

ACTIVITATS UT 7. EMMAGATZEMAMENT D'INFORMACIÓ

XQuery

Activitat 1. Utilitza el fitxer sucursales.xml

Escribe usando Xquery y FLWOR las consultas oportunas para obtener la siguiente información:

1. Los datos de las cuentas de tipo AHORRO.

```
for $cuenta in //cuenta[@tipo="AHORRO"]  
return $cuenta
```

2. De cada sucursal la concatenación de su código y el nº de cuentas de ahorro que tiene.

```
for $sucursal in //sucursal  
return concat($sucursal/@codigo, " - ", count($sucursal/cuenta[@tipo="AHORRO"]))
```

3. Las cuentas de tipo PENSIONES de la sucursal de código SUC3.

```
for $sucursal in //sucursal  
where $sucursal/@codigo = "SUC3"  
return $sucursal/cuenta[@tipo = "PENSIONES"]
```

4. De cada sucursal la concatenación de los datos código de sucursal, director y total saldo haber.

```
for $sucursal in //sucursal  
let $saldo := sum($sucursal/cuenta/saldohaber)  
return concat($sucursal/@codigo, " - ", $sucursal/director, " - ", $saldo)
```

5. Todos los elementos de las sucursales con más de 3 cuentas.

```
for $sucursal in //sucursal  
where (count($sucursal/cuenta) > 3)  
return $sucursal
```

6. Todos los elementos de las sucursales con más de 3 cuentas del tipo AHORRO.

```
for $sucursal in //sucursal  
where (count($sucursal/cuenta[@tipo="AHORRO"]) > 3)  
return $sucursal
```

7. Los nodos del director y la población de las sucursales de con más de 3 cuentas.

```
for $sucursal in //sucursal[count(cuenta) > 3]
return ($sucursal/director, $sucursal/poblacion)
```

8. El número de sucursales de Madrid.

```
for $sucursal in //sucursal/poblacion = "Madrid"
return count($sucursal)
```

9. De cada sucursal, su código y la suma de las aportaciones de las cuentas de tipo PENSIONES.

```
for $sucursal in //sucursal
let $saldo := sum($sucursal/cuenta[@tipo="PENSIONES"]/aportacion)
return concat($sucursal/@codigo, " - ", $saldo)
```

10. Los nodos número de cuenta, nombre de cuenta y el saldo haber de las cuentas con saldo haber mayor de 10000.

```
for $cuenta in //cuenta[./saldohaber > 10000]
return ($cuenta/numero, $cuenta/nombre, $cuenta/saldohaber)
```

11. De cada sucursal con más de 3 cuentas de tipo AHORRO, su código, y la suma del saldo debe de esas cuentas.

```
for $sucursal in //sucursal
where (count($sucursal/cuenta[@tipo="AHORRO"]) > 3)
let $saldo := sum($sucursal/cuenta/saldodebe)
return ($sucursal/@codigo, $saldo)
```

12. Por cada sucursal, el código de la sucursal y el número de cuentas que tiene de tipo AHORRO y de tipo PENSIONES.

```
for $sucursal in //sucursal
return (
  <SUCURSAL>{$sucursal/@codigo}</SUCURSAL>,
  <AHORRO>{count($sucursal/cuenta[@tipo='AHORRO'])}</AHORRO>,
  <PENSIONES>{count($sucursal/cuenta[@tipo='PENSIONES'])}</PENSIONES>
)
```

13. Por cada sucursal el código de sucursal, el director, la población, la suma del total debe y la suma del total haber de sus cuentas.

```
for $sucursal in //sucursal
return (
  <codigo>{$sucursal/@codigo}</codigo>,
  <director>{$sucursal/director/text()}</director>,
  <poblacion>{$sucursal/poblacion/text()}</poblacion>,
  <suma_debe>{sum($sucursal/cuenta/saldodebe)}</suma_debe>,
  <suma_haber>{sum($sucursal/cuenta/saldohaber)}</suma_haber>
)
```

14. El nombre de los directores, el código de sucursal y la población de las sucursales con más de 3 cuentas.

```
for $sucursal in //sucursal[count(cuenta) > 3]
return (
  <nombre_director>{$sucursal/director/text()}</nombre_director>,
  <codigo_sucursal>{$sucursal/@codigo}</codigo_sucursal>,
  <poblacion>{$sucursal/poblacion/text()}</poblacion>
)
```

15. Por cada sucursal el código de sucursal y los datos de las cuentas con más saldo debe.

```
for $sucursal in //sucursal
let $maxSaldo := max($sucursal/cuenta/saldodebe)
where $sucursal/cuenta[saldodebe = $maxSaldo]
return (
  <codigo_sucursal>{$sucursal/@codigo}</codigo_sucursal>,
  <cuenta>{$sucursal/cuenta[saldodebe = $maxSaldo]}</cuenta>
)
```

16. La cuenta del tipo PENSIONES que ha hecho más aportación.

```
let $maxAportacion := max(//cuenta[@tipo = "PENSIONES"]/aportacion)
return //cuenta[@tipo = "PENSIONES" and aportacion = $maxAportacion]
```

Activitat 2. Utilitza el fitxer productos.xml

Escribe con XQuery y FLWOR las consultas oportunas para obtener la siguiente información:

1. Por cada zona el número de productos que tiene.

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

2. Las denominaciones de los productos encerrándolas en las etiquetas si son del código de zona 10, si son de la zona 20, si son de la zona 30, si son de la 40 y en cualquier otro caso.

```
for $producto in //producto
return
if ($producto/cod_zona = "10") then
<zona_10>{data($producto/denominacion)}</zona_10>
else if ($producto/cod_zona = "20") then
<zona_20>{data($producto/denominacion)}</zona_20>
else if ($producto/cod_zona = "30") then
<zona_30>{data($producto/denominacion)}</zona_30>
else if ($producto/cod_zona = "40") then
<zona_40>{data($producto/denominacion)}</zona_40>
else
<otra_zona>{data($producto/denominacion)}</otra_zona>
```

3. Las denominaciones de los productos encerrándolas en las etiquetas siendo XX la zona del producto.

```
for $producto in //producto
let $zona := $producto/cod_zona
let $etiqueta_zona :=
if ($zona = 10) then "zona_10"
else if ($zona = 20) then "zona_20"
else if ($zona = 30) then "zona_30"
else if ($zona = 40) then "zona_40"
else "zona_XX"
return
element {$etiqueta_zona} {data($producto/denominacion)}
```

4. Por cada zona la denominación del o de los productos más caros.

```
for $zona in distinct-values(//cod_zona)
let $productosZona := //producto[cod_zona = $zona]
let $precioMax := max($productosZona/precio)
let $productoCaro := $productosZona[precio = $precioMax]
return
<zona codigo="{ $zona}">
<productoCaro>
<denominacion>{data($productoCaro/denominacion)}</denominacion>
<precio>{data($precioMax)}</precio>
</productoCaro>
</zona>
```

5. La denominación de los productos, estas denominaciones deberán estar encerradas en las etiquetas para los productos en cuya denominación aparece la palabra Placa Base, entre las etiquetas para los que contienen la palabra Memoria, encerradas en , para los que contienen la palabra Micro y para el resto de los productos.

Activitat 3. Utilitza els fitxers productos.xml i zonas.xml

Escribe las instrucciones con XQuery y FLWOR para obtener:

1. Los datos denominación, precio y nombre de la zona de cada producto, ordenado por nombre de zona.

```
for $zona in doc("zonas.xml")//zona
let $productos := doc("productos.xml")//producto[cod_zona = $zona/cod_zona]
order by $zona/nombre
return
for $producto in $productos
return
<producto>
<denominacion>{data($producto/denominacion)}</denominacion>
<precio>{data($producto/precio)}</precio>
<nombreZona>{data($zona/nombre)}</nombreZona>
</producto>
```

2. Por cada zona, el nombre de la zona y el número de productos que tiene.

```
for $zona in doc("zonas.xml")//zona
return
<zona>
<nombre>{data($zona/nombre)}</nombre>
<numero_productos>{count(doc("productos.xml")//producto[cod_zona =
$zona/cod_zona])}</numero_productos>
</zona>
```

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

```
ffor $zona in doc("zonas.xml")//zona
let $productosZona := doc("productos.xml")//producto[cod_zona = $zona/cod_zona]
let $menosStock := $productosZona[stock_actual = min($productosZona/stock_actual)]
return
<zona>
<nombre>{data($zona/nombre)}</nombre>
<codigo>{data($zona/cod_zona)}</codigo>
<producto_menos_stock>{data($menosStock/denominacion)}</producto_menos_stock>
</zona>
```

4. La denominación de los productos con su nombre de zona. Utiliza dos for en la consulta.

```
for $producto in doc("productos.xml")//producto
for $zona in doc("zonas.xml")//zona[cod_zona = $producto/cod_zona]
return
<producto>
<denominacion>{data($producto/denominacion)}</denominacion>
<nombre_zona>{data($zona/nombre)}</nombre_zona>
</producto>
```

5. La denominación de los productos con stock-minimo mayor que 5, su código de zona, el nombre de la zona y el director de esa zona. Utiliza dos for en la consulta.

```
for $producto in doc("productos.xml")//producto
for $zona in doc("zonas.xml")//zona
where $producto/stock_minimo > 5 and $producto/cod_zona = $zona/cod_zona
return
<producto_con_info>
<denominacion>{$producto/denominacion/text()}</denominacion>
<cod_zona>{$producto/cod_zona/text()}</cod_zona>
<nombre_zona>{$zona/nombre/text()}</nombre_zona>
<director_zona>{$zona/director/text()}</director_zona>
</producto_con_info>
```