1. Introducción

De manera rápida podemos definir XQuery con un símil en el que XQuery es a XML lo mismo que SQL es a las bases de datos relacionales.

XQuery es un lenguaje de consulta diseñado para escribir consultas sobre colecciones de datos expresadas en XML. Abarca desde archivos **XML** hasta bases de datos relacionales con funciones de conversión de registros a XML. **Su principal función es extraer información de un conjunto de datos organizados como un árbol nário de etiquetas XML**. En este sentido **XQuery** es independiente del origen de los datos.

XQuery es un lenguaje funcional, lo que significa que, en vez de ejecutar una lista de comandos como un lenguaje procedimental clásico, cada consulta es una expresión que es evaluada y devuelve un resultado, al igual que en **SQL**. Diversas expresiones pueden combinarse de una manera muy flexible con otras expresiones para crear nuevas expresiones más complejas y de mayor potencia semántica.

XQuery está llamado a ser el futuro estándar de consultas sobre documentos **XML**. Actualmente, **XQuery es un conjunto de borradores en el que trabaja el grupo W3C**. Sin embargo, a pesar de no tener una redacción definitiva ya existen o están en proceso numerosas implementaciones de motores y herramientas que lo soportan.

2. Requerimientos técnicos de XQuery

El grupo de trabajo en **XQuery** del **W3C** ha definido un conjunto de requerimientos técnicos para este lenguaje. Los más importantes se detallan a continuación.

- Cuando se va a analizar un documento **XML**, se crea un árbol de nodos del mismo.
- Ese árbol tiene un elemento raíz y una serie de hijos. Los hijos del nodo raíz pueden tener más hijos.
- Si repetimos ese proceso llegará un momento en el que el último nodo no tiene ningún hijo, lo que se denomina nodo hoja.
- Los tipos de nodos se puede encontrar en ese recorrido son los siguientes:
    - **Nodo raíz o "1":** Es el primer nodo del documento **XML**.
    - **Nodo elemento:** Cualquier elemento de un documento **XML** es un nodo elemento en el árbol. El nodo raíz es un caso especial de Nodo elemento (no tiene padre). Cada nodo elemento posee un

padre y puede o no poseer hijos. En el caso que no tenga hijos, sería un nodo hoja.
- **Nodo texto:** Cualquier elemento del documento que no esté marcado con una etiqueta de la DTD del documento XML.
- **Nodo atributo:** Un nodo elemento puede tener etiquetas que complementen la información de ese elemento. Eso sería un nodo atributo.

La extracción de la información durante el recorrido del árbol será tan simple como la detección de los nodos a buscar y el procesamiento de la información que se quiere extraer de ese nodo concreto.
Esto que parece tan complicado, se realiza fácilmente con **Xpath**.
XPath es la herramienta que utiliza **XQuery** para procesar el árbol de nodos de un documento **XML**.

XQuery Reglas de sintaxis básica

- XQuery distingue entre mayúsculas y minúsculas
- Elementos de XQuery, atributos y variables deben ser nombres XML válidos
- Un valor de cadena de XQuery puede estar entre comillas simples o dobles
- Una variable de XQuery se define con un $ seguido de un nombre, por ejemplo, $librería
- Los comentarios están delimitados por (: y :), por ejemplo, (: Esto es un comentario :)

Para los ejemplos utilizaremos el fichero **europa.xml** que se puede encontrar depositado en la plataforma **aules**.

Lectura de archivos

Con la función doc () se lee un documento XML que se indique como parámetro y devuelve el nodo raíz o los elementos que se indiquen mediante una expresión XPath

```
doc("europa.xml")
```

La expresión siguiente se utiliza para seleccionar todos los nombres de los países en el fichero "**europa.xml**"

```
doc("europa.xml")//pais/nombre
```

El resultado sería el siguiente:

```
<nombre>Albania</nombre>
<nombre>Alemania</nombre>
<nombre>Andorra</nombre>
<nombre>Austria</nombre>
<nombre>Bélgica</nombre>
<nombre>Bielorrusia</nombre>
<nombre>Bosnia y Herzegovina</nombre>
<nombre>Bulgaria</nombre>
<nombre>Chequia</nombre>
<nombre>Chipre</nombre>
<nombre>Ciudad Vaticano</nombre>
<nombre>Croacia</nombre>
<nombre>Dinamarca</nombre>
<nombre>Eslovaquia</nombre>
<nombre>Eslovenia</nombre>
<nombre>España</nombre>
<nombre>Estonia</nombre>
<nombre>Finlandia</nombre>
<nombre>Francia</nombre>
<nombre>Gibraltar</nombre>
<nombre>Grecia</nombre>
<nombre>Guernsey</nombre>
<nombre>Hugría</nombre>
<nombre>Irlanda</nombre>
<nombre>Islandia</nombre>
<nombre>Italia</nombre>
<nombre>Jersey</nombre>
<nombre>Letonia</nombre>
<nombre>Liechtenstein</nombre>
<nombre>Lituania</nombre>
<nombre>Luxemburgo</nombre>
<nombre>Macedonia</nombre>
<nombre>Malta</nombre>
<nombre>Man</nombre>
<nombre>Moldavia</nombre>
<nombre>Mónaco</nombre>
<nombre>Montenegro</nombre>
<nombre>Noruega</nombre>
<nombre>Paises Bajos</nombre>
<nombre>Polonia</nombre>
<nombre>Portugal</nombre>
<nombre>Reino Unido</nombre>
```

```
<nombre>Rumanía</nombre>
<nombre>Rusia</nombre>
<nombre>San Marino</nombre>
<nombre>Serbia</nombre>
<nombre>Suecia</nombre>
<nombre>Suiza</nombre>
<nombre>Ucrania</nombre>
```

XQuery utiliza filtros para limitar los datos extraídos de documentos XML.

```
doc("europa.xml")//pais[poblacion>50000000]
```

El filtro anterior se utiliza para seleccionar todos los países cuya población es mayor que 50000000

El resultado sería el siguiente:

```
<pais>
   <cod_continente>f0_1100</cod_continente>
   <cod_pais>f0_110002</cod_pais>
   <nombre>Alemania</nombre>
   <capital>Berlín</capital>
   <poblacion anio="2020">083155031</poblacion>
    <superficie>00357375.62</superficie>
   <estado>República Federal</estado>
   <lenguas>
          <lengua porcentaje="87">Alemán</lengua>
          <lengua porcentaje="04">Ruso</lengua>
          <lengua porcentaje="02">Turco</lengua>
   </lenguas>
   <grupos_etnicos>
          <grupo_etnico porcentaje="80">alemanes</grupo_etnico>
          <grupo_etnico porcentaje="03.7">Turcos</grupo_etnico>
          <grupo_etnico
   porcentaje="01.9">Polacos</grupo_etnico>
     <grupo_etnico porcentaje="01.5">Rusos</grupo_etnico>
   </grupos_etnicos>
   <religiones>
          <religion porcentaje="30.8">Católicos</religion>
          <religion porcentaje="30.2">Protestantes</religion>
          <religion porcentaje="05">Islamismo</religion>
   </religiones>
   <fronteras>
          <frontera cod_pais="f0_110004">784</frontera>
```

```
            <frontera cod_pais="f0_110009">646</frontera>
            <frontera cod_pais="f0_110039">577</frontera>
            <frontera cod_pais="f0_110040">456</frontera>
      <frontera cod_pais="f0_110019">451</frontera>
            <frontera cod_pais="f0_110048">334</frontera>
            <frontera cod_pais="f0_110005">167</frontera>
            <frontera cod_pais="f0_110031">138</frontera>
            <frontera cod_pais="f0_110013">068</frontera>
   </fronteras>
</pais>
<pais>
   <cod_continente>f0_1100</cod_continente>
   <cod_pais>f0_110019</cod_pais>
   <nombre>Francia</nombre>
   <capital>París</capital>
   <poblacion anio="2020">067656682</poblacion>
   <superficie>00675417</superficie>
   <estado>República Semipresidencialista Unitaria</estado>
   <lenguas>
            <lengua porcentaje="90.00">Francés</lengua>
            <lengua porcentaje="03.69">Criollo</lengua>
            <lengua porcentaje="02.96">Occitano</lengua>
            <lengua porcentaje="01.04">Alsaciano</lengua>
            <lengua porcentaje="00.37">Bretón</lengua>
            <lengua porcentaje="00.19">Corso</lengua>
            <lengua porcentaje="00.15">Flamenco</lengua>
   </lenguas>
   <grupos_etnicos>
            <grupo_etnico
   porcentaje="92">Franceses</grupo_etnico>
            <grupo_etnico
   porcentaje="03">Norteafricanos</grupo_etnico>
            <grupo_etnico porcentaje="02">Alemanes</grupo_etnico>
            <grupo_etnico porcentaje="01">Bretones</grupo_etnico>
   </grupos_etnicos>
   <religiones>
            <religion porcentaje="50">Católicos</religion>
            <religion porcentaje="10">Musulmanes</religion>
            <religion porcentaje="02">Protestantes</religion>
   </religiones>
   <fronteras>
            <frontera cod_pais="f0_110016">656.3</frontera>
            <frontera cod_pais="f0_110005">620</frontera>
            <frontera cod_pais="f0_110048">573</frontera>
            <frontera cod_pais="f0_110026">515</frontera>
            <frontera cod_pais="f0_110002">451</frontera>
```

```
            <frontera cod_pais="f0_110031">073</frontera>
            <frontera cod_pais="f0_110003">057</frontera>
            <frontera cod_pais="f0_110036">005.5</frontera>
      </fronteras>
</pais>
<pais>
    <cod_continente>f0_1100</cod_continente>
    <cod_pais>f0_110026</cod_pais>
    <nombre>Italia</nombre>
    <capital>Roma</capital>
    <poblacion anio="2021">060257566</poblacion>
    <superficie>00301340</superficie>
    <estado>República Parlamentaria</estado>
    <lenguas>
            <lengua porcentaje="97">Italiano</lengua>
            <lengua porcentaje="00.42">Alemán</lengua>
            <lengua porcentaje="00.30">Occitano</lengua>
            <lengua porcentaje="00.13">Albanés</lengua>
            <lengua porcentaje="00.08">Esloveno</lengua>
            <lengua porcentaje="00.04">Catalán</lengua>
            <lengua porcentaje="00.03">Francés</lengua>
            <lengua porcentaje="00.03">Griego</lengua>
    </lenguas>
    <grupos_etnicos>
            <grupo_etnico
    porcentaje="91.5">Italianos</grupo_etnico>
            <grupo_etnico
    porcentaje="01.5">Rumanos</grupo_etnico>
            <grupo_etnico
    porcentaje="01">Norteafricanos</grupo_etnico>
    </grupos_etnicos>
     <religiones>
      <religion porcentaje="79.2">Católicos</religion>
            <religion porcentaje="03.5">Ortodoxos</religion>
            <religion porcentaje="01">Islamismo</religion>
            <religion porcentaje="00.4">Budistas</religion>
    </religiones>
    <fronteras>
            <frontera cod_pais="f0_110048">740</frontera>
            <frontera cod_pais="f0_110019">515</frontera>
            <frontera cod_pais="f0_110004">430</frontera>
            <frontera cod_pais="f0_110015">218</frontera>
            <frontera cod_pais="f0_110045">037</frontera>
            <frontera cod_pais="f0_110011">003.2</frontera>
     </fronteras>
</pais>
```

```
<pais>
   <cod_continente>f0_1100</cod_continente>
   <cod_pais>f0_110042</cod_pais>
   <nombre>Reino Unido</nombre>
   <capital>Lóndres</capital>
   <poblacion anio="2020">067747826</poblacion>
   <superficie>00243610</superficie>
   <estado>Monarquía Parlamentaria</estado>
   <lenguas>
          <lengua porcentaje="95">Inglés</lengua>
   </lenguas>
   <grupos_etnicos>
          <grupo_etnico
   porcentaje="87">Británicos</grupo_etnico>
          <grupo_etnico
   porcentaje="02.3">Hindúes</grupo_etnico>
          <grupo_etnico
   porcentaje="01.9">Pakistaníes</grupo_etnico>
          <grupo_etnico
   porcentaje="00.7">Bangladesíes</grupo_etnico>
          <grupo_etnico porcentaje="00.7">Chinos</grupo_etnico>
   </grupos_etnicos>
   <religiones>
          <religion porcentaje="55.1">Protestantes</religion>
          <religion porcentaje="22">Católicos</religion>
          <religion porcentaje="04.4">Islamismo</religion>
          <religion porcentaje="01.3">Hinduistas</religion>
   </religiones>
   <fronteras>
               <frontera cod_pais="f0_110024">499</frontera>
    </fronteras>
</pais>
<pais>
   <cod_continente>f0_1100</cod_continente>
    <cod_pais>f0_110044</cod_pais>
   <nombre>Rusia</nombre>
   <capital>Moscú</capital>
   <poblacion anio="2020">145558000</poblacion>
   <superficie>17098250</superficie>
   <estado>República Federal Semiparlamentaria</estado>
   <lenguas>
          <lengua porcentaje="55">Ruso</lengua>
          <lengua porcentaje="29">Tártaro</lengua>
          <lengua porcentaje="08.5">Chuvasio</lengua>
          <lengua porcentaje="07.5">Baskir</lengua>
   </lenguas>
```

```
    <grupos_etnicos>
            <grupo_etnico porcentaje="80.9">Rusos</grupo_etnico>
            <grupo_etnico
    porcentaje="03.9">Tártaros</grupo_etnico>
            <grupo_etnico
    porcentaje="01.4">Ucranianos</grupo_etnico>
            <grupo_etnico
    porcentaje="01.1">Baskires</grupo_etnico>
            <grupo_etnico
    porcentaje="01">Chuvasios</grupo_etnico>
            <grupo_etnico
    porcentaje="01">Chechenos</grupo_etnico>
    </grupos_etnicos>
    <religiones>
            <religion porcentaje="41.1">Ortodoxos</religion>
            <religion porcentaje="06.5">Islamismo</religion>
            <religion porcentaje="06.3">Católicos</religion>
    </religiones>
    <fronteras>
            <frontera nombre_pais="Kazajistán">7513</frontera>
            <frontera nombre_pais="China">4209</frontera>
            <frontera nombre_pais="Mongolia">3485</frontera>
            <frontera cod_pais="f0_110049">2094</frontera>
            <frontera cod_pais="f0_110018">1271</frontera>
            <frontera cod_pais="f0_110006">1239</frontera>
            <frontera nombre_pais="Georgia">0876</frontera>
            <frontera nombre_pais="Azerbaiyán">0327</frontera>
            <frontera cod_pais="f0_110017">0325</frontera>
            <frontera cod_pais="f0_110028">0271</frontera>
            <frontera cod_pais="f0_110030">0266</frontera>
            <frontera cod_pais="f0_110040">0204</frontera>
            <frontera cod_pais="f0_110038">0196</frontera>
            <frontera nombre_pais="Corea del
    Norte">0019</frontera>
    </fronteras>
</pais>
```

Si sólo queremos los nombres de los países que tienen una población menor de 750000, la consulta sería la siguiente:

```
doc("europa.xml")//pais[poblacion<750000]/nombre
```

El resultado de la consulta:

```
<nombre>Andorra</nombre>
<nombre>Ciudad Vaticano</nombre>
<nombre>Gibraltar</nombre>
<nombre>Guernsey</nombre>
<nombre>Islandia</nombre>
<nombre>Jersey</nombre>
<nombre>Liechtenstein</nombre>
<nombre>Luxemburgo</nombre>
<nombre>Malta</nombre>
<nombre>Man</nombre>
<nombre>Mónaco</nombre>
<nombre>Montenegro</nombre>
<nombre>San Marino</nombre>
```

3. Consultas en XQuery

Las consultas en XQuery suelen combinar la información de una o más fuentes de datos reestructurándola para crear un nuevo resultado.

Las expresiones **FLWOR** (siglas de las cláusulas **for**, **let**, **where**, **order** y **return**), son una de las características más importantes y típicas de XQuery para conseguirlo. Son similares a **SELECT-FROM-WHERE** de **SQL**.

Sin embargo, una expresión **FLWOR** no está definida en términos de tablas, filas y columnas; una expresión **FLOWR** asocia expresiones **XPath** a las variables de las cláusulas for y let, y utiliza dichas asociaciones para crear nuevos resultados. En **XQuery**, cuando usamos el término de **tupla**, nos estamos refiriendo a cada uno de los valores que toma esa asociación.

for	Asocia una (o más) variables a una secuencia de valores, creando una serie de tuplas en el que cada tupla está vinculada a una de las variables	EJEMPLO for $b in doc(libros)/libro let $c:=$b/autor where count ($c)>2 order by $b/titulo return $b/titulo Devuelve los títulos ordenados alfabéticamente de cada uno de los libros que tienen más de dos autores.
let	Asocia una variable al resultado entero de una expresión y devuelve una única tupla, pudiéndose añadir esta nueva asociación a las tupla generadas por una cláusula for si esta existe	
Where	Filtra las tuplas producidas por las cláusulas let y for eliminando todos los valores que no cumplan las condiciones dadas	
order by	Ordena las tuplas según el criterio dado	
return	Construye el resultado de la consulta para cada tupla dada, después de haber sido filtrada por la cláusula where y ordenada por la cláusula order by	

A continuación, en la tabla siguiente, se describe la función de cada bloque:

For: Esta sentencia permite seleccionar los nodos que se quieren consultar, guardándose en la variable

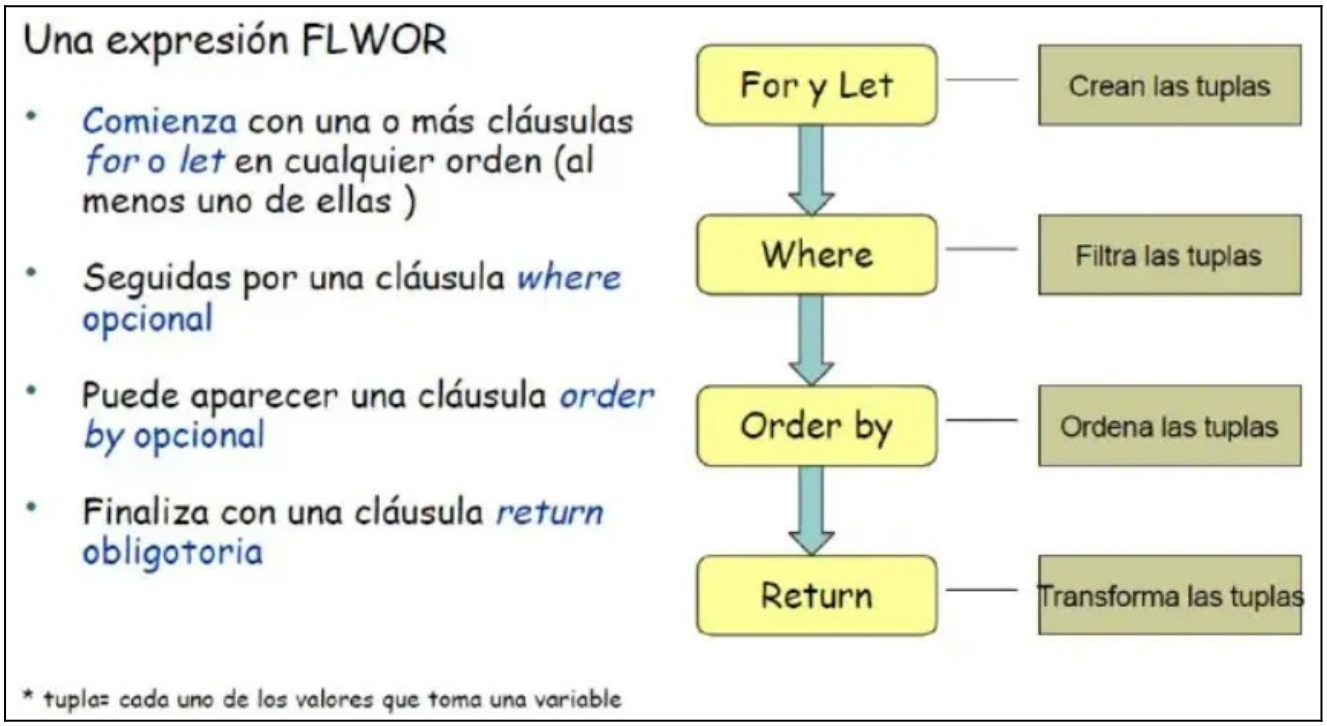
Let: Permite declarar variables a las que se les asignan valores.

Where: Filtra las tuplas eliminando todos los valores que no cumplan las condiciones dadas.

Order by: Ordena las tuplas según el criterio dado.

Return: Devuelve los resultados

En **XQuery**, cuando usamos el término tupla, nos estamos refiriendo a cada uno de los valores que toma una variable.



Con estas cláusulas se consigue buena parte de la funcionalidad que diferencia a **XQuery** de **Xpath**.

Entre otras cosas permite construir el documento que será la salida de la consulta

Diferencias entre for y let

A parte de las diferencias sintácticas (for $x in expresión y let $x:= expresión), se diferencian en los resultados que generan. La cláusula **for** asocia una variable con cada nodo que encuentre en la colección de datos, es una secuencia de datos. La cláusula **let**, en cambio, asocia una variable con todo el resultado de una expresión.

Consulta	Devuelve
for $d in doc("libros.xml")/bib/libro/titulo return <titulos>{ $d }</titulos>	<titulos> <titulo>TCP/IP Illustrated</titulo> </titulos> <titulos> <titulo>Advan Programming for Unix environment</titulo> </titulos> <titulos> <titulo>Data on the Web</titulo> </titulos> <titulos> <titulo> Economics of Technology for Digital TV</titulo> </titulos>
let $d := doc("libros.xml")/bib/libro/titulo return <titulos>{ $d }</titulos>	<titulos> <titulo>TCP/IP Illustrated</titulo> <titulo>Advan Programming for Unix environment</titulo> <titulo>Data on the Web</titulo> <titulo> Economics of Technology for Digital TV</titulo> </titulos>

Reglas generales

Toda consulta escrita en **XQuery** debe cumplir las siguientes reglas:

- **for** y **let** sirven para crear las tuplas con las que trabajará el resto de las cláusulas de la consulta y pueden usarse tantas veces como se desee en una consulta, incluso dentro de otras cláusulas.
- Sin embargo solo pueden declararse una única cláusula **where** (aunque puede tener condiciones anidadas por and, or, …), **una única** cláusula **order by** (aunque es posible especificar varios criterios de ordenación separándolos por comas) y **una única** cláusula **return**
- Ninguna de las cláusulas **FLWOR** es **obligatoria** en una consulta **Xquery.**

Operadores y funciones principales

El conjunto de funciones y operadores soportado por **XQuery** es el mismo conjunto de funciones y operadores utilizados en **XPath** y **XSLT**.

XQuery soporta operadores y funciones matemáticas, de cadenas, para el tratamiento de expresiones regulares, comparaciones de fechas y horas, manipulación de nodos **XML**, manipulación de secuencias, comprobación y conversión de tipos y lógica booleana. Además permite definir funciones propias y funciones dependientes del entorno de ejecución del motor **XQuery**:

- **Matemáticos**: +, -, *, div, idiv (división entera entre enteros), mod(resto)
- **Comparación**: =, !=, <, <=, >, >=, not()
- **Secuencia**: | (unión), intersect, except
- **Redondeo**: floor(), ceiling(), round()
- **Funciones de agrupación**: count(), min(), max(), avg(), sum()
- **Funciones de cadena**: concat(), string-length(), startswith(), ends-with(), substring(), upper-case(), lwer-case(), string()
- **Uso general**: distinct-values(), empty(), exists()

Constructores de nodos

Si se desea que el resultado de la consulta se aun elemento o atributo, debe incluirse dentro de la cláusula **return** de la siguiente manera:

- Etiquetas de apertura y cierre del elemento
- Y en medio y entre **{ }** una lista de expresiones que devuelven el contenido del elemento.

Expresiones constituidas por valores atómicos

Hemos trabajado con expresiones sobre nodos (expresiones **XPath**), pero **XQuery** también admite las llamadas expresiones constituidas por valores atómicos. Ejemplos:

- let $a := 53 → literal integer
- for $a in (1,3,5) → operador, concatena valores para formar una secuencialet
- let $a := (1 to 3) → devuelve al secuencia, 1,2,3

Iteraciones

La cláusula **for** puede también utilizarse para repetir un número fijo de veces una acción. Ejemplo:

```
<cars>
{
    for $i in (1 to 5, 7, 8)
    return <car>{$i}</car>
}
</cars>
```

Variable posicional at

Usada en una cláusula **for** permite ligar una variable a la posición del elemento en la expresión. Ejemplo:

```
let $books := $doc/bookstore/book
for $book at $i in $books
where $i eq 2

return $book
```

Expresiones condicionales (if-then-else)

Además de la cláusula **WHERE**, **XQuery** también soporta expresiones condicionales del tipo "**if-then-else**" con la misma semántica que en los lenguajes de programación más habituales.

La cláusula **WHERE** de una consulta permite filtrar las tuplas que aparecerán en el resultado, mientras que una expresión condicional nos permite crear una u otra estructura de nodos en el resultado que dependa de los valores de las tuplas filtradas.

La cláusula "**else**" **es obligatoria** y debe aparecer **siempre** en la expresión condicional. El motivo de esto es que toda expresión en **XQuery** debe devolver un valor. Si no existe ningún valor a devolver al no cumplirse la cláusula "**if**", devolvemos una secuencia vacía con "**else ()**".

Ejemplo:

```
for $b in doc("libros.xml")//libro
return
<libro>
        { $b/titulo }
        {
                for $a at $i in $b/autor
                where $i <= 2
                return <autor>{string($a/apellido), ", ", string($a/nombre)}</autor>
        }
        {
                if (count($b/autor) > 2)
                then <autor>et al.</autor>
                else ()
        }
</libro>
```

Ejemplos de consultas.

Las siguientes consultas devuelven los mismos resultados

Visualizar el nombre de todos los ríos del fichero europa.xml

- `doc("europa.xml")//rios/rio/nombre`
- ```
  for $x in doc("europa.xml")//rios/rio/nombre
  return $x
  ```
  ```

```
<nombre>Río Adigio</nombre>
<nombre>Río Aller</nombre>
<nombre>Río Amstel</nombre>
<nombre>Río Ångerman</nombre>
<nombre>Río Aqueronte</nombre>
<nombre>Río Argeș</nombre>
<nombre>Río Aspropótamos</nombre>
<nombre>Río Bîc</nombre>
<nombre>Río Bolshói Yegorlyk</nombre>
<nombre>Río Bojana</nombre>
<nombre>Río Bojana</nombre>
<nombre>Río Bosna</nombre>
<nombre>Río Bug Meridional</nombre>
<nombre>Río Bug Occidental</nombre>
<nombre>Río Buzău</nombre>
<nombre>Río Cando</nombre>
<nombre>Río Cem</nombre>
<nombre>Río  Criș</nombre>
<nombre>Río Corrib</nombre>
<nombre>Río Dal</nombre>
<nombre>Río Dâmbovița</nombre>
<nombre>Río Danubio</nombre>
<nombre>Río Dargle</nombre>
<nombre>Río Daugava</nombre>
<nombre>Río Desná</nombre>
<nombre>Río Dhoo</nombre>
<nombre>Río Dniéper</nombre>
<nombre>Río Dniéster</nombre>
<nombre>Río Don</nombre>
<nombre>Río Donets</nombre>
<nombre>Río Douglas</nombre>
<nombre>Río Drammen</nombre>
<nombre>Río Drava</nombre>
<nombre>Río Drin</nombre>
<nombre>Río Drina</nombre>
<nombre>Río Duero</nombre>
<nombre>Río Durance</nombre>
<nombre>Río Dviná Septentrional</nombre>
<nombre>Río Dyle</nombre>
<nombre>Río Ebro</nombre>
<nombre>Río Elba</nombre>
<nombre>Río Eno</nombre>
<nombre>Río Escalda</nombre>
<nombre>Río Fiumicello</nombre>
<nombre>Río Foyle</nombre>
```

```
<nombre>Río Garona</nombre>
<nombre>Río Gauja</nombre>
<nombre>Río Glomma</nombre>
<nombre>Río Göta</nombre>
<nombre>Río Gudenå</nombre>
<nombre>Río Guadalquivir</nombre>
<nombre>Río Guadiana</nombre>
<nombre>Río Gran Morava</nombre>
<nombre>Río Gudbrandsdalslågen</nombre>
<nombre>Río Haliacmon</nombre>
<nombre>Río Havel</nombre>
<nombre>Río Horýn</nombre>
<nombre>Río Ialomița</nombre>
<nombre>Río Ijssel</nombre>
<nombre>Río Ilek</nombre>
<nombre>Río Ilovlia</nombre>
<nombre>Río Indals</nombre>
<nombre>Río Inhul</nombre>
<nombre>Río Inhulets</nombre>
<nombre>Río Izhma</nombre>
<nombre>Río Jökulsá á Fjöllum</nombre>
<nombre>Río Jiu</nombre>
<nombre>Río Jopior</nombre>
<nombre>Río Jorol</nombre>
<nombre>Río Kanlı Dere</nombre>
<nombre>Río Kalix</nombre>
<nombre>Río Kem</nombre>
<nombre>Río Kemijoki</nombre>
<nombre>Río Kuma</nombre>
<nombre>Río Ljungan</nombre>
<nombre>Río Ljusnan</nombre>
<nombre>Río Loira</nombre>
<nombre>Río Lovat</nombre>
<nombre>Río Lúga</nombre>
<nombre>Río Lule</nombre>
<nombre>Río Marano</nombre>
<nombre>Río Marequia</nombre>
<nombre>Río Maritsa</nombre>
<nombre>Río Medvéditsa</nombre>
<nombre>Río Mezén</nombre>
<nombre>Río Miño</nombre>
<nombre>Río Morača</nombre>
<nombre>Río Morava</nombre>
<nombre>Río Morava meridional</nombre>
<nombre>Río Morava occidental</nombre>
<nombre>Río Mosa</nombre>
```

```
<nombre>Río Moscova</nombre>
<nombre>Río Mosela</nombre>
<nombre>Río Msta</nombre>
<nombre>Río Mura</nombre>
<nombre>Río Mureş</nombre>
<nombre>Río Narew</nombre>
<nombre>Río Narva</nombre>
<nombre>Río Neris</nombre>
<nombre>Río Nevá</nombre>
<nombre>Río Niemen</nombre>
<nombre>Río Nore</nombre>
<nombre>Río Numedalslågen</nombre>
<nombre>Río Óder</nombre>
<nombre>Río Oise</nombre>
<nombre>Río Oká</nombre>
<nombre>Río Olt</nombre>
<nombre>Río Onega</nombre>
<nombre>Río Oskol</nombre>
<nombre>Río Pasvikelva</nombre>
<nombre>Río Pechora</nombre>
<nombre>Río Peneo</nombre>
<nombre>Río Pilica</nombre>
<nombre>Río Pite</nombre>
<nombre>Río Po</nombre>
<nombre>Río Prípiat</nombre>
<nombre>Río Prut</nombre>
<nombre>Río Psel</nombre>
<nombre>Río Raut</nombre>
<nombre>Río Rhin</nombre>
<nombre>Ródano</nombre>
<nombre>Río Ros'</nombre>
<nombre>Río Rupel</nombre>
<nombre>Río Sal</nombre>
<nombre>Río Salzach</nombre>
<nombre>Río Samara</nombre>
<nombre>Río San</nombre>
<nombre>Río San Marino</nombre>
<nombre>Río Sava</nombre>
<nombre>Río Segre</nombre>
<nombre>Río Seim</nombre>
<nombre>Río Semeni</nombre>
<nombre>Río Sena</nombre>
<nombre>Río Severn</nombre>
<nombre>Río Siret</nombre>
<nombre>Río Skellefte</nombre>
<nombre>Río Sluch</nombre>
```

```
<nombre>Río Someș</nombre>
<nombre>Río Spree</nombre>
<nombre>Río Styr</nombre>
<nombre>Río Sújona</nombre>
<nombre>Río Sula</nombre>
<nombre>Río Tajo</nombre>
<nombre>Río Támesis</nombre>
<nombre>Río Tana</nombre>
<nombre>Río Terek</nombre>
<nombre>Río Téteriv</nombre>
<nombre>Río Tíber</nombre>
<nombre>Río Timis</nombre>
<nombre>Río Tisza</nombre>
<nombre>Río Trent</nombre>
<nombre>Río Tuloma</nombre>
<nombre>Río Ume</nombre>
<nombre>Río Una</nombre>
<nombre>Río Ural</nombre>
<nombre>Río Váh</nombre>
<nombre>Río Valira del Norte</nombre>
<nombre>Río Vardar</nombre>
<nombre>Río Vashka</nombre>
<nombre>río Velíkaya</nombre>
<nombre>Río Venta</nombre>
<nombre>Río Víchegda</nombre>
<nombre>Río Vilaine</nombre>
<nombre>Río Vim</nombre>
<nombre>Río Viosa</nombre>
<nombre>Río Vístula</nombre>
<nombre>Río Vodla</nombre>
<nombre>Río Volga</nombre>
<nombre>Río Vóljov</nombre>
<nombre>Río Vorskla</nombre>
<nombre>Río Vuoksi</nombre>
<nombre>Río Waal</nombre>
<nombre>Río Weser</nombre>
<nombre>Río Wieprz</nombre>
<nombre>Río Yug</nombre>
<nombre>Río Zeta</nombre>
```

Visualizar el nombre de los países cuya superficie sea inferior a 300 km²

- **`doc("europa.xml")//paises/pais[superficie<300]/nombre`**

- ```
  for $x in doc("europa.xml")//paises/pais
  where $x/superficie<300
  return $x/nombre
  ```

```
<nombre>Andorra</nombre>
<nombre>Ciudad Vaticano</nombre>
<nombre>Gibraltar</nombre>
<nombre>Guernsey</nombre>
<nombre>Jersey</nombre>
<nombre>Liechtenstein</nombre>
<nombre>Mónaco</nombre>
<nombre>San Marino</nombre>
```

La siguiente consulta devuelve los nombres de los países cuyo estado es una República Parlamentaria.

```
for $x in doc("europa.xml")//paises/pais
where $x/estado='República Parlamentaria'
return $x/nombre
```

```
<nombre>Albania</nombre>
<nombre>Bulgaria</nombre>
<nombre>Chequia</nombre>
<nombre>Croacia</nombre>
<nombre>Eslovaquia</nombre>
<nombre>Eslovenia</nombre>
<nombre>Estonia</nombre>
<nombre>Finlandia</nombre>
<nombre>Irlanda</nombre>
<nombre>Islandia</nombre>
<nombre>Italia</nombre>
<nombre>Letonia</nombre>
<nombre>Lituania</nombre>
<nombre>Macedonia</nombre>
```
```

```
<nombre>Malta</nombre>
<nombre>Moldavia</nombre>
<nombre>Montenegro</nombre>
<nombre>Polonia</nombre>
<nombre>San Marino</nombre>
<nombre>Serbia</nombre>
```

Para devolver más de un resultado:

Ejemplo: Visualizar el nombre y la capital de todos los países del fichero europa.xml

```
for $x in doc("europa.xml")//paises/pais
let $b:=$x/nombre
let $c:=$x/capital
return
<Salida>
          {$b}
          {$c}
</Salida>
```

Resultado:

```
<Salida>
     <nombre>Albania</nombre>
     <capital>Tirana</capital>
</Salida>
<Salida>
      <nombre>Alemania</nombre>
      <capital>Berlín</capital>
</Salida>
<Salida>
     <nombre>Andorra</nombre>
     <capital>Andorra la Vella</capital>
</Salida>
<Salida>
     <nombre>Austria</nombre>
     <capital>Viena</capital>
</Salida>
<Salida>
      <nombre>Bélgica</nombre>
```

```
      <capital>Bruselas</capital>
</Salida>

<Salida>
      <nombre>Bielorrusia</nombre>
      <capital>Minsk</capital>
</Salida>
<Salida>
      <nombre>Bosnia y Herzegovina</nombre>
      <capital>Sarajevo</capital>
</Salida>
<Salida>
      <nombre>Bulgaria</nombre>
      <capital>Sofía</capital>
</Salida>
<Salida>
      <nombre>Chequia</nombre>
      <capital>Praga</capital>
</Salida>
<Salida>
      <nombre>Chipre</nombre>
      <capital>Nicosia</capital>
</Salida>
<Salida>
      <nombre>Ciudad Vaticano</nombre>
      <capital>Ciudad Vaticano</capital>
</Salida>
<Salida>
      <nombre>Croacia</nombre>
      <capital>Zagreb</capital>
</Salida>
<Salida>
      <nombre>Dinamarca</nombre>
      <capital>Copenhague</capital>
</Salida>
<Salida>
      <nombre>Eslovaquia</nombre>
      <capital>Bratislava</capital>
</Salida>
<Salida>
      <nombre>Eslovenia</nombre>
      <capital>Liubliana</capital>
</Salida>

<Salida>
      <nombre>España</nombre>
```

```
      <capital>Madrid</capital>
</Salida>
<Salida>
      <nombre>Estonia</nombre>
      <capital>Tallín</capital>
</Salida>
<Salida>
      <nombre>Finlandia</nombre>
      <capital>Helsinki</capital>
</Salida>
<Salida>
      <nombre>Francia</nombre>
      <capital>París</capital>
</Salida>
<Salida>
      <nombre>Gibraltar</nombre>
      <capital>Gibraltar</capital>
</Salida>
<Salida>
      <nombre>Grecia</nombre>
      <capital>Atenas</capital>
</Salida>
<Salida>
      <nombre>Guernsey</nombre>
      <capital>Puerto de San Pedro</capital>
</Salida>
<Salida>
      <nombre>Hugría</nombre>
      <capital>Budapest</capital>
</Salida>
<Salida>
      <nombre>Irlanda</nombre>
      <capital>Dublín</capital>
</Salida>
<Salida>
      <nombre>Islandia</nombre>
      <capital>Reikiavik</capital>
</Salida>

<Salida>
      <nombre>Italia</nombre>
      <capital>Roma</capital>
</Salida>
<Salida>
      <nombre>Jersey</nombre>
      <capital>Saint Helier</capital>
```

```
</Salida>
<Salida>
     <nombre>Letonia</nombre>
     <capital>Riga</capital>
</Salida>
<Salida>
     <nombre>Liechtenstein</nombre>
     <capital>Vaduz</capital>
</Salida>
<Salida>
     <nombre>Lituania</nombre>
     <capital>Vilna</capital>
</Salida>
<Salida>
     <nombre>Luxemburgo</nombre>
     <capital>Luxemburgo</capital>
</Salida>
<Salida>
     <nombre>Macedonia</nombre>
     <capital>Skopie</capital>
</Salida>
<Salida>
     <nombre>Malta</nombre>
     <capital>La Valeta</capital>
</Salida>
<Salida>
     <nombre>Man</nombre>
     <capital>Douglas</capital>
</Salida>
<Salida>
     <nombre>Moldavia</nombre>
     <capital>Chisináu</capital>
</Salida>

<Salida>
     <nombre>Mónaco</nombre>
     <capital>Ciudad de Mónaco</capital>
</Salida>
<Salida>
     <nombre>Montenegro</nombre>
     <capital>Podgorica</capital>
</Salida>
<Salida>
     <nombre>Noruega</nombre>
     <capital>Oslo</capital>
</Salida>
```

```
<Salida>
     <nombre>Paises Bajos</nombre>
     <capital>Ámsterdam</capital>
</Salida>
<Salida>
     <nombre>Polonia</nombre>
     <capital>Varsovia</capital>
</Salida>
<Salida>
     <nombre>Portugal</nombre>
     <capital>Lisboa</capital>
</Salida>
<Salida>
     <nombre>Reino Unido</nombre>
     <capital>Lóndres</capital>
</Salida>
<Salida>
     <nombre>Rumanía</nombre>
     <capital>Bucarest</capital>
</Salida>
<Salida>
     <nombre>Rusia</nombre>
     <capital>Moscú</capital>
</Salida>
<Salida>
     <nombre>San Marino</nombre>
     <capital>San Marino</capital>
</Salida>

<Salida>
     <nombre>Serbia</nombre>
     <capital>Belgrado</capital>
</Salida>
<Salida>
     <nombre>Suecia</nombre>
     <capital>Estocolmo</capital>
</Salida>
<Salida>
     <nombre>Suiza</nombre>
     <capital>Berna</capital>
</Salida>
<Salida>
     <nombre>Ucrania</nombre>
     <capital>Kiev</capital>
</Salida>
```

Los resultados de las consultas salen siempre con las etiquetas XML, si queremos obtener un listado sin las etiquetas, utilizaremos la función data().

La siguiente consulta devuelve los nombres de los países cuyo estado es una República Federal.

```
for $x in doc("europa.xml")//paises/pais
where $x/estado='República Federal'
return
data($x/nombre)
```

Resultado:

```
Alemania
Austria
Bosnia y Herzegovina
```

Podemos unir dentro de **XQuery** etiquetas **HTML** y expresiones o clausulas **FLWOR**, pero la parte que se procesa como consulta debe ir obligatoriamente entre **{ }**

Ejemplo: visualizar en modo de lista los nombres de los países de Europa cuya superficie es menor que 300Km^2

```
<html>
     <head>
          <title> ejemplo en modo lista</title>
     </head>
     <body>
        <p>Paises con superficie menor de 300Km<sup>2</sup></p>
        <ul>
           {
           for $x in doc("europa.xml")//paises/pais
           where $x/superficie<300
           order by $x/nombre
           return
                <li>{data($x/nombre)}</li>
           }
        </ul>
     </body></html>
```

Obtenemos como salida:

```
<html>
```

```
      <head>
            <title> ejemplo en modo lista</title>
      </head>
      <body>
            <p>Paises con superficie menor de
300Km<sup>2</sup></p>
            <ul>
                  <li>Andorra</li>
                  <li>Ciudad Vaticano</li>
                  <li>Gibraltar</li>
                  <li>Guernsey</li>
                  <li>Jersey</li>
                  <li>Liechtenstein</li>
                  <li>Mónaco</li>
                  <li>San Marino</li>
            </ul>
      </body>
</html>
```

Al abrir este fichero **html** la salida sería la siguiente:

Paises con superficie menor de 300Km^2

- Andorra
- Ciudad Vaticano
- Gibraltar
- Guernsey
- Jersey
- Liechtenstein
- Mónaco
- San Marino

Ejemplo: visualizar en modo de tabla los nombres de los países de Europa cuya superficie es menor que 300Km^2

```
<html>
     <head>
            <title> ejemplo en modo tabla</title>
     </head>
     <body>
            <table border="1">
               <tr style="background-color:orange">
                 <td>Paises con superficie menor de
               300Km<sup>2</sup></td>
```

```
            </tr>

        {

            for $x in doc("europa.xml")//paises/pais
            where $x/superficie<300
            order by $x/nombre
            return
                <tr>
                    <td>{data($x/nombre)}</td>
                </tr>

            }

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

```
<html>
     <head>
        <title> ejemplo en modo tabla</title>
      </head>
      <body>
        <table border="1">
                  <tr style="background-color:orange">
                        <td>Paises    con    superficie    menor    de
        300Km<sup>2</sup></td>
            </tr>
            <tr>
                  <td>Andorra</td>
            </tr>
            <tr>
                  <td>Ciudad Vaticano</td>
            </tr>
            <tr>
                  <td>Gibraltar</td>
            </tr>
            <tr>
                  <td>Guernsey</td>
            </tr>
            <tr>
                  <td>Jersey</td>
            </tr>
            <tr>
                  <td>Liechtenstein</td>
            </tr>
            <tr>
                  <td>Mónaco</td>
            </tr>
            <tr>
                  <td>San Marino</td>
            </tr>
            </table>
     </body>
</html>
```

Si abrimos el fichero **html**, nos saldrá lo siguiente:

Paises con superficie menor de 300Km^2
Andorra
Ciudad Vaticano
Gibraltar
Guernsey
Jersey
Liechtenstein
Mónaco
San Marino

Consultas sencillas

1. **Visualizar el nombre, capital y superficie de los paises cuya superficie es mayor o igual que 25000 y menor que 250000 km^2 ordenados por el nombre del país.**

```
<html>
	<head>
		<title> ejemplo 1</title>
	</head>
	<body>
		<table border="1">
			<tr style="background-color:green">
				<td>Nombre</td>
				<td>Capital</td>
				<td>Superficie</td>
			</tr>

		{
		for $x in doc("europa.xml")//paises/pais
		where $x/superficie >=25000 and $x/superficie<250000
		order by $x/superficie
		return
			<tr>
				<td>{$x/nombre/data()}</td>
				<td>{$x/capital/data()}</td>
				<td>{$x/superficie/data()}</td>
			</tr>
		}
```

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

El resultado en HTML sería:

```
<html>
      <head>
            <title> ejemplo 1</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color:green">
                        <td>Nombre</td>
                        <td>Capital</td>
                        <td>Superficie</td>
                  </tr>
                  <tr>
                        <td>Macedonia</td>
                        <td>Skopie</td>
                        <td>00025713</td>
                  </tr>
                  <tr>
                        <td>Albania</td>
                        <td>Tirana</td>
                        <td>00028748</td>
                  </tr>
                  <tr>
                        <td>Bélgica</td>
                        <td>Bruselas</td>
                        <td>00030528</td>
                  </tr>
                  <tr>
                        <td>Moldavia</td>
                        <td>Chisináu</td>
                        <td>00033851</td>
                  </tr>
                  <tr>
                        <td>Suiza</td>
                        <td>Berna</td>
                        <td>00041277</td>
                  </tr>
                  <tr>
```

```
                <td>Paises Bajos</td>
                <td>Ámsterdam</td>
                <td>00041543</td>
            </tr>
            <tr>
                <td>Dinamarca</td>
                <td>Copenhague</td>
                <td>00043094</td>
            </tr>
            <tr>
                <td>Estonia</td>
                <td>Tallín</td>
                <td>00045228</td>
        </tr>
            <tr>
                <td>Eslovaquia</td>
                <td>Bratislava</td>
                <td>00049035</td>
            </tr>
            <tr>
                <td>Bosnia y Herzegovina</td>
                <td>Sarajevo</td>
                <td>00051797</td>
            </tr>
            <tr>
                <td>Croacia</td>
                <td>Zagreb</td>
                <td>00056594</td>
            </tr>

            <tr>
                <td>Letonia</td>
                <td>Riga</td>
                <td>00064589</td>
            </tr>
            <tr>
                <td>Lituania</td>
                <td>Vilna</td>
                <td>00065300</td>
            </tr>
            <tr>
                <td>Irlanda</td>
                <td>Dublín</td>
                <td>00070273</td>
            </tr>
        <tr>
```

```
                <td>Chequia</td>
                <td>Praga</td>
                <td>00078867</td>
            </tr>
            <tr>
                <td>Austria</td>
                <td>Viena</td>
                <td>00083871</td>
        </tr>
            <tr>
                <td>Serbia</td>
                <td>Belgrado</td>
                <td>00088361</td>
            </tr>
            <tr>
                <td>Portugal</td>
                <td>Lisboa</td>
                <td>00092090</td>
            </tr>
            <tr>
                <td>Hugría</td>
                <td>Budapest</td>
                <td>00093030</td>
            </tr>

            <tr>
                <td>Islandia</td>
                <td>Reikiavik</td>
                <td>00103000</td>
            </tr>
            <tr>
                <td>Bulgaria</td>
                <td>Sofía</td>
                <td>00110879</td>
            </tr>
            <tr>
                <td>Grecia</td>
                <td>Atenas</td>
                <td>00131957</td>
            </tr>
            <tr>
                <td>Bielorrusia</td>
                <td>Minsk</td>
                <td>00207600</td>
            </tr>
            <tr>
```

```
                        <td>Rumanía</td>
                        <td>Bucarest</td>
                        <td>00238391</td>
                    </tr>
                    <tr>
                        <td>Reino Unido</td>
                        <td>Lóndres</td>
                        <td>00243610</td>
                    </tr>
            </table>
        </body>
</html>
```

En un navegador obtenemos la siguiente salida:

Nombre	Capital	Superficie
Macedonia	Skopie	00025713
Albania	Tirana	00028748
Bélgica	Bruselas	00030528
Moldavia	Chisináu	00033851
Suiza	Berna	00041277
Paises Bajos	Ámsterdam	00041543
Dinamarca	Copenhague	00043094
Estonia	Tallín	00045228
Eslovaquia	Bratislava	00049035
Bosnia y Herzegovina	Sarajevo	00051797
Croacia	Zagreb	00056594
Letonia	Riga	00064589
Lituania	Vilna	00065300
Irlanda	Dublín	00070273
Chequia	Praga	00078867
Austria	Viena	00083871
Serbia	Belgrado	00088361
Portugal	Lisboa	00092090
Hugría	Budapest	00093030
Islandia	Reikiavik	00103000
Bulgaria	Sofía	00110879
Grecia	Atenas	00131957
Bielorrusia	Minsk	00207600
Rumanía	Bucarest	00238391
Reino Unido	Lóndres	00243610

2. **Visualizar el nombre la capital y la población de todos los países que tienen una población superior a 35 millones de habitantes**

```
<html>
      <head>
            <title> ejemplo 2</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color:red">
                        <td>Nombre</td>
                        <td>Capital</td>
                        <td>Población</td>
                  </tr>

            {
                for $x in doc("europa.xml")//paises/pais
                where $x/poblacion >35000000
                order by $x/poblacion
                return
                 <tr>
                        <td>{$x/nombre/data()}</td>
                        <td>{$x/capital/data()}</td>
                        <td>{$x/poblacion/data()}</td>
                 </tr>

            }

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

La salida es:

```
<html>
      <head>
            <title> ejemplo 2</title>
       </head>
      <body>
            <table border="1">
                        <tr style="background-color:red">
                              <td>Nombre</td>
                              <td>Capital</td>
```

```
                        <td>Población</td>
                    </tr>
                    <tr>
                        <td>Polonia</td>
                        <td>Varsovia</td>
                        <td>037840001</td>
                </tr>
                    <tr>
                        <td>Ucrania</td>
                        <td>Kiev</td>
                        <td>039510726</td>
                    </tr>
                    <tr>
                        <td>España</td>
                        <td>Madrid</td>
                        <td>047398695</td>
                    </tr>
                    <tr>
                        <td>Italia</td>
                        <td>Roma</td>
                        <td>060257566</td>
                    </tr>
                    <tr>
                        <td>Francia</td>
                        <td>París</td>
                        <td>067656682</td>
                    </tr>
                    <tr>
                        <td>Reino Unido</td>
                        <td>Lóndres</td>
                        <td>067747826</td>
                    </tr>
                    <tr>
                        <td>Alemania</td>
                        <td>Berlín</td>
                        <td>083155031</td>
                    </tr>
                    <tr>
                        <td>Rusia</td>
                        <td>Moscú</td>
                        <td>145558000</td>
                    </tr>
            </table>
        </body>
</html>
```

Abriendo el fichero **html** obtenemos:

Nombre	Capital	Población
Polonia	Varsovia	037840001
Ucrania	Kiev	039510726
España	Madrid	047398695
Italia	Roma	060257566
Francia	París	067656682
Reino Unido	Lóndres	067747826
Alemania	Berlín	083155031
Rusia	Moscú	145558000

3. **Visualizar los distintos Estados de los Países:**

```
<html>
      <head>
            <title> ejemplo 2</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color:blue">
                        <td>Nombre de los estados de los
                  paises</td>
                  </tr>

      {
for $x in distinct-values(doc("europa.xml")//paises/pais/estado)
return
      <tr>
            <td>{data($x)}</td>
      </tr>
      }

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

El resultado sería:

```
<html>
	<head>
		<title> ejemplo 2</title>
	</head>
	<body>
		<table border="1">
			<tr style="background-color:blue">
				<td>Nombre de los estados de los paises</td>
			</tr>
			<tr>
				<td>República Parlamentaria</td>
			</tr>
			<tr>
				<td>República Federal</td>
			</tr>
			<tr>
				<td>Principado</td>
			</tr>
			<tr>
				<td>Monarquía Federal Parlamentaria</td>
			</tr>
			<tr>
				<td>República Presidencialista</td>
			</tr>
			<tr>
				<td>Monarquía Absoluta, Electiva y Teocrática</td>
			</tr>
			<tr>
				<td>Monarquía Parlamentaria</td>
			</tr>
			<tr>
				<td>República Semipresidencialista Unitaria</td>
			</tr>
```

```
                <tr>
                    <td>República Parlamentaria Unitaria</td>
                </tr>
                 <tr>
                    <td>Monarquía Constitucional Parlamentaria</td>
                </tr>
                <tr>
                    <td>Dependencia de la Corona Británica</td>
                </tr>
                <tr>
                    <td>Monarquía Constitucional</td>
                </tr>
                <tr>
                    <td>República Unitaria Semipresidencialista</td>
                </tr>
                <tr>
                    <td>República Semipresidencialista</td>
                </tr>
                <tr>
                    <td>República Federal Semiparlamentaria</td>
                </tr>
                <tr>
                    <td>República Parlamentaria Confederal</td>
                </tr>
            </table>
        </body>
</html>
```

Nombre de los estados de los paises
República Parlamentaria
República Federal
Principado
Monarquía Federal Parlamentaria
República Presidencialista
Monarquía Absoluta, Electiva y Teocrática
Monarquía Parlamentaria
República Semipresidencialista Unitaria
República Parlamentaria Unitaria
Monarquía Constitucional Parlamentaria
Dependencia de la Corona Británica
Monarquía Constitucional
República Unitaria Semipresidencialista
República Semipresidencialista
República Federal Semiparlamentaria
República Parlamentaria Confederal

4. **Visualizar la población de los países que tengan menos de 1 millón de habitantes así como su población aumentada en un 10%.**

```
<html>
	<head>
		<title> ejemplo 2</title>
	</head>
	<body>
		<table border="1">
			<tr style="background-color:yellow">
				<td>Población</td>
				<td>Población aumentada en un 10%</td>

			</tr>
		{
		for $x in doc("europa.xml")//paises/pais
		where $x/poblacion<1000000
		order by $x/poblacion
		let $poblacion_nueva:=round($x/poblacion*1.1,2)
		return
			<tr>
				<td>{$x/poblacion/data()}</td>
```

```
                    <td>{$poblacion_nueva}</td>
                </tr>

            }
            </table>
      </body>
</html>
```

El resultado en HTML es:

```
<html>
      <head>
            <title> ejemplo 2</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color:yellow">
                        <td>Población</td>
                        <td>Población aumentada en un 10%</td>
                  </tr>
                  <tr>
                        <td>000000825</td>
                        <td>907.5</td>
                  </tr>
                  <tr>
                        <td>000022256</td>
                        <td>24481.6</td>
                  </tr>
                  <tr>
                        <td>000033553</td>
                        <td>36908.3</td>
                  </tr>
                  <tr>
                        <td>000033691</td>
                        <td>37060.1</td>
                  </tr>
                  <tr>
                        <td>000037540</td>
                        <td>41294</td>
                  </tr>
                  <tr>
                        <td>000038100</td>
                        <td>41910</td>
                  </tr>
```

```
                <tr>
                    <td>000038749</td>
                    <td>42623.9</td>
                </tr>
                <tr>
                    <td>000062792</td>
                    <td>69071.2</td>
                </tr>
                <tr>
                    <td>000085032</td>
                    <td>93535.2</td>
                </tr>
                <tr>
                    <td>000366425</td>
                    <td>403067.5</td>
                </tr>
                <tr>
                    <td>000525285</td>
                    <td>577813.5</td>
                </tr>
                <tr>
                    <td>000621718</td>
                    <td>683889.8</td>
                </tr>
                <tr>
                    <td>000626108</td>
                    <td>688718.8</td>
                </tr>
                <tr>
                    <td>000896007</td>
                    <td>985607.7</td>
                </tr>
            </table>
    </body>
</html>
```

Población	Población aumentada en un 10%
000000825	907.5
000022256	24481.6
000033553	36908.3
000033691	37060.1
000037540	41294
000038100	41910
000038749	42623.9
000062792	69071.2
000085032	93535.2
000366425	403067.5
000525285	577813.5
000621718	683889.8
000626108	688718.8
000896007	985607.7

Consultas con funciones

1. **Visualizar el nombre de los ríos que empiezan por J**

```
<html>
	<head>
		<title> ejemplo 2</title>
	</head>
	<body>
		<table border="1">
			<tr style="background-color:yellow">
				<td>Nombre del Río</td>
				<td>Comprobación que el río empieza por
			J</td>

			</tr>

		{
			for $x in  doc("europa.xml")//rios/rio
			where  substring($x/nombre,5,1)="J"
			return
				<tr>
					<td>{$x/nombre/data()}</td>
					<td>{substring($x/nombre,5,1)}</td>
				</tr>

		}

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

Obtenemos el siguiente resultado:

```
<html>
	<head>
		<title> ejemplo 2</title>
	</head>
	<body>
		<table border="1">
			<tr style="background-color:yellow">
				<td>Nombre del Río</td>
```

```
                    <td>Comprobación que el río empieza por
J</td>
                </tr>
                <tr>
                    <td>Río Jökulsá á Fjöllum</td>
                    <td>J</td>
                </tr>
                <tr>
                    <td>Río Jiu</td>
                    <td>J</td>
                </tr>
                <tr>
                    <td>Río Jopior</td>
                    <td>J</td>
                </tr>
                <tr>
                    <td>Río Jorol</td>
                    <td>J</td>
                </tr>
            </table>
        </body>
</html>
```

Nombre del Río	Comprobación que el río empieza por J
Río Jökulsá á Fjöllum	J
Río Jiu	J
Río Jopior	J
Río Jorol	J

2. **Visualizar las ciudades cuyo nombre empieza por N, así como el número de caracteres que tiene.**

```
<html>
    <head>
        <title> ejemplo 2</title>
    </head>
    <body>
```

```
            <table border="1">
                  <tr style="background-color:Moccasin">
                        <td>Nombre de la ciudad</td>
                        <td>Número de caracteres que tiene</td>
                  </tr>


      {
      for $x in  doc("europa.xml")//ciudades/ciudad
      where substring($x/nombre,1,1)= "N"
      order by  $x/nombre
      let $b:=string-length($x/nombre)
      return
            <tr>
                  <td>{$x/nombre/data()}</td>
                  <td>{$b}</td>
            </tr>
      }

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

Obtenemos el siguiente resultado:

```
<html>
      <head>
            <title> ejemplo 2</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color:Moccasin">
                        <td>Nombre de la ciudad</td>
                        <td>Número de caracteres que tiene</td>
                  </tr>
                  <tr>
                        <td>Namur</td>
                        <td>5</td>
                  </tr>
                  <tr>
                        <td>Nantes</td>
                        <td>6</td>
                  </tr>
                  <tr>
```

```
      <td>Narián-Mar</td>
      <td>10</td>
</tr>
<tr>
      <td>Narva</td>
      <td>5</td>
</tr>
<tr>
      <td>Naváshino</td>
      <td>9</td>
</tr>
<tr>
      <td>Neuss</td>
      <td>5</td>
</tr>
<tr>
      <td>Nicolaiev</td>
      <td>9</td>
</tr>

<tr>
      <td>Nicsic</td>
      <td>6</td>
</tr>
<tr>
      <td>Nikolsk</td>
      <td>7</td>
</tr>
<tr>
      <td>Nimega</td>
      <td>6</td>
</tr>
<tr>
      <td>Nisco</td>
      <td>5</td>
</tr>
<tr>
      <td>Nizhni Nóvgorod</td>
      <td>15</td>
</tr>
<tr>
      <td>Niš</td>
      <td>3</td>
</tr>
<tr>
      <td>Nore og Uvdal</td>
```

```
                    <td>13</td>
                </tr>
                <tr>
                    <td>Nottingham</td>
                    <td>10</td>
                </tr>
                <tr>
                    <td>Novi Grad</td>
                    <td>9</td>
                </tr>
                <tr>
                    <td>Novi Sad</td>
                    <td>8</td>
                </tr>

                <tr>
                    <td>Novodvinsk</td>
                    <td>10</td>
                </tr>
                <tr>
                    <td>Novohrad-Volynskyi</td>
                    <td>18</td>
                </tr>
                <tr>
                    <td>Năsăud</td>
                    <td>6</td>
                </tr>
            </table>
    </body>
</html>
```

Nombre de la ciudad	Número de caracteres que tiene
Namur	5
Nantes	6
Narián-Mar	10
Narva	5
Naváshino	9
Neuss	5
Nicolaiev	9
Nicsic	6
Nikolsk	7
Nimega	6
Nisco	5
Nizhni Nóvgorod	15
Niš	3
Nore og Uvdal	13
Nottingham	10
Novi Grad	9
Novi Sad	8
Novodvinsk	10
Novohrad-Volynskyi	18
Năsăud	6

3. **Visualizar el nombre y la altura de las montañas cuyo nombre empieza por V**

```
<html>
     <head>
          <title> ejemplo 2</title>
     </head>
     <body>
          <table border="1">
               <tr style="background-color:Olive">
                    <td>Nombre de la Montaña</td>
                    <td>Comprobación de que su nombre empieza
               por V</td>
                    <td>Altura de la Montaña</td>
               </tr>

          {
```

```
            for $x in
            doc("europa.xml")//montanas/montana/nombre
            where starts-with(data($x),'V')
            return
                <tr>
                      <td>{$x/data()}</td>
                      <td>{substring($x,1,1)}</td>
                      <td>{$x/../altura/data()}</td>
                </tr>

          }

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

Obtenemos como resultado:

```
<html>
     <head>
          <title> ejemplo 2</title>
     </head>
     <body>
          <table border="1">
               <tr style="background-color:Olive">
                    <td>Nombre de la Montaña</td>
                    <td>Comprobación de que su nombre empieza
por V</td>
                    <td>Altura de la Montaña</td>
               </tr>
               <tr>
                    <td>Vaalserberg</td>
                    <td>V</td>
                    <td>323</td>
               </tr>
               <tr>
                    <td>Vlašić</td>
                    <td>V</td>
                    <td>1943</td>
               </tr>
               <tr>
                    <td>Vihren</td>
                    <td>V</td>
                    <td>2914</td>
```

```
                </tr>
                <tr>
                      <td>Velká Deštná</td>
                      <td>V</td>
                      <td>1115</td>
                </tr>
                <tr>
                      <td>Vistofte</td>
                      <td>V</td>
                      <td>169</td>
                </tr>
                <tr>
                      <td>Väike Munamägi</td>
                      <td>V</td>
                      <td>208</td>
                </tr>
                <tr>
                      <td>Vestrahorn</td>
                      <td>V</td>
                      <td>0889</td>
                </tr>
                <tr>
                      <td>Visitor</td>
                      <td>V</td>
                      <td>2209</td>
                </tr>
                <tr>
                      <td>Vaalserberg </td>
                      <td>V</td>
                      <td>321</td>
                </tr>
                <tr>
                      <td>Vardenik</td>
                      <td>V</td>
                      <td>1874</td>
                </tr>
          </table>
     </body>
</html>
```

Nombre de la Montaña	Comprobación de que su nombre empieza por V	Altura de la Montaña
Vaalserberg	V	323
Vlašić	V	1943
Vihren	V	2914
Velká Deštná	V	1115
Vistofte	V	169
Väike Munamägi	V	208
Vestrahorn	V	0889
Visitor	V	2209
Vaalserberg	V	321
Vardenik	V	1874

4. **Visualizar la cuarta y quinta letra de los nombres de las islas que empiezan por T**

```
<html>
      <head>
            <title> ejemplo 2</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color: Bisque">
                        <td>Nombre de la Islas que empiezan por
                  T</td>
                        <td>posiciones cuarta y quinta del nombre
                  de las islas</td>
                  </tr>
            {
                  for $x in  doc("europa.xml")//islas/isla/nombre
                  where substring($x,1,1)= "T"
                  let $ti:=substring($x,4,2)
                  return
                        <tr>
                              <td>{$x/data()}</td>
                              <td>{$ti}</td>
                        </tr>
            }

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

Obtenemos:

```
<html>
      <head>
            <title> ejemplo 2</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color: Bisque">
                        <td>Nombre de la Islas que empiezan por
T</td>
                        <td>posiciones cuarta y quinta del nombre
de las islas</td>
                  </tr>
```

```
<tr>
      <td>Tåsinge</td>
      <td>in</td>
</tr>
<tr>
      <td>Tenerife</td>
      <td>er</td>
</tr>
<tr>
      <td>Tasos</td>
      <td>os</td>
</tr>
<tr>
      <td>Tinos</td>
      <td>os</td>
</tr>
<tr>
      <td>Tilos</td>
      <td>os</td>
</tr>
<tr>
      <td>Tysnesøy</td>
      <td>ne</td>
</tr>
<tr>
      <td>Tjeldøya</td>
      <td>ld</td>
</tr>


<tr>
      <td>Tustna</td>
      <td>tn</td>
</tr>
<tr>
      <td>Tunøya</td>
      <td>øy</td>
</tr>
<tr>
      <td>Tromsøya</td>
      <td>ms</td>
</tr>
<tr>
      <td>Texel</td>
      <td>el</td>
</tr>
```

```
                <tr>
                      <td>Terschelling</td>
                      <td>sc</td>
                </tr>
                <tr>
                      <td>Tiree</td>
                      <td>ee</td>
                </tr>
                <tr>
                      <td>Tierra de Jorge</td>
                      <td>rr</td>
                </tr>
                <tr>
                      <td>Tierra de Wilczek</td>
                      <td>rr</td>
                </tr>
                <tr>
                      <td>Tierra de Alexandra</td>
                      <td>rr</td>
                </tr>
                <tr>
                      <td>Tierra del Príncipe Rodolfo</td>
                      <td>rr</td>
                </tr>

                <tr>
                      <td>Tjörn</td>
                      <td>rn</td>
                </tr>
                <tr>
                      <td>Tosterön-Aspön</td>
                      <td>te</td>
                </tr>
                <tr>
                      <td>Torsö</td>
                      <td>sö</td>
                </tr>
          </table>
     </body>
</html>
```

Nombre de la Islas que empiezan por T	posiciones cuarta y quinta del nombre de las islas
Tåsinge	in
Tenerife	er
Tasos	os
Tinos	os
Tilos	os
Tysnesøy	ne
Tjeldøya	ld
Tustna	tn
Tunøya	øy
Tromsøya	ms
Texel	el
Terschelling	sc
Tiree	ee
Tierra de Jorge	rr
Tierra de Wilczek	rr
Tierra de Alexandra	rr
Tierra del Príncipe Rodolfo	rr
Tjörn	rn
Tosterön-Aspön	te
Torsö	sö

5. **Visualizar el nombre, población y extensión de todas las ciudades cuyo nombre termina por u**

```
<html>
      <head>
            <title> ejemplo 2</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color: Beige">
                        <td>Nombre de las ciudades que terminan en
u</td>
                        <td>Población</td>
                        <td>Extensión</td>
                  </tr>

{
  for $x in  doc("europa.xml")//ciudades/ciudad
  where  substring($x/nombre,string-length($x/nombre),1)="u"
  return
      <tr>
            <td>{$x/nombre/data()}</td>
            <td>{$x/poblacion/data()}</td>
            <td>{$x/extension/data()}</td>
      </tr>
}

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

Se obtiene como resultado:

```
<html>
      <head>
            <title> ejemplo 2</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color: Beige">
                        <td>Nombre de las ciudades que terminan en
u</td>
                        <td>Población</td>
                        <td>Extensión</td>
```

```
</tr>
<tr>
      <td>Passau</td>
      <td>0190504</td>
      <td>1530.4</td>
</tr>
<tr>
      <td>Decines Charpieu</td>
      <td>0028930</td>
      <td>017.01</td>
</tr>
<tr>
      <td>Meyzieu</td>
      <td>0031438</td>
      <td>023.01</td>
</tr>
<tr>
      <td>Montalieu-Vercieu</td>
      <td>0002178</td>
      <td>008.66</td>
</tr>
<tr>
      <td>Chisinau</td>
      <td>681000</td>
      <td>123</td>
</tr>
<tr>
      <td>Ringebu</td>
      <td>004462</td>
      <td>1248</td>
</tr>
<tr>
      <td>Bacău</td>
      <td>0196883</td>
      <td>043.19</td>
</tr>
<tr>
      <td>Buzău</td>
      <td>0134227</td>
      <td>081.3</td>
</tr>
<tr>
      <td>Giurgiu</td>
      <td>0073587</td>
      <td>046.94</td>
</tr>
```

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

Nombre de las ciudades que terminan en u	Población	Extensión
Passau	0190504	1530.4
Decines Charpieu	0028930	017.01
Meyzieu	0031438	023.01
Montalieu-Vercieu	0002178	008.66
Chisinau	681000	123
Ringebu	004462	1248
Bacău	0196883	043.19
Buzău	0134227	081.3
Giurgiu	0073587	046.94

6. **Visualizar los nombres, así como las tres ultimas letras de las islas que terminan en e.**

```
<html>
      <head>
            <title> ejemplo 2</title>
      </head>
      <body>
            <table border="1">
                  <tr style="background-color: Coral">
```

```
                    <td>Nombre de las islas que terminan en
                e</td>
                    <td>Tres ultimas letras de los nombres de
                las islas</td>
                </tr>
    {
    for $x in  doc("europa.xml")//islas/isla
    where  substring($x/nombre,string-length($x/nombre),1)="e"
    order by $x/nombre
    let $ti:=substring($x/nombre,string-length($x/nombre)-2,3)
    return
        <tr>
            <td>{$x/nombre/data()}</td>
            <td>{$ti}</td>
        </tr>
    }

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

Obtenemos como resultado de la consulta:

```
<html>
    <head>
        <title> ejemplo 2</title>
    </head>
    <body>
        <table border="1">
            <tr style="background-color: Coral">
                <td>Nombre de las islas que terminan en
e</td>
```

```
                    <td>Tres ultimas letras de los nombres de
las islas</td>
                </tr>
                <tr>
                    <td>Belle Île</td>
                    <td>Île</td>
                </tr>
                <tr>
                    <td>Isla Bruce</td>
                    <td>uce</td>
                </tr>
                <tr>
                    <td>Isla de Bute</td>
                    <td>ute</td>
                </tr>
                <tr>
                    <td>Isla de São Jorge</td>
                    <td>rge</td>
                </tr>
                <tr>
                    <td>Lanzarote</td>
                    <td>ote</td>
                </tr>
                <tr>
                    <td>Léucade</td>
                    <td>ade</td>
                </tr>
                <tr>
                    <td>Skye</td>
                    <td>kye</td>
                </tr>

                <tr>
                    <td>Tenerife</td>
                    <td>ife</td>
                </tr>
                <tr>
                    <td>Tierra de Jorge</td>
                    <td>rge</td>
                </tr>
                <tr>
                    <td>Tiree</td>
                    <td>ree</td>
                </tr>
                <tr>
                    <td>Tåsinge</td>
```

```
                    <td>nge</td>
                </tr>
            </table>
        </body>
</html>
```

Nombre de las islas que terminan en e	Tres ultimas letras de los nombres de las islas
Belle Île	Île
Isla Bruce	uce
Isla de Bute	ute
Isla de São Jorge	rge
Lanzarote	ote
Léucade	ade
Skye	kye
Tenerife	ife
Tierra de Jorge	rge
Tiree	ree
Tåsinge	nge

For anidados

1. **Visualiza para cada país, en una tabla distinta, su nombre en la primera fila, grupos étnicos y porcentaje en la segunda fila y a partir de la tercera fila, en filas distintas, el nombre de cada grupo étnico junto con su porcentaje.**

```
<html>
	<head>
		<title>for anidados</title>
	</head>
	<body>
{
	for $pais in
	doc("europa.xml")//paises/pais/nombre/string()
	return

	<br>

	<table border="1">
		<tr bgcolor=" PaleGreen">
			<td colspan="2">
				<h2><center>{$pais}</center></h2>
			</td>
		</tr>
		<tr bgcolor=" Lavender">
			<td>
				<h2>grupo étnico</h2>
			</td>
			<td>
				<h3>porcentaje</h3>
			</td>
		</tr>
	{
	for $grupo in
	doc("europa.xml")//paises/pais[nombre=$pais]/grupos_etnico
	s/grupo_etnico
	return

	<tr>
		<td>
			{$grupo/string()}
		</td>
		<td>
			{$grupo/@porcentaje/string()}
```

```
            </td>
      </tr>
      }
       </table>
      </br>

}

   </body>
</html>
```

Se obtiene como resultado:

```
<html>
      <head>
            <title>for anidados</title>
      </head>
      <body>
            <br>
                  <table border="1">
                        <tr bgcolor=" PaleGreen">
                              <td colspan="2">
                                    <h2>
                                          <center>Albania</center>
                                    </h2>
                              </td>
                        </tr>
                        <tr bgcolor=" Lavender">
                              <td>
                                    <h2>grupo étnico</h2>
                              </td>
                              <td>
                                    <h3>porcentaje</h3>
                              </td>
                        </tr>
                        <tr>
                              <td>Albanos</td>
                              <td>92</td>
                        </tr>
                        <tr>
                              <td>Griegos</td>
                              <td>06</td>
                        </tr>
                        <tr>
```

```
                <td>Macedonios</td>
                <td>02</td>
            </tr>
        </table>
</br>
<br>
.
</br>
<br>
.
</br>
<br>
.
</br>
<br>
.
</br>
<br>
        <table border="1">
            <tr bgcolor=" PaleGreen">
                <td colspan="2">
                    <h2>
                        <center>Ucrania</center>
                    </h2>
                </td>
            </tr>
            <tr bgcolor=" Lavender">
                <td>
                    <h2>grupo étnico</h2>
                </td>
                <td>
                    <h3>porcentaje</h3>
                </td>
            </tr>


            <tr>
                <td>Ucranianos</td>
                <td>77.8</td>
            </tr>
            <tr>
                <td>Rusos</td>
                <td>17.3</td>
```

```
                    </tr>
                </table>
            </br>
        </body>
</html>
```

Albania	
grupo étnico	**porcentaje**
Albanos	92
Griegos	06
Macedonios	02

.

.

.

.

.

Ucrania	
grupo étnico	**porcentaje**
Ucranianos	77.8
Rusos	17.3

2. **Visualiza para cada país, en una tabla distinta, su nombre en la primera fila, frontera con y kilómetros en la segunda fila y a partir de la tercera fila, en filas distintas, el código de cada país con el que hace frontera junto con el número de kilómetros que tiene de frontera, al final debe aparecer una fila que indique números totales de frontera terrestre del país x y la suma de todos los kilómetros de frontera de todos los países.**

```
<html>
    <head>
        <title>for anidados</title>
    </head>
    <body>
```

```
{
      for $pais in doc("europa.xml")//paises/pais/nombre/string()
      return

            <br>
            <table border="1">
                  <tr bgcolor="Salmon">
                        <td colspan="2">
                              <h2><center>{$pais}</center></h2>
                        </td>
                  </tr>
                  <tr bgcolor=" Khaki">
                        <td>
                              <h2>frontera con</h2>
                        </td>
                        <td>
                              <h3>Kilómetros</h3>
                        </td>
                  </tr>

      {
      for $frontera in
      doc("europa.xml")//paises/pais[nombre=$pais]/fronteras/fro
      ntera
      return

      <tr>
            <td>
                  {$frontera/@cod_pais/string()}
            </td>
            <td>
                  {$frontera/string()}
            </td>
      </tr>

      }

      <tr bgcolor="Gold" >
            <td>
numero total de kilómetros de frontera terrestre de {$pais}
            </td>
            <td>
            {sum(//paises/pais[nombre=$pais]/fronteras/frontera)}
            </td>
      </tr>
```

```
</table>
</br>

}

      </body>
</html>
```

Se obtiene como resultado:

```
<html>
      <head>
            <title>for anidados</title>
      </head>
      <body>
            <br>
            <table border="1">
                  <tr bgcolor="Salmon">
                        <td colspan="2">
                              <h2>
                                    <center>Albania</center>
                              </h2>
                        </td>
                  </tr>
                  <tr bgcolor=" Khaki">
                        <td>
                              <h2>frontera con</h2>
                        </td>
                        <td>
                              <h3>Kilómetros</h3>
                        </td>
                  </tr>
                  <tr>
                        <td>f0_110021</td>
                        <td>282</td>
                  </tr>
                  <tr>
                        <td>f0_110037</td>
                        <td>172</td>
                  </tr>
                  <tr>
                        <td>f0_110032</td>
                        <td>151</td>
                  </tr>
                  <tr>
```

```
            <td>f0_110046</td>
            <td>115</td>
      </tr>
      <tr bgcolor="Gold">
            <td>
                  número total de kilómetros de
                  frontera terrestre de Albania
            </td>
            <td>720</td>
      </tr>
</table>
</br>
<br>
.
</br>
<br>
.
</br>
<br>
.
</br>
<br>
<table border="1">
      <tr bgcolor="Salmon">
            <td colspan="2">
                  <h2>
                        <center>Ucrania</center>
                  </h2>
            </td>
      </tr>
      <tr bgcolor=" Khaki">
            <td>
                  <h2>frontera con</h2>
            </td>
            <td>
                  <h3>Kilómetros</h3>
            </td>
      </tr>
      <tr>
            <td>f0_110044</td>
            <td>2094</td>
      </tr>
      <tr>
            <td>f0_110035</td>
            <td>0940</td>
      </tr>
```

```
                <tr>
                    <td>f0_110006</td>
                    <td>0891</td>
                </tr>
                <tr>
                    <td>f0_110043</td>
                    <td>0531</td>
                </tr>
                <tr>
                    <td>f0_110040</td>
                    <td>0428</td>
                </tr>
                <tr>
                    <td>f0_110023</td>
                    <td>0136</td>
                </tr>
                <tr>
                    <td>f0_110014</td>
                    <td>0090</td>
                </tr>
                <tr bgcolor="Gold">
                    <td>
                        número total de kilómetros de
                        frontera terrestre de Ucrania
                    </td>
                    <td>5110</td>
                </tr>
            </table>
            </br>
        </body>
</html>
```

Albania	
frontera con	**Kilómetros**
f0_110021	282
f0_110037	172
f0_110032	151
f0_110046	115
numero total de kilómetros de frontera terrestre de Albania	720

.

.

.

Ucrania	
frontera con	**Kilómetros**
f0_110044	2094
f0_110035	0940
f0_110006	0891
f0_110043	0531
f0_110040	0428
f0_110023	0136
f0_110014	0090
numero total de kilómetros de frontera terrestre de Ucrania	5110

Subconsultas

1. **Visualizar para cada montaña, su nombre, altura, y el nombre del país en el que se encuentra.**

```
<html>
	<head>
		<title>subconsultas</title>
	</head>
	<body>
		<table border="1">
			<tr bgcolor="GreenYellow">
```

```
                    <td>Nombre de la Montaña</td>
                    <td>Altura</td>
                    <td>Nombre del País en el que se
encuentra</td>
                </tr>
                {
                    for $mon in(//montanas/montana)
                    let $nom:=$mon/nombre
                    let $alt:=$mon/altura
                    let $c_pais:=$mon/cod_pais
                    let $nom_pais:=
                    (//paises/pais[cod_pais=$c_pais]/nombre)
                    return
                        <tr>
                            <td>{$nom/data()}</td>
                            <td>{$alt/data()}</td>
                            <td>{$nom_pais/data()}</td>
                        </tr>
                }
            </table>
        </body>
</html>
```

2. **Visualiza para cada país, en una tabla distinta, su nombre en la primera fila, frontera con y kilómetros en la segunda fila y a partir de la tercera fila, en filas distintas, el nombre de cada país con el que hace frontera junto con el número de kilómetros que tiene de frontera, al final debe aparecer una fila que indique números totales de frontera terrestre del país x y la suma de todos los kilómetros de frontera de todos los países.**

```
<html>
	<head>
		<title>for anidados</title>
	</head>

	<body>

	{
	for $pais in doc("europa.xml")//paises/pais/nombre/string()
	return

	<br>
	<table border="1">
		<tr bgcolor="Salmon">
			<td colspan="2">
				<h2><center>{$pais}</center></h2>
			</td>
		</tr>
		<tr bgcolor=" Khaki">
			<td>
				<h2>frontera con</h2>
			</td>
			<td>
				<h3>Kilómetros</h3>
			</td>
		</tr>

	{
	for $frontera in
	doc("europa.xml")//paises/pais[nombre=$pais]/fronteras/fro
	ntera
	let $c_pais:=$frontera/@cod_pais
	let $nom_pais:= (//paises/pais[cod_pais=$c_pais]/nombre)
	return

		<tr>
		   <td>{$nom_pais/string()}</td>
```

```
                <td>{$frontera/string()}</td>
            </tr>
        }

            <tr bgcolor="Gold" >
                <td>
numero total de kilómetros de frontera terrestre de {$pais}
                </td>
                <td>
{sum(//paises/pais[nombre=$pais]/fronteras/frontera)}
                </td>
            </tr>
      </table>
      </br>

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

3. **Visualizar para cada pais, su nombre, el numero de ciudades que tiene y la media de la población de las ciudades de cada país.**

```
<html>
      <head>
            <title>subconsultas</title>
      </head>
      <body>
            <table border="1">
                  <tr bgcolor="GreenYellow">
                        <td>País</td>
                        <td>Numero de ciudades</td>
                        <td>Media Poblacional</td>
                  </tr>
                  {
                  for $paises in(//paises/pais)
                  let $cod_p:=$paises/cod_pais
                  let
                  $media:=avg(//ciudades/ciudad[@cod_pais=$cod_p]/
                  poblacion)
                  let
                  $cu:=count(//ciudades/ciudad[@cod_pais=$cod_p])
                  return
                  <tr>
                        <td>{$paises/nombre/string()}</td>
```

```
                    <td>{$cu}</td>
                    <td>{$media}</td>
                </tr>
                }

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

4. **Visualizar para cada ciudad su nombre, población, el nombre del país al que pertenece, así como el nombre de los ríos que pasan por la ciudad.**

```
<html>
    <head>
        <title>subconsultas</title>
    </head>
    <body>
        <table border="1">
            <tr bgcolor="Magenta">
                <td>nombre ciudad</td>
                <td>Pais al que pertenece</td>
                <td>Nombre de los ríos que tiene</td>
            </tr>
            {
            for $ciudades in(//ciudades/ciudad)
            let $nom:=$ciudades/nombre
            let $pob:=$ciudades/poblacion
            let $c_pais:=$ciudades/@cod_pais
            let
            $nom_pais:=(//paises/pais[cod_pais=$c_pais]/nomb
            re)
            let
            $cod_rio:=(//ciudades/ciudad[nombre=$nom]/rios/c
            od_rio)
            let
            $nom_rio:=(//rios/rio[cod_rio=$cod_rio]/nombre)
            return

            <tr>
                <td>{$nom/string()}</td>
                <td>{$nom_pais/string()}</td>
                <td>{$nom_rio/string()}</td>
            </tr>
            }
```

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

Joins

1. **Visualizar el nombre de las ciudades y el nombre del país al que pertenece siempre que la población de la ciudad sea mayor que 2 millones de habitantes ordenados por población.**

```
<html>
      <head>
            <title>joins</title>
      </head>
      <body>
      <table border="1">
            <tr bgcolor="GreenYellow">
                  <td>Nombre Ciudad</td>
                  <td>Población de la ciudad</td>
                  <td>País al que pertenece la ciudad</td>
            </tr>

            {

            for $nom in //ciudades/ciudad
            for $pais in //paises/pais
            where $nom/@cod_pais= $pais/cod_pais and
            $nom/poblacion>2000000
            order by $nom/poblacion/number()
            return

            <tr>
                  <td>{$nom/nombre}</td>
                  <td>{$nom/poblacion}</td>
                  <td>{$pais/nombre}</td>
            </tr>

            }
            </table>
      </body>
</html>
```

2. **Visualizar una tabla para cada país en la que tengamos en la primera fila el nombre del país, en la segunda fila tengamos nombre de la ciudad y nombre del rio, y a partir de la tercera fila los nombres de las ciudades del país y los nombres de los ríos que pasan por dicha ciudad.**

```
<html>
      <head>
            <title>Joins</title>
      </head>
      <body>
      {
      for $nompais in //paises/pais
      return
      <br>
      <table border="1">
            <tr bgcolor="Salmon">
                  <td colspan="2">
                  <h2><center>{$nompais/nombre}</center></h2>
                  </td>
            </tr>
            <tr bgcolor=" Khaki">
                  <td>
                        <h2>nombre ciudad</h2>
                  </td>
                  <td>
                  <h3>nombre de los ríos que pasan por ella</h3>
                  </td>
            </tr>
            {
            for $ciud in //ciudades/ciudad
            for $rio in //rios/rio
            where $nompais/cod_pais= $ciud/@cod_pais and
            $ciud/rios/cod_rio=$rio/cod_rio
            return

            <tr>
                  <td>{$ciud/nombre/data()}</td>
                  <td>{$rio/nombre/data()}</td>
            </tr>
            }

       </table>
      }
      </body></html>
```