Las colecciones son una capeta

* **//** es el directorio **raíz**
* **/** **se entra en diferentes niveles de los xml**

La // son busquedas recursivas, diferencia de mayusculas y minusculas

Hay varios operadores:

* El operador **..** para el padre
* El operador **.** para el local

# Funciones unitarias

**La función data** se aplica mediante una consulta y se sacan todos los codigos

## La funcion number

pasa de tipo texto a tipo numérico

//departamento/empleado[nombre="Juan Parra"]/number(@salario)

## La funcion String

convierte un numero a cadena de texto

concat(string(333), 'hola')

## La función substring

substring(cadena,posicion)

substring(//produc[cod\_prod=1010]/denominacion,4) -> Un parámetro

substring(//EMP\_ROW[EMP\_NO='7369']/OFICIO,1,4) -> Dos parámetros

## UpperCase (Mayusculas)

upper-case(cadena)

Primer en mayúsculas y resto en minúsculas:

**EJEMPLOS**

concat(substring(//EMP\_ROW[EMP\_NO='7369']/APELLIDO,1,1),lower-case(substring(//EMP\_ROW[EMP\_NO='7369']/APELLIDO,2)))

//produc/upper-case(denominacion)

Visualizar la denominación de un producto junto con el precio multiplicado 1.21

//produc/concat(string(denominacion),precio\*1.21)

//produc/concat(denominacion,' ',number(precio\*1.21))

## LowerCase (Minusculas)

Contains permite devolver verdadero o falso

**EJEMPLOS**

//produc[contains(denominacion,'i5')]

//EMP\_ROW[contains(APELLIDO,'A')]

**Stars with**, es una función que devuelve true o false, si empieza por una determinada cadena de texto

Funciones de agrupacion, se utiliza para obtener

AVG, la media del precio //produc[precio>avg(//precio)]/concat('articulo: ',denominacion)

**Función count**

Une dos cadenas de texto

count(//produc[precio>avg(//precio)]/concat('Articulo: ',denominacion))

**Funcion Max min**

MAX Muestra el producto mas alto

max(//stock\_actual) **Stock actual máximo**

MIN Muestra el producto mas pequeño

Distint values Permite mostrar valores no repetidos dentro de un conjunto

distinct-values(//product/cod\_zona)

**La funcion sum()**

Suma todos los precios

- Mostrar la denominación de los productos que están bajo mínimos, es decir, su stock\_actual es inferior al mínimo.

//produc[number(stock\_actual)<number(stock\_minimo)]/concat("producto:",denominacion)

(: //product[(number(stock\_actual)-number(stock\_minimo))<0]/concat("producto:",denominacion)

Consulta cruzadas

Quien es el director de la oficina que tiene mas stock actual a la venta

//zona[cod\_zona=//produc[stock\_actual=max(//stock\_actual)]/cod\_zona]/director

**De la zona 10, quiero saber su nombre y la suma de todods los stock actual y minimo de cada zona**

concat(sum(//produc[cod\_zona=10]/stock\_actual),"-", sum(//produc[cod\_zona=10]/stock\_minimo),"-",//zona[cod\_zona=10]/nombre)

Mostrar el directorio de zona en donde se venden productos de tarjeta o del memoria

//zona[cod\_zona=distinct-values(//produc[contains(denominacion,'Memoria') || contains(denominacion,'Tarjeta') ]/cod\_zona)]/director

Denominación del producto, cuyos precios supera la media de los precios de la zona 20

//produc[precio>avg(//produc[cod\_zona=20]/precio)]/denominacion

XQUERY

Manipula y transforma los resultados de las consultas

for <variable> in <expresión XPath>

let <variables vinculadas>

where <condición XPath>

order by <expresión>

return <expresión de salida>

FOR

EJEMPLO

**for $productos in //produc**

**let $stock\_real:=($productos/stock\_actual)-($productos/stock\_minimo)**

**where $productos/cod\_zona=10**

**order by $productos/precio**

**return <stock\_real>{$stock\_real}</stock\_real>**

EJEMPLO

for $emp in /EMPLEADOS/EMP\_ROW

let $nom:=$emp/APELLIDO, $ofi:=$emp/OFICIO

order by $emp/OFICIO

return <APEOFI>{concat($nom,' ',$ofi)}</APEOFI>

Ejemplos de finalización en la consulta anterior para el return:

return <APEOFI>{concat($nom,' ',$ofi)}</APEOFI>

return $nom

En

for $dep in /universidad/departamento

return if ($dep/@tipo='A')

then <tipoA>{data($dep/nombre)}</tipoA>

else () aunque no devuelva nada, hay que poner else

**for $productos in //produc**

**let $stock\_real:=($productos/stock\_actual)-($productos/stock\_minimo)**

(:where $productos/cod\_zona=10:)

order by $productos/precio

return if ($productos/cod\_zona=10)

then <stock\_real>{$stock\_real}</stock\_real

Ifs anidados for $dep in /universidad/departamento

return if ($dep/@tipo='A' )

then <tipoA>{data($dep/nombre)}</tipoA>

else if ($dep/@tipo='B')

then <tipoB>{data($dep/nombre)}</tipoB>

else ()

Mostrar los nombres de profesores del departamento A

**for $prof in /universidad/departamento[@tipo='A']/empleado**

**let $profe:=$prof/nombre, $puesto:=$prof/puesto**

**(:where $puesto='Profesor'**

**return $profe:)**

**return if ($puesto='Profesor')**

**then $profe**

Mostrar el nombre de los departamentos que son del tipo A y tipo B en dos nodos diferenciados: <tipoA> <tipoB>

**for $dep in /universidad/departamento**

**return if ($dep/@tipo='A')**

**then <tipoA>{data($dep/nombre)}</tipoA>**

**else <tipoB>{data($dep/nombre)}</tipoB>**

**Mostrar el nombre de departamento y número de empleados que tiene por cada uno de ellos con el siguiente formato:**

**<depart>Informática</depart>**

**<emple>2</emple>**

**<depart>Matemáticas</depart>**

**<emple>4</emple>**

**…**

for $dep in /universidad/departamento

let $nom:=$dep/empleado

return <depart>{data($dep/nombre)} </depart> |

<emple>{count($nom)}</emple>

Otra opción es

for $dept in /universidad/departamento

let $nom:=$dept/nombre, $num:=count($dept/empleado)

return (<depart>{data($nom)}</depart>, <emple>{data($num)}</emple>)