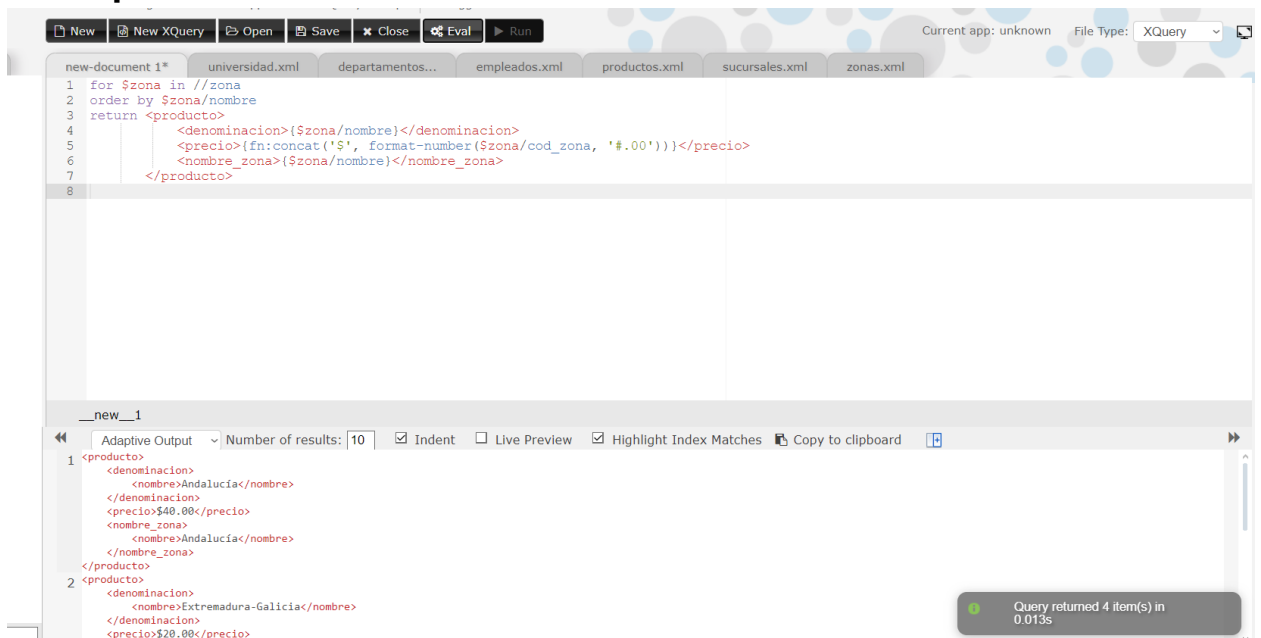


PRÁCTICA 7: Consultas complejas

Sube a la colección *Prueba* el documento **zonas.xml**, contiene información de las zonas donde se venden los productos del documento **productos.xml**. Utilizando estos dos documentos realizar las siguientes consultas XQuery:

1. Obtener los datos denominación, precio y nombre de zona de cada producto, ordenado por nombre de zona.

```
for $zona in //zona
order by $zona/nombre
return <producto>
  <denominacion>{$zona/nombre}</denominacion>
  <precio>{fn:concat('$', format-number($zona/cod_zona,
'#.00'))}</precio>
  <nombre_zona>{$zona/nombre}</nombre_zona>
</producto>
```

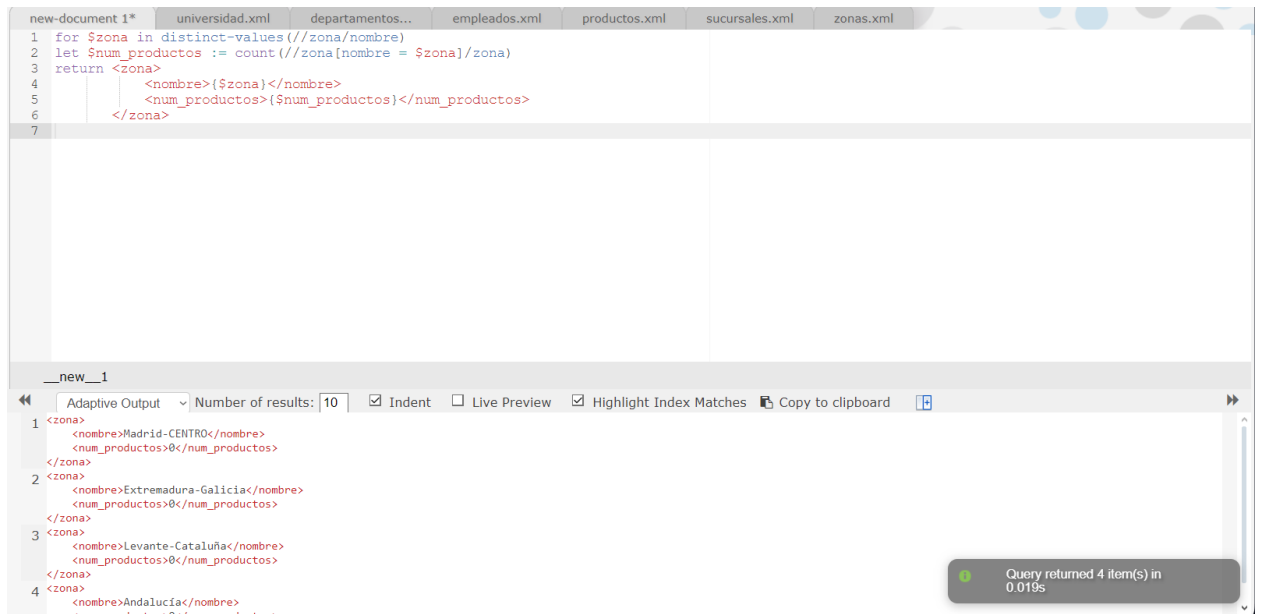


2. Obtener por cada zona, el nombre de zona y el número de productos que tiene.

```

for $zona in distinct-values(//zona/nombre)
let $num_productos := count(//zona[nombre = $zona]/zona)
return <zona>
    <nombre>{$zona}</nombre>
    <num_productos>{$num_productos}</num_productos>
</zona>

```



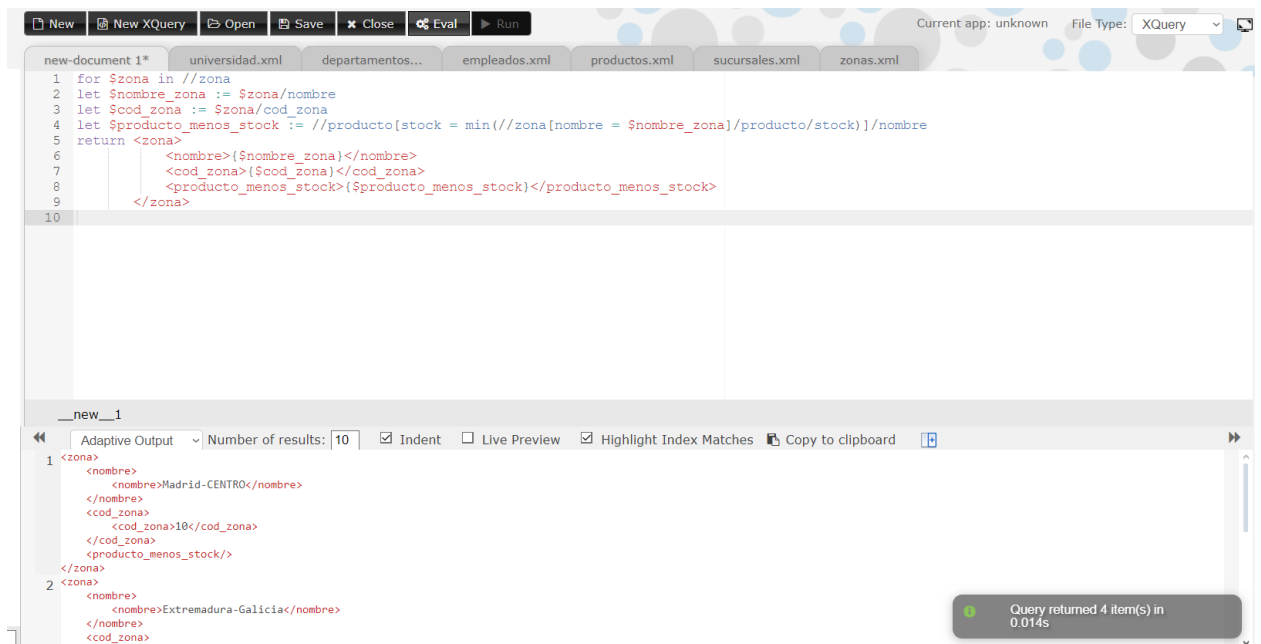
3. Obtener por cada zona, el nombre de la zona, su código y el nombre del producto con menos stock actual.

```

for $zona in //zona
let $nombre_zona := $zona/nombre
let $cod_zona := $zona/cod_zona
let $producto_menos_stock := //producto[stock = min(//zona[nombre = $nombre_zona]/producto/stock)]/nombre
return <zona>
    <nombre>{$nombre_zona}</nombre>
    <cod_zona>{$cod_zona}</cod_zona>

<producto_menos_stock>{$producto_menos_stock}</producto_me
nos_stock>
</zona>

```



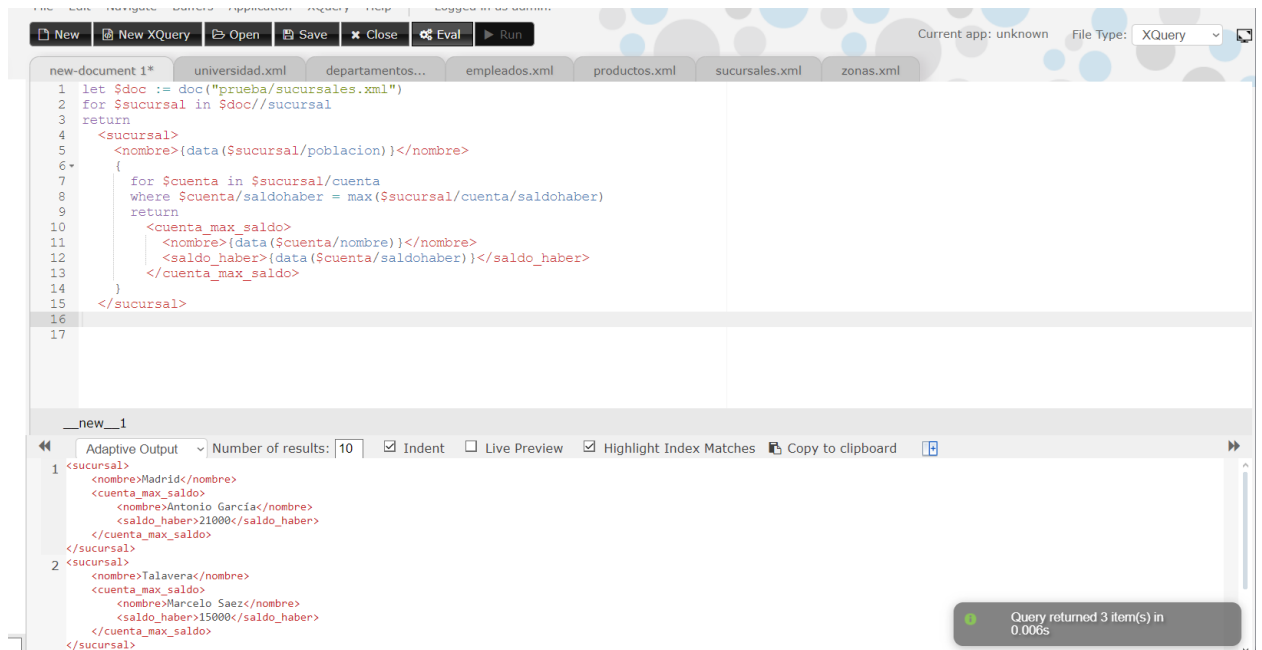
Utiliza el documento **sucursales.xml** para realizar las siguientes consultas XQuery usando dos for:

4. Obtener por cada sucursal el mayor saldo haber y el nombre de la cuenta que tiene ese saldo.

```

let $doc := doc("prueba/sucursales.xml")
for $sucursal in $doc//sucursal
return
  <sucursal>
    <nombre>{data($sucursal/poblacion)}</nombre>
    {
      for $cuenta in $sucursal/cuenta
      where $cuenta/saldohaber = max($sucursal/cuenta/saldohaber)
      return
        <cuenta_max_saldo>
          <nombre>{data($cuenta/nombre)}</nombre>
          <saldo_haber>{data($cuenta/saldohaber)}</saldo_haber>
        </cuenta_max_saldo>
    }
  </sucursal>

```



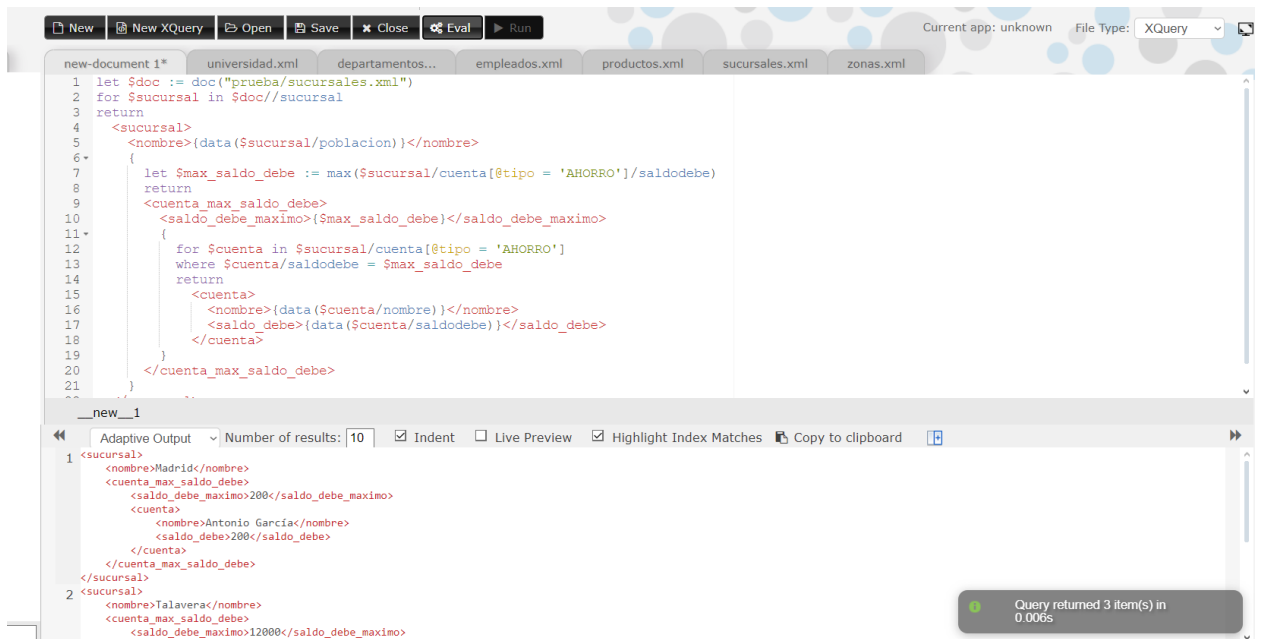
5. Obtener por cada sucursal el nombre de la cuenta del tipo AHORRO cuyo saldo debe ser el máximo. Sacar también el máximo.

```

let $doc := doc("prueba/sucursales.xml")
for $sucursal in $doc//sucursal
return
  <sucursal>
    <nombre>{data($sucursal/poblacion)}</nombre>
    {
      let $max_saldo_debe := max($sucursal/cuenta[@tipo =
'AHORRO']/saldodebe)
      return
        <cuenta_max_saldo_debe>
          <saldo_debe_maximo>{$max_saldo_debe}</saldo_debe_maximo>
          {
            for $cuenta in $sucursal/cuenta[@tipo = 'AHORRO']
            where $cuenta/saldodebe = $max_saldo_debe
            return
              <cuenta>
                <nombre>{data($cuenta/nombre)}</nombre>
                <saldo_debe>{data($cuenta/saldodebe)}</saldo_debe>
              </cuenta>
            }
          }
        </cuenta_max_saldo_debe>
    }
  }

```

```
}
</sucursal>
```

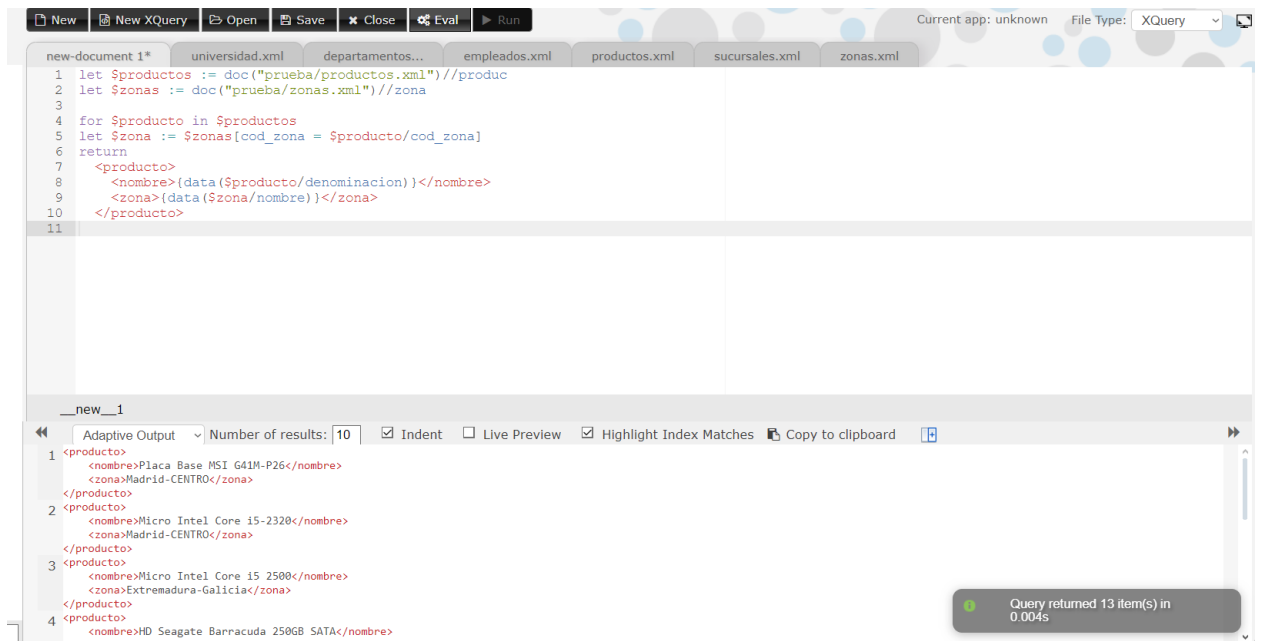


Utiliza los documentos *productos.xml* y *zonas.xml*

6. Visualiza los nombres de productos con su nombre de zona. Utiliza dos for en la consulta.

```
let $productos := doc("prueba/productos.xml")//produc
let $zonas := doc("prueba/zonas.xml")//zona
```

```
for $producto in $productos
let $zona := $zonas[cod_zona = $producto/cod_zona]
return
  <producto>
    <nombre>{data($producto/denominacion)}</nombre>
    <zona>{data($zona/nombre)}</zona>
  </producto>
```



7. Visualiza los nombres de productos con stock_minimo > 5. su código de zona, su nombre y el director de esa zona. Utiliza dos for en la consulta.

```

let $productos := doc("prueba/productos.xml")//produc
let $zonas := doc("prueba/zonas.xml")//zona

```

```

for $producto in $productos
for $zona in $zonas
where $producto/stock_minimo > 5 and $producto/cod_zona =
$zona/cod_zona
return
  <producto>
    <nombre>{data($producto/denominacion)}</nombre>
    <codigo_zona>{data($zona/cod_zona)}</codigo_zona>
    <nombre_zona>{data($zona/nombre)}</nombre_zona>
    <director_zona>{data($zona/director)}</director_zona>
  </producto>

```

NewNew XQueryOpenSaveCloseEvalRun

Current app: unknownFile Type: XQuery

new-document 1*universidad.xmldepartamentos...empleados.xmlproductos.xmlsucursales.xmlzonas.xml

```
1 let $productos := doc("prueba/productos.xml")//produc
2 let $zonas := doc("prueba/zonas.xml")//zona
3
4 for $producto in $productos
5 for $zona in $zonas
6 where $producto/stock_minimo > 5 and $producto/cod_zona = $zona/cod_zona
7 return
8   <producto>
9     <nombre>{data($producto/denominacion)}</nombre>
10    <codigo_zona>{data($zona/cod_zona)}</codigo_zona>
11    <nombre_zona>{data($zona/nombre)}</nombre_zona>
12    <director_zona>{data($zona/director)}</director_zona>
13  </producto>
14
```

__new__1

Adaptive OutputNumber of results: 10IndentLive PreviewHighlight Index MatchesCopy to clipboard

```
1 <producto>
  <nombre>Micro Intel Core i5 2500</nombre>
  <codigo_zona>20</codigo_zona>
  <nombre_zona>Extremadura-Galicia</nombre_zona>
  <director_zona>Alicia Pérez</director_zona>
</producto>
2 <producto>
  <nombre>HD Caviar Blue 500GB SATA3</nombre>
  <codigo_zona>30</codigo_zona>
  <nombre_zona>Levante-Cataluña</nombre_zona>
  <director_zona>Carlos Grau</director_zona>
</producto>
3 <producto>
  <nombre>Tarjeta gráfica Gigabyte GeForce 1GB</nombre>
```

Query returned 4 item(s) in 0.005s