**Unidad 5. XPATH-XQUERY**

**1.Introducción**

- / → Indicar la estructura jerárquica de los nodos

- // → Busca recursiva

- .. → Selecciona el nodo padre

- . → Selecciona el nodo actual

- [] → Filtra datos

- Operadores: =,>,<,!=,>=,<=, and, or o || → Se usan para filtrar los datos dentro de los corchetes

- @ → Selecciona atributos para filtrar en las selecciones

- (: Comentario :) → Para comentar el código

**2.Funciones**

- data → Recoge unicamente los datos de las etiquetas

- number → Convierte el valor al número

- string → Convierte el valor en un String

- concat → Concatena valores Strings

- substring → Coge una subcadena de la cadena indicada indicando el inicio y el número de caracteres que se quieran coger

- upper-case → Convierte el texto a mayúscula

- lower-case → Convierte el texto a minúscula

- contains → Devuelve un boolean, busca el dato indicado dentro del nodo indicado

- starts-with → Devuelve un boolean, busca el dato indicado dentro del nodo indicado comenzando por el principio

**3.Funciones de agrupación**

- count → Cuenta los elementos que hay en el XML

- sum → Suma los valores de ese tipo

- avg → Hace la media del valor elegido

- max → El valor máximo del elementos

- min → El valor mínimo del elementos

- distinct-values → Elimina valores repetidos

**5.XQuery**

- for <variable> in <expresión XPath> (recuperar una vuelta por cada valor repetido en una linea)

- let <variables vinculadas> (asignamos a una valiable un valor)

- where <condición Xpath> (filtro)

- order by <expresión> (ordena los resultados)

- return <expresión de salida> (muestra cada vuelta del bucle 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 $productos in //produc[cod\_zona = 10]

let $subida\_precio:=$productos/precio \* 1.10

where $productos/precio > 45

order by $productos/cod\_prod

return concat($productos/cod\_prod,' - ',$productos/precio,' - ', $subida\_precio)

**6. Sacar datos de dos XML distintos**

Ejermplo:

for $emp in (/EMPLEADOS/EMP\_ROW)

let $emple:= $emp/APELLIDO, $dep:= $emp/DEPT\_NO, $dnom:= (/departamentos/DEP\_ROW[DEPT\_NO =$dep]/DNOMBRE)

return <res>{$emple, $dep, $dnom} </res>

Ejercicio:

for $emp in //EMP\_ROW

let $MaxEmple := max(//EMP\_ROW/SALARIO)

let $Localidad:= //DEP\_ROW[DEPT\_NO=$emp/DEPT\_NO]/LOC

return if ($emp/SALARIO=$MaxEmple)

{data($Localidad)}

**7. Doble for**

Ejemplo: for $dep in /universidad/departamento

for $pue in distinct-values($dep/empleado/puesto)

let $cu:=count($dep/empleado[puesto=$pue])

return

<departamentos><depart>{data($dep/nombre)}</depart><puesto>{data($pue)}</ puesto><profes>{$cu}</profes></departamentos>

<HTML>

<BODY>

<table>

{

for $dept in (//DEP\_ROW)

let $nodept:=$dept/DEPT\_NO , $nomdep:=$dept/DNOMBRE

for $emple in (//EMP\_ROW[DEPT\_NO=$nodept])

let $empleado:=$emple/APELLIDO

return <tr><td>{data($nodept)}</td><td>{data

($nomdep)}</td><td>{data($empleado)}</td></tr>

}

</table>

</BODY>

</HTML>

Ejercicio: for $dept in //DEP\_ROW

for $max in max(//EMP\_ROW[DEPT\_NO=$dept/DEPT\_NO]/SALARIO)

let $emp:=data(//EMP\_ROW[SALARIO=$max and DEPT\_NO=$dept/DEPT\_NO]/APELLIDO)

return

<departamento>

{$dept}

<empleadoMax.sal>{$emp}</empleadoMax.sal>

</departamento>

for $puestos in distinct-values(//OFICIO)

for $maxSalario in max(//EMP\_ROW[OFICIO = $puestos]/SALARIO)

let $empleado:= //EMP\_ROW[SALARIO = $maxSalario and OFICIO = $puestos]

return $empleado/APELLIDO