2005
return <lib2005>{$b/title, $b/author}</lib2005>

```

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

```

(: Con FOR :)
for $y in doc("libros.xml")//book/year
return <publicacion>{data($y)}</publicacion>,

```

```

(: Con LET :)
let $y := doc("libros.xml")//book/year
return <publicacion>{data($y)}</publicacion>

```

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

```

for $b in doc("libros.xml")//book
order by $b/@category, $b/title
return $b

```

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

```

let $total := count(doc("libros.xml")//book)
return <total>{$total}</total>

```

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

```
let $books := doc("libros.xml")//book
return (
  for $b in $books return $b/title,
  <total>{count($books)}</total>
)
```

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

```
let $prices := doc("libros.xml")//book/price
return (
  <min>{min($prices)}</min>,
  <max>{max($prices)}</max>
)
```

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)

```
for $b in doc("libros.xml")//book
return (
  $b/title,
  $b/price,
  <precio_con_iva>{$b/price * 1.04}</precio_con_iva>
)
```

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

```
for $b in doc("libros.xml")//book
order by ($b/price * 1.04)
return (
  $b/title,
  $b/price,
  <precio_con_iva>{$b/price * 1.04}</precio_con_iva>
)
```

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

```
let $suma := sum(doc("libros.xml")//book/price)
return <total>{$suma}</total>
```

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

```
let $books := doc("libros.xml")//book
return (
  for $b in $books return $b/price,
  <total>{sum($books/price)}</total>
)
```

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")//book
return (
  $b/title,
```

```
<num_autores>{count($b/author)}</num_autores>
)
```

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

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

23. Mostrar los libros cuya categoría empiece por C

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

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

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

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

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

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

```
for $b in doc("libros.xml")//book
return (
  $b/title,
  <caracteres>{string-length($b/title)}</caracteres>
)
```

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

```
for $year in distinct-values(doc("libros.xml")//book/year)
return <año>{$year}</año>
```

28. Mostrar los títulos en una tabla de HTML.

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

