

### 9. Resuelve las siguientes consultas:

- a. Devuelve el apellido del penúltimo empleado (NOTA: utilizar last())
  - `//EMP_ROW[position()=last()-1]/APELLIDO`
- b. Obtén los elementos del empleado que ocupa la posición 3 (position())
  - `//EMP_ROW[position()=3]`
- c. Cuenta el número de empleados del departamento 10
  - `count(//EMP_ROW[DEPT_NO=10])`
- d. Obtén la suma de SALARIO de los empleados del DEPT\_NO =20
  - `sum(//EMP_ROW[DEPT_NO=20]/SALARIO)`
- e. Obtén el salario máximo, el mínimo de los empleados con OFICIO=ANALISTA
  - `concat(min(//EMP_ROW[OFICIO = "ANALISTA"]/SALARIO)," :  
",max(//EMP_ROW[OFICIO = "ANALISTA"]/SALARIO))`
- f. Obtén la media de salario en el DEPT\_NO=10
  - `avg(//EMP_ROW[DEPT_NO = 10]/SALARIO)`
- g. Devuelve la concatenación de apellido, oficio y salario
  - `concat(//EMP_ROW/APELLIDO, //EMP_ROW/SALARIO, //EMP_ROW/OFICIO)`
- h. Obtén los elementos de los empleados cuyo apellido empieza por 'A'
  - `//EMP_ROW[starts-with(APELLIDO,"A")]`
- i. Devuelve los oficios que contienen la sílaba 'OR'
  - `//EMP_ROW[contains(OFICIO,"OR")]/OFICIO`
- j. Obtén los datos de los empleados cuyo apellido tiene menos de 4 caracteres
  - `//EMP_ROW/APELLIDO[string-length()<4]`

### 10. Resuelve las siguientes consultas referentes al documento productos.xml. Este documento contiene los datos de los productos de una distribuidora de componentes informáticos. La estructura del documento es:

- a. Obtén la denominación y precio de todos los productos
  - `//produc/denominacion|//produc/precio`
- b. Obtén los productos que sean "Placa base"
  - `//produc[contains(denominacion,"Placa Base")]`
- c. Obtén los productos cuyo precio sea mayor que 60€ y de la zona 20
  - `//produc[precio>60 and cod_zona=20]`
- d. Obtén el número de los productos que sean memorias y de la zona 10
  - `count(//produc[contains(denominacion,"Memoria") and cod_zona=10])`
- e. Obtén la media de los precios de los micros
  - `avg(//produc[contains(denominacion,"Micro")]/precio)`
- f. Obtén los datos de los productos cuyo stock mínimo sea mayor que el stock actual (NOTA: usa función number())
  - `//produc[number(stock_minimo)>number(stock_actual)]`

g. Obtén el producto más caro

- `max(//produc/precio)`

h. Obtén el producto más barato de la zona 20

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