

Xpath

9. Resuelve las siguientes consultas:

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

```
//EMP_ROW[position()=last()-1]/APELLIDO/text()
```

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

```
//EMP_ROW[position()=3]
```

c) Cuenta el número de empleados del departamento 10

```
count(//EMP_ROW[DEPT_NO=10])
```

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

```
sum(//EMP_ROW[DEPT_NO=20]/SALARIO)
```

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

```
//EMP_ROW/concat(APELLIDO,"-",OFICIO,"-",SALARIO)
```

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

```
avg(//EMP_ROW[DEPT_NO=10]/SALARIO)
```

g) Devuelve la concatenación de apellido, oficio y salario

```
concat(//EMP_ROW/APELLIDO,//EMP_ROW/OFICIO,//EMP_ROW/SALARIO)
```

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

```
//EMP_ROW[starts-with(*,"A")]
```

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

```
//EMP_ROW[contains(OFICIO,"OR")]/OFICIO/text()
```

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

```
//EMP_ROW[string-length(APELLIDO)<4]
```

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

`//produc/concat(denominacion," ",precio)`

b) Obtén los productos que sean “Placa base”

`//produc[contains(denominacion, "Placa Base")]`

c) Obtén los productos cuyo precio sea mayor que 60€ y de la zona 20 Obtén el número de los productos que sean memorias y de la zona 10

`//produc[precio>60 and cod_zona=20]`

d) Obtén la media de los precios de los micros

`avg(//produc[contains(denominacion, "Micro")]/precio)`

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

`//produc[number(stock_minimo) > number(stock_actual)]`

f) Obtén el producto más caro

`//produc[precio = max(precio)]`

g) Obtén el producto más barato de la zona 20

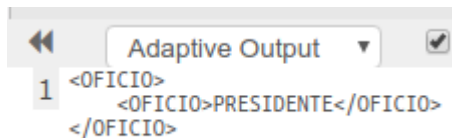
`//produc[cod_zona=20][precio=min(precio)]`

Problemas XQuery_2

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

a) Obtén los nombres de oficio que empiezan por P

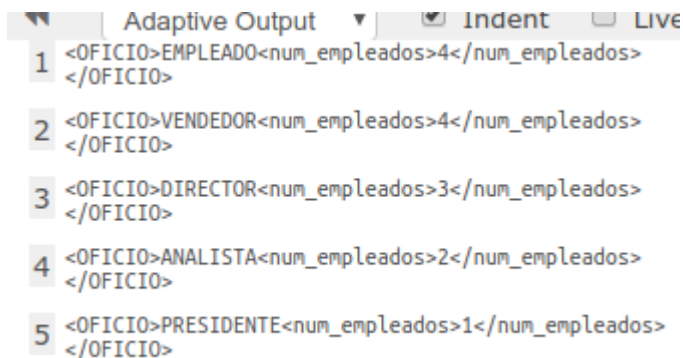
```
for $emp in /EMPLEADOS/EMP_ROW  
where starts-with($emp/OFICIO, "P")  
return <OFICIO>{$emp/OFICIO}</OFICIO>
```



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

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

```
for $ofi in distinct-values(//EMP_ROW/OFICIO)  
return  
  <OFICIO>{data($ofi)}  
    <num_empleados>{count(//EMP_ROW[OFICIO=$ofi])}</  
    num_empleados>  
  </OFICIO>
```



```
1 <OFICIO>EMPLEADO<num_empleados>4</num_empleados>  
  </OFICIO>  
2 <OFICIO>VENDEDOR<num_empleados>4</num_empleados>  
  </OFICIO>  
3 <OFICIO>DIRECTOR<num_empleados>3</num_empleados>  
  </OFICIO>  
4 <OFICIO>ANALISTA<num_empleados>2</num_empleados>  
  </OFICIO>  
5 <OFICIO>PRESIDENTE<num_empleados>1</num_empleados>  
  </OFICIO>
```

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

```
for $dep in distinct-values(//EMP_ROW/DEPT_NO)  
return  
  <OFICIO>{data($dep)}  
    <num_empleados>{count(//EMP_ROW[DEPT_NO=$dep])}</  
    num_empleados>
```

```
<media_salario>{round(avg(//EMP_ROW[DEPT_NO=$dep]/S
ALARIO))}</media_salario>
</OFICIO>
```

```
Adaptive Output  Indent
1 <OFICIO>20<num_empleados>5</num_empleados>
  <media_salario>2274</media_salario>
  </OFICIO>
2 <OFICIO>30<num_empleados>6</num_empleados>
  <media_salario>1736</media_salario>
  </OFICIO>
3 <OFICIO>10<num_empleados>3</num_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**
 for \$cod in distinct-values(//cod_zona)
 return

```
  <zona>{data($cod)}
  <num_productos>{count(//produc[cod_zona=$cod])}</
num_productos>
  </zona>
```

```
Adaptive Output  Indent
1 <zona>10<num_productos>4</num_productos>
  </zona>
2 <zona>20<num_productos>3</num_productos>
  </zona>
3 <zona>30<num_productos>4</num_productos>
  </zona>
4 <zona>40<num_productos>2</num_productos>
  </zona>
```

- 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.**

```
for $prod in //produc
let $tag:=concat("zona",$prod/cod_zona)
return
  element{$tag} {data($prod/denominacion)}
```

```

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.

```

for $prod in distinct-values(//cod_zona)
let $precioMax:=max(//produc[cod_zona=$prod]/precio)
let $denom:=//produc[precio=$precioMax]/denominacion
return
  <denominacion>{$denom/text()}</denominacion>

```

```

1 <denominacion>Micro Intel Core i5-2320</denominacion>
2 <denominacion>Micro Intel Core i5 2500</denominacion>
3 <denominacion>HD Caviar Blue 500GB SATA3</denominacion>
4 <denominacion>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

```

for $prod in //produc
return
  if(contains($prod/denominacion, "Placa"))
    then <placa>{data($prod)}</placa>
  else if(contains($prod/denominacion, "Memoria"))
    then <memoria>{data($prod)}</memoria>
  else if(contains($prod/denominacion, "Micro"))
    then <micro>{data($prod)}</micro>
  else <otro>{data($prod)}</otro>

```

```

1 <placa>
  1010
  Placa Base MSI G41M-P26
  50
  10
  3
  10
</placa>
2 <micro>
  1011
  Micro Intel Core i5-2320
  120
  3
  5
  10
</micro>
3 <micro>
  1012
  Micro Intel Core i5 2500
  170
  5
  6
  20
</micro>
4 <otro>
  1013
  HD Seagate Barracuda 250GB SATA
  80
  10
  5
  20
</otro>
5 <otro>
  1014
  HD Caviar Blue 500GB SATA3
  150
  5
  6
  --
5 <otro>
  1014
  HD Caviar Blue 500GB SATA3
  150
  5
  6
  30
</otro>
6 <otro>
  1015
  Tarjeta gráfica Asus GeForce EN210 Silent 1GB
  40
  10
  5
  30
</otro>
7 <otro>
  1016
  Tarjeta gráfica Gigabyte GeForce 1GB
  50
  5
  6
  40
</otro>
8 <otro>
  1017
  Tarjeta gráfica Nvidia Express 1GB
  45
  10
  5
  30
</otro>
9 <micro>
  1018
  Micro Intel Dual Core G620
  60
  15
  5
  40
</micro>
10 <memoria>
  1019
  Memoria DDR3 G.Skill 2GB
  10
  5
  3
  10
</memoria>

```

