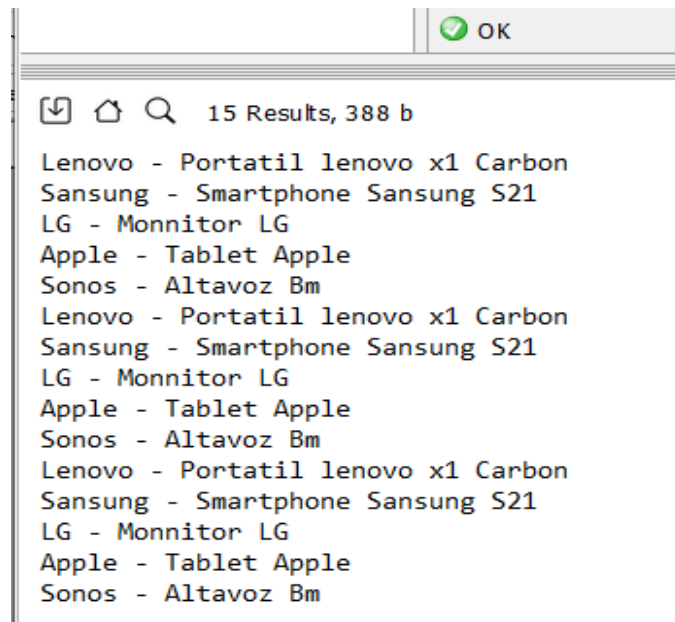


### Consulta 1: Listar todos los productos y sus fabricantes.

For \$producto in //producto return concat(\$producto/fabricante/text(), ' - ', \$producto/nombreproducto/text())



### Consulta 2: Listar todos los productos con un precio mayor a \$100.

for \$producto in //producto where \$producto/precio >=100 return concat(\$producto/nombreproducto/text(), ' - ', \$producto/precio/text(), ' €')



### Consulta 3: Listar todos los productos con un precio menor o igual a \$50

for \$producto in //producto where \$producto/precio <=100 return concat(\$producto/nombreproducto/text(), ' - ', \$producto/precio/text(), ' €')

XQuery `for $producto in //producto where $producto/precio <=100 return concat($producto/nombreproducto/text(), ' - ', $producto/precio/text(), ' €')`

Context: db:get("phoneland... Editor

1 <productos>  
 2 <producto>  
 3 <nombreproducto>Portatil lenovo x1 Carbon</nombreproducto>  
 4 <fabricante>Lenovo</fabricante>  
 5 <precio>1200</precio>  
 6 </producto>  
 7  
 8 <producto>  
 9 <nombreproducto>Smartphone Samsung Galaxy S21</nombreproducto>  
 10 <fabricante>Samsung</fabricante>  
 11 <precio>900</precio>  
 12 </producto>  
 13  
 14 </productos>

productos.xml (1937 b)

producto	precio
Portatil lenovo x1 Carbon	1200
Smartphone Samsung Galaxy S21	900

3 Results, 67 b

Monnitor LG - 100 €  
 Monnitor LG - 100 €  
 Monnitor LG - 100 €

Compiling:  
 - merge: descendant::producto  
 - rewrite <= comparison: (\$producto\_0/precio <= 100) -> \$producto\_0/precio <= 100

#### Consulta 4: Listar los nombres de los productos que fabrica "Fabricante A"

XQuery `for $name in //producto where $name/fabricante='Lenovo' return $name/nombreproducto`

Context: db:get("phoneland... Editor

1 <productos>  
 2 <producto>  
 3 <nombreproducto>Portatil lenovo x1 Carbon</nombreproducto>  
 4 <fabricante>Lenovo</fabricante>  
 5 <precio>1200</precio>  
 6 </producto>  
 7  
 8 <producto>  
 9 <nombreproducto>Smartphone Samsung Galaxy S21</nombreproducto>  
 10 <fabricante>Samsung</fabricante>  
 11 <precio>900</precio>  
 12 </producto>  
 13  
 14 </productos>

productos.xml (1937 b)

3 Results, 178 b

<nombreproducto>Portatil lenovo x1 Carbon</nombreproducto>  
 <nombreproducto>Portatil lenovo x1 Carbon</nombreproducto>  
 <nombreproducto>Portatil lenovo x1 Carbon</nombreproducto>

Compiling:  
 - merge: descendant  
 - rewrite to predicate  
 - inline for \$name\_0

## 5.- Listar los nombres de los productos que tienen un precio mayor a \$100 y que son fabricados por "Fabricante B"

XQuery `for $price in //producto[precio>=100 and fabricante="Samsung"] return $price/nombreproducto`

Context: db:get("phonek")

3 Results, 169 b

Result

Compiling:

- rewrite >= comparison
- rewrite to predicate: (p
- merge: descendant::pr

## 6.- Listar los nombres de los productos y sus precios, ordenados por precio ascendente.

XQuery `for $product in //producto where $product/precio let $price := $product/precio order by $price ascending return concat("Producto:", $product/nombreproducto, ", Precio: ", $price)`

Context: db:get("phonek")

15 Results, 619 b

Result

Info

Compiling:

- merge: descendant::producto
- rewrite fn:concat(value1,value2[,...]): concat("Producto:", \$product\_0/nombreproducto, ", Precio: ", \$price\_1) -> ("Producto:" || \$product\_0/nombreproducto || ", Precio: " || \$price\_1)
- rewrite to predicate: ./precio

Optimizing:

- rewrite context value: . -> db:get-pre("phoneland", 0)
- rewrite util:root(nodes): util:root(db:get-pre("phoneland", 0)) -> db:get-pre("phoneland", 0)
- convert to child steps: descendant::producto[precio]

Optimized Query:

```
(for $product_0 in db:get-pre("phoneland", 0)/*-productos/*-producto[precio]
let $price_1 := $product_0/precio order by $price_1 empty least return ("
```

## Consulta 7: Listar los nombres de los productos y sus precios, ordenados por precio descendente.

The screenshot shows the XQuery Studio interface. The query editor contains the following XQuery:

```
for $product in //producto where $product/precio let $price := $product/precio order by $price descending return concat("Producto:", $product/nombreproducto, " ", Precio: ", $price)
```

The left pane shows the file explorer with the 'phoneland' directory selected. The right pane shows a preview of the XML data, displaying product names and prices.

The bottom pane shows the results of the query, which are 15 results totaling 619 bytes. The results are displayed in a table format:

Producto	Precio
Smartphone Samsung S21	900
Smartphone Samsung S21	900
Smartphone Samsung S21	900
Altavoz Bm	400
Altavoz Bm	400
Altavoz Bm	400
Portatil lenovo x1 Carbon	1200
Portatil lenovo x1 Carbon	1200
Portatil lenovo x1 Carbon	1200
Tablet Apple	1100
Tablet Apple	1100
Tablet Apple	1100
Monitor LG	100
Monitor LG	100
Monitor LG	100

The bottom right pane shows the 'Compiling' and 'Optimizing' information for the query.

## Consulta 8: Listar los nombres de los productos que contienen la palabra "Cámara".

The screenshot shows the XQuery Studio interface. The query editor contains the following XQuery:

```
for $product in //producto where contains($product/nombreproducto, "Tablet") return $product/nombreproducto
```

The left pane shows the file explorer with the 'phoneland' directory selected. The right pane shows a preview of the XML data, displaying product names and prices.

The bottom pane shows the results of the query, which are 3 results totaling 139 bytes. The results are displayed in a table format:

nombreproducto
Tablet Apple
Tablet Apple
Tablet Apple

The bottom right pane shows the 'Compiling' and 'Optimizing' information for the query.

### Consulta 9: Listar los nombres de los productos que contienen la letra "E" en su nombre.

Query: `for $find in //producto where contains($find/nombreproducto, "e")return $find/nombreproducto`

Context: db:get(C:/baseX/baseX/data/phoneland/)

Find contents...: \*.xml, \*.xq\*

phoneland

- atv.baseX (0 b)
- atvl.baseX (4 b)
- atvr.baseX (0 b)
- inf.baseX (1154 b)
- productos.xml (1937 b)
- tbl.baseX (4096 b)
- tbli.baseX (6 b)
- txt.baseX (287 b)
- txtl.baseX (143 b)
- txtr.baseX (90 b)

productos.xml

```

41 <fabricante>Samsung</fabricante>
42 <precio>900</precio>
43 </producto>
44
45 <producto>
46 <nombreproducto>Monnitor LG</nombreproducto>
47 <fabricante>LG</fabricante>
48 <precio>100</precio>
49 </producto>
50
51 <producto>
52 <nombreproducto>Tablet Apple</nombreproducto>
53 <fabricante>Apple</fabricante>
54 <precio>1100</precio>
55 </producto>
56
57 <producto>
58 <nombreproducto>Altavoz Bm</nombreproducto>
59 <fabricante>Sonos</fabricante>

```

OK

9 Results, 490 b

Result

```

<nombreproducto>Portatil lenovo x1 Carbon</nombreproducto>
<nombreproducto>Smartphone Samsung S21</nombreproducto>
<nombreproducto>Tablet Apple</nombreproducto>
<nombreproducto>Portatil lenovo x1 Carbon</nombreproducto>
<nombreproducto>Smartphone Samsung S21</nombreproducto>
<nombreproducto>Tablet Apple</nombreproducto>
<nombreproducto>Portatil lenovo x1 Carbon</nombreproducto>
<nombreproducto>Smartphone Samsung S21</nombreproducto>
<nombreproducto>Tablet Apple</nombreproducto>

```

Compi

- mergi
- rewrit
- inline
- simpl
- e"))/noi

Optimi

- rewrit
- rewrit

### Consulta 10: Listar los nombres de los productos cuyo fabricante comienza con la letra "S"

New... (Ctrl+N)

Query: `for $find in //producto where contains($find/nombreproducto, "S")return $find/nombreproducto`

Context: db:get(C:/baseX/baseX/data/phoneland/)

Find contents...: \*.xml, \*.xq\*

phoneland

- atv.baseX (0 b)
- atvl.baseX (4 b)
- atvr.baseX (0 b)
- inf.baseX (1154 b)
- productos.xml (1937 b)
- tbl.baseX (4096 b)
- tbli.baseX (6 b)
- txt.baseX (287 b)
- txtl.baseX (143 b)
- txtr.baseX (90 b)

productos.xml

```

41 <fabricante>Samsung</fabricante>
42 <precio>900</precio>
43 </producto>
44
45 <producto>
46 <nombreproducto>Monnitor LG</nombreproducto>
47 <fabricante>LG</fabricante>
48 <precio>100</precio>
49 </producto>
50
51 <producto>
52 <nombreproducto>Tablet Apple</nombreproducto>
53 <fabricante>Apple</fabricante>
54 <precio>1100</precio>
55 </producto>
56
57 <producto>
58 <nombreproducto>Altavoz Bm</nombreproducto>
59 <fabricante>Sonos</fabricante>

```

OK

3 Results, 169 b

Result

```

<nombreproducto>Smartphone Samsung S21</nombreproducto>
<nombreproducto>Smartphone Samsung S21</nombreproducto>
<nombreproducto>Smartphone Samsung S21</nombreproducto>

```

Compi

- mergi
- rewrit
- inline