Guía de Laboratorio: Conexión a MySQL con JDBC

1. Introducción

Esta guía explica cómo conectar una aplicación Java a una base de datos MySQL utilizando JDBC. Se implementará un sistema de gestión de usuarios con operaciones CRUD (Crear, Leer, Actualizar y Eliminar).

2. Requisitos Previos

- MySQL instalado y en ejecución.
- JDK 17.
- Dependencia JDBC para MySQL.

3. Creación de la Base de Datos

Ejecuta los siguientes comandos en MySQL:

```
CREATE DATABASE mi_base;
USE mi_base;
CREATE TABLE usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    edad INT NOT NULL
);
```

4. Código Java

El siguiente código establece la conexión a MySQL y permite realizar operaciones CRUD:

```
import java.sql.*;
import java.util.Scanner;

public class UserManager {
    private static final String URL = "jdbc:mysql://localhost:3306/mi_base";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public static void main(String[] args) {
        try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
            Scanner scanner = new Scanner(System.in);
            boolean running = true;

            while (running) {
                System.out.println("\nGestión de Usuarios");
                System.out.println("1. Listar usuarios");
                System.out.println("2. Agregar usuario");
```

```
                System.out.println("3. Actualizar usuario");
                System.out.println("4. Eliminar usuario");
                System.out.println("5. Salir");
                System.out.print("Seleccione una opción: ");
                int option = scanner.nextInt();
                scanner.nextLine();

                switch (option) {
                    case 1:
                        listUsers(conn);
                        break;
                    case 2:
                        addUser(conn, scanner);
                        break;
                    case 3:
                        updateUser(conn, scanner);
                        break;
                    case 4:
                        deleteUser(conn, scanner);
                        break;
                    case 5:
                        running = false;
                        break;
                    default:
                        System.out.println("Opción no válida.");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private static void listUsers(Connection conn) throws SQLException {
        String sql = "SELECT * FROM usuarios";
        try (Statement stmt = conn.createStatement(); ResultSet rs =
stmt.executeQuery(sql)) {
            System.out.println("\nLista de Usuarios:");
            while (rs.next()) {
                System.out.println(rs.getInt("id") + ". " + rs.getString("nombre")
+ " - " + rs.getString("email") + " - " + rs.getInt("edad") + " años");
            }
        }
    }

    private static void addUser(Connection conn, Scanner scanner) throws
SQLException {
        System.out.print("Ingrese nombre: ");
        String nombre = scanner.nextLine();
        System.out.print("Ingrese email: ");
        String email = scanner.nextLine();
        System.out.print("Ingrese edad: ");
        int edad = scanner.nextInt();
        scanner.nextLine();
```

```
        String sql = "INSERT INTO usuarios (nombre, email, edad) VALUES (?, ?,
?)";
        try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, nombre);
            pstmt.setString(2, email);
            pstmt.setInt(3, edad);
            pstmt.executeUpdate();
            System.out.println("Usuario agregado correctamente.");
        }
    }

    private static void updateUser(Connection conn, Scanner scanner) throws
SQLException {
        System.out.print("Ingrese ID del usuario a actualizar: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Nuevo nombre: ");
        String nombre = scanner.nextLine();
        System.out.print("Nuevo email: ");
        String email = scanner.nextLine();
        System.out.print("Nueva edad: ");
        int edad = scanner.nextInt();
        scanner.nextLine();

        String sql = "UPDATE usuarios SET nombre = ?, email = ?, edad = ? WHERE id
= ?";
        try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, nombre);
            pstmt.setString(2, email);
            pstmt.setInt(3, edad);
            pstmt.setInt(4, id);
            int rows = pstmt.executeUpdate();
            if (rows > 0) {
                System.out.println("Usuario actualizado correctamente.");
            } else {
                System.out.println("Usuario no encontrado.");
            }
        }
    }

    private static void deleteUser(Connection conn, Scanner scanner) throws
SQLException {
        System.out.print("Ingrese ID del usuario a eliminar: ");
        int id = scanner.nextInt();
        scanner.nextLine();

        String sql = "DELETE FROM usuarios WHERE id = ?";
        try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setInt(1, id);
            int rows = pstmt.executeUpdate();
            if (rows > 0) {
                System.out.println("Usuario eliminado correctamente.");
            } else {
                System.out.println("Usuario no encontrado.");
```

```
            }
        }
    }
}
```

5. Ejecución del Proyecto

Caso 2: Gestión de Pedidos en un Restaurante

1. Creación de la Base de Datos

Ejecuta los siguientes comandos en MySQL:

```sql
CREATE DATABASE restaurante;
USE restaurante;

CREATE TABLE pedidos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cliente VARCHAR(100) NOT NULL,
    platillo VARCHAR(100) NOT NULL,
    precio DECIMAL(8,2) NOT NULL,
    estado ENUM('Pendiente', 'Preparando', 'Entregado') DEFAULT 'Pendiente'
);
```

---

2. Código en Java

Clase `DatabaseConnection` (Manejo de conexión)

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/restaurante";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}
```

Clase `Pedido` (Entidad del pedido)

```
public class Pedido {
    private int id;
    private String cliente;
    private String platillo;
    private double precio;
    private String estado;

    public Pedido(int id, String cliente, String platillo, double precio, String
estado) {
        this.id = id;
        this.cliente = cliente;
        this.platillo = platillo;
        this.precio = precio;
        this.estado = estado;
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Cliente: " + cliente + ", Platillo: " + platillo +
", Precio: " + precio + ", Estado: " + estado;
    }
}
```

Clase `PedidoDAO` (Acceso a datos)

```
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class PedidoDAO {
    public List<Pedido> listarPedidos() throws SQLException {
        List<Pedido> pedidos = new ArrayList<>();
        String query = "SELECT * FROM pedidos";
        try (Connection conn = DatabaseConnection.getConnection();
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(query)) {
            while (rs.next()) {
                pedidos.add(new Pedido(
                        rs.getInt("id"),
                        rs.getString("cliente"),
                        rs.getString("platillo"),
                        rs.getDouble("precio"),
                        rs.getString("estado")
                ));
            }
        }
        return pedidos;
    }

    public void agregarPedido(String cliente, String platillo, double precio)
```

```
throws SQLException {
        String query = "INSERT INTO pedidos (cliente, platillo, precio) VALUES (?,
?, ?)";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(query)) {
            pstmt.setString(1, cliente);
            pstmt.setString(2, platillo);
            pstmt.setDouble(3, precio);
            pstmt.executeUpdate();
            System.out.println("Pedido agregado con éxito.");
        }
    }

    public void actualizarEstadoPedido(int id, String estado) throws SQLException
{
        String query = "UPDATE pedidos SET estado = ? WHERE id = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(query)) {
            pstmt.setString(1, estado);
            pstmt.setInt(2, id);
            int rowsAffected = pstmt.executeUpdate();
            System.out.println(rowsAffected > 0 ? "Estado actualizado con éxito."
: "Pedido no encontrado.");
        }
    }

    public void eliminarPedido(int id) throws SQLException {
        String query = "DELETE FROM pedidos WHERE id = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(query)) {
            pstmt.setInt(1, id);
            int rowsAffected = pstmt.executeUpdate();
            System.out.println(rowsAffected > 0 ? "Pedido eliminado con éxito." :
"Pedido no encontrado.");
        }
    }
}
```

Clase `PedidoManager` (Interfaz de usuario)

```
import java.sql.SQLException;
import java.util.List;
import java.util.Scanner;

public class PedidoManager {
    public static void main(String[] args) {
        PedidoDAO pedidoDAO = new PedidoDAO();
        Scanner scanner = new Scanner(System.in);
        boolean running = true;

        while (running) {
```

```
            System.out.println("\nGestión de Pedidos");
            System.out.println("1. Listar pedidos");
            System.out.println("2. Agregar pedido");
            System.out.println("3. Actualizar estado de pedido");
            System.out.println("4. Eliminar pedido");
            System.out.println("5. Salir");
            System.out.print("Seleccione una opción: ");
            int option = scanner.nextInt();
            scanner.nextLine();

            try {
                switch (option) {
                    case 1:
                        List<Pedido> pedidos = pedidoDAO.listarPedidos();
                        pedidos.forEach(System.out::println);
                        break;
                    case 2:
                        System.out.print("Ingrese el nombre del cliente: ");
                        String cliente = scanner.nextLine();
                        System.out.print("Ingrese el platillo: ");
                        String platillo = scanner.nextLine();
                        System.out.print("Ingrese el precio: ");
                        double precio = scanner.nextDouble();
                        scanner.nextLine();
                        pedidoDAO.agregarPedido(cliente, platillo, precio);
                        break;
                    case 3:
                        System.out.print("Ingrese el ID del pedido a actualizar:
");
                        int idUpdate = scanner.nextInt();
                        scanner.nextLine();
                        System.out.print("Ingrese el nuevo estado (Pendiente,
Preparando, Entregado): ");
                        String estado = scanner.nextLine();
                        pedidoDAO.actualizarEstadoPedido(idUpdate, estado);
                        break;
                    case 4:
                        System.out.print("Ingrese el ID del pedido a eliminar: ");
                        int idDelete = scanner.nextInt();
                        scanner.nextLine();
                        pedidoDAO.eliminarPedido(idDelete);
                        break;
                    case 5:
                        running = false;
                        break;
                    default:
                        System.out.println("Opción no válida.");
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```