

Problemas XQuery_1

Prueba las siguientes expresiones en eXide y averigua qué devuelven:

```
for $emp in /EMPLEADOS/EMP_ROW
order by $emp/APELLIDO
return if ($emp/OFICIO='DIRECTOR')
then <DIRECTOR>{$emp/APELLIDO/text()}</DIRECTOR>
else <EMPLE> {data($emp/APELLIDO)} </EMPLE>
```

- **devuelve con la etiqueta DIRECTOR a los empleados con oficio DIRECTOR al resto los devuelve con la etiqueta EMPL**

```
for $prof in /universidad/departamento[@tipo="A"]/empleado
let $profe:=$prof/nombre, $puesto:=$prof/puesto
where $puesto="Profesor"
return $profe
```

- **Devuelve los profesores del departamento A**

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

- **Devuelve los departamentos entre etiquetas si es un departamento de Tipo A lo pone entre las etiquetas tipoA si no lo pone entre las etiquetas tipoB**

```
for $dep in /universidad/departamento
let $nom:=$dep/empleado
return <depart>{data($dep/nombre)}
<emple>{count($nom)} </emple>
</depart>
```

- **Devuelve el departamento y el numero de empleados de cada uno**

```
for $dep in /universidad/departamento
let $emp:=$dep/empleado
let $sal:=$dep/empleado/@salario
return
  <depart>{data($dep/nombre)}
```

```
<emple>{count($emp)}</emple>
<medsal>{avg($sal)}</medsal>
</depart>
```

- **Devuelve lo mismo que el anterior pero calculando el sueldo medio**

```
for $dep in /universidad/departamento
let $emp:=$dep/empleado
let $sal:=$dep/empleado/@salario
let $maxi:=max($dep/empleado/@salario)
let $emplmax:=$dep/empleado[@salario=$maxi]
return
  <depart>{data($dep/nombre)}
  <emple>{count($emp)}</emple>
  <medsal>{avg($sal)}</medsal>
  <salmax>{$maxi}</salmax>
  <emplemax>{$emplmax/nombre/text()} -
  {data($emplmax/@salario)}</emplemax>
</depart>
```

- **Devuelve lo mismo que antes añadiendo el salario maximo de cada departamento y quien lo cobra**

Problemas XQuery_2

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

- Obtén los nombres de oficio que empiezan por P

```
xquery version "3.1";
for $emp in /EMPLEADOS/EMP_ROW
order by $emp/APELLIDO
return $emp[starts-with(OFICIO, "P")]/OFICIO
```

```
<OFICIO>PRÉSIDENTE</OFICIO>
```

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

```
for $emp in distinct-values(/EMPLEADOS/EMP_ROW/OFICIO)
let $count:=count(/EMPLEADOS/EMP_ROW[OFICIO=$emp])
return
```

```
  <depart>{$emp}
  <emple>{$count}</emple>
</depart>
```

```
1 <depart>EMPLEADO<emple>4</emple>
  </depart>
2 <depart>VENDEDOR<emple>4</emple>
  </depart>
3 <depart>DIRECTOR<emple>3</emple>
  </depart>
4 <depart>ANALISTA<emple>2</emple>
  </depart>
5 <depart>PRESIDENTE<emple>1</emple>
  </depart>
```

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

```
for $emp in distinct-values(/EMPLEADOS/EMP_ROW/OFICIO)
let $count:=count(/EMPLEADOS/EMP_ROW[OFICIO=$emp])
let $medsal:=avg(/EMPLEADOS/EMP_ROW[OFICIO=$emp]/SALARIO)
```

```
return
```

```
  <depart>{$emp}
  <emple>{$count}</emple>
  <medsal>{round($medsal)}</medsal>
</depart>
```

```
1 <depart>EMPLEADO<emple>4</emple>
  </depart>
2 <depart>VENDEDOR<emple>4</emple>
  </depart>
3 <depart>DIRECTOR<emple>3</emple>
  </depart>
4 <depart>ANALISTA<emple>2</emple>
  </depart>
5 <depart>PRESIDENTE<emple>1</emple>
  </depart>
```

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

- Obtén por cada zona el número de productos que tiene

```
let $pro := /productos/producc/cod_zona
let $proDis := distinct-values($pro)
return
  <Prod>
    {for $pro in $proDis
     return
       <numPro>
         {concat("producte: ", $pro, " cantitat: ",
count(/productos/producc[cod_zona=$pro]))}
       </numPro>}
    </Prod>
```

```
<Prod>
  <numPro>producte: 10 cantitat: 4</numPro>
  <numPro>producte: 20 cantitat: 3</numPro>
  <numPro>producte: 30 cantitat: 4</numPro>
  <numPro>producte: 40 cantitat: 2</numPro>
</Prod>
```

- Obtén la denominación de los productos entre las etiquetas <zona10></zona10> si son del código de zona 10, <zona20></zona20> si son del código de zona 20, etc.

```
let $pro := /productos/produc
return
<Prod>
{for $pro in $pro
let $den := $pro/denominacion
let $codZon := $pro/cod_zona
return
    if ($codZon=10)
    then <zona10>
    {data($den)}
    </zona10>
    else if ($codZon = 20)
    then <zona20>
    {data($den)}
    </zona20>
    else if ($codZon = 30)
    then <zona30>
    {data($den)}
    </zona30>
    else if ($codZon = 40)
    then <zona40>
    {data($den)}
    </zona40>
    else ()
}
</Prod>
```

```
<Prod>
  <zona10>Placa Base MSI G41M-P26</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>Tarjeta gráfica Gigabyte GeForce 1GB</zona40>
  <zona30>Tarjeta gráfica Nvidia Express 1GB</zona30>
  <zona40>Micro Intel Dual Core G620</zona40>
  <zona10>Memoria DDR3 G.Skill 2GB</zona10>
  <zona10>Memoria DDR3 G.Skill 4GB</zona10>
  <zona20>Memoria DDR3 Kingston HyperX 4GB</zona20>
  <zona30>Placa Base ASRock G41M-S3 </zona30>
</Prod>
```

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

```
let $pro := /productos/produc/cod_zona
let $proDis := distinct-values($pro)
return
  <Prod>
  {for $pro in $proDis
   return
     <numPro>
     {concat("Zona: ", $pro, " Precio maximo: ",
max(/productos/produc[cod_zona=$pro]/precio))}
     </numPro>}
  </Prod>
```

```
<Prod>
  <numPro>Zona: 10 Precio maximo: 120</numPro>
  <numPro>Zona: 20 Precio maximo: 170</numPro>
  <numPro>Zona: 30 Precio maximo: 150</numPro>
  <numPro>Zona: 40 Precio maximo: 60</numPro>
</Prod>
```

- Obtén 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 la palabra Memoria <micro></micro>, para los que contienen la palabra Micro y <otros></otros> para el resto de productos

```
let $pro := /productos/produc
return
  <Prod>
  {
    for $pro in $pro
    let $den := $pro/denominacion
    return
      if (starts-with($den, "Placa"))
      then
        <Placa>
        {data($den)}
        </Placa>
      else if (starts-with($den, "Memoria"))
      then
        <Memoria>
        {data($den)}
        </Memoria>
      else if (starts-with($den, "Micro"))
      then
        <Micro>
        {data($den)}
        </Micro>
```

```

else
<Otros>
{data($den)}
</Otros>
}
</Prod>
<Prod>
<Placa>Placa Base MSI G41M-P26</Placa>
<Micro>Micro Intel Core i5-2320</Micro>
<Micro>Micro Intel Core i5 2500</Micro>
<Otros>HD Seagate Barracuda 250GB SATA</Otros>
<Otros>HD Caviar Blue 500GB SATA3</Otros>
<Otros>Tarjeta gráfica Asus GeForce EN210 Silent 1GB</Otros>
<Otros>Tarjeta gráfica Gigabyte GeForce 1GB</Otros>
<Otros>Tarjeta gráfica Nvidia Express 1GB</Otros>
<Micro>Micro Intel Dual Core G620</Micro>
<Memoria>Memoria DDR3 G.Skill 2GB</Memoria>
<Memoria>Memoria DDR3 G.Skill 4GB</Memoria>
<Memoria>Memoria DDR3 Kingston HyperX 4GB</Memoria>
<Placa>Placa Base ASRock G41M-S3 </Placa>
</Prod>

```