

Documentación Técnica del CRUD de Usuarios en Java y MySQL

Angie Tatiana Herran Corredor
ADSO - SENA

31 de mayo de 2025

1. Descripción General del Proyecto

Este proyecto implementa un CRUD (Crear, Leer, Actualizar, Eliminar) para la gestión de usuarios en un sistema desarrollado en Java utilizando MySQL como base de datos. El sistema permite registrar usuarios, buscar registros, actualizar datos de forma dinámica y eliminar usuarios de la base de datos.

2. Estructura de la Base de Datos

```
CREATE TABLE Usuario (  
    idUsuario INT PRIMARY KEY AUTOINCREMENT,  
    nombreUsuario VARCHAR(80) NOT NULL,  
    contraseña VARCHAR(30) NOT NULL,  
    rol ENUM( 'Tecnico', 'Analista', 'Administrador', 'Sysadmin' )  
    NOT NULL,  
    nombre VARCHAR(80) NOT NULL,  
    telefono VARCHAR(15) NOT NULL,  
    correoElectronico VARCHAR(100) NOT NULL,  
    numeroIdentificacion VARCHAR(12) UNIQUE NOT NULL,  
    estadoUsuario ENUM( 'Activo', 'Inactivo' ) DEFAULT 'Activo'  
);
```

3. Módulos y Funciones CRUD

3.1. Registrar Usuario

Método: insertarUsuario(Usuario usuario, Connection connection)

Entradas: Objeto Usuario con todos los campos completados.

Salidas: Inserta el registro en la tabla Usuario.

Pruebas: Se probó con datos de ejemplo, como:

- idUsuario: 5

- nombreUsuario: juanperez
- contrasena: segura123
- rol: Tecnico
- nombre: Juan Perez
- telefono: 3134567890
- correoElectronico: juan.perez@example.com
- numeroIdentificacion: 123456789012
- estadoUsuario: Activo

3.2. Buscar Usuarios

Método: `buscarUsuario(Connection connection)`

Descripción: Muestra todos los usuarios registrados, organizados en forma de tabla ordenada por el campo `idUsuario` de forma ascendente.

Entradas: Ninguna.

Salidas: Muestra los datos en consola, por ejemplo:

ID	Usuario	Contraseña	Rol
1	juanperez	segura123	Tecnico

Nombre	Teléfono
Juan Perez	3134567890

3.3. Actualizar Usuario

Método: `actualizarUsuario(int idUsuario, List<String>campos, List<String>valores, Connection connection)`

Entradas:

- `idUsuario`: ID del usuario a actualizar.
- `campos`: Lista de campos a actualizar (`nombreUsuario`, `contrasena`, etc.).
- `valores`: Lista de nuevos valores para esos campos.

Funcionamiento:

1. Se pregunta al usuario qué campos desea actualizar.
2. Se construye una consulta SQL dinámica según los campos seleccionados.
3. Se ejecuta la actualización en la base de datos.

Salidas: Actualización de los campos seleccionados en el registro correspondiente.

Pruebas: Actualización de rol, teléfono y correo de un usuario existente.

3.4. Eliminar Usuario

Método: `eliminarUsuario(int idUsuario, Connection connection)`

Entradas: `idUsuario` del usuario a eliminar.

Salidas: Elimina el usuario de la base de datos.

Pruebas: Eliminación exitosa de usuarios existentes. Mensaje en consola: **Usuario eliminado correctamente.**

4. Configuraciones del Entorno

- **Servidor MySQL:** MySQL 8.0.41
- **Conector JDBC:** `mysql-connector-java-9.2.0.jar`
- **Entorno de Desarrollo:** IntelliJ IDEA Community
- **Base de Datos:** Localhost, puerto 3306, base de datos `technical_visit`.

5. Manual Técnico de Uso

Registro de Usuario

Ejecutar el programa y responder a las preguntas solicitadas para llenar los datos del usuario.

Búsqueda de Usuarios

Los registros se muestran automáticamente después de cada operación.

Actualización de Usuario

El sistema pregunta qué campos desea actualizar. Ingresar solo los campos deseados y dejar los demás sin cambiar.

Eliminación de Usuario

Ingresar el `idUsuario` cuando el programa lo solicite.

6. Pruebas Realizadas

Pruebas Unitarias por Módulo

- **Insertar Usuario:** Inserción de datos completos con diferentes roles y estados.
- **Buscar Usuarios:** Visualización correcta de la tabla.
- **Actualizar Usuario:** Pruebas dinámicas con campos seleccionados.
- **Eliminar Usuario:** Eliminación exitosa de registros.

Resultados de Pruebas

Todos los módulos han sido verificados y cumplen con la funcionalidad esperada.

7. Código Principal

0.1. Archivo: Main.java

```
import java.sql.Connection;
import java.sql.SQLException;
import java.util.*;

public class Main {
    public static void main(String[] args) throws SQLException {
        Scanner lectorEntrada = new Scanner(System.in);
        System.out.println("====REGISTRAR NUEVO USUARIO====");
        System.out.print("Ingrese el ID del usuario: ");
        int idUsuario = lectorEntrada.nextInt();
        lectorEntrada.nextLine(); // Limpiar el buffer

        System.out.print("Ingrese el nombre de usuario: ");
        String nombreUsuario = lectorEntrada.nextLine();

        System.out.print("Ingrese la contraseña: ");
        String contrasena = lectorEntrada.nextLine();

        System.out.print("Ingrese el rol: ");
        String rol = lectorEntrada.nextLine();

        System.out.print("Ingrese el Nombre Completo: ");
        String nombre = lectorEntrada.nextLine();

        System.out.print("Ingrese el Teléfono: ");
        String telefono = lectorEntrada.nextLine();

        System.out.print("Ingrese el Correo Electrónico: ");
        String correoElectronico = lectorEntrada.nextLine();

        System.out.print("Ingrese el Número de Identificación: ");
        String numeroIdentificacion = lectorEntrada.nextLine();

        System.out.print("Ingrese el estado (Activo/Inactivo): ");
        String estadoUsuario = lectorEntrada.nextLine();

        Usuario objUsuario = new Usuario(idUsuario,
            nombreUsuario, contrasena, rol, nombre, telefono,
            correoElectronico, numeroIdentificacion, estadoUsuario);
        Connection connection = SQLConexion.connect();
```

```

UsuarioCRUD objUsuarioCRUD = new UsuarioCRUD();
objUsuarioCRUD.insertarUsuario(objUsuario, connection);
System.out.println(" Usuario registrado exitosamente.");

System.out.println(" *****TABLA-DE-USUARIOS*****");
objUsuarioCRUD.buscarUsuario(connection);

System.out.println
("=====ACTUALIZAR-DATOS-USUARIO=====");
System.out.print(" Ingrese el ID del usuario a actualizar: ");
int idActualizar = lectorEntrada.nextInt();
lectorEntrada.nextLine(); // Limpiar buffer

List<String> campos = new ArrayList<>();
List<String> valores = new ArrayList<>();

System.out.print(" Desea actualizar
-----nombre de Usuario? (s/n): ");
if (lectorEntrada.nextLine().equalsIgnoreCase("s")) {
    System.out.print(" Ingrese nombre de Usuario: ");
    campos.add("nombreUsuario");
    valores.add(lectorEntrada.nextLine());
}

System.out.print(" Desea actualizar la contrase a? (s/n): ");
if (lectorEntrada.nextLine().equalsIgnoreCase("s")) {
    System.out.print(" Ingrese Nueva contrasea: ");
    campos.add("contrasena");
    valores.add(lectorEntrada.nextLine());
}

System.out.print(" Desea actualizar el rol? (s/n): ");
if (lectorEntrada.nextLine().equalsIgnoreCase("s")) {
    System.out.print(" Ingrese Nuevo rol
----- (Tecnico/Analista/Administrador/Sysadmin): ");
    campos.add("rol");
    valores.add(lectorEntrada.nextLine());
}

System.out.print(" Desea actualizar su nombre? (s/n): ");
if (lectorEntrada.nextLine().equalsIgnoreCase("s")) {
    System.out.print(" Ingrese nombre: ");
    campos.add("nombre");
    valores.add(lectorEntrada.nextLine());
}

System.out.print(" Desea actualizar su
-----No. de telefono? (s/n): ");

```

```

        if (lectorEntrada.nextLine().equalsIgnoreCase("s")) {
            System.out.print(" Ingrese -Nuevo-No.-de-telefono:-");
            campos.add("telefono");
            valores.add(lectorEntrada.nextLine());
        }

        System.out.print(" Desea -actualizar-el-correo
-----Electronico?(s/n):-");
        if (lectorEntrada.nextLine().equalsIgnoreCase("s")) {
            System.out.print(" Ingrese -Nuevo-correo-Electronico:-");
            campos.add("correoElectronico");
            valores.add(lectorEntrada.nextLine());
        }

        System.out.print(" Desea -actualizar
-----estado-de-Usuario?(s/n):-");
        if (lectorEntrada.nextLine().equalsIgnoreCase("s")) {
            System.out.print(" Ingrese -Nuevo-estado-de-Usuario
------(Activo/Inactivo):-");
            campos.add("estadoUsuario");
            valores.add(lectorEntrada.nextLine());
        }

        if (campos.isEmpty()) {
            System.out.println("No-se-seleccionaron-campos
-----para-actualizar.");
        } else {
            objUsuarioCRUD.actualizarUsuario(idActualizar,
            campos, valores, connection);
        }

        System.out.println("*****-TABLA-DE-USUARIOS-*****");
        objUsuarioCRUD.buscarUsuario(connection);

        System.out.println("\nPRUEBA-ELIMINAR-REGISTRO-USUARIO");
        System.out.print(" Ingrese -el-ID-del-usuario
-----para-eliminar:-");
        int idEliminar = lectorEntrada.nextInt();
        lectorEntrada.nextLine();
        objUsuarioCRUD.eliminarUsuario(idEliminar, connection);

        System.out.println("*****-TABLA-DE-USUARIOS-*****");
        objUsuarioCRUD.buscarUsuario(connection);

        connection.close();
    }
}

```

8. Código CRUD

0.2. Archivo: UsuarioCRUD.java

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;

public class UsuarioCRUD {
    public void insertarUsuario(Usuario usuario,
        Connection connection) throws SQLException {
        String instructionStorage = "INSERT-INTO- usuario -(idUserio ,
-----nombreUsuario , -contrasena , -rol , -nombre , -telefono ,
-----correoElectronico , -numeroIdentificacion , -estadoUsuario)
-----VALUES-(? ,? ,? ,? ,? ,? ,? ,? ,? ,? );";
        PreparedStatement statement = connection.prepareStatement
            (instructionStorage);
        statement.setInt(1, usuario.getIdUsuario());
        statement.setString(2, usuario.getNombreUsuario());
        statement.setString(3, usuario.getContrasena());
        statement.setString(4, usuario.getRol());
        statement.setString(5, usuario.getNombre());
        statement.setString(6, usuario.getTelefono());
        statement.setString(7, usuario.getCorreoElectronico());
        statement.setString(8, usuario.getNumeroIdentificacion());
        statement.setString(9, usuario.getEstadoUsuario());
        statement.executeUpdate();
        statement.close();
    }

    public void buscarUsuario(Connection connection) throws
        SQLException {
        String searchIntruccion = "SELECT-idUsuario , -nombreUsuario ,
-----contrasena , -rol , -nombre , -telefono , -correoElectronico ,
-----numeroIdentificacion , -estadoUsuario -FROM- usuario -ORDER-BY
-----idUserio -ASC;";
        PreparedStatement statement = connection.prepareStatement
            (searchIntruccion);
        ResultSet resultSet = statement.executeQuery();

        System.out.printf(" %-10s %-15s %-21s %-15s %-30s
----- %-15s %-25s %-20s %-10s%n", "ID",
            "Usuario", "Contrase a", "Rol", "Nombre", "Tel fono",
            "Correo", "Identificaci n", "Estado");
        System.out.println("-----");
    }
}
```

```

while (resultSet.next()){
    int idUsuario =resultSet.getInt("idUsuario");
    String nombreUsuario =resultSet.getNString
("nombreUsuario");
    String contrasena =resultSet.getNString
("contrasena");
    String rol =resultSet.getNString("rol");
    String nombre =resultSet.getNString("nombre");
    String telefono =resultSet.getNString("telefono");
    String correoElectronico =resultSet.getNString
("correoElectronico");
    String numeroIdentificacion =resultSet.getNString
("numeroIdentificacion");
    String estadoUsuario =resultSet.getNString
("estadoUsuario");
    System.out.printf(" %-10d- %-15s- %-21s
----- %-15s- %-30s- %-15s- %-25s- %-20s- %-10s%n" ,
idUsuario , nombreUsuario , contrasena , rol , nombre ,
telefono , correoElectronico , numeroIdentificacion ,
estadoUsuario);
}

resultSet.close();
statement.close();
}

public void actualizarUsuario(int idUsuario ,
List<String> campos , List<String> valores ,
Connection connection) throws SQLException
{
    StringBuilder sql = new StringBuilder
("UPDATE Usuario SET ");
    for (int i = 0; i < campos.size(); i++) {
        sql.append(campos.get(i)).append(" =?");
        if (i < campos.size() - 1) {
            sql.append(", ");
        }
    }
    sql.append(" WHERE idUsuario =?");
    PreparedStatement statement =
connection.prepareStatement(sql.toString());
    for (int i = 0; i < valores.size(); i++) {
        statement.setString(i + 1, valores.get(i));
    }
    statement.setInt(valores.size() + 1, idUsuario);
    int rowsUpdated = statement.executeUpdate();
    if (rowsUpdated > 0) {
        System.out.println

```



```

        ("\\n-Usuario-actualizado-exitosamente.");
    } else {
        System.out.println
            ("\\n-No-se-encontr -el-usuario-para
-----actualizar.");
    }
    statement.close();
}

public void eliminarUsuario(int idUsuario, Connection
connection)
throws SQLException {
    String eliminarSQL = "DELETE-FROM- Usuario
-----WHERE-idUsuario-=?";
    PreparedStatement statement = connection.prepareStatement
        (eliminarSQL);
    statement.setInt(1, idUsuario);
    int filasAfectadas = statement.executeUpdate();
    if (filasAfectadas > 0) {
        System.out.println(" Usuario-eliminado
-----correctamente.");
    } else {
        System.out.println("No-se-encontr -el
-----usuario-con-ese-ID.");
    }
    statement.close();
}
}

```