

# ACCESO A DATOS

Jesús Martín Robles

DAM a distancia

Curso 23-24

I.E.S. Aguadulce

Tarea 6 :  
Programación  
de  
componentes  
de acceso a  
datos.

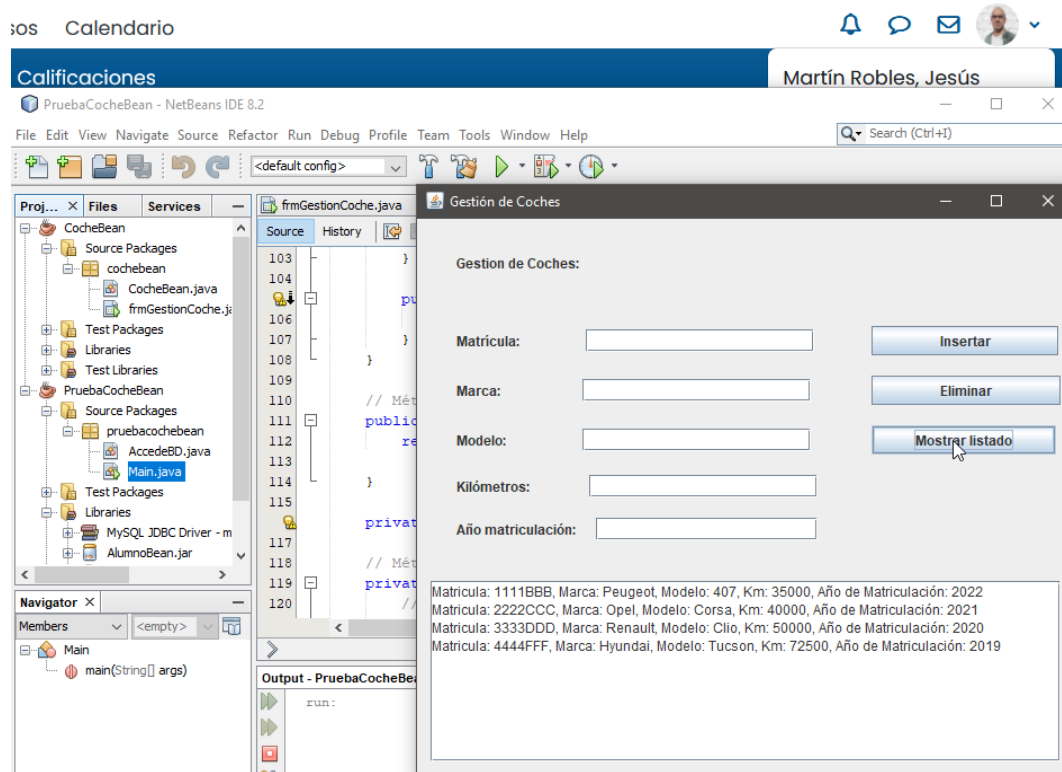
# Índice

Base de Datos .....	2
Componente en proyecto de prueba .....	3
Tipos de acceso a datos .....	6
Dificultades .....	7

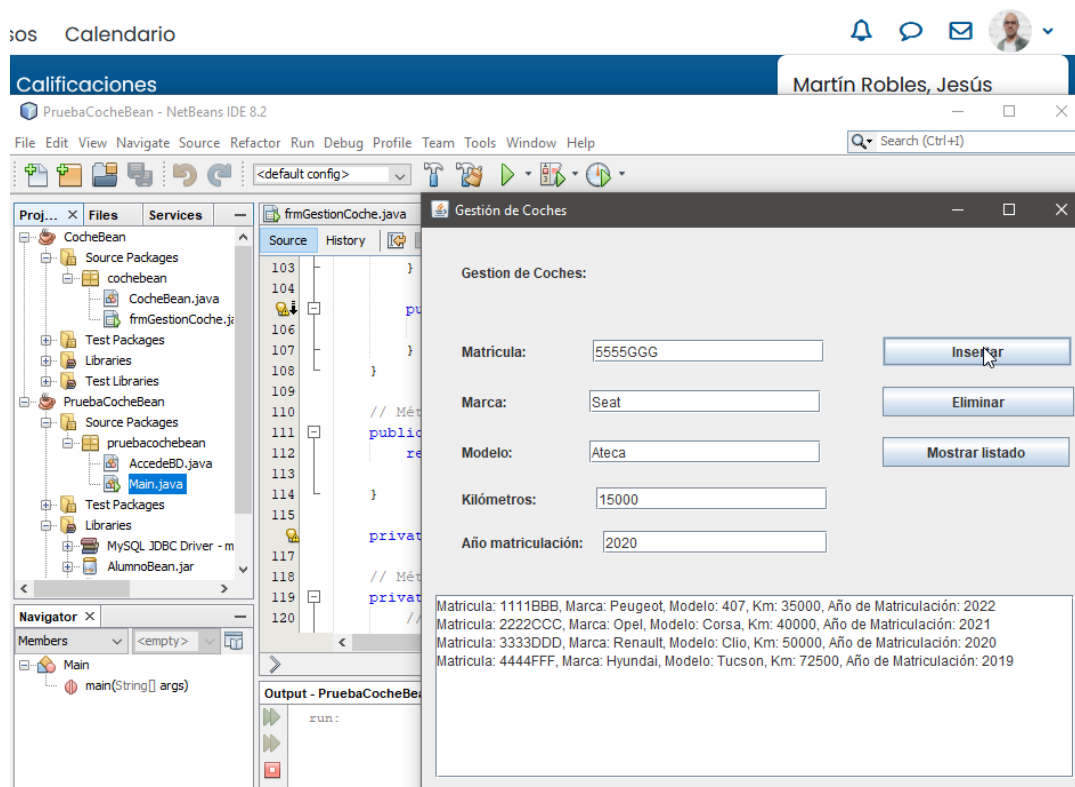
## Script de la base de datos propuesta en el ejercicio 1 de la tarea:

```
1 • CREATE DATABASE IF NOT EXISTS coches;
2
3 • USE coches;
4
5 • CREATE TABLE IF NOT EXISTS coches (
6     matricula VARCHAR(7),
7     marca VARCHAR(30),
8     modelo VARCHAR(30),
9     km BIGINT,
10    yearmat INT
11 );
12
13 • INSERT INTO coches (matricula, marca, modelo, km, yearmat)
14 VALUES
15     ('1111BBB', 'Peugeot', '407', 35000, 2022),
16     ('2222CCC', 'Opel', 'Corsa', 40000, 2021),
17     ('3333DDD', 'Renault', 'Clio', 50000, 2020),
18     ('4444FFF', 'Hyundai', 'Tucson', 72500, 2019);
19
```

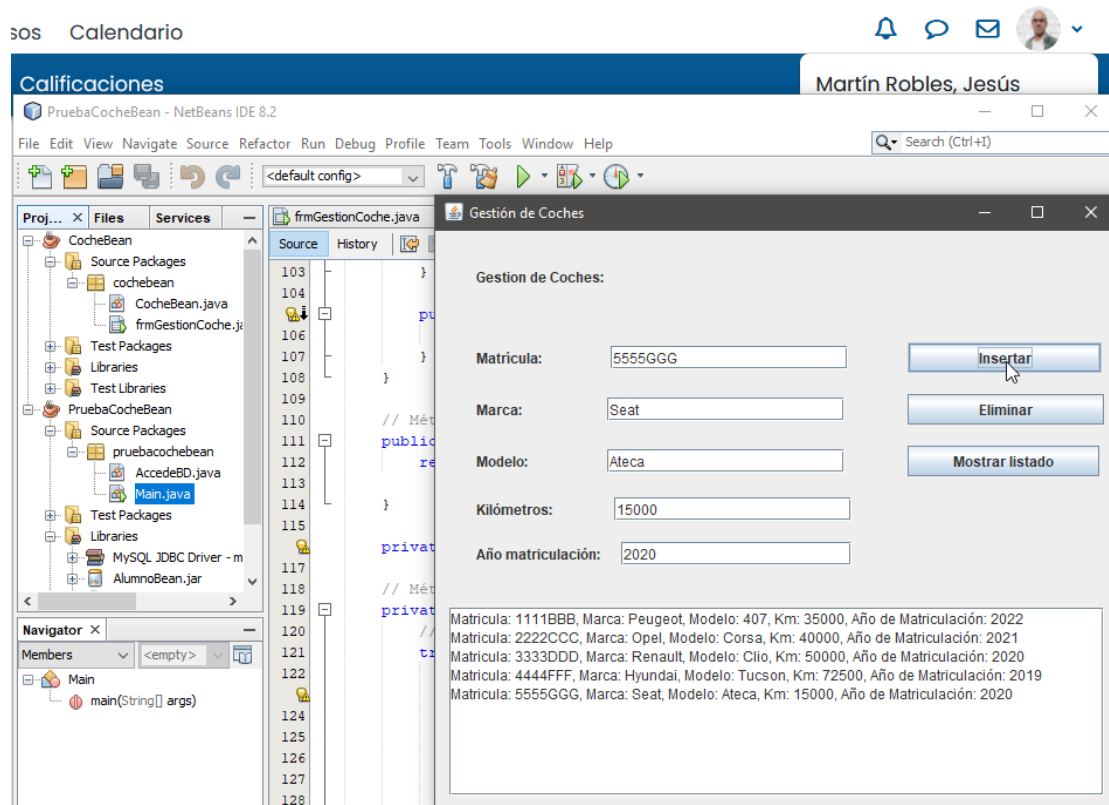
## Pruebas del componente en un proyecto de prueba:



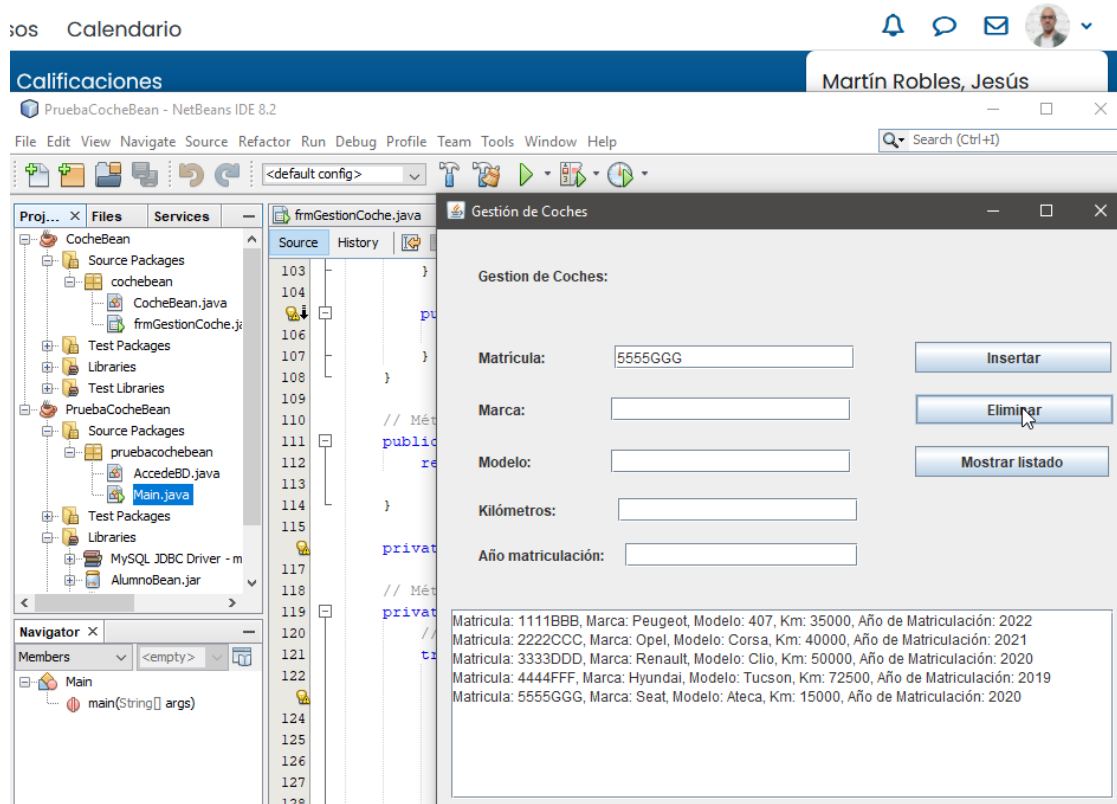
- Se muestran los datos de la base de datos de los coches.



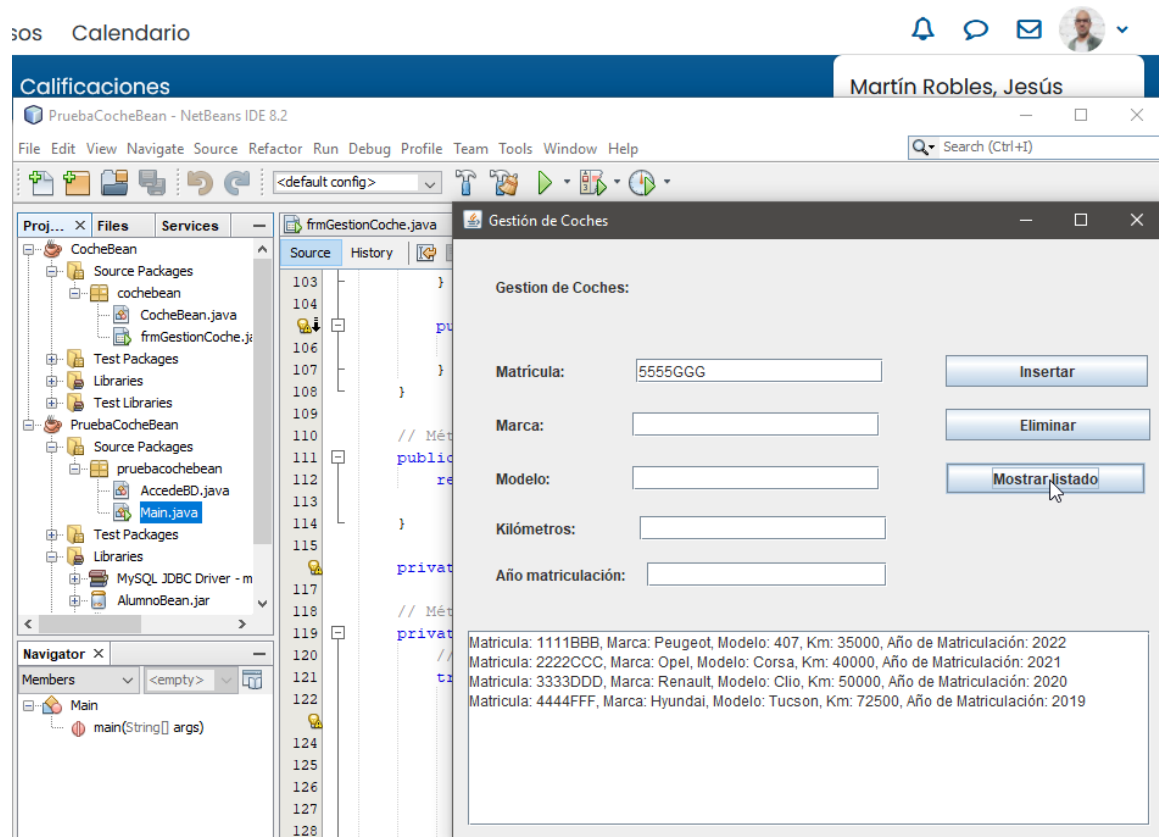
- Se insertan datos nuevos en los campos correspondientes.



- Se inserta correctamente el nuevo coche.



- Eliminamos coche por la matrícula.



- Coche eliminado correctamente.

## **Tipos de acceso a datos:**

### **1. Uso de ficheros:**

- Los componentes JavaBeans pueden ser útiles para el acceso a datos mediante ficheros cuando se necesita leer o escribir datos en ficheros de texto o binarios de manera estructurada y modular.

- Este componente podría ser útil en proyectos de gestión de archivos de configuración, logs, o cualquier otro tipo de datos que se almacenen en ficheros.

### **2. Gestión de información usando mapeo objeto-relacional (ORM):**

- El uso de JavaBeans en este contexto facilita la integración entre el modelo de datos relacional y el modelo de objetos de la aplicación.

- Sería útil en proyectos que requieran acceso a una base de datos relacional donde se quiera mapear las entidades de la base de datos a objetos Java de manera automática y transparente, utilizando un framework ORM como Hibernate o JPA.

### **3. Gestión de información usando bases de datos orientadas a objetos:**

- En este caso, los componentes JavaBeans pueden ser útiles para representar las entidades de la base de datos orientada a objetos como objetos Java.

- Se aplicaría en proyectos que utilizan bases de datos orientadas a objetos como MongoDB o db4o, donde los datos se almacenan directamente como objetos Java.

### **4. Gestión de información usando bases de datos XML:**

- Aquí, los JavaBeans pueden ser útiles para representar la estructura de los datos XML como objetos Java.

- Se podría utilizar en proyectos que necesiten leer, escribir o manipular datos en archivos XML de manera programática, como aplicaciones de procesamiento de documentos XML o integración de sistemas que intercambian datos en formato XML.

## Algunas dificultades encontradas:

- Una dificultad fue la obtención de la lista de coches, que la declaré como privada por lo que tuve que crear un método que únicamente devolviera esa lista:  

```
public List<Coche> getCochesList()  
{  
    return Coches;  
}
```
- Por otro lado, tuve problemas con la dimensión de la ventana al cargarlo desde el proyecto test, que se debía a haber arrastrado esta línea de una práctica anterior y no redimensionaba correctamente la ventana ni sus campos.

```
frame.setLayout(new FlowLayout());
```