

# Actividades

## Actividad 1.

A partir del XML empleados.xml crea rutas de localización XPath que seleccionen los siguientes nodos o conjuntos de nodos. Recuerda que XPATH es sensible a mayúsculas y minúsculas:

- a) selecciona todos los elementos o nodos de los empleados del departamento 10.

`/EMPLEADOS/EMPLEADO[number(DEPT_NO)=10]`

- b) selecciona los nodos APELLIDO y DEPT\_NO de los empleados.

`/EMPLEADOS/EMPLEADO/(APELLIDO|DEPT_NO)`

- c) selecciona los apellidos de los empleados del departamento 10.

`/EMPLEADOS/EMPLEADO[number(DEPT_NO)=10]/APELLIDO`

- d) selecciona todos los elementos o nodos de los empleados que NO pertenezcan al departamento 10. `/EMPLEADOS/EMPLEADO[number(DEPT_NO)!=10]`

- e) selecciona los APELLIDOS de los empleados que NO son analistas.

`/EMPLEADOS/EMPLEADO[OFICIO!="ANALISTA"]/APELLIDO`

- f) selecciona el APELLIDO y el OFICIO de los empleados del departamento 10.

`/EMPLEADOS/EMPLEADO[number(DEPT_NO)=10]/(APELLIDO|OFICIO)`

- g) devuelve los datos de los empleados con SALARIO mayor de 1300 y del departamento

10. `/EMPLEADOS/EMPLEADO[number(SALARIO)>1300 and number(DEPT_NO)=10]`

- h) devuelve el APELLIDO y el OFICIO de los empleados con SALARIO mayor de 1300

y del departamento 20.

`/EMPLEADOS/EMPLEADO[number(SALARIO)>1300 and number(DEPT_NO)=20]/(SALARIO|OFICIO)`

- i) devuelve todos los datos del primer empleado.

`/EMPLEADOS/EMPLEADO[number(1)]`

- j) devuelve el APELLIDO del quinto empleado.

`/EMPLEADOS/EMPLEADO[number(5)]/APELLIDO`

- k) selecciona todos los datos del último empleado.

`/EMPLEADOS/EMPLEADO[last()]`

- l) devuelve el APELLIDO del penúltimo empleado.

`/EMPLEADOS/EMPLEADO[last()-1]`

m) selecciona todos los elementos del empleado que ocupa la posición 3.

`/EMPLEADOS/EMPLEADO[3]`

n) selecciona el apellido de los elementos cuya posición es menor de 3, es decir devuelve

los apellidos del primer y segundo empleado `/EMPLEADOS/EMPLEADO[position() < 3]`

o) cuenta el número de empleados del departamento 10.

`/EMPLEADOS/count(EMPLEADO[number(DEPT_NO)=10])`

p) cuenta el nº de empleados con oficio EMPLEADO y SALARIO mayor de 1300.

`/EMPLEADOS/count(EMPLEADO[number(SALARIO)>1300 and OFICIO = "EMPLEADO"])`

q) devuelve la suma de SALARIO de los empleados del departamento 20.

`/EMPLEADOS/sum(EMPLEADO[number(DEPT_NO)=20]/SALARIO)`

r) devuelve el salario máximo

`/EMPLEADOS/max(EMPLEADO/SALARIO)`

s) devuelve el salario mínimo

`/EMPLEADOS/min(EMPLEADO/SALARIO)`

t) devuelve el salario mínimo de los empleados con oficio Analista.

`/EMPLEADOS/min(EMPLEADO[OFICIO="ANALISTA"]/SALARIO)`

u) devuelve la media del salario redondeada a 3 decimales.

`/EMPLEADOS/round(avg(EMPLEADO/SALARIO),3)`

v) devuelve la media del salario de los empleados del departamento 20.

`/EMPLEADOS/avg(EMPLEADO[number(DEPT_NO)=20]/SALARIO)`

w) devuelve el apellido y el oficio concatenados de los empleados del departamento 10.

`/EMPLEADOS/EMPLEADO[number(DEPT_NO)=10]/concat(APELLIDO,"_",OFICIO)`

x) selecciona los elementos de los empleados cuyo APELLIDO empieza por A

`/EMPLEADOS/EMPLEADO[starts-with(APELLIDO,"A")]`

y) selecciona APELLIDO y OFICIO concatenados de los empleados cuyo OFICIO

empieza por "A"

`/EMPLEADOS/EMPLEADO[starts-with(APELLIDO,"A")]/concat(APELLIDO,"_",OFICIO)`

z) devuelve los oficios que contienen la sílaba 'OR'.

`/EMPLEADOS/EMPLEADO[contains(OFICIO,"OR")]/OFICIO`

aa) devuelve los apellidos que contienen una "A".

`/EMPLEADOS/EMPLEADO[contains(APELLIDO,"A")]/APELLIDO`

## Actividad 2.

El documento productos.xml contiene los datos de los productos de una distribuidora de componentes informáticos. Por cada producto tenemos el código, la denominación, el precio, el stock actual, el stock mínimo y el código de zona. Estos datos son:

```
<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>
```

Realiza las siguientes consultas XPath:

- a) Obtén los nodos denominación y precio de todos los productos.

`productos/produc/(denominacion|precio)`

- b) Obtén los nodos de los productos que sean placas base.

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

- c) Obtén los nodos de los productos con precio mayor de 60 € y de la zona 20.

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

- d) Obtén el número de productos que sean memorias y de la zona 10.

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

- e) Obtén la media de precio de los micros.

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

- f) Obtén los datos de los productos cuyo stock mínimo sea mayor que su stock actual.

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

- g) Obtén el nombre de producto y el precio de aquellos cuyo stock mínimo sea mayor que

su stock actual y sean de la zona 40.

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

- h) Obtén el producto más caro.

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

- i) Obtén el producto más barato de la zona 20.

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

- j) Obtén el producto más caro de la zona 10.

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

## Actividad 3.

El documento sucursales.xml 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. Realiza las siguientes consultas XPath:

- a) Obtén los datos de las cuentas bancadas cuyo tipo sea AHORRO.

```
sucursales//cuenta[@tipo="AHORRO"]
```

- b) Obtén 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"]))
```

- c) Obtén las cuentas de tipo PENSIONES de la sucursal con código SUC3.

```
sucursales/sucursal[@codigo="SUC3"]/cuenta[@tipo="PENSIONES"]
```

- d) Obtén por cada sucursal la concatenación de los datos, código sucursal, director y total

```
sucursales/sucursal/concat(@codigo,"_",director,"_",sum(cuenta/saldohaber),"€")
```

- saldo haber. e) Obtén todos los elementos de las sucursales con más de 3 cuentas.

```
sucursales/sucursal[count(cuenta)>3]
```

- f) Obtén todos los elementos de las sucursales con más de 3 cuentas del tipo AHORRO.

```
sucursales/sucursal[count(cuenta[@tipo="AHORRO"])>3]
```

- g) Obtén los nodos del director y la población de las sucursales con más de 3 cuentas.

```
sucursales/sucursal[count(cuenta)>3]/concat(director,"_",poblacion)
```

- h) Obtén el número de sucursales cuya población sea Madrid.

```
sucursales/count(sucursal[poblacion="Madrid"])
```

- i) Obtén 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),"€")
```

## Actividad 4.

. Tomando el documento XML `datospersonales.xml`, crea rutas de localización XPath que seleccionen los siguientes nodos o conjuntos de nodos:

a) Nombre de la persona.

`persona/nombre`

b) Año de nacimiento.

`persona/fechanacimiento/anio`

c) Lista de teléfonos.

`persona/telefono`

## Actividad 5.

A partir del documento XML `contactos.xml`, realiza las siguientes consultas XPath: a)

Lista de nombres de todos los contactos

`contactos//nombre`

b) Lista de años de nacimiento de los contactos nacidos después de 1970

`contactos/persona/fechanacimiento[number(anio)>1970]/anio`

c) Lista de nombres de los contactos nacidos entre 1950 y 1970 d) Nombres de las contactos/persona/fechanacimiento[number(anio)>1950 and number(anio)<1970]/../nombre personas que tengan exactamente 2 teléfonos.

`contactos/persona[count(telefono)=2]/nombre`

e) Lista de teléfonos móviles (que empiecen por 6) de los contactos

`contactos/persona/telefono[starts-with(., "6")]`

f) Promedio de edad de los contactos (toma la edad como la diferencia entre 2017 y el

año de nacimiento)

`avg(/contactos/persona/fechanacimiento/anio/number(2017 - .))` EL PUNTO MOLA

g) Dirección postal completa del primer contacto, con el siguiente formato: #calle#,

#numero#, #poblacion# (#pais#).

`contactos/persona/direccion/concat("#",calle,"#",numero,"#",poblacion,"#",pais)`