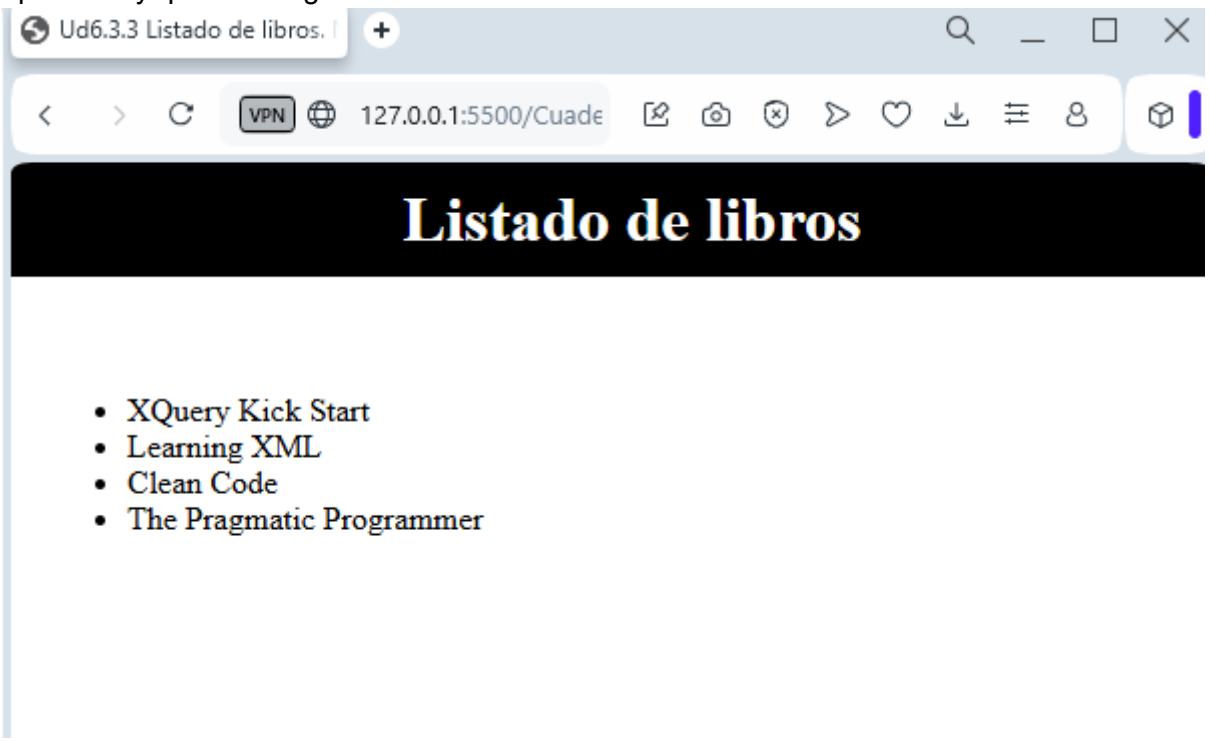
Actividad 7 Tenemos almacenados en el documento XML libros.xml los datos relativos libros.

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



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**>.

```xml
<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>
```

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

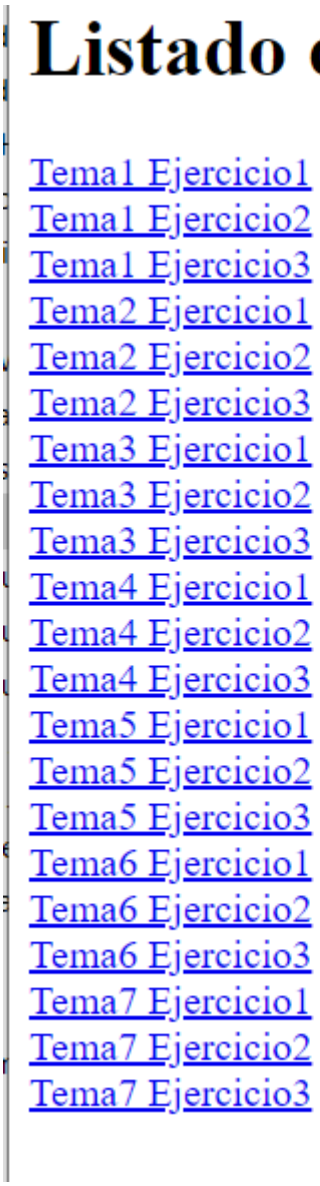
6. Lista de libros ordenados por categoría de forma descendiente, y como segundo criterio por título de forma descendiente. Mostrando el título con precio, donde el precio es mayor que 30 $. El resultado (06.txt) es un fichero de texto:

```
Learning XML:39.95$
XQuery Kick Start:49.99$
```

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 valorinicial to valorfinal, por ejemplo 1 to 100
   Generar automáticamente la siguiente página web (08.html)



Donde cada ejercicio es un enlace

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

```
for $x in doc("libros.xml")/bookstore/book
where count($x/author) > 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:**

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

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

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

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

let $x := doc("libros.xml")/bookstore/book
return <publicacion>{data($x/year)}</publicacion>
```

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

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

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

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

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

```
let $precios := doc("libros.xml")/bookstore/book/price
return
<resultado>
<minimo>{min($precios)}</minimo>
<maximo>{max($precios)}</maximo>
 </resultado>
```

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 $x in doc("libros.xml")/bookstore/book
let $iva := $x/price * 1.04
order by $iva
return <libro>
<titulo>{data($x/title)}</titulo>
<precio>{data($x/price)}</precio>
<precio_iva>{$iva}</precio_iva>
</libro>
```

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

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

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

```
let $libros := doc("libros.xml")/bookstore/book
return (
  for $l in $libros return data($l/price),
  <suma>{sum($libros/price)}</suma>)
```

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

```
for $x in doc("libros.xml")/bookstore/book
return  {data($x/title)} <num_autores>{count($x/author)}</num_autores>
```

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

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

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

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

23. **Mostrar los libros cuya categoría empiece por C**

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

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

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

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

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

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

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

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

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

28. **Mostrar los títulos en una tabla de HTML.**

```
<table>
  <tr><th>Título</th></tr>
  {
 for $x in doc("libros.xml")/bookstore/book/title
    return tr(td(data($x)))
  }
</table>
```

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.