**AE-2.**

**Manejo de conectores de BBDD**

**Ivan Nuñez Rodriguez**

2º DAM

Acceso a Datos

**Índice:**

1. Código.
2. Creación Base de Datos
3. Ejecución y resultado.
4. Enlace a GitHub.

**Código:**

Clase CocheFun

package controller;  
  
import dao.CocheDAO;  
import model.Coche;  
  
import java.sql.SQLException;  
import java.util.Scanner;  
  
public class CocheFun {  
 private final CocheDAO cocheDAO = new CocheDAO();  
  
 public void nuevoCoche(Scanner scanner) {  
 System.*out*.print("Marca del coche: ");  
 String marca = scanner.next();  
 System.*out*.print("Modelo del coche: ");  
 String modelo = scanner.next();  
 System.*out*.print("Año del coche: ");  
 int anio = scanner.nextInt();  
  
 try {  
 cocheDAO.nuevoCoche(new Coche(0, marca, modelo, anio));  
 System.*out*.println("Coche añadido correctamente.");  
 } catch (SQLException e) {  
 System.*out*.println("Error al añadir coche: " + e.getMessage());  
 }  
 }  
  
 public void borrarCoche(Scanner scanner) {  
 System.*out*.print("ID del coche a borrar: ");  
 int id = scanner.nextInt();  
  
 try {  
 cocheDAO.borrarCoche(id);  
 System.*out*.println("Coche borrado correctamente.");  
 } catch (SQLException e) {  
 System.*out*.println("Error al borrar coche: " + e.getMessage());  
 }  
 }  
  
 public void buscarCoche(Scanner scanner) {  
 System.*out*.print("ID del coche a consultar: ");  
 int id = scanner.nextInt();  
  
 try {  
 Coche coche = cocheDAO.buscaCochePorId(id);  
 if (coche != null) {  
 System.*out*.println("Coche encontrado: " + coche);  
 } else {  
 System.*out*.println("Coche no encontrado.");  
 }  
 } catch (SQLException e) {  
 System.*out*.println("Error al consultar coche: " + e.getMessage());  
 }  
 }  
  
 public void modificarCoche(Scanner scanner) {  
 System.*out*.print("ID del coche a modificar: ");  
 int id = scanner.nextInt();  
  
 System.*out*.print("Nueva marca: ");  
 String marca = scanner.next();  
 System.*out*.print("Nuevo modelo: ");  
 String modelo = scanner.next();  
 System.*out*.print("Nuevo año: ");  
 int anio = scanner.nextInt();  
  
 try {  
 cocheDAO.actulizaCoche(id, marca, modelo, anio);  
 System.*out*.println("Coche actualizado correctamente.");  
 } catch (SQLException e) {  
 System.*out*.println("Error al modificar coche: " + e.getMessage());  
 }  
 }  
  
 public void listarCoches() {  
 try {  
 cocheDAO.listarCoches().forEach(System.*out*::println);  
 } catch (SQLException e) {  
 System.*out*.println("Error al listar coches: " + e.getMessage());  
 }  
 }  
}

clase PasajeroFun

package controller;  
  
import dao.PasajeroDAO;  
import model.Pasajero;  
  
import java.sql.SQLException;  
import java.util.Scanner;  
  
public class PasajeroFun {  
 private final PasajeroDAO pasajeroDAO = new PasajeroDAO();  
  
 public void nuevoPasajero(Scanner scanner) {  
 System.*out*.print("Nombre del pasajero: ");  
 String nombre = scanner.next();  
 System.*out*.print("Edad del pasajero: ");  
 int edad = scanner.nextInt();  
 System.*out*.print("Peso del pasajero: ");  
 double peso = scanner.nextDouble();  
  
 try {  
 pasajeroDAO.añadirPasajero(new Pasajero(0, nombre, edad, peso));  
 System.*out*.println("Pasajero añadido correctamente.");  
 } catch (SQLException e) {  
 System.*out*.println("Error al añadir pasajero: " + e.getMessage());  
 }  
 }  
  
 public void borrarPasajero(Scanner scanner) {  
 System.*out*.print("ID del pasajero a borrar: ");  
 int id = scanner.nextInt();  
  
 try {  
 pasajeroDAO.borrarPasajero(id);  
 System.*out*.println("Pasajero borrado correctamente.");  
 } catch (SQLException e) {  
 System.*out*.println("Error al borrar pasajero: " + e.getMessage());  
 }  
 }  
  
 public void consultarPasajero(Scanner scanner) {  
 System.*out*.print("ID del pasajero a consultar: ");  
 int id = scanner.nextInt();  
  
 try {  
 Pasajero pasajero = pasajeroDAO.getPasajeroById(id);  
 if (pasajero != null) {  
 System.*out*.println("Pasajero encontrado: " + pasajero);  
 } else {  
 System.*out*.println("Pasajero no encontrado.");  
 }  
 } catch (SQLException e) {  
 System.*out*.println("Error al consultar pasajero: " + e.getMessage());  
 }  
 }  
  
 public void listarPasajeros() {  
 try {  
 pasajeroDAO.listarPasajeros().forEach(System.*out*::println);  
 } catch (SQLException e) {  
 System.*out*.println("Error al listar pasajeros: " + e.getMessage());  
 }  
 }  
  
 public void añadirPasajeroACoche(Scanner scanner) {  
 System.*out*.print("ID del pasajero: ");  
 int pasajeroId = scanner.nextInt();  
 System.*out*.print("ID del coche: ");  
 int cocheId = scanner.nextInt();  
  
 try {  
 pasajeroDAO.añadirPasajeroEnCoche(pasajeroId, cocheId);  
 System.*out*.println("Pasajero añadido al coche correctamente.");  
 } catch (SQLException e) {  
 System.*out*.println("Error al añadir pasajero al coche: " + e.getMessage());  
 }  
 }  
  
 public void eliminarPasajeroDeCoche(Scanner scanner) {  
 System.*out*.print("ID del pasajero: ");  
 int pasajeroId = scanner.nextInt();  
 System.*out*.print("ID del coche: ");  
 int cocheId = scanner.nextInt();  
  
 try {  
 pasajeroDAO.borrarPasajeroEnCoche(pasajeroId, cocheId);  
 System.*out*.println("Pasajero eliminado del coche correctamente.");  
 } catch (SQLException e) {  
 System.*out*.println("Error al eliminar pasajero del coche: " + e.getMessage());  
 }  
 }  
  
 public void listarPasajerosDeCoche(Scanner scanner) {  
 System.*out*.print("ID del coche: ");  
 int cocheId = scanner.nextInt();  
  
 try {  
 var pasajeros = pasajeroDAO.listarPasajerosPorCoche(cocheId);  
 if (pasajeros.isEmpty()) {  
 System.*out*.println("El coche con ID " + cocheId + " no tiene pasajeros asociados.");  
 } else {  
 pasajeros.forEach(System.*out*::println);  
 }  
 } catch (SQLException e) {  
 System.*out*.println("Error al listar pasajeros del coche: " + e.getMessage());  
 }  
 }  
}

clase cocheDao

package dao;  
  
import database.DatabaseConnection;  
import database.DatabaseSchema;  
import model.Coche;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.List;  
  
public class CocheDAO {  
 private Connection connection;  
 private PreparedStatement prepareStatement;  
 private ResultSet resultSet;  
  
 public CocheDAO() {  
 connection = DatabaseConnection.*getConnection*();  
 }  
  
 public void nuevoCoche(Coche coche) throws SQLException {  
 String query = "INSERT INTO " + DatabaseSchema.*TABLE\_COCHES* + " (" +  
 DatabaseSchema.*COLUMN\_COCHE\_MARCA* + ", " +  
 DatabaseSchema.*COLUMN\_COCHE\_MODELO* + ", " +  
 DatabaseSchema.*COLUMN\_COCHE\_ANIO* + ") VALUES (?, ?, ?)";  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setString(1, coche.getMarca());  
 prepareStatement.setString(2, coche.getModelo());  
 prepareStatement.setInt(3, coche.getAnio());  
 prepareStatement.executeUpdate();  
  
 }  
  
  
 public Coche buscaCochePorId(int id) throws SQLException {  
 String query = "SELECT \* FROM " + DatabaseSchema.*TABLE\_COCHES* +  
 " WHERE " + DatabaseSchema.*COLUMN\_COCHE\_ID* + " = ?";  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setInt(1, id);  
 resultSet = prepareStatement.executeQuery();  
 if (resultSet.next()) {  
 return new Coche(resultSet.getInt(DatabaseSchema.*COLUMN\_COCHE\_ID*),  
 resultSet.getString(DatabaseSchema.*COLUMN\_COCHE\_MARCA*),  
 resultSet.getString(DatabaseSchema.*COLUMN\_COCHE\_MODELO*),  
 resultSet.getInt(DatabaseSchema.*COLUMN\_COCHE\_ANIO*));  
 }  
 return null;  
 }  
  
  
 public void actulizaCoche(int id, String marca, String modelo, int anio) throws SQLException {  
 String query = "UPDATE " + DatabaseSchema.*TABLE\_COCHES* + " SET " +  
 DatabaseSchema.*COLUMN\_COCHE\_MARCA* + " = ?, " +  
 DatabaseSchema.*COLUMN\_COCHE\_MODELO* + " = ?, " +  
 DatabaseSchema.*COLUMN\_COCHE\_ANIO* + " = ? WHERE " + DatabaseSchema.*COLUMN\_COCHE\_ID* + " = ?";  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setString(1, marca);  
 prepareStatement.setString(2, modelo);  
 prepareStatement.setInt(3, anio);  
 prepareStatement.setInt(4, id);  
 prepareStatement.executeUpdate();  
  
 }  
  
  
 public void borrarCoche(int id) throws SQLException {  
 String query = "DELETE FROM " + DatabaseSchema.*TABLE\_COCHES* +  
 " WHERE " + DatabaseSchema.*COLUMN\_COCHE\_ID* + " = ?";  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setInt(1, id);  
 prepareStatement.executeUpdate();  
  
 }  
  
  
 public ArrayList<Coche> listarCoches() throws SQLException {  
 ArrayList<Coche> coches = new ArrayList<>();  
 String query = "SELECT \* FROM " + DatabaseSchema.*TABLE\_COCHES*;  
  
 prepareStatement = connection.prepareStatement(query);  
 resultSet = prepareStatement.executeQuery();  
 while (resultSet.next()) {  
 coches.add(new Coche(resultSet.getInt(DatabaseSchema.*COLUMN\_COCHE\_ID*),  
 resultSet.getString(DatabaseSchema.*COLUMN\_COCHE\_MARCA*),  
 resultSet.getString(DatabaseSchema.*COLUMN\_COCHE\_MODELO*),  
 resultSet.getInt(DatabaseSchema.*COLUMN\_COCHE\_ANIO*)));  
 }  
  
 return coches;  
 }  
}

clase pasajeroDao

package dao;  
  
import database.DatabaseConnection;  
import database.DatabaseSchema;  
import model.Pasajero;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.List;  
  
public class PasajeroDAO {  
  
 private Connection connection;  
 private PreparedStatement prepareStatement;  
 // statement  
 private ResultSet resultSet;  
  
 // constructor por defecto  
 public PasajeroDAO() {  
 connection = DatabaseConnection.*getConnection*();  
 }  
  
 public void añadirPasajero(Pasajero pasajero) throws SQLException {  
 String query = "INSERT INTO " + DatabaseSchema.*TABLE\_PASAJEROS* + " (" +  
 DatabaseSchema.*COLUMN\_PASAJERO\_NOMBRE* + ", " +  
 DatabaseSchema.*COLUMN\_PASAJERO\_EDAD* + ", " +  
 DatabaseSchema.*COLUMN\_PASAJERO\_PESO* + ") VALUES (?, ?, ?)";  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setString(1, pasajero.getNombre());  
 prepareStatement.setInt(2, pasajero.getEdad());  
 prepareStatement.setDouble(3, pasajero.getPeso());  
 prepareStatement.executeUpdate();  
  
 }  
  
 public Pasajero getPasajeroById(int id) throws SQLException {  
 String query = "SELECT \* FROM " + DatabaseSchema.*TABLE\_PASAJEROS* +  
 " WHERE " + DatabaseSchema.*COLUMN\_PASAJERO\_ID* + " = ?";  
  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setInt(1, id);  
 resultSet = prepareStatement.executeQuery();  
 if (resultSet.next()) {  
 return new Pasajero(resultSet.getInt(DatabaseSchema.*COLUMN\_PASAJERO\_ID*),  
 resultSet.getString(DatabaseSchema.*COLUMN\_PASAJERO\_NOMBRE*),  
 resultSet.getInt(DatabaseSchema.*COLUMN\_PASAJERO\_EDAD*),  
 resultSet.getDouble(DatabaseSchema.*COLUMN\_PASAJERO\_PESO*));  
 }  
 return null;  
 }  
  
  
 public void borrarPasajero(int id) throws SQLException {  
 String query = "DELETE FROM " + DatabaseSchema.*TABLE\_PASAJEROS* + " WHERE " + DatabaseSchema.*COLUMN\_PASAJERO\_ID* + " = ?";  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setInt(1, id);  
 prepareStatement.executeUpdate();  
 }  
  
  
 public List<Pasajero> listarPasajeros() throws SQLException {  
 List<Pasajero> pasajeros = new ArrayList<>();  
 String query = "SELECT \* FROM " + DatabaseSchema.*TABLE\_PASAJEROS*;  
  
 prepareStatement = (PreparedStatement) connection.createStatement();  
 resultSet = prepareStatement.executeQuery(query);  
 while (resultSet.next()) {  
 pasajeros.add(new Pasajero(resultSet.getInt(DatabaseSchema.*COLUMN\_PASAJERO\_ID*),  
 resultSet.getString(DatabaseSchema.*COLUMN\_PASAJERO\_NOMBRE*),  
 resultSet.getInt(DatabaseSchema.*COLUMN\_PASAJERO\_EDAD*),  
 resultSet.getDouble(DatabaseSchema.*COLUMN\_PASAJERO\_PESO*)));  
 }  
  
 return pasajeros;  
 }  
  
  
 public void añadirPasajeroEnCoche(int pasajeroId, int cocheId) throws SQLException {  
 String query = "INSERT INTO " + DatabaseSchema.*TABLE\_COCHE\_PASAJERO* + " (" +  
 DatabaseSchema.*COLUMN\_COCHE\_PASAJERO\_COCHE\_ID* + ", " +  
 DatabaseSchema.*COLUMN\_COCHE\_PASAJERO\_PASAJERO\_ID* + ") VALUES (?, ?)";  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setInt(1, cocheId);  
 prepareStatement.setInt(2, pasajeroId);  
 prepareStatement.executeUpdate();  
 }  
  
  
 public void borrarPasajeroEnCoche(int pasajeroId, int cocheId) throws SQLException {  
 String query = "DELETE FROM " + DatabaseSchema.*TABLE\_COCHE\_PASAJERO* +  
 " WHERE " + DatabaseSchema.*COLUMN\_COCHE\_PASAJERO\_COCHE\_ID* + " = ? AND " +  
 DatabaseSchema.*COLUMN\_COCHE\_PASAJERO\_PASAJERO\_ID* + " = ?";  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setInt(1, cocheId);  
 prepareStatement.setInt(2, pasajeroId);  
 prepareStatement.executeUpdate();  
 }  
  
  
 public List<Pasajero> listarPasajerosPorCoche(int cocheId) throws SQLException {  
 List<Pasajero> pasajeros = new ArrayList<>();  
 String query = "SELECT p.\* FROM " + DatabaseSchema.*TABLE\_PASAJEROS* + " p " +  
 "JOIN " + DatabaseSchema.*TABLE\_COCHE\_PASAJERO* + " cp ON p." +  
 DatabaseSchema.*COLUMN\_PASAJERO\_ID* + " = cp." + DatabaseSchema.*COLUMN\_COCHE\_PASAJERO\_PASAJERO\_ID* +  
 " WHERE cp." + DatabaseSchema.*COLUMN\_COCHE\_PASAJERO\_COCHE\_ID* + " = ?";  
  
 prepareStatement = connection.prepareStatement(query);  
 prepareStatement.setInt(1, cocheId);  
 resultSet = prepareStatement.executeQuery();  
 while (resultSet.next()) {  
 pasajeros.add(new Pasajero(resultSet.getInt(DatabaseSchema.*COLUMN\_PASAJERO\_ID*),  
 resultSet.getString(DatabaseSchema.*COLUMN\_PASAJERO\_NOMBRE*),  
 resultSet.getInt(DatabaseSchema.*COLUMN\_PASAJERO\_EDAD*),  
 resultSet.getDouble(DatabaseSchema.*COLUMN\_PASAJERO\_PESO*)));  
 }  
 return pasajeros;  
 }  
  
  
}

clase DatabaseConnetion

package database;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
  
public class DatabaseConnection {  
 public static Connection getConnection() {  
 Connection connection = null;  
 try {  
 connection = DriverManager.*getConnection*(DatabaseSchema.*DB\_URL*, DatabaseSchema.*DB\_USER*, DatabaseSchema.*DB\_PASSWORD*);  
 System.*out*.println("Conexión a la base de datos establecida con éxito.");  
 } catch (SQLException e) {  
 System.*out*.println("Error al conectar a la base de datos: " + e.getMessage());  
 }  
 return connection;  
 }  
}

clase DatabseSchema

package database;  
  
public class DatabaseSchema {  
 public static final String *DB\_URL* = "jdbc:mysql://localhost:3306/gestion\_coches";  
 public static final String *DB\_USER* = "root";  
 public static final String *DB\_PASSWORD* = "";  
 public static final String *TABLE\_COCHES* = "coches";  
 public static final String *COLUMN\_COCHE\_ID* = "id";  
 public static final String *COLUMN\_COCHE\_MARCA* = "marca";  
 public static final String *COLUMN\_COCHE\_MODELO* = "modelo";  
 public static final String *COLUMN\_COCHE\_ANIO* = "anio";  
 public static final String *TABLE\_PASAJEROS* = "pasajeros";  
 public static final String *COLUMN\_PASAJERO\_ID* = "id";  
 public static final String *COLUMN\_PASAJERO\_NOMBRE* = "nombre";  
 public static final String *COLUMN\_PASAJERO\_EDAD* = "edad";  
 public static final String *COLUMN\_PASAJERO\_PESO* = "peso";  
 public static final String *TABLE\_COCHE\_PASAJERO* = "coche\_pasajero";  
 public static final String *COLUMN\_COCHE\_PASAJERO\_COCHE\_ID* = "coche\_id";  
 public static final String *COLUMN\_COCHE\_PASAJERO\_PASAJERO\_ID* = "pasajero\_id";  
}

clase Coche

package model;  
  
import lombok.\*;  
  
@Setter  
@Getter  
@NoArgsConstructor  
@AllArgsConstructor  
@ToString  
public class Coche {  
 private int id;  
 private String marca;  
 private String modelo;  
 private int anio;  
  
public void mostrarCoche() {  
 System.*out*.println("id = " + id);  
 System.*out*.println("marca = " + marca);  
 System.*out*.println("modelo = " + modelo);  
 System.*out*.println("año = " + anio);  
  
}  
  
}

clase Pasajero

package model;  
  
import lombok.\*;  
  
@Setter  
@Getter  
@AllArgsConstructor  
@NoArgsConstructor  
@ToString  
public class Pasajero {  
 private int id;  
 private String nombre;  
 private int edad;  
 private double peso;  
  
public void mostrarPasajero(){  
  
 System.*out*.println("id = " + id);  
 System.*out*.println("nombre = " + nombre);  
 System.*out*.println("edad = " + edad);  
 System.*out*.println("peso = " + peso);  
  
}  
}

clase Main

import controller.CocheFun;  
import controller.PasajeroFun;  
  
import java.util.Scanner;  
  
public class Main {  
 private static final CocheFun *cocheFun* = new CocheFun();  
 private static final PasajeroFun *pasajeroFun* = new PasajeroFun();  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 int opcion;  
  
 do {  
 System.*out*.println("\n===== MENÚ PRINCIPAL =====");  
 System.*out*.println("1. Añadir nuevo coche");  
 System.*out*.println("2. Borrar coche por ID");  
 System.*out*.println("3. Consulta coche por ID");  
 System.*out*.println("4. Modificar coche por ID");  
 System.*out*.println("5. Listado de coches");  
 System.*out*.println("6. Gestión de pasajeros");  
 System.*out*.println("7. Terminar el programa");  
 System.*out*.print("Elija una opción: ");  
 opcion = scanner.nextInt();  
  
 switch (opcion) {  
 case 1 -> *cocheFun*.nuevoCoche(scanner);  
 case 2 -> *cocheFun*.borrarCoche(scanner);  
 case 3 -> *cocheFun*.buscarCoche(scanner);  
 case 4 -> *cocheFun*.modificarCoche(scanner);  
 case 5 -> *cocheFun*.listarCoches();  
 case 6 -> *menuGestionPasajeros*(scanner);  
 case 7 -> System.*out*.println("Saliendo del programa...");  
 default -> System.*out*.println("Opción no válida. Inténtelo de nuevo.");  
 }  
 } while (opcion != 7);  
 }  
  
 private static void menuGestionPasajeros(Scanner scanner) {  
 int opcion;  
  
 do {  
 System.*out*.println("\n===== MENÚ GESTIÓN DE PASAJEROS =====");  
 System.*out*.println("1. Añadir nuevo pasajero");  
 System.*out*.println("2. Borrar pasajero por ID");  
 System.*out*.println("3. Consultar pasajero por ID");  
 System.*out*.println("4. Listar todos los pasajeros");  
 System.*out*.println("5. Añadir pasajero a coche");  
 System.*out*.println("6. Eliminar pasajero de coche");  
 System.*out*.println("7. Listar pasajeros de un coche");  
 System.*out*.println("8. Volver al menú principal");  
 System.*out*.print("Elija una opción: ");  
 opcion = scanner.nextInt();  
  
 switch (opcion) {  
 case 1 -> *pasajeroFun*.nuevoPasajero(scanner);  
 case 2 -> *pasajeroFun*.borrarPasajero(scanner);  
 case 3 -> *pasajeroFun*.consultarPasajero(scanner);  
 case 4 -> *pasajeroFun*.listarPasajeros();  
 case 5 -> *pasajeroFun*.añadirPasajeroACoche(scanner);  
 case 6 -> *pasajeroFun*.eliminarPasajeroDeCoche(scanner);  
 case 7 -> *pasajeroFun*.listarPasajerosDeCoche(scanner);  
 case 8 -> System.*out*.println("Volviendo al menú principal...");  
 default -> System.*out*.println("Opción no válida. Inténtelo de nuevo.");  
 }  
 } while (opcion != 8);  
 }  
}

**Creación de Base de Datos**

**Texto

Descripción generada automáticamente**

**Ejecución y resultado.**

Ejecución clase main

Imagen que contiene Forma

Descripción generada automáticamente

Ejecución añadir coche

Texto

Descripción generada automáticamente

Ejecución listar coche y comprobación de añadir coche

Texto

Descripción generada automáticamente

Ejecución eliminar coche y comprobación

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ejecución bucar coche por id

Texto

Descripción generada automáticamente

Ejecución modificar coche y comprobación

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

Estado de base datos con todos estos cambios

Pantalla de un video juego

Descripción generada automáticamente con confianza media

Ejecución menú pasajeros

Texto

Descripción generada automáticamente

Ejecución añadir pasajero y comprobación

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ejecución borrar pasajero y comprobación

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Pantalla de computadora con letras

Descripción generada automáticamente con confianza baja

Ejecución busca pasajero por id

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Ejecución listar pasajero

Pantalla de computadora con letras

Descripción generada automáticamente con confianza baja

Estado Base de datos con cambios

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ejecución añadir pasajero a coche y comprobación

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Ejecución eliminar pasajero de coche y comprobación

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Ejecución listado pasajeros de un coche

Texto

Descripción generada automáticamente

Ejecución volver menú principal

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Ejecución termina programa

**Texto

Descripción generada automáticamente**

**Enlace a GitHub:**