

a. /EMPLEADOS/EMP_ROW[DEPT_NO=10]

Toda la información de los empleados del departamento número 10

b. /EMPLEADOS/EMP_ROW/APELLIDO/EMPLEADOS/EMP_ROW/DEPT_NO

Apellidos y departamentos de los empleados

c. /EMPLEADOS/EMP_ROW [DEPT_NO=10]/APELLIDO/text()

solo el texto de los apellidos de los empleados del departamento 10

d. /EMPLEADOS/EMP_ROW [not(OFICIO="ANALISTA")]

Toda la información de los empleados que no son analistas

e. /EMPLEADOS/EMP_ROW[SALARIO>1300 and DEPT_NO=20]/APELLIDO

apellidos de los empleados con salario mayor a 1300 del departamento 20

f. /EMPLEADOS/EMP_ROW[1]

Toda la información del primer empleado

a. last()

//EMP_ROW[last()]/SALARIO

salario del ultimo empleado

b. position()

//EMP_ROW[position() = 2]

segundo empleado (posicion 2)

c. count()

count(//EMP_ROW)

cuantos empleados existen

d. sum(),div(),mod()

sum(//EMP_ROW[DEPT_NO=20]/SALARIO)

suma los salarios de los empleados del departamento 20

e. max(), min(),avg()

min(//EMP_ROW/SALARIO)

minimo salario

f. concat(cadena1, cadena2,...)

concat(//EMP_ROW[position()=2]/APELLIDO," y ", //EMP_ROW[position()=1]/APELLIDO)

concatena cadenas de texto, en este caso ARROYO y SANCHEZ.

g. starts-with (cadena1, cadena2)

los elementos cadena1 que empiezan por cadena2

//EMP_ROW[starts-with(apellido,"s")]

h. contains(cad1,cad2)

los elementos cadena1 que contengan la cadena2

//EMP_ROW[contains(apellido,"s")]

i. string-length(argumento)

empleado que su apellido sea de 5 letras

//EMP_ROW/apellido[string-length(5)]

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)]`

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)`