**Práctica 5 Aprendiendo Xquery biblioteca**

**Creamos la colección biblioteca y subimos los ficheros libros.xml y comentarios.xml**



<?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>

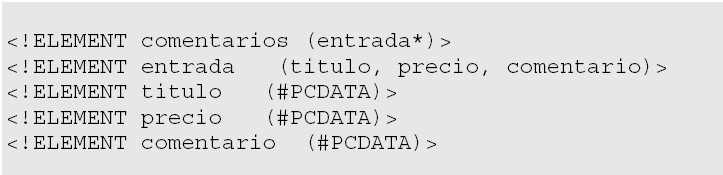
<editorial>Kluwer Academic editorials</editorial>

<precio>129.95</precio>

</libro>

</bib>

En algunas consultas también necesitaremos combinar la información contenida en el archivo “libros.xml” con la información contenida en el archivo “comentarios.xml”. El contenido de este archivo se muestra a continuación. A continuación se muestra el contenido del DTD correspondiente al archivo "comentarios.xml".



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

<comentarios>

<entrada>

<titulo>Data on the Web</titulo>

<precio>34.95</precio>

<comentario>

Un libro muy bueno sobre bases de datos.

</comentario>

</entrada>

<entrada>

<titulo>Advanced Programming in the Unix environment</titulo> <precio>65.95</precio>

<comentario>

Un libro claro y detallado de programación en UNIX.

</comentario>

</entrada>

<entrada>

<titulo>TCP/IP Illustrated</titulo>

<precio>65.95</precio>

<comentario>

Uno de los mejores libros de TCP/IP

</comentario>

</entrada> </comentarios>

# Lo subimos a la base de datos Exists



# CONSULTAS (da un pantallazo de la salida)

1. **Obtener todos los nodos de la biblioteca** //bib

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **Obtener todos los libros**

//bib/libro

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **Sacar los libros de precio <40** for $x in//bib/libro where $x/precio<40 return $x

Texto

Descripción generada automáticamente

1. **Sacar los títulos de los libros de precio >40.**

for $x in//bib/libro where $x/precio>40

return $x/titulo

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. **Ordena la consulta anterior por titulo** for $x in//bib/libro where $x/precio>40 order by $x/titulo

return $x/titulo

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. **Obtener los títulos según su año, si son del año 2000 visualizarlos entre las etiquetas sigloXXI sino entre las eitquetas sigloXX.**

for $x in//bib/libro return if($x/@año="2000") then <sigloXXI>{data($x/titulo)}</sigloXXI>

else <sigloXX>{data($x/titulo)}</sigloXX>

Captura de pantalla de un celular

Descripción generada automáticamente

1. **Obtener el nombre y el año de todos los libros publicados por AddisonWesley después de 1991.** for $b in //bib/libro

where $b/editorial= "Addison-Wesley" and $b/@año>1991 return <libro año="{$b/@año}"> {$b/titulo}</libro>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **Poner la salida anterior en un Html**

<html>

<body>

<h1>bib</h1>

<ul>

{

for $b in //bib/libro

where $b/editorial= "Addison-Wesley" and $b/@año>1991 return <libro año="{$b/@año}"> {$b/titulo}</libro>

}

</ul>

</body>

</html>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. **Podemos poner un intervalo en la cláusula for (1 to 5) probar**

for $x in (1 to 5) return <test>{$x}</test>

Texto, Carta

Descripción generada automáticamente

1. **La palabra reservada at permite contar iteraciones, listar los libros numerándolos.**

for $b at $i in //bib/libro/titulo

return <libro> {$i}.{data($b)}</libro>

Captura de pantalla de un celular

Descripción generada automáticamente

1. **Cuando hay más de una expresión en la cláusula for se separa por comas. Probar esta expresión:**

for $x in (10,20), $y in (100,200)

return <test> x={$x} and y={$y}</test>

Imagen de la pantalla de un celular de un mensaje en letras blancas

Descripción generada automáticamente con confianza baja

1. **Consulta devuelve los títulos de los libros que tengan más de dos autores ordenados por su título.**

for $b in //libro let $c := $b//autor where count($c) > 2 order by $b/titulo

return $b/ titulo



1. **Consulta devuelve los títulos de los libros del año 2.000. Como “año” es un atributo y no una etiqueta se le antecede con un carácter “@”.**

for $b in //libro

where $b/@año = "2000" return $b/titulo



1. **Diferencias entre las cláusulas for y let. Compara** for $d in /bib/libro/titulo return

<titulos>{ $d }</titulos>

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

let $d := /bib/libro/titulo return

<titulos>{ $d }</titulos>

Captura de pantalla de un celular

Descripción generada automáticamente

1. **Si una cláusula let aparece en una consulta que ya posee una o más cláusulas for, los valores de la variable vinculada por la cláusula let se añaden a cada una de las tuplas generadas por la cláusula for. Un ejemplo se muestra en la siguiente consulta:** Esta consulta devuelve el título de cada uno de los libros de archivo "libros.xml"junto con el número de autores de cada libro.

for $b in //libro let $c := $b/autor return

<libro>{ $b/titulo, <autores>{ count($c) }</autores>}</libro>

Imagen de la pantalla de un celular con texto

Descripción generada automáticamente con confianza media

1. **Expresiones condicionales. Consulta que devuelva los títulos de todos los libros almacenados en el archivo libros.xml y sus dos primeros autores, en el caso que existan más de dos autores para un mismo libro se añade un tercer autor “ et. al”**

for $b in //libro return <libro>

{ $b/titulo }

{

for $a at $i in $b/autor where $i <= 2 return <autor>{data($a)}</autor>

} { if (count($b/autor) > 2)

then <autor>et al.</autor>

else () }

</libro>

Imagen de la pantalla de un celular con texto

Descripción generada automáticamente con confianza media

1. **Cuantificadores existenciales:** some”,“every” y ”satisfies”

**Consulta devuelva todos los títulos de los libros en los que todos los autores de cada libro sean W. Stevens**

for $b in //libro where every $a in $b/autor satisfies ($a/last="Stevens" and $a/first="W.") return $b/titulo

El último título devuelto como resultado de la consulta es un libro que no tiene autores. Cuando un cuantificador universal se aplica sobre un nodo vacío, siempre devuelve cierto, como se ve en el ejemplo anterior.

1. El operador de sustracción (except) recibe dos secuencias de nodos como operandos y devuelve una secuencia conteniendo todos los nodos del primer operando que no aparezcan en el segundo operando. **Consulta que usa el operador sustracción para obtener un nodo libro con todos sus nodos hijos salvo el nodo <precio>.**

for $b in //libro where $b/titulo = "TCP/IP Illustrated" return <libro>

{ $b/@\* }

{ $b/\* except $b/precio }

</libro>

Captura de pantalla de un celular

Descripción generada automáticamente

La función **distinct-values()** extrae los valores de una secuencia de nodos y crea una nueva secuencia con valores únicos, eliminando los nodos duplicados. **Consulta devuelva todos los apellidos distintos de los autores**.

for $l in distinct-values(//bib/libro/autor/apellido)

return <apellidos>{ $l }</apellidos>

Imagen de la pantalla de un celular de un mensaje en letras blancas

Descripción generada automáticamente con confianza baja

1. La función empty() devuelve cierto cuando la expresión entre paréntesis está vacía.

**Consulta que devuelve todos los nodos libro que tengan al menos un nodo autor.**

for $b in //libro where not(empty($b/autor))

return $b

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. La función opuesta a empty() es exists(), la cual devuelve cierto cuando una secuencia contiene, al menos, un elemento.

**Por ejemplo, como la consulta anterior tiene una cláusula where que comprueba una negación sobre empty(), podemos rescribirla usando la función exists()**

for $b in //libro where exists($b/autor) return $b

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente