

## **HOMERJ: THE RELATIONAL XML DATABASE PROJECT**

### **Autores:**

#### **Estudiantes de 5to año de la Licenciatura en Sistemas**

**Ambrosi, Lucas; Baioni, Sebastián**

Colaboradores: Broglino, Florencia; Chiara, Laura; Dieguez, Romina; Farías, Martín; Fernández, María Laura; Fionda, César; Gómez, Estela; Ledesma, Máximo.

Universidad Argentina John F. Kennedy

Departamento de Diseño de Sistemas

Don Bosco 3729

1206 – Ciudad de Buenos Aires

República Argentina

T.E.: 54-11-4982-2736

Secoor@kennedy.edu.ar

### **RESUMEN**

HomerJ es una herramienta que facilita la Gestión de la Información contenida en documentos XML. A diferencia de otros motores de manejo de XML, éste utiliza como lenguaje de consulta el SQL, el mismo lenguaje que es utilizado por los servidores de bases de datos estándar que se pueden encontrar en el mercado.

De esta forma combina la flexibilidad y capacidad para almacenar información del XML con la potencia de un lenguaje de consulta sencillo, claro y universal como el SQL, ahorrándole al desarrollador de software el tiempo y la complejidad de aprender un lenguaje de consulta nuevo.

Se presentan las etapas de su desarrollo, herramientas utilizadas y los subsistemas que lo componen.

**Palabras clave:** XML, SQL, Database, Gestión de la Información.

## INTRODUCCIÓN

Una empresa tiene inevitablemente esparcida su información en ámbitos diversos (desde computadoras de escritorio hasta la red global, pasando por Intranets y redes locales) y aplicaciones heredadas útiles cuya transformación en cualquier dirección (plataforma de desarrollo, entorno operativo, etc.) no tiene sentido y debe poder hacer algo con todo eso. Y no sólo “algo” sino “algo que le sea de utilidad práctica”, ya que en el mercado actual la información se cotiza como el más estratégico de los recursos.

No estamos hablando de mostrar información sino de integrar procesos mediante esa información. XML no es un lenguaje sino un metalenguaje, es una forma de estructurar que nos permite generar la sintaxis adecuada para cada situación. Es una base que define un lenguaje que resulta comprensible, para cualquier aplicación. Por eso XML cambia todo, porque nos abre la puerta a otra forma de comunicarnos y de trabajar, que va a ser imprescindible en los próximos años.

Muchas de las aplicaciones desarrolladas hoy en día, trabajan guardando y/o leyendo información de archivos XML. La forma estándar de acceso a esta información, es mediante lenguajes de consulta como el Xpath[1], XQL[2], entre otros. Estos lenguajes presentan gramáticas complejas y en cierto modo limitadas a comparación de la capacidad de extensibilidad que posee XML, aunque cumplen con su objetivo de una forma eficaz.

Por otra parte, el SQL, es el lenguaje por excelencia de las bases de datos modernas. Permite realizar consultas de alta complejidad de una forma sencilla y veloz. Es un lenguaje muy difundido y fácil de aprender.

La idea surge a partir de la necesidad de gestionar la información contenida en los XML de una forma eficiente, sencilla, rápida y útil.

Este proyecto fue desarrollado en el marco de la asignatura Práctica Profesional, perteneciente al Departamento de Diseño de Sistemas, y su

objetivo fue eliminar barreras de tiempo y complejidad en el acceso a la información, agilizando de esta forma los tiempos de desarrollo y codificación del software, y por ende los tiempos de desarrollo de los sistemas. Estas son las bases sobre las cuales se cimienta HomerJ.

## **¿QUÉ ES XML?**

Es un formato basado en texto, específicamente diseñado para almacenar y transmitir datos. Un documento XML se compone de elementos XML cada uno de los cuales consta de una etiqueta de inicio, de una etiqueta de fin y de los datos comprendidos entre ambas etiquetas. Al igual que los documentos HTML, un documento XML contiene texto anotado por etiquetas. Sin embargo, a diferencia de HTML, XML admite un conjunto ilimitado de etiquetas, no para indicar el aspecto que debe tener algo, sino lo que significa. Por ejemplo: un elemento XML puede estar etiquetado como precio, número de pedido o nombre. El autor del documento es quien decide qué tipo de datos va a utilizar y qué etiquetas son las más adecuadas. Los documentos XML son fáciles de crear.

## **INICIOS DE HOMERJ**

HomerJ es concebido desde su origen como un servidor de bases de datos estándar, pero con la gran diferencia de realizar la persistencia de los datos[3] en archivos con formato XML, facilitando por otra parte el acceso a la información mediante consultas SQL.

Desde un inicio tanto autores como colaboradores, nos vimos inmersos en el dilema de cómo encarar la persistencia y como procesar las consultas. A lo largo de 8 meses, fuimos atravesando distintas etapas, abordando en cada una de ellas una perspectiva diferente de cómo llevaríamos a buen puerto este proyecto.

**Etapa I** (duración 2 días):

- ?? Análisis de requerimientos.
- ?? Estudio avanzado de las gramáticas de los distintos lenguajes SQL, XPath, XQuery.
- ?? Diseño de un modelo de persistencia de datos.
- ?? Elección de las herramientas de trabajo: lenguajes, sistemas operativos, compiladores e IDE's de desarrollo.

**Etapa II** (duración 2 mes):

- ?? Programación básica de una aplicación cliente servidor.
- ?? Especificación de un protocolo de comunicación entre el cliente y el servidor.
- ?? Estudio de protocolos de bases de datos (JDBC [4] y ODBC).
- ?? Testeo del funcionamiento en red de los componentes.

**Etapa III** (duración 6 mes):

- ?? Programación avanzada del servidor.
- ?? Interpretación de instrucciones SQL para resolver una consulta.
- ?? Programación de comandos básicos del servidor.

**Etapa IV Investigaciones Auxiliares** (a lo largo del proyecto):

- ?? Como desarrollar un protocolo JDBC.
- ?? Aplicación de políticas de seguridad en aplicaciones de red.
- ?? Investigación de servidores web: Apache Tomcat. y JBoss.
- ?? Investigación de especificaciones Java: JTA [5], JDO, JTS[6], JNDI, Java Beans [7]
- ?? XML:DB, una API para bases de datos XML.
- ?? XML-QL, un lenguaje de consulta para XML [8]

## **HERRAMIENTAS UTILIZADAS PARA SU DESARROLLO:**

- ?? Se utilizaron PC's con procesador Intel® Pentium III de 700 Mhz con 128 Mb de memoria RAM.
- ?? Java como lenguaje de desarrollo, nos permitió desarrollar un producto multiplataforma.
- ?? J2SE 1.4.1\_02 como maquina virtual.
- ?? Microsoft® Windows® XP Professional fue el sistema operativo utilizado para el desarrollo, aunque el producto fue testeado exitosamente sobre Windows® 98, Linux RedHat 8.0 y Linux Mandrake 7.1.
- ?? Eclipse 2.0 y Sun ONE Studio fueron los IDE's utilizados para el desarrollo.
- ?? Para el manejo de versiones, se utilizó Microsoft® Visual Source Safe®.

## **SUBSISTEMAS HOMERJ**

HomerJ no es una simple aplicación, HomerJ es un sistema compuesto por tres subsistemas. HomerJ Server, HomerJ Client y el Client-Server Protocol.

### **HomerJ Server**

Se caracteriza por su capacidad para atender varios clientes al mismo tiempo y su estabilidad. HomerJ esta diseñado para mantener el control de las conexiones aunque alguna de ellas falle o intente realizar operaciones indebidas. Como se puede ver en la Figura 1, HomerJ posee un sencillo panel de control para realizar funciones básicas como el apagado. Sobre la ventana de la consola, en la figura 2, se podrá ir observando toda la actividad del servidor .



Figura 1 – Panel de Control de HomerJ Server

```
*****  
**** HomerJ The Relational XML Project :: Database Server ****  
*****  
***** Iniciando ejecucion del servidor  
[2003-12-13 19:38:46.719] Loading configuration file: "F:\RXMLDB_Server\conf\server.xml"  
[2003-12-13 19:38:47.039] Trying to initialize HomerJ server on 127.0.0.1:13718  
[2003-12-13 19:38:47.149] Loading databases...  
[2003-12-13 19:38:47.179] Loading database file: agenda.xml  
[2003-12-13 19:38:47.860] Loading database file: homerj.xml  
[2003-12-13 19:38:47.921] Loading database file: peliculas.xml  
[2003-12-13 19:38:48.211] HomerJ running...
```

Figura 2 – Consola para el log de las actividades de HomerJ Server.

Es el servidor de bases de datos. El mismo, una vez iniciado, se quedará escuchando un puerto a la espera de conexiones con algún cliente. Se encargará de resolver todas las consultas originadas por los clientes.

Por el momento, HomerJ maneja consultas de tipo SELECT además de los comandos básicos de un servidor, aunque se encuentra en desarrollo el motor de ABM, para poder realizar las actualizaciones.

Los XML manejados por este proyecto poseen una estructura propia y especialmente definida. Nuestro próximo objetivo, es poder liberarnos de dicha estructura para lograr que el servidor sea tan flexible como lo puede ser un documento XML.

HomerJ Server, tiene una gran capacidad de proceso, una de sus cualidades, quizás la mas importante, es la posibilidad de devolver los resultados con distintos formatos. ¿Qué es esto?. Es un servidor de bases de datos, normalmente están estructuradas en formato de tablas. Llamemos a esta vista clásica o tradicional.

HomerJ puede retornar los resultados de una búsqueda con formato tabular o bien devolverlos con una estructura de documento XML, lo que resultaría algo más natural sabiendo que los datos están almacenados de esta forma.

## HomerJ Client

Es una aplicación cliente tipo consola. Este cliente es capaz de conectarse con un servidor HomerJ para la realización de las consultas, como se puede observar en la figura 3.

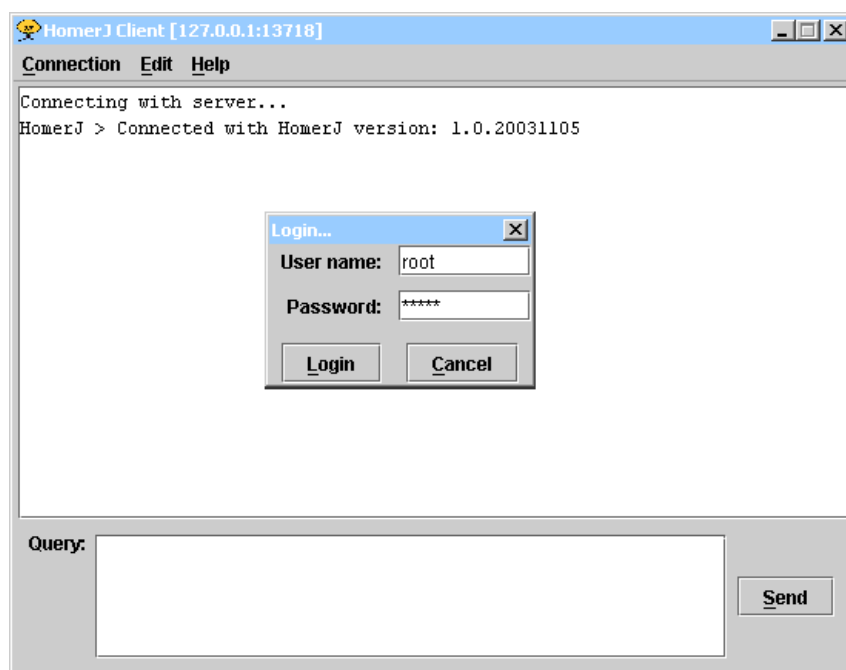


Figura 3 – Consola HomerJ Client conectada a un servidor.

Una vez establecida la conexión, el cliente deberá iniciar una sesión con su nombre de usuario y su contraseña antes de poder realizar alguna consulta. Ya iniciada la sesión, como se ve en la figura 4, se tendrá que escoger entre alguna de las bases de datos disponibles.

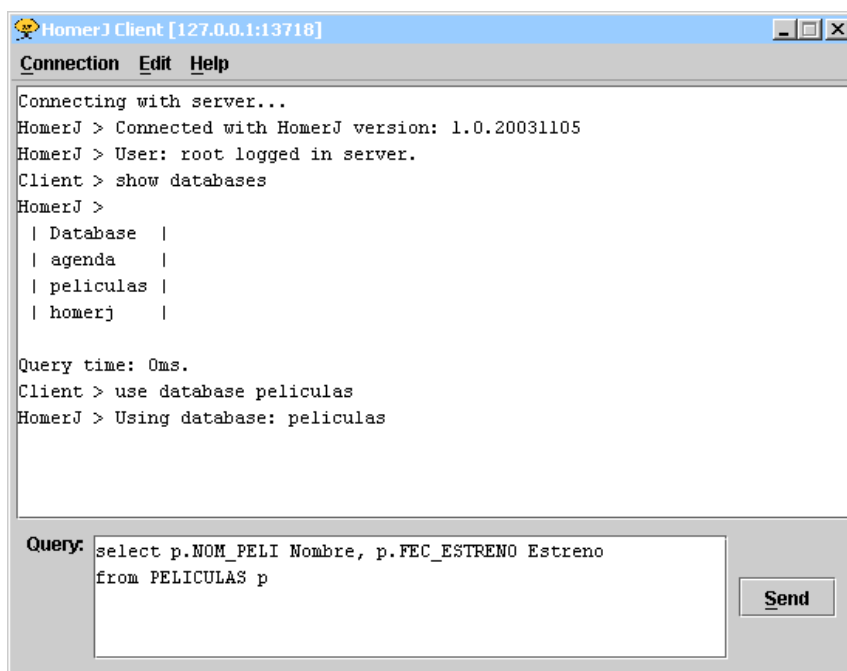


Figura 4 – Elección de una base de datos.

Como ya fue expuesto en el apartado anterior, el servidor HomerJ tiene la capacidad de responder de acuerdo a las necesidades del cliente, es por eso que hay dos formas de solicitarle los resultados de una consulta:

1. Por defecto, las respuestas serán devueltas en formato de tabla, como se aprecia en la figura 5.

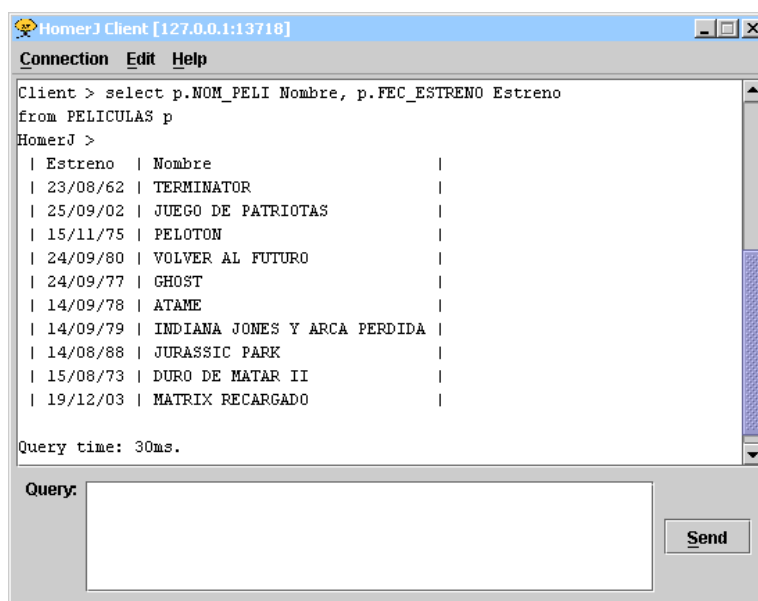
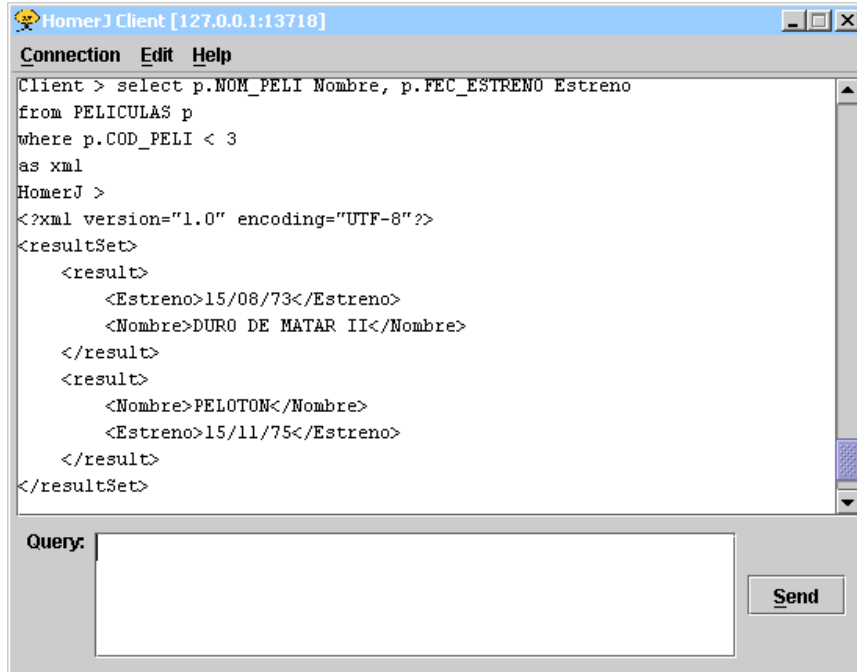


Figura 5 – Resultados de la consulta en forma tradicional.



2. En caso que nuestra aplicación requiera que las respuestas estén como un documento XML, se podrá aclarar agregando al final de la consulta el sufijo “as xml” , lo que podemos ver en la figura 6.



```
HomerJ Client [127.0.0.1:13718]
Connection Edit Help
Client > select p.NOM_PELI Nombre, p.FEC_ESTRENO Estreno
from PELICULAS p
where p.COD_PELI < 3
as xml
HomerJ >
<?xml version="1.0" encoding="UTF-8"?>
<resultSet>
  <result>
    <Estreno>15/08/73</Estreno>
    <Nombre>DURO DE MATAR II</Nombre>
  </result>
  <result>
    <Nombre>PELOTON</Nombre>
    <Estreno>15/11/75</Estreno>
  </result>
</resultSet>

Query:
Send
```

Figura 6 – Resultados de la consulta en formato XML.

### Client – Server Protocol

La comunicación entre el cliente y el servidor no podría ser posible sin la existencia de un protocolo que lo facilitara y diera el soporte necesario a todas las nuevas posibilidades que se nos están presentando.

En un principio, se investigo la alternativa de utilizar protocolos universales y establecidos como estándares, tal es el caso del JDBC. Este protocolo es un estándar en lo que se refiere a las conexiones de clientes Java a servidores de bases de datos. Lamentablemente, los protocolos de este tipo no están preparados para trabajar con XML. A lo largo del desarrollo se intentó agregar funcionalidad para facilitar el soporte, pero no fue posible concluir con una versión estable.

Finalmente, se optó por simplificar esta fase y desarrollar un protocolo propio que cumpliera de forma eficiente con la transmisión de la información; y

proporcionara toda la flexibilidad necesaria para transportar tanto tablas como el formato de documento XML.

## **CONCLUSIÓN**

SQL es un lenguaje universal de consulta de bases de datos y XML se ha convertido ya en un estándar de almacenamiento de información. HomerJ ha demostrado exitosamente, que una aproximación entre ambos es claramente posible.

A partir de ahora, quedan abiertos puntos para futuras investigaciones tales como: el desarrollo de un gestor de ABM completo y la integración de otro estándar como lo es el JDBC para el intercambio de información entre aplicaciones cliente y el servidor.

## REFERENCIAS

- [1] Definición del Lenguaje de consulta Xpath <http://www.w3.org/tr/xpath> Consultado: Abril de 2003.
- [2] Definición del Lenguaje de consulta XQL <http://www.w3.org/tands/ql/ql98/pp/xql.html> Consultado: Abril de 2003.
- [3] Interface de bases de datos en XML <http://www.xmldb.org/> Consultado: Abril 2003.
- [4] API estándar de conexión a bases de datos desde Java <http://www.programacion.com/bbdd/tutorial/jdbc> Consultado: Junio 2003.
- [5] Java Transaction API (JTA) <http://java.sun.com/products/jta> Consultado: Julio 2003.
- [6] Java Transaction Service (JTS) <http://java.sun.com/products/jts> Consultado: Julio 2003.
- [7] Enterprise JavaBeans (EJB) <http://java.sun.com/products/ejb> Consultado: Junio 2003.
- [8] Definición del Lenguaje de consulta XML-QL <http://www.w3.org/TR/NOTE-xml-ql> Consultado: Abril de 2003.