

Ingeniería en Sistemas y Ciencias de la Computación

## **Tarea**

**Estudiante:**

**José Carlos Morataya Enrique**



Programación 1

**Jutiapa, marzo 2025**

## Proceso

### Punto 1: Creé las clases derivadas de la clase `Vehículo`

Lo primero que hice fue en la parte de `Vehículo`, ya que pues es como nuestro plano de lo que vamos a hacer, o sea, lo primero que modifiqué fue la clase `Vehículo`. Hice el encendido y el apagado, modifiqué la parte de `acelerar` y `frenar` donde me hace una validación si está encendido o no, o sea, si no está encendido no acelera, pero si está encendido, pues acelera. Entonces fui agarrando mediante esa lógica lo mismo con `frenar`, y en la parte de `encender` hice una validación donde si le dan dos veces a encender que aparezca, pues, de que el vehículo ya está encendido. Y de la misma manera con el vehículo, si le dan dos veces a `apagar`, pues ya está apagado, y que salga ese mensaje.

Luego derivé la clase `CarroDeCombustion` de la clase `Vehiculo`. Le puse, 3 encapsulamientos donde uno es `gas`, otro es `capacidadTanque` y otro es `tipoGasolina`, donde yo hice que fueran propiedades, o sea, que el usuario ingresará en esas propiedades lo que quiere darles.

Después derivé la clase `Motocicleta` de `Vehiculo`, hice un encapsulamiento de 3 variables que sería `cilindrada`, `gas` y `manubrio`, donde de igual manera se lo pedí al usuario, o sea, lo ingresé a propiedades.

Y por último derivé la clase `Camion` de `Vehiculo`, y en el encapsulamiento puse `cargaMaxima`, `gas` y `tipoGasolina`, que siempre se lo almacena en propiedades para que el usuario ingrese qué es lo que va a meter o qué tipo.

### Punto 2: Sobrescribí los métodos en las clases derivadas

En la clase base `Vehiculo`, ya tenía los métodos `acelerar`, `frenar`, `encender` y `apagar`, y los hice `virtual` para que se pudieran sobrescribir. En `CarroDeCombustion`, edité la información de vehículo, donde le agregué el `tipoGasolina` y la `capacidadTanque` para que dé la información, y de igual manera le puse un tipo de separadores para que se logre entender el código al momento de leerlo. Luego, agarré la lógica de `frenar` de `Vehiculo`, lo mandé a llamar con un `override` donde dice que gastará un 1% de gasolina. Luego de eso, creé un `void` donde vamos a cargar gasolina y hace una validación donde no supere la capacidad del tanque, o sea, que no sea mayor a la capacidad del tanque, que no pueda meterle, si en dado caso la capacidad del tanque son 100 litros, que no le pueda meter 101 L, o sea, más de eso. Lo hice un `void` donde muestra la gasolina restante. Modifiqué el `void` de `acelerar` con `override`

donde si no hay gasolina no puede acelerar y donde se le gasta la gasolina un 2%. Y luego, al momento de `encender`, lo sobrescribí para que si en dado caso no tiene gasolina, no encenderá.

En la clase `Motocicleta`, modifiqué la información del vehículo, agregando la `cilindrada` y el `manubrio`. Luego agarré la lógica de validación, siempre de que al momento de `encender` debe tener gas, lo hice con `override`, al `frenar` gasta un 1% de gasolina, también con `override`, y al momento de `acelerar`, sobrescribí para que acelere más rápido que un carro normal, o sea, un 1 más rápido, y hice un `void` donde aparece el gas restante y otro para cuando cargo gas.

En la clase `Camión`, lo que hice fue primero modificar `acelerar` con `override` y dice que hice que aceleraran menos que un carro de combustión, específicamente un 0, o sea, que no acelera tanto. Luego de eso, sobrescribí `frenar`, y sé que frenará más lento que un carro normal, o sea, un 0, que gastará un 2% de gas, y en dado caso de velocidad que `acelerar` gastará un 5%. Y hice la lógica de cargar gasolina, hice la lógica de la gasolina restante, y modifiqué la información del vehículo, agregando la `cargaMaxima` y el `tipoGasolina`.

## Resumen general

En general, lo que hice fue primero modificar la clase `Vehiculo` para que tuviera las validaciones de encendido y apagado, y que `acelerar` y `frenar` dependieran de eso. Luego hice el `CarroDeCombustion` con sus 3 propiedades y sobrescribí los métodos para que gastara gasolina y tuviera validaciones. Después hice la `Motocicleta`, también con 3 propiedades, y modifiqué los métodos para que fuera más rápida y gastara menos gas. Y por último el `Camion`, que hice que fuera más lento, gastara más gas, y tuviera sus propiedades específicas. Todo lo hice usando **herencia y polimorfismo**, y encapsulé las propiedades para que fueran privadas. El proyecto funciona creando vehículos que heredan de `Vehiculo`, y cada uno tiene su comportamiento especial, como gastar gasolina o ser más rápido o lento.