

Xpath

9. Resuelve las siguientes consultas:

a) Devuelve el apellido del penúltimo empleado (NOTA: utilizar last())

```
new-document 1*
1 xquery version "3.1";
2 doc('nueva/empleados.xml')/EMPLEADOS/EMP_ROW[position()=last()-1]/APELLIDO/text()

__new__1
1 FERNANDEZ
```

b) Obtén los elementos del empleado que ocupa la posición 3 (position())

```
new-document 1* empleados.xml
1 xquery version "3.1";
2 doc('nueva/empleados.xml')/EMPLEADOS/EMP_ROW[position()=3]

__new__1
1 <EMP_ROW>
  <EMP_NO>7521</EMP_NO>
  <APELLIDO>SALA</APELLIDO>
  <OFICIO>VENDEDOR</OFICIO>
  <DIR>7698</DIR>
  <FECHA_ALT>1991-02-22</FECHA_ALT>
  <SALARIO>1625</SALARIO>
  <COMISION>650</COMISION>
  <DEPT_NO>30</DEPT_NO>
</EMP_ROW>
```

c) Cuenta el número de empleados del departamento 10

```
new-document 1* empleados.xml
1 xquery version "3.1";
2 count(doc('nueva/empleados.xml')/EMPLEADOS/EMP_ROW[DEPT_NO=10])

__new__1
1 3
```

d) Obtén la suma de SALARIO de los empleados del DEPT_NO=20

```
new-document 1*  empleados.xml
1  xquery version "3.1";
2  sum(doc('nueva/empleados.xml')/EMPLEADOS/EMP_ROW[DEPT_NO=20]/SALARIO)

__new__1
1  1.137e4
```

e) Obtén el salario máximo, el mínimo de los empleados con OFICIO=ANALISTA

```
New  New XQuery  Open  Save  Close  Eval  Run

new-document 1*  empleados.xml*
1  xquery version "3.1";
2  concat(min(doc('nueva/empleados.xml')/EMPLEADOS/EMP_ROW[OFICIO="ANALISTA"]/SALARIO)
3  , "-"
4  , max(doc('nueva/empleados.xml')/EMPLEADOS/EMP_ROW[OFICIO="ANALISTA"]/SALARIO))

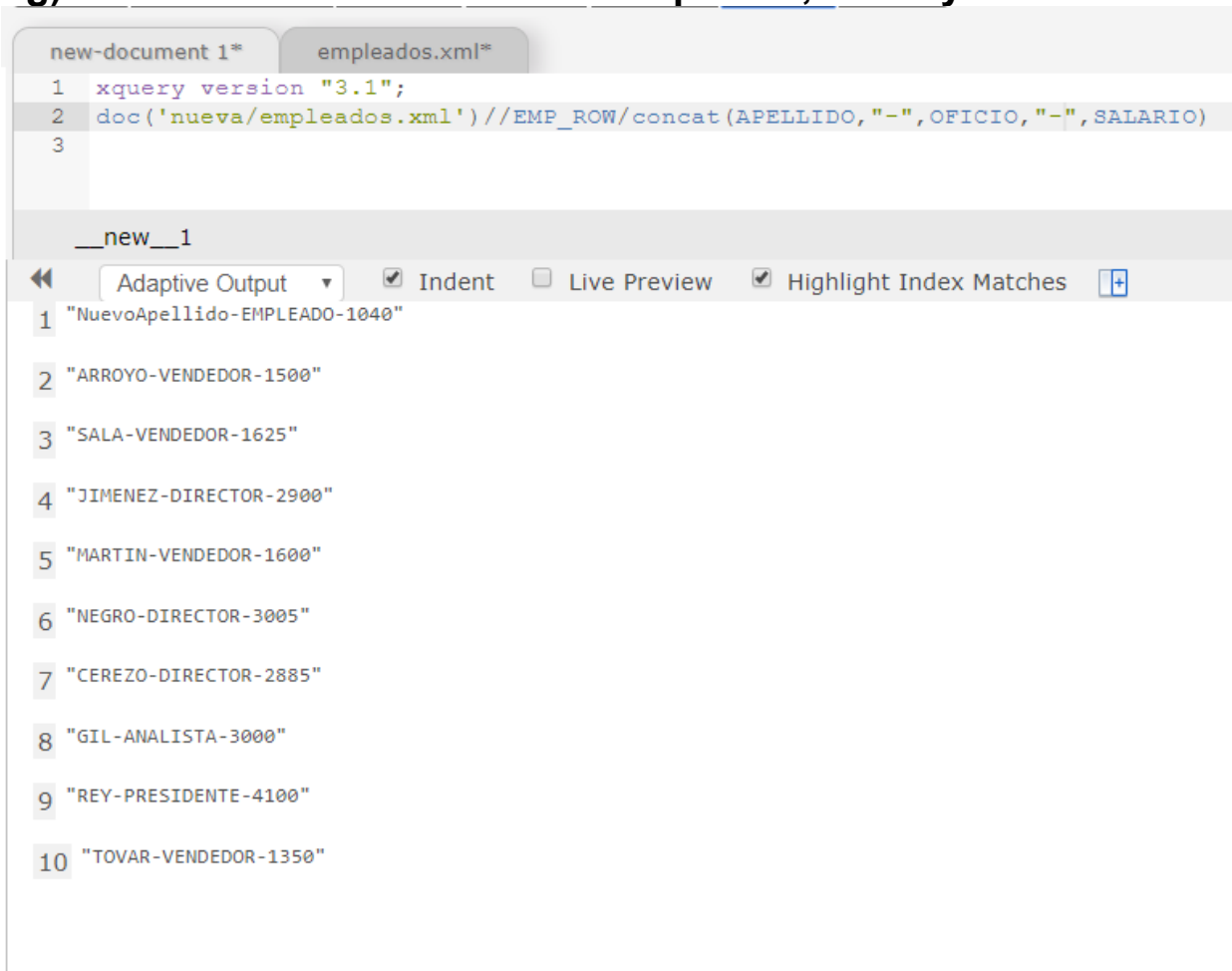
__new__1
1  "3000-3000"
```

f) Obtén la media de salario en el DEPT_NO=10

```
new-document 1*  empleados.xml*
1  xquery version "3.1";
2  concat(avg(doc('nueva/empleados.xml')/EMPLEADOS/EMP_ROW[DEPT_NO=10]/SALARIO), "€")
3

__new__1
1  "2891.6666666666665€"
```

g) Devuelve la concatenación de apellido, oficio y salario



The screenshot shows an XQuery editor with two tabs: "new-document 1*" and "empleados.xml*". The query in the editor is:

```
1 xquery version "3.1";
2 doc('nueva/empleados.xml')//EMP_ROW/concat(APELLIDO,"-",OFICIO,"-",SALARIO)
3
```

Below the editor, the results are displayed in a table with the title "__new__1". The table has a single column containing the concatenated strings of last name, job title, and salary for each employee.

1 "NuevoApellido-EMPLEADO-1040"
2 "ARROYO-VENDEDOR-1500"
3 "SALA-VENDEDOR-1625"
4 "JIMENEZ-DIRECTOR-2900"
5 "MARTIN-VENDEDOR-1600"
6 "NEGRO-DIRECTOR-3005"
7 "CEREZO-DIRECTOR-2885"
8 "GIL-ANALISTA-3000"
9 "REY-PRESIDENTE-4100"
10 "TOVAR-VENDEDOR-1350"

h) Obtén los elementos de los empleados cuyo apellido empieza por 'A'

```
new-document 1*  empleados.xml*
1  xquery version "3.1";
2  doc('nueva/empleados.xml')//EMP_ROW[starts-with(*,"A")]
3

__new__1
Adaptive Output  Indent  Live Preview  Highlight Index Matches  +
1  <EMP_ROW>
    <EMP_NO>7499</EMP_NO>
    <APELLIDO>ARROYO</APELLIDO>
    <OFICIO>VENDEDOR</OFICIO>
    <DIR>7698</DIR>
    <FECHA_ALT>1990-02-20</FECHA_ALT>
    <SALARIO>1500</SALARIO>
    <COMISION>390</COMISION>
    <DEPT_NO>30</DEPT_NO>
  </EMP_ROW>
2  <EMP_ROW>
    <EMP_NO>7788</EMP_NO>
    <APELLIDO>GIL</APELLIDO>
    <OFICIO>ANALISTA</OFICIO>
    <DIR>7566</DIR>
    <FECHA_ALT>1991-11-09</FECHA_ALT>
    <SALARIO>3000</SALARIO>
    <DEPT_NO>20</DEPT_NO>
  </EMP_ROW>
3  <EMP_ROW>
    <EMP_NO>7876</EMP_NO>
    <APELLIDO>ALONSO</APELLIDO>
    <OFICIO>EMPLEADO</OFICIO>
    <DIR>7788</DIR>
    <FECHA_ALT>1991-09-23</FECHA_ALT>
    <SALARIO>1430</SALARIO>
    <DEPT_NO>20</DEPT_NO>
  </EMP_ROW>
4  <EMP_ROW>
    <EMP_NO>7902</EMP_NO>
    <APELLIDO>FERNANDEZ</APELLIDO>
    <OFICIO>ANALISTA</OFICIO>
    <DIR>7566</DIR>
    <FECHA_ALT>1991-12-03</FECHA_ALT>
    <SALARIO>3000</SALARIO>
    <DEPT_NO>20</DEPT_NO>
  </EMP_ROW>
```

i) Devuelve los oficios que contienen la sílaba 'OR'

```
new-document 1*  empleados.xml*
1  xquery version "3.1";
2  doc('nueva/empleados.xml')//EMP_ROW[contains(OFICIO,"OR")]/OFICIO/text()
3

__new__1
Adaptive Output  Indent  Live Preview  Highlight Index Matches  +
1  VENDEDOR
2  VENDEDOR
3  DIRECTOR
4  VENDEDOR
5  DIRECTOR
6  DIRECTOR
7  VENDEDOR
```

j) Obtén los datos de los empleados cuyo apellido tiene menos de 4 caracteres

```
new-document 1*  empleados.xml*
1  xquery version "3.1";
2  doc('nueva/empleados.xml')//EMP_ROW[string-length(APELLIDO)<4]
3

__new__1
Adaptive Output  Indent  Live Preview  Highlight Index Matches  +
1  <EMP_ROW>
    <EMP_NO>7788</EMP_NO>
    <APELLIDO>GIL</APELLIDO>
    <OFICIO>ANALISTA</OFICIO>
    <DIR>7566</DIR>
    <FECHA_ALT>1991-11-09</FECHA_ALT>
    <SALARIO>3000</SALARIO>
    <DEPT_NO>20</DEPT_NO>
  </EMP_ROW>
2  <EMP_ROW>
    <EMP_NO>7839</EMP_NO>
    <APELLIDO>REY</APELLIDO>
    <OFICIO>PRESIDENTE</OFICIO>
    <FECHA_ALT>1991-11-17</FECHA_ALT>
    <SALARIO>4100</SALARIO>
    <DEPT_NO>10</DEPT_NO>
  </EMP_ROW>
```

10. Resuelve las siguientes consultas referentes al documento productos.xml. Este documento contiene los datos de los productos de una distribuidora de componentes informáticos. La estructura del documento es:

a) Obtén la denominación y precio de todos los productos

```
new-document 1*  empleados.xml*  productos.xml
1  xquery version "3.1";
2  doc('nueva/productos.xml')//produc/concat(denominacion," -> ",precio,"€")
3

__new__1
Adaptive Output  Indent  Live Preview  Highlight Index Matches  +
1  "Placa Base MSI G41M-P26 -> 50€"
2  "Micro Intel Core i5-2320 -> 120€"
3  "Micro Intel Core i5 2500 -> 170€"
4  "HD Seagate Barracuda 250GB SATA -> 80€"
5  "HD Caviar Blue 500GB SATA3 -> 150€"
6  "Tarjeta gráfica Asus GeForce EN210 Silent 1GB -> 40€"
7  "Tarjeta gráfica Gigabyte GeForce 1GB -> 50€"
8  "Tarjeta gráfica Nvidia Express 1GB -> 45€"
9  "Micro Intel Dual Core G620 -> 60€"
10 "Memoria DDR3 G.Skill 2GB -> 10€"
```

b) Obtén los productos que sean “Placa base”

```
new-document 1*  empleados.xml*  productos.xml
1  xquery version "3.1";
2  doc('nueva/productos.xml')//produc[contains(denominacion, "Placa Base")]
3

__new__1
Adaptive Output  Indent  Live Preview  Highlight Index Matches  +
1  <produc>
    <cod_prod>1010</cod_prod>
    <denominacion>Placa Base MSI G41M-P26</denominacion>
    <precio>50</precio>
    <stock_actual>10</stock_actual>
    <stock_minimo>3</stock_minimo>
    <cod_zona>10</cod_zona>
  </produc>
2  <produc>
    <cod_prod>1022</cod_prod>
    <denominacion>Placa Base ASRock G41M-S3 </denominacion>
    <precio>52</precio>
    <stock_actual>2</stock_actual>
    <stock_minimo>2</stock_minimo>
    <cod_zona>30</cod_zona>
  </produc>
```

c) Obtén los productos cuyo precio sea mayor que 60€ y de la zona 20

```
new-document 1*  empleados.xml*  productos.xml
1  xquery version "3.1";
2  doc('nueva/productos.xml')//produc[precio>60 and cod_zona=20]
3

__new__1
Adaptive Output  Indent  Live Preview  Highlight Index Matches  +
1 <produc>
  <cod_prod>1012</cod_prod>
  <denominacion>Micro Intel Core i5 2500</denominacion>
  <precio>170</precio>
  <stock_actual>5</stock_actual>
  <stock_minimo>6</stock_minimo>
  <cod_zona>20</cod_zona>
</produc>
2 <produc>
  <cod_prod>1013</cod_prod>
  <denominacion>HD Seagate Barracuda 250GB SATA</denominacion>
  <precio>80</precio>
  <stock_actual>10</stock_actual>
  <stock_minimo>5</stock_minimo>
  <cod_zona>20</cod_zona>
</produc>
```

d) Obtén el número de los productos que sean memorias y de la zona 10

```
new-document 1*  empleados.xml*  productos.xml
1  xquery version "3.1";
2  doc('nueva/productos.xml')//produc[contains(denominacion, "Memoria") and cod_zona=10]
3

__new__1
Adaptive Output  Indent  Live Preview  Highlight Index Matches  +
1 <produc>
  <cod_prod>1019</cod_prod>
  <denominacion>Memoria DDR3 G.Skill 2GB</denominacion>
  <precio>10</precio>
  <stock_actual>5</stock_actual>
  <stock_minimo>3</stock_minimo>
  <cod_zona>10</cod_zona>
</produc>
2 <produc>
  <cod_prod>1020</cod_prod>
  <denominacion>Memoria DDR3 G.Skill 4GB</denominacion>
  <precio>30</precio>
  <stock_actual>30</stock_actual>
  <stock_minimo>10</stock_minimo>
  <cod_zona>10</cod_zona>
</produc>
```

e) Obtén la media de los precios de los micros

```
new-document 1*  empleados.xml*  productos.xml
1  xquery version "3.1";
2  concat(avg(doc('nueva/productos.xml')//produc[contains(denominacion, "Micro")]/precio),"€")
3

__new__1
1  "116.66666666666667€"
```

f) Obtén los datos de los productos cuyo stock mínimo sea mayor que el stock actual (NOTA: usa función number())

```
new-document 1*  empleados.xml*  productos.xml
1  xquery version "3.1";
2  doc('nueva/productos.xml')//produc[number(stock_minimo) > number(stock_actual)]
3

__new__1
1  <produc>
   <cod_prod>1011</cod_prod>
   <denominacion>Micro Intel Core i5-2320</denominacion>
   <precio>120</precio>
   <stock_actual>3</stock_actual>
   <stock_minimo>5</stock_minimo>
   <cod_zona>10</cod_zona>
</produc>
2  <produc>
   <cod_prod>1012</cod_prod>
   <denominacion>Micro Intel Core i5 2500</denominacion>
   <precio>170</precio>
   <stock_actual>5</stock_actual>
   <stock_minimo>6</stock_minimo>
   <cod_zona>20</cod_zona>
</produc>
3  <produc>
   <cod_prod>1014</cod_prod>
   <denominacion>HD Caviar Blue 500GB SATA3</denominacion>
   <precio>150</precio>
   <stock_actual>5</stock_actual>
   <stock_minimo>6</stock_minimo>
   <cod_zona>30</cod_zona>
</produc>
4  <produc>
   <cod_prod>1016</cod_prod>
   <denominacion>Tarjeta gráfica Gigabyte GeForce 1GB</denominacion>
   <precio>50</precio>
   <stock_actual>5</stock_actual>
   <stock_minimo>6</stock_minimo>
   <cod_zona>40</cod_zona>
</produc>
```


g) Obtén el producto más caro

The screenshot shows an XML editor with three tabs: 'new-document 1*', 'empleados.xml*', and 'productos.xml'. The 'productos.xml' tab is active, displaying an XQuery on line 2: `doc('nueva/productos.xml')//produc[precio = max(precio)]`. Line 1 contains `xquery version "3.1";` and line 3 is empty. Below the editor, the output window titled '__new__1' shows the result of the query: an XML element `<produc>` with attributes `<cod_prod>1012</cod_prod>`, `<denominacion>Micro Intel Core i5 2500</denominacion>`, `<precio>170</precio>`, `<stock_actual>5</stock_actual>`, `<stock_minimo>6</stock_minimo>`, and `<cod_zona>20</cod_zona>`. The output window also features a toolbar with 'Adaptive Output', 'Indent', 'Live Preview', and 'Highlight Index Matches' options.

```
1 xquery version "3.1";
2 doc('nueva/productos.xml')//produc[precio = max(precio)]
3
```

```
1 <produc>
  <cod_prod>1012</cod_prod>
  <denominacion>Micro Intel Core i5 2500</denominacion>
  <precio>170</precio>
  <stock_actual>5</stock_actual>
  <stock_minimo>6</stock_minimo>
  <cod_zona>20</cod_zona>
</produc>
```

h) Obtén el producto más barato de la zona 20

The screenshot shows the same XML editor interface. The 'productos.xml' tab is active, displaying an XQuery on line 2: `doc('nueva/productos.xml')//produc[cod_zona=20][precio=min(precio)]`. Line 1 contains `xquery version "3.1";` and line 3 is empty. The output window titled '__new__1' shows the result: an XML element `<produc>` with attributes `<cod_prod>1021</cod_prod>`, `<denominacion>Memoria DDR3 Kingston HyperX 4GB</denominacion>`, `<precio>16</precio>`, `<stock_actual>15</stock_actual>`, `<stock_minimo>4</stock_minimo>`, and `<cod_zona>20</cod_zona>`. The output window also features a toolbar with 'Adaptive Output', 'Indent', 'Live Preview', and 'Highlight Index Matches' options.

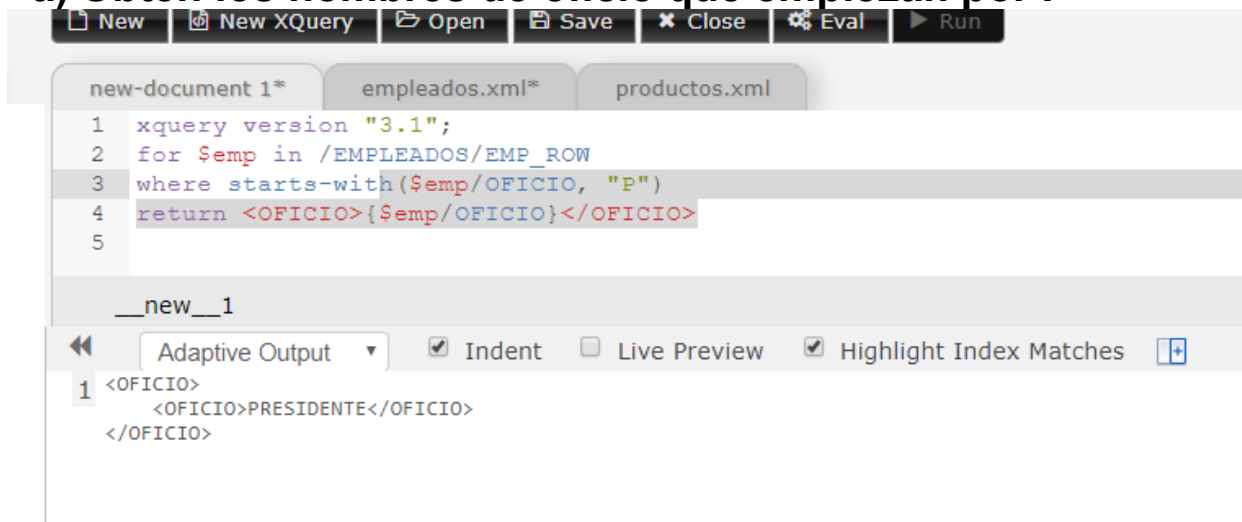
```
1 xquery version "3.1";
2 doc('nueva/productos.xml')//produc[cod_zona=20][precio=min(precio)]
3
```

```
1 <produc>
  <cod_prod>1021</cod_prod>
  <denominacion>Memoria DDR3 Kingston HyperX 4GB</denominacion>
  <precio>16</precio>
  <stock_actual>15</stock_actual>
  <stock_minimo>4</stock_minimo>
  <cod_zona>20</cod_zona>
</produc>
```

Problemas XQuery_2

1. Resuelve las siguientes consultas utilizando el documento EMPLEADOS.xml

a) Obtén los nombres de oficio que empiezan por P



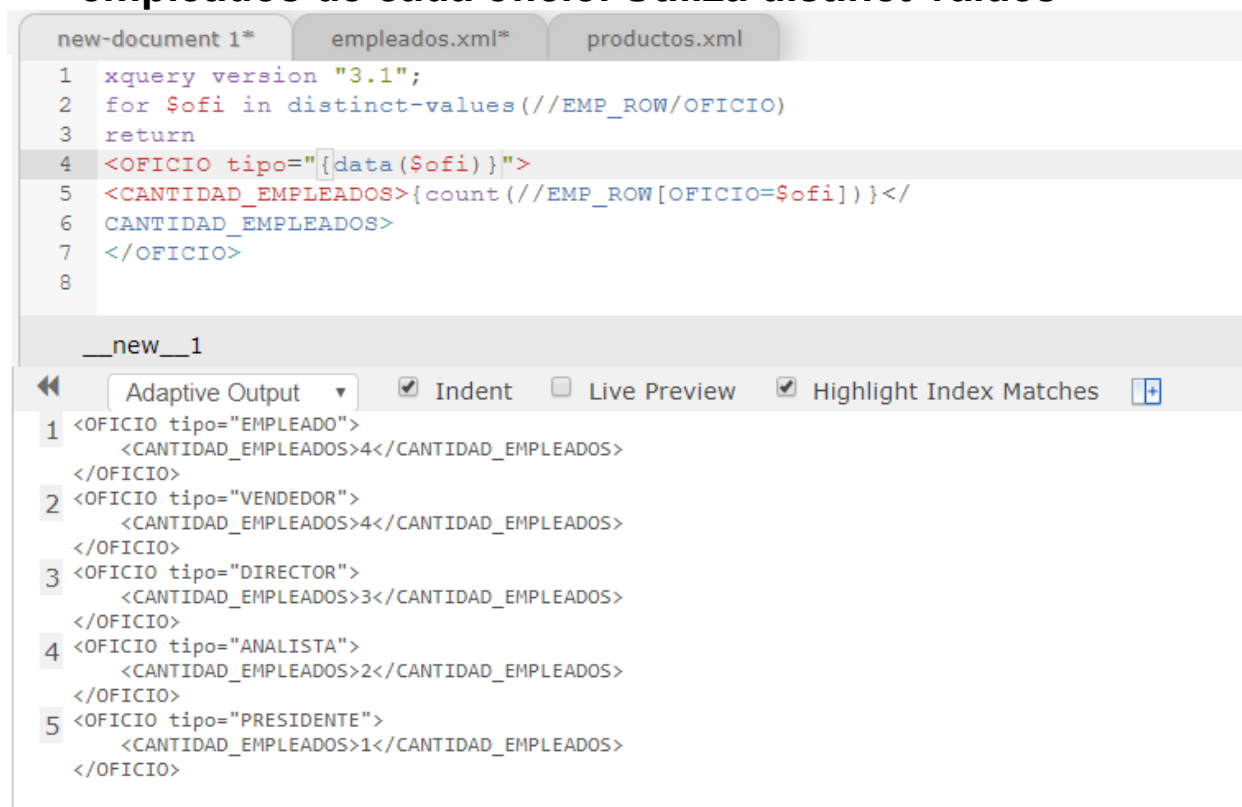
The screenshot shows an XQuery editor with a menu bar (New, New XQuery, Open, Save, Close, Eval, Run) and a tab bar with 'new-document 1*', 'empleados.xml*', and 'productos.xml'. The query editor contains the following XQuery:

```
1 xquery version "3.1";
2 for $emp in /EMPLEADOS/EMP_ROW
3 where starts-with($emp/OFICIO, "P")
4 return <OFICIO>{$emp/OFICIO}</OFICIO>
```

The output pane shows the result:

```
1 <OFICIO>
  <OFICIO>PRESIDENTE</OFICIO>
</OFICIO>
```

b) Obtén los nombres de oficio y el número de los empleados de cada oficio. Utiliza distinct-values



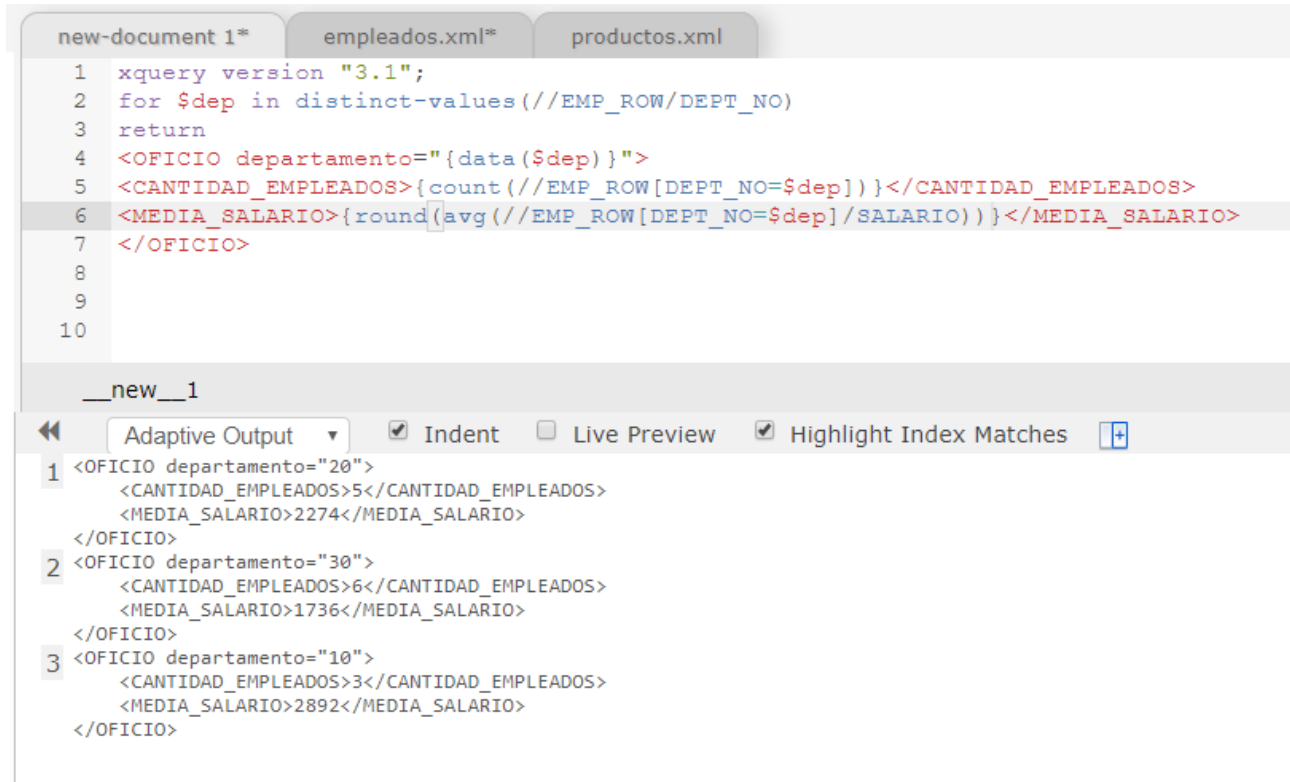
The screenshot shows an XQuery editor with a menu bar (New, New XQuery, Open, Save, Close, Eval, Run) and a tab bar with 'new-document 1*', 'empleados.xml*', and 'productos.xml'. The query editor contains the following XQuery:

```
1 xquery version "3.1";
2 for $ofi in distinct-values(//EMP_ROW/OFICIO)
3 return
4 <OFICIO tipo="{data($ofi)}">
5 <CANTIDAD_EMPLEADOS>{count(//EMP_ROW[OFICIO=$ofi])}</
6 CANTIDAD_EMPLEADOS>
7 </OFICIO>
```

The output pane shows the result:

```
1 <OFICIO tipo="EMPLEADO">
  <CANTIDAD_EMPLEADOS>4</CANTIDAD_EMPLEADOS>
</OFICIO>
2 <OFICIO tipo="VENDEDOR">
  <CANTIDAD_EMPLEADOS>4</CANTIDAD_EMPLEADOS>
</OFICIO>
3 <OFICIO tipo="DIRECTOR">
  <CANTIDAD_EMPLEADOS>3</CANTIDAD_EMPLEADOS>
</OFICIO>
4 <OFICIO tipo="ANALISTA">
  <CANTIDAD_EMPLEADOS>2</CANTIDAD_EMPLEADOS>
</OFICIO>
5 <OFICIO tipo="PRESIDENTE">
  <CANTIDAD_EMPLEADOS>1</CANTIDAD_EMPLEADOS>
</OFICIO>
```

c) Obtén el número de empleados que tiene cada departamento y la media de salario redondeada



The screenshot shows an XML editor with three tabs: 'new-document 1*', 'empleados.xml*', and 'productos.xml'. The 'new-document 1*' tab is active, displaying an XQuery. The query iterates over distinct department numbers and returns XML elements for each, including the count of employees and the rounded average salary. Below the editor, the output is shown in a panel titled '__new__1', displaying three XML snippets corresponding to departments 20, 30, and 10.

```
1 xquery version "3.1";
2 for $dep in distinct-values(//EMP_ROW/DEPT_NO)
3 return
4 <OFICIO departamento="{data($dep)}">
5 <CANTIDAD_EMPLEADOS>{count(//EMP_ROW[DEPT_NO=$dep])}</CANTIDAD_EMPLEADOS>
6 <MEDIA_SALARIO>{round(avg(//EMP_ROW[DEPT_NO=$dep]/SALARIO))}</MEDIA_SALARIO>
7 </OFICIO>
8
9
10
```

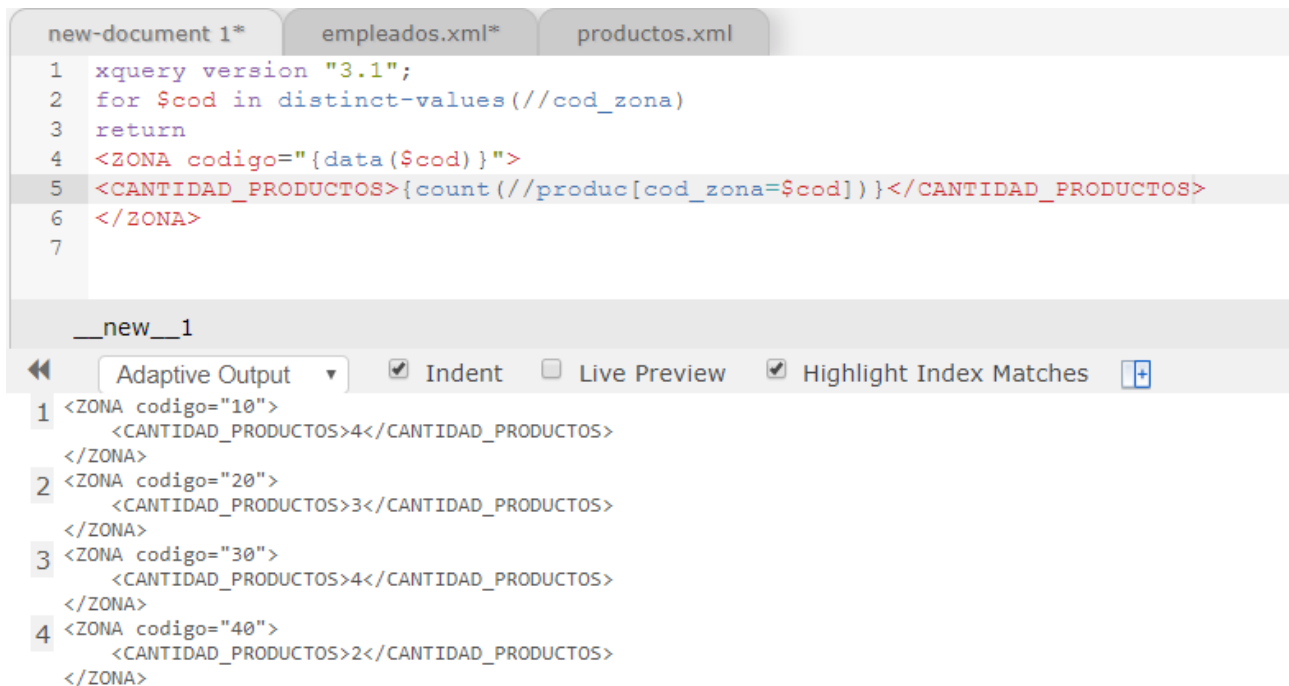
__new__1

Adaptive Output ☒ Indent ☐ Live Preview ☒ Highlight Index Matches [+](#)

```
1 <OFICIO departamento="20">
  <CANTIDAD_EMPLEADOS>5</CANTIDAD_EMPLEADOS>
  <MEDIA_SALARIO>2274</MEDIA_SALARIO>
</OFICIO>
2 <OFICIO departamento="30">
  <CANTIDAD_EMPLEADOS>6</CANTIDAD_EMPLEADOS>
  <MEDIA_SALARIO>1736</MEDIA_SALARIO>
</OFICIO>
3 <OFICIO departamento="10">
  <CANTIDAD_EMPLEADOS>3</CANTIDAD_EMPLEADOS>
  <MEDIA_SALARIO>2892</MEDIA_SALARIO>
</OFICIO>
```

2. Utilizando el documento productos.xml, resuelve con Xquery:

a) Obtén por cada zona el número de productos que tiene



The screenshot shows an XQuery editor with three tabs: 'new-document 1*', 'empleados.xml*', and 'productos.xml'. The query is written in the first tab and executed, showing the result in the second tab.

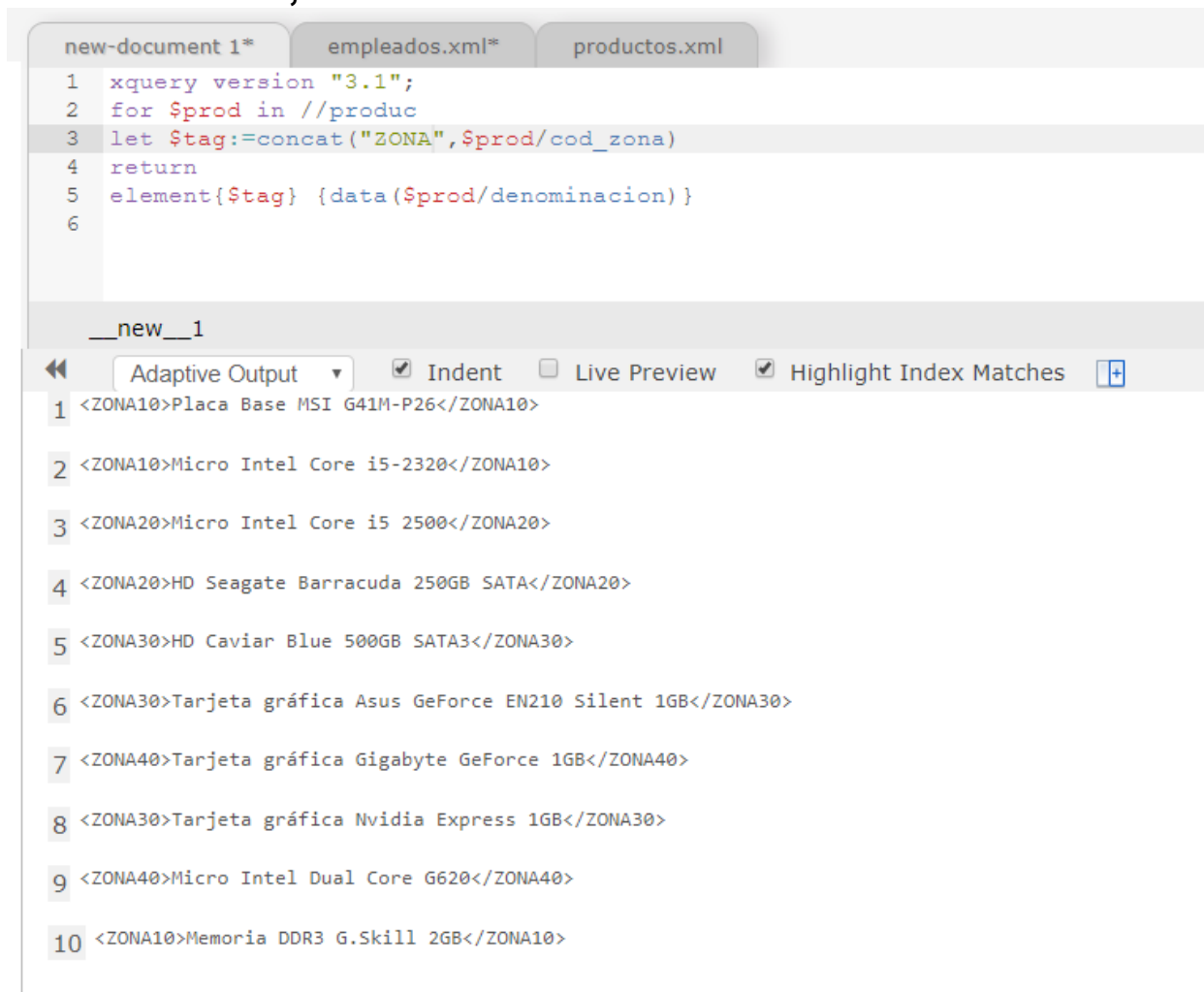
```
1 xquery version "3.1";
2 for $cod in distinct-values(//cod_zona)
3 return
4 <ZONA codigo="{data($cod)}">
5 <CANTIDAD_PRODUCTOS>{count(//produc[cod_zona=$cod])}</CANTIDAD_PRODUCTOS>
6 </ZONA>
7
```

The result, titled '__new__1', is displayed in the second tab. It shows the XML output of the query, with the following structure:

```
1 <ZONA codigo="10">
2   <CANTIDAD_PRODUCTOS>4</CANTIDAD_PRODUCTOS>
3 </ZONA>
4 <ZONA codigo="20">
5   <CANTIDAD_PRODUCTOS>3</CANTIDAD_PRODUCTOS>
6 </ZONA>
7 <ZONA codigo="30">
8   <CANTIDAD_PRODUCTOS>4</CANTIDAD_PRODUCTOS>
9 </ZONA>
10 <ZONA codigo="40">
11   <CANTIDAD_PRODUCTOS>2</CANTIDAD_PRODUCTOS>
12 </ZONA>
```

The editor interface includes a toolbar with options: 'Adaptive Output' (dropdown), 'Indent' (checked), 'Live Preview' (unchecked), and 'Highlight Index Matches' (checked). A plus icon is also visible on the right.

b) Obtén la denominación de los productos entre las etiquetas si son del código de zona 10, si son del código de zona 20, etc.



The screenshot shows an XML editor with three tabs: 'new-document 1*', 'empleados.xml*', and 'productos.xml'. The 'new-document 1*' tab is active, displaying an XQuery script. Below the script, the output is shown in a preview window titled '__new__1'. The output consists of ten XML elements, each representing a product with its zone code and name. The elements are numbered 1 through 10 on the left side of the preview window.

```
1 xquery version "3.1";
2 for $prod in //produc
3 let $tag:=concat("ZONA", $prod/cod_zona)
4 return
5 element{$tag} {data($prod/denominacion)}
6
```

__new__1

Adaptive Output Indent Live Preview Highlight Index Matches

- 1 <ZONA10>Placa Base MSI G41M-P26</ZONA10>
- 2 <ZONA10>Micro Intel Core i5-2320</ZONA10>
- 3 <ZONA20>Micro Intel Core i5 2500</ZONA20>
- 4 <ZONA20>HD Seagate Barracuda 250GB SATA</ZONA20>
- 5 <ZONA30>HD Caviar Blue 500GB SATA3</ZONA30>
- 6 <ZONA30>Tarjeta gráfica Asus GeForce EN210 Silent 1GB</ZONA30>
- 7 <ZONA40>Tarjeta gráfica Gigabyte GeForce 1GB</ZONA40>
- 8 <ZONA30>Tarjeta gráfica Nvidia Express 1GB</ZONA30>
- 9 <ZONA40>Micro Intel Dual Core G620</ZONA40>
- 10 <ZONA10>Memoria DDR3 G.Skill 2GB</ZONA10>

c) Obtén por cada zona la denominación del o de los productos más caros.

```
new-document 1*  empleados.xml*  productos.xml
1  xquery version "3.1";
2  for $prod in distinct-values(//cod_zona)
3  let $precioMax:=max(//produc[cod_zona=$prod]/precio)
4  let $denom:=//produc[precio=$precioMax]/denominacion
5  return
6  <DENOMINACION precio="{data($precioMax)}">{$denom/text()}</DENOMINACION>

__new__1
Adaptive Output  Indent  Live Preview  Highlight Index Matches
1 <DENOMINACION precio="120">Micro Intel Core i5-2320</DENOMINACION>
2 <DENOMINACION precio="170">Micro Intel Core i5 2500</DENOMINACION>
3 <DENOMINACION precio="150">HD Caviar Blue 500GB SATA3</DENOMINACION>
4 <DENOMINACION precio="60">Micro Intel Dual Core G620</DENOMINACION>
```

d) Obtén la denominación de los productos contenida entre las etiquetas para los productos en cuya denominación aparece la palabra Placa Base, , para los que contienen la palabra Memoria , para los que contienen la palabra Micro y para el resto de productos

```
new-document 1*  empleados.xml*  productos.xml
1  xquery version "3.1";
2  for $prod in //produc
3  return
4  if(contains($prod/denominacion, "Placa"))
5  then <placa>{data($prod/denominacion)}</placa>
6  else if(contains($prod/denominacion, "Memoria"))
7  then <memoria>{data($prod/denominacion)}</memoria>
8  else if(contains($prod/denominacion, "Micro"))
9  then <micro>{data($prod/denominacion)}</micro>
10 else <otro>{data($prod/denominacion)}</otro>
```

__new__1

⏪ Adaptive Output ☒ Indent ☐ Live Preview ☒ Highlight Index Matches [+](#)

```
1 <placa>Placa Base MSI G41M-P26</placa>
2 <micro>Micro Intel Core i5-2320</micro>
3 <micro>Micro Intel Core i5 2500</micro>
4 <otro>HD Seagate Barracuda 250GB SATA</otro>
5 <otro>HD Caviar Blue 500GB SATA3</otro>
6 <otro>Tarjeta gráfica Asus GeForce EN210 Silent 1GB</otro>
7 <otro>Tarjeta gráfica Gigabyte GeForce 1GB</otro>
8 <otro>Tarjeta gráfica Nvidia Express 1GB</otro>
9 <micro>Micro Intel Dual Core G620</micro>
10 <memoria>Memoria DDR3 G.Skill 2GB</memoria>
```