Ejercicio: Simulación de una Carrera de Fórmula 1

Enunciado:

Vas a crear una pequeña simulación de una carrera de Fórmula 1 utilizando Programación Orientada a Objetos (POO). En este ejercicio, debéis crear clases que representen distintos tipos de **vehículos** de carrera (**CocheF1** y **CocheElectricoF1**), con sus atributos y métodos correspondientes.

Los conceptos que se van a aplicar son:

- Clases para representar los vehículos.
- Herencia para crear diferentes tipos de vehículos que hereden de una clase base.
- Métodos **protegidos** para los cálculos internos de los coches.
- Métodos **públicos** para interactuar con los coches desde el exterior.
- Constructor para inicializar los coches con ciertos atributos.
- **Destructor** para indicar cuando un coche se ha "retirado" de la carrera.

Instrucciones:

1. Clase base VehiculoCarrera:

- Atributos protegidos:
 - marca: La marca del coche.
 - **modelo**: El modelo del coche.
 - velocidad: La velocidad máxima del coche.
 - **combustible**: Nivel de combustible.
- Métodos protegidos:
 - consumirCombustible(): Método que reduce el combustible al moverse el coche.
- Métodos públicos:
 - **arrancar**(): Arranca el coche.
 - acelerar(): Incrementa la velocidad del coche.
 - **detener**(): Detiene el coche.
 - mostrarEstado(): Muestra el estado actual del coche (marca, modelo, velocidad y combustible).
- o Constructor:
 - Inicializa los atributos marca, modelo, velocidad, y combustible.
- o Destructor:
 - Muestra un mensaje indicando que el coche se ha retirado de la carrera.

2. Clase hija CocheF1:

- o Hereda de VehiculoCarrera.
- o Atributo extra:
 - **alerones**: Indica si tiene alerones mejorados.
- Métodos específicos:
 - activarDRS(): Método que simula la activación del DRS para aumentar la velocidad temporalmente.

3. Clase hija CocheElectricoF1:

o Hereda de VehiculoCarrera.

- o Atributo extra:
 - bateria: Nivel de batería del coche.
- o Métodos específicos:
 - recargar(): Método que simula la recarga de la batería.