

PRÁCTICA 5: Xquery

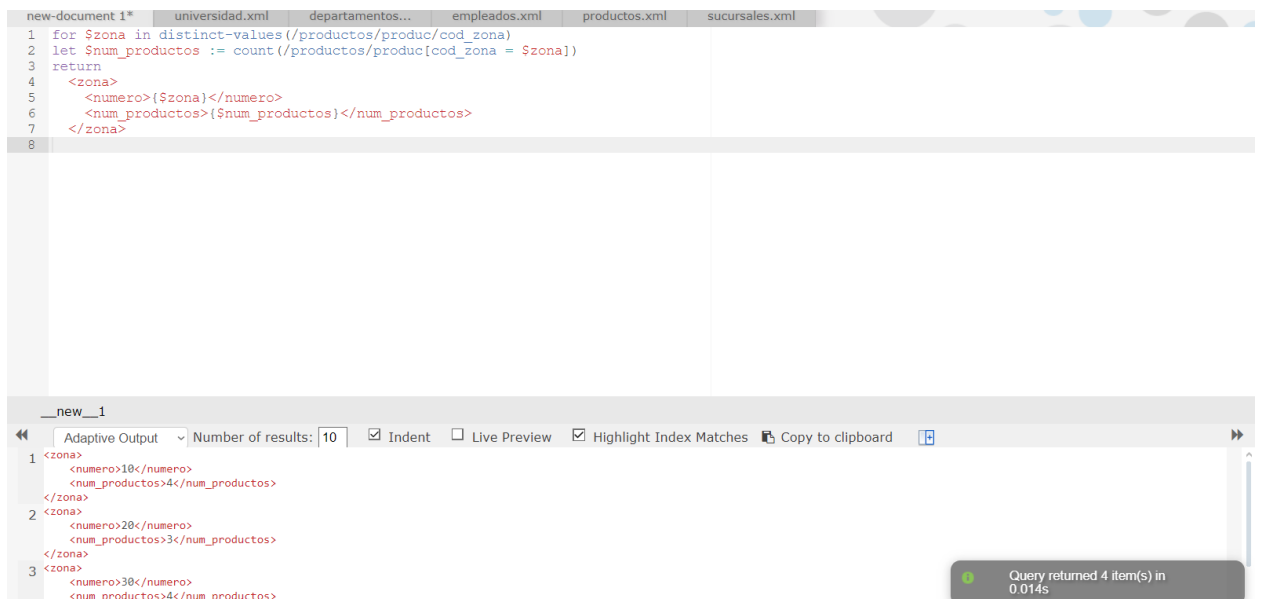
REALIZA LAS SIGUIENTES CONSULTAS Xquery:

Utilizando el documento **productos.xml**.

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

Zona: 10. Num prod = 4

```
for $zona in distinct-values(/productos/produccion/cod_zona)
let $num_productos := count(/productos/produccion[cod_zona = $zona])
return
  <zona>
    <numero>{$zona}</numero>
    <num_productos>{$num_productos}</num_productos>
  </zona>
```



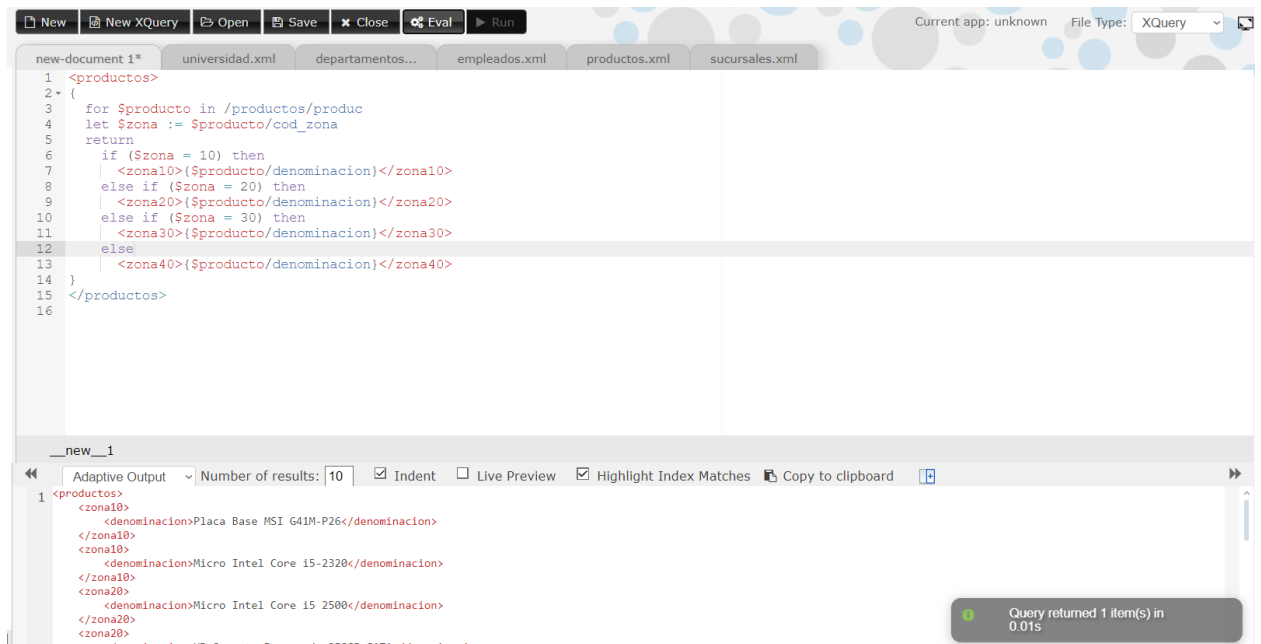
2. Obtener la denominación de los productos entre las etiquetas <zona10></zona10> si son del código de zona 10, <zona20></zona20> si son de la zona 20, <zona30></zona30> si son de la 30 y <zona40></zona40> si son de la 40.

```
<productos>
{
```

```

for $producto in /productos/produc
let $zona := $producto/cod_zona
return
  if ($zona = 10) then
    <zona10>{$producto/denominacion}</zona10>
  else if ($zona = 20) then
    <zona20>{$producto/denominacion}</zona20>
  else if ($zona = 30) then
    <zona30>{$producto/denominacion}</zona30>
  else
    <zona40>{$producto/denominacion}</zona40>
}
</productos>

```



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

```

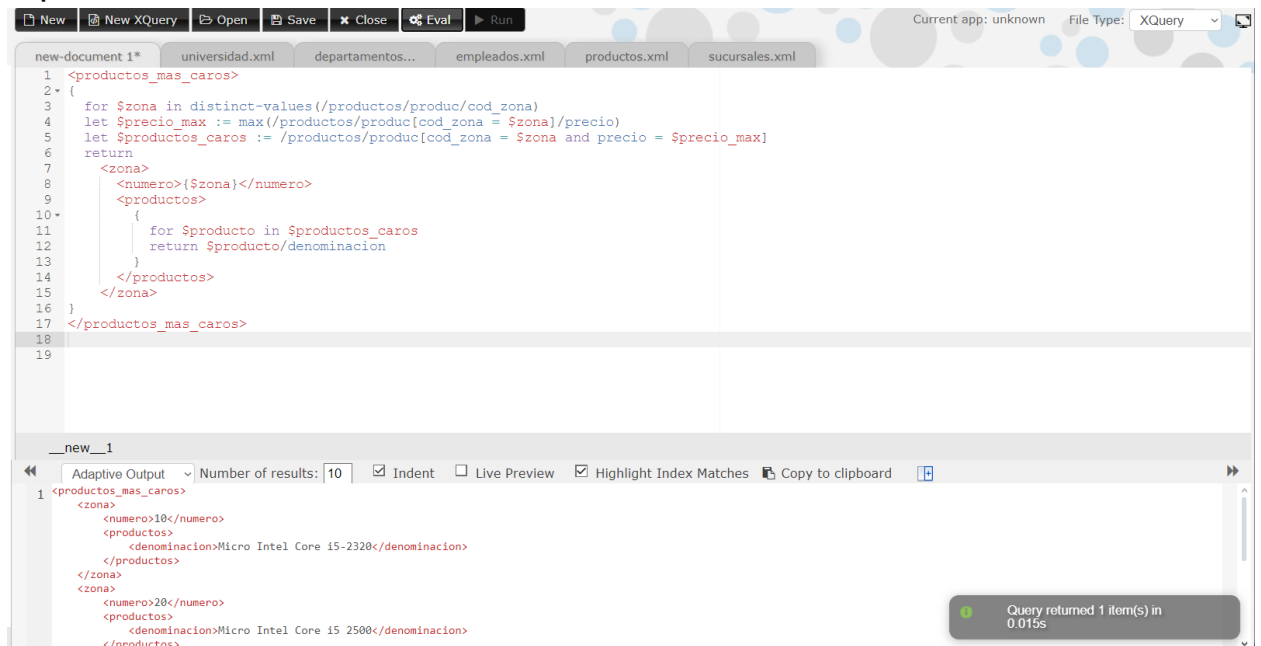
<productos_mas_caros>
{
  for $zona in distinct-values(/productos/produc/cod_zona)
  let $precio_max := max(/productos/produc[cod_zona = $zona]/precio)
  let $productos_caros := /productos/produc[cod_zona = $zona and precio
= $precio_max]
  return
    <zona>

```

```

<numero>{$zona}</numero>
<productos>
{
  for $producto in $productos_caros
  return $producto/denominacion
}
</productos>
</zona>
}
</productos_mas_caros>

```



4. 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.

```

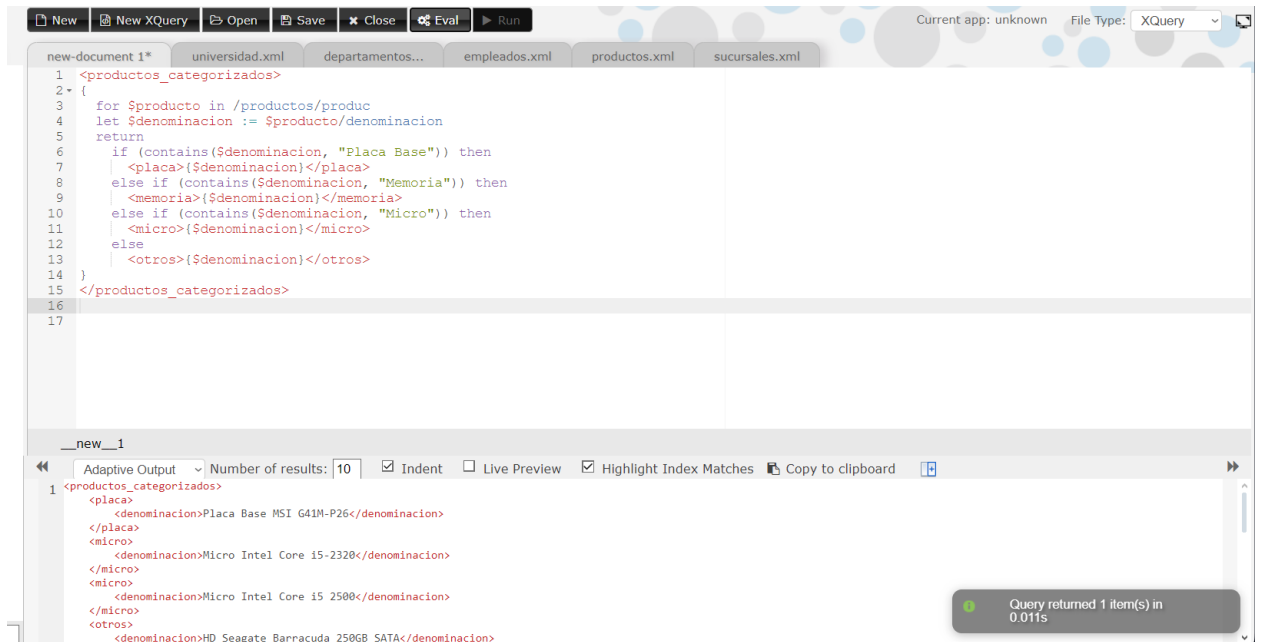
<productos_categorizados>
{
  for $producto in /productos/produccion
  let $denominacion := $producto/denominacion
  return
    if (contains($denominacion, "Placa Base")) then
      <placa>{$denominacion}</placa>
    else if (contains($denominacion, "Memoria")) then

```

```

    <memoria>{$denominacion}</memoria>
  else if (contains($denominacion, "Micro")) then
    <micro>{$denominacion}</micro>
  else
    <otros>{$denominacion}</otros>
}
</productos_categorizados>

```



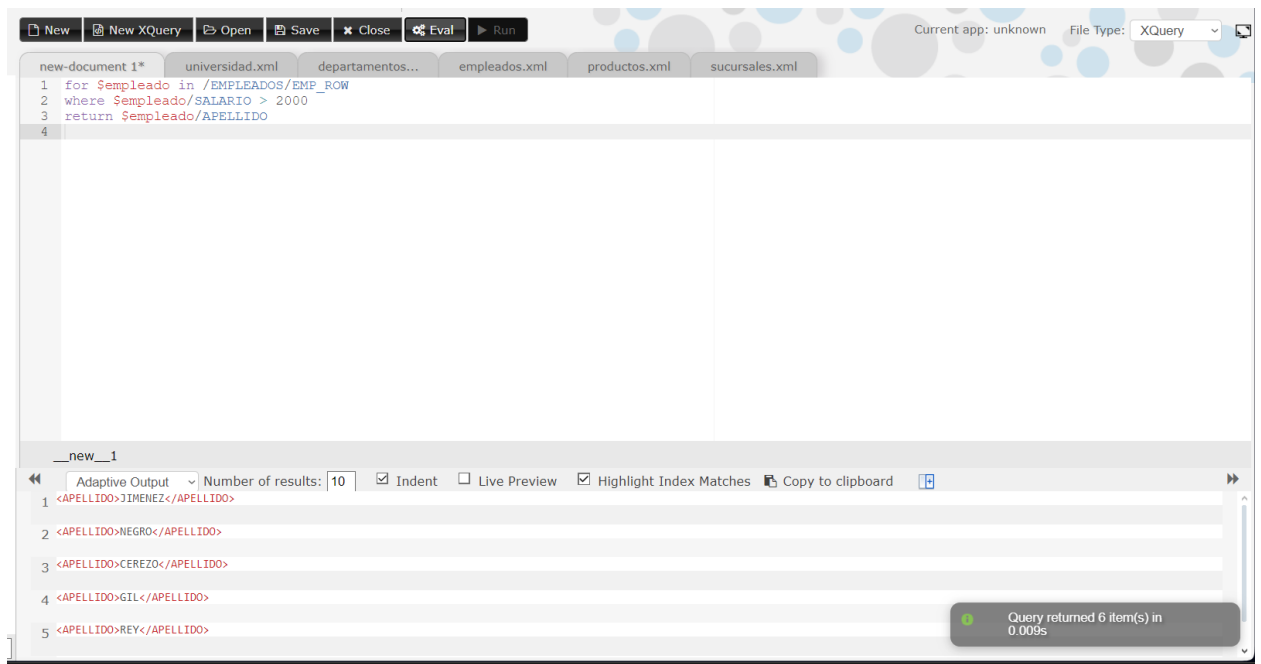
5. Invéntate 5 consultas XQuery, pon enunciado y solución.

Consulta 1: Nombres de empleados que ganan más de \$2000.
 Enunciado: Obtener los nombres de los empleados que tienen un salario mayor a \$2000. Solución:

```

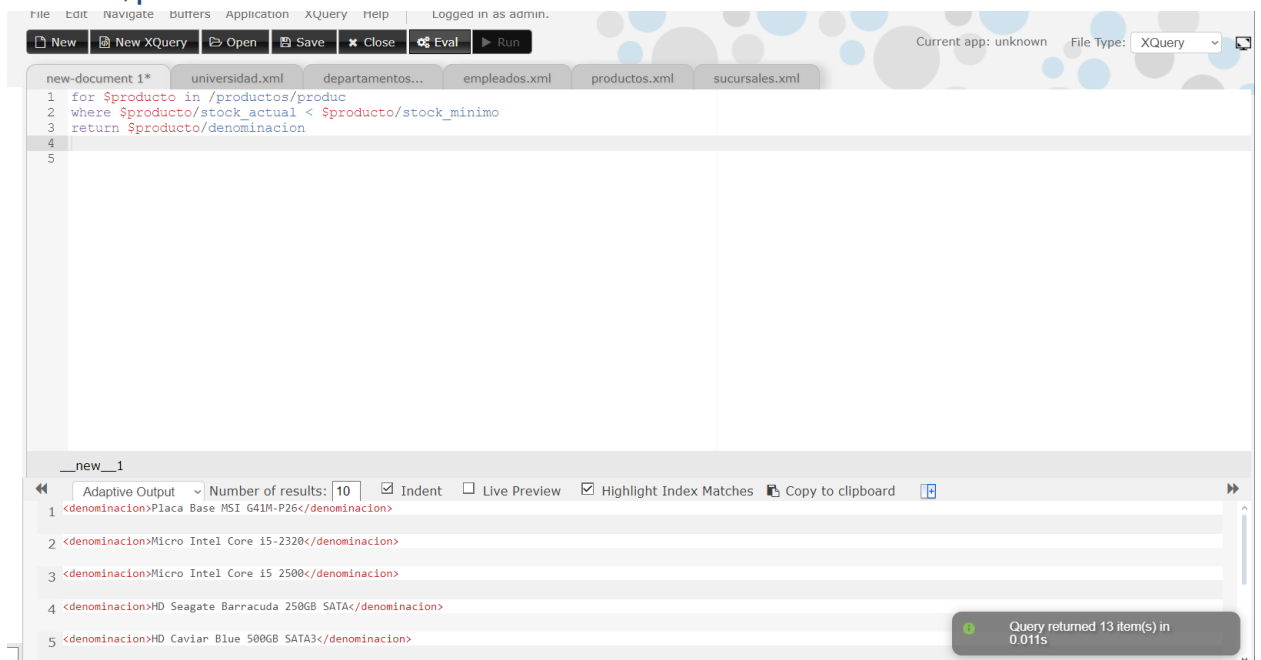
for $empleado in /EMPLEADOS/EMP_ROW
where $empleado/SALARIO > 2000
return $empleado/APELLIDO

```



Consulta 2: Productos con un stock actual menor que el stock mínimo. Enunciado: Obtener los productos que tienen un stock actual menor que su stock mínimo. Solución:

```
for $producto in /productos/produc
where $producto/stock_actual < $producto/stock_minimo
return $producto/denominación
```



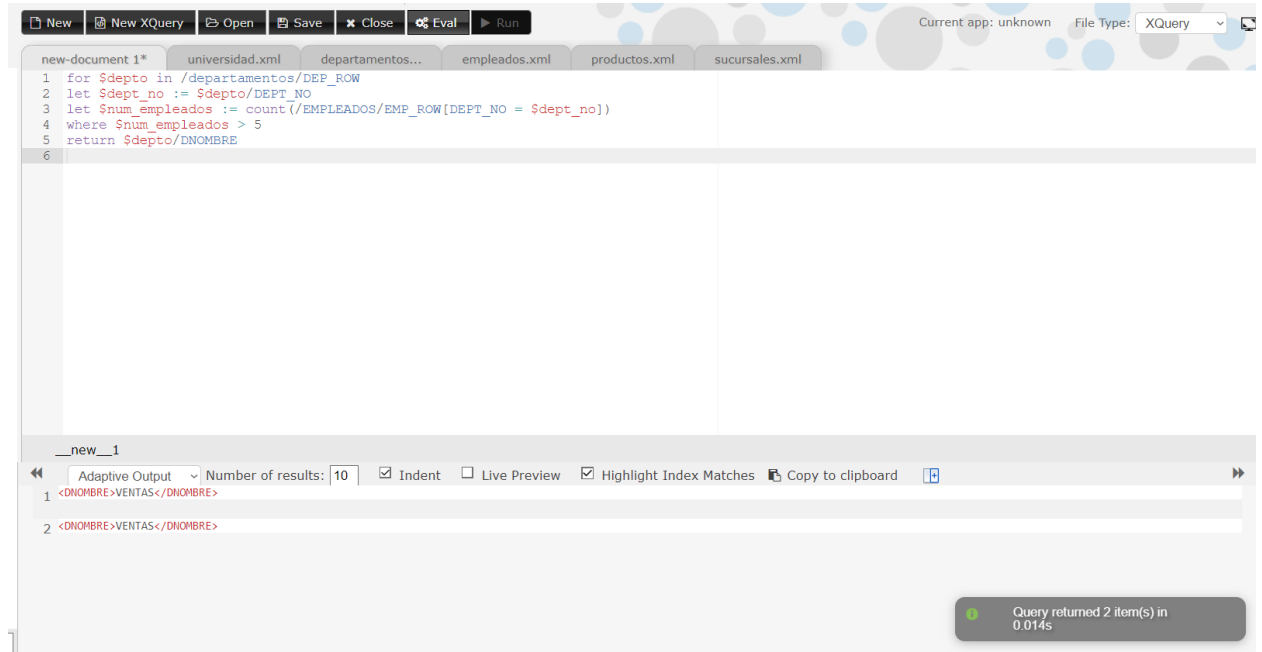
Consulta 3: Departamentos con más de 5 empleados. Enunciado: Obtener los nombres de los departamentos que tienen más de 5 empleados. Solución:

```

for $depto in /departamentos/DEP_ROW
let $dept_no := $depto/DEPT_NO
let $num_empleados := count(/EMPLEADOS/EMP_ROW[DEPT_NO =
$dept_no])
where $num_empleados > 5
return

```

\$depto/DNOMBRE



Consulta 4: Productos cuyo precio es múltiplo de 10. Enunciado: Obtener los nombres de los productos cuyo precio es un múltiplo de 10. Solución:

```

for $producto in /productos/produc
where $producto/precio mod 10 eq 0

```

return

\$producto/denominacion

The screenshot shows an XQuery editor interface. The top toolbar includes buttons for New, New XQuery, Open, Save, Close, Eval, and Run. Below the toolbar, there are tabs for various XML documents: new-document 1*, universidad.xml, departamentos..., empleados.xml, productos.xml, and sucursales.xml. The main editor area contains the following XQuery code:

```
1 for $producto in /productos/produc
2 where $producto/precio mod 10 eq 0
3 return $producto/denominacion
4
```

Below the editor, there is a results pane labeled __new__1. It shows the results of the query, which are 10 XML elements. The results are displayed in a table-like format with columns for the XML element and its value. The results are:

XML Element	Value
<denominacion>Placa Base MSI G41M-P26</denominacion>	Placa Base MSI G41M-P26
<denominacion>Micro Intel Core i5-2320</denominacion>	Micro Intel Core i5-2320
<denominacion>Micro Intel Core i5 2500</denominacion>	Micro Intel Core i5 2500
<denominacion>HD Seagate Barracuda 250GB SATA</denominacion>	HD Seagate Barracuda 250GB SATA
<denominacion>HD Caviar Blue 500GB SATA3</denominacion>	HD Caviar Blue 500GB SATA3

A status bar at the bottom right indicates: Query returned 10 item(s) in 0.008s.

Consulta 5: Empleados que fueron contratados después del año 2000. Enunciado: Obtener los nombres de los empleados que fueron contratados después del año 2000. Solución:

for \$empleado in /EMPLEADOS/EMP_ROW

where substring(\$empleado/FECHA_ALT, 1, 4) > '2000'

return

\$empleado/APELLIDO

The screenshot shows an XQuery editor interface. The top toolbar includes buttons for New, New XQuery, Open, Save, Close, Eval, and Run. Below the toolbar, there are tabs for various XML documents: new-document 1*, universidad.xml, departamentos..., empleados.xml, productos.xml, and sucursales.xml. The main editor area contains the following XQuery code:

```
1 for $empleado in /EMPLEADOS/EMP_ROW
2 where substring($empleado/FECHA_ALT, 1, 4) > '2000'
3 return $empleado/APELLIDO
```

Below the editor, there is a results pane labeled __new__1. It shows the results of the query, which are 0 XML elements. The results are displayed in a table-like format with columns for the XML element and its value. The results are:

XML Element	Value
-------------	-------

A status bar at the bottom right indicates: Query returned 0 item(s) in 0.008s.