

## CAPÍTULO 5

### IMPLEMENTACIÓN DEL SISTEMA

#### 5.1 Servidor Web Jakarta-Tomcat

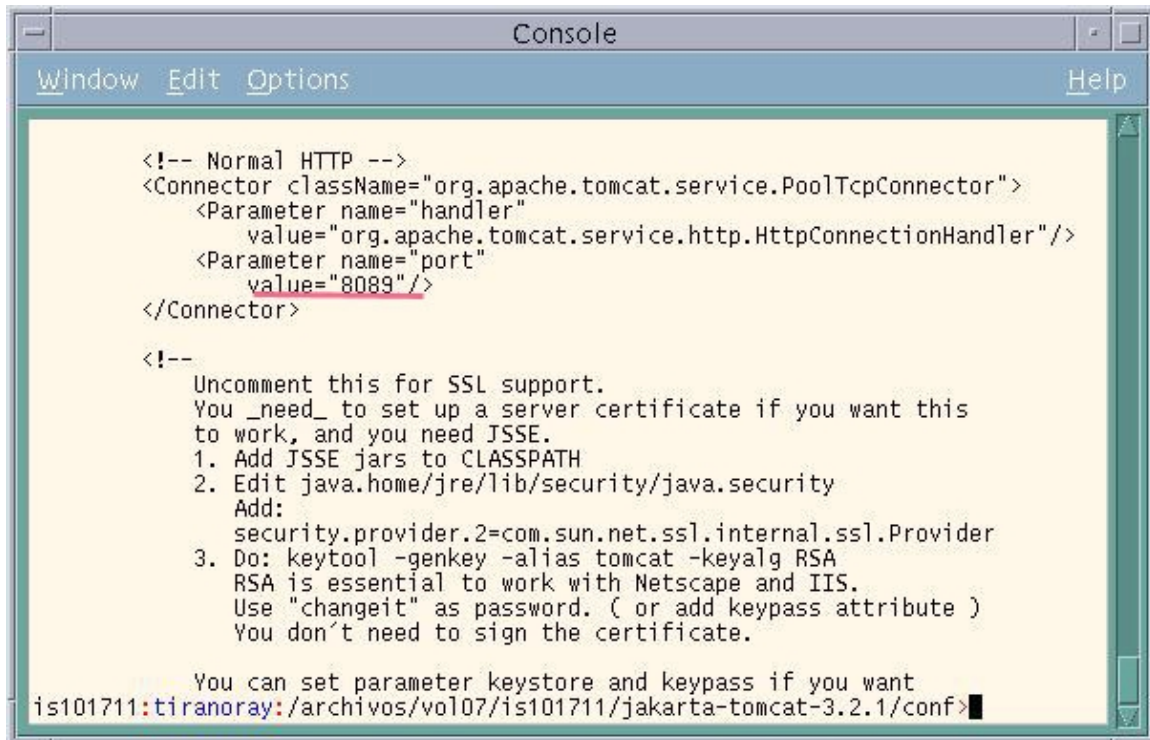
Para el desarrollo de un Sistema de Comercio Electrónico (SCE) se usó un servidor Web compatible con la tecnología Java HTTP Servlets y JSP, para esto se utilizó la implementación del Proyecto Jakarta: Servidor Web Jakarta-Tomcat.

El Proyecto Jakarta tiene el propósito de proveer de soluciones comerciales de calidad en una arquitectura Cliente/Servidor basadas sobre la plataforma Java [Jakarta, 2001]. Tomcat, Watchdog y Taglibs son los tres subproyectos de Jakarta. Tomcat es la implementación de JSP 1.1 y Servlets 2.2 y en su versión de jakarta-tomcat-3.2.1 se desarrolló el sistema de aplicación del e-Commerce Constructor. Cabe mencionar que éste servidor es gratuito, y hasta existe la posibilidad de bajar el código fuente.

Tomcat se puede instalar en cualquier máquina dígame computadora personal, la aplicación binaria esta disponible para instalarse y trabajar en cualquier sistema operativo que soporte Java desde una versión 1.1.7 y/o mayor, se probó en computadoras PC, Sun (véase capítulo 1), y sobre sistemas operativos Unix, Linux y Windows, desafortunadamente en Macintosh no se pudo probar satisfactoriamente, pues la aplicación binaria no cuenta con los scripts necesarios para su ejecución, pero la versión en código fuente al tratar de instalarlo con un servidor Web su desempeño fue bastante deficiente pues la integración con Java en Mac es demasiado complicada debido a que el sistema operativo MacOS 9 y/o menor controla los threads de Java con el MacOS Runtime for Java (MRJ), su manejador de Java, el cual procesa las ejecuciones de la Java Virtual Machine (JVM), haciendo que se procesen dos trabajos por separados y sea más pesado para el servidor, pues depende de que el proceso de la JVM ejecutado por el MRJ sea terminado y asignado al Servidor Web. Sería interesante observar el desempeño sobre MacOS X el cual ya esta implementado completamente en Java y trabaja en sobre Java 2 Standard Edition 1.3 y con el MacOS X Server que soporta tecnología Servlet y JSP.

Con todo y estas situaciones se puede observar que la decisión de implementar el sistema en JSP fue muy acertada, pues por el momento la presencia de desarrollar aplicaciones en Java se postula como una propuesta real en el presente y el una opción para el futuro.

Por último para la configuración del servidor se debe tener la especificación del nombre de la máquina que fungirá como servidor y soportará las operaciones del e-CC así como de la Base de Datos. La configuración del servidor depende de un archivo llamado `server.xml` (véase figura 5.1), este archivo contiene la configuración que necesita el servidor para ejecutarse correctamente. Para configurar el puerto en el que el servidor va a trabajar, se edita el archivo `server.xml` que se localiza en `/jakarta-tomcat-3.2.1/conf/` si este no se modifica el puerto por *default* es el 8080.



```

<!-- Normal HTTP -->
<Connector className="org.apache.tomcat.service.PoolTcpConnector">
  <Parameter name="handler"
    value="org.apache.tomcat.service.http.HttpConnectionHandler"/>
  <Parameter name="port"
    value="8089"/>
</Connector>

<!--
Uncomment this for SSL support.
You _need_ to set up a server certificate if you want this
to work, and you need JSSE.
1. Add JSSE jars to CLASSPATH
2. Edit java.home/jre/lib/security/java.security
   Add:
   security.provider.2=com.sun.net.ssl.internal.ssl.Provider
3. Do: keytool -genkey -alias tomcat -keyalg RSA
   RSA is essential to work with Netscape and IIS.
   Use "changeit" as password. ( or add keypass attribute )
   You don't need to sign the certificate.

You can set parameter keystore and keypass if you want
is101711:tiranoray:/archivos/vol07/is101711/jakarta-tomcat-3.2.1/conf>
```

Figura 5.1 Edición del Archivo `server.xml`.

Para iniciar el servidor se debe ejecutar el archivo `startup`, dependiendo del sistema operativo `.sh` o `.bat` y para apagarlo el archivo `shutdown` se localiza en `/jakarta-tomcat-3.2.1/bin/` estos archivos forman parte del servidor `jakarta-tomcat-3.2.1`.

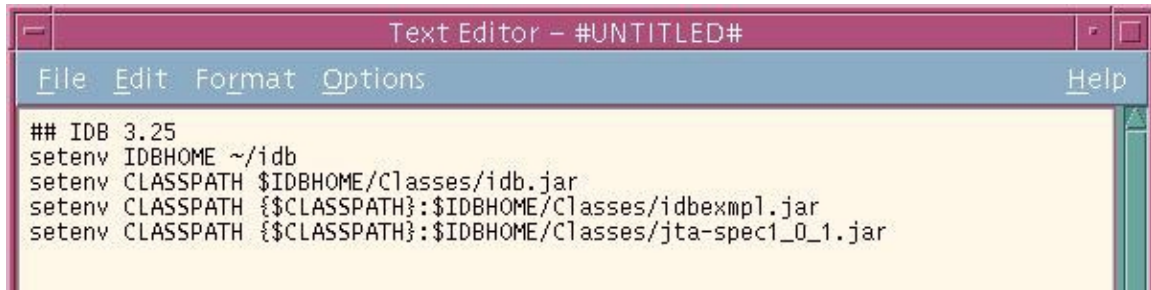
## **5.2 Base de Datos**

### *5.2.1 DBMS: Instant DataBase (IDB)*

Instant DataBase es un Manejador de Bases de Datos Relacionada (RDBMS Relational DataBase Management System) implementada completamente en Java. Soporta proyecciones, transacciones, triggers, sub-selecciones, renombramiento de tablas así como muchas opciones más. Es sencilla, eficiente, pequeña en espacio y fácil de instalar, ésta es accesada a través de su propio Driver de JDBC y administrada usando SQL standard. Cabe mencionar que es gratuita, es un producto avalado por Lutris Technology. Se puede observar las especificaciones, API de IDB y bajar un DBMS desde su página:

<http://www.lutris.com/products/projects/instantDB/>

Para disponer de IDB es necesario establecer las variables de ambiente de Java, estas variables le indicaran a la hora de arrancar el servidor en donde buscar los archivos necesarios para su funcionamiento. Dependiendo del sistema operativo varían las variables, tales como `PATH`, `JAVA_HOME`, `CLASSPATH`, pero en este caso solo se especificará la variable `IDBHOME` y `CLASSPATH` con el path o ruta en donde se encuentran los archivos de trabajo, así como las librerías de IDB, las cuales son `idb.tar`, `idbexmpl.tar` y `jta-spec1_0_1.jar` éstas son agregadas al classpath y salvadas en un archivo de configuración correspondiente (`autoexec.bat` en caso de ser Windows y `.cshrc` y/o `.config` en caso de ser Unix o Linux el sistema operativo o alguno otro). En la figura 5.2 se muestra un ejemplo en un ambiente Unix.

A screenshot of a text editor window titled "Text Editor - #UNTITLED#". The window has a menu bar with "File", "Edit", "Format", "Options", and "Help". The main text area contains the following shell configuration code:

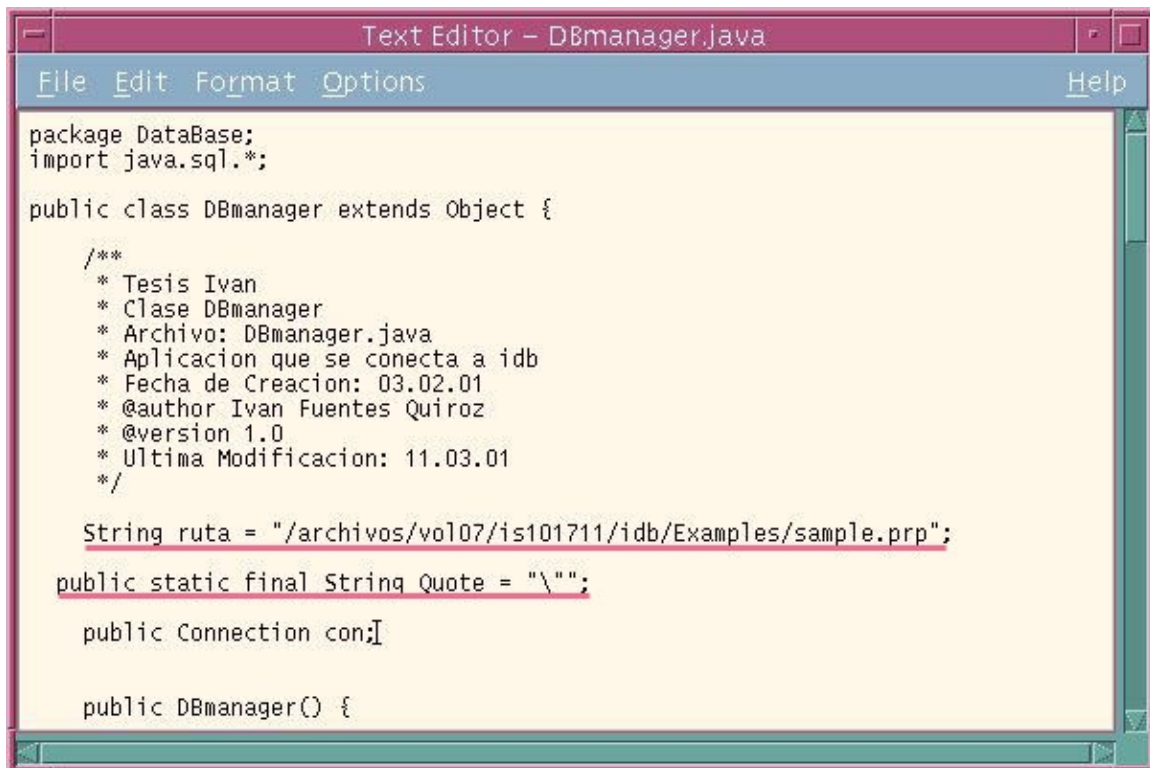
```
## IDB 3.25
setenv IDBHOME ~/idb
setenv CLASSPATH $IDBHOME/Classes/idb.jar
setenv CLASSPATH {$CLASSPATH}:$IDBHOME/Classes/idbexmpl.jar
setenv CLASSPATH {$CLASSPATH}:$IDBHOME/Classes/jta-spec1_0_1.jar
```

Figura 5.2 Ejemplo de Configuración del CLASSPATH para IDB en Unix.

IDB es la base de datos con la cual fue desarrollado el presente proyecto, pero no está obligado a trabajar con él. La utilización de JDBC le da la característica de portabilidad con la base de datos.

### *5.2.2 Portabilidad del DBMS*

La portabilidad de Java le da la oportunidad de cambiar de Manejador de Base de Datos si es que el sistema en el que se va a implementar cuenta con otra base de datos y/o se necesitan almacenar tipos de datos especiales los cuales IDB no maneja o contiene. Para esto se diseñó una clase en Java llamada DBmanager (véase fig. 5.3), esta clase es la que se encarga de realizar la conexión con la base de datos, y maneja los elementos como la ruta en donde se localiza la base de datos, y algunos aspectos como el manejo de comilla (`quote`) sencilla o doble que varía al momento de hacer las queries a la base de datos.



```
Text Editor - DBmanager.java
File Edit Format Options Help
package DataBase;
import java.sql.*;

public class DBmanager extends Object {

    /**
     * Tesis Ivan
     * Clase DBmanager
     * Archivo: DBmanager.java
     * Aplicacion que se conecta a idb
     * Fecha de Creacion: 03.02.01
     * @author Ivan Fuentes Quiroz
     * @version 1.0
     * Ultima Modificacion: 11.03.01
     */

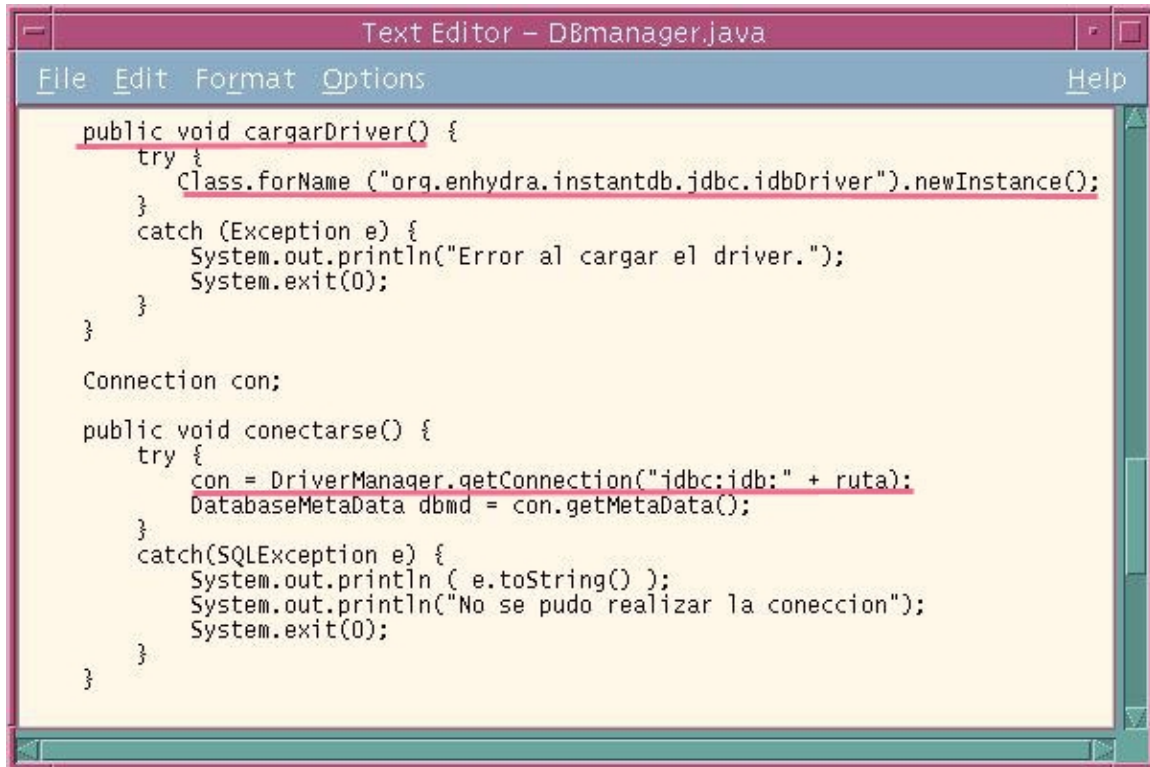
    String ruta = "/archivos/vol07/is101711/idb/Examples/sample.prp";
    public static final String Quote = "\"";

    public Connection con;

    public DBmanager() {
```

Figura 5.3 Archivo DBmanager.java aplicación de conexión a Base de Datos.

También se puede especificar la ruta en donde se encuentra físicamente el archivo de la BD y el tipo de comilla que utiliza. Para cambiar de DBMS lo que se debe de hacer es agregar la especificación del driver a manejar, además de indicar la tecnología utilizada JDBC u ODBC. En la figura 5.4 se muestra la especificación de éstas características.



```
Text Editor - DBmanager.java
File Edit Format Options Help

public void cargarDriver() {
    try {
        Class.forName ("org.enhydra.instantdb.jdbc.idbDriver").newInstance();
    }
    catch (Exception e) {
        System.out.println("Error al cargar el driver.");
        System.exit(0);
    }
}

Connection con;

public void conectarse() {
    try {
        con = DriverManager.getConnection("idbc:idb:" + ruta);
        DatabaseMetaData dbmd = con.getMetaData();
    }
    catch(SQLException e) {
        System.out.println ( e.toString() );
        System.out.println("No se pudo realizar la coneccion");
        System.exit(0);
    }
}
```

Figura 5.4 Especificaciones de Driver y Conectividad a la Base de Datos.

De esta manera se tiene la opción de implementar el sistema no sólo en cualquier plataforma, sino también con cualquier base de datos, con tecnologías JDBC u ODBC. Claro que para poder ejecutarlas, deben crearse las tablas correspondientes específicas en el capítulo 4.

Con el propósito de administrar la base de datos se desarrolló una aplicación que funge como manejador de la base de datos. Ésta se llama Query, en el se pueden realizar *queries* para la consulta y manejo, así como la inserción y modificación de datos a través de *queries* en SQL. Cabe mencionar que Query consta con su propia clase DBmanager.

La aplicación Query (véase fig. 5.5) se conecta a la base de datos dando la opción de ejecutar *queries* en un campo de texto y en otro independiente para presentar el o los resultados de éste así como limpiar los campos para la ejecución de un nuevo *query*, y

por último desconectarse de la base y salir de la aplicación misma.

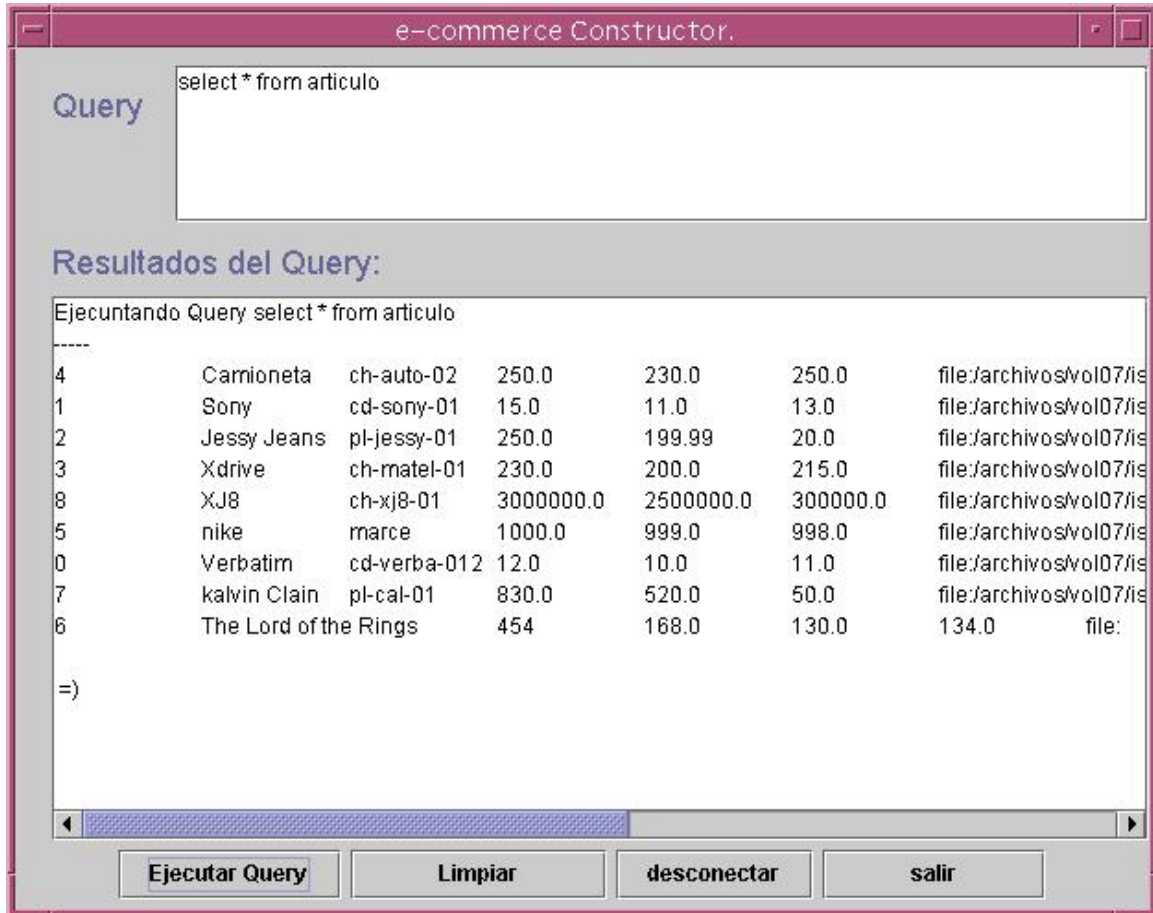


Figura 5.5 Aplicación Query.

Teniendo además esta aplicación es más sencillo la portabilidad de la base de datos, así como su administración, pues de éste modo se pueden realizar pruebas de validación y verificación de una nueva base de datos y de la integridad de los mismos, asegurando la conexión a la base.

### 5.3 Ubicación del Sistema

Tras éstas observaciones, no queda nada mas por hacer que iniciar el servidor, y

## **Capítulo 7: Resultados y Trabajo a Futuro**

posteriormente a través de cualquier navegador dando la dirección electrónica, se podrá acceder a la página principal del ambiente. En el caso particular de la presentación de esta tesis la dirección de inicio del sistema se depositó en el servidor de la UDLA-P llamado `acadaplic` en el puerto 8089, siendo ubicada en el directorio "tesis", y la dirección electrónica <http://acadaplic.pue.udlap.mx:8089/tesis> para ser accesada vía Internet.

La tienda puede ser accesada a través del mismo servidor, pero en la siguiente dirección electrónica: <http://acadaplic.pue.udlap.mx:8089/tesis/tienda> para navegar en ella.