**JDBC**

**Java Data Base Connectivity**

Es la API que maneja Java para la conexión a bases de datos y poder ejecutar las sentencias hacia las mismas.

Está compuesto por las siguientes Clases o Componentes:

**DriverManager:** Nos permite crear una instancia de la conexión  
**Connection:** Genera la sesión, maneja todo el ciclo de vida de una sesión cuando nos conectamos a una base de datos.  
**Statement:** Nos ayuda a traer datos de una tabla.  
**PreparedStatement:** Hace lo mismo que Statement con la diferencia de que éste nos permite recibir parámetros para la clausula where.  
**ResultSet:** Es una interfaz que nos ayudará a manejar los datos obtenidos convirtiendo los datos en objetos.

Para generar un CRUD también tenemos métodos a partir de Statement/PreparedStatement:  
Para ejecutar un SELECT el método a usar será el: executeQuery(sqlString)  
Para ejecutar INSERT,UPDATE,DELETE or a DDL usaremos el método: executeUpdate(sqlString)  
Para ejecutar otros comandos de SQL usaremos: execute(sqlString).

Para comenzar a trabajar con JDBC primero se debe descargar el driver de la base de datos correspondiente, cada base de datos que soporta Java tiene su propio Driver.

1. Agregando el Driver al proyecto

Solo es necesario agregarlo a la carpeta de libs del proyecto

1. Agregarlo al Java Build Path , para que sea empaquetado

En eclipse en click derecho en el proyecto y luego en propiedades, luego en la sección de Libraries en la lista de Classpath , luego Add JARs…

Y luego refrescar el proyecto

Como buena práctica se debe crear un paquete donde contendrá todas las clases que conectan a la base de datos, Esto es lo que se conoce como **DAO (Data Access Object) .**

Se creara una interfaz que dara el esquema para la conexión a la base de datos.

import java.sql.Connection;

import java.sql.DriverManager;

import static DataBase.*\**;

public interface IDBConnection{

default *Connection* connectToDB(){

*Connection* connection = null;

try{

Class.forName("com.mysql.jdbc.Driver"); // Nombre completo que se utilizara para la conexión

connection = DiverManager.getConnection(URL+DB,USER,PASSWORD);

if(connection != null){

System.out.println("Se establecio la conexion");

}

} catch(*Exceiption* *ex*){

e.printStackTrace(); // esto solo para cuestiones de debug

}finally{

return connection;

}

}

}

/\* Crear todas las constantes que

de la base de datos

\*/

public class DataBase{

public static final *String* URL = "jdbc:mysql://localhost:3306/"; // especificar base de datos y su ubicacion

public static final *String* DB = "Nombre de la base de datos";

public static final *String* USER = "usuario de la base de datos";

public static final *String* PASSWORD = "password de la base de datos";

}

**Sentencia SELECT**

import java.sql.PreparedStatement;

import java.sql.ResultSet;

public interface MovieDAO extends *IDBConnection*{

default *ArrayList*<Movie> read(){

*ArrayList*<Movie> movies = **new** ArrayList();

try(*Connection* connection = connectToDB()){

*String* sqlQuery = "SELECT \* FROM Movie";

*PreparedStatement* preparedStatement = conneciton.preparedStatement(sqlQuery);

*ResultSet* rs = preparedStatement.executeQuery();

while (resultSet.next()) {

*Movie* movie = **new** Movie(

resultSet.getString(TMOVIE\_TITLE),

resultSet.getString(TMOVIE\_GENRE),

resultSet.getString(TMOVIE\_CREATOR),

resultSet.getInt(TMOVIE\_DURATION),

resultSet.getShort(TMOVIE\_YEAR));

movies.add(movie);

}

}catch(*SQLException* *ex*){

ex.printStackTrace();

}finally{

return movies;

}

}

}

**SELECT con parámetros para el Where**

private *boolean* getMovieViewed(*PreparedStatement* preparedStatement,

*Connection* connection,

*int* idMovie) {

*boolean* viewed = false;

*String* query = "SELECT \* FROM " + TVIEWED +

" WHERE " + TVIEWED\_IDMATERIAL + " = ?" +

" AND " + TVIEWED\_IDELEMENT + " = ?";

*ResultSet* resultSet = null;

try {

preparedStatement = connection.prepareStatement(query);

preparedStatement.setInt(1, *TMATERIALS\_ID*[0]);

/\*\*

// el numero que recibe por parametro sera la parte

que sera reemplaza en la consuta segun la posicion que

se le indique , es decir donde encuentre el ?

\*/

preparedStatement.setInt(2, idMovie);

resultSet = preparedStatement.executeQuery();

viewed = resultSet.next();

preparedStatement.close();

} catch (*Exception* *e*) {

e.printStackTrace();

} finally {

return viewed;

}

}

**INSERT insertar datos**

*String* insertTableSQL = "INSERT INTO DBUSER"

        + "(USER\_ID, USERNAME, CREATED\_BY, CREATED\_DATE) VALUES"

        + "(?,?,?,?)";

*PreparedStatement* preparedStatement = dbConnection.prepareStatement(insertTableSQL);

preparedStatement.setInt(1, 11);

preparedStatement.setString(2, "mkyong");

preparedStatement.setString(3, "system");

preparedStatement.setTimestamp(4, getCurrentTimeStamp());

// execute insert SQL stetement

preparedStatement .executeUpdate();

**UPDATE actulizar datos**

*String* updateTableSQL = "UPDATE DBUSER SET USERNAME = ? WHERE USER\_ID = ?";

*PreparedStatement* preparedStatement = dbConnection.prepareStatement(updateTableSQL);

preparedStatement.setString(1, "mkyong\_new\_value");

preparedStatement.setInt(2, 1001);

// execute insert SQL stetement

preparedStatement .executeUpdate();