

# Conectar a base de datos MySQL con Java

Ángel Gabriel García Morones

Instituto de Educación Media y Superior María Esther Zuno de Echeverría (MEZE)

RESUMEN — Almacenando información en una Base de datos utilizado MySQL y creando consultas en entorno IDE Java.

## INTRODUCTION

En el desarrollo de una aplicación que captura información se requiere usualmente *conectar a base de datos, la BD que más auge tiene es MySQL*, pero existen otros manejadores de BD ,en mi caso utilizare MySQL cabe mencionar que puede variar algunos pasos en el proceso en cuanto en otros entornos, pero reitero el proceso es bastante similar, en algunas ocasiones varia solo datos de conexión, usare API JDBC del lenguaje de programación Java y como IDE haremos uso de NetBeans 8.2.

## BASE DE DATOS

Una base de datos (BD) o, mejor dicho, un sistema gestor de bases de datos (SGBD), es un software que gestiona, almacena gran cantidad de información y la vez proporciona seguridad.

Una vez teniendo en cuenta la definición (BD) nos concentraremos en el trabajo a realizar

## SERVIDOR MySQL

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.



Imagen 1 – logo MySQL

## BASE DE DATOS MYSQL CON JAVA

Lo primero que haremos será preparar el servidor y los datos de prueba, debes descargar e instalar en servidor [MySQL](#), también requieres a la aplicación [MySQL Workbench](#), esta última no es necesaria, pero nos facilitará el trabajo.

Realizare la creación de la tabla dentro del entorno MySQL Workbench con todos sus valores, admitiré la explicación del código ya que se debe tener noción de que está sucediendo.

```
/*mostrar tabla de una BD*/
• show tables;
/*mostrar todo las columnas de la tabla seleccion
• select * from user;

• use agenda;
• create table contacto (
  id INT auto_increment, primary key(id),
  Nombre varchar (20),
  apellidos varchar (40),
  telefonos varchar (20),
  ubicacion_casa int
);
```

Imagen 2- creación de la tabla

Una vez terminado con la creación de tablas y llenado pasaremos a el entorno IDE para dichas consultas.

## ENTORNO DE PROGRAMACION (IDE)

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de integrated development environment), es un entorno que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

### Java

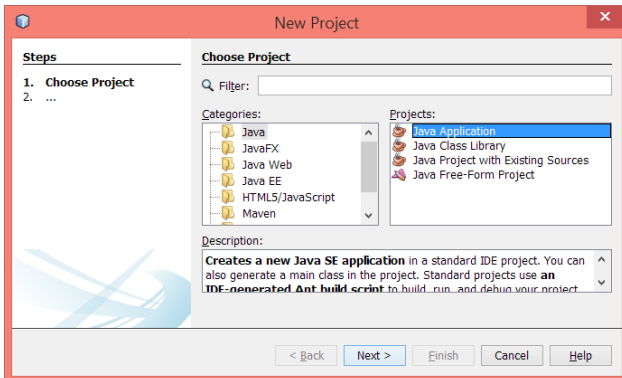
Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

### NetBeans

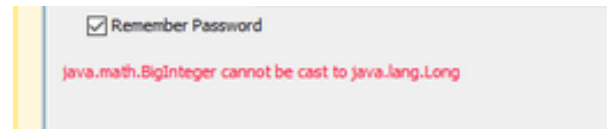
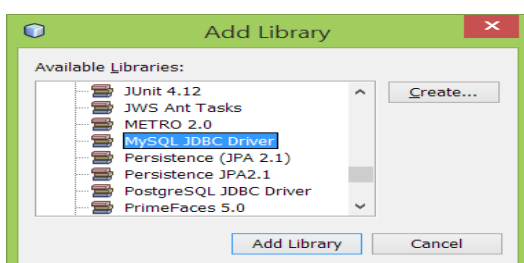
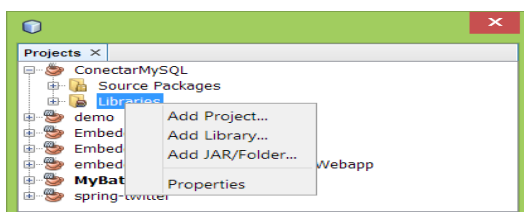
NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java.

Una vez teniendo en cuenta la definición (IDE) y (Java) nos concentraremos en el trabajo a realizar.

creamos el proyecto con Netbeans IDE, seleccionamos el tipo Java | Java Application.

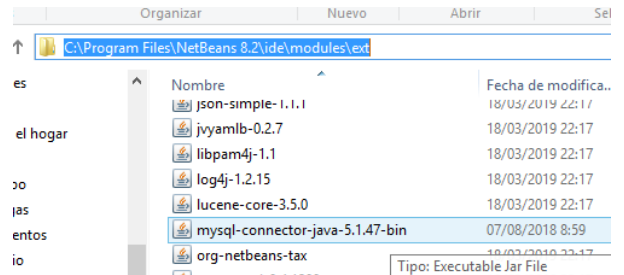


Una vez tengamos el proyecto damos clic derecho sobre Libraries y seleccionamos Add Library...

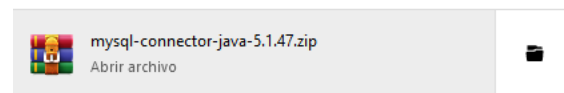


*Imagen de error puede que tu librería JDBC no soporte el numero de caracteres de datos que va a leer*

### SOLUCIÓN

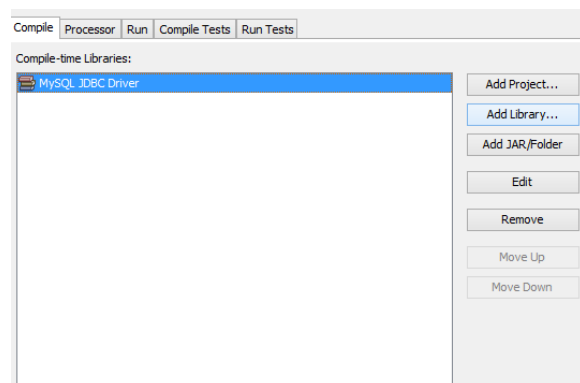


primero elimina archivo mysql-connector-java de la ruta C:\Program Files\NetBeans 8.2\ide\modules\ext



Después de eliminar la extensión, bajaremos una nueva desde la página oficial <https://dev.mysql.com/downloads/connector/j/5.1.html>, descomprime el archivo, y copia el archivo mysql-connector-java-5.1.46-bin en la ruta C:\Program Files\NetBeans 8.2\ide\modules\ext

En el NetBeans modificaremos la extensión



*Imagen muestra de agregar librería (extensión que fue descargada)*

*Nota: una vez borrada la extensión anterior se tornará letras en rojo en la extensión solo tienes que volver agregar y seleccionar la extensión descargada y guardada en la ruta C:\Program Files\NetBeans 8.2\ide\modules\ext.*

## Conectar a base de datos MySQL

Ahora ya podemos empezar a escribir el código Java requerido para realizar la conexión y consultar los datos.

Lo primero que veremos es un pequeño fragmento de código que nos permitirá saber si el driver está correctamente instalado, cuando no se pueda encontrar el driver se lanzará la excepción y mostramos el mensaje indicado.

```
try {
    Class.forName("com.mysql.jdbc.Driver").newInstance();
} catch (Exception ex) {
    System.out.println("Error, no se ha podido cargar MySQL JDBC Driver");
}
```

*Imagen-codigo de validacion del dirver en funcionamiento*

```
String url = "jdbc:mysql://localhost:3306/prueba";
String username = "root";
String password = "123456";

Connection connection = DriverManager.getConnection(url, username, password);
```

Primero indicamos la URL de conexión, para ello utilizamos una cadena de texto que tendrá la siguiente forma: jdbc:mysql://hostname:portNumber/databaseName, en donde hostname es el nombre del servidor, como estamos usando un servidor local indicaremos localhost, lo siguiente que debemos indicar es el puerto portNumber, en mi caso instale MySQL server en el puerto por defecto, este es 3306, y al final databaseName indica el nombre de la base de datos a la que nos conectaremos.

```
Statement statement = connection.createStatement();
ResultSet rs = statement.executeQuery("SELECT * FROM persona");
```

*Imagen Código de consulta.*

Lo que sigue es crear un objeto Statement que nos permitirá lanzar consultas contra la base de datos, para ello usamos el método createStatement() de la clase Connection.

El objeto ResultSet nos permite obtener los resultados de la consulta que ejecutamos con el método executeQuery(), en nuestro caso la misma selecciona todos los datos de la tabla persona.

```
while (rs.next()) {

    String id = rs.getString("id");
    String Nombre = rs.getString("nombre");
    String apellidos = rs.getString("apellidos");
    String telefonos = rs.getString("Telefonos");
    String ubicacion_casa = rs.getString("ubicacion_casa");

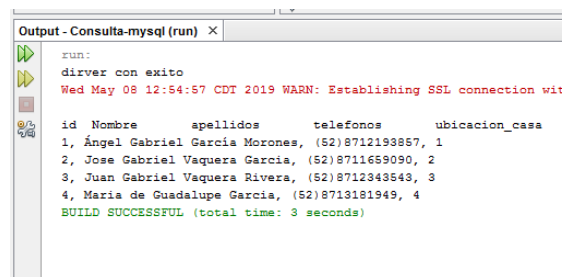
    System.out.println(String.format("%s, %s %s, %s, %s", id, Nombre, apellidos, telefonos, ubicacion_casa ));
}
```

En cada iteración obtenemos la columna correspondiente usando los diversos métodos getXxx(...), estos dependerán del tipo de la columna, por ejemplo, getInt(...) para obtener un entero o getString(...) para una cadena de texto, debemos indicarle el nombre de la columna que deseamos leer o también podemos usar su índice.

Al finalizar cerramos la conexión

```
rs.close();
statement.close();
connection.close();
```

Si ejecutamos la aplicación veremos el siguiente resultado, si todo está correcto.



```
run:
dirver con éxito
Wed May 08 12:54:57 CDT 2019 WARN: Establishing SSL connection with
id Nombre      apellidos      telefonos      ubicacion_casa
1, Ángel Gabriel Garcia Morones, (52)8712193857, 1
2, Jose Gabriel Vaquera Garcia, (52)8711659090, 2
3, Juan Gabriel Vaquera Rivera, (52)8712343543, 3
4, Maria de Guadalupe Garcia, (52)8713181949, 4
BUILD SUCCESSFUL (total time: 3 seconds)
```

## CONCLUSIÓN

Para realizar la clase conexión cabe mencionar que no se aplica en todos los entornos ide en este caso fue en uno específico al igual tomar en cuenta la versión de su sistema operativo obviamente la plataforma del S.O y la versión de la librería JDBC de java.