

Acceso a Base de Datos

- JDBC fue diseñado para mantener sencillas las cosas sencillas.
- JDBC nos permite hacer una conexión hacia una Base de datos.
- Podemos realizar cualquier tipo de consultas, modificación y creación de nuevas tablas.

Acceso a Base de Datos

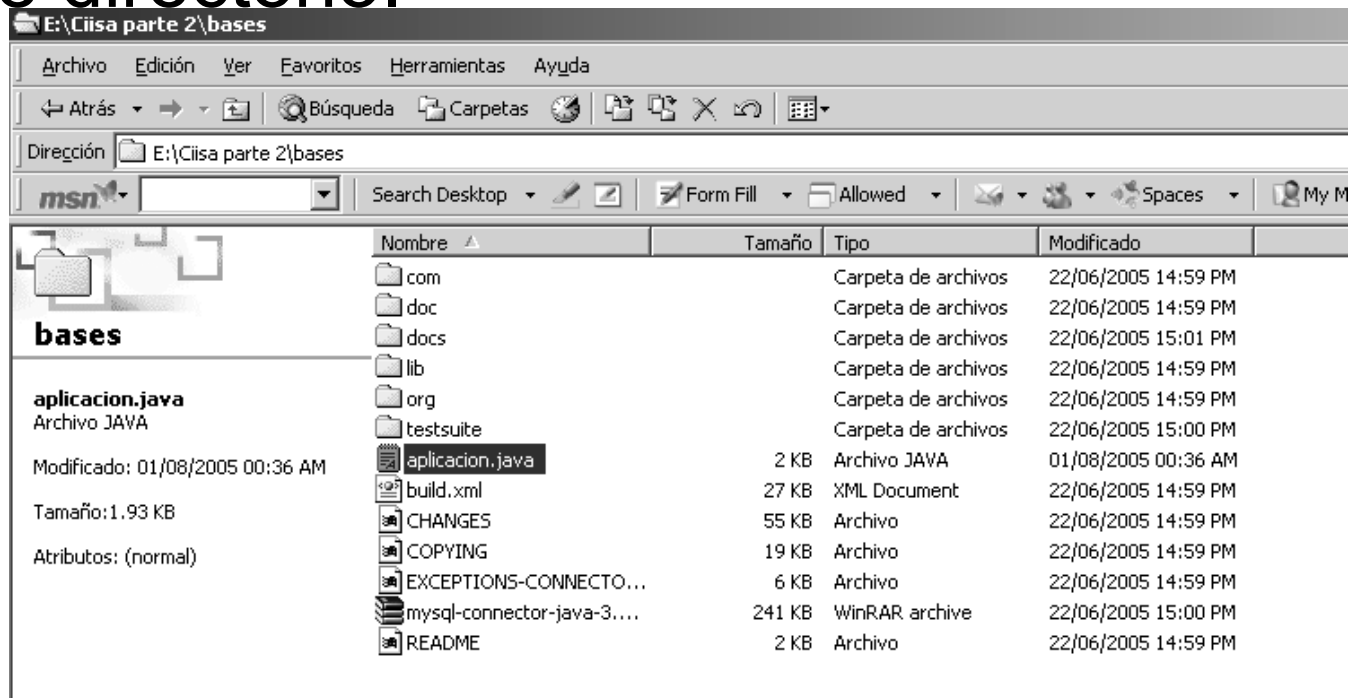
- Los pasos necesarios:
 - Cargar el driver (el driver se obtiene del proveedor de la BD).
 - Establecer la conexión.
 - Crear sentencias.
 - Ejecutar las sentencias.

Acceso a Base de Datos

- Pero que hacer con el driver? Como lo uso?
- La mejor respuesta disponible es esta:
 - *"Once you have un-archived the distribution archive, you can install the driver in one of two ways:*
 - ▶ *Either copy the "com" and "org" subdirectories and all of their contents to anywhere you like, and put the directory holding the "com" and "org" subdirectories in your classpath,*
 - ▶ *or put mysql-connector-java-[version]-bin.jar in your classpath, either by adding the FULL path to it to your CLASSPATH environment variable,*
 - ▶ *or by copying the .jar file to \$JAVA_HOME/jre/lib/ext."*

Acceso a Base de Datos

- Pero hay una opción más sencilla en Windows:
- Descomprimir el .zip, y mover el .java a ese directorio.



Acceso a Base de Datos

- Cargar el driver
 - En nuestro caso usaremos la BD MySql, el driver se puede obtener desde : <http://www.mysql.org>
 - La carga del driver consiste en esta linea:
 - `Class.forName("com.mysql.jdbc.Driver").newInstance();`

Acceso a Base de Datos

- **La línea anterior es solamente válida para MySql.**
- Establecer una conexión
 - `String url = "jdbc:mysql://localhost"`
 - `Connection con = DriverManager.getConnection(url, "login", "pass") ;`

Acceso a Base de Datos

- Crear sentencias
 - Un objeto Statement es el que envía nuestras sentencias SQL al controlador de la base de datos.
 - Simplemente creamos un objeto Statement y lo ejecutamos, suministrando el método SQL apropiado con la sentencia SQL que queremos enviar.

Acceso a Base de Datos

- Para una sentencia SELECT, el método a ejecutar es executeQuery.
- Para sentencias que crean o modifican tablas, el método a utilizar es executeUpdate.
- Statement stmt = con.createStatement();
- Donde con es la conexión establecida anteriormente.

Acceso a Base de Datos

- Ejecutar sentencias
 - En este momento stmt existe, pero no tiene ninguna sentencia SQL que pasarle al controlador de la base de datos.
 - `stmt.executeQuery("SELECT * FROM tabla");`
 - `stmt.executeUpdate("INSERT INTO tabla VALUE");`

Acceso a Base de Datos

- Recuperar valores desde una tabla
 - JDBC devuelve los resultados en un objeto ResultSet.
 - `ResultSet rs = stmt.executeQuery("SELECT * FROM tabla");`
 - La variable rs, que es un ejemplar de ResultSet, contiene las filas de tabla.

Acceso a Base de Datos

- Para recorrer todas las filas existentes en rs, utilizamos el método next().
 - while(rs.next()) { // código }
 - Para recuperar una columna, se puede utilizar tanto el nombre de la columna como su posición.
 - rs.getString("nombre");
 - rs.getString(1);

Acceso a Base de Datos

- Las sentencias que podemos ejecutar pueden ser tan complejas como nos sea posible.
- La labor de los métodos getString, getFloat, etc., es transformar los valores de la base de datos a valores de JAVA.

Acceso a Base de Datos

- Utilización de transacciones
 - La decisión de utilizar o no las transacciones es un atributo de la conexión.
 - `con.setAutoCommit(false);`
 - Donde con es nuestro objeto Connection
 - `con.commit();`
 - `con.rollback();`

Acceso a Base de Datos

- Ejercicios