

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 (?, ?,
?)" ;

        preparedStatement = connection.prepareStatement(query);
        preparedStatement.setString(1, coche.getMarca());
        preparedStatement.setString(2, coche.getModelo());
        preparedStatement.setInt(3, coche.getAnio());
        preparedStatement.executeUpdate();

    }

    public Coche buscaCochePorId(int id) throws SQLException {
        String query = "SELECT * FROM " + DatabaseSchema.TABLE_COCHES
+
            " WHERE " + DatabaseSchema.COLUMN_COCHE_ID + " = ?";

        preparedStatement = connection.prepareStatement(query);
        preparedStatement.setInt(1, id);
        resultSet = preparedStatement.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 + " = ?";

        preparedStatement = connection.prepareStatement(query);
        preparedStatement.setString(1, marca);
        preparedStatement.setString(2, modelo);
        preparedStatement.setInt(3, anio);
        preparedStatement.setInt(4, id);
        preparedStatement.executeUpdate();

    }

    public void borrarCoche(int id) throws SQLException {
        String query = "DELETE FROM " + DatabaseSchema.TABLE_COCHES +
            " WHERE " + DatabaseSchema.COLUMN_COCHE_ID + " = ?";

```

```

        preparedStatement = connection.prepareStatement(query);
        preparedStatement.setInt(1, id);
        preparedStatement.executeUpdate();
    }

    public ArrayList<Coche> listarCoches() throws SQLException {
        ArrayList<Coche> coches = new ArrayList<>();
        String query = "SELECT * FROM " + DatabaseSchema.TABLE_COCHES;

        preparedStatement = connection.prepareStatement(query);
        resultSet = preparedStatement.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 preparedStatement;
    // 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 (?, ?,
?)" ;

        preparedStatement = connection.prepareStatement(query);
        preparedStatement.setString(1, pasajero.getNombre());
        preparedStatement.setInt(2, pasajero.getEdad());
        preparedStatement.setDouble(3, pasajero.getPeso());
        preparedStatement.executeUpdate();

    }

    public Pasajero getPasajeroById(int id) throws SQLException {
        String query = "SELECT * FROM " +
DatabaseSchema.TABLE_PASAJEROS +
        " WHERE " + DatabaseSchema.COLUMN_PASAJERO_ID + " =
?";

        preparedStatement = connection.prepareStatement(query);
        preparedStatement.setInt(1, id);
        resultSet = preparedStatement.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 + " = ?";

        preparedStatement = connection.prepareStatement(query);
        preparedStatement.setInt(1, id);
        preparedStatement.executeUpdate();
    }

    public List<Pasajero> listarPasajeros() throws SQLException {
        List<Pasajero> pasajeros = new ArrayList<>();
        String query = "SELECT * FROM " +
DatabaseSchema.TABLE_PASAJEROS;

        preparedStatement = (PreparedStatement)
connection.createStatement();
        resultSet = preparedStatement.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 (?, ?)";

    preparedStatement = connection.prepareStatement(query);
    preparedStatement.setInt(1, cocheId);
    preparedStatement.setInt(2, pasajeroId);
    preparedStatement.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 + " =
?";

    preparedStatement = connection.prepareStatement(query);
    preparedStatement.setInt(1, cocheId);
    preparedStatement.setInt(2, pasajeroId);
    preparedStatement.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 + " = ?";

    preparedStatement = connection.prepareStatement(query);
    preparedStatement.setInt(1, cocheId);
    resultSet = preparedStatement.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;

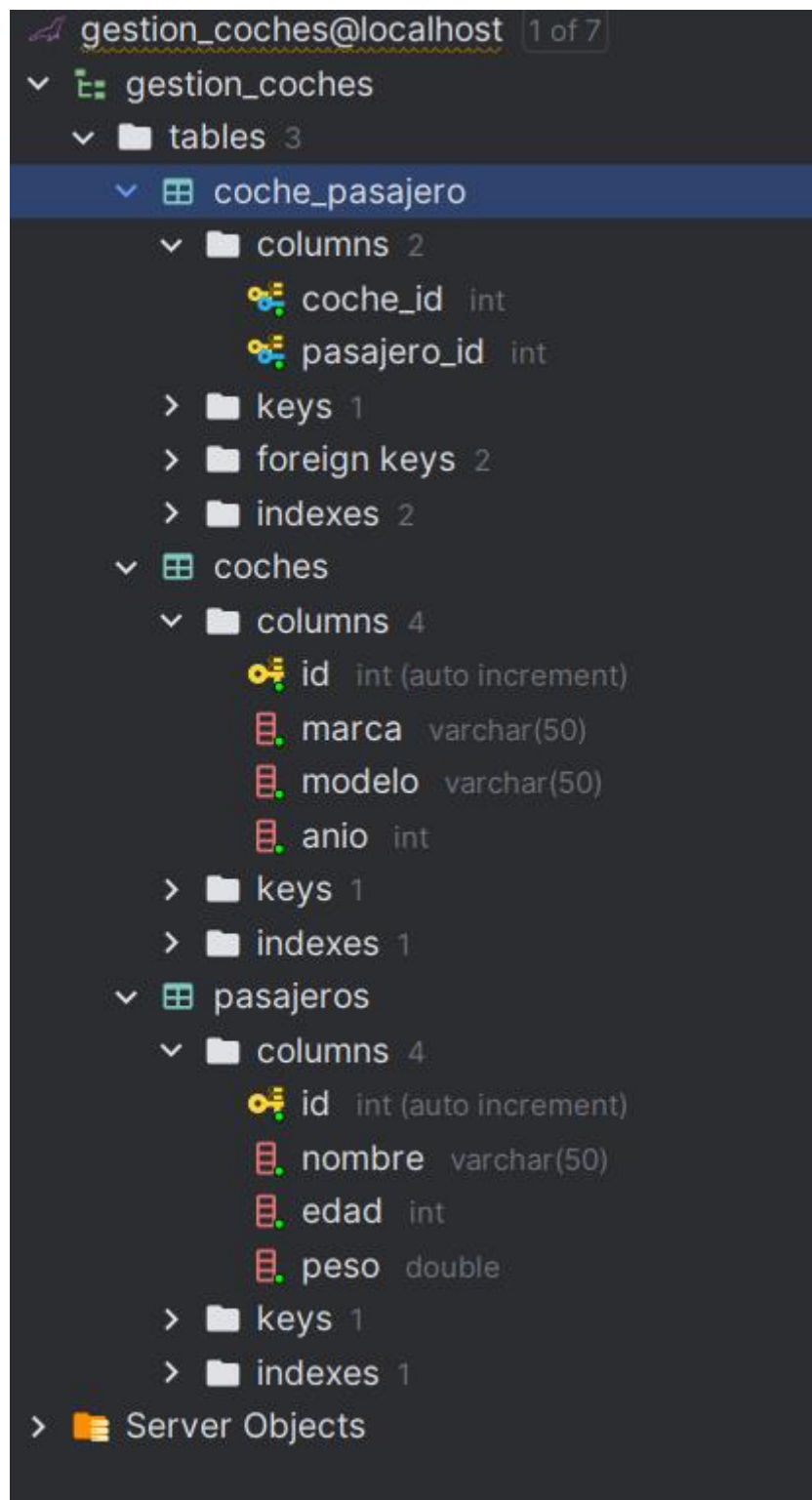
        do {
            System.out.println("\n===== MENÚ GESTIÓN DE PASAJEROS
=====");
            System.out.println("1. Añadir nuevo pasajero");
            System.out.println("2. Borrar pasajero por ID");
            System.out.println("3. Consultar pasajero por ID");
            System.out.println("4. Listar 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



Ejecución y resultado.

Ejecución clase main

```
===== MENÚ PRINCIPAL =====  
1. Añadir nuevo coche  
2. Borrar coche por ID  
3. Consulta coche por ID  
4. Modificar coche por ID  
5. Listado de coches  
6. Gestión de pasajeros  
7. Terminar el programa  
Elija una opción:  
_____
```

Ejecución añadir coche

```
7. Terminar el programa  
Elija una opción: 1  
Marca del coche: golf  
Modelo del coche: golf  
Año del coche: 2010  
Coche añadido correctamente.
```

Ejecución listar coche y comprobación de añadir coche

```
Elija una opción: 5  
Coche(id=4, marca=audi, modelo=a3, anio=2009)  
Coche(id=6, marca=renault, modelo=5, anio=1800)  
Coche(id=7, marca=golf, modelo=golf, anio=2010)
```

Ejecución eliminar coche y comprobación

```
Elija una opción: 2  
ID del coche a borrar: 7  
Coche borrado correctamente.
```

```
Elija una opción: 5  
Coche(id=4, marca=audi, modelo=a3, anio=2009)  
Coche(id=6, marca=renault, modelo=5, anio=1800)
```

Ejecución bucar coche por id

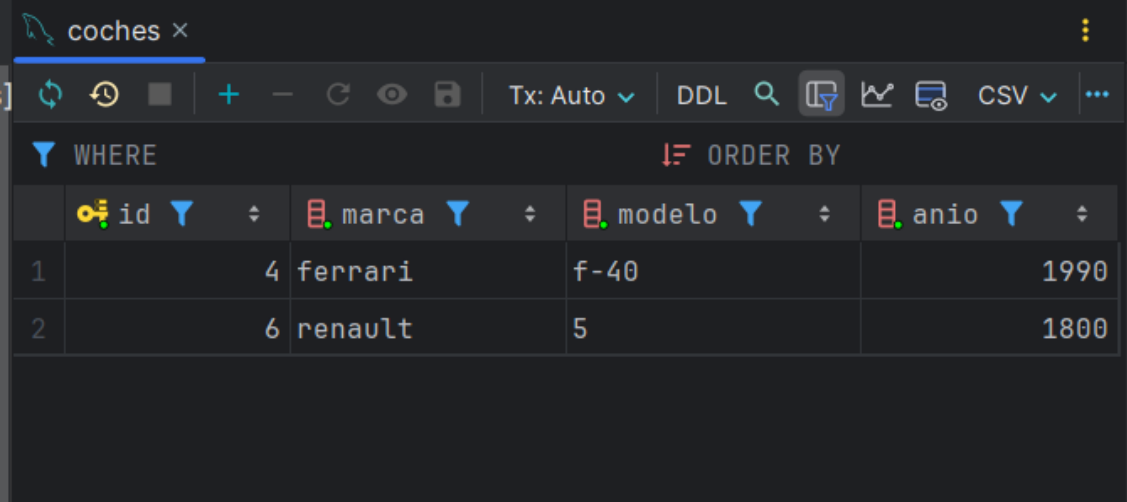

```
Elija una opción: 3
ID del coche a consultar: 4
Coche encontrado: Coche(id=4, marca=audi, modelo=a3, anio=2009)
```

Ejecución modificar coche y comprobación

```
Elija una opción: 4
ID del coche a modificar: 4
Nueva marca: ferrari
Nuevo modelo: f-40
Nuevo año: 1990
Coche actualizado correctamente.
```

```
Elija una opción: 5
Coche(id=4, marca=ferrari, modelo=f-40, anio=1990)
Coche(id=6, marca=renault, modelo=5, anio=1800)
```

Estado de base datos con todos estos cambios



	id	marca	modelo	anio
1	4	ferrari	f-40	1990
2	6	renault	5	1800

Ejecución menú pasajeros

```
===== MENÚ GESTIÓN DE PASAJEROS =====  
1. Añadir nuevo pasajero  
2. Borrar pasajero por ID  
3. Consultar pasajero por ID  
4. Listar todos los pasajeros  
5. Añadir pasajero a coche  
6. Eliminar pasajero de coche  
7. Listar pasajeros de un coche  
8. Volver al menú principal  
Elija una opción:
```

Ejecución añadir pasajero y comprobación

```
Elija una opción: 1  
Nombre del pasajero: juan  
Edad del pasajero: 35  
Peso del pasajero: 70  
Pasajero añadido correctamente.
```

```
Elija una opción: 4  
Pasajero(id=5, nombre=ana, edad=1, peso=15.0)  
Pasajero(id=7, nombre=ivan, edad=41, peso=95.0)  
Pasajero(id=8, nombre=juan, edad=35, peso=70.0)
```

Ejecución borrar pasajero y comprobación

```
Elija una opción: 2  
ID del pasajero a borrar: 8  
Pasajero borrado correctamente.
```

```
Elija una opción: 4  
Pasajero(id=5, nombre=ana, edad=1, peso=15.0)  
Pasajero(id=7, nombre=ivan, edad=41, peso=95.0)
```

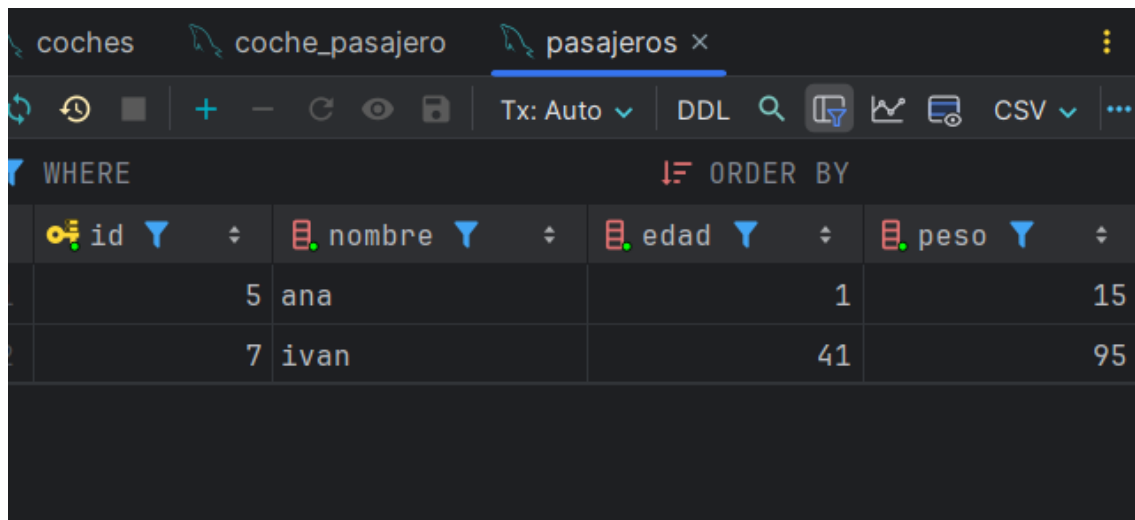
Ejecución busca pasajero por id

```
Elija una opción: 3
ID del pasajero a consultar: 8
Pasajero encontrado: Pasajero(id=8, nombre=juan, edad=35, peso=70.0)
```

Ejecución listar pasajero

```
Elija una opción: 4
Pasajero(id=5, nombre=ana, edad=1, peso=15.0)
Pasajero(id=7, nombre=ivan, edad=41, peso=95.0)
```

Estado Base de datos con cambios



id	nombre	edad	peso
5	ana	1	15
7	ivan	41	95

Ejecución añadir pasajero a coche y comprobación

```
Elija una opción: 5
ID del pasajero: 7
ID del coche: 4
Pasajero añadido al coche correctamente.
```

```
Elija una opción: 7
ID del coche: 4
Pasajero(id=7, nombre=ivan, edad=41, peso=95.0)
```

Ejecución eliminar pasajero de coche y comprobación

```
Elija una opción: 6
ID del pasajero: 7
ID del coche: 4
Pasajero eliminado del coche correctamente.
```

```
Elija una opción: 7
ID del coche: 4
El coche con ID 4 no tiene pasajeros asociados.
```

Ejecución listado pasajeros de un coche

```
Elija una opción: 7
ID del coche: 4
Pasajero(id=7, nombre=ivan, edad=41, peso=95.0)
----- MENÚ GESTIÓN DE PASAJEROS -----
```

Ejecución volver menú principal

```
Elija una opción: 8
Volviendo al menú principal...
```

Ejecución termina programa

```
Elija una opción: 7
Saliendo del programa...
```

Enlace a GitHub: