

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

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

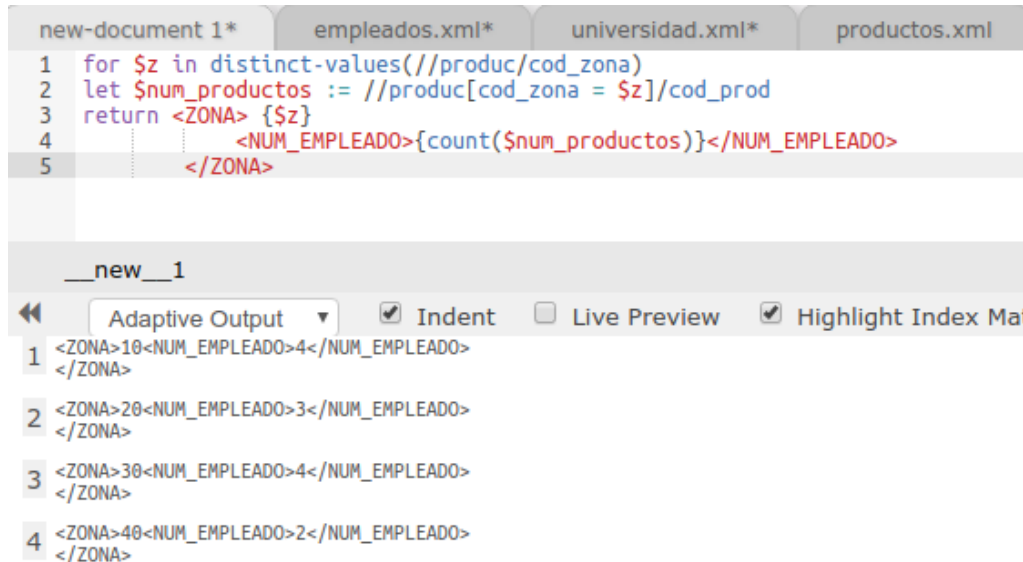
for \$z in distinct-values(//produc/cod_zona)

let \$num_productos := //produc[cod_zona = \$z]/cod_prod

return <ZONA> {\$z}

<NUM_EMPLEADO>{count(\$num_productos)}</NUM_EMPLEADO>

</ZONA>



b. 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.

for \$producto in /productos/produc

return if (\$producto/cod_zona = 10)

then <zona10>

{ \$producto/denominacion }

</zona10>

else if (\$producto/cod_zona = 20)

then <zona20>

{ \$producto/denominacion }

</zona20>

else if (\$producto/cod_zona = 30)

then <zona30>

{ \$producto/denominacion }

</zona30>

else <zona40>

{ \$producto/denominacion }

</zona40>

```

new-document 1*  empleados.xml*  universidad.xml*  productos.xml  departamentos...
1  for $producto in /productos/produ
2  return if ($producto/cod_zona = 10)
3  then <zona10>
4  { $producto/denominacion}
5  </zona10>
6  else if ($producto/cod_zona = 20)
7  then <zona20>
8  { $producto/denominacion}
9  </zona20>
10 else if ($producto/cod_zona = 30)
11 then <zona30>
12 { $producto/denominacion}
13 </zona30>
14 else <zona40>
15 { $producto/denominacion}
16 </zona40>

```

```

__new__1
1  <zona10>
2  <denominacion>Placa Base MSI G41M-P26</denominacion>
3  </zona10>
4  <zona10>
5  <denominacion>Micro Intel Core i5-2320</denominacion>
6  </zona10>
7  <zona20>
8  <denominacion>Micro Intel Core i5 2500</denominacion>
9  </zona20>
10 <zona20>
11 <denominacion>HD Seagate Barracuda 250GB SATA</denominacion>
12 </zona20>
13 <zona30>
14 <denominacion>HD Caviar Blue 500GB SATA3</denominacion>
15 </zona30>
16 <zona30>
17 <denominacion>Tarjeta gráfica Asus GeForce EN210 Silent 1GB</denominacion>
18 </zona30>
19 <zona40>
20 <denominacion>Tarjeta gráfica Gigabyte GeForce 1GB</denominacion>
21 </zona40>

```

c. Obtén por cada zona la denominación del o de los productos más caros.
for \$z in distinct-values(/productos/produ/cod_zona)
let \$denominacion := /productos/produ[cod_zona = \$z and precio = max(precio)]/denominacion
return
<zona>{\$z}
<denominacion>{\$denominacion}</denominacion>
</zona>

```

new-document 1*  empleados.xml*  universidad.xml*  productos.xml  departamentos...
1  for $z in distinct-values(/productos/produ/cod_zona)
2  let $denominacion := /productos/produ[cod_zona = $z and precio = max(precio)]/denominacion
3  return
4  <zona>{$z}
5  <denominacion>{$denominacion}</denominacion>
6  </zona>

```

```

__new__1
1  <zona>10<denominacion>
2  <denominacion>Micro Intel Core i5-2320</denominacion>
3  </denominacion>
4  </zona>
5  <zona>20<denominacion>
6  <denominacion>Micro Intel Core i5 2500</denominacion>
7  </denominacion>
8  </zona>
9  <zona>30<denominacion>
10 <denominacion>HD Caviar Blue 500GB SATA3</denominacion>
11 </denominacion>
12 </zona>
13 <zona>40<denominacion>
14 <denominacion>Micro Intel Dual Core G620</denominacion>
15 </denominacion>
16 </zona>

```

d. 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

for \$producto in //produ

```

return if ($producto[starts-with(denominacion, 'Placa Base')])
then <Placa>
    {$producto/denominacion}
</Placa>
else if ($producto[starts-with(denominacion, 'Memoria')])
then <Memoria>
    {$producto/denominacion}
</Memoria>
else if ($producto[starts-with(denominacion, 'Micro')])
then <micro>
    {$producto/denominacion}
</micro>
else <otros>
    {$producto/denominacion}
</otros>

```

The screenshot shows an XML editor with multiple tabs: 'new-document 1*', 'empleados.xml*', 'universidad.xml*', 'productos.xml', and 'departamentos...'. The 'new-document 1*' tab is active, displaying XSLT code. The code iterates through a list of products and outputs XML elements based on their denominations. The output is shown in a separate pane below the code editor.

XSLT Code:

```

1 for $producto in //produc
2 return if ($producto[starts-with(denominacion, 'Placa Base')])
3 then <Placa>
4     {$producto/denominacion}
5 </Placa>
6 else if ($producto[starts-with(denominacion, 'Memoria')])
7 then <Memoria>
8     {$producto/denominacion}
9 </Memoria>
10 else if ($producto[starts-with(denominacion, 'Micro')])
11 then <micro>
12     {$producto/denominacion}
13 </micro>
14 else <otros>
15     {$producto/denominacion}
16 </otros>

```

Output:

```

1 <Placa>
2   <denominacion>Placa Base MSI G41M-P26</denominacion>
3 </Placa>
4 <micro>
5   <denominacion>Micro Intel Core i5-2320</denominacion>
6 </micro>
7 <micro>
8   <denominacion>Micro Intel Core i5 2500</denominacion>
9 </micro>
10 <otros>
11   <denominacion>HD Seagate Barracuda 250GB SATA</denominacion>
12 </otros>
13 <otros>
14   <denominacion>HD Caviar Blue 500GB SATA3</denominacion>
15 </otros>
16 <otros>
17   <denominacion>Tarjeta gráfica Asus GeForce EN210 Silent 1GB</denominacion>
18 </otros>

```