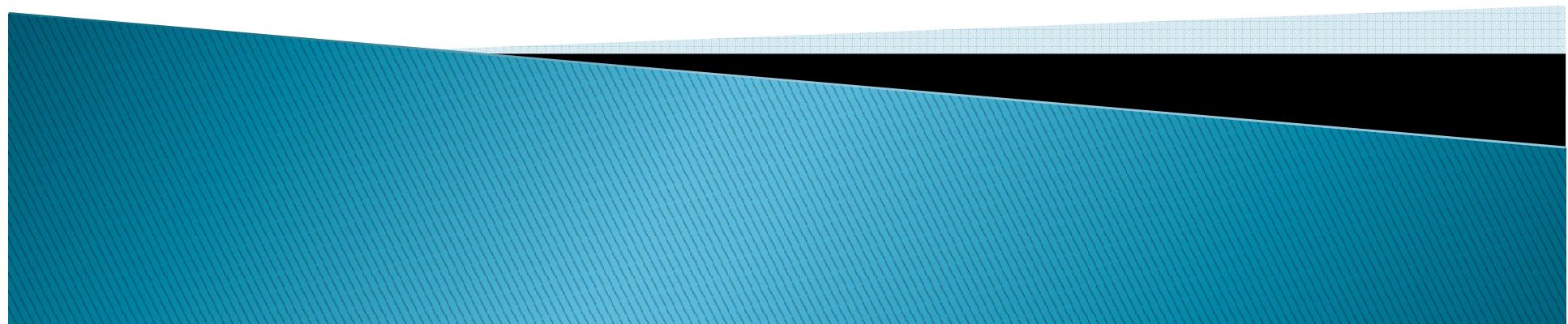


XQUERY

LENGUAJE DE MARCAS



LENGUAJE DE MARCAS XQuery.-teoria

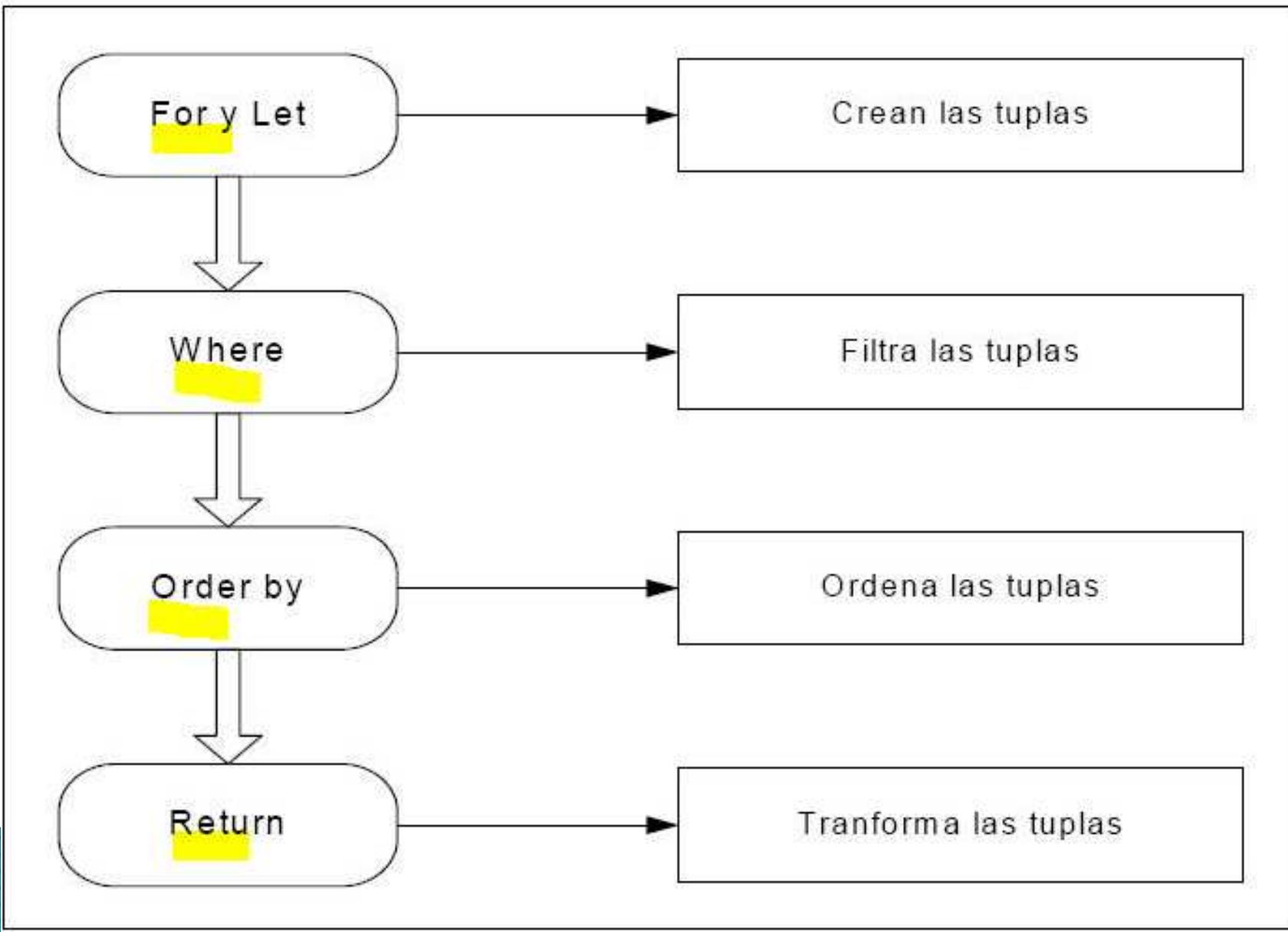
En XQuery las consultas pueden estar compuestas por cláusulas de hasta cinco tipos distintos. Las consultas siguen la norma FLWOR (leído como flower), siendo FLWOR las siglas de For, Let, Where, Order y Return. A continuación, en la tabla 1, se describe la función de cada bloque:

For	Vincula una o más variables a expresiones escritas en XPath, creando un flujo de tuplas en el que cada tupla está vinculada a una de las variables.
Let	Vincula una variable al resultado completo de una expresión añadiendo esos vínculos a las tuplas generadas por una cláusula for o, si no existe ninguna cláusula for, creando una única tupla que contenga esos vínculos.
Where	Filtrá las tuplas eliminando todos los valores que no cumplen las condiciones dadas.
Order by	Ordena las tuplas según el criterio dado.
Return	Construye el resultado de la consulta para una tupla dada, después de haber sido filtrada por la cláusula where y ordenada por la cláusula order by.

Tabla 1. Posibles cláusulas en una consulta XQuery.

En XQuery, cuando usamos el término tupla, nos estamos refiriendo a cada uno de los valores que toma una variable.

LENGUAJE DE MARCAS XQuery.-teoria



LENGUAJE DE MARCAS XQuery.-teoria

Entrada (alumnos.xml)

```
<alumnos>
  <alumno dni="93940">
    <nombre>Jose</nombre>
    <apells>Bernardo</apells>
    <nota>7</nota>
  </alumno>
  <alumno dni="93940">
    <nombre>Juan</nombre>
    <apells>López</apells>
    <nota>4</nota>
  </alumno>
</alumnos>
```

Consulta (XQuery)

```
for $a in doc("alumnos.xml")//alumno
  where $a/nota > 5
return
<aprobado>{
  $a/@dni,
  $a/nota
}</aprobado>
```

Resultado

```
<aprobado dni="93940">
  <nota>7</nota>
</aprobado>
```

LENGUAJE DE MARCAS XQuery.-teoria

Permite iterar sobre una secuencia de valores, ligando una variable a cada valor de la secuencia y evaluando una expresión por cada valor de la variable

```
for $n in (1 to 4)
return $n * $n
```



Salida

```
1 4 9 16
```

LENGUAJE DE MARCAS XQuery.-teoria

Ejemplo: Obtener nombres de alumnos cuya nota está entre 6 y 8

```
for $a in doc("alumnos.xml")//alumno  
where $a/nota > 6 and $a/nota < 8  
return $a/nombre
```

Mas ejemplos: obtén los nombres de los alumnos ordenados

- por el campo apellido , en el 1^{er} y
- por el dni en el 2º

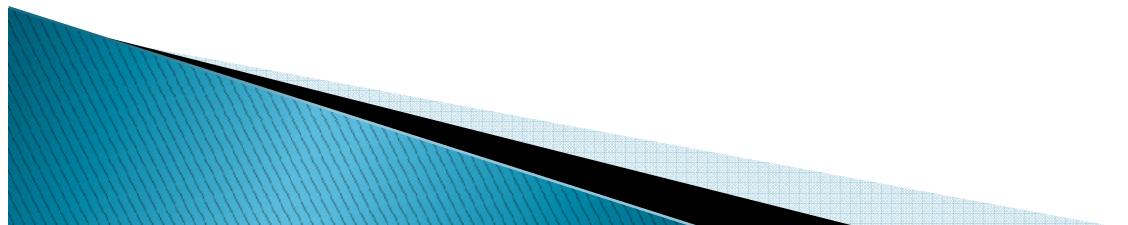
```
for $a in doc("alumnos.xml")//alumno  
order by $a/apellidos ascending  
return $a/nombre
```

```
for $a in doc("alumnos.xml")//alumno  
order by number($a/@dni) ascending  
return $a/nombre
```

LENGUAJE DE MARCAS XQuery.-teoria

Para cada alumno, muestra los nombres de los alumnos y un mensaje de aprobado o suspendido según el valor de si nota

```
for $a in doc("alumnos.xml")//alumno
return
<alumno>
{$a/nombre}
<nota>
{if ($a/nota > 5) then "Aprobado"
else "Suspensos" }
</nota>
</alumno>
```



LENGUAJE DE MARCAS XQuery.-teoria

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bib>
    <libro año="1994">
        <titulo>TCP/IP Illustrated</titulo>
        <autor>
            <apellido>Stevens</apellido>
            <nombre>W.</nombre>
        </autor>
        <editorial>Addison-Wesley</editorial>
        <precio> 65.95</precio>
    </libro>

    <libro año="1992">
        <titulo>Advan Programming for Unix environment</titulo>
        <autor>
            <apellido>Stevens</apellido>
            <nombre>W.</nombre>
        </autor>
        <editorial>Addison-Wesley</editorial>
        <precio>65.95</precio>
    </libro>

    <libro año="2000">
        <titulo>Data on the Web</titulo>
        <autor>
            <apellido>Abiteboul</apellido>
            <nombre>Serge</nombre>
        </autor>
        <autor>
            <apellido>Buneman</apellido>
            <nombre>Peter</nombre>
        </autor>
        <autor>
            <apellido>Suciu</apellido>
            <nombre>Dan</nombre>
        </autor>
        <editorial>Morgan Kaufmann editorials</editorial>
        <precio>39.95</precio>
    </libro>

```

```

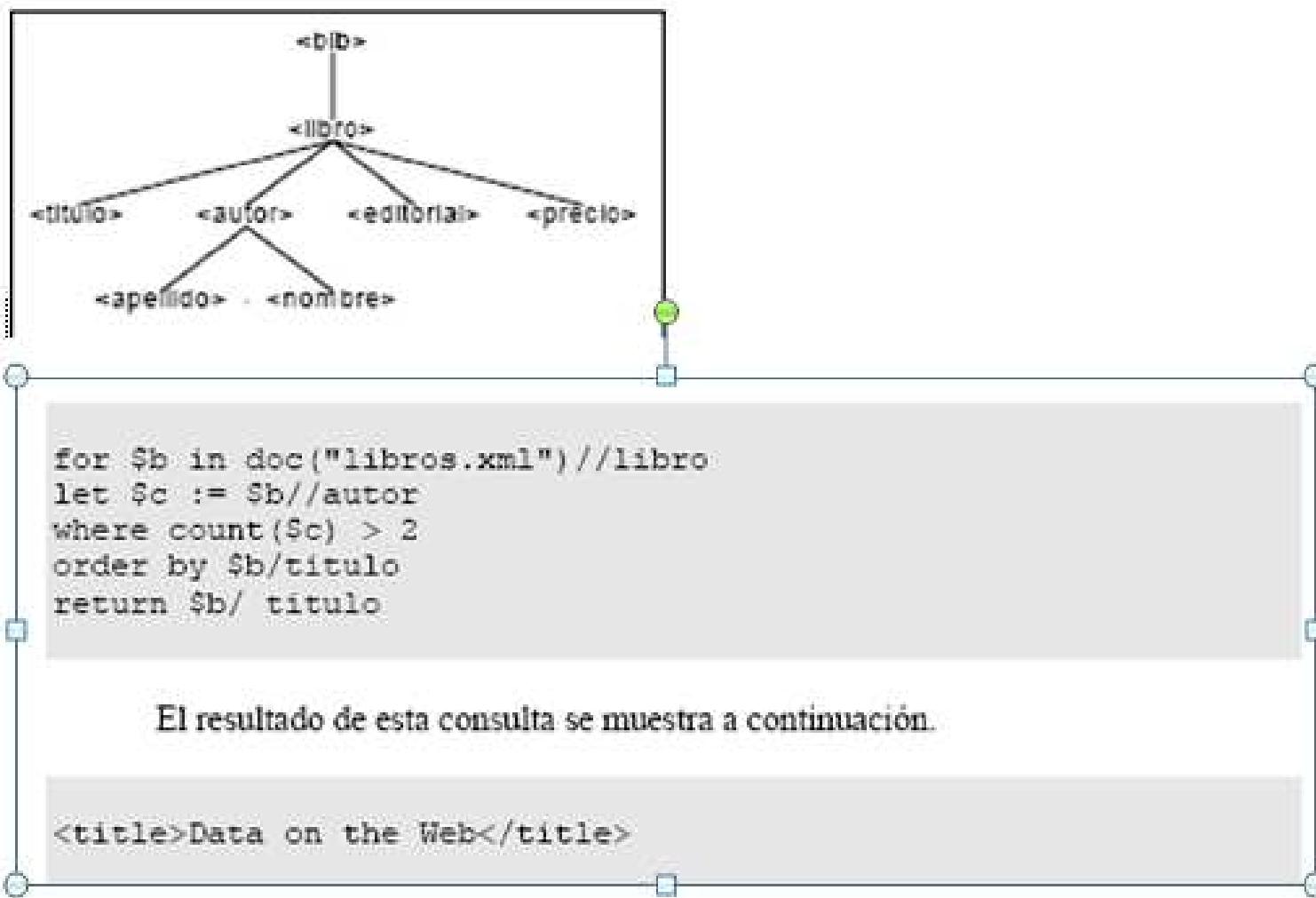
    <libro año="1999">
        <titulo> Economics of Technology for Digital TV</titulo>
        <editor>
            <apellido>Gerbarg</apellido>
            <nombre>Darcy</nombre>
            <afiliacion>CITI</afiliacion>
        </editor>
        <editorial>Kluwer Academic editorials</editorial>
        <precio>129.95</precio>
    </libro>

```

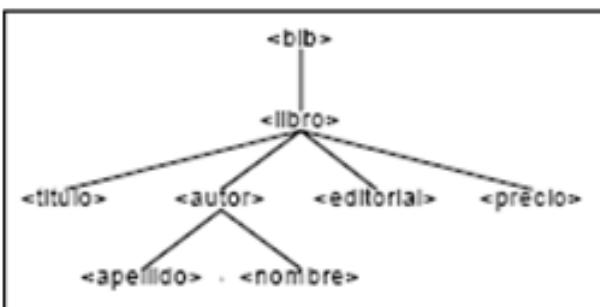
```

</bib>
```

LENGUAJE DE MARCAS XQuery.-teoria



LENGUAJE DE MARCAS XQuery.-teoria



```
for $b in doc("libros.xml")//libro
where $b/@año = "2000"
return $b/titulo
```

El resultado de la consulta anterior se muestra a continuación.

```
<title>Data on the Web</title>
```

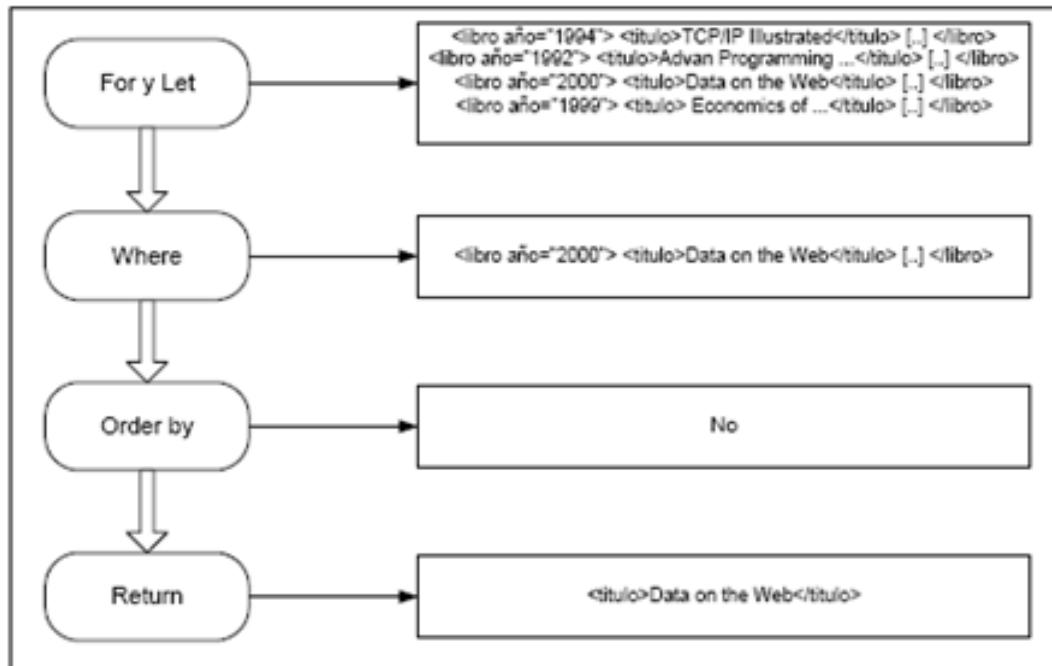
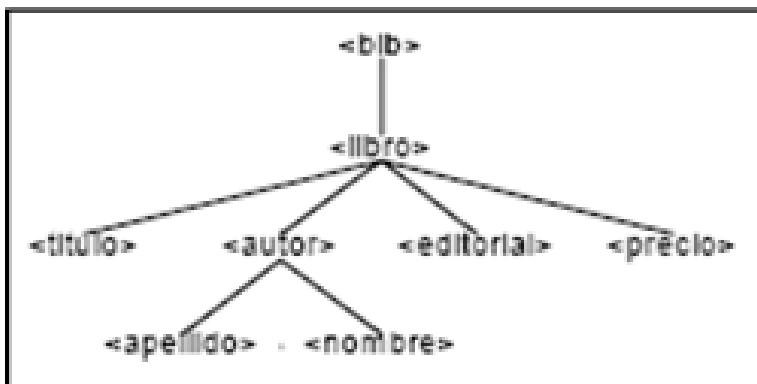


Figura 3. Ejecución de una consulta XQuery con los resultados de cada cláusula.

LENGUAJE DE MARCAS XQuery.-teoria



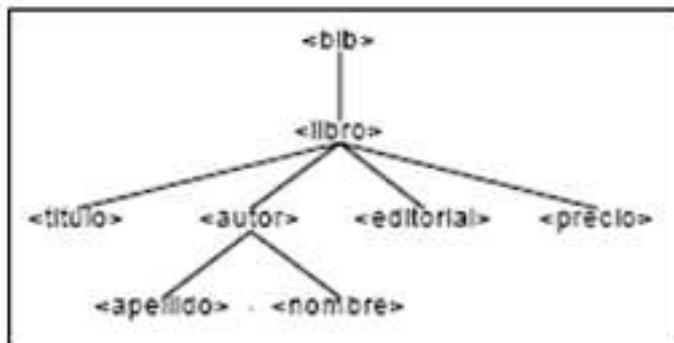
Escribir una consulta que obtenga el título de los libros cuyo precio esté por debajo de 50.00€.

```
for $b in doc("libros.xml")//libro  
where $b/precio < 50.00  
return $b/titulo
```

El resultado de esta consulta se muestra a continuación.

```
<titulo>Data on the Web</titulo>
```

LENGUAJE DE MARCAS XQuery.-teoria

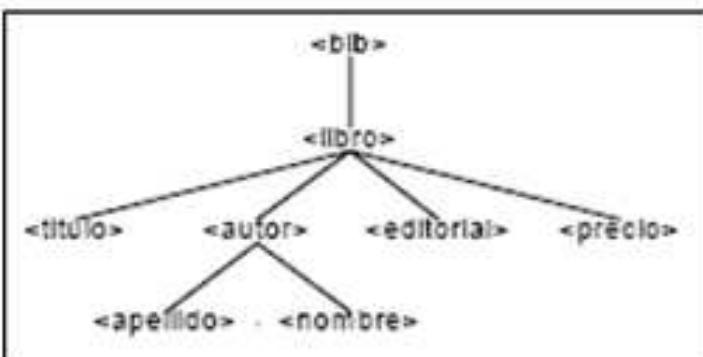


```
for $d in doc("libros.xml")/bib/libro/titulo  
return  
    <titulos>{ $d }</titulos>
```

El resultado de esta consulta se muestra a continuación:

```
<titulos>  
    <titulo>TCP/IP Illustrated</titulo>  
</titulos>  
<titulos>  
    <titulo>Advan Programming for Unix environment</titulo>  
</titulos>  
<titulos>  
    <titulo>Data on the Web</titulo>  
</titulos>  
<titulos>  
    <titulo> Economics of Technology for Digital TV</titulo>  
</titulos>
```

LENGUAJE DE MARCAS XQuery.-teoria



A continuación repetimos la misma consulta sustituyendo la cláusula `for` una cláusula `let`.

```
let $d := doc("libros.xml")/bib/libro/titulo  
return  
<titulos>{ $d }</titulos>
```

El resultado de esta consulta se muestra a continuación.

```
<titulos>  
  <titulo>TCP/IP Illustrated</titulo>  
  <titulo>Advanced Programming for Unix environment</titulo>  
  <titulo>Data on the Web</titulo>  
  <titulo>Economics of Technology for Digital TV</titulo>  
</titulos>
```

LENGUAJE DE MARCAS XQuery.-teoria

. Operadores y funciones principales.

El conjunto de funciones y operadores soportado por XQuery 1.0 es el mismo conjunto de funciones y operadores utilizado en XPath 2.0 y XSLT 2.0.

XQuery soporta operadores y funciones matemáticas, de cadenas, para el tratamiento de expresiones regulares, comparaciones de fechas y horas, manipulación de nodos XML, manipulación de secuencias, comprobación y conversión de tipos y lógica booleana. Además permite definir funciones propias y funciones dependientes del entorno de ejecución del motor XQuery. Los operadores y funciones más importantes se muestran en la tabla 2 y se detallan a lo largo de este apartado.

Matemáticos:	+ , - , * , div(*), idiv(*), mod.
Comparación:	=, !=, <, >, <=, >=, not()
Secuencia:	union (), intersect, except
Redondeo:	floor(), ceiling(), round().
Funciones de agrupación:	count(), min(), max(), avg(), sum().
Funciones de cadena:	concat(), string-length(), starts-with(), ends-with(), substring(), upper-case(), lower-case(), string()
Uso general:	distinct-values(), empty(), exists()

LENGUAJE DE MARCAS XQuery.-teoria

... Comentarios.

Los comentarios en XQuery, a diferencia de XML, van encerrados entre caras sonrientes, tal y como se muestra a continuación.

```
(: Esto es un comentario, y parece muy feliz :)
```

