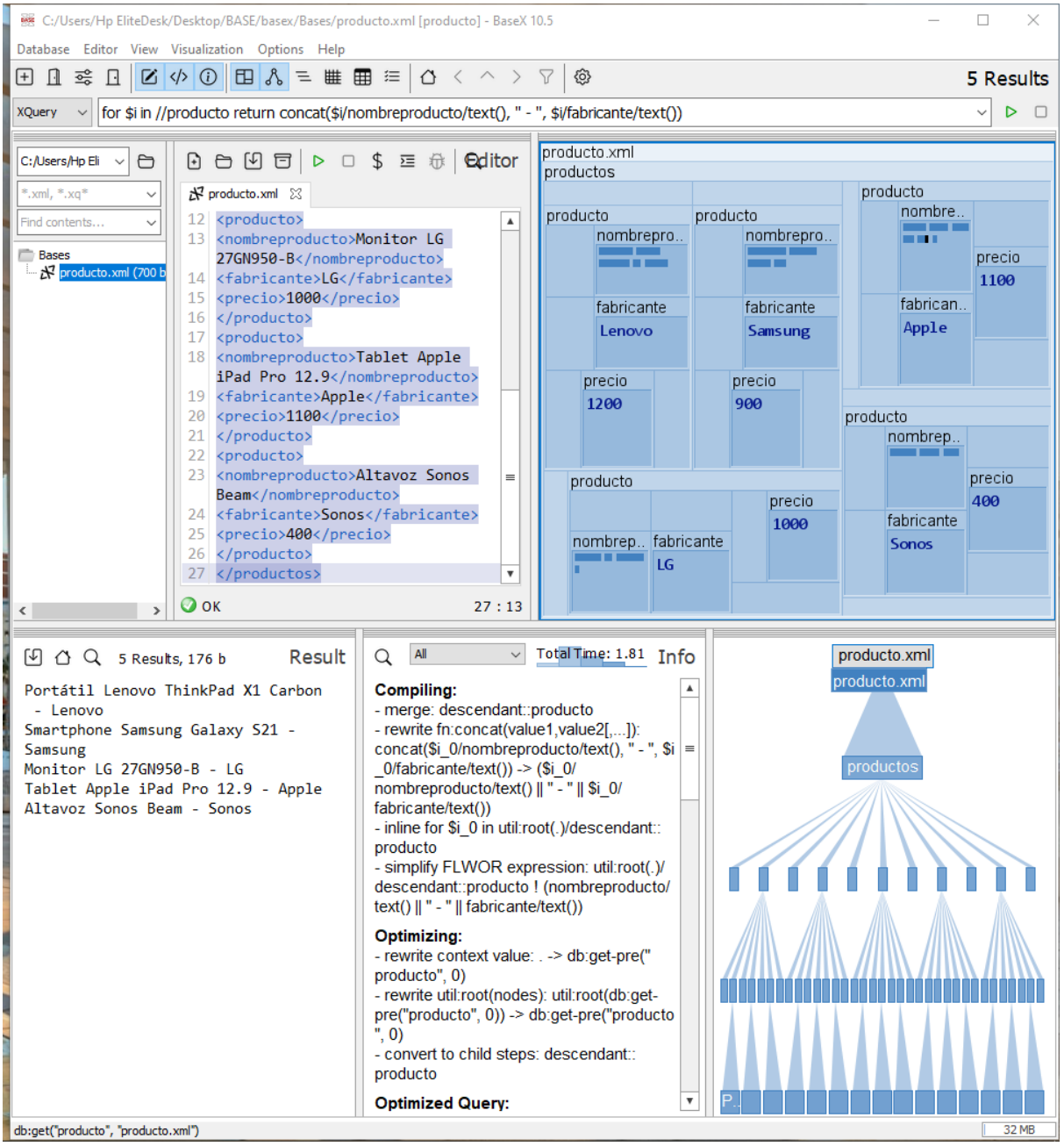
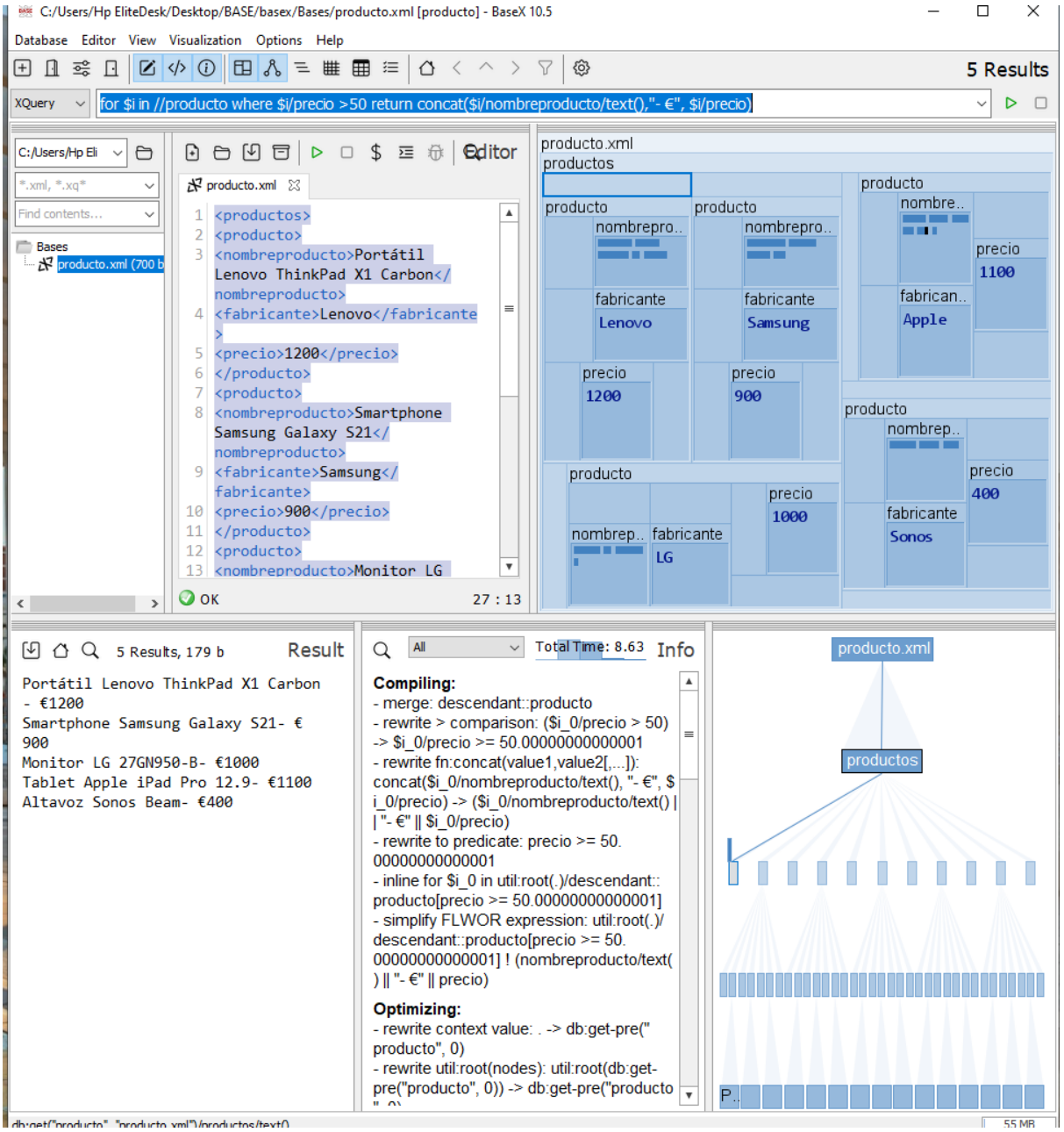
Listar todos los productos y sus fabricantes.

for $i in //producto return concat($i/nombreproducto/text(), " - ", $i/fabricante/text())



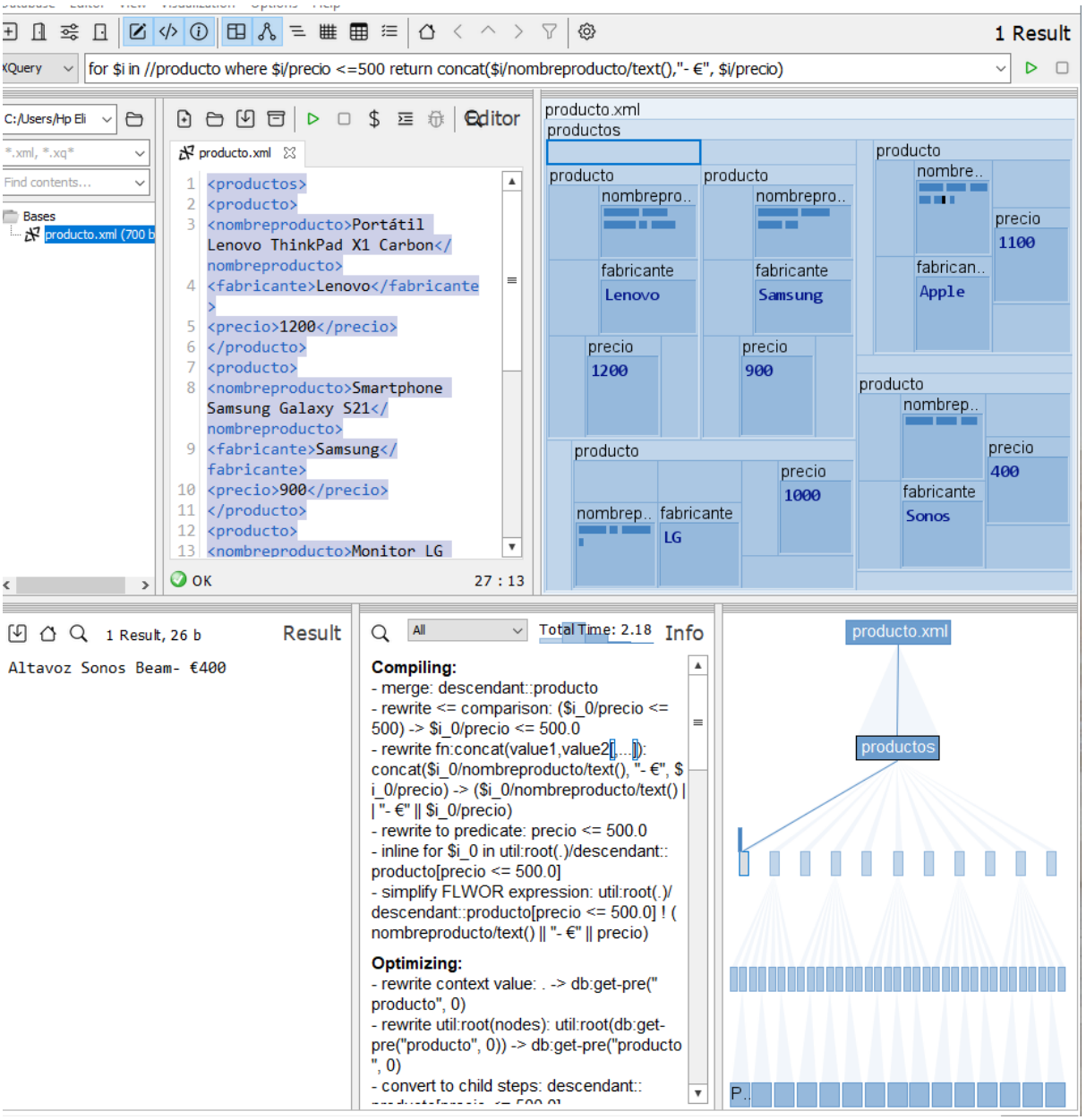
Listar todos los productos con un precio mayor a $500.

for $i in //producto where $i/precio >500 return concat($i/nombreproducto/text(),"- €", $i/precio)



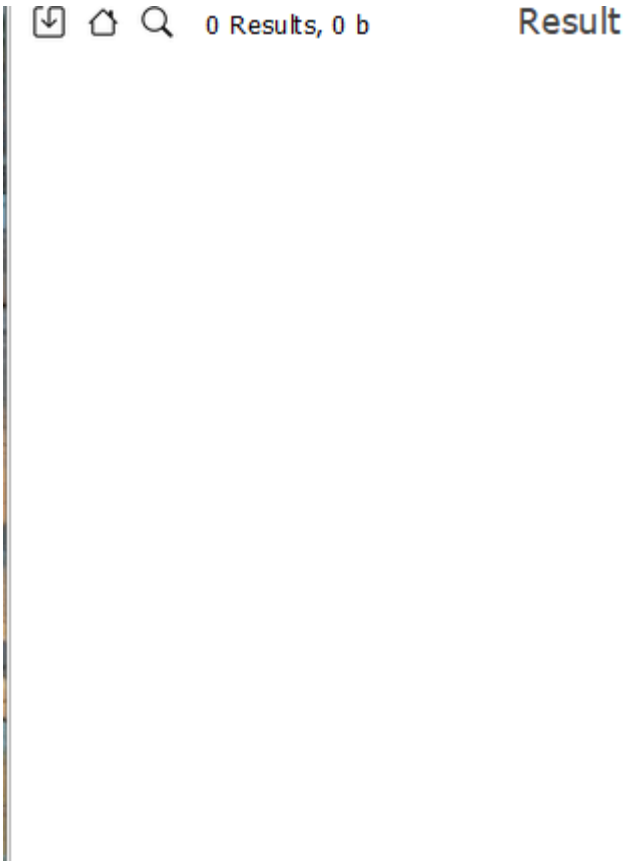
Listar todos los productos con un precio menor o igual a $500.

for $i in //producto where $i/precio <=500 return concat($i/nombreproducto/text(),"- €", $i/precio)



Listar los nombres de los productos que fabrica "Fabricante A".

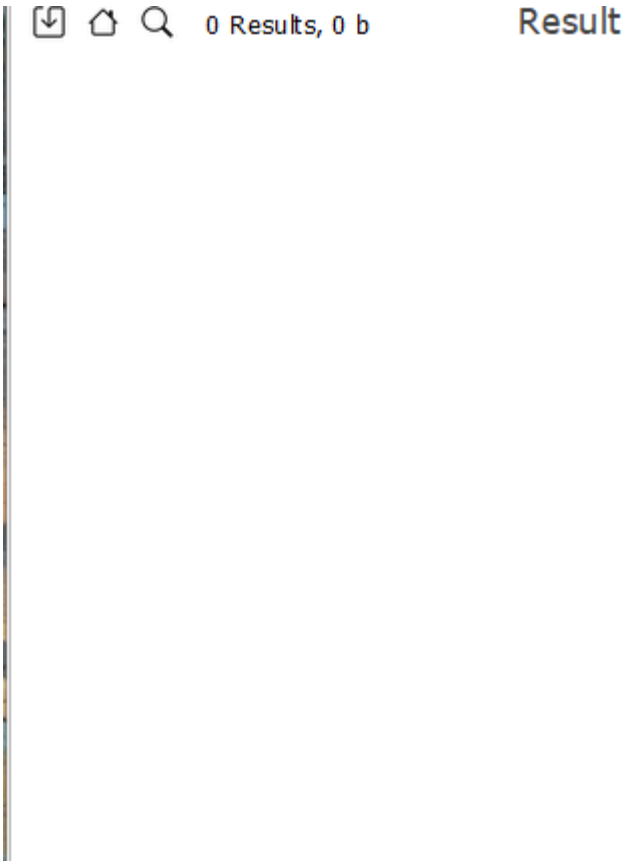
Está vacía la lista



for $i in //producto where $i/fabricante="FABRICANTE A" return $i/nombreproducto/text()

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

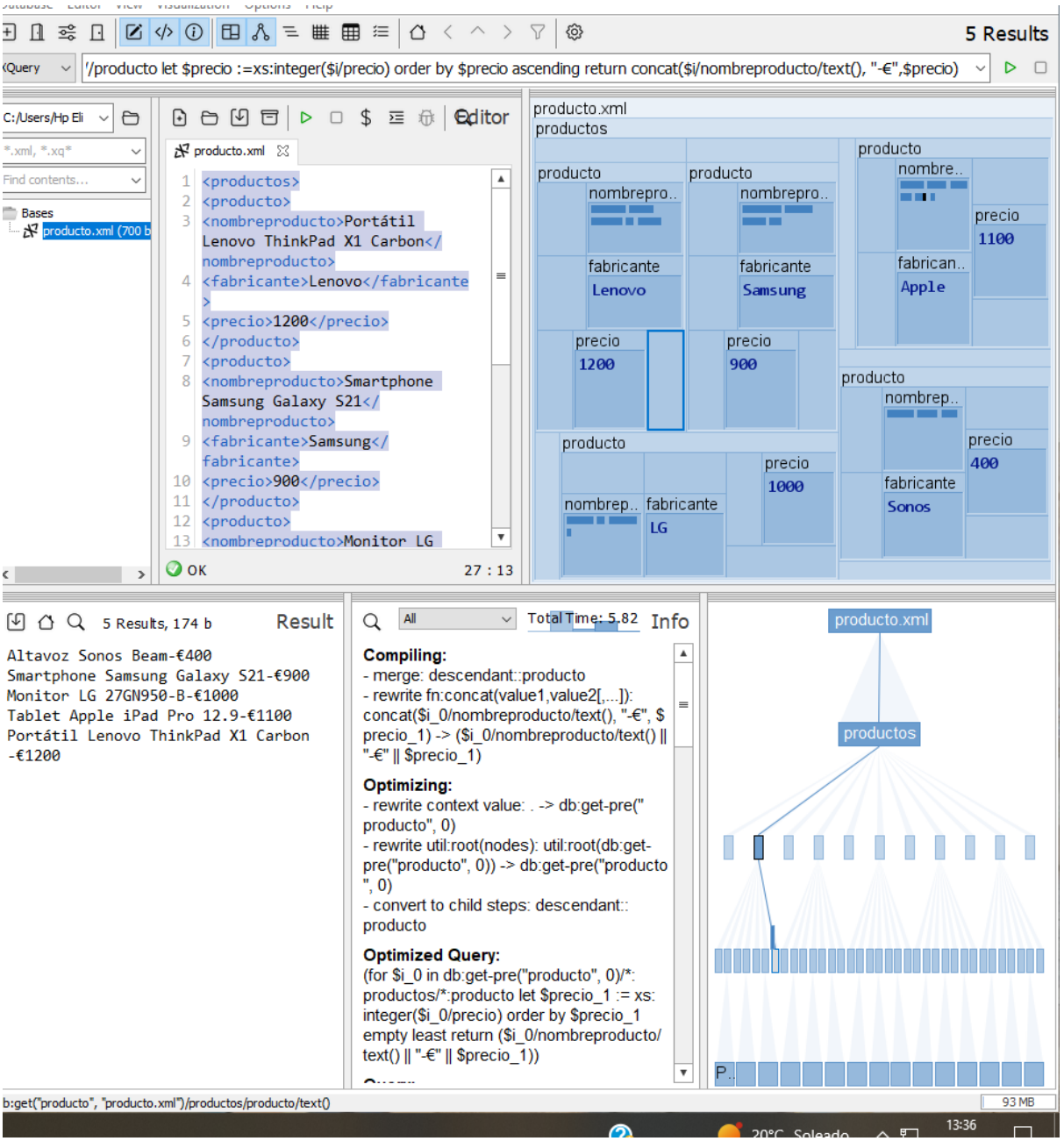
Está vacía la lista




```
for $i in //producto where $i/fabricante = "Fabricante B" and $i/precio > 100 return
$i/nombreproducto/text()
```

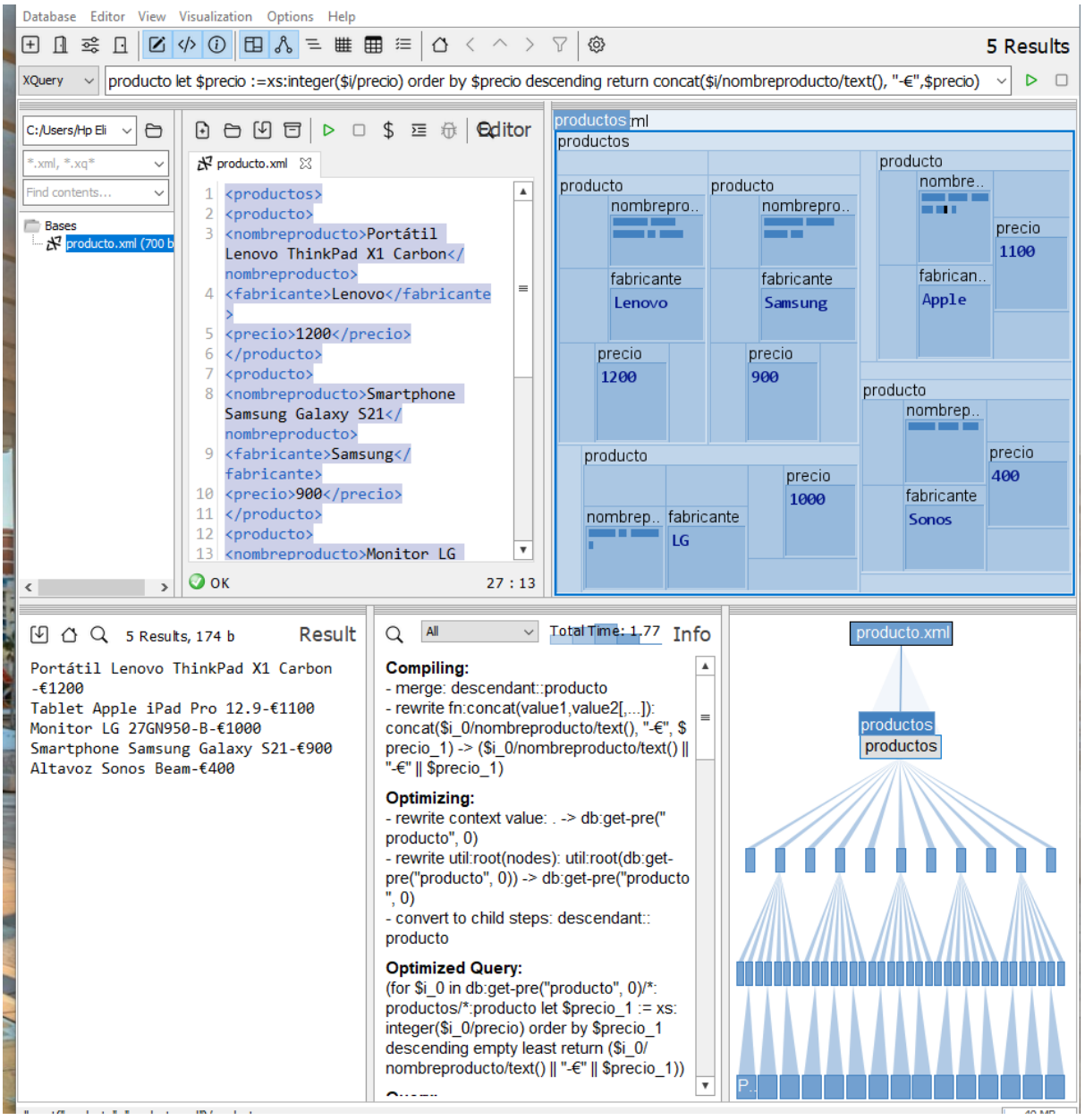
Listar los nombres de los productos y sus precios, ordenados por precio ascendente.

```
for $i in //producto let $precio :=xs:integer($i/precio) order by $precio ascending return
concat($i/nombreproducto/text(), "-€",$precio)
```


5 Results
/producto let $precio :=xs:integer($i/precio) order by $precio ascending return concat($i/nombreproducto/text(), "-€",$precio)
producto.xml
1 <productos>
2 <producto>
3 <nombreproducto>Portátil Lenovo ThinkPad X1 Carbon</nombreproducto>
4 <fabricante>Lenovo</fabricante>
5 <precio>1200</precio>
6 </producto>
7 <producto>
8 <nombreproducto>Smartphone Samsung Galaxy S21</nombreproducto>
9 <fabricante>Samsung</fabricante>
10 <precio>900</precio>
11 </producto>
12 <producto>
13 <nombreproducto>Monitor LG
OK
27 : 13
productos
producto
nombrepro..
fabricante
Lenovo
precio
1200
Samsung
900
nombre..
Apple
1100
nombrep..
Sonos
400
LG
1000
5 Results, 174 b
Result
Altavoz Sonos Beam-€400
Smartphone Samsung Galaxy S21-€900
Monitor LG 27GN950-B-€1000
Tablet Apple iPad Pro 12.9-€1100
Portátil Lenovo ThinkPad X1 Carbon -€1200
All
Total Time: 5.82
Info
Compiling:
- merge: descendant::producto
- rewrite fn:concat(value1,value2[,...]): concat($i_0/nombreproducto/text(), "-€", $precio_1) -> ($i_0/nombreproducto/text() || "-€" || $precio_1)
Optimizing:
- rewrite context value: . -> db:get-pre("producto", 0)
- rewrite util:root(nodes): util:root(db:get-pre("producto", 0)) -> db:get-pre("producto", 0)
- convert to child steps: descendant::producto
Optimized Query:
(for $i_0 in db:get-pre("producto", 0)/*:productos/*:producto let $precio_1 := xs:integer($i_0/precio) order by $precio_1 empty least return ($i_0/nombreproducto/text() || "-€" || $precio_1))
producto.xml
productos
P..
b:get("producto", "producto.xml")/productos/producto/text()
93 MB
13:36

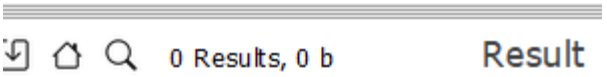
Listar los nombres de los productos y sus precios, ordenados por precio descendente.

for $i in //producto let $precio :=xs:integer($i/precio) order by $precio descending return concat($i/nombreproducto/text(), "-€",$precio)



Listar los nombres de los productos que contienen la palabra "Cámara"

for $p in //producto where contains($p/nombreproducto/text(), "Cámara") return $p/nombreproducto/text()



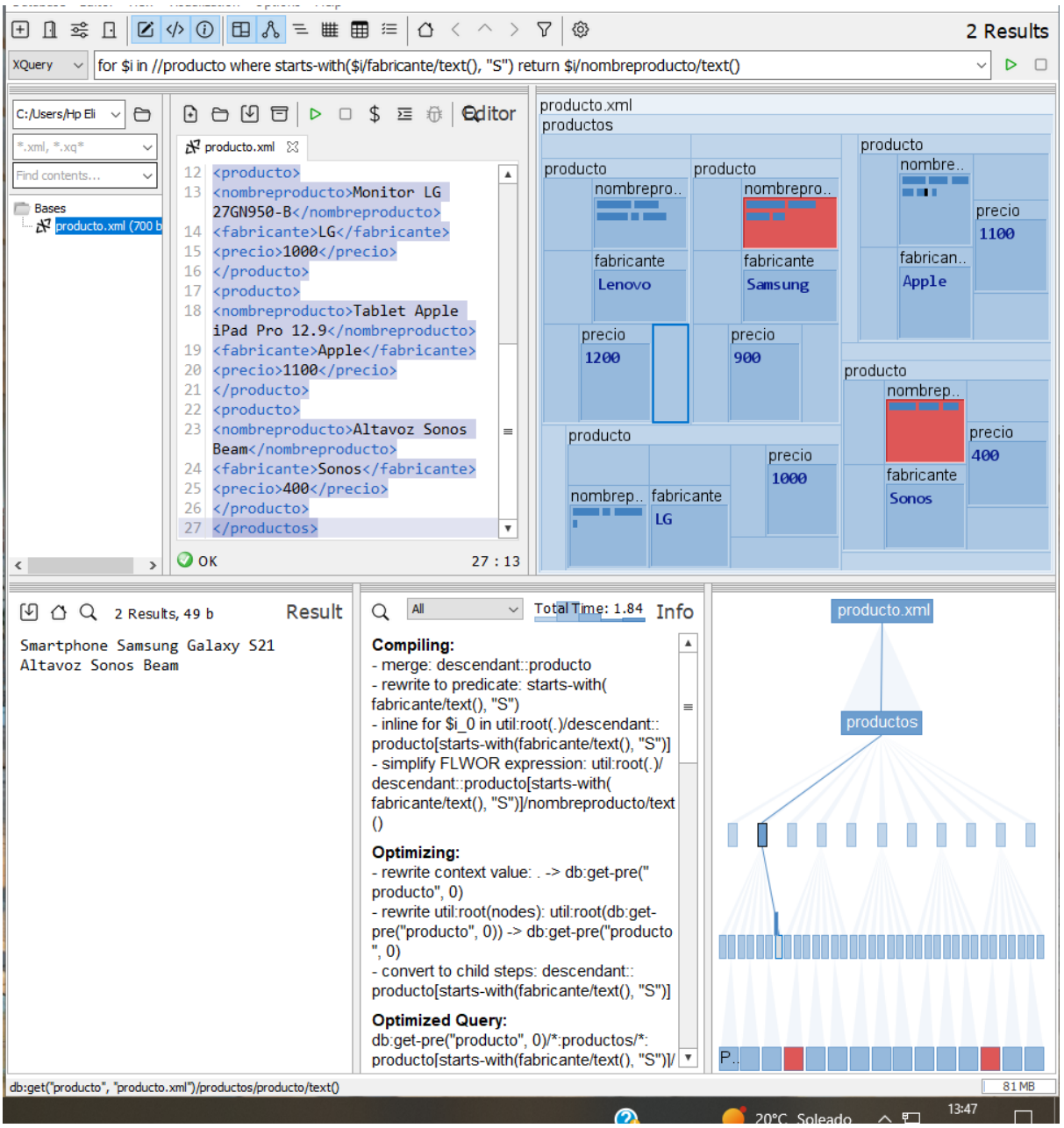
Listar los nombres de los productos que contienen la letra "E" en su nombre.

for $i in //producto where contains($i/nombreproducto/text(), "E") return $i/nombreproducto/text()

//no sale nada

Listar los nombres de los productos cuyo fabricante comienza con la letra "S"

for $i in //producto where starts-with($i/fabricante/text(), "S") return $i/nombreproducto/text()


2 Results
XQuery
for $i in //producto where starts-with($i/fabricante/text(), "S") return $i/nombreproducto/text()
C:/Users/Hp Eli
*.xml, *.xq*
Find contents...
Bases
producto.xml (700 b
Editor
producto.xml
12 <producto>
13 <nombreproducto>Monitor LG 27GN950-B</nombreproducto>
14 <fabricante>LG</fabricante>
15 <precio>1000</precio>
16 </producto>
17 <producto>
18 <nombreproducto>Tablet Apple iPad Pro 12.9</nombreproducto>
19 <fabricante>Apple</fabricante>
20 <precio>1100</precio>
21 </producto>
22 <producto>
23 <nombreproducto>Altavoz Sonos Beam</nombreproducto>
24 <fabricante>Sonos</fabricante>
25 <precio>400</precio>
26 </producto>
27 </productos>
OK
27 : 13
producto.xml
productos
producto
nombrepro..
fabricante
Lenovo
precio
1200
Samsung
900
nombre..
1100
fabrican..
Apple
nombrep..
Sonos
400
LG
1000
2 Results, 49 b
Result
Smartphone Samsung Galaxy S21
Altavoz Sonos Beam
All
Total Time: 1.84
Info
Compiling:
- merge: descendant::producto
- rewrite to predicate: starts-with(fabricante/text(), "S")
- inline for $i_0 in util:root(.)/descendant:: producto[starts-with(fabricante/text(), "S")]
- simplify FLWOR expression: util:root(.)/ descendant::producto[starts-with(fabricante/text(), "S")]/nombreproducto/text ()
Optimizing:
- rewrite context value: . -> db:get-pre(" producto", 0)
- rewrite util:root(nodes): util:root(db:get-pre("producto", 0)) -> db:get-pre("producto ", 0)
- convert to child steps: descendant:: producto[starts-with(fabricante/text(), "S")]
Optimized Query:
db:get-pre("producto", 0)/*:productos/*: producto[starts-with(fabricante/text(), "S")]/
producto.xml
productos
db:get("producto", "producto.xml")/productos/producto/text()
81 MB
20°C Soleado
13:47