

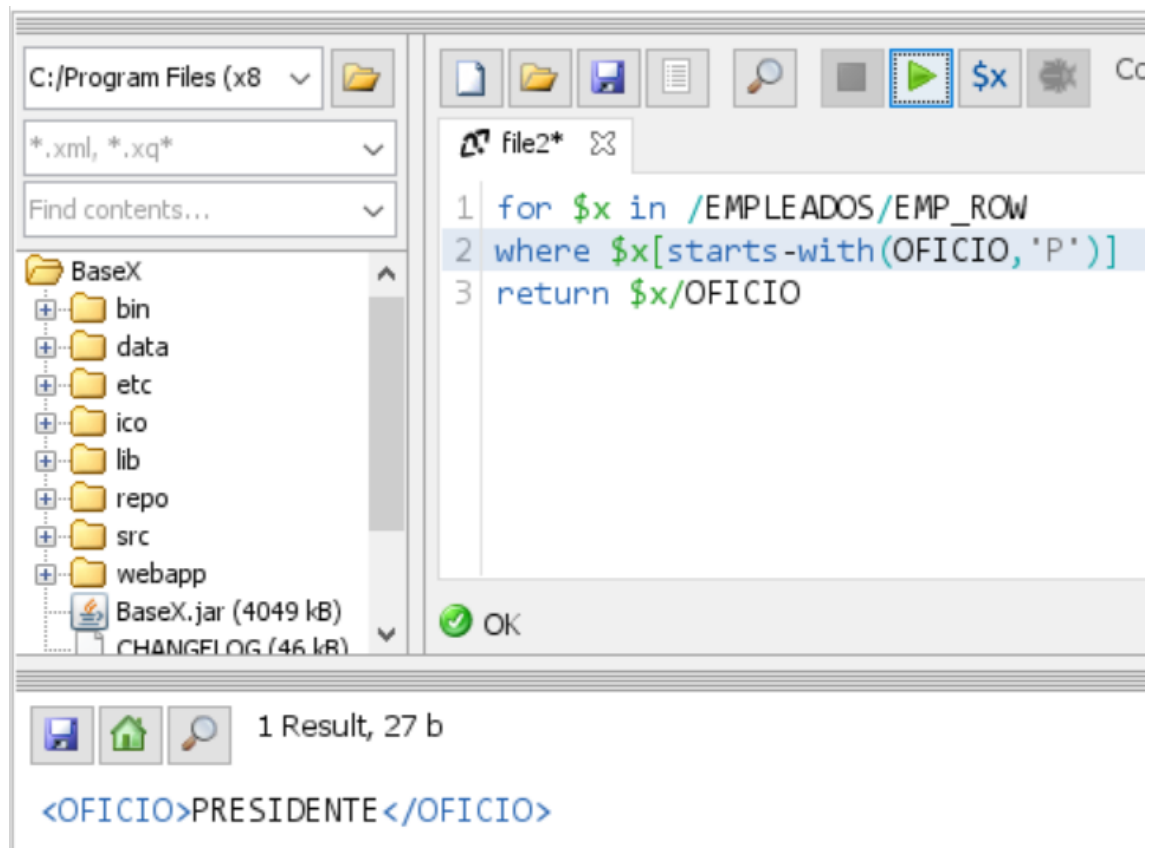
PRÁCTICA . Xquery

Con los fichero **empleados.xml**, **universidad.xml** y **productos.xml**

REALIZA LAS SIGUIENTES CONSULTAS Xquery:

Con el fichero empleados.xml

1. Nombres de oficio que empiezan por P



2. Nombres de oficio y los nº empleados de cada oficio

The screenshot shows the BaseX IDE interface. The left pane displays a file explorer with a tree structure including folders like bin, data, etc, ico, lib, repo, src, and webapp. The main editor area contains an XQuery script:

```

1 for $x in distinct-values(/EMPLEADOS/EMP_ROW/OFICIO)
2 let $y := count(/EMPLEADOS/EMP_ROW[OFICIO=$x])
3 return concat($x, ' ', $y)

```

The context bar at the top right indicates "Context: db:open('PRACTIC...)". Below the editor, the results pane shows "5 Results, 60 b" and a "Result" table with the following data:

EMPLEADO	Count
VENDEDOR	4
DIRECTOR	3
ANALISTA	2
PRESIDENTE	1

On the right side of the results pane, there is a diagram showing a hierarchy of XML elements: "emplea.." (containing "empleados.xml" and "EMPLEADOS"), "productos.xml" (containing "productos"), and a series of vertical bars representing individual data rows.

3. El número de empleados que tiene cada departamento y la media de salario redondeada.

La salida se mostrará:

Departamento: 10. Num emples=3. Media salario=2892

The screenshot shows the BaseX IDE interface. The left pane displays a file explorer with a tree structure including folders like bin, data, etc, ico, lib, repo, src, and webapp. The main editor area contains an XQuery script:

```

1 for $x in distinct-values(EMPLEADOS/EMP_ROW/DEPT_NO)
2 let $y := count(EMPLEADOS/EMP_ROW[DEPT_NO=$x])
3 let $z := round(avg(EMPLEADOS/EMP_ROW[DEPT_NO=$x]/SALARIO))
4 return concat('Departamento= ', $x, ' Empleados= ', $y, ' Media de salarios redondeado= ', $z)

```

The context bar at the top right indicates "Context: db:open('PRACTICA\QUERYV2')". Below the editor, the results pane shows "3 Results, 196 b" and a "Result" table with the following data:

Departamento	Empleados	Media de salarios redondeado
20	5	2274
30	6	1736
10	3	2892

On the right side of the results pane, there is a diagram showing a hierarchy of XML elements: "emplea.." (containing "empleados.xml" and "EMPLEADOS"), "productos.xml" (containing "productos"), and a series of vertical bars representing individual data rows.

4. Modifica la salida anterior para que quede entre etiquetas

```

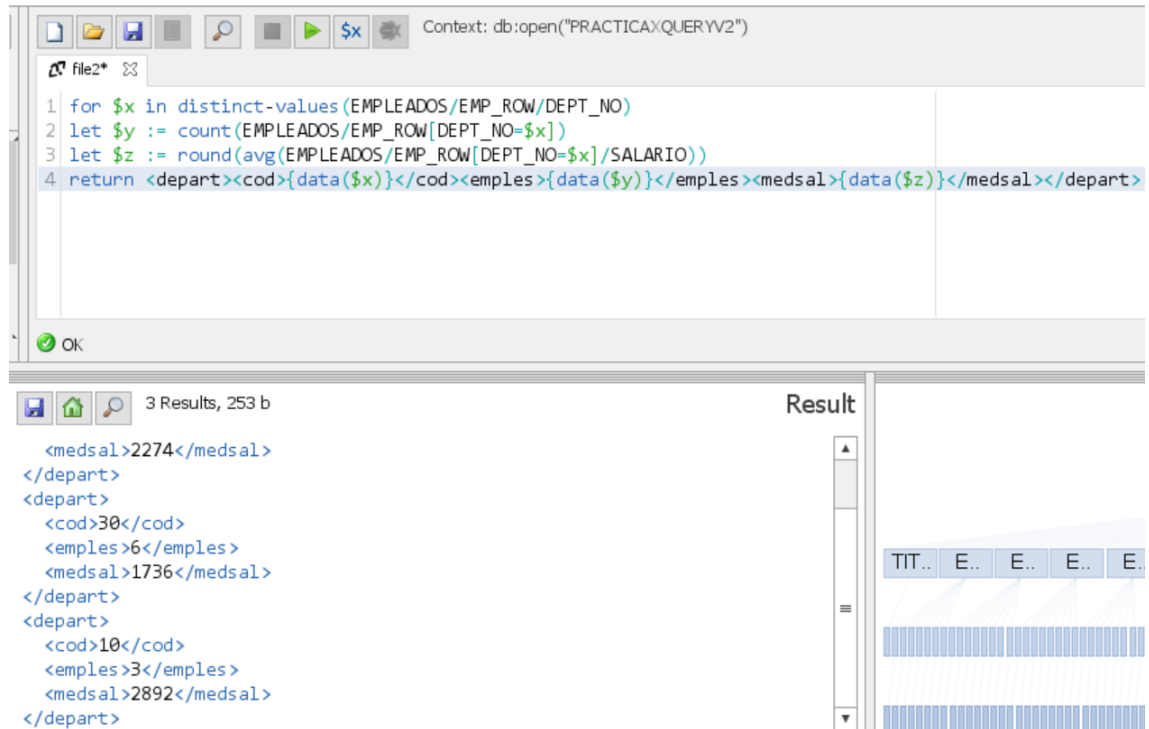
<depart>
  <cod>20</cod>
  <emples>5</emples>
  <medsal>2274</medsal>
</depart>

```

```

for $x in distinct-values(EMPLEADOS/EMP_ROW/DEPT_NO)
let $y := count(EMPLEADOS/EMP_ROW[DEPT_NO=$x])
let $z :=
round(avg(EMPLEADOS/EMP_ROW[DEPT_NO=$x]/SALARIO))
return
<depart><cod>{data($x)}</cod><emples>{data($y)}</emples><med
sal>{data($z)}</medsal></depart>

```



5. Los nombres de los empleados, los que son directores entre etiquetas <DIRECTOR></DIRECTOR> y los que no lo son entre etiquetas <EMPLE></EMPLE>

```

for $x in EMPLEADOS/EMP_ROW
return if ($x/OFICIO = 'DIRECTOR') then
<DIRECTOR>{data($x/APELLIDO)}</DIRECTOR>
else <EMPLE>{data($x/APELLIDO)}</EMPLE>

```

Context: db:open("PRACTICA\XQUER\YV2")

```

1 for $x in EMPLEADOS/EMP_ROW
2 return if ($x/OFICIO = 'DIRECTOR') then <DIRECTOR>{data($x/APELLIDO)}</DIRECTOR>
3 else <EMPLE>{data($x/APELLIDO)}</EMPLE>

```

OK

14 Results, 333 b

Result

```

<EMPLE>SANCHEZ</EMPLE>
<EMPLE>ARROYO</EMPLE>
<EMPLE>SALA</EMPLE>
<DIRECTOR>JIMENEZ</DIRECTOR>
<EMPLE>MARTIN</EMPLE>
<DIRECTOR>NEGRO</DIRECTOR>
<DIRECTOR>CEREZO</DIRECTOR>
<EMPLE>GIL</EMPLE>
<EMPLE>REY</EMPLE>
<EMPLE>TOVAR</EMPLE>
<EMPLE>ALONSO</EMPLE>
<EMPLE>JIMENO</EMPLE>

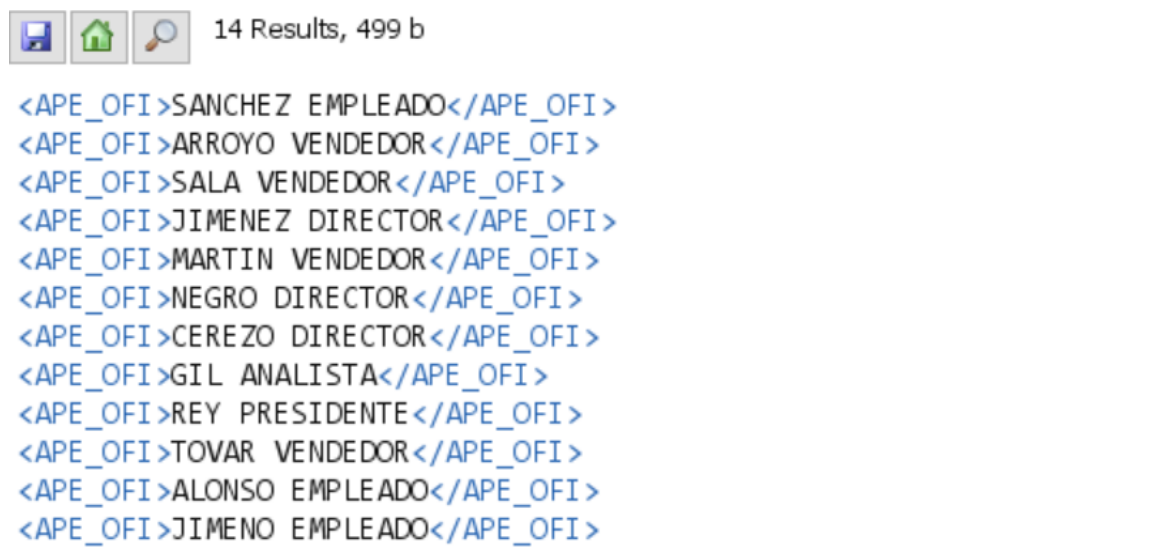
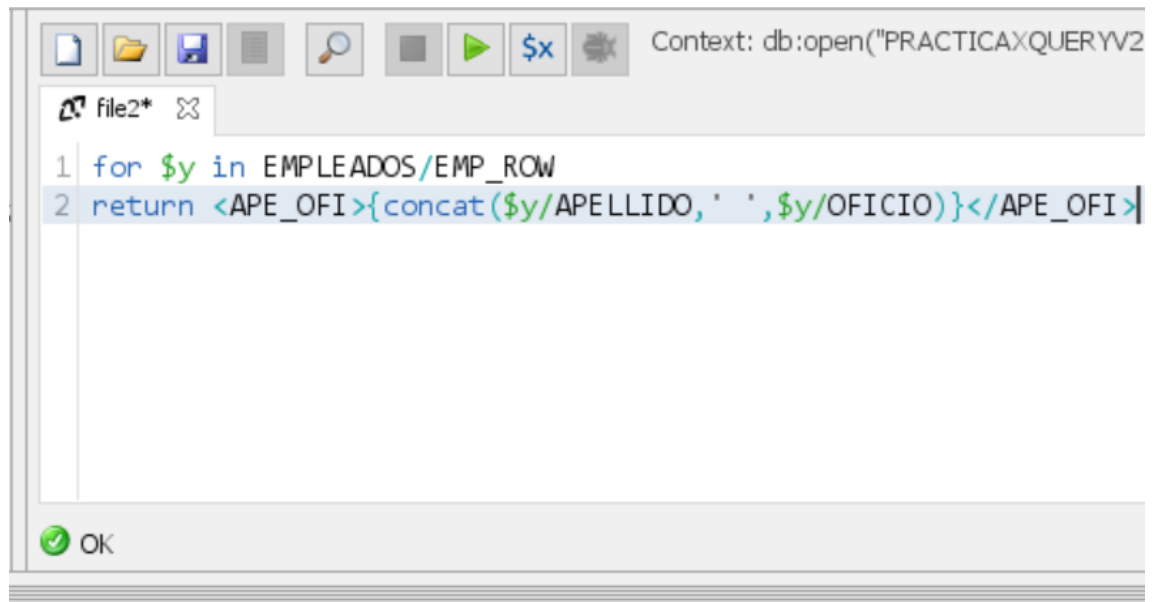
```

6. Nombre y el oficio concatenado entre las etiquetas

<APE_OFI> </APE_OFI>

for \$y in EMPLEADOS/EMP_ROW

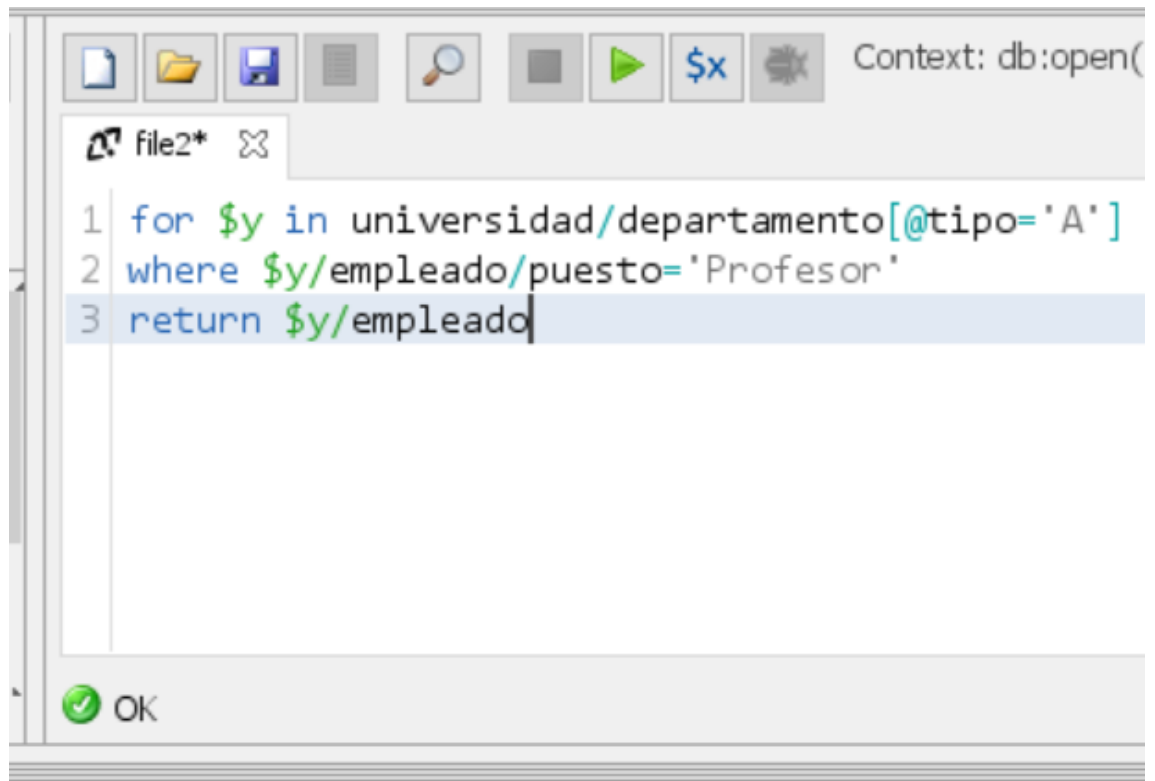
return <APE_OFI>{concat(\$y/APELLIDO,'',\$y/OFICIO)}</APE_OFI>



Con el fichero universidad.xml

7. Los nombres de empleados de los departamentos de tipo A, cuyo puesto es profesor.

```
for $y in universidad/departamento[@tipo='A']
where $y/empleado/puesto='Profesor'
return $y/empleado
```

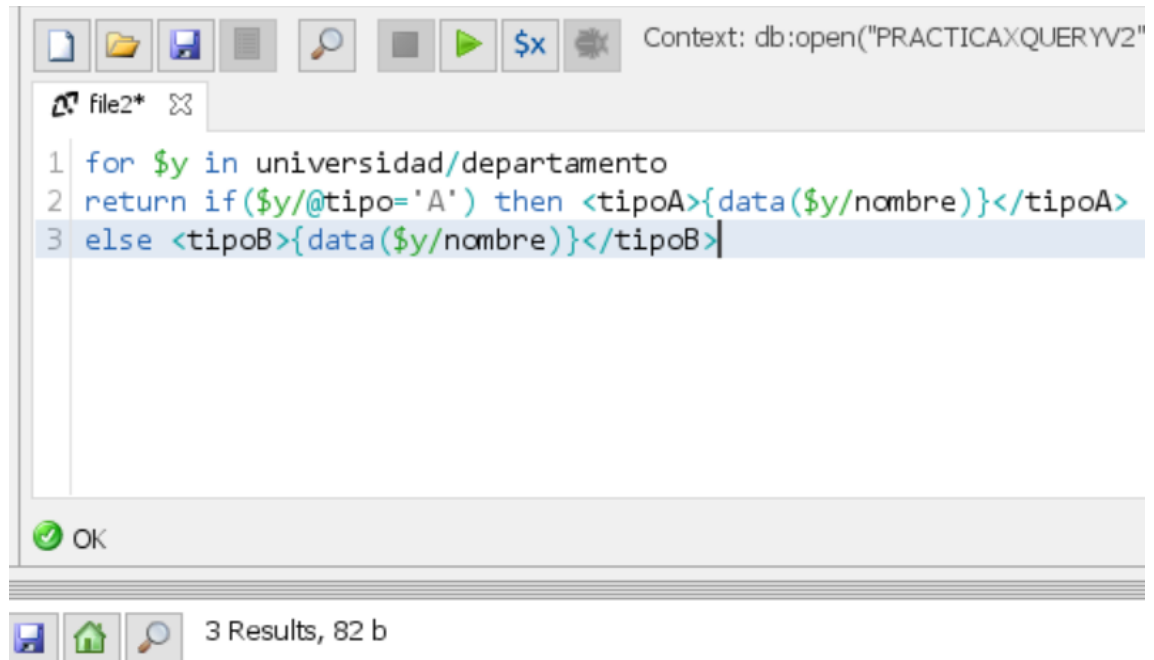


6 Results, 616 b

```
<empleado salario="2000">
  <puesto>Asociado</puesto>
  <nombre>Juan Parra</nombre>
</empleado>
<empleado salario="2300">
  <puesto>Profesor</puesto>
  <nombre>Alicia Martín</nombre>
</empleado>
<empleado salario="1900">
  <puesto>Técnico</puesto>
  <nombre>Juan Parra</nombre>
</empleado>
```

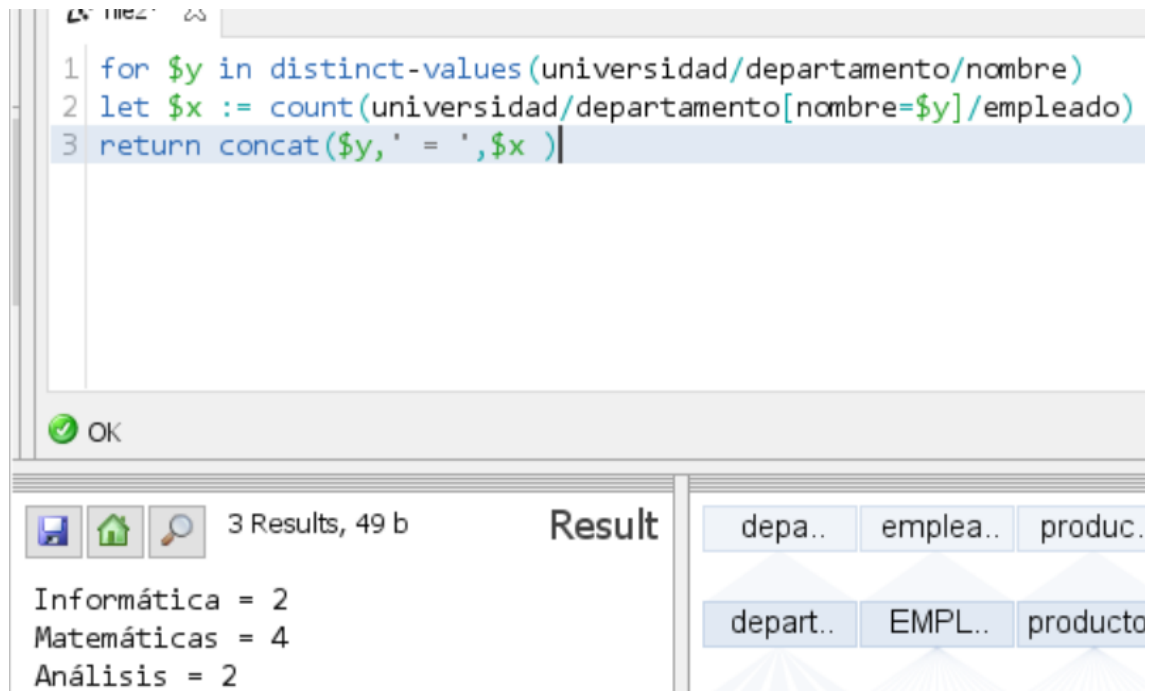
8. Nombre de departamento encerrado entre las etiquetas `<tipo=A></tipo=A>` si es de tipo A, y entre las etiquetas `<tipo=B></tipo=B>` , si no lo es.

```
for $y in universidad/departamento
return if($y/@tipo='A') then <tipoA>{data($y/nombre)}</tipoA>
else <tipoB>{data($y/nombre)}</tipoB>
```



9. Nombres de los departamentos y los empleados que tiene cada departamento.

```
for $y in distinct-values(universidad/departamento/nombre)
let $x := count(universidad/departamento[nombre=$y]/empleado)
return concat($y,' = ', $x )
```



10. Nombres de los departamentos, los empleados que tiene cada departamento y la media del salario entre etiquetas.

```

for $y in universidad/departamento/nombre
let $x := count(universidad/departamento[nombre=$y]/empleado)
let $z :=
avg(universidad/departamento[nombre=$y]/empleado/@salario)
return
<departamento><nombre>{data($y)}</nombre><numemple>{data(
$x)}</numemple><mediasalar>{data($z)}</mediasalar></departam
ento>

```


Context: db:open("PRACTICAXQUERYV2")

```

1 for $y in universidad/departamento/nombre
2 let $x := count(universidad/departamento[nombre=$y]/empleado)
3 let $z := avg(universidad/departamento[nombre=$y]/empleado/@salario)
4 return <departamento><nombre>{data($y)}</nombre><numemple>{data($x)}<
    /numemple><mediasalar>{data($z)}</mediasalar></departamento>

```

OK 4 : 1

3 Results, 370 b

Result

```

<departamento>
  <nombre>Informática</nombre>
  <numemple>2</numemple>
  <mediasalar>2150</mediasalar>
</departamento>
<departamento>
  <nombre>Matemáticas</nombre>
  <numemple>4</numemple>
  <mediasalar>2200</mediasalar>
</departamento>
<departamento>
  <nombre>Análisis</nombre>

```

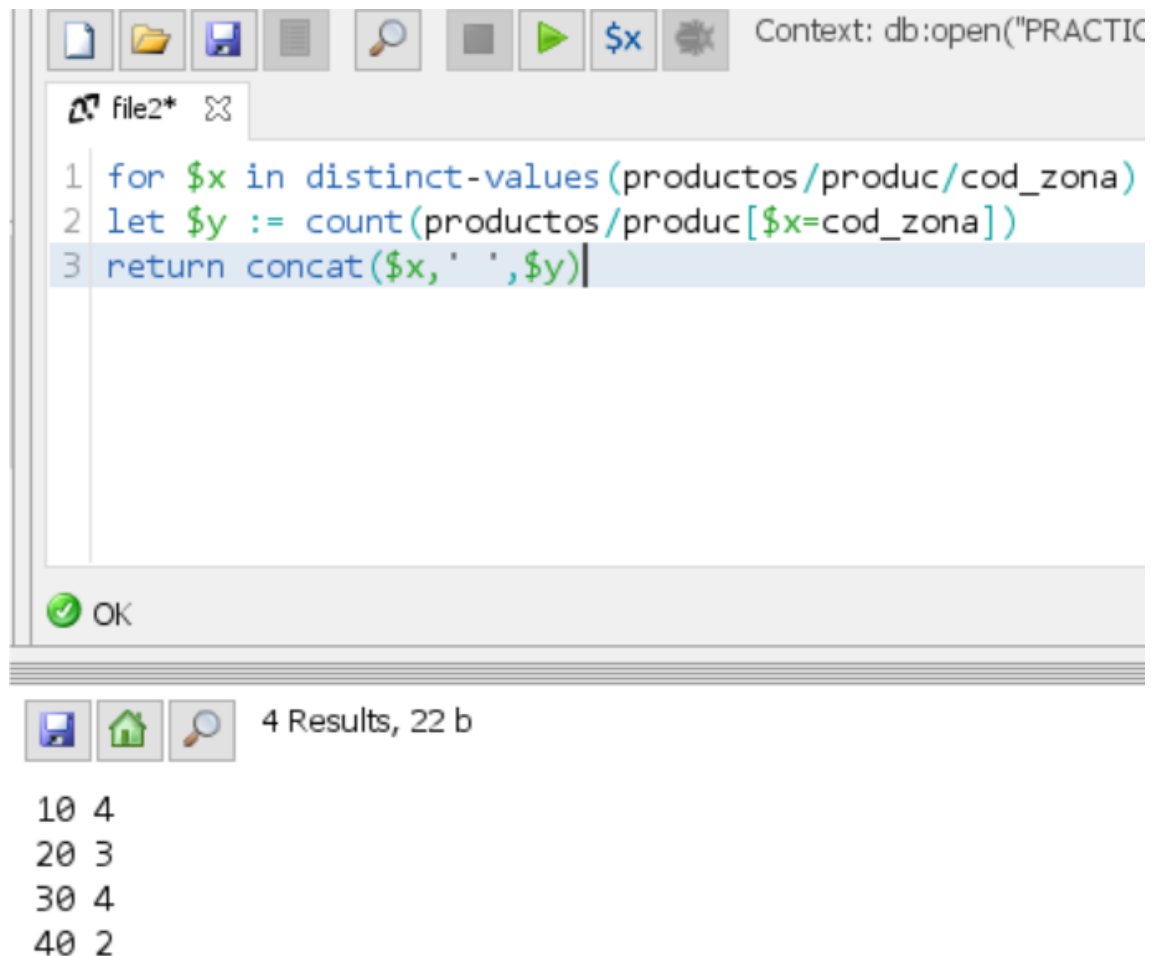
Utilizando el documento **productos.xml**.

11. Obtener por cada zona el número de productos que tiene.

```

for $x in distinct-values(productos/produccion/cod_zona)
let $y := count(productos/produccion[$x=cod_zona])
return concat($x, ' ', $y)

```



The screenshot shows a database query editor window. The query is as follows:

```

1 for $x in distinct-values(productos/produc/cod_zona)
2 let $y := count(productos/produc[$x=cod_zona])
3 return concat($x, ' ', $y)

```

The results of the query are displayed below the editor:

```

10 4
20 3
30 4
40 2

```

12.Obtener la denominación de los productos entres las etiquetas <zona10></zona10> si son del código de zona 10, <zona20></zona20> si son de la zona 20, <zona30></zona30> si son de la 30 y <zona40></zona40> si son de la 40.

```

for $x in productos/produc
return if ($x/cod_zona=10) then
<zona10>{data($x/denominacion)}</zona10>
else if ($x/cod_zona=20) then
<zona20>{data($x/denominacion)}</zona20>
else if ($x/cod_zona=30) then
<zona30>{data($x/denominacion)}</zona30>
else if ($x/cod_zona=40) then
<zona40>{data($x/denominacion)}</zona40>

```

Context: db:open("PRACTICAXQUERY")

```

1 for $x in productos/produc
2 return if ($x/cod_zona=10) then <zona10>{data($x/denominacion)}</
  zona10>
3 else if ($x/cod_zona=20) then <zona20>{data($x/denominacion)}</zona20
  >
4 else if ($x/cod_zona=30) then <zona30>{data($x/denominacion)}</zona30
  >
5 else if ($x/cod_zona=40) then <zona40>{data($x/denominacion)}</zona40
  >

```

OK 2

13 Results, 622 b

Result

```

zona10>
<zona10>Micro Intel Core i5-2320</
zona10>
<zona20>Micro Intel Core i5 2500</
zona20>
<zona20>HD Seagate Barracuda 250GB
SATA</zona20>
<zona30>HD Caviar Blue 500GB SATA3
</zona30>
<zona30>Tarjeta gráfica Asus
GeForce EN210 Silent 1GB</zona30>
<zona40>Tarieta gráfica Gigabvte

```

productos.xml

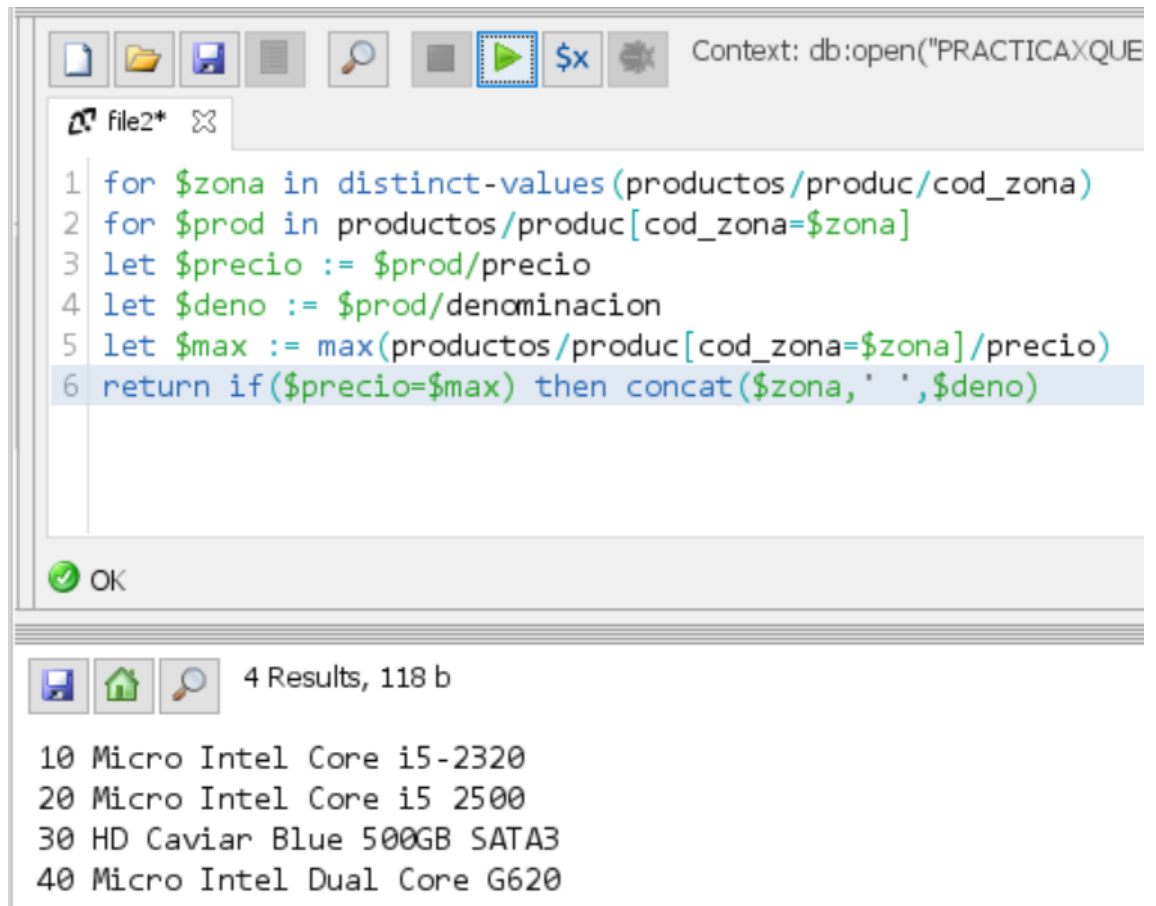
productos

13. Obtener por cada zona la denominación del o de los productos más caros.

```

for $zona in distinct-values(productos/produc/cod_zona)
for $prod in productos/produc[cod_zona=$zona]
let $precio := $prod/precio
let $deno := $prod/denominacion
let $max := max(productos/produc[cod_zona=$zona]/precio)
return if($precio=$max) then concat($zona,' ', $deno)

```

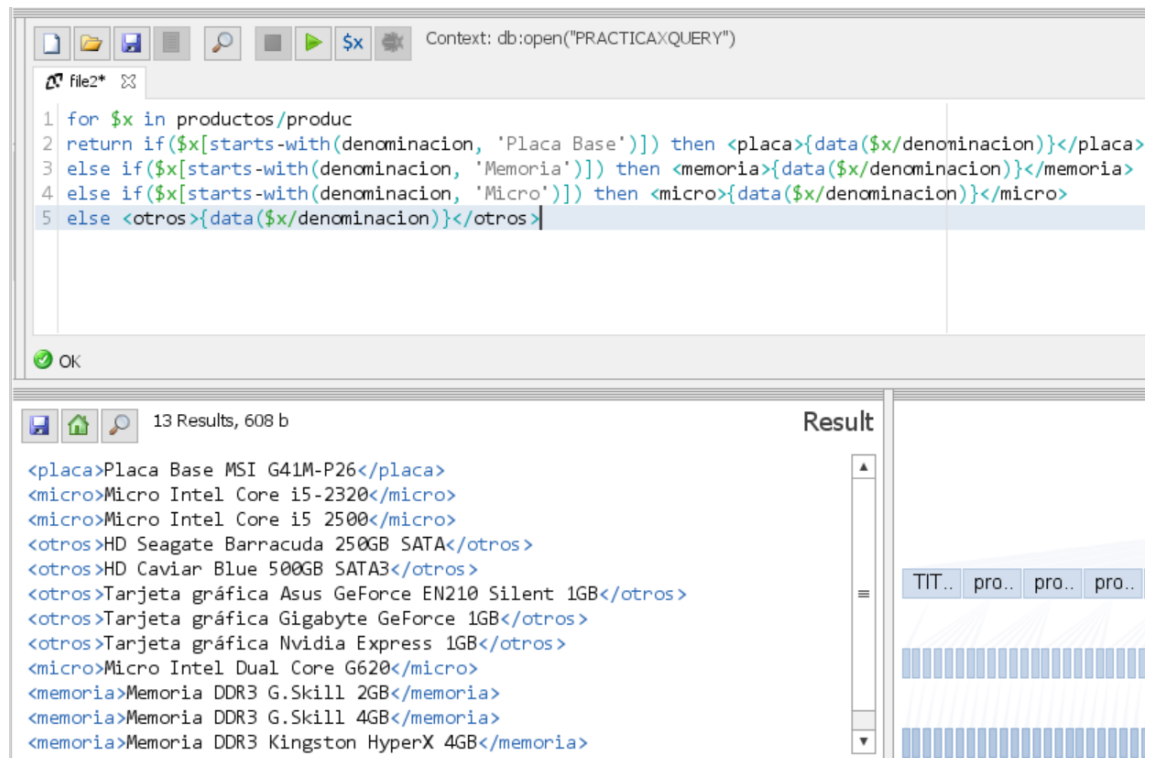


14. Obtener la denominación de los productos contenida entre las etiquetas <placa></placa> para los productos en cuya denominación aparece la palabra Placa Base, <memoria></memoria>, para los que contienen a la palabra Memoria <micro></micro>, para los que contienen la palabra Micro y <otros></otros> para el resto de productos.

```

for $x in productos/produc
return if($x[starts-with(denominacion, 'Placa Base')]) then
  <placa>{data($x/denominacion)}</placa>
else if($x[starts-with(denominacion, 'Memoria')]) then
  <memoria>{data($x/denominacion)}</memoria>
else if($x[starts-with(denominacion, 'Micro')]) then
  <micro>{data($x/denominacion)}</micro>
else <otros>{data($x/denominacion)}</otros>

```



15.A partir del fichero xml, genera mediante una consulta XQuery un HTML con los datos de los productos en forma de tabla y dale formato con un css externo.

HTML Y CSS adjuntos en la entrega.

```

<html>
  <body>
    <h3>TABLA PRODUCTOS</h3>
    <table border="3">
      <tr>
        <th>CODIGO PRODUCTO</th>
        <th>DENOMINACION</th>
        <th>PRECIO</th>
        <th>STOCK ACTUAL</th>
        <th>STOCK MINIMO</th>
        <th>CODIGO ZONA</th>
      </tr>{
      for $x in productos/produc
      return
        <tr>
          <td>{$x/cod_prod}</td>
          <td>{$x/denominacion}</td>
          <td>{$x/precio}</td>
          <td>{$x/stock_actual}</td>
          <td>{$x/stock_minimo}</td>
          <td>{$x/cod_zona}</td>
        </tr>
      }</table>
    </body>
  </html>

```

TABLA PRODUCTOS

CODIGO PRODUCTO	DENOMINACION	PRECIO	STOCK ACTUAL	STOCK MINIMO	CODIGO ZONA
1010	Placa Base MSI G41M-P26	50	10	3	10
1011	Micro Intel Core i5-2320	120	3	5	10
1012	Micro Intel Core i5 2500	170	5	6	20
1013	HD Seagate Barracuda 250GB SATA	80	10	5	20
1014	HD Caviar Blue 500GB SATA3	150	5	6	30
1015	Tarjeta gráfica Asus GeForce EN210 Silent 1GB	40	10	5	30
1016	Tarjeta gráfica Gigabyte GeForce 1GB	50	5	6	40
1017	Tarjeta gráfica Nvidia Express 1GB	45	10	5	30
1018	Micro Intel Dual Core G620	60	15	5	40
1019	Memoria DDR3 G.Skill 2GB	10	5	3	10
1020	Memoria DDR3 G.Skill 4GB	30	30	10	10
1021	Memoria DDR3 Kingston HyperX 4GB	16	15	4	20
1022	Placa Base ASRock G41M-S3	52	2	2	30