Ejercicio 3. Consultas DTO y Paging.

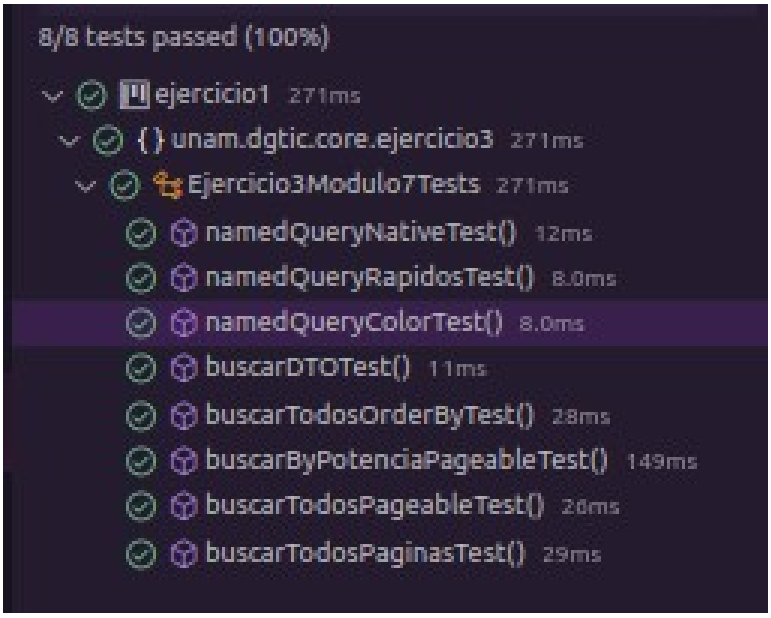
```java
package unam.dgtic.core.ejercicio3;

import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.data.domain.PageRequest;
import org.springframework.data.domain.Pageable;
import org.springframework.data.domain.Sort;

@SpringBootTest
public class Ejercicio3Modulo7Tests {

    @Autowired
    AutomovilRepositorio repositorioAutomovil;

    @Autowired
    AutomovilPagingAndSortingRepository repositorioAutomovilPS;

    @Test
    void namedQueryNativeTest() {
        Iterable<Automovil> iterable = repositorioAutomovil.buscarTodosConDueno();
        iterable.forEach(System.out::println);

    }

    @Test
    void namedQueryRapidosTest() {
        Iterable<Automovil> iterable = repositorioAutomovil.buscarRapidos();
        iterable.forEach(System.out::println);

    }

    @Test
    void namedQueryColorTest() {
        Iterable<Automovil> iterable = repositorioAutomovil.buscarPorColor();
        iterable.forEach(System.out::println);
    }

    @Test
    void buscarDTOTest() {
        Iterable<AutomovilDTO> iterable = repositorioAutomovil.findAutomovilDTO();

        iterable.forEach(System.out::println);
    }

    @Test
    void buscarTodosOrderByTest() {
        Iterable<Automovil> iterable = repositorioAutomovilPS.findByOrderByMarca();
        iterable.forEach(System.out::println);

        iterable = repositorioAutomovilPS.findByOrderByMarcaDescModeloDesc();
        iterable.forEach(System.out::println);
    }

    @Test
    void buscarByPotenciaPageableTest() {
        Pageable pag1 = PageRequest.of(page: 0, size: 5);
        Iterable<Automovil> automovi = repositorioAutomovilPS.findAllByPotencia(potencia: 280, pag1);
        automovi.forEach(System.out::println);

        Pageable pag2 = PageRequest.of(page: 1, size: 5);
        automovi = repositorioAutomovilPS.findAllByPotencia(potencia: 717, pag2);
        automovi.forEach(System.out::println);
    }
```

```java
    @Test
    void buscarTodosPageableTest() {
        Pageable pag1 = PageRequest.of(page: 0, size: 2, Sort.by(...properties: "marca").ascending());
        Iterable<Automovil> automovi = repositorioAutomovilPS.findAll(pag1);
        automovi.forEach(System.out::println);

        Pageable pag2 = PageRequest.of(page: 1, size: 2, Sort.by(...properties: "modelo").descending());
        automovi = repositorioAutomovilPS.findAll(pag2);
        automovi.forEach(System.out::println);
    }

    @Test
    void buscarTodosPaginasTest() {
        Pageable pagina;
        Iterable<Automovil> iterable;

        for (int i = 0; i <= 4; i++) {
            System.out.println("Página " + i);
            pagina = PageRequest.of(i, size: 2, Sort.by(...properties: "color").descending());

            iterable = repositorioAutomovilPS.findAll(pagina);
            iterable.forEach(System.out::println);
        }
    }
}
```

Repositorio.

```java
    // Named Queries.
    List<Automovil> buscarRapidos();

    List<Automovil> buscarTodosConDueno();

    // Native Query.
    List<Automovil> buscarPorColor();

    // Query en repositorio usando JPQL.
    @Query("select avg(a.torque) from Automovil a")
    public double buscarTorquePromedioAutomovil();

    // Query en repositorio usando SQL.
    @Query(value = "select distinct a.* from Automovil a \n" +
            "join Dueno c on (a.id_automovil = c.id_automovil) \n" +
            "order by marca", nativeQuery = true)
    public List<Automovil> buscarAutomovilConDueno();

    // Automovil DTO.
    @Query(value = "select distinct new unam.dgtic.core.ejercicio3.AutomovilDTO(" +
            "a.marca, a.modelo, a.color, a.torque, a.potencia, d.nombre) " +
            "from Automovil a, Dueno d " +
            "where a.idAutomovil = d.automovil")
    List<AutomovilDTO> findAutomovilDTO();
```

Ejecución pruebas.



8 pruebas realizadas con éxito.