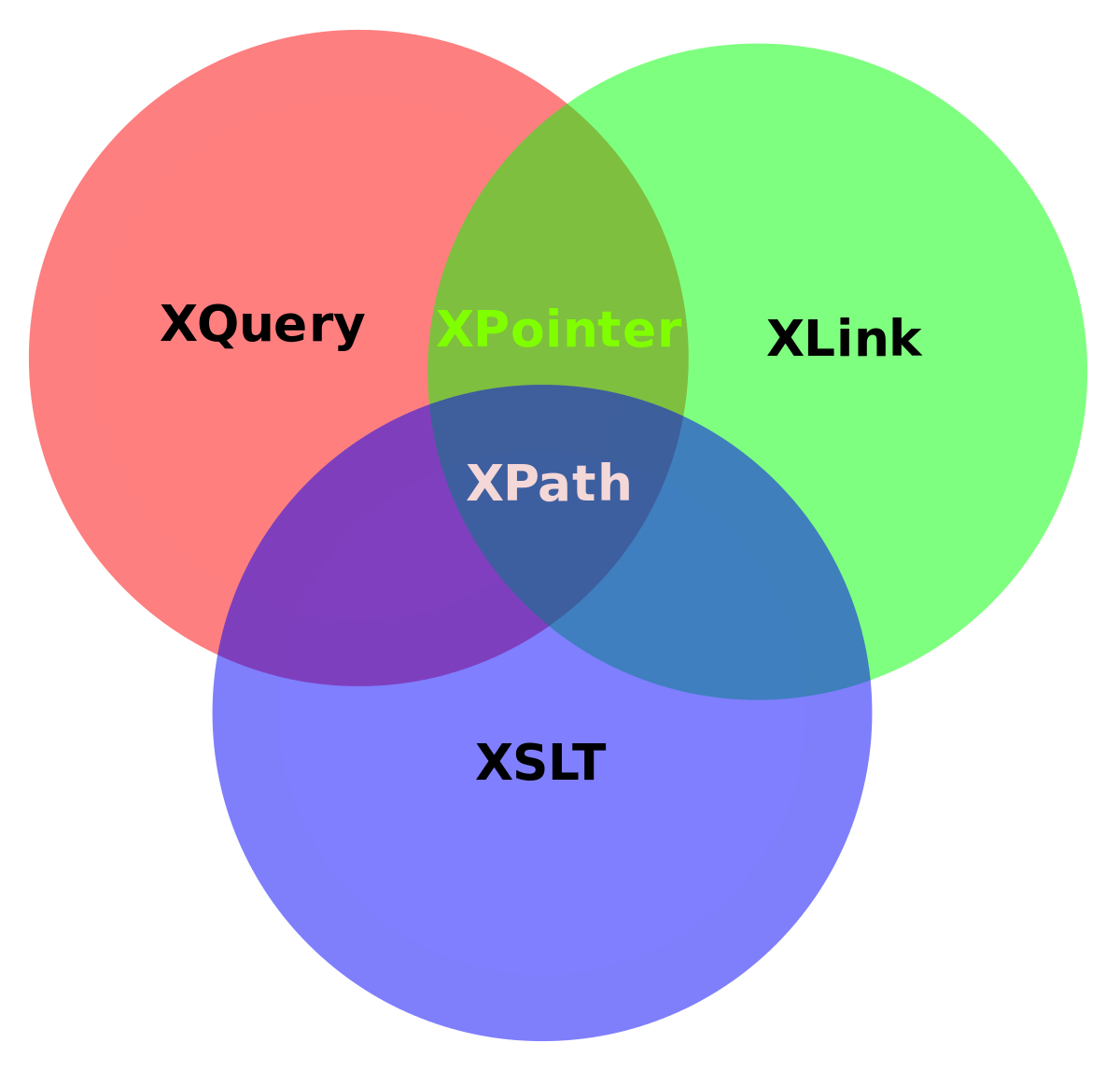
Actividad 8

Rubén Beltrán Muñoz

Lenguaje De Marcas

1º DAM



Tratamiento De La Información Con XML

Índice

[**1.** **Ejemplo de uso de la cláusula FOR. Obtener todos los títulos de los libros del fichero.** 2](#_Toc11869238)

[**1.** **Ejemplo de uso de la cláusula LET. Obtener todos los títulos de los libros del fichero.** 2](#_Toc11869239)

[**2.** **Ejemplo de uso de la cláusula FOR y LET juntas. Obtener todos los títulos de los libros del fichero, junto con los autores de cada libro.** 3](#_Toc11869240)

[**3.** **Listar publicación y título de los libros que tienen más de un autor.** 4](#_Toc11869241)

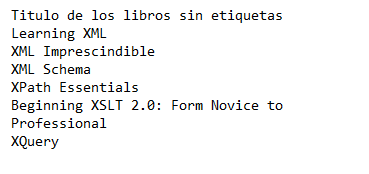
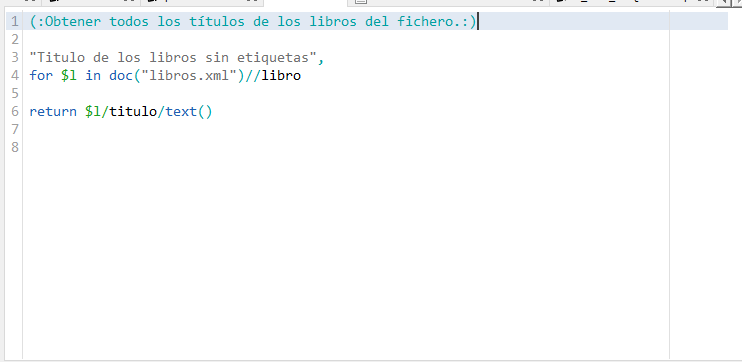
[**4.** **Por cada libro, obtener su título y el número de autores, agrupados en un elemento** 5](#_Toc11869242)

[**5.** **Ejemplo de uso de las cláusulas WHERE y ORDER BY en una consulta con dos ficheros. Obtener los títulos de los libros prestados con sus autores y la fecha de inicio y devolución del préstamo, ordenados por la fecha de inicio del préstamo.** 5](#_Toc11869243)

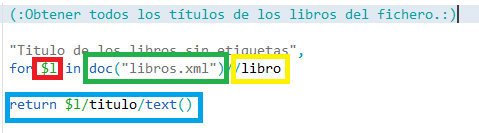
[**6.** **Ejemplo, hacer una consulta que devuelva los títulos de los libros almacenados en el fichero y su primer autor. En caso de que haya más de un autor para un libro se añade un segundo autor “cia”.** 7](#_Toc11869244)

[**Observación del alumno:** 7](#_Toc11869245)

# **Ejemplo de uso de la cláusula FOR. Obtener todos los títulos de los libros del fichero.**

****

**Consulta XQuery Resultado Consulta**

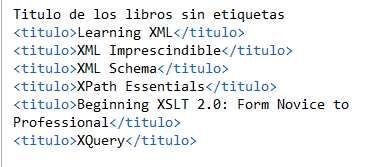
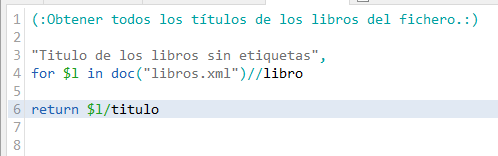
****

El funcionamiento de nuestra consulta es el siguiente: realizamos un **bucle for**, en el cual la **variable $l** (color Rojo) tomara como valor cada uno de los **nodos de libro** (color amarillo) del documento **libros.xml** (color verde).

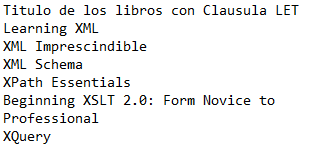
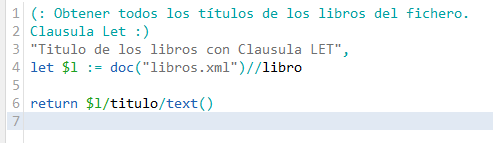
Cada uno de los nodos los retornaremos con la **variable $l** en cada una de las tuplas vinculadas a la misma.

Con la línea **$l/titulo/text()** declaramos que nos retorne dentro de la tupla guardada en la variable $l (libro) el título en formato texto **( text() ).**

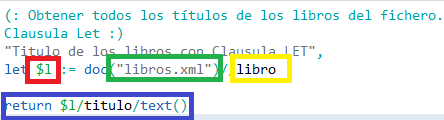
De no especificar ese último dato nos saldría el resultado con etiquetas como muestra la siguiente imagen:

****

# **Ejemplo de uso de la cláusula LET. Obtener todos los títulos de los libros del fichero.**

****

**Consulta XQuery Resultado Consulta**

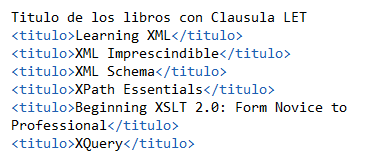
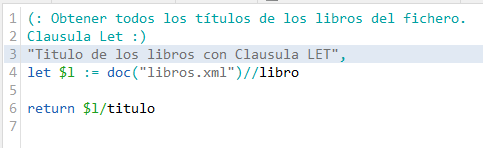
****

El funcionamiento de nuestra consulta es el siguiente: con cláusula **LET** asignamos a la **variable $l** (color Rojo) el valor cada uno de los **nodos de libro** (color amarillo) del documento **libros.xml** (color verde).

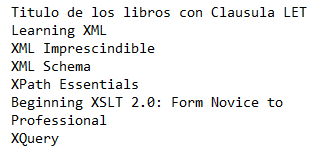
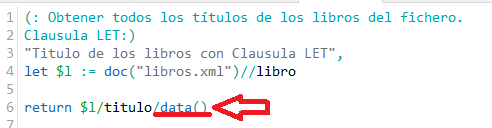
Cada uno de los nodos los retornaremos con la **variable $l** que previamente hemos asignado con nuestra clausula **LET** en cada una de las tuplas vinculadas a la misma.

Con la línea **$l/titulo/text()** declaramos que nos retorne dentro de la tupla guardada en la variable $l (libro) el título en formato texto **( text() ).**

De no especificar ese último dato nos saldría el resultado con etiquetas como muestra la siguiente imagen:

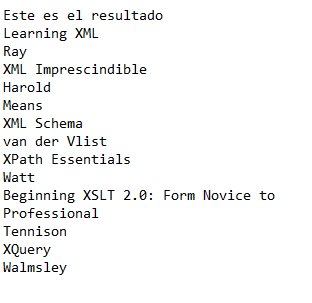
****

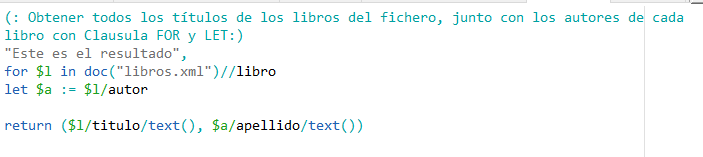
Se visualiza el resultado en vez con el **formato text()** con el **formato data()** siendo visible también:

****

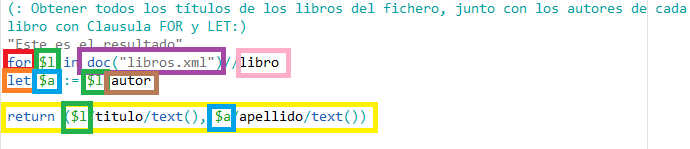
**Consulta XQuery Resultado Consulta**

# **Ejemplo de uso de la cláusula FOR y LET juntas. Obtener todos los títulos de los libros del fichero, junto con los autores de cada libro.**

****

****

**Consulta XQuery Resultado Consulta**

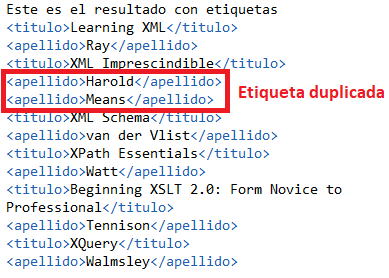
****

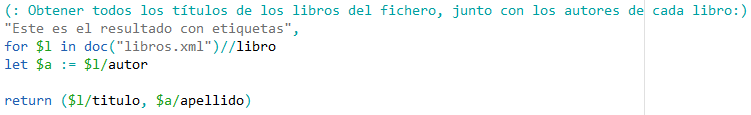
En esta consulta tenemos un **clausula for** la cual la asignamos en la **variable $l (libro)**el valor del nodo libro, del **archivo libros.xml**

Después tenemos una **cláusula Let** con un **variable $a** con la cual asignaremos en cada **bucle for** el valor de la tupla de la variable anterior **$l (libro) mas /autor**.

Después retornaremos por medio de la **variable $l (libro) /** titulo (el título de ese nodo de libros) seguido de la$a **(libro/autor) /**apellido el apellido de la misma tupla en la que nos encontramos. En formato texto en ambas sin etiquetas

En esta consulta seria mas visible los resultados obtenidos con las etiquetas de los mismos, con lo que eliminamos los formatos **Text()** de nuestra consulta:

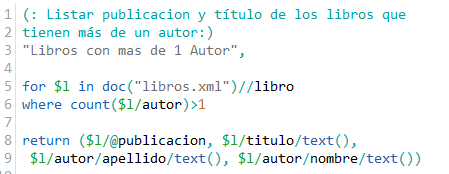
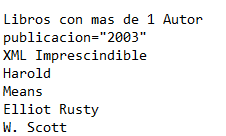




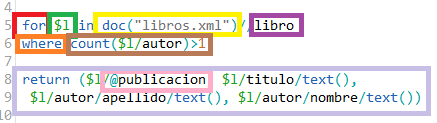
**Consulta XQuery Resultado Consulta**

De esta manera podemos observar que por un mismo titulo nos ha sacado dos apellidos distintos debido a que ese libro tiene dos autores distintos. De la cual en la otra consulta no quedaba tan claro

# **Listar publicación y título de los libros que tienen más de un autor.**

****

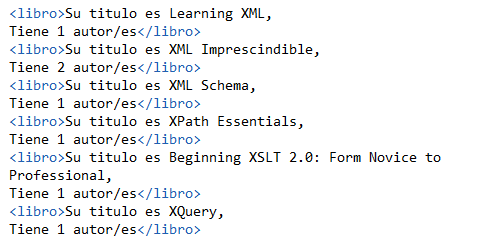
**Consulta XQuery Resultado Consulta**

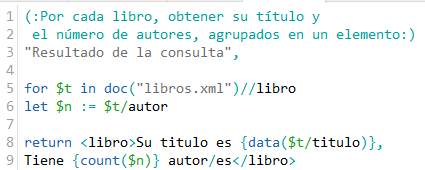
****

Esta consulta la realizamos por medio de un **clausula for** almacenándola en la **variable $l** del **archivo libros.xml** los **nodos de libro.** Con la **cláusula Where** declaramos que solo almacén las tuplas que cumplan la siguiente condición, la cual en este caso es que **cuente los autores** por tupla y solo **almacene los que sean mayores de 1 autor**.

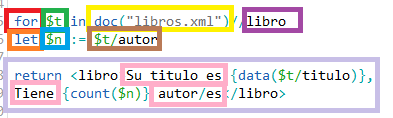
En el mensaje de retorno podemos observar también que mostramos un **atributo** perteneciente a la **etiqueta libro ( $l** ), con el **símbolo @ seguido del nombre** del atributo. **Mostraremos también el título, apellido del autor y nombre del autor, todos ellos en formato texto**

# **Por cada libro, obtener su título y el número de autores, agrupados en un elemento**

****

****

**Consulta XQuery Resultado Consulta**

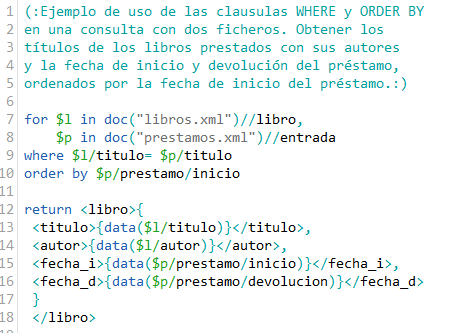
****

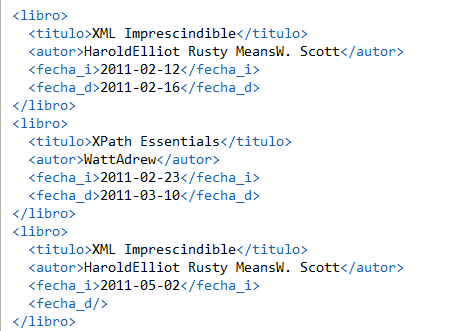
Esta consulta realizamos por medio de la **clausula For** un bucle el cual almacenamos en la **variable $t** de los **nodos libro** correspondiente al **archivo libros.xml**

Después por medio de la **cláusula Let** asignamos en la **variable $n** los datos correspondientes a la tupla de la **variable $t (libro) / autor**

Esta forma de retornar la consulta es mas visible ya **incluyendo un texto** podemos realizar una descripción de lo que es cada dato que nos va a devolver la consulta. En este caso nos devolver **el titulo y el numero de autores por libro.** Como observamos el resultado es más fácil de interpretar, en consultas anteriores para saber a que pertenece los datos incluíamos la etiqueta de la misma eliminando el formato Text() de la misma.

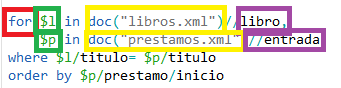
# **Ejemplo de uso de las cláusulas WHERE y ORDER BY en una consulta con dos ficheros. Obtener los títulos de los libros prestados con sus autores y la fecha de inicio y devolución del préstamo, ordenados por la fecha de inicio del préstamo.**

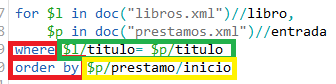
****

****

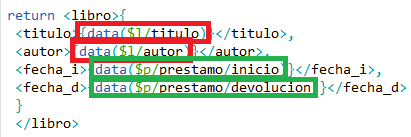
En esta consulta por medio de una **clausula For** vamos a recorrer dos archivos .xml y vamos a incluir en dos variables distintas dos nodos distintos. Después por medio de la **cláusula Where** vamos a realizar una comparación de dos datos de los nodos incluidos en nuestras variables. Una vez realizado la comparación las vamos a ordenar por medio de la cláusula **Order By** de un valor que vamos a establecer nosotros.

En vamos a retornar una serie de datos, correspondientes a nuestras variables y las vamos a encapsular en unas etiquetas para que la consulta se mas reconocible.

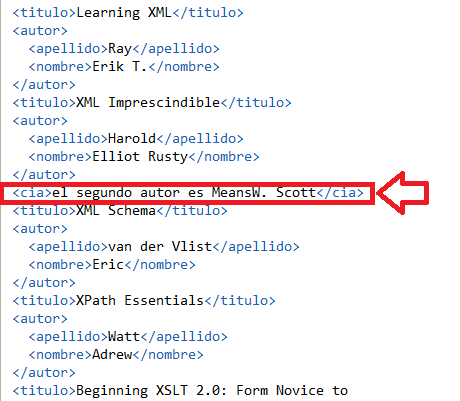
Realizamos unas **clausula For**, por la cual vamos a recorrer dos archivos. En el **bucle For** que realizamos en el archivo **libros.xml** declaramos una **variable $l** en la cual vamos a incluir los **nodos libro**. En el otro bucle que estamos realizamos en el archivo **prestamos.xml** declaramos una **variable $p** en la que incluimos los **nodos correspondientes a entrada**

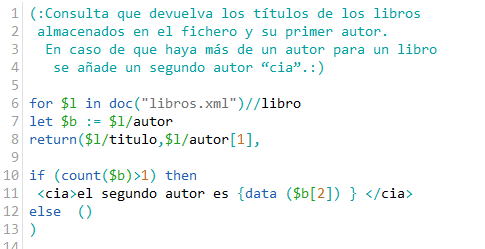
Tras realizar el paso anterior tenemos un clausula **Where** que vamos a guardar solo los nodos en los cuales tengan el mismo titulo tanto la tupla perteneciente **a libro ( $l )** como la perteneciente a **entrada ($p ).**   
Una vez realizado el filtro anterior las vamos a ordenar

por medio de la cláusula **Order By** y con los datos de **$p (entrada) / préstamo /inicio**

Los datos que nos va a retornar los tenemos entre etiquetas para una mayor claridad de los mismo. Como podemos observar solicitarnos datos pertenecientes a la **variable libro ($l)[archivo libros.xml]** y **variable entrada ($p)[archivo prestamos.xml]** 

# **Ejemplo, hacer una consulta que devuelva los títulos de los libros almacenados en el fichero y su primer autor. En caso de que haya más de un autor para un libro se añade un segundo autor “cia”.**

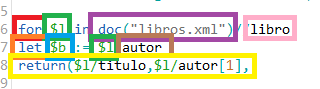




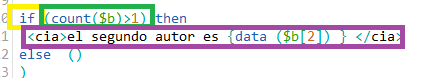
En esta consulta al igual que en el punto 3, en esta consulta tenemos un **clausula For** la cual la asignamos en la **variable $l (libro)**el valor del nodo libro, del **archivo libros.xml**

Después tenemos una **cláusula Let** con un **variable $b** con la cual asignaremos en cada **bucle For** el valor de la tupla de la variable anterior **$l (libro) mas /autor**.

Después retornaremos por medio de la **variable $l (libro) /** titulo (el título de ese nodo de libros) seguido de la **$b (libro/autor)** de la misma tupla en la que nos encontramos. Tenemos al igual que una Array [1] para especificar que utilicemos este formato en el **autor [1]**



Por medio de una **cláusula if** y de una comparativa de **tuplas de autor mayor a 1,** en el caso de que **encuentre más de 1 autor** **mostrara también los siguientes autores con la etiqueta <cia></cia>**



# **Observación del alumno:** Una vez realizada la actividad de consulta por medio del Software se observa que es muy similar a la realización de consultas en un base de datos por medio de una aplicación destinado a ello como SqlDeveloper, con la peculiaridad de que trabajamos con las etiquetas y las podemos mostrar para su fácil interpretación