[explicación conexión-20250305\_225208-Grabación de la reunión.mp4](https://uniminuto0-my.sharepoint.com/:v:/g/personal/jorman_chaves_uniminuto_edu_co/Ec4MxjxsaixNpKDwhz7NVkkB-zWCHHF4RkolXHGBHC8UcA?e=2VQNZ7)

/\*

\* Clase para manejar la conexión a una base de datos MySQL.

\*/

package database; // Define el paquete donde se encuentra la clase

import java.sql.Connection; // Importa la clase Connection para manejar la conexión con la base de datos

import java.sql.DriverManager; // Importa la clase DriverManager para gestionar los controladores de bases de datos

import java.sql.SQLException; // Importa la clase SQLException para manejar errores de SQL

import javax.swing.JOptionPane; // Importa la clase JOptionPane para mostrar mensajes emergentes

public class Conexion {

// Definición de constantes para la conexión a la base de datos

private final String DRIVER="com.mysql.jdbc.Driver"; // Especifica el driver de conexión para MySQL

private final String URL="jdbc:mysql://localhost:3306/"; // URL base del servidor MySQL

private final String DB="dbsistemaventas"; // Nombre de la base de datos a conectar

private final String USER="root"; // Usuario de la base de datos

private final String PASSWORD=""; // Contraseña de la base de datos (vacía por defecto)

public Connection cadena; // Variable para almacenar la conexión a la base de datos

public static Conexion instancia; // Instancia única de la clase (patrón Singleton)

// Constructor privado para evitar instanciación directa

private Conexion(){

this.cadena=null; // Inicializa la conexión como nula

}

// Método para establecer la conexión con la base de datos

public Connection conectar(){

try {

Class.forName(DRIVER); // Carga el driver de MySQL

this.cadena=DriverManager.getConnection(URL+DB,USER,PASSWORD); // Establece la conexión con la base de datos

} catch (ClassNotFoundException | SQLException e) { // Captura excepciones si hay errores en la conexión

JOptionPane.showMessageDialog(null, e.getMessage()); // Muestra un mensaje de error en caso de fallo

System.exit(0); // Finaliza la ejecución del programa si la conexión falla

}

return this.cadena; // Retorna el objeto Connection

}

// Método para cerrar la conexión con la base de datos

public void desconectar(){

try {

this.cadena.close(); // Cierra la conexión con la base de datos

} catch (SQLException e) { // Captura excepciones en caso de error al cerrar la conexión

JOptionPane.showMessageDialog(null, e.getMessage()); // Muestra un mensaje de error si la desconexión falla

}

}

// Método para obtener la instancia única de la clase (patrón Singleton)

public synchronized static Conexion getInstancia(){

if (instancia==null){ // Verifica si la instancia aún no ha sido creada

instancia=new Conexion(); // Crea una nueva instancia si no existe

}

return instancia; // Retorna la única instancia de la clase

}

}

Texto

El contenido generado por IA puede ser incorrecto.