

## UF3. Persistencia en BD nativas XML.

### Actividades PAC01 (Solución):

Para poder realizar esta PAC deberéis instalar el gestor de bases de datos XML eXist tal como se os explica en el libro. Además los ficheros de datos con los que trabajaréis los encontraréis en la página: [http://www.elaltozano.es/ACCESO\\_A\\_DATOS/](http://www.elaltozano.es/ACCESO_A_DATOS/) buscando los del tema 5 del libro. Ahí también está una versión de eXist, aunque igualmente os podéis descargar una versión más actualizada en [exist-db.org](http://exist-db.org)

**Importante:** eXist usa como puerto para gestionar la base de datos el 8080, el cual puede que esté ocupado por alguna otra aplicación. Si es el caso, no arrancará y os dará un error. Para solucionarlo deberéis cambiar el puerto estándar de eXist modificando el fichero `/tools/jetty/etc/jetty.xml` dentro del directorio de eXist.

#### 1. Realiza la actividad 1 de la página 231 del libro.

- Obtén los nodos denominación y precio de todos los productos:  
`/productos/produc/denominacion | /productos/produc/precio`
- Obtén los nodos de los productos que sean placas base:  
`/productos/produc[starts-with(denominacion,'Placa Base')]`
- Obtén los nodos de los productos con precio mayor de 60 € y de la zona 20:  
`/productos/produc[precio>60 and cod_zona=20]`
- Obtén el número de los productos que sean memorias y de la zona 10:  
`count(/productos/produc[starts-with(denominacion,'Memoria') and cod_zona=10])`
- Obtén la media de precio de los micros:  
`avg(/productos/produc[starts-with(denominacion,'Micro')]/precio)`
- Obtén los datos de los productos cuyo stock mínimo sea mayor que su stock actual:  
`/productos/produc[number(stock_minimo)>number(stock_actual)]`
- 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
cod_zona=40]/denominacion |
/productos/produc[number(stock_minimo)>number(stock_actual) and
cod_zona=40]/precio
```

- Obtén el producto más caro:  
`/productos/produc[precio=max(/productos/produc/precio)]`
- Obtén el producto más barato de la zona 20:  
`/productos/produc[precio=min(/productos/produc[cod_zona="20"]/precio) and cod_zona="20"]`
- Obtén el producto más caro de la zona 10:  
`/productos/produc[precio=max(/productos/produc[cod_zona="10"]/precio) and cod_zona="10"]`

2. Realiza la actividad 3 de la página 242 del libro.

- Obtén por cada zona el número de productos que tiene:  
`for $produc in distinct-values(/productos/produc/cod_zona)
let $cu:=count(/productos/produc[cod_zona=$produc])
return concat('Hay:', $cu , 'productos', 'en la zona nº',$produc)`
- Obtén la denominación de los productos entre las etiquetas 10 si son del código 10, 20 si son de la zona 20, 30 si son de la zona 30, 40 si son de la zona 40:  
`for $prod in //produc
return
concat('<zona',$prod/cod_zona,'>',$prod/denominacion,'</zona',$prod/cod_zona,'>')`
- Obtén por cada zona la denominación de los productos más caros:  
`for $prod in //produc for $zon in $prod/cod_zona
let $den:=$prod/denominacion let $precio:=$prod/precio
return if ($precio=max(/productos/produc[cod_zona=$zon]/precio) and
/productos/produc/cod_zona=$zon)
then
< zona>{data($zon)}<preciomax>{data($precio)}</preciomax><producto>{data (
$den)}</producto></zona>
else()`
- Obtén la denominación de los productos contenida entre las etiquetas placa para los productos en cuya denominación aparece la palabra Placa Base, <memoria> para los que contienen la palabra memoria, <micro> para los que contienen la palabra Micro y <otros> para el resto de productos:

```
for $prod in (/productos/produc/denominacion)
  let $deno:=data(/productos/produc[contains (denominacion,
$prod)]/denominacion)
return if (contains($prod, 'Placa Base')) then <placa>{$deno}</placa> else(
if (contains($prod, 'Memoria')) then <memoria>{$deno}</memoria> else(
if (contains($prod, 'Micro')) then <micro>{$deno}</micro> else()))
```

- Devuelve el código de sucursal y el número de cuentas que tiene de tipo ahorro y de tipo pensiones.

```
for $sucursal in /sucursales/sucursal
  let $numeroAhorro := count($sucursal/cuenta[@tipo = 'AHORRO'])
  let $numeroPensiones := count($sucursal/cuenta[@tipo = 'PENSIONES'])
return <Sucursal> <Codigo>{data($sucursal/@codigo)}</Codigo>
<Ahorro>{$numeroAhorro}</Ahorro>
<Pensiones>{data($numeroPensiones)}</Pensiones> </Sucursal>
```

- Devuelve por cada sucursal el código de sucursal, el director, la población, la suma del total debe y la suma de total haber de sus cuentas:

```
for $suc in distinct-values(/sucursales/sucursal/@codigo)
  let $codigo:=//sucursales/sucursal[@codigo=$suc]/@codigo
  let $director:=//sucursales/sucursal[@codigo=$suc]/director
  let $poblac:=//sucursales/sucursal[@codigo=$suc]/poblacion
  let $haber:=sum(/sucursales/sucursal[@codigo=$suc]/cuenta/saldohaber)
  let $debe:=sum(/sucursales/sucursal[@codigo=$suc]/cuenta/saldodebe)
return
<sucursal>
  <codigo>{$suc}</codigo>
  {$director}
  {$poblac}
  <suma_debe>{$debe}</suma_debe>
  <suma_haber>{$haber}</suma_haber>
</sucursal>
```

- Devuelve el nombre de los directores, el código de la sucursal, y la población de las sucursales con más de tres cuentas.

```
for $sucursal in /sucursales/sucursal
where (count($sucursal/cuenta) > 3)
return <Sucursal>{($sucursal/@codigo)} {($sucursal/director)}
{($sucursal/poblacion)} </Sucursal>
```

- Devuelve por cada sucursal, el código de sucursal y los datos de las cuentas con más saldo debe

```
for $suc in distinct-values(/sucursales/sucursal/@codigo)
  let $codigo:=//sucursales/sucursal[@codigo=$suc]/@codigo
  let
cuenta:=//sucursales/sucursal[@codigo=$suc]/cuenta[number(saldodebe)=number
(max(/sucursales/sucursal[@codigo=$suc]/cuenta/saldodebe))]
```

```
return
<sucursal>
  <codigo>{$suc}</codigo>
  {$cuenta}
</sucursal>
```

- Devuelve la cuenta del tipo pensiones que ha hecho más aportación:  

```
for $suc in distinct-values(//sucursales/sucursal/@codigo)
  let
    $cuenta:=/sucursales/sucursal[@codigo=$suc]/cuenta[number(aportacion)=number(max(//sucursales/sucursal[@codigo=$suc]/cuenta/aportacion))]
  return
  <sucursal>
    <codigo>{$suc}</codigo>
    {$cuenta}
  </sucursal>
```

**Nota:** Hay distintas posibles maneras de realizar un mismo ejercicio por lo que puede que existan soluciones correctas distintas de las aquí plasmadas.