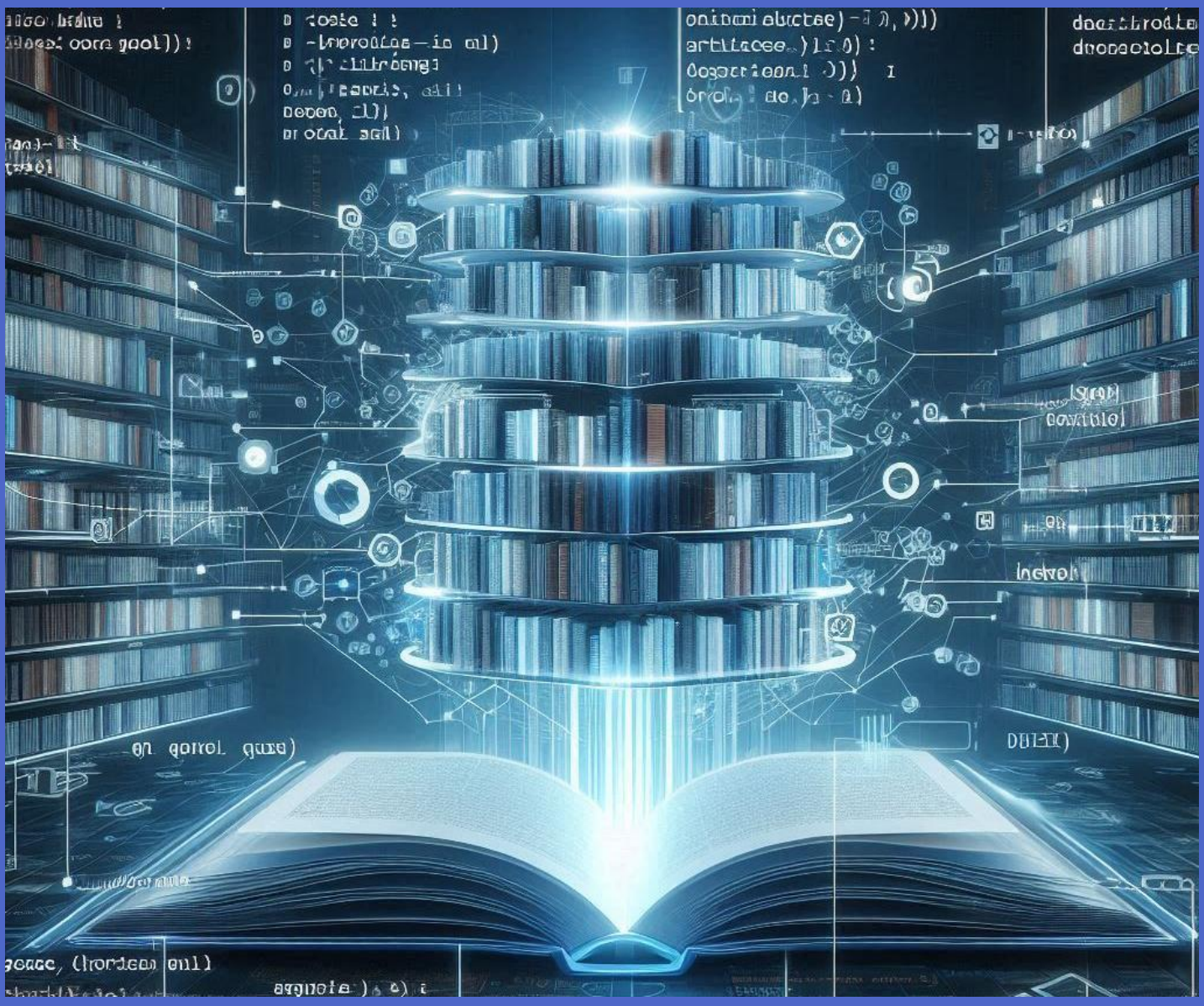
L E N G U A J E D E M A R C A S

CASO PRACTICO 1UD6



ALUMNO CESUR

24/25

Alejandro Muñoz de la Sierra

PROFESOR

Mercedes Piedra

I N T R O D U C C I O N

Hoy nos lanzamos a explorar la gestión de datos en XML, utilizando XQuery de una forma muy práctica. Usaremos BaseX, un sistema de bases de datos que, en esencia, está diseñado para trabajar con información en XML de modo bastante eficiente.

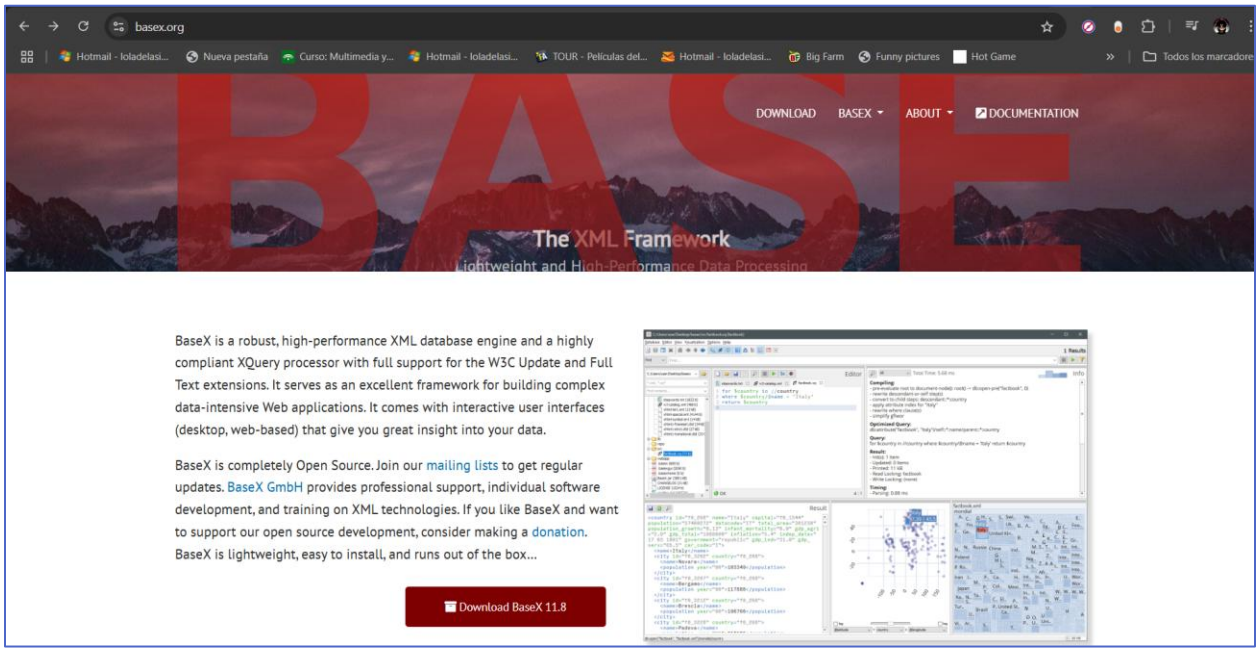
Imagina por un momento un entorno laboral en una librería digital, donde aparecen consultas puntuales sobre un extenso catálogo de libros; aquí, normalmente, se busca optimizar procesos internos a través del análisis y la extracción de información relevante.

Con este ejercicio, nos familiarizaremos con el manejo de herramientas especializadas como XQuery, al mismo tiempo que ponemos en práctica conceptos clave sobre el almacenamiento y búsqueda de datos estructurados. En la mayoría de los casos, resulta ser una oportunidad ideal para adentrarse en entornos donde la información se maneja de forma profesional, acercándonos poco a poco a esos escenarios tan dinámicos.

0 1

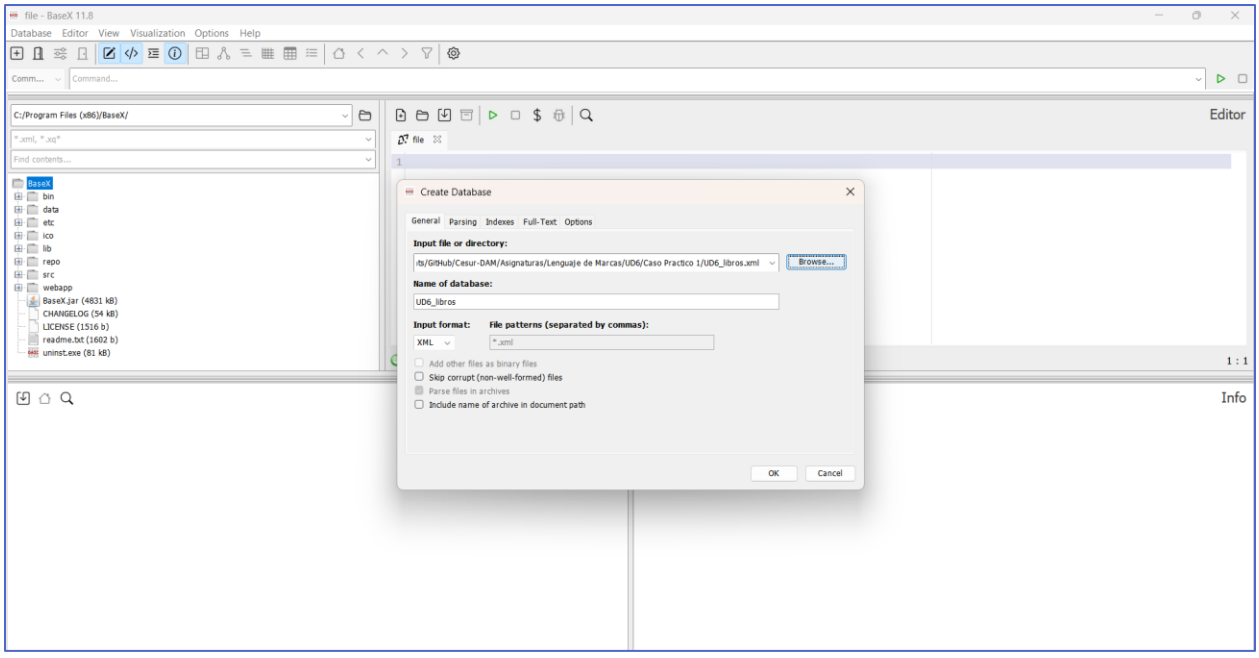
INSTALACION DE BASEX Y PRIMEROS PASOS

Vamos al sitio de baseX y procedemos a descargar el instalador y ejecutarlo:

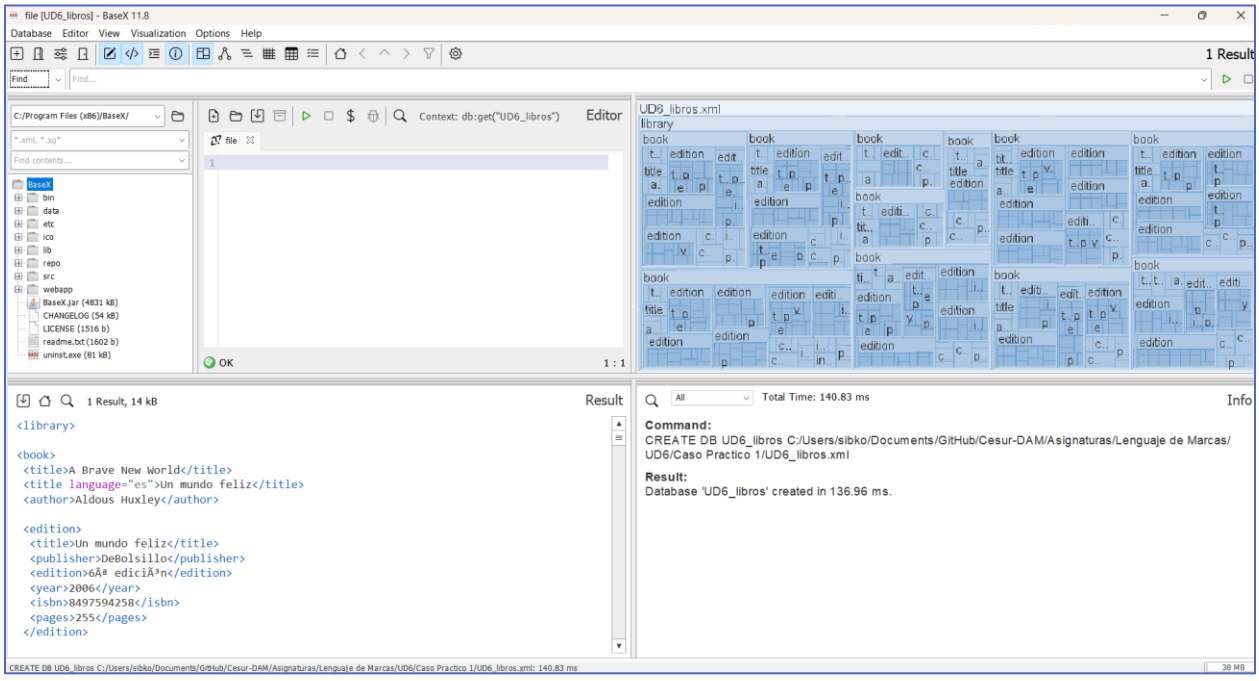


Damos contraseña de administrador, ejecutamos el programa.
Procedemos a abrir el archivo xml, pero el programa nos recomiendo crear la base de datos para el mismo, buscamos en internet y encontramos que si vamos a usarlo para varias consultas y guardar resultados es mejor crear la base.

El proceso es muy simple, elegimos la ubicación y el archivo libros.xml y le damos un nombre a la base:



Base creada con la estructura del archivo libros.xml:



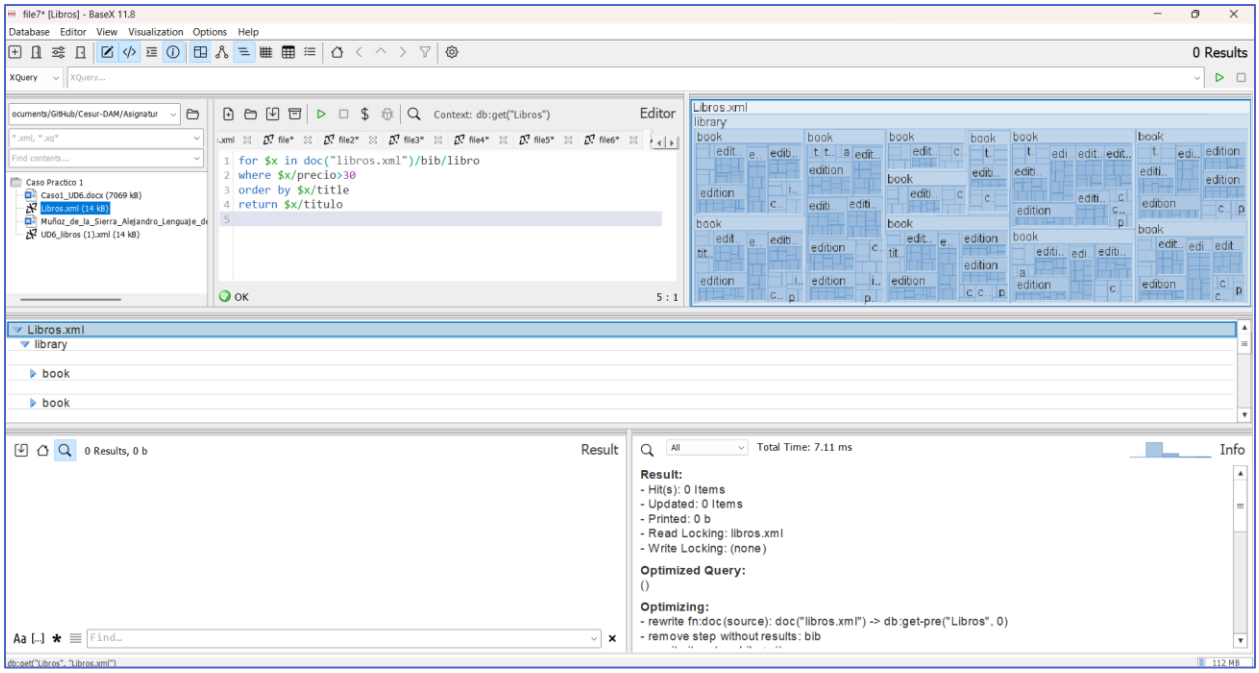
0 2

C O N S U L T A S C O N X Q U E R Y

1. ¿Qué obtendrás con la siguiente consulta?

```
for $x in doc("libros.xml")/bib/libro
where $x/precio>30
order by $x/title
return $x/titulo
```

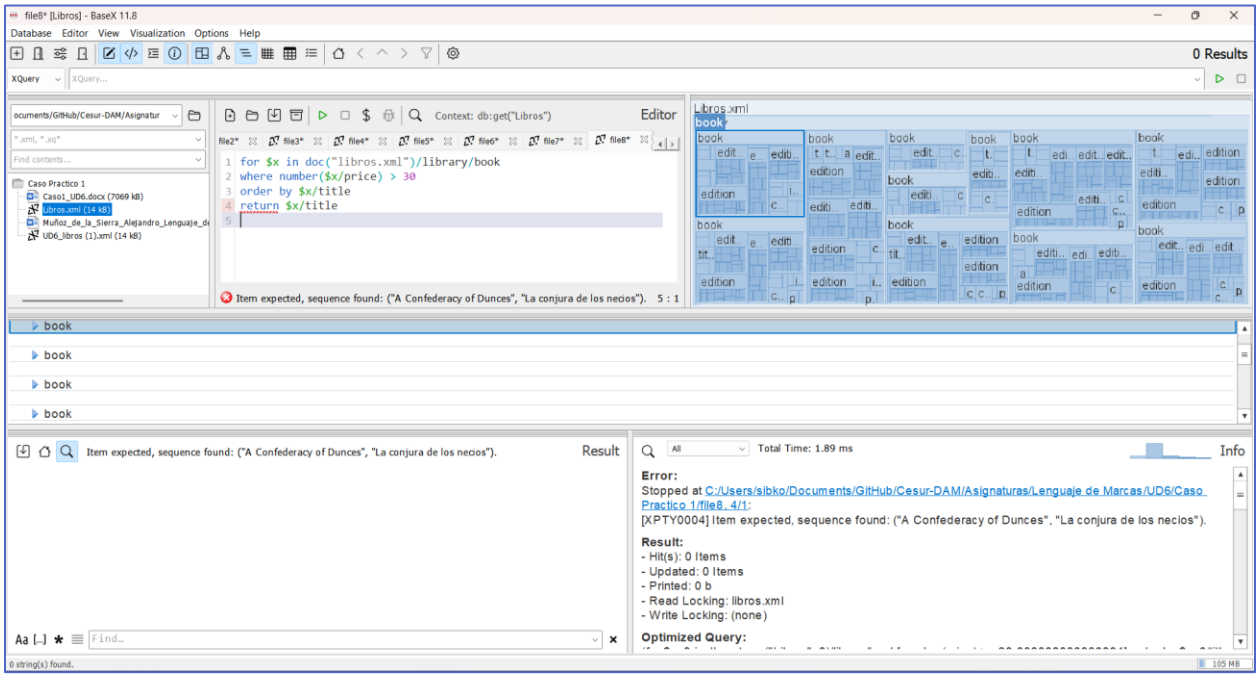
La siguiente consulta no devuelve ningún resultado, debido a que estamos buscando en nodos con nombres incorrectos y en español, cuando nuestro documento xml tiene la estructura en inglés.



Corregimos la consulta para que devuelva el contenido que se espera:

```
for $x in doc("libros.xml")/library/book
where number($x/price) > 30
order by $x/title
return $x/title
```

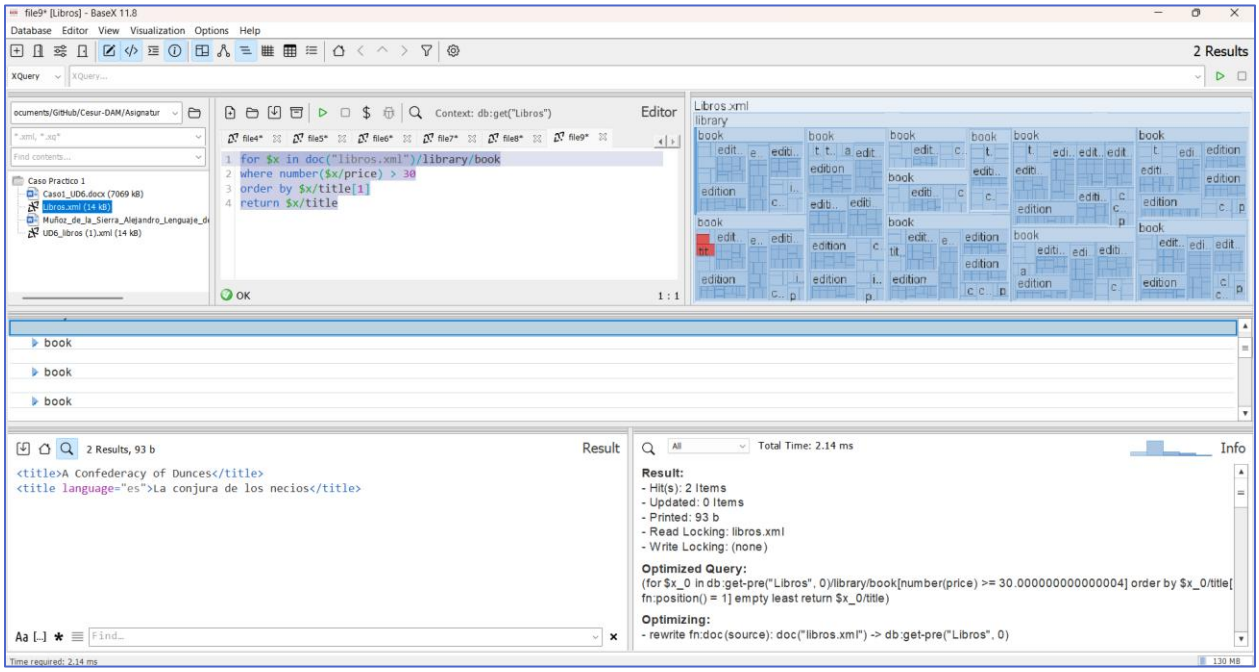
en este caso los nodos están con sus nombres correctos en inglés, y podemos acceder al título del libro que cumpla cuyo precio es mayor a 30:



Nos devuelve un solo resultado, aunque nos indica con un error que estamos devolviendo una secuencia, porque en realidad hay dos nodos title.

Una forma sencilla de resolver el error y tener una consulta más limpia, sería ordenado la secuencia por uno de los dos nodos, en este caso el primer title y que devuelva todos los nodos title, con el código que exponemos a continuación.

```
for $x in doc("libros.xml")/library/book
where number($x/price) > 30
order by $x/title[1]
return $x/title
```



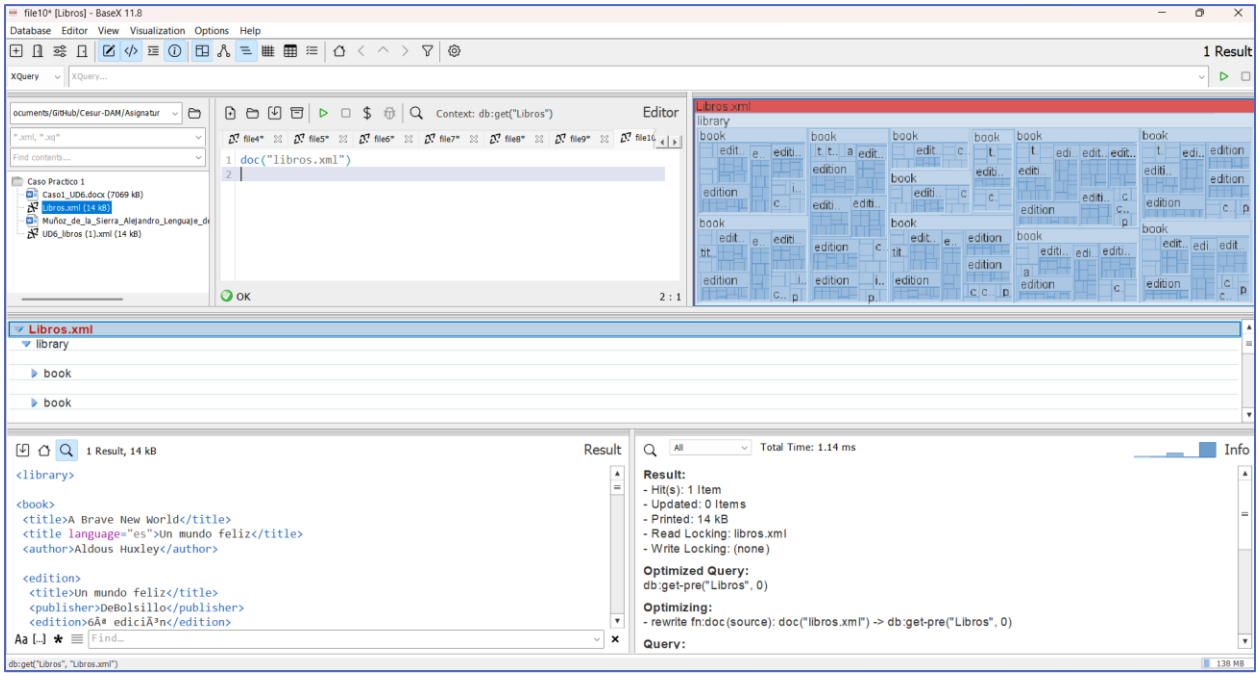
Como vemos la consulta devuelve sin errores esta vez el contenido exigido y ordenado correctamente. Incluso vemos marcado en rojo en el mapa el resultado.

2.Obtén todo el documento entero.

Esta consulta a diferencia de la anterior es muy sencilla:

doc("libros.xml")

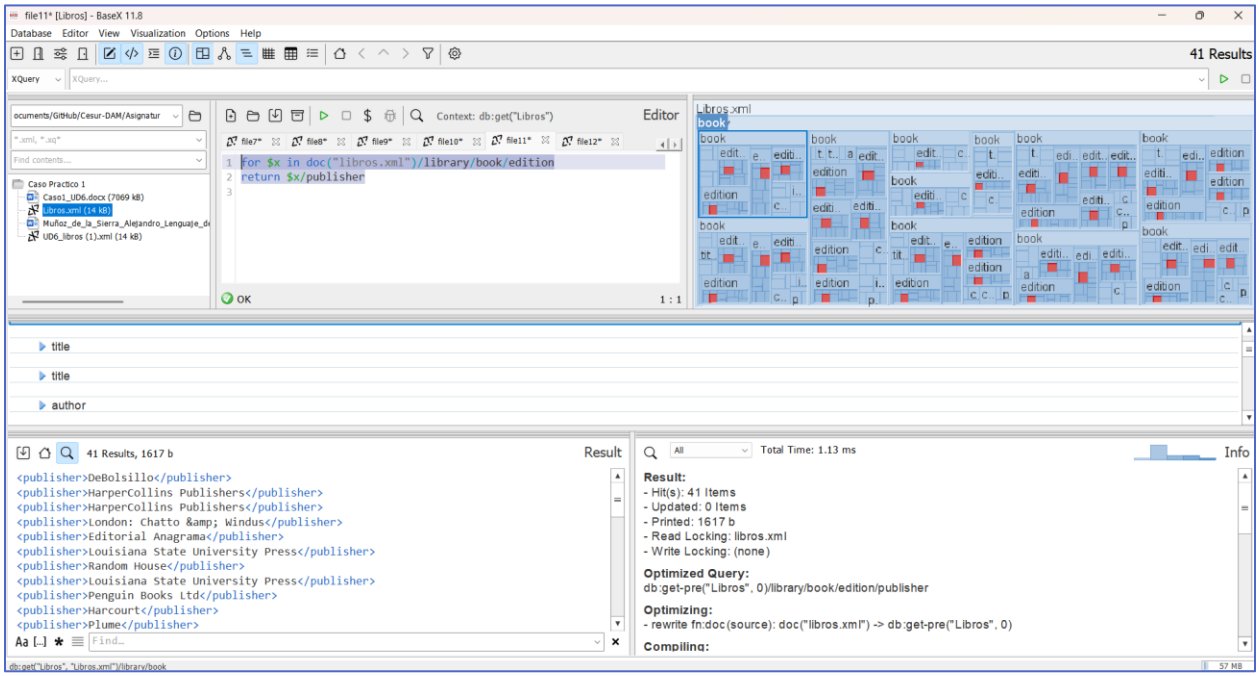
No necesitamos decirle la dirección del archivo libros.xml porque ya tenemos creada la base de datos con el nodo raíz el directorio de ese archivo.



3.Obtén las editoriales de los diferentes libros.

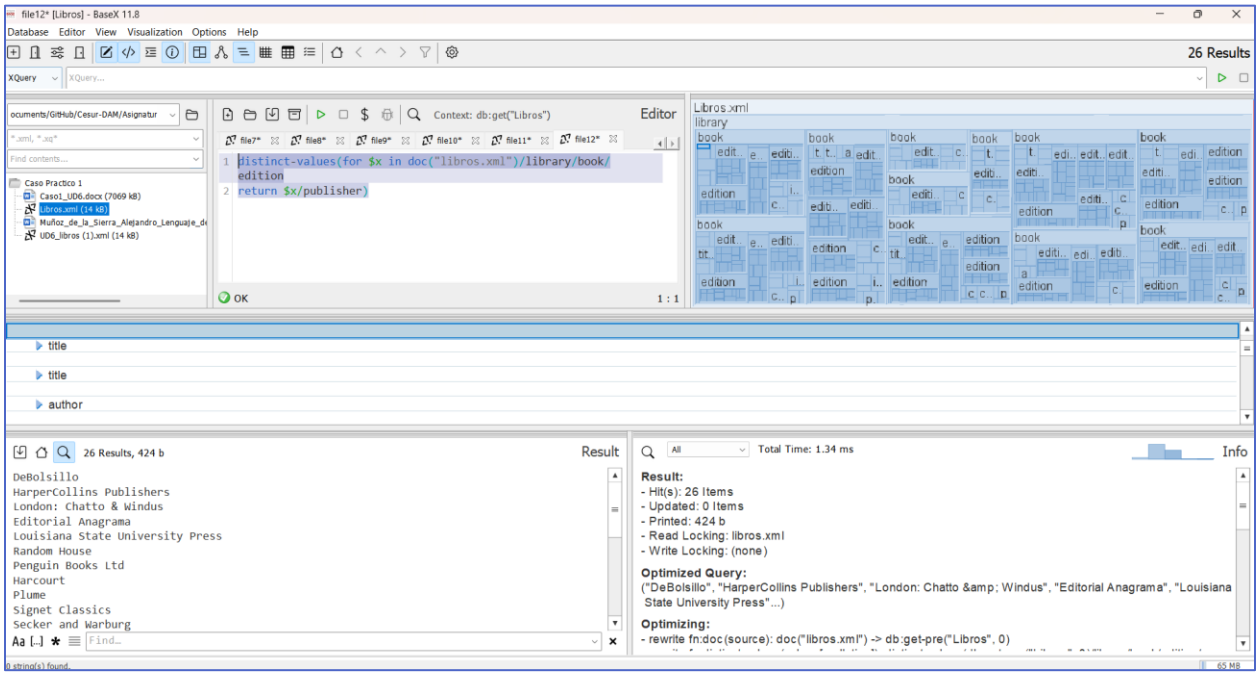
Para esta consulta e interpretado que se piden sólo las editoriales sin que se vea el libro de cada editorial. Para ello simplemente accedemos al nodo donde está esa información con el siguiente código:

```
for $x in doc("libros.xml")/library/book/edition
return $x/Publisher
```



Aunque vemos que no necesitamos tener los datos repetidos, porque si queremos saber todas las editoriales, interpreto que se puede omitir duplicados. Resolvemos a una consulta más limpia con una función que sólo devuelve valores únicos.

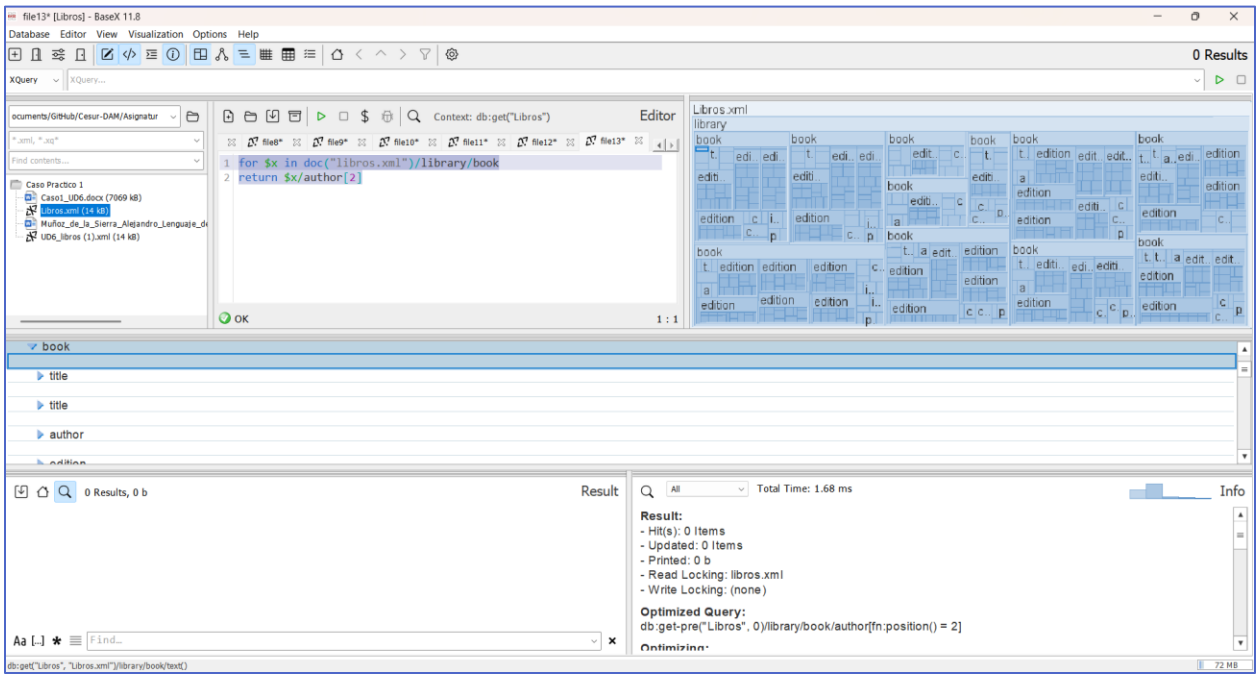
distinct-values(for $x in doc("libros.xml")/library/book/edition
return $x/publisher)



4.Obtén el segundo autor de cada libro.

Esta consulta es extraña porque realmente está pidiendo el segundo autor de cada libro, suponemos que habrá algún libro con más de un autor.

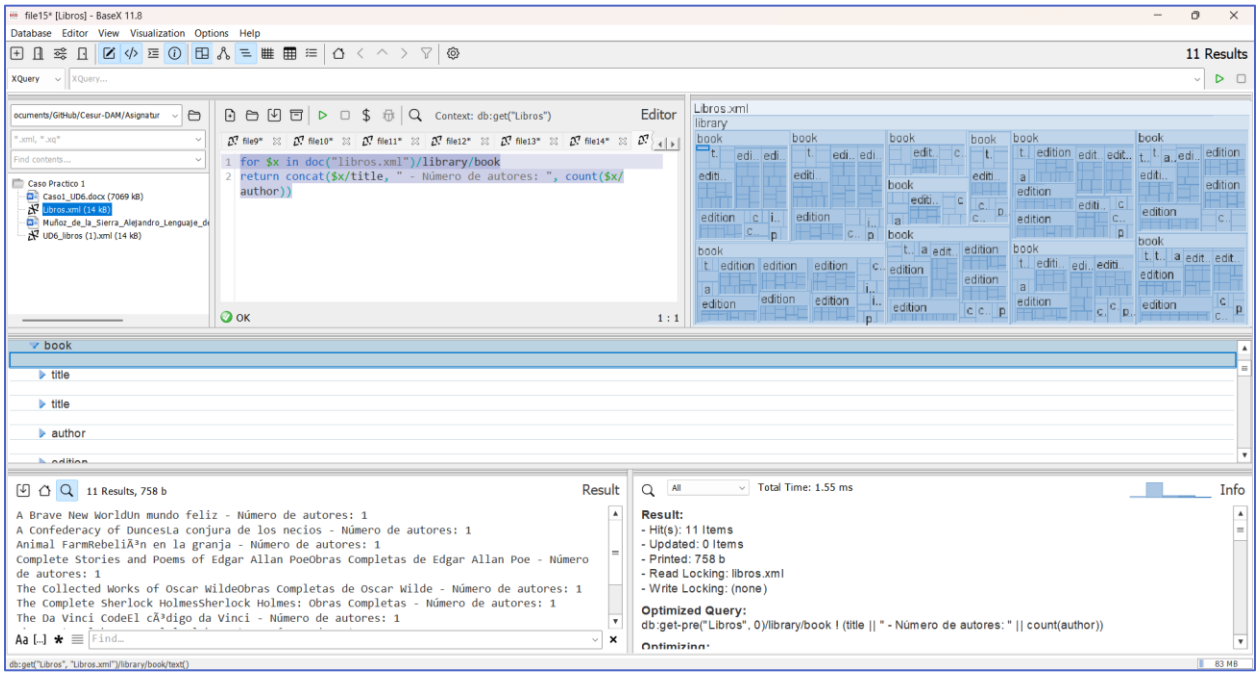
```
for $x in doc("libros.xml")/library/book
return $x/author[2]
```



Vemos que no existe ningún libro con dos nodos autor.

Podríamos hacer una consulta extra para mostrar cada libro y que cuente el número de autores que tiene. Algo como así:

```
for $x in doc("libros.xml")/library/book
return concat($x/title, " - Número de autores: ", count($x/author))
```

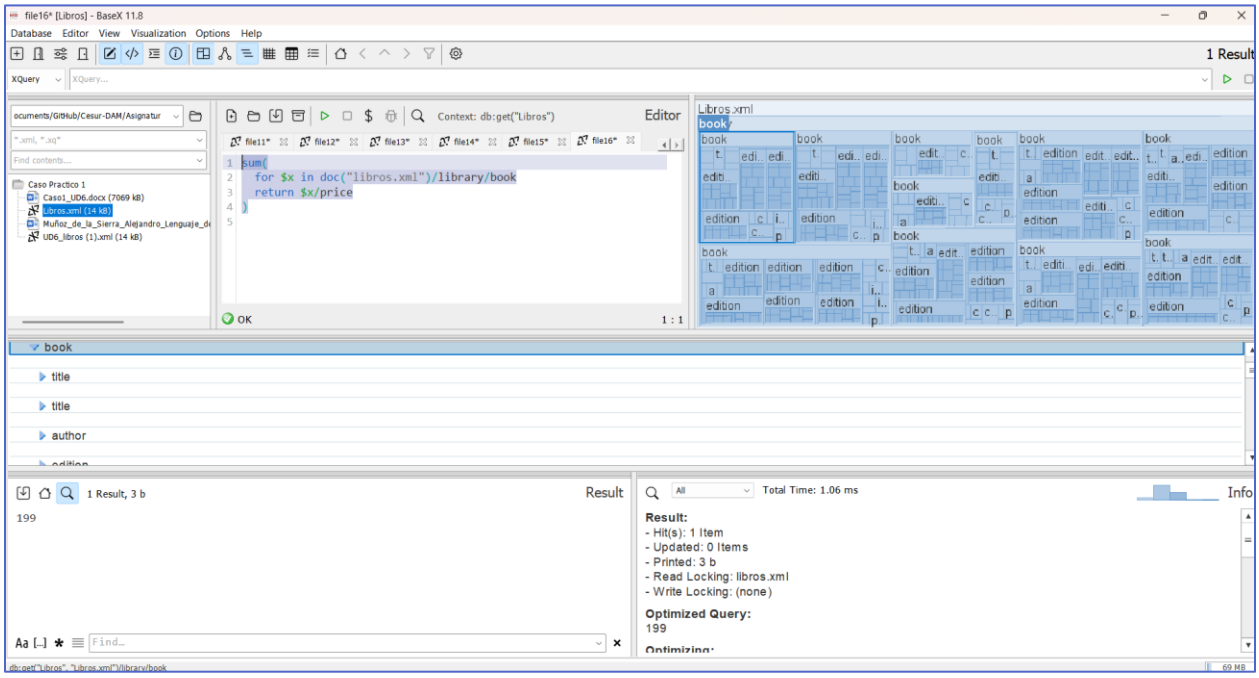


Ahí observamos que no hay libros con dos autores.

5.Suma todos los precios de los libros.

Esta consulta es simplemente , acceder al nodo precio de cada libro como parámetro de la función sum():

```
sum(
  for $x in doc("libros.xml")/library/book
  return $x/price
)
```



0 4

C O N C L U S I O N E S

En este proyecto nos pusimos a trabajar con XQuery sobre un documento XML que se asemeja a una biblioteca repleta de libros. Con expresiones FLWOR (for, let, where, order by, return) logramos extraer datos muy concretos –títulos, editoriales, precios y hasta el número de autores en cada obra–, lo que en la mayoría de los casos nos sorprendió por su claridad.

A lo largo del proceso, fuimos cuidadosos al conservar la estructura y los nombres originales de los nodos, aunque la navegación entre elementos se volvió, en parte, un reto interesante. Se experimentó con distintas formas de presentar los resultados, mezclando valores sencillos con formatos algo más refinados; en pocas palabras, buscamos que la información se mostrara de manera natural y variada.

En definitiva, este trabajo nos ayudó a fijar ideas básicas sobre la gestión y consulta de datos estructurados, fortaleciendo nuestro análisis y la atención al detalle –algo fundamental al dominar la sintaxis de XQuery. Generalmente hablando, cada pequeño reto fue una oportunidad para aprender y avanzar, haciendo la experiencia aún más enriquecedora.

0 5

R E F E R E N C I A S

https://docs.basex.org

https://www.w3.org/TR/xquery-31/

https://www.w3schools.com/xml/xquery_intro.asp

https://www.freeformatter.com/xquery-formatter.html

https://www.lawebdelprogramador.com/cursos/XQuery/

https://www.youtube.com/watch?v=OMewFZNs3wY