

Descripción del Proyecto

Este proyecto es una aplicación Java que utiliza JDBC (Java Database Connectivity) para interactuar con una base de datos MySQL. La aplicación permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en una tabla `users` dentro de la base de datos. Es un ejemplo sencillo y efectivo para comprender cómo Java puede conectarse y manipular bases de datos relacionales.

Instrucciones para Configurar y Ejecutar la Aplicación

Requisitos Previos

- JDK (Java Development Kit): Asegúrate de tener instalado JDK 8 o superior.
- IDE: Un entorno de desarrollo integrado como IntelliJ IDEA, Eclipse, o NetBeans.
- MySQL Server: Asegúrate de tener MySQL Server instalado y funcionando.
- Conector JDBC de MySQL: Descarga el archivo JAR del conector JDBC de MySQL desde [aquí](https://dev.mysql.com/downloads/connector/j/) (https://dev.mysql.com/downloads/connector/j/).

Crear la Base de Datos y la Tabla

1. Abre MySQL Workbench o cualquier cliente de MySQL.
2. Conéctate a tu servidor MySQL.
3. Ejecuta el siguiente script SQL para crear la base de datos y la tabla:

```
```sql
CREATE DATABASE testdb;
USE testdb;

CREATE TABLE users (
 id INT AUTO_INCREMENT PRIMARY KEY,
 name VARCHAR(50) NOT NULL,
 email VARCHAR(50) NOT NULL
);
```
```

Configuración del Proyecto

Eclipse

1. Crear un nuevo proyecto:
 - Abre Eclipse.
 - Ve a `File -> New -> Java Project`.
 - Dale un nombre a tu proyecto y haz clic en `Finish`.
2. Añadir el conector JDBC de MySQL:
 - Descarga el archivo JAR del conector JDBC de MySQL.
 - Añade el archivo JAR al proyecto:
 - Haz clic derecho en el proyecto en el Explorador de proyectos.
 - Selecciona `Build Path -> Configure Build Path`.
 - Ve a la pestaña `Libraries`.
 - Haz clic en `Add External JARs`.
 - Navega hasta donde descargaste el archivo JAR y selecciónalo.
 - Aplica los cambios y cierra la ventana.

Ejecutar el Programa

1. Ejecutar la clase principal:
 - Haz clic derecho en la clase `UserCRUD`.

- Selecciona `Run As -> Java Application`.

Si todo está configurado correctamente, deberías ver los mensajes en la consola indicando que las operaciones CRUD se han realizado con éxito.

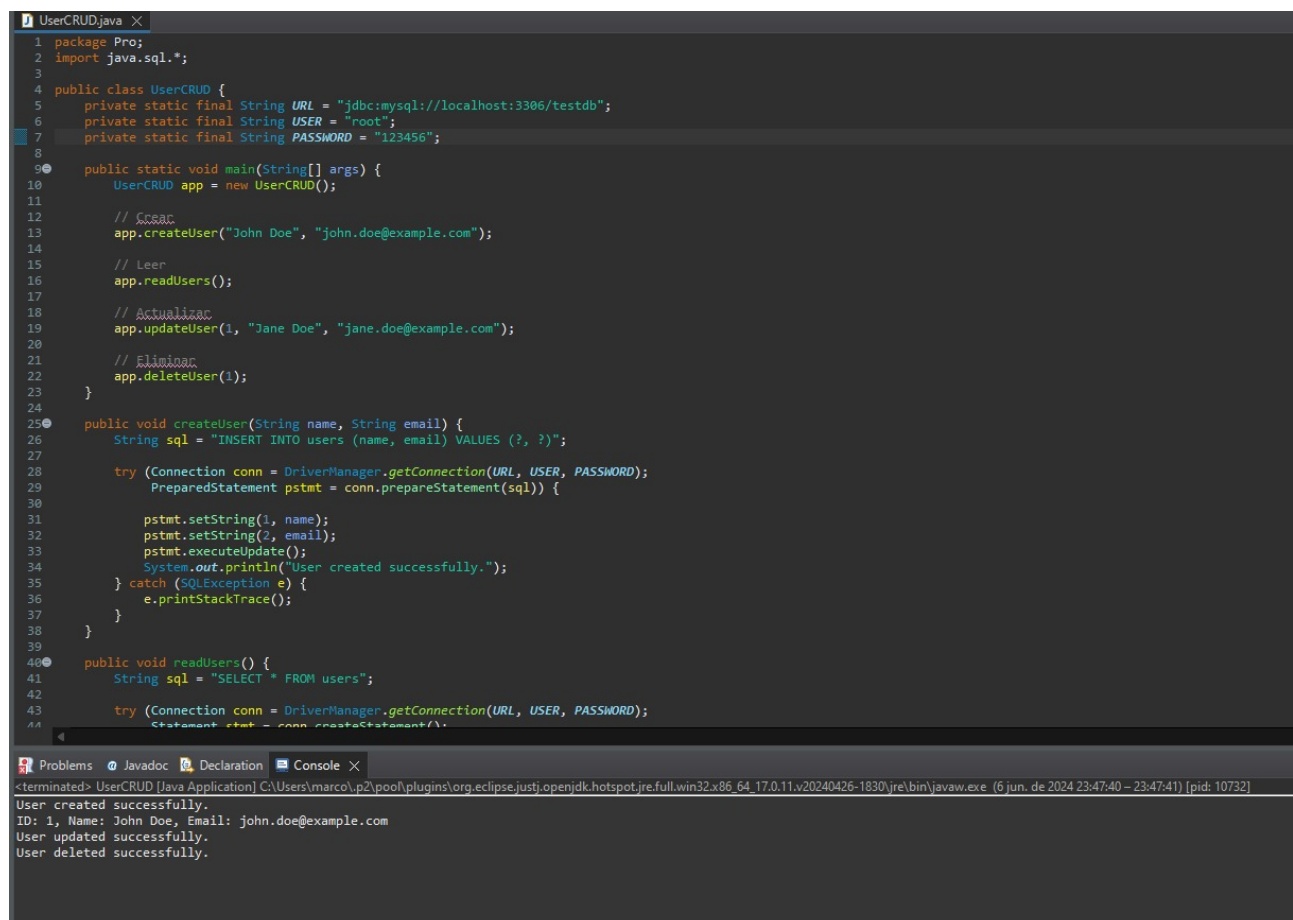
Explicación de las Funcionalidades Implementadas

Conexión a la Base de Datos

- URL de Conexión: La URL de conexión `jdbc:mysql://localhost:3306/testdb` especifica el protocolo JDBC, el nombre del host (localhost), el puerto (3306) y el nombre de la base de datos (testdb).
- Usuario y Contraseña: Las constantes `USER` y `PASSWORD` contienen las credenciales para acceder a la base de datos MySQL.

Métodos CRUD

- Crear Usuario (CREATE): El método `createUser` inserta un nuevo usuario en la tabla `users` usando una sentencia SQL `INSERT INTO`.
- Leer Usuarios (READ): El método `readUsers` obtiene todos los usuarios de la tabla `users` usando una sentencia SQL `SELECT` y los muestra en la consola.
- Actualizar Usuario (UPDATE): El método `updateUser` actualiza los detalles de un usuario existente basado en su ID usando una sentencia SQL `UPDATE`.
- Eliminar Usuario (DELETE): El método `deleteUser` elimina un usuario de la tabla `users` basado en su ID usando una sentencia SQL `DELETE`.



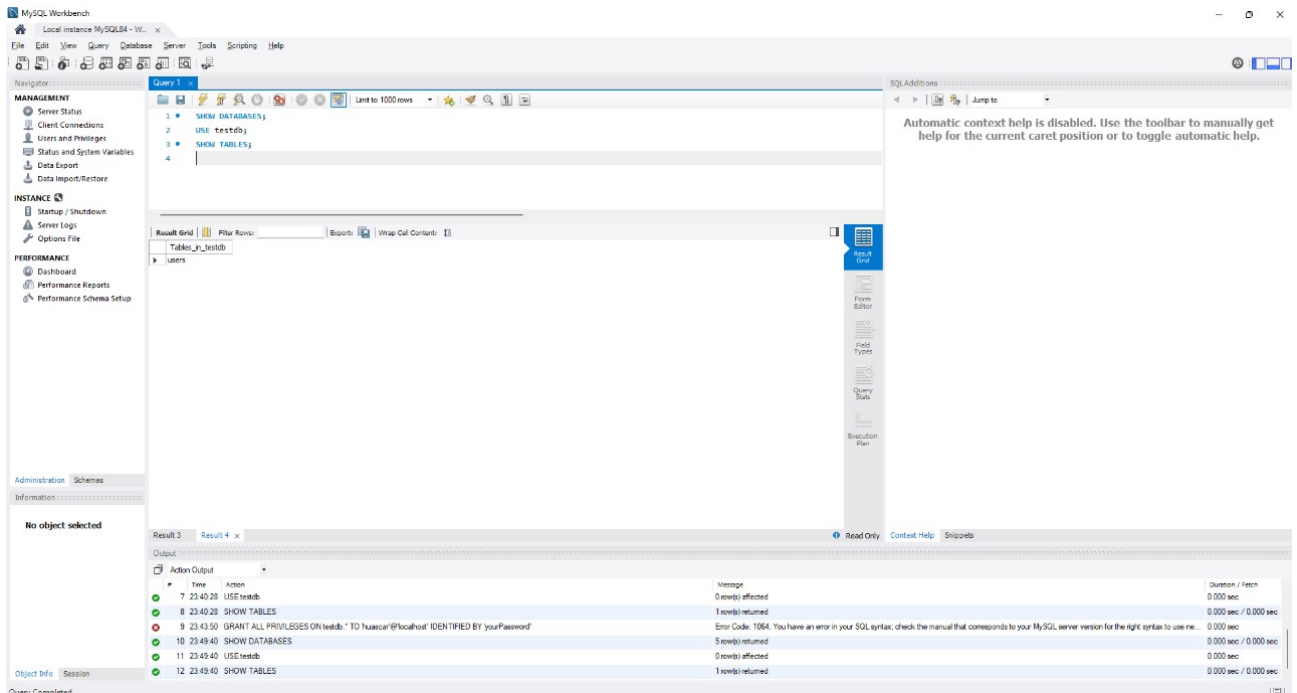
```
1 package Pro;
2 import java.sql.*;
3
4 public class UserCRUD {
5     private static final String URL = "jdbc:mysql://localhost:3306/testdb";
6     private static final String USER = "root";
7     private static final String PASSWORD = "123456";
8
9     public static void main(String[] args) {
10         UserCRUD app = new UserCRUD();
11
12         // Crear
13         app.createUser("John Doe", "john.doe@example.com");
14
15         // Leer
16         app.readUsers();
17
18         // Actualizar
19         app.updateUser(1, "Jane Doe", "jane.doe@example.com");
20
21         // Eliminar
22         app.deleteUser(1);
23     }
24
25     public void createUser(String name, String email) {
26         String sql = "INSERT INTO users (name, email) VALUES (?, ?)";
27
28         try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
29             PreparedStatement pstmt = conn.prepareStatement(sql)) {
30
31             pstmt.setString(1, name);
32             pstmt.setString(2, email);
33             pstmt.executeUpdate();
34             System.out.println("User created successfully.");
35         } catch (SQLException e) {
36             e.printStackTrace();
37         }
38     }
39
40     public void readUsers() {
41         String sql = "SELECT * FROM users";
42
43         try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
44             Statement stmt = conn.createStatement()) {
45
46             ResultSet rs = stmt.executeQuery(sql);
47             while (rs.next()) {
48                 int id = rs.getInt("id");
49                 String name = rs.getString("name");
50                 String email = rs.getString("email");
51                 System.out.println("ID: " + id + ", Name: " + name + ", Email: " + email);
52             }
53         }
54     }
55
56     public void updateUser(int id, String name, String email) {
57         String sql = "UPDATE users SET name = ?, email = ? WHERE id = ?";
58
59         try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
60             PreparedStatement pstmt = conn.prepareStatement(sql)) {
61
62             pstmt.setInt(1, id);
63             pstmt.setString(2, name);
64             pstmt.setString(3, email);
65             pstmt.executeUpdate();
66             System.out.println("User updated successfully.");
67         } catch (SQLException e) {
68             e.printStackTrace();
69         }
70     }
71
72     public void deleteUser(int id) {
73         String sql = "DELETE FROM users WHERE id = ?";
74
75         try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
76             PreparedStatement pstmt = conn.prepareStatement(sql)) {
77
78             pstmt.setInt(1, id);
79             pstmt.executeUpdate();
80             System.out.println("User deleted successfully.");
81         } catch (SQLException e) {
82             e.printStackTrace();
83         }
84     }
85 }
```

Problems Javadoc Declaration Console X

<terminated> UserCRUD [Java Application] C:\Users\marco\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.11.v20240426-1830\jre\bin\javaw.exe (6 jun. de 2024 23:47:40 - 23:47:41) [pid: 10732]

User created successfully.
ID: 1, Name: John Doe, Email: john.doe@example.com
User updated successfully.
User deleted successfully.

En la captura de pantalla se muestra las funciones CRUD y como las mismas son ejecutadas correctamente, generando el cambio deseado en la base de datos vinculada al programa



Se muestra la base de datos en MySQL después de generar los datos y haberlos eliminado con las funciones CRUD.