

Programación en J2EE

Anexo JDBC

INDICE:

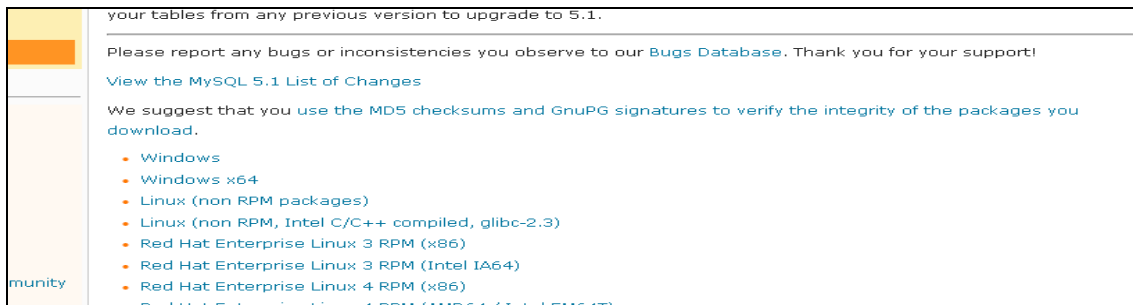
Programación en J2EE.....	1
Anexo JDBC.....	1
1 Instalación de Entorno.....	2
1.1 MySQL	2
1.2 MySQL Tools	5
2 Drivers JDBC	9
2.1 Definición	9
2.2 Tipos de Driver JDBC	9

1 Instalación de Entorno

En nuestros ejercicios usaremos **MySQL 5.1**. (tamaño aprox. 60-80 MB)

<http://dev.mysql.com/downloads/mysql/5.1.html>

Está tanto para Linux, como para Windows. El alumno elegirá el entorno que prefiera



En esta sesión se verá como acceder y manipular la información de una base de datos. Ahora bien, no se verá el diseño Entidad-Relación de la misma y su paso a tablas, dado que eso correspondería a otro tipo de curso y llevaría mucho más tiempo. Es por ello, que partiremos de una base de datos común que el profesor aportará. Como herramientas para el manejo de la base de datos usaremos las **Mysql Tools**:

<http://dev.mysql.com/downloads/gui-tools/5.0.html>

Está tanto para Linux, como para Windows. El alumno elegirá el entorno que prefiera

Windows downloads

The install package uses the Windows Installer, which is built in to Windows XP and more recent Microsoft Windows versions. [An update for Windows 2000 can be downloaded here.](#)

Windows (x86)	5.0-r12	17.4M	Pick a mirror
	MD5: 7d6d546069554900e6d9a324b6840336		
Without installer (unzip in C:\)	5.0-r12	16.6M	Pick a mirror
	MD5: 9f396065bc095ff73dbd6e478554ee62		

Mac OSX downloads

MAC OSX 10.4 (Universal binaries)	5.0-r12	12.4M	Pick a mirror
	MD5: 87989a094c5edb6efb28cdad50dd1a44		

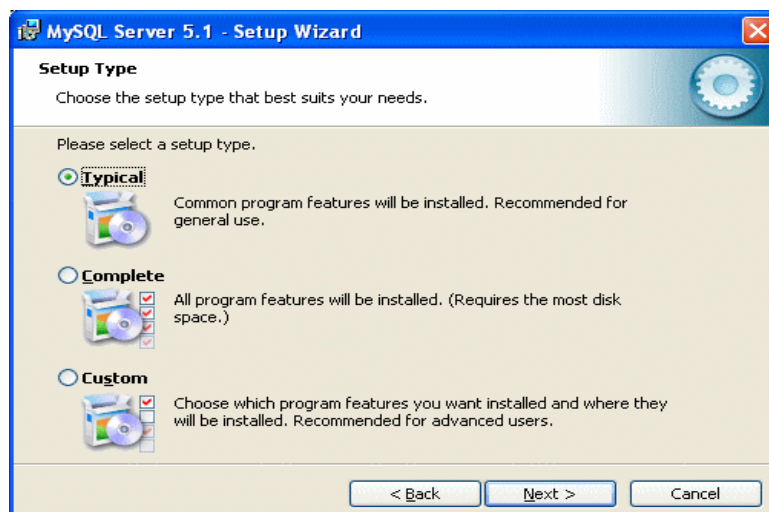
Linux downloads

Individual RPM packages are supplied for the GUI Tools components and combined into a single TAR archive.

RedHat Enterprise Linux 3 (x86) RPM (bundled dependencies)	5.0r12	19.4M	Pick a Mirror
	MD5: 45efb53691698dae0a6badbb27f9c742		

1.1 MySQL

Una vez hemos descargado MySQL hemos de instalarlo. Los pasos para ello son, una vez hemos pulsado Setup.exe, pulsaremos siempre Next, y prestaremos especial atención a las siguientes pantallas:

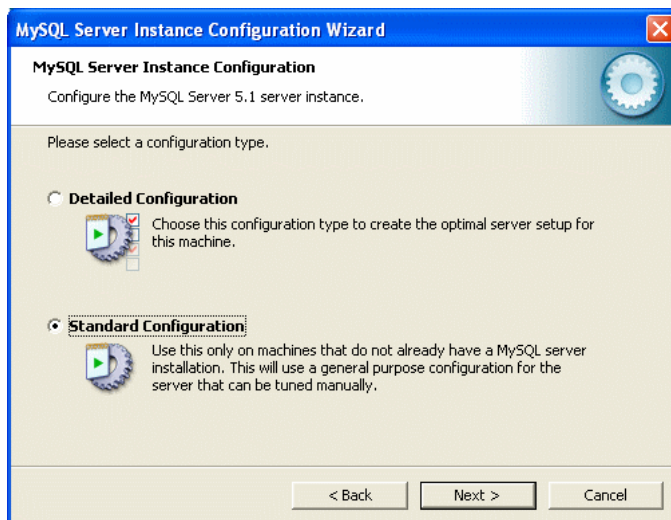


En la pantalla de la figura elegiremos Typical

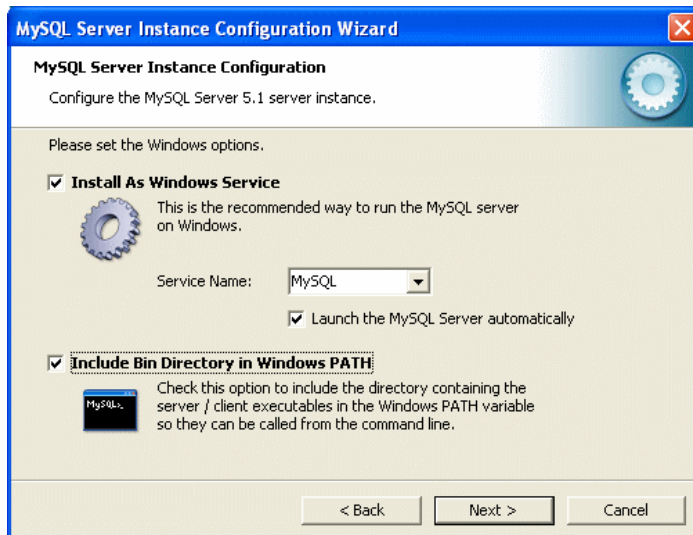
En esta pantalla indicaremos que deseamos configurar el servidor de Base de Datos



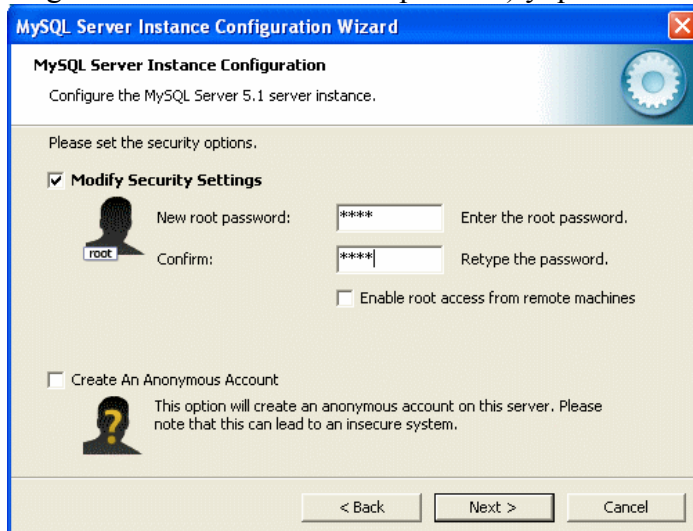
En la siguiente pantalla indicamos que deseamos utilizar la configuración estándar



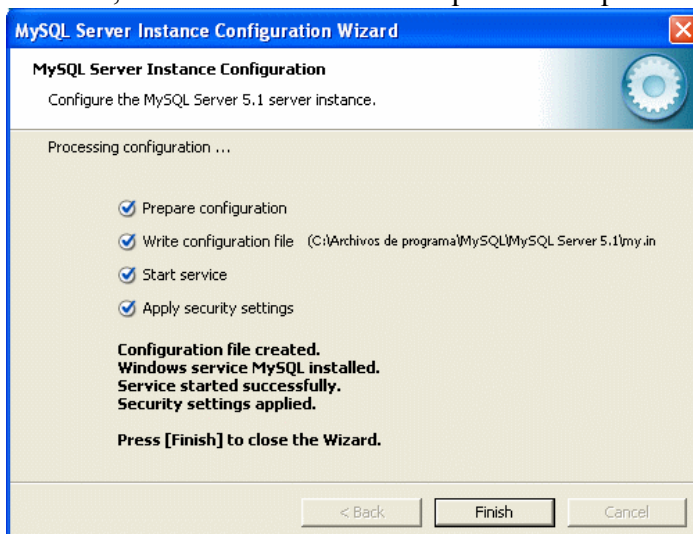
En esta pantalla indicaremos que queremos tener la Base de Datos funcionando como un servicio del Sistema Operativo, y además queremos que el PATH incluya los comandos de gestión de la Base de Datos.



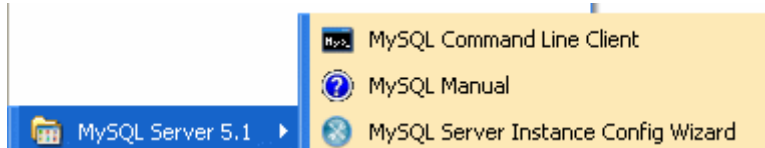
En la siguiente pantalla indicamos el password de root (pondremos todos root, para luego facilitar la corrección al profesor) y que no cree ninguna cuenta anónima.



En esta pantalla muestra que todo ha ido correctamente (es importante que salga todo correcto, en caso contrario habría que revisar que ocurre)



Una vez hemos instalado todo, veremos una entrada de programas como la que se muestra a continuación

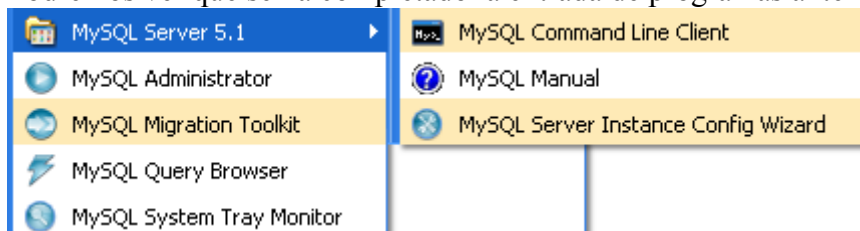


1.2 MySQL Tools

Una vez hemos descargado MySQL Tools hemos de instalarlo. Los pasos para ello es seguir el instalador pulsando siempre Next. Prácticamente la única pantalla a considerar es la siguiente, en la cual elegiremos el tipo de instalación completa:



Podremos ver que se ha completado la entrada de programas anterior con



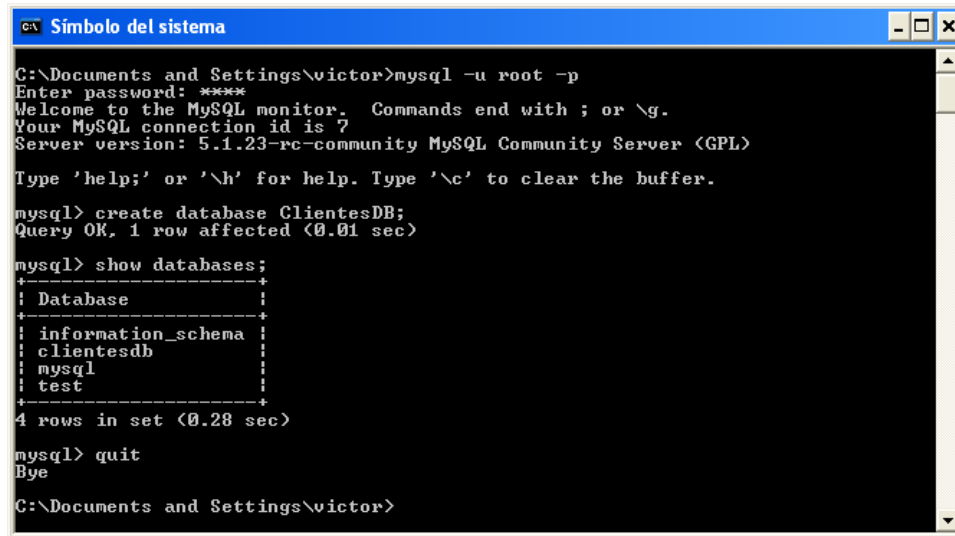
1.2.1 Crear Base de Datos de ejemplo

Una vez tenemos instalado el entorno procederemos a crear nuestra primera base de datos (técnicamente lo que crearemos es un esquema de base de datos, ya que la base de datos es MySQL y lo que contiene son distintos esquemas) que podremos usar el resto del tema para los ejemplos. Para ello haremos:

- 1- Acceder al símbolo de Sistema
- 2- Ejecutar el monitor de MySQL, mediante **mysql -u root -p** (esto es, conéctate a localhost como root y pide el password)
- 3- Escribir la orden **create database ClientesDB;** (de este modo crea el esquema)
- 4- Podemos escribir **show databases;** para cerciorarnos que la ha creado

5- Escribimos **quit** para salida del monitor de MySQL.

Esto es:



```

C:\Documents and Settings\victor>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.1.23-rc-community MySQL Community Server <GPL>

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database ClientesDB;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| clientesdb |
| mysql |
| test |
+-----+
4 rows in set (0.28 sec)

mysql> quit
Bye
C:\Documents and Settings\victor>

```

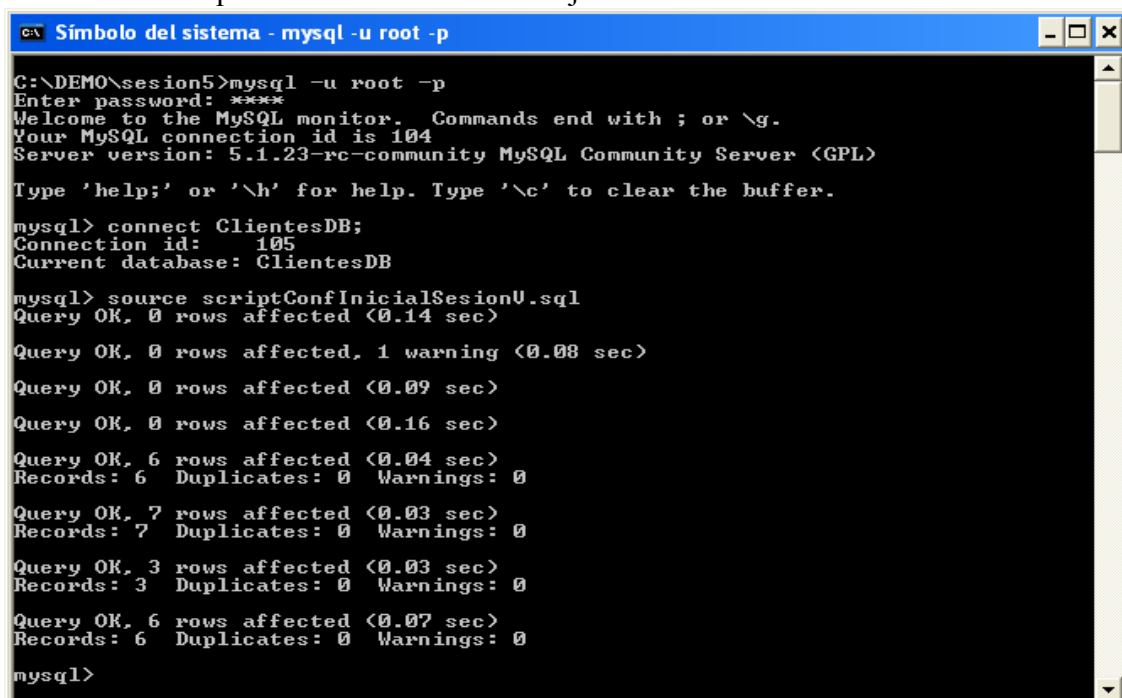
1.2.2 Rellenar la Base de Datos de ejemplo

Una vez creada la base de datos vamos a usar el script llamado `scriptConfInicialSesionV.sql` que creará las tablas de la Base de Datos y las rellenará con datos de prueba.

Para ello haremos:

- 1- Descargar el fichero **scriptConfInicialSesionV.sql** que nos proporciona el profesor y dejarlo en una carpeta llamada `sesion5` de nuestra carpeta `DEMO` (creada en sesiones anteriores)
- 2- Acceder al símbolo de Sistema
- 3- Ir a la carpeta `sesion5` de `DEMO` (`cd C:\DEMO\sesion5`)
- 4- Ejecutar el monitor de MySQL, mediante **mysql -p -u root**
- 5- Nos conectaremos a nuestra base de datos, mediante **connect ClientesDB;**
- 6- Ejecutaremos nuestro script mediante **source scriptConfInicialSesionV.sql;**

De este modo podemos ver si todo se ha ejecutado correctamente.



```

C:\DEMO\sesion5>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 104
Server version: 5.1.23-rc-community MySQL Community Server <GPL>

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> connect ClientesDB;
Connection id: 105
Current database: ClientesDB

mysql> source scriptConfInicialSesionV.sql
Query OK, 0 rows affected (0.14 sec)

Query OK, 0 rows affected, 1 warning (0.08 sec)
Query OK, 0 rows affected (0.09 sec)
Query OK, 0 rows affected (0.16 sec)
Query OK, 6 rows affected (0.04 sec)
Records: 6 Duplicates: 0 Warnings: 0
Query OK, 7 rows affected (0.03 sec)
Records: 7 Duplicates: 0 Warnings: 0
Query OK, 3 rows affected (0.03 sec)
Records: 3 Duplicates: 0 Warnings: 0
Query OK, 6 rows affected (0.07 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql>

```

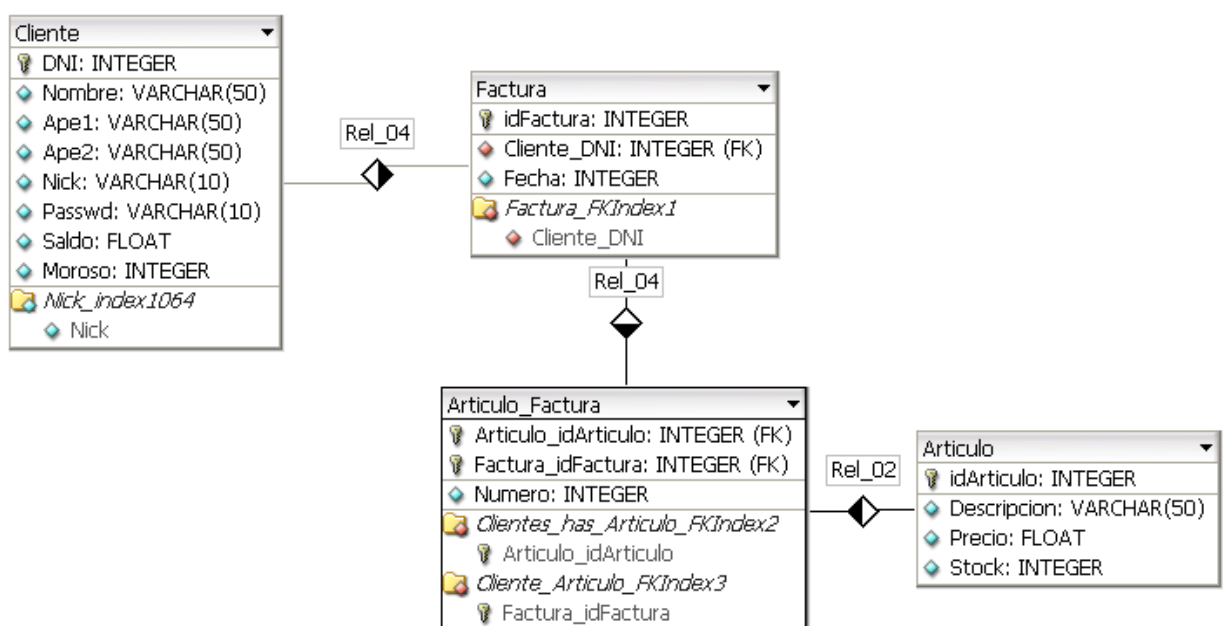
Veamos ahora mediante MySQL Query Browser las tablas creadas. Rellenaremos el dialogo de entrada como se indica en la figura



Así que

podremos ver las tablas conforman ClientesDB

Añado el esquema (no se ha obtenido con Mysql Tools) de estas tablas y su inter-relación, para su mejor entendimiento



Podemos realizar las consultas SQL que deseemos a través de la entrada de comandos de MySQL Query Browser arrastrando la tabla sobre la entrada de comandos



De este modo, por ejemplo podemos ver el contenido de la tabla clientes

MySQL Query Browser - Connection: root@localhost:3306 / ClientesDB

File Edit View Query Script Tools Window MySQL Enterprise Help

Go back Next Refresh

`SELECT * FROM clientesdb.cliente c;`

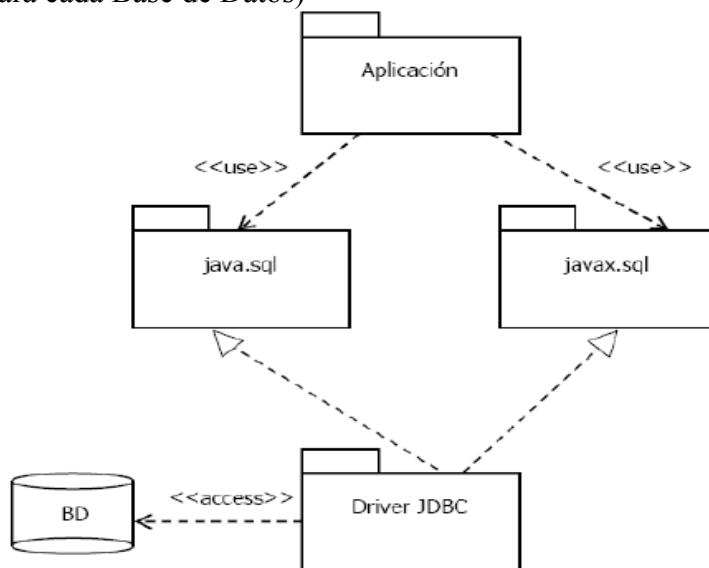
Resultset 1

DNI	Nombre	Ape1	Ape2	Nick	Passwd	Saldo	Moroso
14	Francisco	Lopez	Sanchis	franlo	fran1	1000	0
15	Pedro	Martinez	NULL	peter	pet1	1000	0
25	Manolo	Perez	Aspar	manu	man1	1000	0
33	Marcos	Heredia	Buendia	marcs	mar1	1000	0
367809	Guillermo	Toro	Fuentes	guille	gui1	1000	0
456781	Julia	Sanches	Guarner	juli	jul1	1000	0
31678901	Ana	Bermejo	Beltan	anaber	ana1	1000	0

2 Drivers JDBC

2.1 Definición

Los **drivers** permiten conectarnos con una base de datos determinada y lanzar queries para manipularla. Un driver suele ser un fichero .jar que contiene una implementación de todos los interfaces del API de JDBC. Idealmente nuestro código nunca dependerá del driver, dado que siempre trabaja contra los paquetes `java.sql` y `javax.sql` (en la práctica existen diferentes dialectos de SQL-a pesar de ser un estandard-para cada Base de Datos)



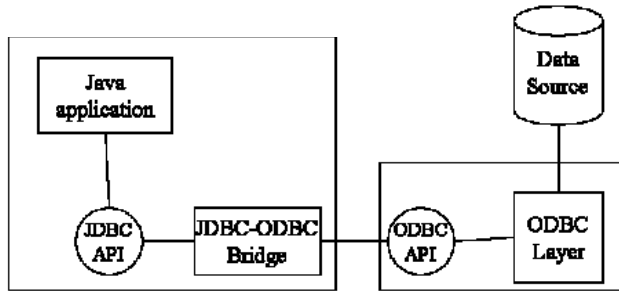
2.2 Tipos de Driver JDBC

Existen cuatro tipos de drivers JDBC, cada uno con una filosofía de trabajo diferente.

2.2.1 Tipo 1. JDBC-ODBC:"BRIDGE"

Este driver permite al programador acceder a fuentes de datos ODBC existentes mediante JDBC. El JDBC-ODBC Bridge (puente JDBC-ODBC) implementa operaciones JDBC traduciéndolas a operaciones ODBC, se encuentra dentro del paquete `sun.jdbc.odbc` y contiene librerías nativas para acceder a ODBC. Cada máquina que use el driver, debe tener librerías de ODBC (dll propias del S.O)

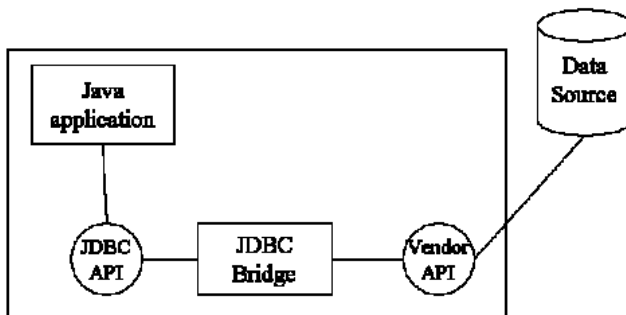
- **Ventajas:** Buena forma de aprender JDBC. También puede ser buena idea usarlo en sistemas donde cada máquina cliente tenga ya instalado los drivers ODBC. También es posible que sea la única forma de acceder a ciertos motores de Bases de Datos.
- **Inconvenientes:** No es buena idea usar esta solución para aplicaciones que exijan un gran rendimiento, ya que la transformación JDBC-ODBC es costosa. Tampoco es buena solución para aplicaciones con alto nivel de escalabilidad.



2.2.2 Driver Java parciales: “NATIVE”

Traducen las llamadas al API de JDBC Java en llamadas propias del motor de Base de Datos (Oracle, Informix...). Al igual que el tipo anterior, exige en las máquinas clientes código binario propio del cliente de la Base de datos específica y del sistema operativo.

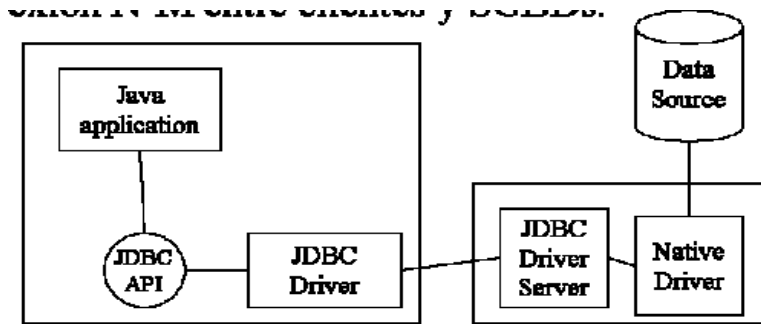
- **Ventajas:** Mejor rendimiento que el anterior. Quizá puede ser buena solución para entornos controlados como intranets. Ejemplo OCI oracle.
- **Inconvenientes:** Principalmente la escalabilidad, ya que estos drivers exigen que en la máquina cliente librerías del cliente de la Base de Datos. (En nuestro caso el cliente de Base de datos es el servidor de aplicaciones)



2.2.3 Driver JDBC a través de Middleware: “NETWORK”

Traduce las llamadas al API JDBC en llamadas propias a un protocolo independiente del DBMS (DataBase Management System, Sistema Gestor de Bases de Datos SGBD). Éste se encargará de traducirlas de nuevo en sentencias propias del motor de Base de Datos de cada caso.

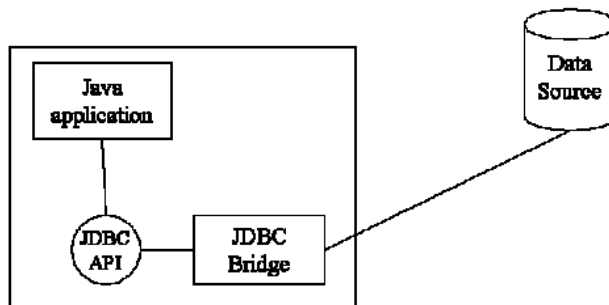
- **Ventajas:** 100 % portable. Buena solución cuando necesitamos acceder a Bases de Datos distintas y se quiere usar un único driver JDBC para acceder a las mismas. Al residir la traducción en el servidor del middleware, los clientes no necesitan librerías específicas, tan solo el driver.
- **Inconvenientes:** Al ser independiente de la plataforma, no aprovecha las características específicas de la Base de Datos



2.2.4 Driver java puro: “THIN”

Convierte o traduce las llamadas al API JDBC en llamadas al protocolo de red usado por el motor de bases de datos, lo que en realidad es una invocación directa al motor de bases de datos.

- *Ventajas:* 100 % portable. Buen rendimiento. El cliente sólo necesita el driver. (como ya se ha comentado nuestro cliente de Base de datos es el servidor de aplicaciones)
- *Inconvenientes:* La desventaja principal reside en la configuración del servidor que accede a Base de Datos, dado que necesitará librerías específicas para cada motor de base de datos distinto.



Nosotros usaremos este último tipo de Drivers. Concretamente usaremos:

<http://www.mysql.com/products/connector/j/>

Es probable encontrarnos varios drivers distintos para la misma fuente de datos. Debemos saber detectar cual es el driver más adecuado para nuestra aplicación, por ejemplo, si elegimos un driver ODBC/JDBC, tendremos más flexibilidad para elegir distintas fuentes de datos, pero si por ejemplo trabajamos con una Base de Datos Oracle, un driver JDBC diseñado para esta base de datos será mucho más eficiente.

Para aclarar los distintos tipos de Drivers veamos el siguiente esquema

