José Carlos Manjon Carrasco

Lenguaje de marcas

Tarea 6

Contenido

[Ejercicio 1. 2](#_Toc166320186)

[Ejercicio 2. 3](#_Toc166320187)

[Ejercicio 3 4](#_Toc166320188)

[Ejercicio 4. 5](#_Toc166320189)

[Ejercicio 5. 6](#_Toc166320190)

[Ejercicio 6. 6](#_Toc166320191)

[Ejercicio 7. 7](#_Toc166320192)

[Ejercicio 8. 8](#_Toc166320193)

[Ejercicio 9. 9](#_Toc166320194)

[Ejercicio 10. 10](#_Toc166320195)

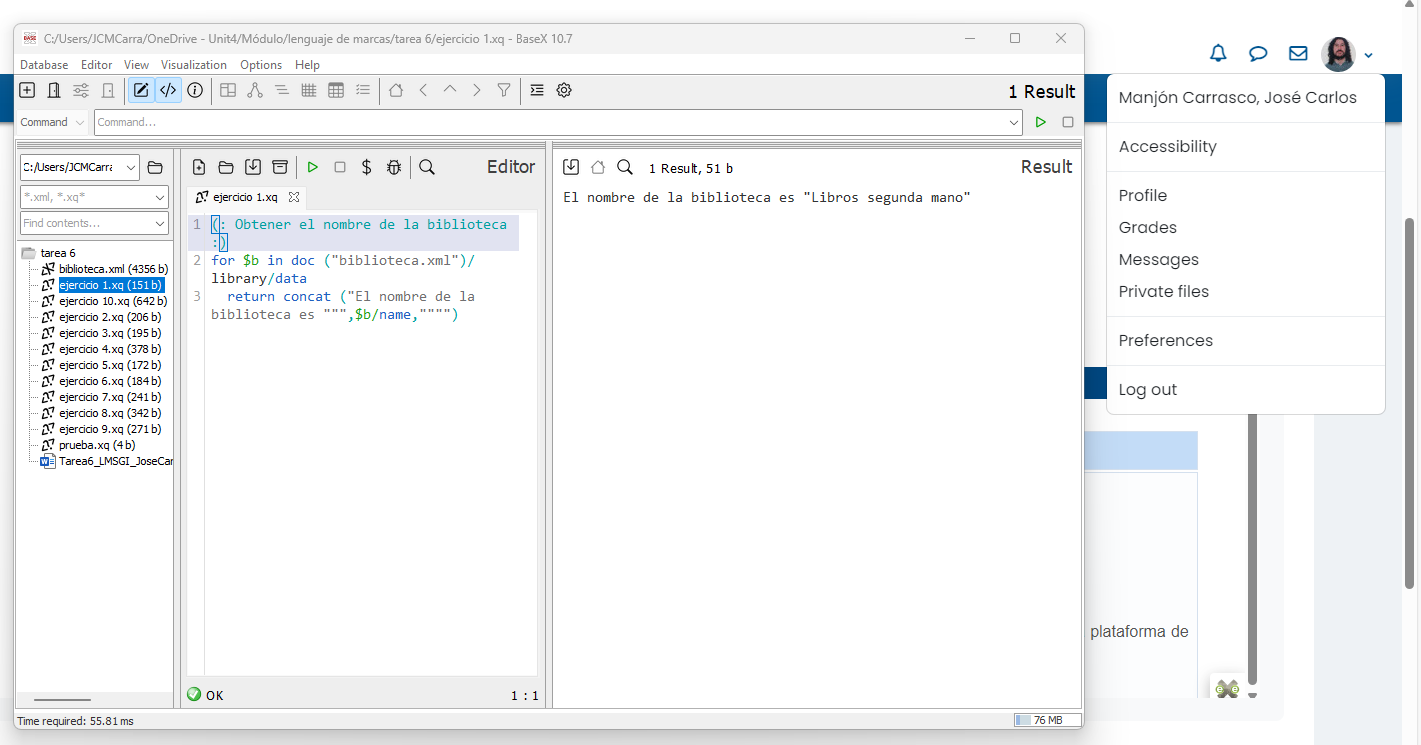
A partir del fichero [biblioteca.xml](https://www.juntadeandalucia.es/educacion/gestionafp/datos/tareas/DAM/LM_10013346/2023-24/DAM_LM_6_2023-24_Individual__867769/biblioteca.zip) deberás realizar los siguientes ejercicios de **XQuery** para obtener información relevante a partir de los datos almacenados en el fichero xml de origen. Deberás generar las consultas y probarlas tal y como se especifica en el **Apartado 2. Información de interés**, en el que se detalla cómo deberás realizar la entrega.

Ejercicio 1. Obtener el nombre de la biblioteca

(: Obtener todos los títulos de los libros del fichero libros.xml junto con los autores de cada libro. Usando solo la cláusula for:)

for $b in doc ("biblioteca.xml")/library/data

return concat ("El nombre de la biblioteca es """,$b/name,"""")



Ejercicio 2. Encontrar todos los libros publicados después de ***2010***. Deberás mostrar la información de la siguiente forma:

<book>  
                <title>Título del libro</title>  
                <year>Año de publicación</year>  
          </book> *(:Encontrar todos los libros publicados después de 2010:)*

*for $b in doc ("biblioteca.xml")//library/catalog/book*

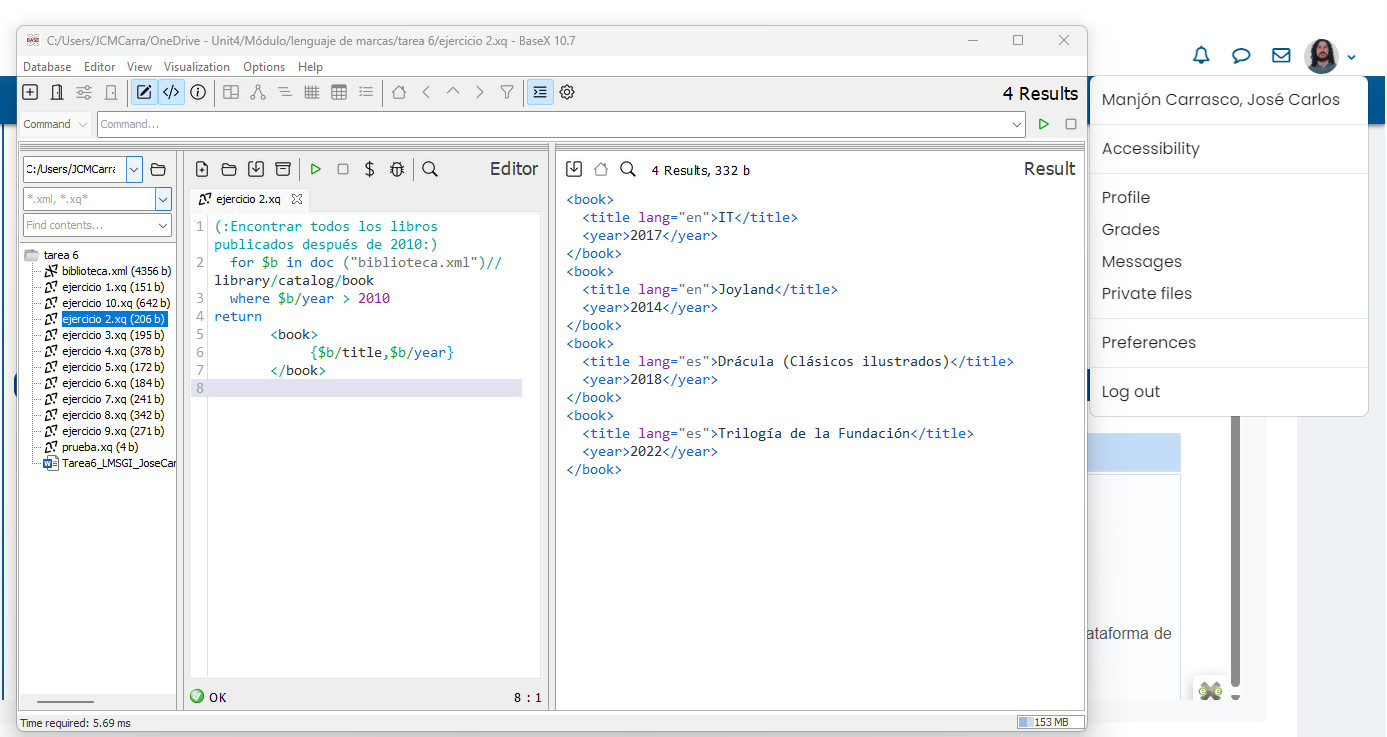
where $b/year > 2010

return

<book>

{$b/title,$b/year}

</book>



Ejercicio 3. Obtener el nombre de los autores de los libros de ***terror***

**(:Obtener el nombre de los autores de los libros de terror:)**

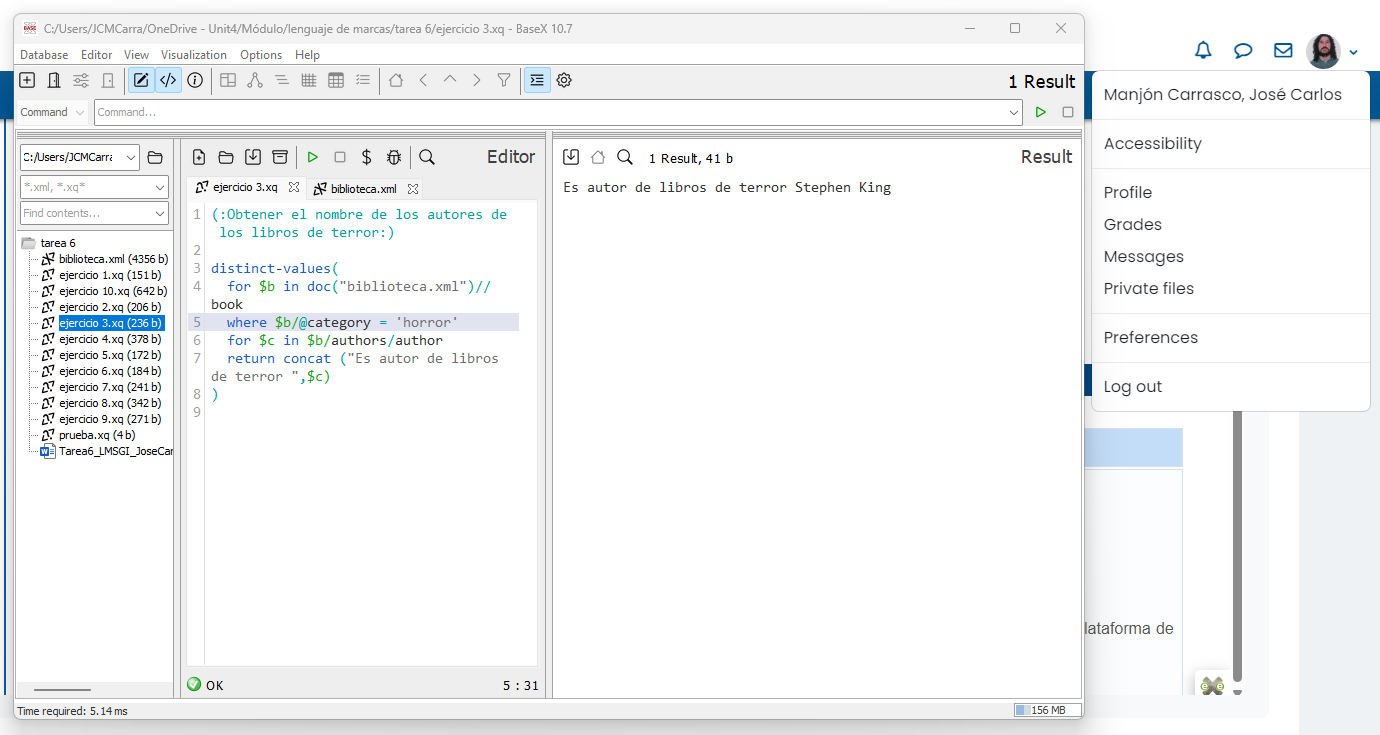
**distinct-values(**

**for $b in doc("biblioteca.xml")//book**

**where $b/@category = 'horror'**

**for $c in $b/authors/author**

**return concat ("Es autor de libros de terror ",$c))**



Ejercicio 4. Obtener información sobre los cursos impartidos en ***enero*** de ***2024*** con el siguiente formato (se tendrán que concatenar los campos adecuados para obtener la fecha):

<course>  
                <title>Título del libro</title>  
                <date>Fecha del curso (en el formato día/mes/año)</date>  
          </course>

for $c in doc("biblioteca.xml")//library/courses/course

where $c/date/year = 2024 and $c/date/month = 1

return

<course>

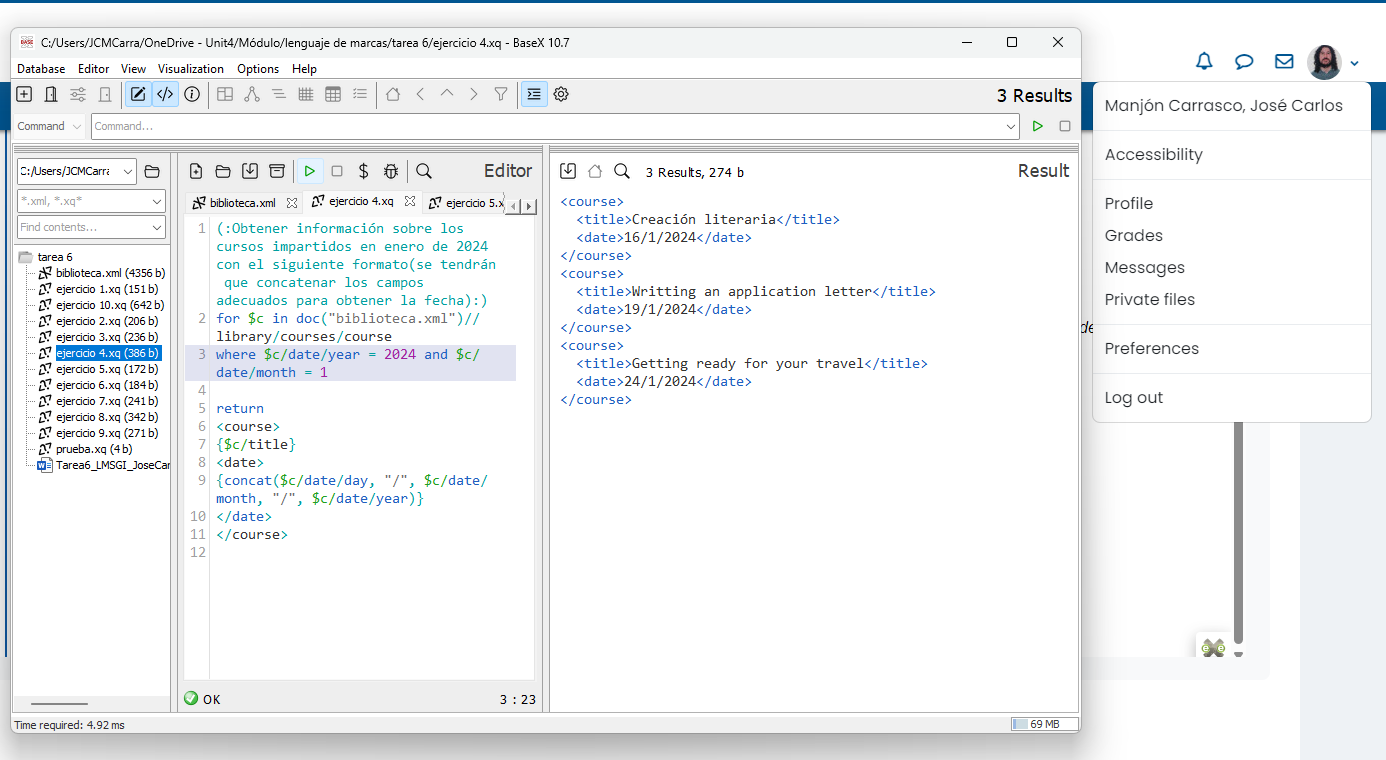
{$c/title}

<date>

{concat($c/date/day, "/", $c/date/month, "/", $c/date/year)}

</date>

</course>

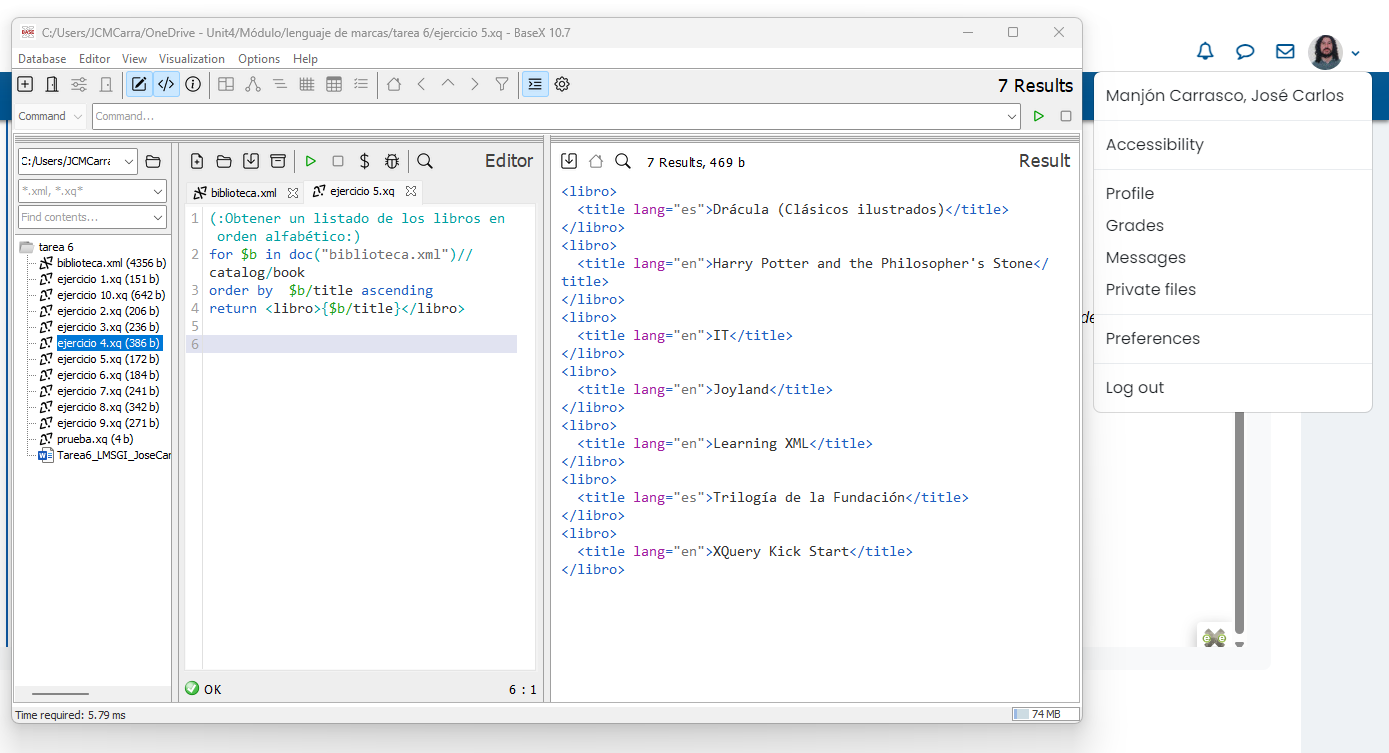


Ejercicio 5. Obtener un listado de los libros en ***orden alfabético***

**for $b in doc("biblioteca.xml")//catalog/book**

**order by $b/title ascending**

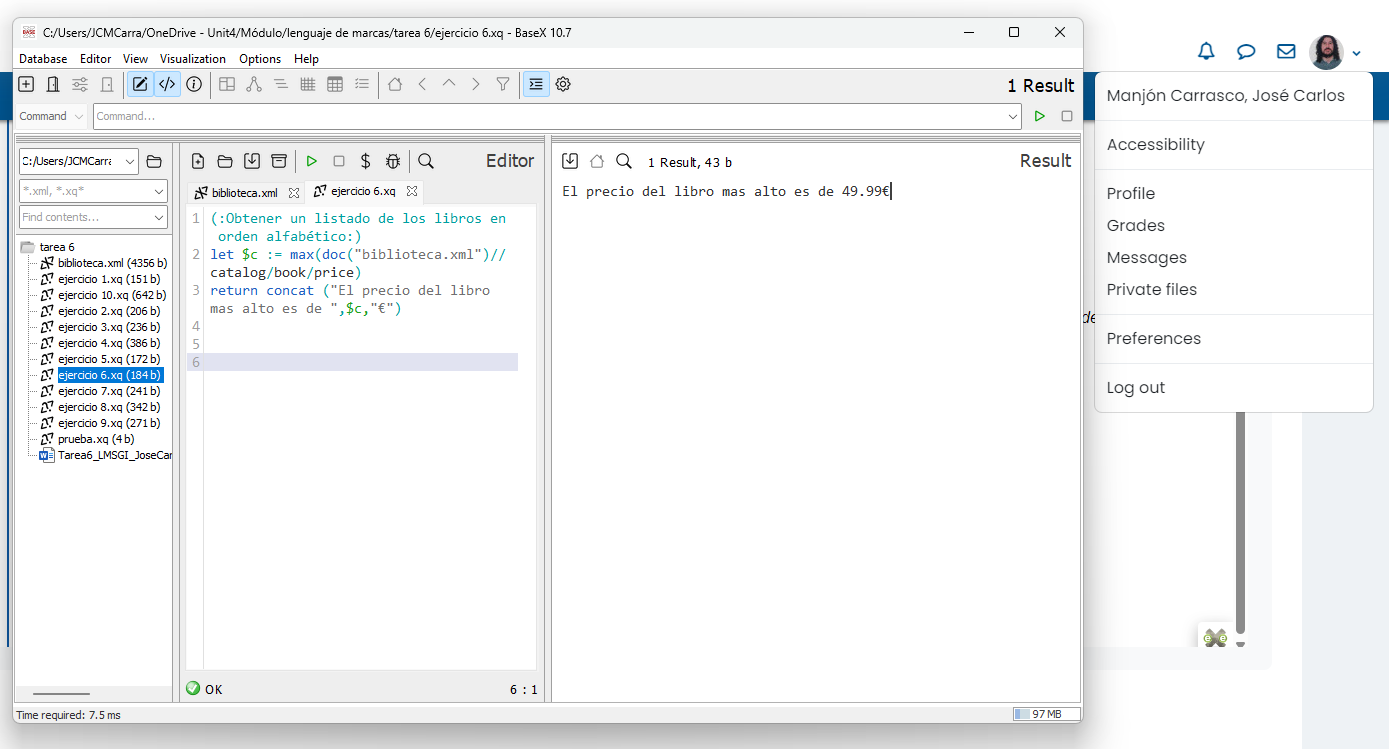
**return <libro>{$b/title}</libro>**



Ejercicio 6. Encontrar el libro ***más caro*** del catálogo

let $c := max(doc("biblioteca.xml")//catalog/book/price)

return concat ("El precio del libro mas alto es de ",$c,"€")

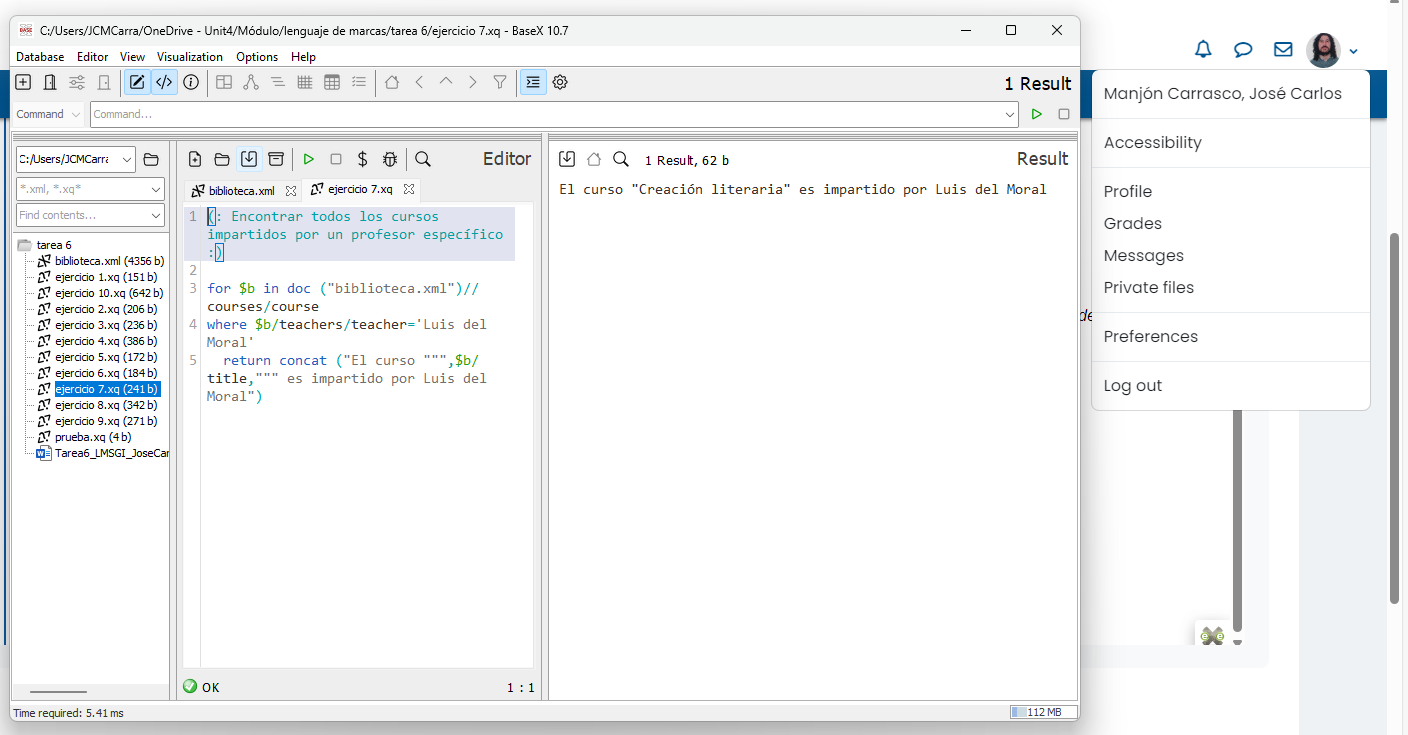


Ejercicio 7. Encontrar todos los cursos impartidos por un ***profesor específico***

**for $b in doc ("biblioteca.xml")//courses/course**

**where $b/teachers/teacher='Luis del Moral'**

**return concat ("El curso """,$b/title,""" es impartido por Luis del Moral")**



Ejercicio 8. Obtener el nombre de todos los libros, junto con el autor y el género, ordenados ***alfabéticamente*** por género

for $b in doc("biblioteca.xml")//book

let $cb := $b/@category

order by $cb

return

<libro>

{

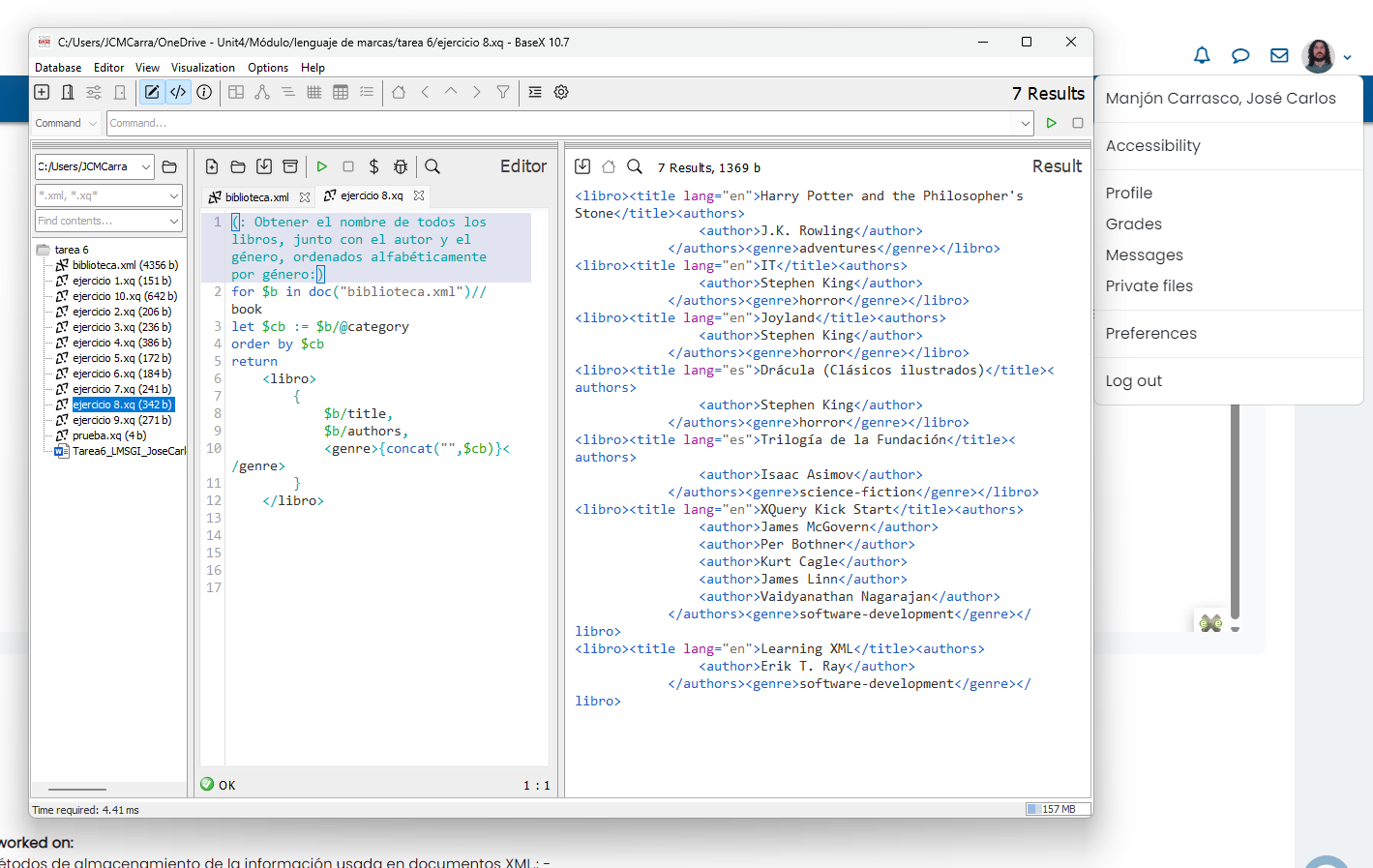
$b/title,

$b/authors,

<genre>{concat("",$cb)}</genre>

}

</libro>



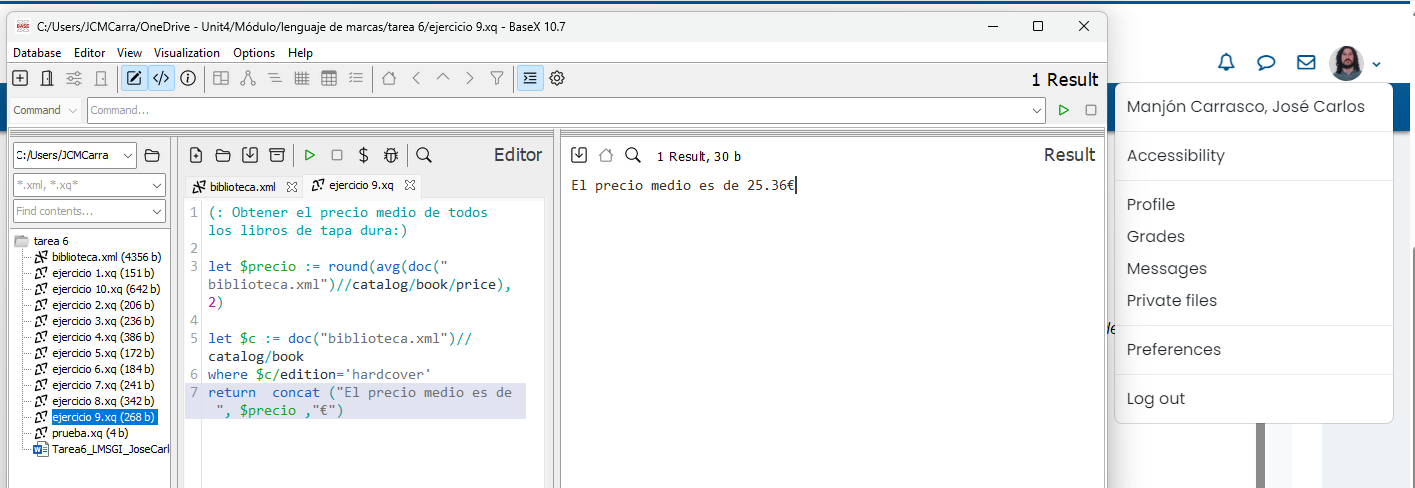
Ejercicio 9. Obtener el precio medio de todos los libros de ***tapa dura***

let $precio := round(avg(doc("biblioteca.xml")//catalog/book/price),2)

let $c := doc("biblioteca.xml")//catalog/book

where $c/edition='hardcover'

return concat ("El precio medio es de ", $precio ,"€")



Ejercicio 10. Encuentra los libros***mas caros*** de cada categoría y muestra su ***categoría***, el ***título***, el ***precio*** y el ***autor***

(:Buscamos en los nódulos book el atributo category , pero no repetimos los atributos:)

for $categoria in distinct-values(doc("biblioteca.xml")//book/@category)

(:Declaramos la variable precio máximo en la ruta /book/price para aquellos nódulos book:)

let $precioMaximo := max(doc("biblioteca.xml")//book[@category=$categoria]/price)

return

<libro>

<categoria>{$categoria}</categoria>

{

for $b in doc("biblioteca.xml")//book

where $b/@category = $categoria and $b/price = $precioMaximo

return $b}

</libro>

