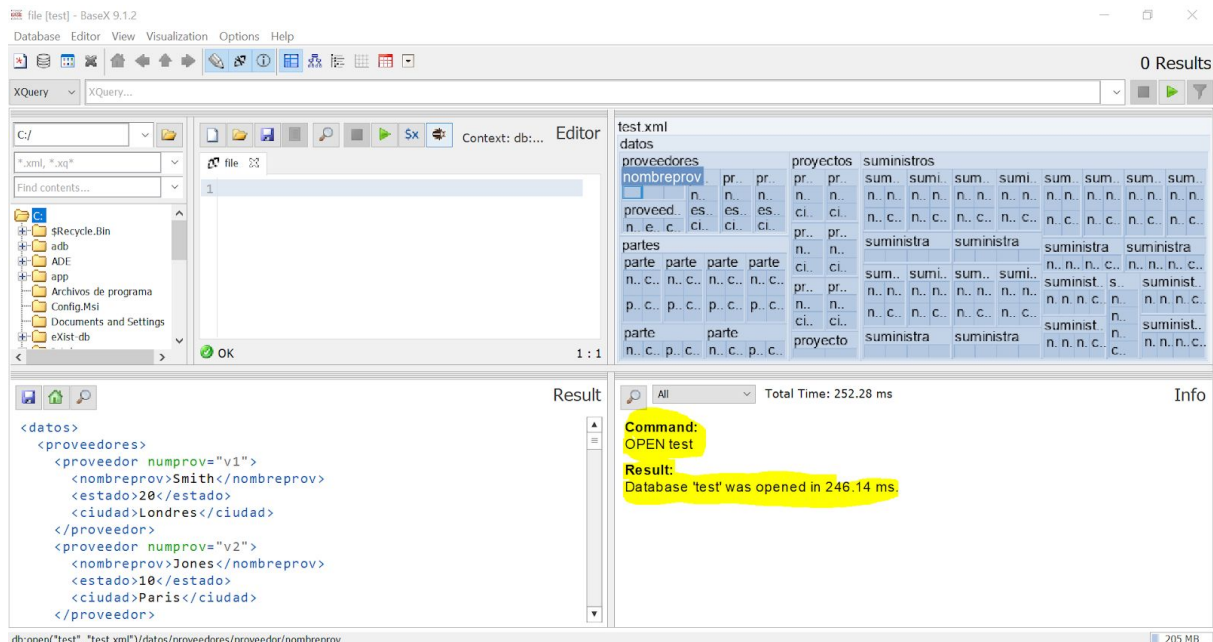


Consultas FLWOR con XBase

Para crear una consulta FLWOR debemos en primer lugar tener cargada la base de datos con el documento XML del que queremos crear consultas. Si se ha cargado correctamente debemos tener una pantalla similar a esta:



En la zona resaltada de amarillo vemos que se ha cargado completamente.

Vamos crear dos consultas a partir del fichero de ejemplo:

Consulta 1: Ciudad de los proveedores

Extraer la ciudad de los proveedores (no debe aparecer la etiqueta) que tengan un estado mayor de 15.

```
for $proveedor in doc("test.xml")/datos/proveedores/proveedor
where $proveedor/estado > 15
return $proveedor/ciudad/text()
```

- La primera línea de for, crea una variable llamada proveedor, donde guardará los elementos situados en el documento *test.xml*, dentro del elemento raíz *datos*, dentro de los *proveedores* y lo que guardará será cada nodo *proveedor*.
- La segunda línea filtra con una cláusula *where*, los proveedores que tienen un estado mayor que 15.
- La tercera línea devuelve la ciudad sin etiquetas (si quisiéramos que aparecieran las etiquetas deberíamos dejar únicamente \$proveedor/ciudad) de cada proveedor, ya que la instrucción return se repetirá tantas veces como tuplas tenga guardadas del *for*.

The screenshot shows the XBase 9.1.2 application window. The top menu bar includes File, Database, Editor, View, Visualization, Options, and Help. The main window is divided into several panes:

- Left Pane:** A file explorer showing the directory structure of the XML document, including folders like 'sRecycle.Bin', 'adb', 'ADE', 'app', 'Archivos de programa', 'Config.Msi', 'Documents and Settings', and 'eXist-db'.
- Top Center Pane:** The XQuery editor. The query entered is: `for $proveedor in doc("test.xml")/datos/proveedores/proveedor where $proveedor/estado > 15 return $proveedor/ciudad/text()`.
- Right Pane:** A tree view of the XML document structure. The 'datos' node is expanded, showing sub-nodes like 'proveedores', 'proyectos', 'suministros', 'partes', and 'proveedor'. The 'proveedor' node is highlighted in red.
- Bottom Left Pane:** The 'Result' pane showing the output of the query: 'Londres', 'Paris', 'Londres', and 'Atenas'.
- Bottom Right Pane:** The 'Info' pane showing query compilation details, including the optimized query: `for $proveedor_0 in db:open-pre("test", 0)/"datos/*:proveedores/*:proveedor[(15.000000000000002 <= *:estado)] return $proveedor_0/*:ciudad/text()`.

Especial detalle al resultado en el cuadrante inferior izquierdo y cuadrante superior derecho donde vemos los resultados que hemos recuperado marcados en rojo dentro del esquema general XML.

Vamos a fijarnos también que en XBase la consulta se escribe en una sola línea, aunque en los ejercicios podemos separarlo con saltos de línea para que sea más legible.

Ahora, ahora vamos a aplicar la función concat para que cada resultado lleve un texto delante:

```
for $proveedor in doc("test.xml")/datos/proveedores/proveedor
where $proveedor/estado > 15
return concat('Ciudad', ' ', $proveedor/ciudad/text())
```

Vamos a fijarnos en la función concat, se encarga de concatenar los resultados, en este caso la palabra Ciudad, un espacio, y el texto de la ciudad de cada proveedor.



```
Ciudad Londres
Ciudad Paris
Ciudad Londres
Ciudad Atenas
```

Si queremos enmarcar cada resultado entre etiquetas. Debemos escribirlo de la siguiente forma:

```
for $proveedor in doc("test.xml")/datos/proveedores/proveedor
where $proveedor/estado > 15
```

```
return <respuesta> { concat('Ciudad', '  
' , $proveedor/ciudad/text()) } </respuesta>
```



```
<respuesta>Ciudad Londres</respuesta>  
<respuesta>Ciudad Paris</respuesta>  
<respuesta>Ciudad Londres</respuesta>  
<respuesta>Ciudad Atenas</respuesta>
```

Vamos a fijarnos en el return, ponemos las etiquetas XML entre las que queremos marcar cada uno de los resultados, unas llaves y dentro el resultado de la consulta FLWOR que estábamos utilizando anteriormente.

Consulta: filas de la tabla partes

Para contar una serie de elemento, podemos utilizar la función count,

En esta consulta vamos a ver cuántas partes existen, es decir, el total de filas de la tabla partes.

Para resolverlo una opción sería resolverlo así:

```
for $partes in doc("test.xml")/datos/partes  
return count($partes/parte)
```



6

Para marcarlo entre etiquetas:

```
for $partes in doc("test.xml")/datos/partes  
return <partestotales> { count($partes/parte) }  
</partestotales>
```



```
<partestotales>6</partestotales>
```

Esta solución funciona en realidad estamos haciendo un bucle de una sola iteración.

En este caso, se puede recurrir directamente a la función count que permite contar el número de elementos de una consulta parcial sin tener que hacer siquiera el recorrido.

```
<partestotales> { count(doc("datos.xml")/datos/partes/parte)
} </partestotales>
```



```
<partestotales>6</partestotales>
```