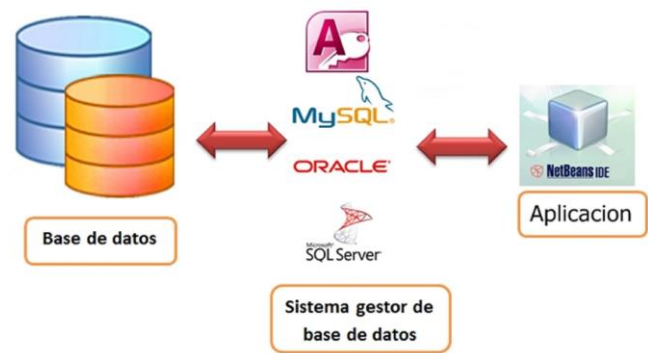


# LABORATORIO N°13

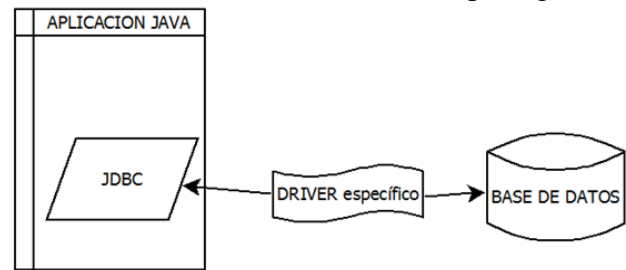
## ACCESO A DATOS

OBJETIVO: CREAR PROGRAMAS EN JAVA QUE ACCEDAN A UNA BASE DE DATOS MYSQL.

En acceso a datos, se requiere conectar una herramienta de programación con el motor de bases de datos que se desea acceder. En nuestro caso, debemos conectar Netbeanz con Mysql.



Las conexiones en JAVA se realizan a través de conexiones JDBC que significa Java Data Base Conection.



JDBC (Java DataBase Connectivity) es la API que permite la conexión de un programa Java y una base de datos relacional.

Se encuentra dentro del paquete java.sql.  
Incluye clases e interfaces que permiten el acceso a la bases de datos para ejecutar consultas, actualizaciones, ejecutar procedimientos, etc.  
Algunas de las clases e interfaces de JDBC son:

Clase / Interface	Función
Clase DriverManager	Establece la conexión con la base de datos
Interface Connection	Representa una conexión con la BD
Interface Statement	Ejecución de consultas SQL
Interface PreparedStatement	Ejecución de consultas preparadas y procedimientos almacenados
Interface ResultSet	Manipulación de registros en consultas de tipo Select
Interface ResultSetMetadata	Proporciona información sobre la estructura de los datos.

El objetivo de las interfaces de JDBC es definir como trabajar con la base de datos: como establecer la conexión, como ejecutar una consulta, etc.

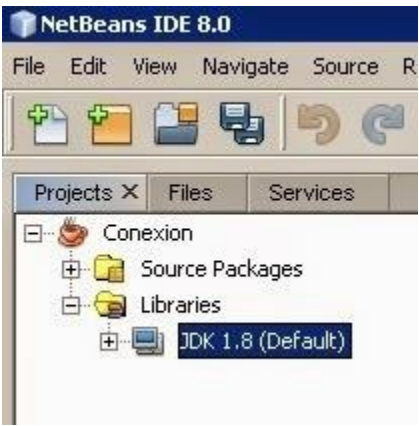
Existen otras conexiones de acceso a datos como ODBC (Open Daba Base Conection, que es de Windows), ADO (Access Data Object), OleDB, entre otras.

En nuestro caso necesitamos disponer de los siguientes requisitos:

- Ambiente de desarrollo (En este caso Netbeanz)
- JDK instalado.
- MySql Instalado
- mysql-connector-java-x.x.x-bin (este conector depende de la versión de mysql instalado, <http://www.mysql.com/downloads/>)
- Conocimientos Básicos de Programación en Java.

### Incluyendo el driver en Netbeanz

Se ingresa a Netbeans , luego barra de herramientas “Projects“.



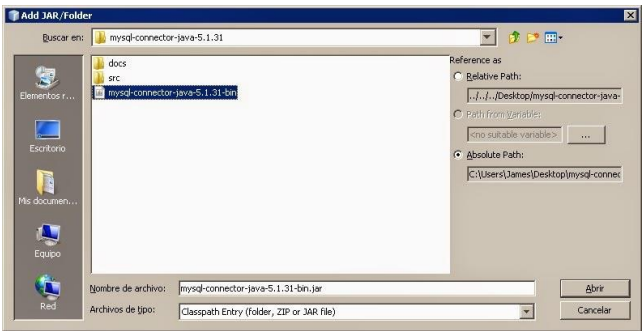
### Explorador de proyectos en Netbeans

clic derecho en la carpeta “Libraries” y presionar la opción “Add JAR/FOLDER...” .



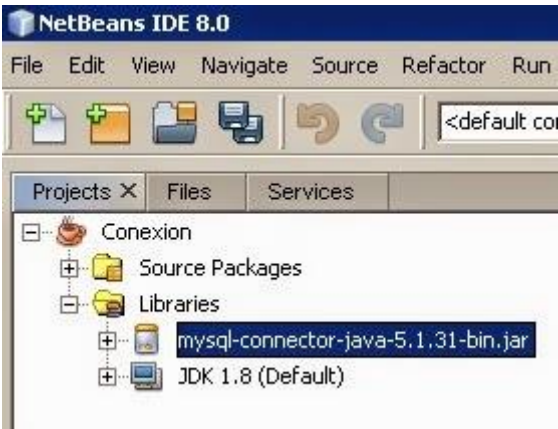
### Añadiendo un nuevo JAR al proyecto

buscar el archivo “mysql-connector-java-5.1.31-bin.jar” en la carpeta descomprimida.



### Seleccionando el conector

Cuando se haya incluido el archivo, el explorador de proyectos mostrará una nueva dependencia hacia el conector. Ahora puede conectar tu aplicación Java y una base de datos en Mysql.



## Nueva dependencia en nuestro proyecto

Esquema general de un programa en Java:

```
import java.sql.*;
public class EjemploAccesoBD1 {
    public static void main(String[] args) {
        Connection conexion = null;
        try {
            // Cargar el driver
            Class.forName("com.mysql.jdbc.Driver");

            // Se obtiene una conexión con la base de datos.
            // En este caso nos conectamos a la base de datos prueba
            // con el usuario root y contraseña 1daw
            conexion = DriverManager.getConnection("jdbc:mysql://localhost/prueba", "root", "1daw");

            // Se crea un Statement, para realizar la consulta
            Statement s = conexion.createStatement();

            // Se realiza la consulta. Los resultados se guardan en el ResultSet rs
            ResultSet rs = s.executeQuery("select * from persona");

            // Se recorre el ResultSet, mostrando por pantalla los resultados.
            while (rs.next()) {
                System.out.println(rs.getInt("Id") + " " + rs.getString(2) + " " + rs.getDate(3));
            }
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        } catch (ClassNotFoundException e) {
            System.out.println(e.getMessage());
        } finally { // Se cierra la conexión con la base de datos.
            try {
                if (conexion != null) {
                    conexion.close();
                }
            } catch (SQLException ex) {
                System.out.println(ex.getMessage());
            }
        }
    }
}
```

## ACTIVIDADES

1. Ejecutar el siguiente script SQL para crear la base de datos y tabla en Mysql

```
CREATE DATABASE Comida;
USE Comida;
CREATE TABLE Pizza(codigoPizza int primary key, valorPizza int, nombrePizza varchar(30));

INSERT INTO Pizza(codigoPizza, valorPizza, nombrePizza ) values (1, 2000, 'Pizza Americana');
INSERT INTO Pizza(codigoPizza, valorPizza, nombrePizza ) values (2, 4000, 'Pizza Vegetariana');
INSERT INTO Pizza(codigoPizza, valorPizza, nombrePizza ) values (3, 4500, 'Pizza Doble Queso');
```

2. Crear un programa en Java que permite leer los datos de la tabla Pizza

```
//Paso 1. Importar los paquetes requeridos
import java.sql.*;

public class SQL {
    // url del driver JDBC y la base de datos
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/Pizzeria";

    // usuario y contraseña de la base de datos
```

```

static final String USER = "root";
static final String PASS = "1234567";

public static void main(String[] args) {
    Connection conn = null;
    Statement stmt = null;
    try{
        //Paso 2: Cargar driver JDBC
        Class.forName("com.mysql.jdbc.Driver");

        //Paso 3: Abrir una conexion
        System.out.println("Conectando a la base de datos...");
        conn = DriverManager.getConnection(DB_URL,USER,PASS);

        //Paso 4: Ejecutar una consulta
        System.out.println("Creando declaracion...");
        stmt = conn.createStatement();
        String sql;
        sql = "SELECT * FROM Pizza";
        ResultSet rs = stmt.executeQuery(sql);

        //Paso 5: Extraer datos del resultado
        while(rs.next()){
            //Recuperar por el nombre de la columna
            int codigoPizza = rs.getInt("codigoPizza");
            int valorPizza = rs.getInt("valorPizza");
            String nombrePizza = rs.getString("nombrePizza");

            //Mostar resultados
            System.out.print("Codigo: " + codigoPizza);
            System.out.print(", Valor: " + valorPizza);
            System.out.println(", Nombre: " + nombrePizza);

        }
        //Paso 6: Limpiar
        rs.close();
        stmt.close();
        conn.close();
    }catch(SQLException se){
        //Errores de jdbc
        se.printStackTrace();
    }catch(Exception e){
        //Errores de Class.forName
        e.printStackTrace();
    }finally{
        //bloque usado para cerrar recursos
        try{
            if(stmt!=null)
                stmt.close();
        }catch(SQLException se2){
        }// nada que hacer
        try{
            if(conn!=null)
                conn.close();
        }catch(SQLException se){
            se.printStackTrace();
        }
    }
    System.out.println("Adios!");
}
}

```

3. Cree un programa en Java que permita insertar un nuevo registro

//Paso 1. Importar los paquetes requeridos

```

import java.sql.*;

public class sql {
    // url del driver JDBC y la base de datos
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/Pizzeria";

    // usuario y contraseña de la base de datos
    static final String USER = "root";
    static final String PASS = "1234567";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try{
            //Paso 2: Cargar driver JDBC
            Class.forName("com.mysql.jdbc.Driver");

            //Paso 3: Abrir una conexion
            System.out.println("Conectando a la base de datos...");
            conn = DriverManager.getConnection(DB_URL,USER,PASS);

            //Paso 4: Ejecutar insert
            System.out.println("Creando declaracion...");
            stmt = conn.createStatement();
            String sql;

            sql = "INSERT INTO Pizza " +
                "VALUES (4, 4000, 'Queso y pepperoni')";
            stmt.executeUpdate(sql);
            sql = "INSERT INTO Pizza " +
                "VALUES (5, 5000, 'Hawaiana')";
            stmt.executeUpdate(sql);
            sql = "INSERT INTO Pizza " +
                "VALUES(6, 6000, 'Pollo')";
            stmt.executeUpdate(sql);
            System.out.println("Insertados...");
            //Paso 5: Limpiar
            rs.close();
            stmt.close();
            conn.close();
        }catch(SQLException se){
            //Errores de jdbc
            se.printStackTrace();
        }catch(Exception e){
            //Errores de Class.forName
            e.printStackTrace();
        }finally{
            //bloque usado para cerrar recursos
            try{
                if(stmt!=null)
                    stmt.close();
            }catch(SQLException se2){
            }// nada que hacer
            try{
                if(conn!=null)
                    conn.close();
            }catch(SQLException se){
                se.printStackTrace();
            }
        }
        System.out.println("Adios!");
    }
}

```

4. Queda de tarea crear un programa en Java que permita eliminar un registro de la tabla