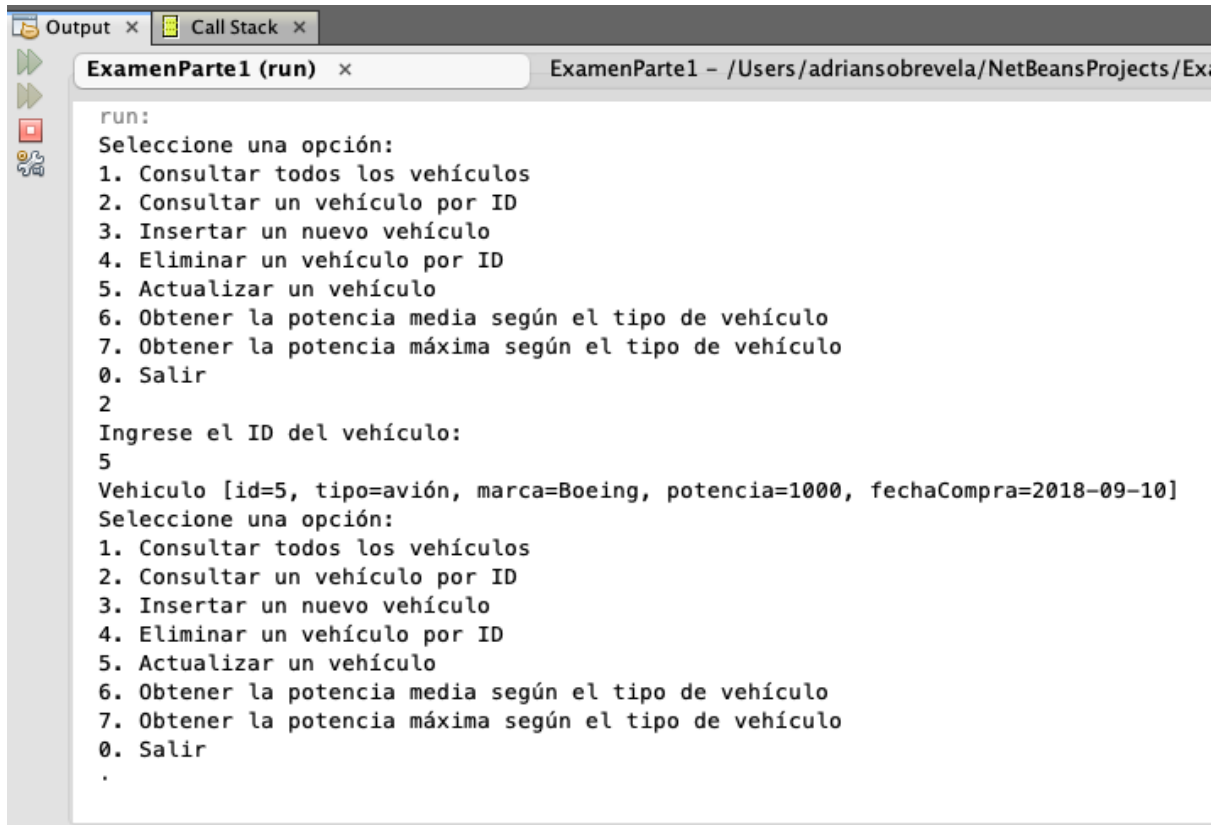


## Primera parte



```
run:
Seleccione una opción:
1. Consultar todos los vehículos
2. Consultar un vehículo por ID
3. Insertar un nuevo vehículo
4. Eliminar un vehículo por ID
5. Actualizar un vehículo
6. Obtener la potencia media según el tipo de vehículo
7. Obtener la potencia máxima según el tipo de vehículo
0. Salir
2
Ingrese el ID del vehículo:
5
Vehiculo [id=5, tipo=avión, marca=Boeing, potencia=1000, fechaCompra=2018-09-10]
Seleccione una opción:
1. Consultar todos los vehículos
2. Consultar un vehículo por ID
3. Insertar un nuevo vehículo
4. Eliminar un vehículo por ID
5. Actualizar un vehículo
6. Obtener la potencia media según el tipo de vehículo
7. Obtener la potencia máxima según el tipo de vehículo
0. Salir
.
```

### databaseconnection.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package examenparte1;

/**
 *
 * @author adriansobrevela
 */
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/vehiculos";
    private static final String USER = "root";
    private static final String PASSWORD = "montanejos";
```

}

 $\ell^*$ 

\*

```

        consultarPorId(scanner);
        break;
    case 3:
        insertarVehiculo(scanner);
        break;
    case 4:
        eliminarVehiculo(scanner);
        break;
    case 5:
        actualizarVehiculo(scanner);
        break;
    case 6:
        obtenerPotenciaMedia(scanner);
        break;
    case 7:
        obtenerPotenciaMaxima(scanner);
        break;
    case 0:
        System.out.println("Saliendo...");
        break;
    default:
        System.out.println("Opción no válida.");
        break;
    }
} while (option != 0);
} catch (SQLException e) {
    e.printStackTrace();
}
}

private static void consultarTodos() throws SQLException {
    List<Vehiculo> vehiculos = vehiculoDAO.consultarTodos();
    for (Vehiculo vehiculo : vehiculos) {
        System.out.println(vehiculo);
    }
}

private static void consultarPorId(Scanner scanner) throws SQLException {
    System.out.println("Ingrese el ID del vehículo:");
    int id = scanner.nextInt();
    Vehiculo vehiculo = vehiculoDAO.consultarPorId(id);
    if (vehiculo != null) {
        System.out.println(vehiculo);
    } else {
        System.out.println("Vehículo no encontrado.");
    }
}
}

```

```

private static void insertarVehiculo(Scanner scanner) throws SQLException {
    System.out.println("Ingrese el tipo del vehículo:");
    String tipo = scanner.nextLine();
    System.out.println("Ingrese la marca del vehículo:");
    String marca = scanner.nextLine();
    System.out.println("Ingrese la potencia del vehículo:");
    int potencia = scanner.nextInt();
    System.out.println("Ingrese la fecha de compra (yyyy-mm-dd):");
    String fecha = scanner.next();

    Vehiculo vehiculo = new Vehiculo(0, tipo, marca, potencia,
java.sql.Date.valueOf(fecha));
    vehiculoDAO.insertarVehiculo(vehiculo);
    System.out.println("Vehículo insertado correctamente.");
}

private static void eliminarVehiculo(Scanner scanner) throws SQLException {
    System.out.println("Ingrese el ID del vehículo a eliminar:");
    int id = scanner.nextInt();
    vehiculoDAO.eliminarVehiculo(id);
    System.out.println("Vehículo eliminado correctamente.");
}

private static void actualizarVehiculo(Scanner scanner) throws SQLException {
    System.out.println("Ingrese el ID del vehículo a actualizar:");
    int id = scanner.nextInt();
    scanner.nextLine(); // Consumir el salto de línea

    System.out.println("Ingrese el nuevo tipo del vehículo:");
    String tipo = scanner.nextLine();
    System.out.println("Ingrese la nueva marca del vehículo:");
    String marca = scanner.nextLine();
    System.out.println("Ingrese la nueva potencia del vehículo:");
    int potencia = scanner.nextInt();
    System.out.println("Ingrese la nueva fecha de compra (yyyy-mm-dd):");
    String fecha = scanner.next();

    Vehiculo vehiculo = new Vehiculo(id, tipo, marca, potencia,
java.sql.Date.valueOf(fecha));
    vehiculoDAO.actualizarVehiculo(vehiculo);
    System.out.println("Vehículo actualizado correctamente.");
}

private static void obtenerPotenciaMedia(Scanner scanner) throws SQLException {
    System.out.println("Ingrese el tipo de vehículo:");
    String tipo = scanner.nextLine();
    double potenciaMedia = vehiculoDAO.obtenerPotenciaMedia(tipo);
}

```

```

        System.out.println("La potencia media de los vehículos tipo " + tipo + " es: " +
potenciaMedia);
    }

    private static void obtenerPotenciaMaxima(Scanner scanner) throws SQLException {
        System.out.println("Ingrese el tipo de vehículo:");
        String tipo = scanner.nextLine();
        int potenciaMaxima = vehiculoDAO.obtenerPotenciaMaxima(tipo);
        System.out.println("La potencia máxima de los vehículos tipo " + tipo + " es: " +
potenciaMaxima);
    }
}
}
}

```

## VEHICULO.JAVA

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package examenparte1;

import java.util.Date;

public class Vehiculo {
    private int id;
    private String tipo;
    private String marca;
    private int potencia;
    private Date fechaCompra;

    public Vehiculo(int id, String tipo, String marca, int potencia, Date fechaCompra) {
        this.id = id;
        this.tipo = tipo;
        this.marca = marca;
        this.potencia = potencia;
        this.fechaCompra = fechaCompra;
    }

    // Getters y Setters
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

```

```

    }

    public String getTipo() {
        return tipo;
    }

    public void setTipo(String tipo) {
        this.tipo = tipo;
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public int getPotencia() {
        return potencia;
    }

    public void setPotencia(int potencia) {
        this.potencia = potencia;
    }

    public Date getFechaCompra() {
        return fechaCompra;
    }

    public void setFechaCompra(Date fechaCompra) {
        this.fechaCompra = fechaCompra;
    }

    @Override
    public String toString() {
        return "Vehiculo [id=" + id + ", tipo=" + tipo + ", marca=" + marca + ", potencia="
+ potencia + ", fechaCompra=" + fechaCompra + "];"
    }
}

```

**VehiculoDao.java**

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

```

package examenparte1;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author adriansobrevela
 */
public class VehiculoDAO {
    private Connection connection;

    public VehiculoDAO(Connection connection) {
        this.connection = connection;
    }

    public List<Vehiculo> consultarTodos() throws SQLException {
        List<Vehiculo> vehiculos = new ArrayList<>();
        String query = "SELECT * FROM vehiculos";
        Statement stmt = connection.createStatement();
        ResultSet rs = stmt.executeQuery(query);

        while (rs.next()) {
            Vehiculo vehiculo = new Vehiculo(
                rs.getInt("id"),
                rs.getString("tipo"),
                rs.getString("marca"),
                rs.getInt("potencia"),
                rs.getDate("fecha_compra")
            );
            vehiculos.add(vehiculo);
        }
        return vehiculos;
    }

    public Vehiculo consultarPorId(int id) throws SQLException {
        String query = "SELECT * FROM vehiculos WHERE id = ?";
        PreparedStatement pstmt = connection.prepareStatement(query);
        pstmt.setInt(1, id);
        ResultSet rs = pstmt.executeQuery();

        if (rs.next()) {
            return new Vehiculo(
                rs.getInt("id"),
                rs.getString("tipo"),
                rs.getString("marca"),
                rs.getInt("potencia"),
                rs.getDate("fecha_compra")
            );
        }
        return null;
    }
}

```

```

        rs.getDate("fecha_compra")
    );
}
return null;
}

public void insertarVehiculo(Vehiculo vehiculo) throws SQLException {
    String query = "INSERT INTO vehiculos (tipo, marca, potencia, fecha_compra)
VALUES (?, ?, ?, ?)";
    PreparedStatement pstmt = connection.prepareStatement(query);
    pstmt.setString(1, vehiculo.getTipo());
    pstmt.setString(2, vehiculo.getMarca());
    pstmt.setInt(3, vehiculo.getPotencia());
    pstmt.setDate(4, new java.sql.Date(vehiculo.getFechaCompra().getTime()));
    pstmt.executeUpdate();
}

public void eliminarVehiculo(int id) throws SQLException {
    String query = "DELETE FROM vehiculos WHERE id = ?";
    PreparedStatement pstmt = connection.prepareStatement(query);
    pstmt.setInt(1, id);
    pstmt.executeUpdate();
}

public void actualizarVehiculo(Vehiculo vehiculo) throws SQLException {
    String query = "UPDATE vehiculos SET tipo = ?, marca = ?, potencia = ?,
fecha_compra = ? WHERE id = ?";
    PreparedStatement pstmt = connection.prepareStatement(query);
    pstmt.setString(1, vehiculo.getTipo());
    pstmt.setString(2, vehiculo.getMarca());
    pstmt.setInt(3, vehiculo.getPotencia());
    pstmt.setDate(4, new java.sql.Date(vehiculo.getFechaCompra().getTime()));
    pstmt.setInt(5, vehiculo.getId());
    pstmt.executeUpdate();
}

public double obtenerPotenciaMedia(String tipo) throws SQLException {
    String query = "SELECT AVG(potencia) FROM vehiculos WHERE tipo = ?";
    PreparedStatement pstmt = connection.prepareStatement(query);
    pstmt.setString(1, tipo);
    ResultSet rs = pstmt.executeQuery();

    if (rs.next()) {
        return rs.getDouble(1);
    }
    return 0.0;
}

```

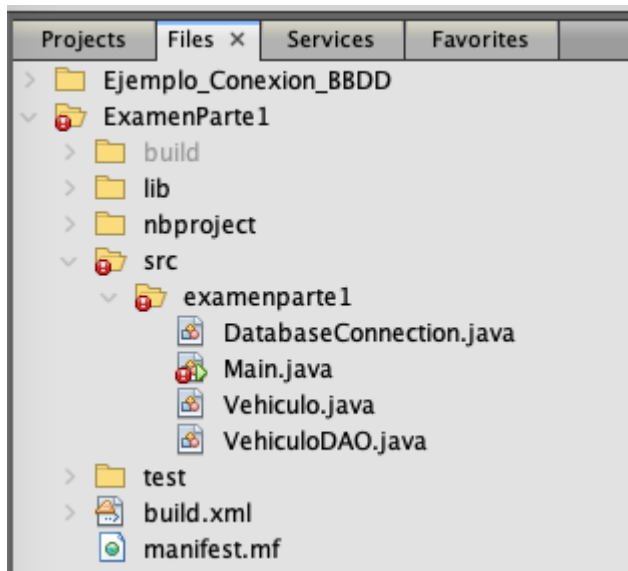


```

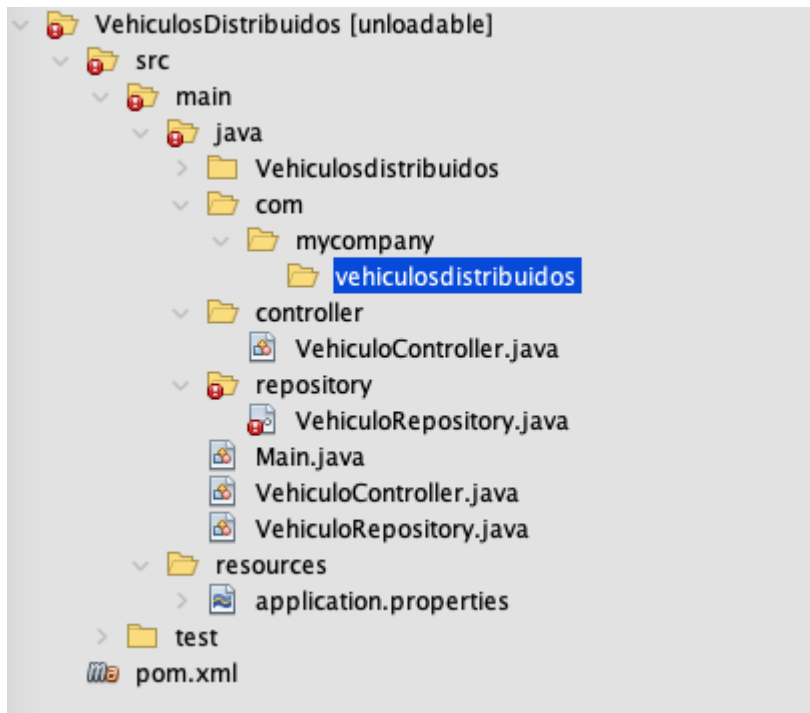
public int obtenerPotenciaMaxima(String tipo) throws SQLException {
    String query = "SELECT MAX(potencia) FROM vehiculos WHERE tipo = ?";
    PreparedStatement pstmt = connection.prepareStatement(query);
    pstmt.setString(1, tipo);
    ResultSet rs = pstmt.executeQuery();

    if (rs.next()) {
        return rs.getInt(1);
    }
    return 0;
}
}

```



## Parte 2



Este no me dio resultado.

pom.xml

```
<dependencies>
  <!-- Dependencias de Spring Boot -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <!-- Driver MySQL -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
  </dependency>
  <!-- Dependencias de prueba -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

### vehiculo.java

```
package Vehiculosdistribuidos;
```

```
import java.sql.Date;
```

```
public class Vehiculo {
    private int id;
    private String tipo;
    private String marca;
    private int potencia;
    private Date fechaCompra;

    // Constructor
    public Vehiculo(int id, String tipo, String marca, int potencia, Date fechaCompra) {
        this.id = id;
        this.tipo = tipo;
        this.marca = marca;
        this.potencia = potencia;
        this.fechaCompra = fechaCompra;
    }

    // Getters and Setters
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getTipo() {
        return tipo;
    }

    public void setTipo(String tipo) {
        this.tipo = tipo;
    }
}
```

```

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public int getPotencia() {
        return potencia;
    }

    public void setPotencia(int potencia) {
        this.potencia = potencia;
    }

    public Date getFechaCompra() {
        return fechaCompra;
    }

    public void setFechaCompra(Date fechaCompra) {
        this.fechaCompra = fechaCompra;
    }

    @Override
    public String toString() {
        return "Vehiculo [id=" + id + ", tipo=" + tipo + ", marca=" + marca + ", potencia=" +
potencia + ", fechaCompra=" + fechaCompra + "]";
    }
}

```

### **VehiculoController.java**

```

package com.example.vehiculosdistribuidos.controller;

import com.example.vehiculosdistribuidos.Vehiculo;
import com.example.vehiculosdistribuidos.repository.VehiculoRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/vehiculos")
public class VehiculoController {

    @Autowired
    private VehiculoRepository vehiculoRepository;

```

```

    @GetMapping
    public List<Vehiculo> consultarTodos() {
        return vehiculoRepository.consultarTodos();
    }

    @GetMapping("/{id}")
    public Vehiculo consultarPorId(@PathVariable int id) {
        return vehiculoRepository.consultarPorId(id);
    }

    @PostMapping
    public void insertarVehiculo(@RequestBody Vehiculo vehiculo) {
        vehiculoRepository.insertarVehiculo(vehiculo);
    }

    @DeleteMapping("/{id}")
    public void eliminarVehiculo(@PathVariable int id) {
        vehiculoRepository.eliminarVehiculo(id);
    }

    @PutMapping("/{id}")
    public void actualizarVehiculo(@PathVariable int id, @RequestBody Vehiculo
vehiculo) {
        vehiculo.setId(id);
        vehiculoRepository.actualizarVehiculo(vehiculo);
    }

    @GetMapping("/potenciaMedia/{tipo}")
    public double obtenerPotenciaMedia(@PathVariable String tipo) {
        return vehiculoRepository.obtenerPotenciaMedia(tipo);
    }

    @GetMapping("/potenciaMaxima/{tipo}")
    public int obtenerPotenciaMaxima(@PathVariable String tipo) {
        return vehiculoRepository.obtenerPotenciaMaxima(tipo);
    }
}

```

#### **Vehiculorepository.java**

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**

```

```

*
* @author adriansobrevela
*/
package vehiculosdistribuidos.repository;

import com.example.vehiculosdistribuidos.Vehiculo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;

@Repository
public class VehiculoRepository {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    public List<Vehiculo> consultarTodos() {
        String sql = "SELECT * FROM vehiculos";
        return jdbcTemplate.query(sql, (rs, rowNum) -> mapRowToVehiculo(rs));
    }

    public Vehiculo consultarPorId(int id) {
        String sql = "SELECT * FROM vehiculos WHERE id = ?";
        return jdbcTemplate.queryForObject(sql, new Object[]{id}, (rs, rowNum) ->
mapRowToVehiculo(rs));
    }

    public void insertarVehiculo(Vehiculo vehiculo) {
        String sql = "INSERT INTO vehiculos (tipo, marca, potencia, fecha_compra) VALUES
(?, ?, ?, ?)";
        jdbcTemplate.update(sql, vehiculo.getTipo(), vehiculo.getMarca(),
vehiculo.getPotencia(), vehiculo.getFechaCompra());
    }

    public void eliminarVehiculo(int id) {
        String sql = "DELETE FROM vehiculos WHERE id = ?";
        jdbcTemplate.update(sql, id);
    }

    public void actualizarVehiculo(Vehiculo vehiculo) {
        String sql = "UPDATE vehiculos SET tipo = ?, marca = ?, potencia = ?, fecha_compra =
? WHERE id = ?";
        jdbcTemplate.update(sql, vehiculo.getTipo(), vehiculo.getMarca(),
vehiculo.getPotencia(), vehiculo.getFechaCompra(), vehiculo.getId());
    }

```

```

    }

    public double obtenerPotenciaMedia(String tipo) {
        String sql = "SELECT AVG(potencia) FROM vehiculos WHERE tipo = ?";
        return jdbcTemplate.queryForObject(sql, new Object[]{tipo}, Double.class);
    }

    public int obtenerPotenciaMaxima(String tipo) {
        String sql = "SELECT MAX(potencia) FROM vehiculos WHERE tipo = ?";
        return jdbcTemplate.queryForObject(sql, new Object[]{tipo}, Integer.class);
    }

    private Vehiculo mapRowToVehiculo(ResultSet rs) throws SQLException {
        return new Vehiculo(
            rs.getInt("id"),
            rs.getString("tipo"),
            rs.getString("marca"),
            rs.getInt("potencia"),
            rs.getDate("fecha_compra")
        );
    }
}

```

## Main.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author adriansobrevela
 */
package VehiculosDistribuidos;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Main {
    public static void main(String[] args) {
        SpringApplication.run(Main.class, args);
    }
}

```

vehiculorepository.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package repository;

/**
 *
 * @author adriansobrevela
 */
package VehiculosDistribuidos.repository;

import com.example.vehiculosDistribuidos.model.Vehiculo;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface VehiculoRepository extends JpaRepository<Vehiculo, Long> {

    @Query("SELECT AVG(v.potencia) FROM Vehiculo v WHERE v.tipo = :tipo")
    Double findPotenciaMediaByTipo(String tipo);

    @Query("SELECT MAX(v.potencia) FROM Vehiculo v WHERE v.tipo = :tipo")
    Integer findPotenciaMaximaByTipo(String tipo);
}
```

application.properties

```
# To change this license header, choose License Headers in Project Properties.
# To change this template file, choose Tools | Templates
# and open the template in the editor.
spring.datasource.url=jdbc:mysql://localhost:3306/vehiculos
spring.datasource.username=root
spring.datasource.password=montanejos
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
```