

---

## ACTIVIDAD 5.1

Dado el documento *productos.xml* que está dentro de la colección *ColeccionPruebas*, con información de datos de productos, y cuya estructura es la siguiente:

```
<produc>
  <cod_prod>xxxxxx</cod_prod>
  <denominacion>xxxxxxxxxxxx</denominacion>
  <precio>xxxx</precio>
  <stock_actual>xxx</stock_actual>
  <stock_minimo>xxxx</stock_minimo>
  <cod_zona>xxxx</cod_zona>
</produc>
```

Realizar las siguientes consultas XPath:

- **Obtener los nodos denominación y precio de todos los productos.**

Solución 1:

```
/productos/produc/denominacion | /productos/produc/precio
```

Solución 2:

```
/productos/produc/(denominacion|precio)
```

- **Obtener los nodos de los productos que sean placas base.**

```
/productos/produc[contains(denominacion,'Placa Base')]
```

- **Obtener los nodos de los productos con precio mayor de 60 € y de la zona 20.**

```
/productos/produc[precio>60 and cod_zona=20]
```

- **Obtener el número de productos que sean memorias y de la zona 10.**

```
count(/productos/produc[contains(denominacion,'Memoria') and  
cod_zona=10])
```

Visualizamos si es cierto el número que nos devuelve mediante:

```
/productos/produc[contains(denominacion,'Memoria') and cod_zona=10]
```

- **Obtener la media de precio de los micros.**

Solución 1:

```
avg(/productos/produc[contains(denominacion,'Micro')]/precio)
```

Solución 2 con formato:

format-

```
number(avg(/productos/produc[contains(denominacion,'Micro')]/precio),'  
0.00')
```

- **Obtener los datos de los productos cuyo stock mínimo sea mayor que su stock actual.**

```
/productos/produc[number(stock_minimo)>number(stock_actual)]
```

Si no utilizas number entonces los resultados no serían correctos:  
/productos/produc[stock\_minimo>stock\_actual]

- **Obtener el nombre de producto y el precio de aquellos cuyo stock mínimo sea mayor que su stock actual y sean de la zona 40**

Solución 1:

```
/productos/produc[number(stock_minimo)>number(stock_actual) and  
cod_zona=40]/denominacion |  
/productos/produc[number(stock_minimo)>number(stock_actual) and  
cod_zona=40]/precio
```

Solución 2:

```
/productos/produc[number(stock_minimo)>number(stock_actual) and  
cod_zona=40]/(denominacion | precio)
```

Solución 3:

```
/productos/produc[number(stock_minimo)>number(stock_actual) and  
cod_zona=40]/concat (denominacion,' ', precio = ',precio)
```

- **Obtener el producto más caro.**

```
/productos/produc[precio=max(precio)]
```

- **Obtener el producto más barato de la zona 20.**

```
/productos/produc[cod_zona=20 and precio=min(precio)]
```

- **Obtener el producto más caro de la zona 10.**

```
/productos/produc[cod_zona=10 and precio=max(precio)]
```

---

## ACTIVIDAD 5.2

Dado el documento *sucursales.xml* que se encuentra dentro de la colección *ColeccionPruebas*. Este documento contiene los datos de las sucursales de un banco. Por cada sucursal tenemos el teléfono, el código, el director de la sucursal, la población y las cuentas de la sucursal. Y por cada cuenta tenemos el tipo de cuenta AHORRO o PENSIONES, el nombre de la cuenta, el número, el saldo haber y el saldo debe. Estos datos son:

```
<sucursales>  
  <sucursal telefono="xxxxxxx" codigo="xxxx">  
    <director>xxxxxxxxxxxxxxxxxx</director>  
    <poblacion>xxxxxxxxxx</poblacion>  
    <cuenta tipo="xxxxxxx">  
      <nombre>xxxx</nombre>  
      <numero>xxxx</numero>  
      <saldohaber>xxxxxx</saldohaber>  
      <saldodebe>xxxxxx</saldodebe>  
    </cuenta>  
    . . . . .  
  </sucursal>  
  . . . . .
```

```
</sucursales>
```

Realiza las siguientes consultas XPath:

- **Obtener los datos de las cuentas bancarias cuyo tipo sea AHORRO.**  
`/sucursales/sucursal/cuenta[@tipo="AHORRO"]`
- **Obtener por cada sucursal la concatenación de su código, y el número de cuentas del tipo AHORRO que tiene**  
`/sucursales/sucursal/concat(@codigo, ' ', count(cuenta[@tipo="AHORRO"]))`
- **Obtener las cuentas de tipo PENSIONES de la sucursal con código SUC3.**  
`/sucursales/sucursal[@codigo='SUC3']/cuenta[@tipo='PENSIONES']`
- **Obtener por cada sucursal la concatenación de los datos, código sucursal, director, y total saldo haber**  
`/sucursales/sucursal/concat(@codigo, ' ', director, ' ', sum(cuenta/saldohaber))`
- **Obtener todos los elementos de las sucursales con más de 3 cuentas.**  
`/sucursales/sucursal[count(cuenta)>3]`
- **Obtener todos los elementos de las sucursales con más de 3 cuentas del tipo AHORRO.**  
`/sucursales/sucursal[count(cuenta[@tipo="AHORRO"])>3]`
- **Obtener los nodos del director y la población de las sucursales con más de 3 cuentas.**  
Solución 1:  
`/sucursales/sucursal[count(cuenta)>3]/director |  
/sucursales/sucursal[count(cuenta)>3]/poblacion`  
Solución 2:  
`/sucursales/sucursal[count(cuenta)>3]/(director|poblacion)`
- **Obtener el número de sucursales cuya población sea Madrid.**  
`count(/sucursales/sucursal[poblacion='Madrid'])`
- **Obtener por cada sucursal, su código y la suma de las aportaciones de las cuentas del tipo PENSIONES.**  
`/sucursales/sucursal/concat(@codigo, '  
' , sum(cuenta[@tipo='PENSIONES']/aportacion))`

- Obtener los nodos número de cuenta, nombre de cuenta y el saldo haber de las cuentas con saldo haber mayor de 10000.

Solución 1:

```
/sucursales/sucursal/cuenta[saldohaber>10000]/numero |  
/sucursales/sucursal/cuenta[saldohaber>10000]/nombre |  
/sucursales/sucursal/cuenta[saldohaber>10000]/saldohaber
```

Solución 2:

```
/sucursales/sucursal/cuenta[saldohaber>10000]/(numero | nombre |  
saldohaber)
```

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

```
/sucursales/sucursal[count(cuenta[@tipo="AHORRO"])>3]/concat(@codigo, '  
' , sum(cuenta[@tipo='AHORRO']/saldodebe))
```

---