

LENGUAJES DE RECUPERACIÓN PARA LA WEB

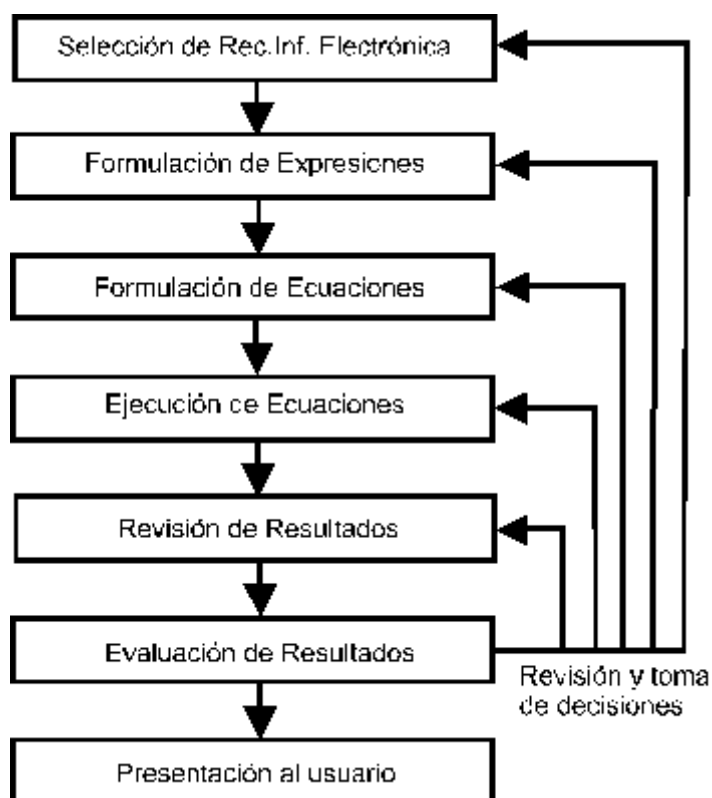
Introducción.....	1
XML-Query	4
XQL.....	9
Tolog.....	12

Introducción

Ya desde la antigüedad, existía la necesidad de desarrollar métodos y técnicas que no sólo permitieran el control y la conservación de las diferentes colecciones de información, sino también la identificación de las mismas con el fin de poder recuperarlas posteriormente de forma correcta.

Actualmente, los sistemas de búsqueda han adquirido una gran importancia en el uso cotidiano de los ordenadores. Sin embargo, la recuperación de información textual lleva consigo asociada una serie de problemas, algunos de los cuales provienen de las características propias del lenguaje natural al resultar éste ambiguo, que todavía no han sido resueltos de forma satisfactoria. Por todo esto, el uso apropiado de lenguajes de recuperación como los que veremos en esta Web es importante. De esta forma, se está viendo un mecanismo de recuperación y organización de la información.

El proceso de recuperación en un entorno informático genérico sería de la siguiente manera:



En general, se puede decir que la recuperación de información se basa en la localización y búsqueda de información puntualmente, y se caracteriza por la utilización de un lenguaje de recuperación que permite comunicarse con la base de datos correspondiente, que es donde se encuentran los documentos que pueden resultar relevantes para el usuario que hizo la búsqueda.

Por tanto, se puede definir un lenguaje de interrogación de bases de datos como un conjunto de órdenes, operadores y estructuras que, organizados según unas normas lógicas, permiten la consulta de fuentes y recursos de información electrónica. El resultado de la combinación de estos elementos, siguiendo las normas establecidas, es una expresión, a la que se identifica con el nombre de ecuación, capaz de interrogar el contenido de la fuente de información con la finalidad de extraer de la base de datos aquellos documentos que cumplan los requisitos establecidos. La definición mínima de un lenguaje de interrogación y de sus componentes puede encontrarse en el borrador de norma ISO 8777-1988.

El objeto que persigue esta Web es dar a conocer cómo la recuperación y organización de la información se hace posible, para lo cual nos centramos en conocer varios

lenguajes de recuperación. En particular, se incluyen mini-tutoriales acerca de los siguientes:

Lenguajes de recuperación en XML:

- XML-Query
- XQL

Lenguajes de recuperación en topic-maps:

- Tolog

Por último cabe destacar que en estas páginas elaboradas por mí sólo se tratan los anteriores lenguajes de recuperación citados, pero existen muchos otros, incluso algunos de los cuáles (SeRQL y SPARQL) también se habla de ellos dentro de lo que es esta WIKI.

A continuación se indican las referencias que han resultado más útiles para recabar la información que reside en este sitio web:

- Tutorial de XML-Query: <http://www.w3schools.com/xquery/default.asp>
- Definición de XQL:
http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci213712,00.html
- Tutorial de XQL: http://en.wikibooks.org/wiki/XML_-_Managing_Data_Exchange/XQL#XQL_Tutorial
- FAQ de XQL: <http://www.ibiblio.org/xql/>
- Tutorial de Tolog : <http://www.ontopia.net/topicmaps/materials/tolog.html#N65>

XML-Query

XQuery, también conocido como XML Query, es un lenguaje para encontrar y extraer elementos y atributos de documentos XML. La mejor forma de entender este lenguaje es diciendo que XQuery es para XML lo que SQL es para las bases de datos. Por ejemplo, una pregunta que XQuery podría resolver sería la siguiente: Seleccionar todos los libros con un precio menor a 20 euros de la colección de libros almacenada en un documento XML llamado catalogo.xml.

XQuery es supervisado por W3C (World Wide Web Consortium), por lo que es compatible con XML, Namespaces, XSLT, XPath y XML Schema.

XQuery es soportado por los principales motores de bases de datos (IBM, Oracle, Microsoft, etc.).

XQuery hace uso de XPath, un lenguaje utilizado para seleccionar partes de XML. De hecho, XQuery 1.0 y XPath 2.0 comparten el mismo modelo de datos y soportan las mismas funciones y operadores. Por ello, si ya conoces XPath, no tendrás problemas en comprender XQuery.

Para entender mejor el lenguaje de recuperación XQuery, a continuación, se muestra un ejemplo, basándonos en el siguiente documento:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<libros>
</libros>
</libro>
<libro tema="Humanidades">
<titulo ISBN="8434453096">La vida eterna</titulo>
<autor>Fernando Savater</autor>
<fecha>2007</fecha>
<precio>17.50</precio>
</libro>
<libro tema="Literatura">
```

```

<titulo ISBN="8476697635">Los irlandeses</titulo>
<autor>Rodolfo Walsh</autor>
<fecha>2007</fecha>
<precio>16</precio>
</libro>
<libro tema="Gastronomía">
<titulo ISBN="8496434559">Galicia en el paladar</titulo>
<autor>Cristóbal Ramírez</autor>
<autor>Miguel Pazos</autor>
<autor>Mantel Xose Santos Díaz</autor>
<fecha>2007</fecha>
<precio>24.90</precio>
</libro>
</libro>
<libro tema="Artes">
<titulo ISBN="8496445550">Los tesoros de Rembrandt</titulo>
<autor>Michiel Roscam Abbing</autor>
<fecha>2006</fecha>
<precio>35</precio>
</libro>
</libros>

```

XQuery utiliza funciones para extraer información de un documento XML. La siguiente función permite abrir el fichero.

```
doc("catalogo.xml")
```

XQuery utiliza expresiones para navegar a través de los elementos de un documento XML. La siguiente expresión sirve para seleccionar el título de todos los libros.

```
doc("catalogo.xml")/libros/libro/titulo
```

El resultado sería:

```
<titulo ISBN="8434453096">La vida eterna</titulo>
<titulo ISBN="8476697635">Los irlandeses</titulo>
<titulo ISBN="8496434559">Galicia en el paladar</titulo>
<titulo ISBN="8496445550">Los tesoros de Rembrandt</titulo>
```

XQuery usa predicados para limitar la información extraída del documento XML. El siguiente predicado sirve para seleccionar aquellos libros cuyo precio sea menor a 20 euros.

```
doc("catalogo.xml")/libros/libro[precio<20]
```

El resultado sería:

```
</libro>
<libro tema="Humanidades">
<titulo ISBN="8434453096">La vida eterna</titulo>
<autor>Fernando Savater</autor>
<fecha>2007</fecha>
<precio>17.50</precio>
</libro>
<libro tema="Literatura">
<titulo ISBN="8476697635">Los irlandeses</titulo>
<autor>Rodolfo Walsh</autor>
<fecha>2007</fecha>
<precio>16</precio>
</libro>
```

Las expresiones FLWOR (For, Let, Where, Order by, Return) son en XQuery equivalentes a las distintas cláusulas usadas en SQL. Usando FLWOR podemos obtener el título de cada libro cuyo precio sea mayor que 20 euros y ordenada alfabéticamente:

```
for $x in doc("catalogo.xml")/libros/libro
where $x/precio>20
order by $x/titulo
```

return \$x/titulo

El resultado sería:

```
<titulo ISBN="8496434559">Galicia en el paladar</titulo>  
<titulo ISBN="8496445550">Los tesoros de Rembrandt</titulo>
```

Además, usando FLWOR podemos después representar el resultado en HTML. Por ejemplo, mediante una lista mostrar sólo el título de cada libro y ordenada alfabéticamente:

```
<ul>  
{  
for $x in doc("catalogo.xml")/libros/libro/titulo  
order by $x  
return <li>{data($x)}</li>  
}  
</ul>
```

El resultado sería:

```
<ul>  
<li>Galicia en el paladar</li>  
<li>La vida eterna</li>  
<li>Los irlandeses</li>  
<li>Los tesoros de Rembrandt</li>  
</ul>
```

Para terminar, vamos a hablar acerca del uso de expresiones condicionales en este tipo de lenguaje de recuperación y que también serán útiles para realizar ciertas consultas. El ejemplo siguiente permite diferenciar entre los libros que considero entretenidos y los que no lo son.

```
for $x in doc("catalogo.xml")/libros/libro
```

```
return if ($x/@tema="Literatura")
then <entretenido>{data($x/titulo)}</entretenido>
else <aburrido>{data($x/titulo)}</aburrido>
```

Esto daría el resultado:

```
<aburrido>La vida eterna</aburrido>
<entretenido>Los irlandeses</entretenido>
<aburrido>Galicia en el paladar</aburrido>
<aburrido>Los tesoros de Rembrandt</aburrido>
```


XQL

XQL (XML Query Language) es un lenguaje de consultas a sistemas de bases de datos y que facilita la recuperación y organización de la información, de tal forma que sirve para hacer consultas contra bases de datos y obtener el resultado en un nuevo documento XML. Por tanto, se trata de SQL incrustado en los documentos XML.

XQL es una manera de localizar y filtrar los elementos (campos de datos) y texto en XML. Está basado en XSL (Extensible Stylesheet Language).

La notación utilizada se centra en una notación de directorios. Por supuesto, a esa notación se le puede aplicar lógica booleana para filtrar los elementos correspondientes. Por ejemplo, "libro/autor" permite seleccionar todos los autores de cada libro, todo ello encontrándose en un determinado contexto (en un documento XML).

Para entender mejor el lenguaje de recuperación XQL, a continuación, se muestra un ejemplo, basándonos en el siguiente documento:

```
<?xml version="1.0"?>
<facturas>
<factura>
<cliente>
Juan Salinas Gracia
</cliente>
<anotacion>
El cliente nos preguntó que garantía le damos en caso de que los productos adquiridos sean defectuosos
</anotacion>
<entradas n="2">
<entrada cantidad="1" precio_total="80.00">
<producto fabricante="ACME" nombre="taladradora" precio="80.00"/>
</entrada>
<entrada cantidad="1" precio_total="50.00">
```

```

<producto fabricante="ACME" nombre="pulidora" precio="50.00"/>
</entrada>
</entradas>
</factura>
<factura>
<cliente>
Almudena Paz Hornos
</cliente>
<entradas n=2>
<entrada cantidad="2" precio_total="32.00">
<producto fabricante="BSA" nombre="linterna" precio="16.00"/>
</entrada>
<entrada cantidad="1" precio_total="13.00">
<producto fabricante="BSA" nombre="bateria" precio="13.00"/>
</entrada>
</entradas>
</factura>
</facturas>

```

Ante la consulta:

```
//cliente
```

El resultado sería:

```

<xql:result>
<cliente>Juan Salinas Gracia</cliente>
<cliente>Almudena Paz Hornos</cliente>
</xql:result>

```

Ante la consulta:

```
//producto [@fabricante="BSA"]
```

El resultado sería:

```
<xql:result>
<producto fabricante="BSA" nombre="linterna" precio="16.00"/>
<producto fabricante="BSA" nombre="bateria" precio="13.00"/>
</xql:result>
```

Ante la consulta:

```
//factura[cliente='Juan Salinas Gracia']//producto
```

El resultado sería:

```
<xql:result>
<producto fabricante="ACME" nombre="taladradora" precio="80.00"/>
<producto fabricante="ACME" nombre="pulidora" precio="50.00"/>
</xql:result>
```

Luego, existe otro lenguaje de recuperación, XIRQL, que representa una extensión de XQL para recuperación de información.

Tolog

Tolog es un lenguaje que facilita la recuperación y organización de la información para topics maps, inspirado por Datalog (un subconjunto de Prolog) y SQL. Un topic map es una estructura abstracta que contiene información de interés para una comunidad de usuarios, de igual forma que sucede en el caso de una base de datos.

Tolog se basa en el uso de la lógica de predicados de primer orden, de tal forma que existe una base de hechos, que es contra la que se realizan las consultas y que, por tanto, se hacen en forma de predicados. En los topic maps la representación se hace mediante asociaciones. Por ejemplo, si tenemos información registrada acerca de libros, podríamos encontrar información del tipo "Los irlandeses es un libro que fue escrito por Rodolfo Walsh" y "Rodolfo Walsh es un escritor". Entonces, se podrían tener predicados tales como:

escrito-por(Los Irlandeses, Rodolfo Walsh)

es-escritor(Rodolfo Walsh)

En el caso de un topic map, podríamos tener expresiones del tipo "Los irlandeses es un libro que fue escrito por Rodolfo Walsh que es un escritor".

En Tolog las consultas son fáciles de realizar, ya que éstas llevarán asociadas unas variables en las que se recojan los resultados de tal consulta. Por ejemplo, ante la siguiente consulta:

escrito-por(\$X, Rodolfo Walsh)

se obtendrían de la base todos los libros escritos por Rodolfo Walsh. Sin embargo, la consulta:

escrito-por(\$X, \$Y)

devolvería todas las posibles asignaciones que hagan esa consulta verdadera. Luego, las consultas se pueden complicar tanto como se quiera, simplemente concatenando consultas.

Otro aspecto importante a destacar acerca de este lenguaje de recuperación son las reglas que se definen en Tolog, las cuales podrán ser consultadas como si de hechos se tratara y que consisten en asociaciones que se desprenden a partir de otros hechos. Por ejemplo:

libros-de-autor-en-fecha(\$X, \$Y, \$Z) :- escrito-por(\$X, \$Y), autor(\$Y), publicado-en(\$X, \$Z)

Con esta regla se podría obtener los libros de un cierto autor publicados en una fecha concreta.

Para terminar, cabe destacar los principales -pros y -contras de Tolog:

Ventajas del lenguaje de recuperación Tolog:

- Es simple y fácil de entender e implementar.
- Permite realizar consultas muy sofisticadas.

Inconvenientes del lenguaje de recuperación Tolog:

- Sólo se pueden realizar consultas basándose en predicados o reglas ya escritos.
- Sólo permite consultar el topic map, no modificarlo.