INSTITUTO DE EDUCACIÓN SUPERIOR CIBERTEC
DIRECCIÓN ACADÉMICA
CARRERA PROFESIONALES

CIBERTEC

CURSO	:	**Desarrollo de Aplicaciones Web I**
PROFESOR	:	**César Enrique Santos Torres**
CICLO	:	**Quinto**
SECCIÓN	:	
GRUPO	:	
FECHA	:	
DURACIÓN	:	**50 minutos**

NOTA

ALUMNO (A) : EINER HUGO CHAVEZ AGUILAR

CASO DE LABORATORIO 3 (EF)

Consideraciones generales:

- El laboratorio consta de 1 Crud implementado con Spring RESTFul + Spring Data JPA, cada operación del CRUD deberá ir acompañada (De forma obligatoria) de capturas de pantalla de lo implementado.

- Sólo debe subir este documento, con sus evidencias y respuestas en él. El código fuente del proyecto debe ser subido a Github (Adjuntar link del repositorio). No se aceptará código zipeado.

- El nombre del presente archivo deberá tener la siguiente estructura: “DAWI-APELLIDOPATERNO-APELLIDOMATERNO-NOMBRES.pdf”.

LOGRO DE LA EVALUACION:
Al término de la evaluación, el alumno deberá implementar un CRUD con Spring RESTful, dicho CRUD deberá incluir las siguientes operaciones:
- **/all** (Consulta de todos los items)
- **/detail** (Consulta de un item)
- **/update** (Actualización de un item)
- **/delete/{id}** (Eliminación de un item)
- **/create** (Creación de un item)

CONSOLIDADO

Pregunta	Puntaje		Llenar solo en caso de Recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	5			
2	5			
3	5			
4	5			
Total	20			

Nota Recalificada	

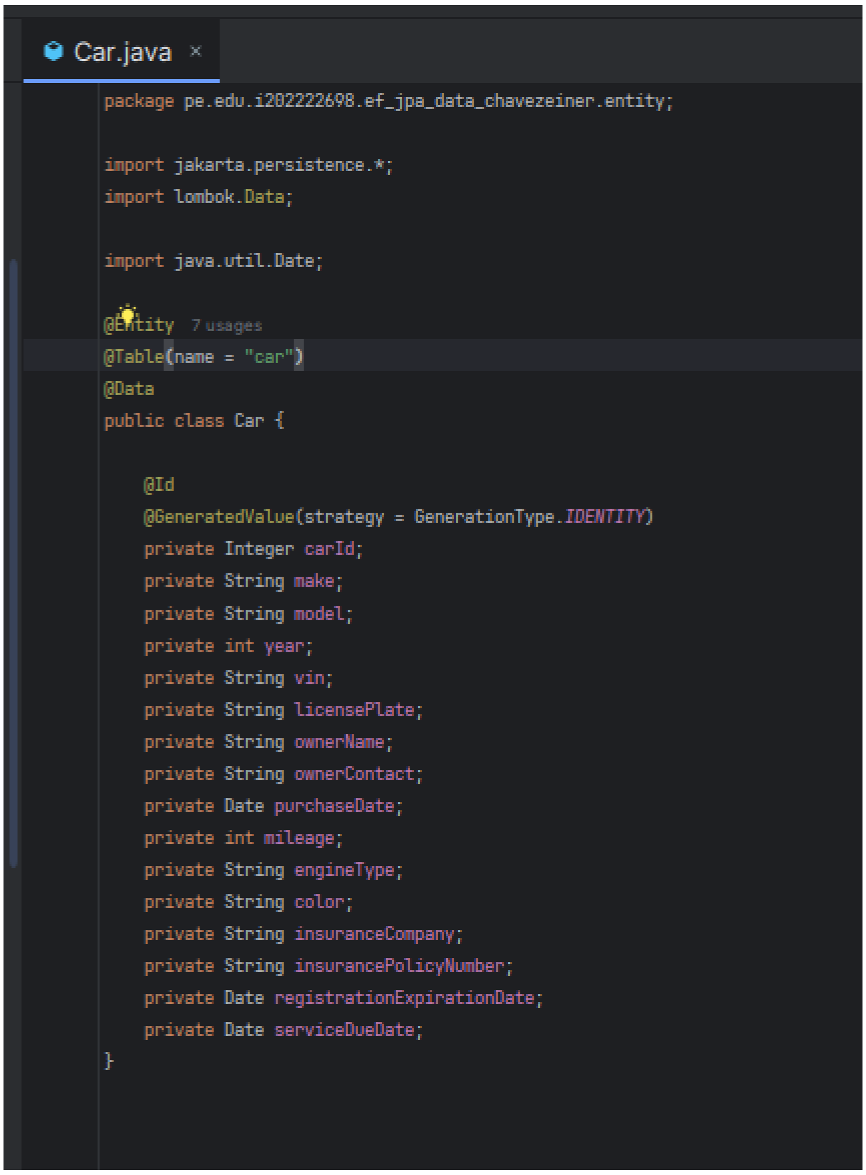
Alcance de la prueba

Implementar un CRUD de la siguiente tabla (Deberá crear un base de datos “**fabric**” y en ella la tabla especificada):

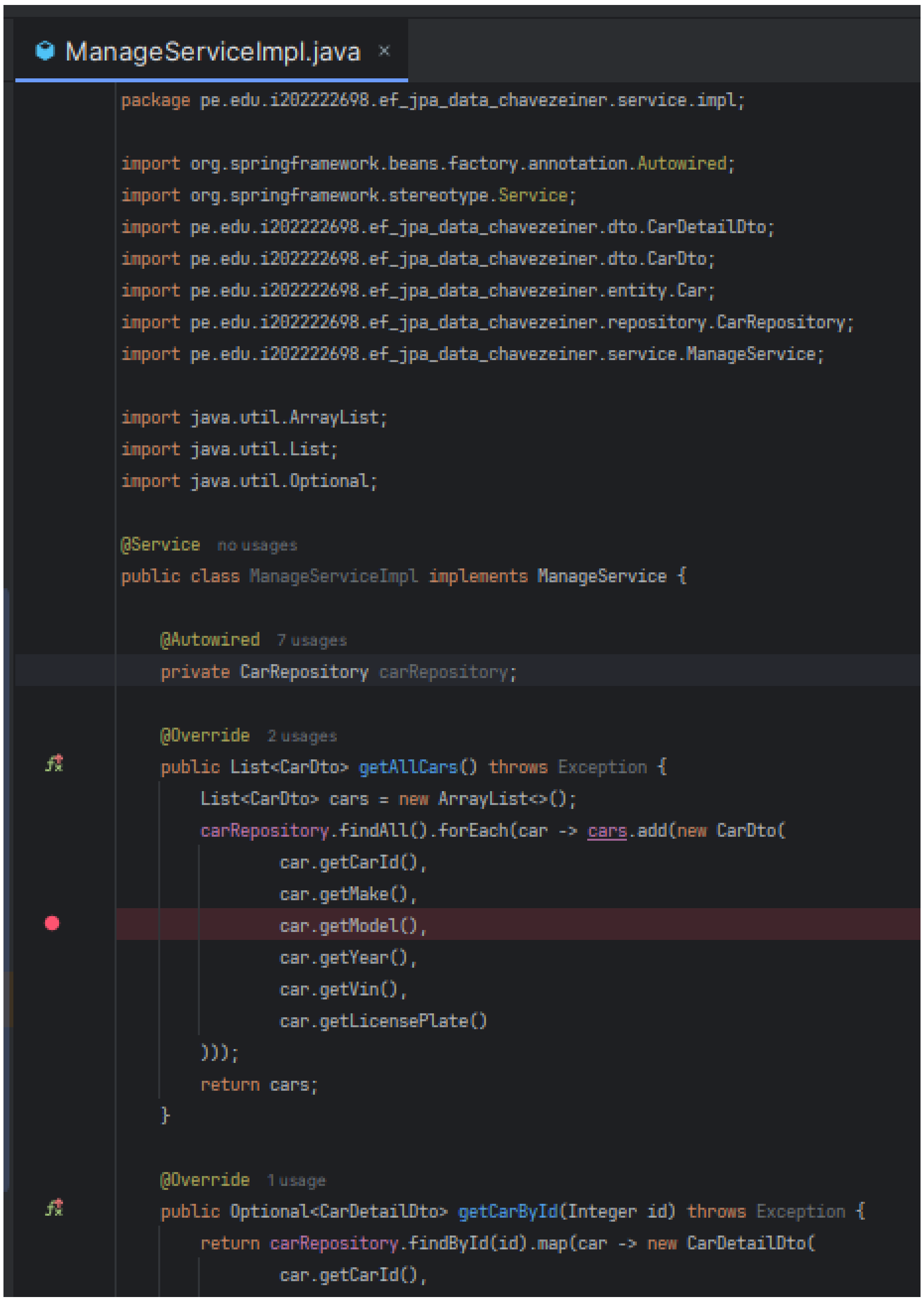
```sql
CREATE TABLE car (
    car_id INT AUTO_INCREMENT PRIMARY KEY,
    make VARCHAR(50),
    model VARCHAR(50),
    year INT,
    vin VARCHAR(50),
    license_plate VARCHAR(20),
    owner_name VARCHAR(100),
    owner_contact VARCHAR(50),
    purchase_date DATE,
    mileage INT,
    engine_type VARCHAR(50),
    color VARCHAR(30),
    insurance_company VARCHAR(100),
    insurance_policy_number VARCHAR(50),
    registration_expiration_date DATE,
    service_due_date DATE
);
```

Ejecutar los siguientes registros de la tabla previa:

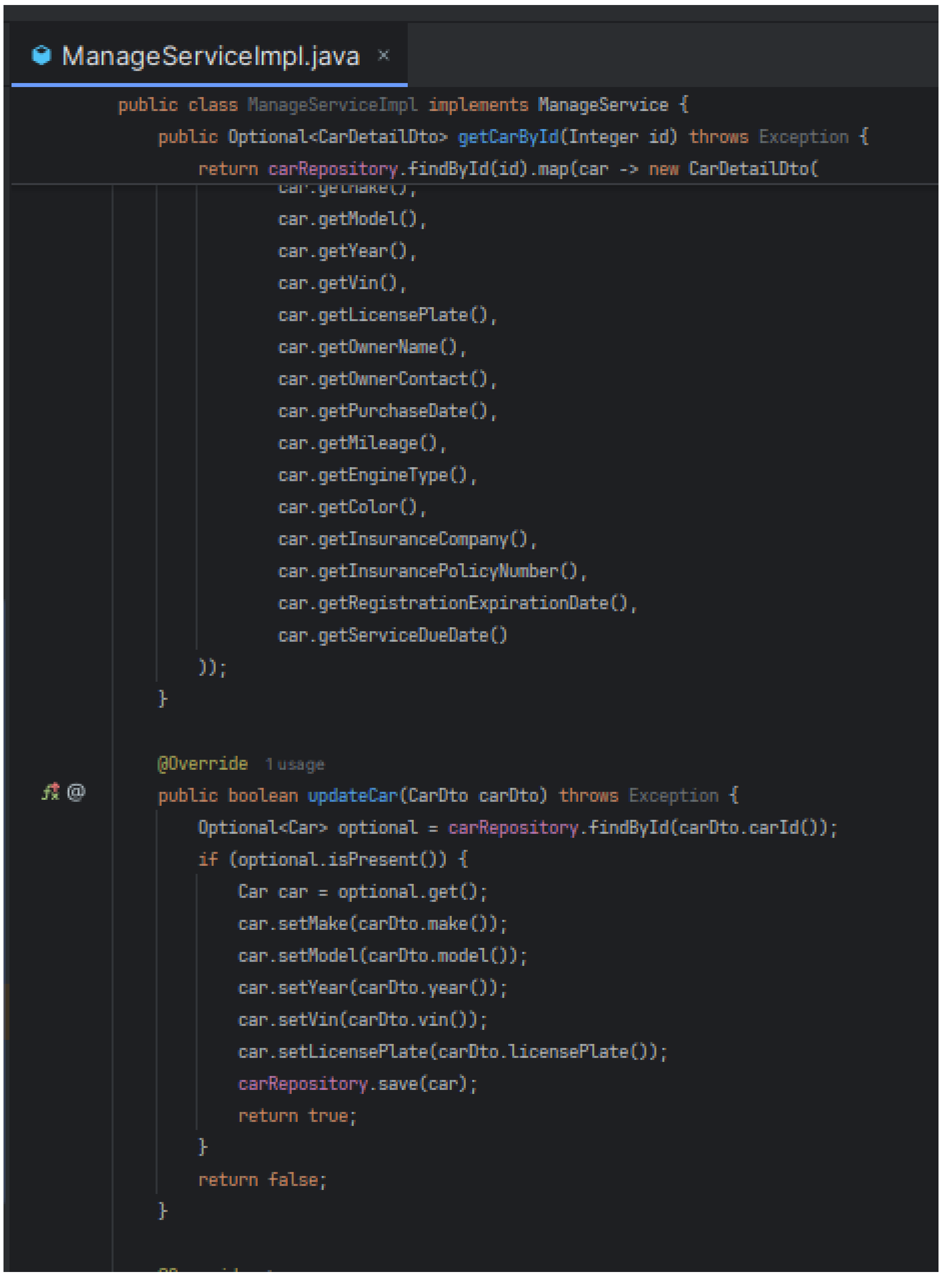
```sql
INSERT INTO car (make, model, year, vin, license_plate, owner_name,
owner_contact, purchase_date, mileage, engine_type, color, insurance_company,
insurance_policy_number, registration_expiration_date, service_due_date) VALUES
('Toyota', 'Corolla', 2018, '1NXBR12E3YZ123456', 'ABC123', 'Juan Perez', '555-
1234', '2018-01-15', 25000, 'Gasoline', 'Red', 'Seguros del Sol', 'INS123456',
'2025-01-15', '2024-07-15'),
('Honda', 'Civic', 2020, '2HGES26785H654321', 'XYZ789', 'Maria Lopez', '555-
5678', '2020-05-10', 15000, 'Gasoline', 'Blue', 'ProtectAuto', 'INS789012',
'2026-05-10', '2025-11-10'),
('Ford', 'Focus', 2019, '1FAFP34N06W765432', 'DEF456', 'Carlos Jimenez', '555-
8765', '2019-03-20', 30000, 'Diesel', 'Black', 'AutoSeguro', 'INS345678', '2024-
03-20', '2023-09-20'),
('Chevrolet', 'Malibu', 2017, '1G1ZD5ST1HF123456', 'GHI123', 'Ana Martinez',
'555-4321', '2017-08-05', 45000, 'Gasoline', 'White', 'AutoProtegido',
'INS901234', '2023-08-05', '2023-02-05'),
('Nissan', 'Altima', 2021, '1N4AL11D75C123456', 'JKL456', 'Luis Rodriguez',
'555-3456', '2021-06-15', 10000, 'Gasoline', 'Silver', 'SegurosTotal',
'INS567890', '2027-06-15', '2026-12-15'),
('Mazda', '3', 2022, 'JM1BPACL1M1234567', 'MNO789', 'Sofia Gonzalez', '555-
6789', '2022-09-20', 5000, 'Gasoline', 'Gray', 'ProtecCar', 'INS234567', '2028-
09-20', '2028-03-20'),
('Hyundai', 'Elantra', 2016, 'KMHDH4AE6DU123456', 'PQR123', 'Pedro Alvarez',
'555-9876', '2016-11-05', 60000, 'Gasoline', 'Green', 'AseguraTodo',
'INS876543', '2022-11-05', '2022-05-05'),
('Kia', 'Optima', 2015, '5XXGT4L34FG123456', 'STU456', 'Isabel Ramirez', '555-
2345', '2015-02-10', 75000, 'Gasoline', 'Yellow', 'CarSeguros', 'INS345678',
'2021-02-10', '2020-08-10'),
('Volkswagen', 'Jetta', 2014, '3VW2K7AJ6EM123456', 'VWX789', 'Manuel Diaz',
'555-8765', '2014-05-15', 90000, 'Gasoline', 'Blue', 'SeguroMovil', 'INS901234',
'2020-05-15', '2019-11-15'),
('Subaru', 'Impreza', 2013, 'JF1GJAA67DG123456', 'YZA123', 'Patricia Sanchez',
'555-6543', '2013-12-01', 100000, 'Gasoline', 'Red', 'AsegurAuto', 'INS567890',
'2019-12-01', '2019-06-01');
```


Car.java

```java
package pe.edu.i202222698.ef_jpa_data_chavezeiner.entity;

import jakarta.persistence.*;
import lombok.Data;

import java.util.Date;

@Entity  7 usages
@Table(name = "car")
@Data
public class Car {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer carId;
    private String make;
    private String model;
    private int year;
    private String vin;
    private String licensePlate;
    private String ownerName;
    private String ownerContact;
    private Date purchaseDate;
    private int mileage;
    private String engineType;
    private String color;
    private String insuranceCompany;
    private String insurancePolicyNumber;
    private Date registrationExpirationDate;
    private Date serviceDueDate;
}
```

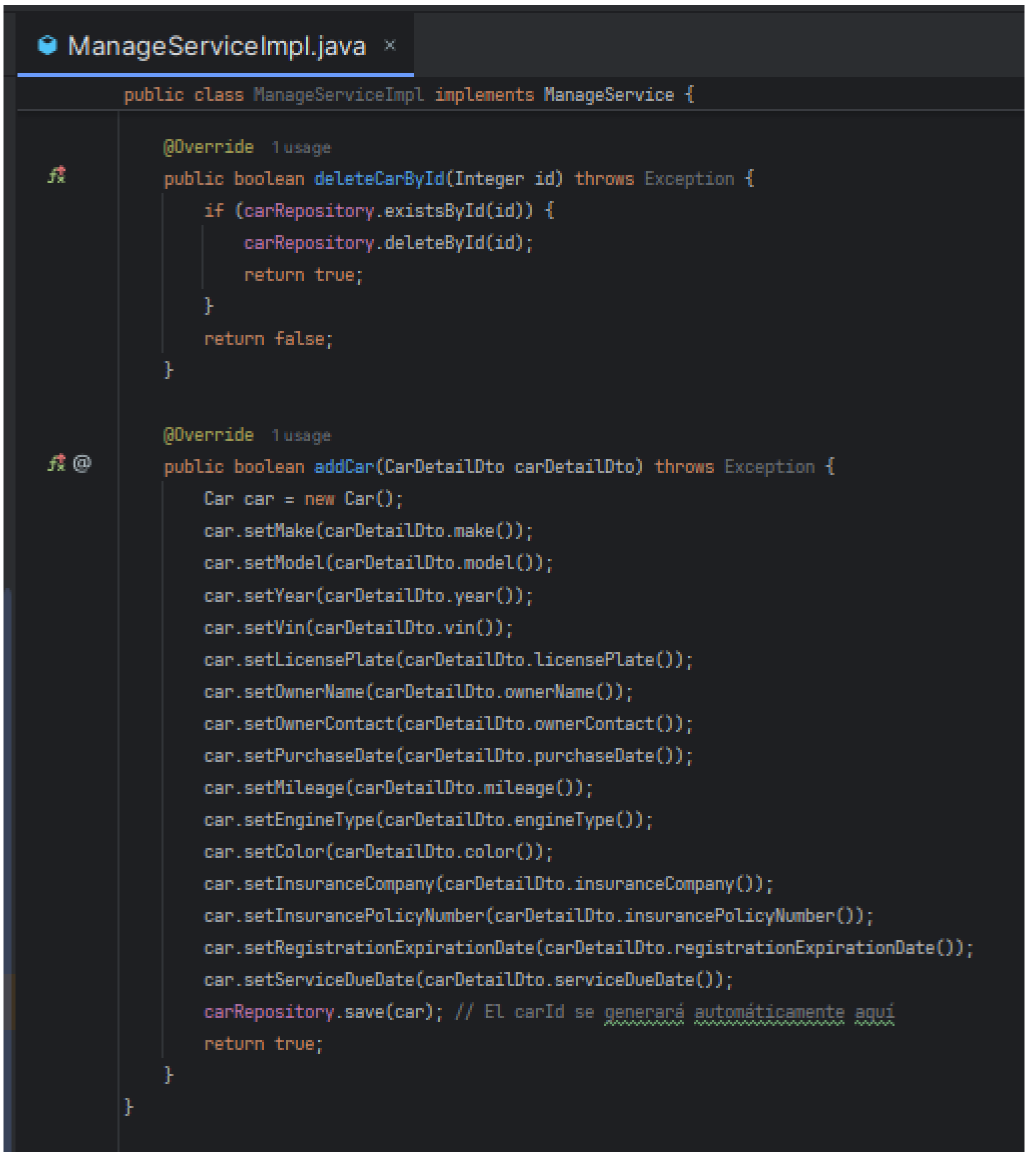

ManageServiceImpl.java

```
package pe.edu.i202222698.ef_jpa_data_chavezeiner.service.impl;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.dto.CarDetailDto;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.dto.CarDto;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.entity.Car;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.repository.CarRepository;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.service.ManageService;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

@Service  no usages
public class ManageServiceImpl implements ManageService {

    @Autowired  7 usages
    private CarRepository carRepository;

    @Override  2 usages
    public List<CarDto> getAllCars() throws Exception {
        List<CarDto> cars = new ArrayList<>();
        carRepository.findAll().forEach(car -> cars.add(new CarDto(
                car.getCarId(),
                car.getMake(),
                car.getModel(),
                car.getYear(),
                car.getVin(),
                car.getLicensePlate()
        )));
        return cars;
    }

    @Override  1 usage
    public Optional<CarDetailDto> getCarById(Integer id) throws Exception {
        return carRepository.findById(id).map(car -> new CarDetailDto(
                car.getCarId(),
```



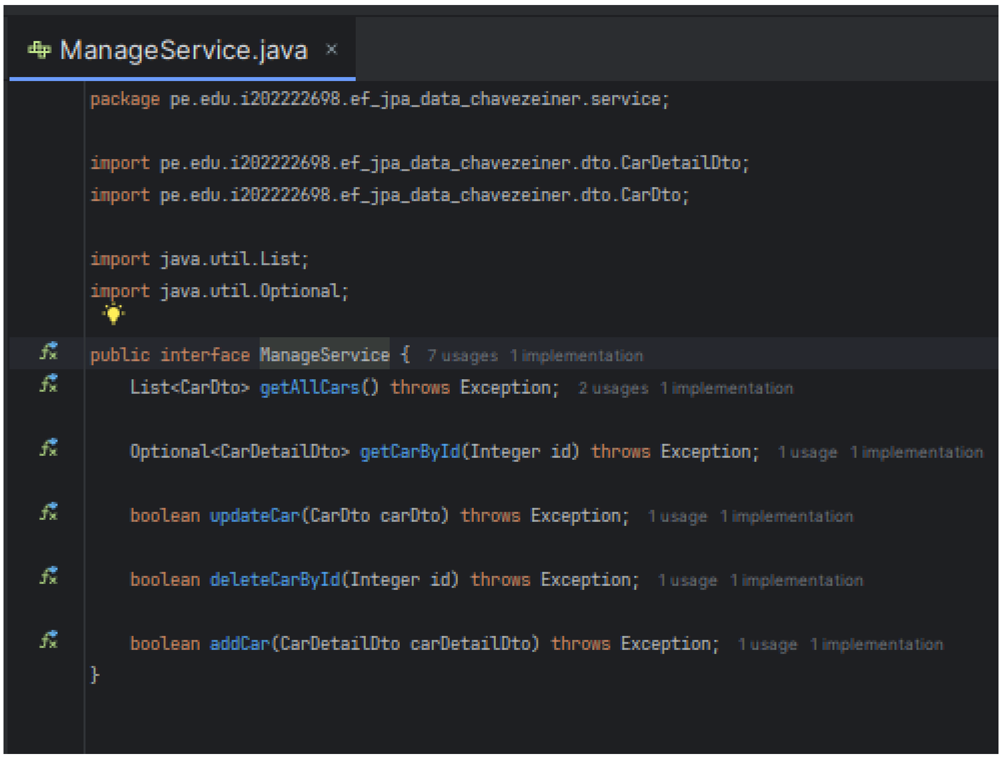
ManageServiceImpl.java ×

```java
public class ManageServiceImpl implements ManageService {
    public Optional<CarDetailDto> getCarById(Integer id) throws Exception {
        return carRepository.findById(id).map(car -> new CarDetailDto(
                car.getMake(),
                car.getModel(),
                car.getYear(),
                car.getVin(),
                car.getLicensePlate(),
                car.getOwnerName(),
                car.getOwnerContact(),
                car.getPurchaseDate(),
                car.getMileage(),
                car.getEngineType(),
                car.getColor(),
                car.getInsuranceCompany(),
                car.getInsurancePolicyNumber(),
                car.getRegistrationExpirationDate(),
                car.getServiceDueDate()
        ));
    }

    @Override  1 usage
    public boolean updateCar(CarDto carDto) throws Exception {
        Optional<Car> optional = carRepository.findById(carDto.carId());
        if (optional.isPresent()) {
            Car car = optional.get();
            car.setMake(carDto.make());
            car.setModel(carDto.model());
            car.setYear(carDto.year());
            car.setVin(carDto.vin());
            car.setLicensePlate(carDto.licensePlate());
            carRepository.save(car);
            return true;
        }
        return false;
    }
```

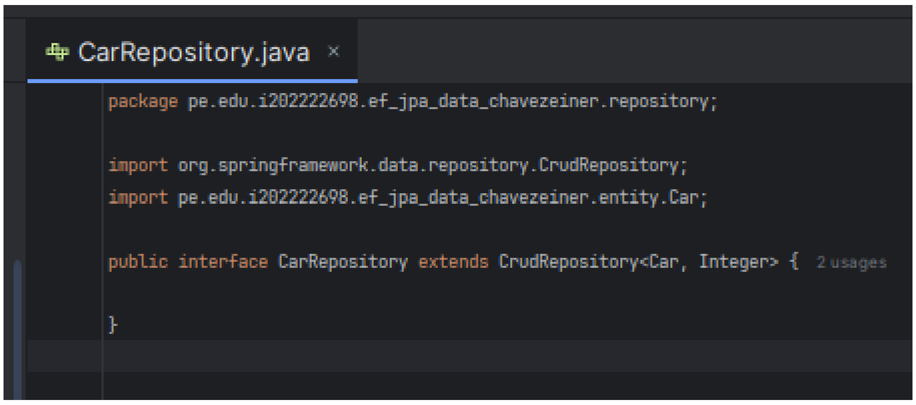


ManageServiceImpl.java ×

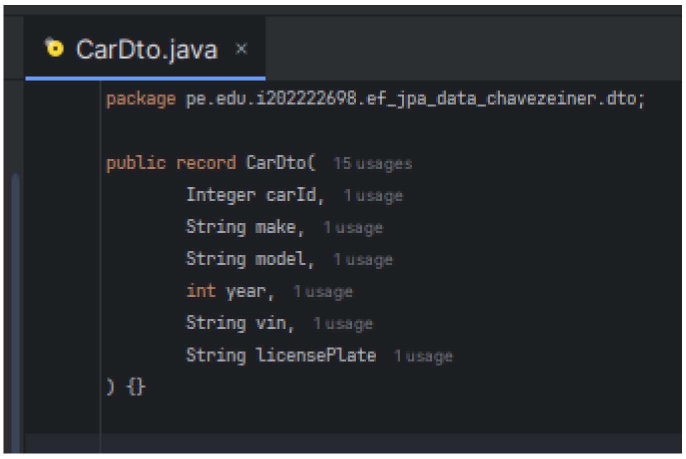
```java
public class ManageServiceImpl implements ManageService {

    @Override  1 usage
    public boolean deleteCarById(Integer id) throws Exception {
        if (carRepository.existsById(id)) {
            carRepository.deleteById(id);
            return true;
        }
        return false;
    }

    @Override  1 usage
    public boolean addCar(CarDetailDto carDetailDto) throws Exception {
        Car car = new Car();
        car.setMake(carDetailDto.make());
        car.setModel(carDetailDto.model());
        car.setYear(carDetailDto.year());
        car.setVin(carDetailDto.vin());
        car.setLicensePlate(carDetailDto.licensePlate());
        car.setOwnerName(carDetailDto.ownerName());
        car.setOwnerContact(carDetailDto.ownerContact());
        car.setPurchaseDate(carDetailDto.purchaseDate());
        car.setMileage(carDetailDto.mileage());
        car.setEngineType(carDetailDto.engineType());
        car.setColor(carDetailDto.color());
        car.setInsuranceCompany(carDetailDto.insuranceCompany());
        car.setInsurancePolicyNumber(carDetailDto.insurancePolicyNumber());
        car.setRegistrationExpirationDate(carDetailDto.registrationExpirationDate());
        car.setServiceDueDate(carDetailDto.serviceDueDate());
        carRepository.save(car); // El carId se generará automáticamente aquí
        return true;
    }
}
```


ManageService.java ×

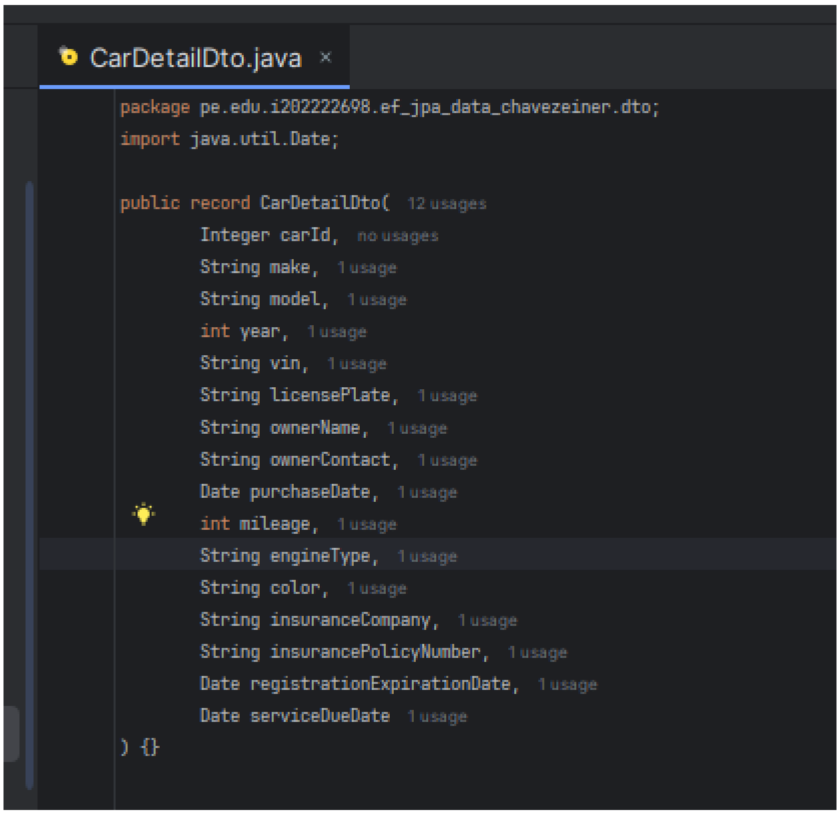
```java
package pe.edu.i202222698.ef_jpa_data_chavezeiner.service;

import pe.edu.i202222698.ef_jpa_data_chavezeiner.dto.CarDetailDto;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.dto.CarDto;

import java.util.List;
import java.util.Optional;

public interface ManageService {  7 usages  1 implementation
    List<CarDto> getAllCars() throws Exception;  2 usages  1 implementation

    Optional<CarDetailDto> getCarById(Integer id) throws Exception;  1 usage  1 implementation

    boolean updateCar(CarDto carDto) throws Exception;  1 usage  1 implementation

    boolean deleteCarById(Integer id) throws Exception;  1 usage  1 implementation

    boolean addCar(CarDetailDto carDetailDto) throws Exception;  1 usage  1 implementation
}
```


CarRepository.java ×

```java
package pe.edu.i202222698.ef_jpa_data_chavezeiner.repository;

import org.springframework.data.repository.CrudRepository;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.entity.Car;

public interface CarRepository extends CrudRepository<Car, Integer> {  2 usages

}
```

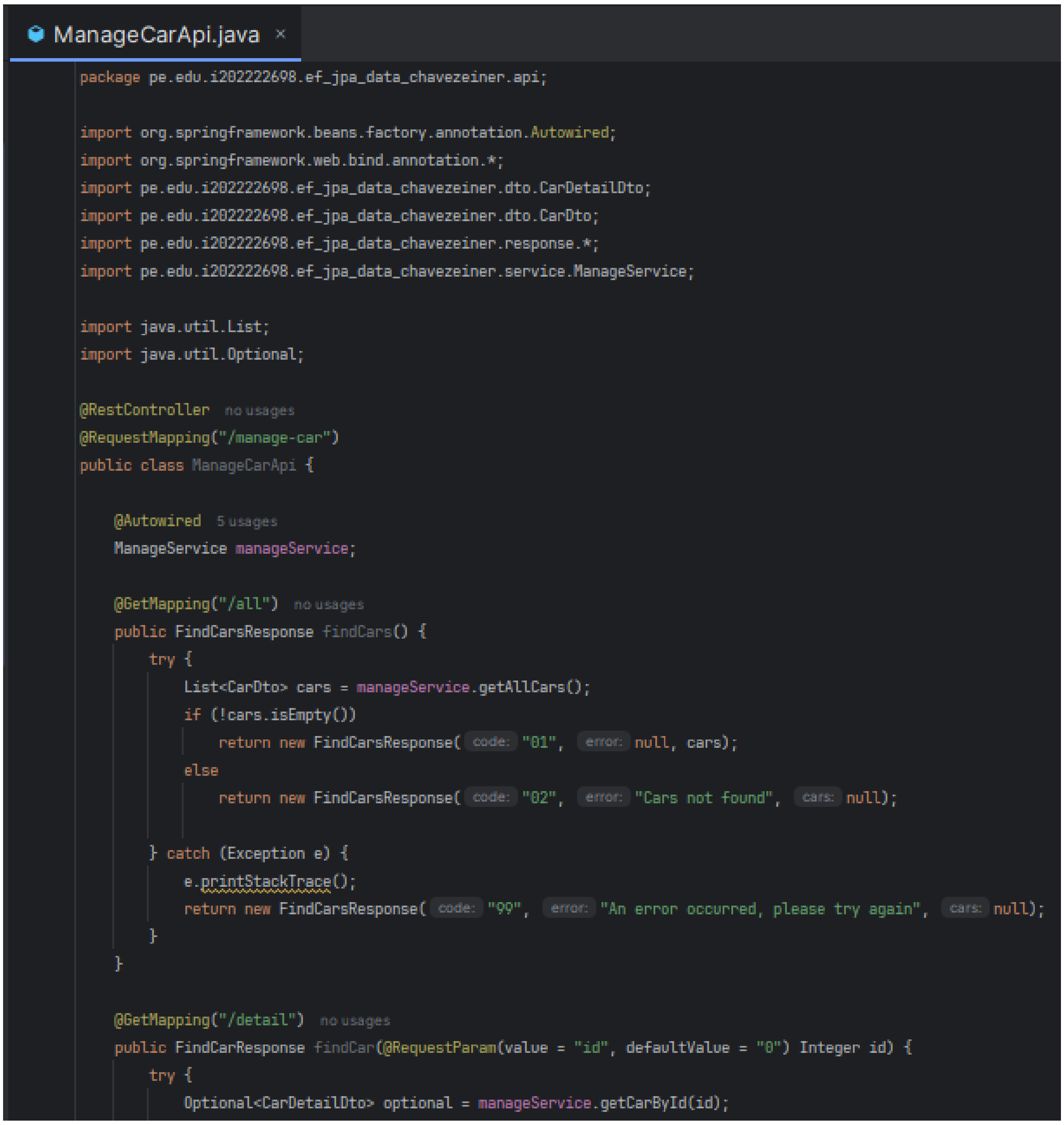

CarDto.java ×

```java
package pe.edu.i202222698.ef_jpa_data_chavezeiner.dto;

public record CarDto(  15 usages
        Integer carId,  1 usage
        String make,  1 usage
        String model,  1 usage
        int year,  1 usage
        String vin,  1 usage
        String licensePlate  1 usage
) {}
```

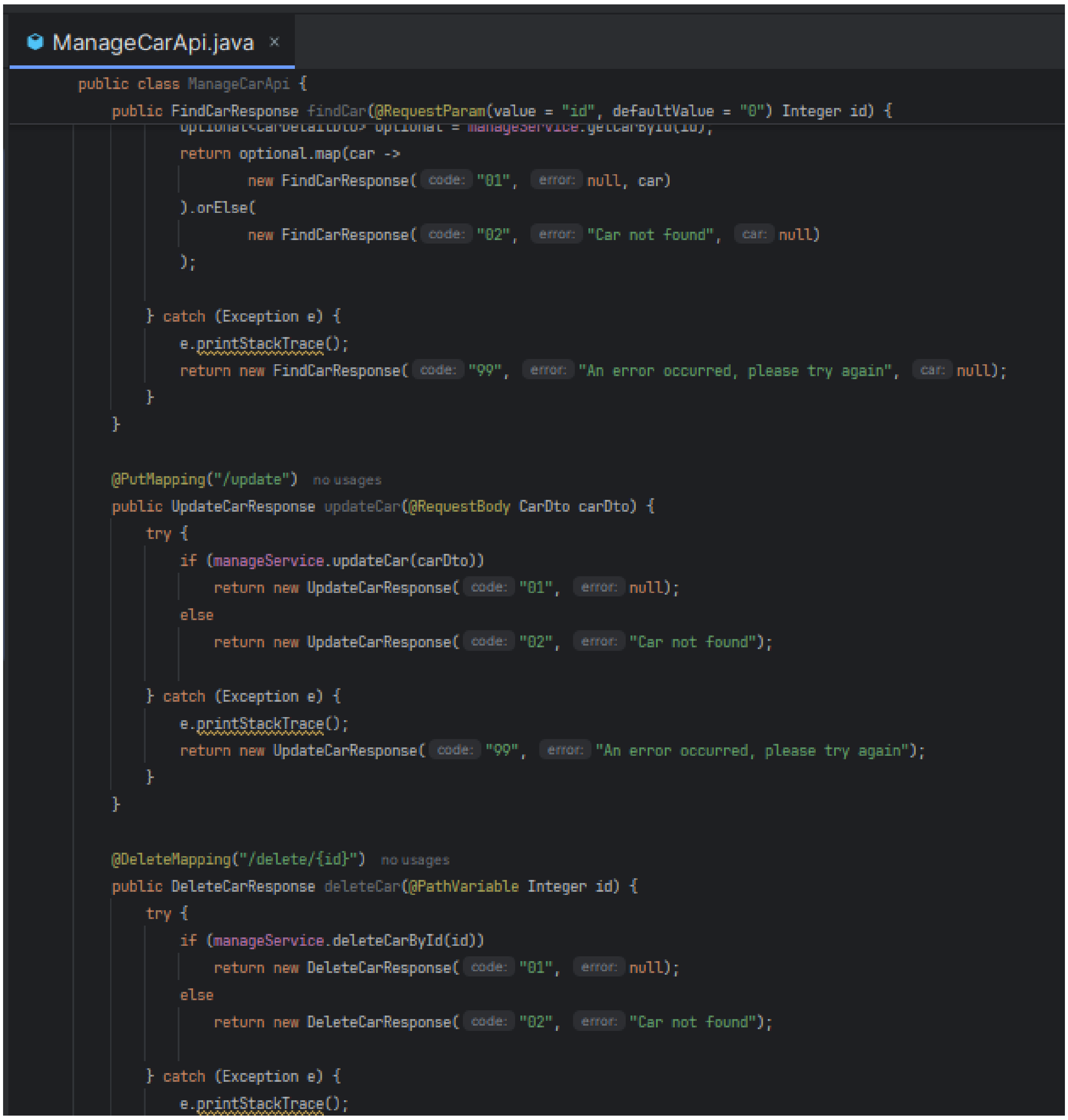


CarDetailDto.java

```java
package pe.edu.i202222698.ef_jpa_data_chavezeiner.dto;
import java.util.Date;

public record CarDetailDto(  12 usages
        Integer carId,  no usages
        String make,  1 usage
        String model,  1 usage
        int year,  1 usage
        String vin,  1 usage
        String licensePlate,  1 usage
        String ownerName,  1 usage
        String ownerContact,  1 usage
        Date purchaseDate,  1 usage
        int mileage,  1 usage
        String engineType,  1 usage
        String color,  1 usage
        String insuranceCompany,  1 usage
        String insurancePolicyNumber,  1 usage
        Date registrationExpirationDate,  1 usage
        Date serviceDueDate  1 usage
) {}
```

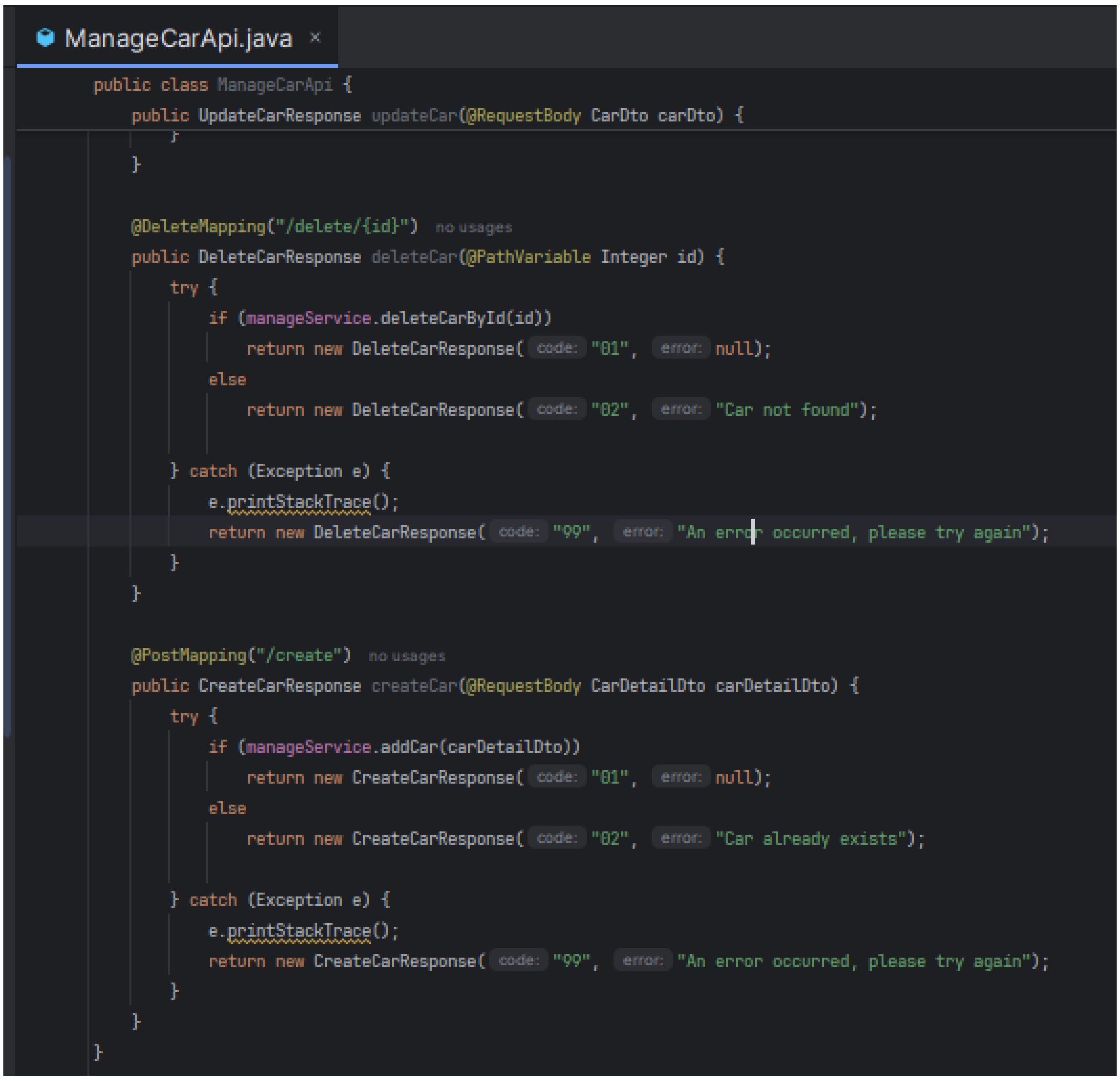


```
package pe.edu.i202222698.ef_jpa_data_chavezeiner.api;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.dto.CarDetailDto;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.dto.CarDto;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.response.*;
import pe.edu.i202222698.ef_jpa_data_chavezeiner.service.ManageService;

import java.util.List;
import java.util.Optional;

@RestController  no usages
@RequestMapping("/manage-car")
public class ManageCarApi {

    @Autowired  5 usages
    ManageService manageService;

    @GetMapping("/all")  no usages
    public FindCarsResponse findCars() {
        try {
            List<CarDto> cars = manageService.getAllCars();
            if (!cars.isEmpty())
                return new FindCarsResponse( code: "01",  error: null, cars);
            else
                return new FindCarsResponse( code: "02",  error: "Cars not found",  cars: null);

        } catch (Exception e) {
            e.printStackTrace();
            return new FindCarsResponse( code: "99",  error: "An error occurred, please try again",  cars: null);
        }
    }

    @GetMapping("/detail")  no usages
    public FindCarResponse findCar(@RequestParam(value = "id", defaultValue = "0") Integer id) {
        try {
            Optional<CarDetailDto> optional = manageService.getCarById(id);
```



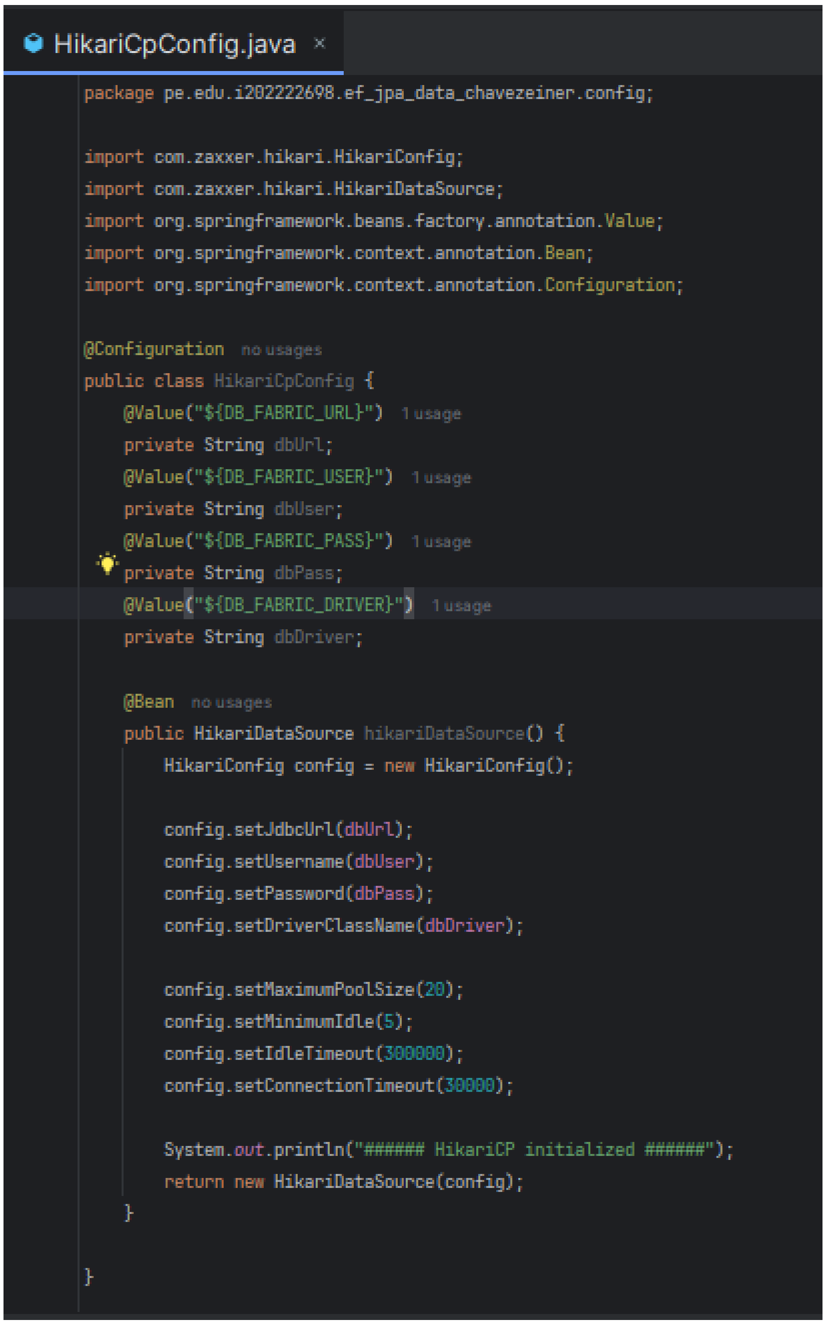
ManageCarApi.java ×

```
    public class ManageCarApi {
        public FindCarResponse findCar(@RequestParam(value = "id", defaultValue = "0") Integer id) {
                Optional<CarDetailDto> optional = manageService.getCarById(id);
                return optional.map(car ->
                        new FindCarResponse( code: "01",  error: null, car)
                ).orElse(
                        new FindCarResponse( code: "02",  error: "Car not found",  car: null)
                );

            } catch (Exception e) {
                e.printStackTrace();
                return new FindCarResponse( code: "99",  error: "An error occurred, please try again",  car: null);
            }
        }

        @PutMapping("/update")  no usages
        public UpdateCarResponse updateCar(@RequestBody CarDto carDto) {
            try {
                if (manageService.updateCar(carDto))
                    return new UpdateCarResponse( code: "01",  error: null);
                else
                    return new UpdateCarResponse( code: "02",  error: "Car not found");

            } catch (Exception e) {
                e.printStackTrace();
                return new UpdateCarResponse( code: "99",  error: "An error occurred, please try again");
            }
        }

        @DeleteMapping("/delete/{id}")  no usages
        public DeleteCarResponse deleteCar(@PathVariable Integer id) {
            try {
                if (manageService.deleteCarById(id))
                    return new DeleteCarResponse( code: "01",  error: null);
                else
                    return new DeleteCarResponse( code: "02",  error: "Car not found");

            } catch (Exception e) {
                e.printStackTrace();
```



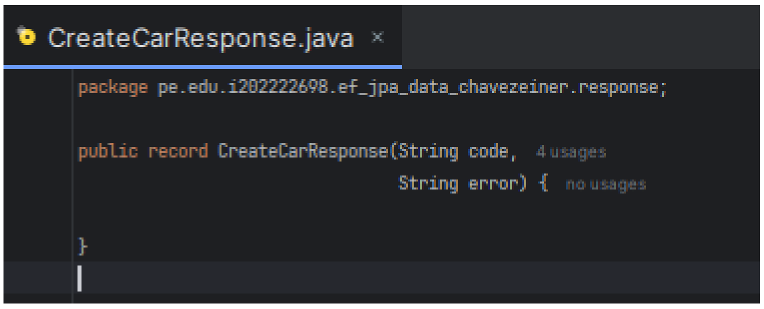
ManageCarApi.java ×

```java
public class ManageCarApi {
    public UpdateCarResponse updateCar(@RequestBody CarDto carDto) {
        }
    }

    @DeleteMapping("/delete/{id}")  no usages
    public DeleteCarResponse deleteCar(@PathVariable Integer id) {
        try {
            if (manageService.deleteCarById(id))
                return new DeleteCarResponse( code: "01",  error: null);
            else
                return new DeleteCarResponse( code: "02",  error: "Car not found");

        } catch (Exception e) {
            e.printStackTrace();
            return new DeleteCarResponse( code: "99",  error: "An error occurred, please try again");
        }
    }

    @PostMapping("/create")  no usages
    public CreateCarResponse createCar(@RequestBody CarDetailDto carDetailDto) {
        try {
            if (manageService.addCar(carDetailDto))
                return new CreateCarResponse( code: "01",  error: null);
            else
                return new CreateCarResponse( code: "02",  error: "Car already exists");

        } catch (Exception e) {
            e.printStackTrace();
            return new CreateCarResponse( code: "99",  error: "An error occurred, please try again");
        }
    }
}
```



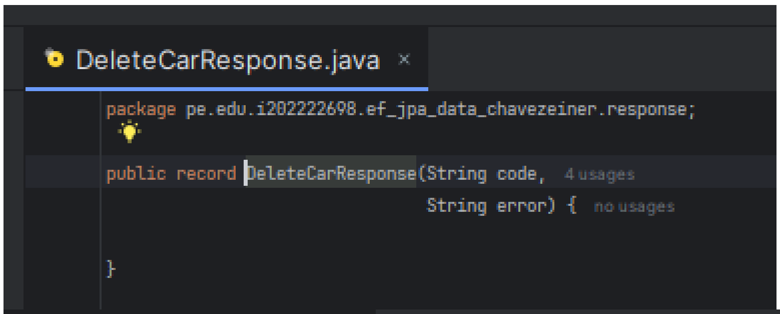
HikariCpConfig.java ×

```java
package pe.edu.i202222698.ef_jpa_data_chavezeiner.config;

import com.zaxxer.hikari.HikariConfig;
import com.zaxxer.hikari.HikariDataSource;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration  no usages
public class HikariCpConfig {
    @Value("${DB_FABRIC_URL}")  1 usage
    private String dbUrl;
    @Value("${DB_FABRIC_USER}")  1 usage
    private String dbUser;
    @Value("${DB_FABRIC_PASS}")  1 usage
    private String dbPass;
    @Value("${DB_FABRIC_DRIVER}")  1 usage
    private String dbDriver;

    @Bean  no usages
    public HikariDataSource hikariDataSource() {
        HikariConfig config = new HikariConfig();

        config.setJdbcUrl(dbUrl);
        config.setUsername(dbUser);
        config.setPassword(dbPass);
        config.setDriverClassName(dbDriver);

        config.setMaximumPoolSize(20);
        config.setMinimumIdle(5);
        config.setIdleTimeout(300000);
        config.setConnectionTimeout(30000);

        System.out.println("###### HikariCP initialized ######");
        return new HikariDataSource(config);
    }

}
```



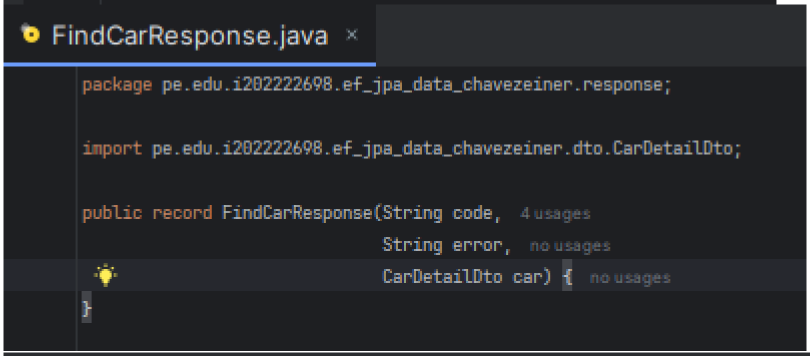
CreateCarResponse.java ×

```java
package pe.edu.i202222698.ef_jpa_data_chavezeiner.response;

public record CreateCarResponse(String code,  4 usages
                                String error) {  no usages

}
```



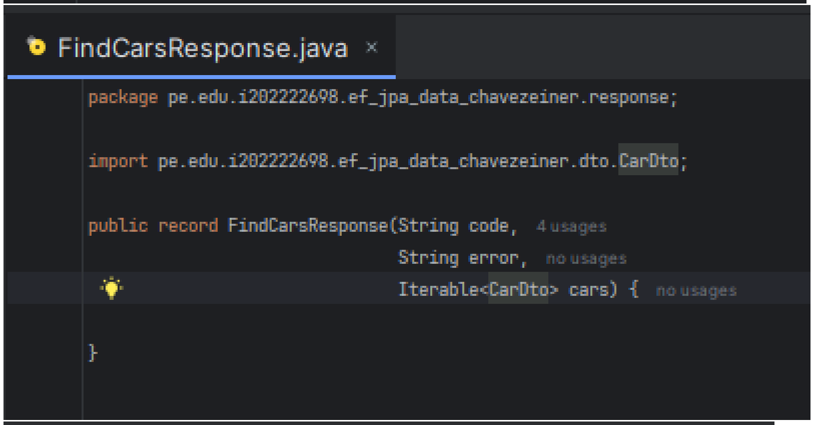
DeleteCarResponse.java

```
package pe.edu.i202222698.ef_jpa_data_chavezeiner.response;

public record DeleteCarResponse(String code,  4 usages
                                String error) {  no usages

}
```



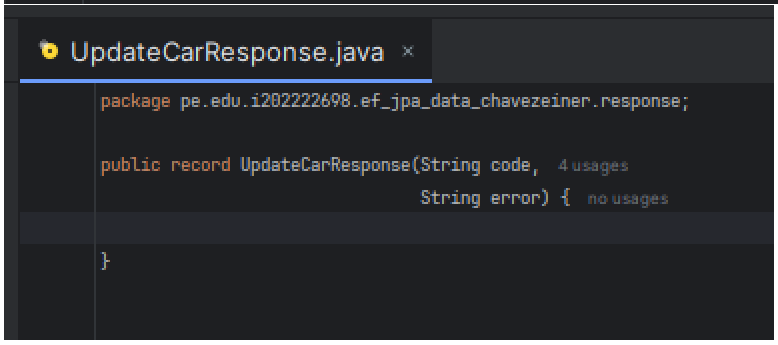
FindCarResponse.java

```
package pe.edu.i202222698.ef_jpa_data_chavezeiner.response;

import pe.edu.i202222698.ef_jpa_data_chavezeiner.dto.CarDetailDto;

public record FindCarResponse(String code,  4 usages
                              String error,  no usages
                              CarDetailDto car) {  no usages
}
```



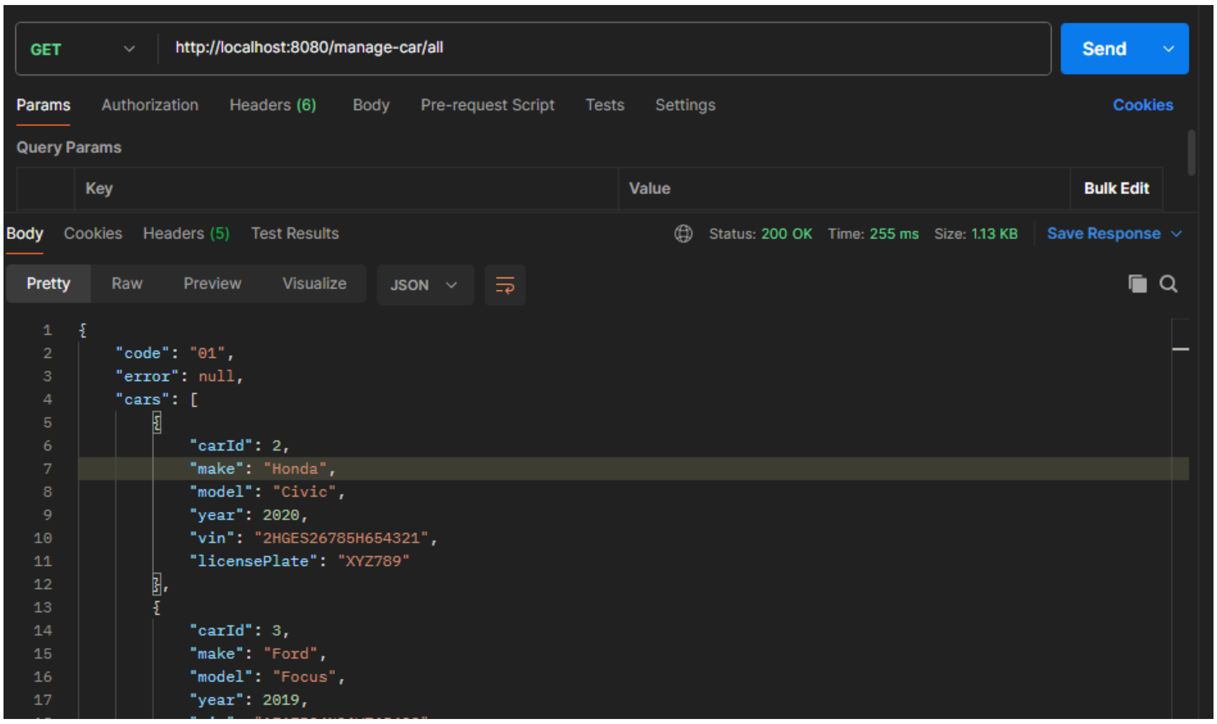
FindCarsResponse.java

```
package pe.edu.i202222698.ef_jpa_data_chavezeiner.response;

import pe.edu.i202222698.ef_jpa_data_chavezeiner.dto.CarDto;

public record FindCarsResponse(String code,  4 usages
                               String error,  no usages
                               Iterable<CarDto> cars) {  no usages

}
```



UpdateCarResponse.java

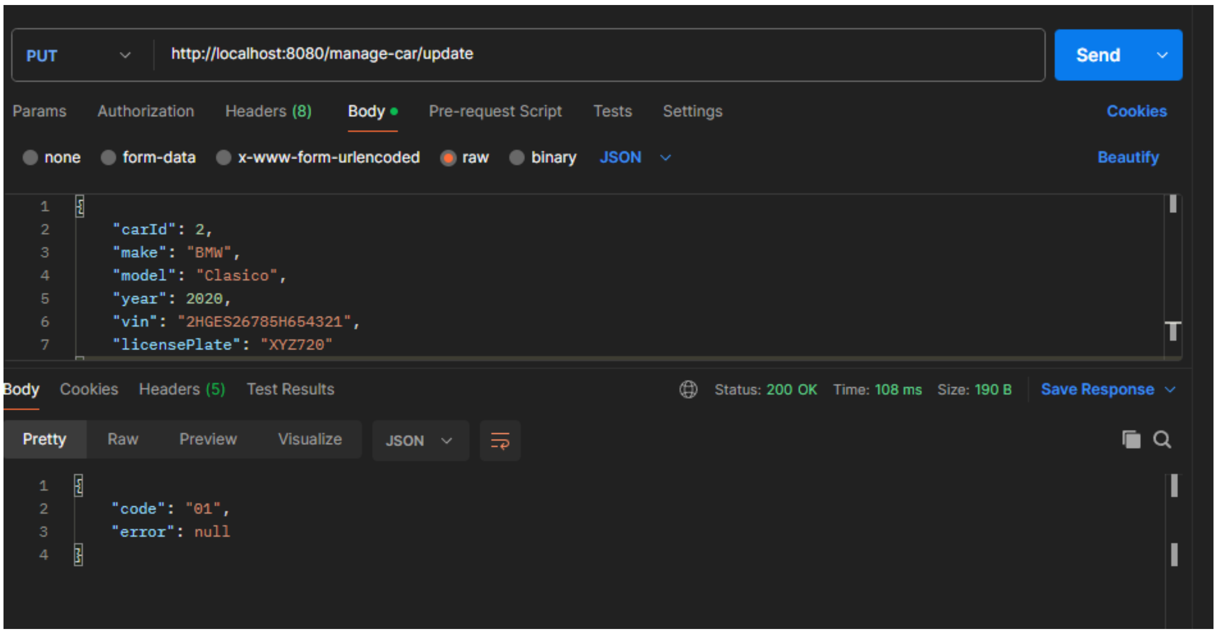
```
package pe.edu.i202222698.ef_jpa_data_chavezeiner.response;

public record UpdateCarResponse(String code,  4 usages
                                String error) {  no usages

}
```



http://localhost:8080/manage-car/all

Status: 200 OK Time: 255 ms Size: 1.13 KB

```
{
    "code": "01",
    "error": null,
    "cars": [
        {
            "carId": 2,
            "make": "Honda",
            "model": "Civic",
            "year": 2020,
            "vin": "2HGES26785H654321",
            "licensePlate": "XYZ789"
        },
        {
            "carId": 3,
            "make": "Ford",
            "model": "Focus",
            "year": 2019,
```

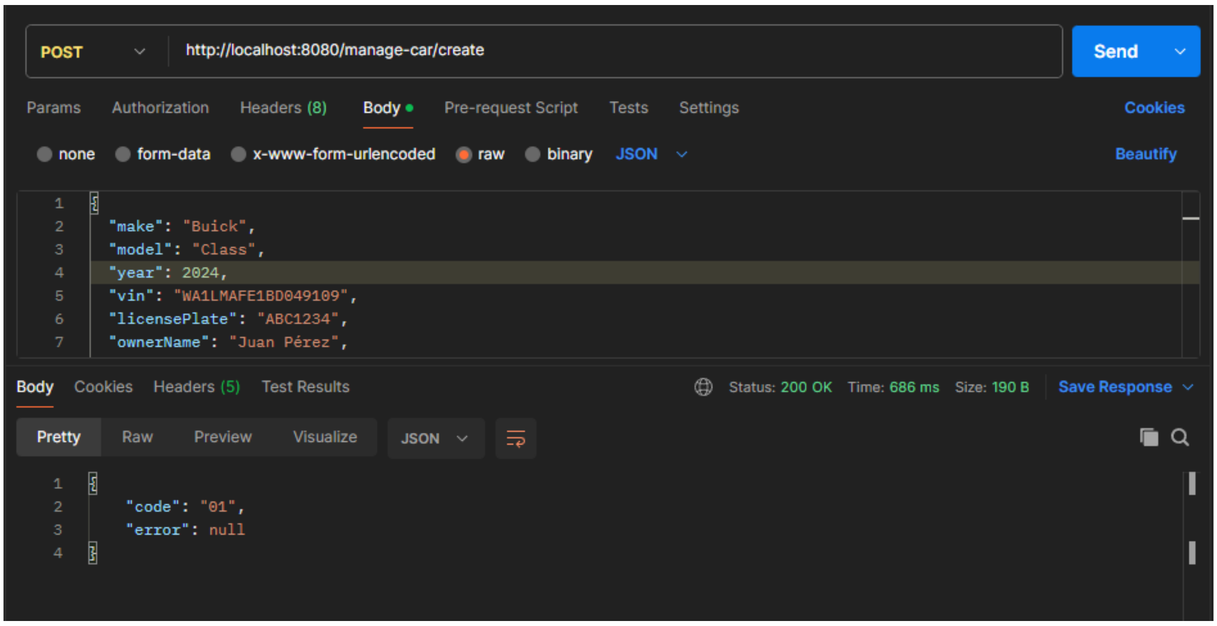


http://localhost:8080/manage-car/update

```
{
    "carId": 2,
    "make": "BMW",
    "model": "Clasico",
    "year": 2020,
    "vin": "2HGES26785H654321",
    "licensePlate": "XYZ720"
```

Status: 200 OK Time: 108 ms Size: 190 B

```
{
    "code": "01",
    "error": null
}
```

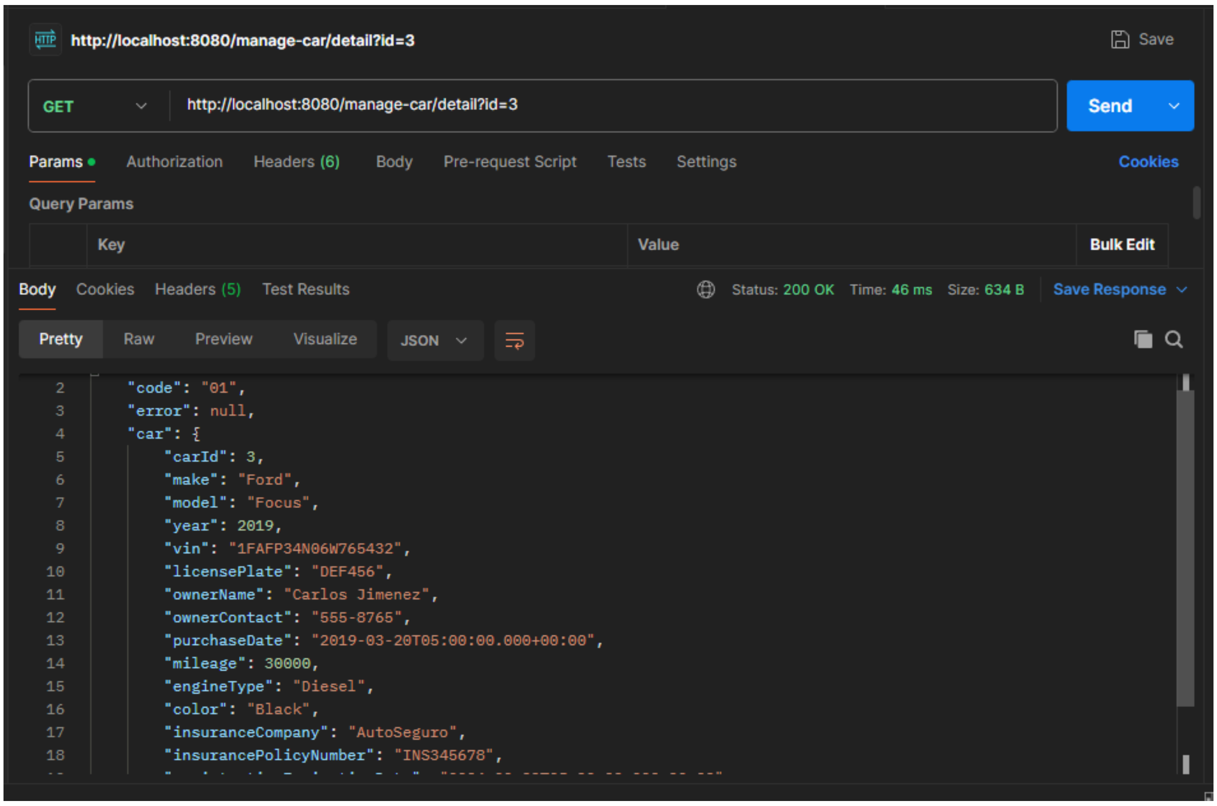


http://localhost:8080/manage-car/create

```
{
  "make": "Buick",
  "model": "Class",
  "year": 2024,
  "vin": "WA1LMAFE1BD049109",
  "licensePlate": "ABC1234",
  "ownerName": "Juan Pérez",
```

Status: 200 OK Time: 686 ms Size: 190 B

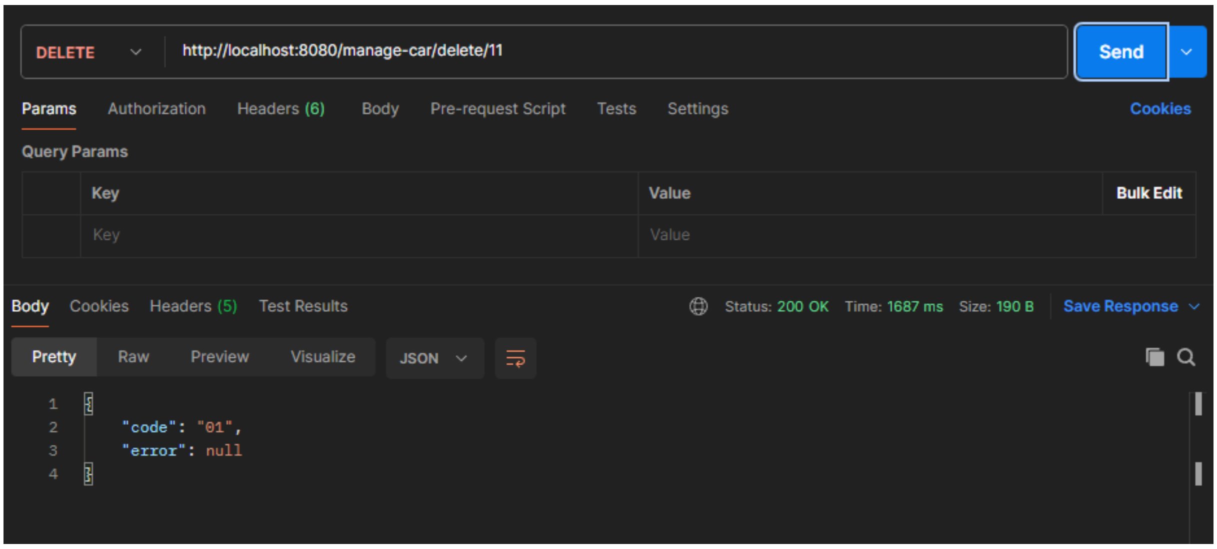
```
{
    "code": "01",
    "error": null
}
```



http://localhost:8080/manage-car/detail?id=3

Status: 200 OK Time: 46 ms Size: 634 B

```
    "code": "01",
    "error": null,
    "car": {
        "carId": 3,
        "make": "Ford",
        "model": "Focus",
        "year": 2019,
        "vin": "1FAFP34N06W765432",
        "licensePlate": "DEF456",
        "ownerName": "Carlos Jimenez",
        "ownerContact": "555-8765",
        "purchaseDate": "2019-03-20T05:00:00.000+00:00",
        "mileage": 30000,
        "engineType": "Diesel",
        "color": "Black",
        "insuranceCompany": "AutoSeguro",
        "insurancePolicyNumber": "INS345678",
```



http://localhost:8080/manage-car/delete/11

Status: 200 OK Time: 1687 ms Size: 190 B

```
{
    "code": "01",
    "error": null
}
```

LINK_GIT HUB: https://github.com/EinerChavez/EF_DATA_DAWI_EINER_CHAVEZ.git