

Documentación del Proceso

Punto 1: Derivé la clase **AutoDeCombustion** de la clase **Vehiculo**

Primero, derivé la clase **AutoDeCombustion** de la clase base **Vehiculo**. Esto me permitió reutilizar las propiedades y métodos comunes a todos los vehículos, como **Modelo**, **Color** y **Year**. Al hacer esto, no tuve que volver a escribir estas propiedades en cada clase derivada. En **AutoDeCombustion**, creé un método **Acelerar** que sobrescribe el comportamiento del método en la clase base. Además, gestioné el nivel de combustible, el cual disminuye cada vez que el vehículo acelera o frena.

Punto 2: Encapsulé la propiedad **private int velocidad = 0**; en la clase base y sobrescribí el método **Frenar** en la clase **Motocicleta**

Luego, decidí encapsular la propiedad **velocidad** dentro de la clase base **Vehiculo**. Esto se hizo para evitar que otras clases accedieran directamente a ella, asegurando que el control sobre la velocidad se mantuviera dentro de la clase base. En la clase **Motocicleta**, sobrescribí el método **Frenar** para agregar un comportamiento adicional. Cada vez que la motocicleta frena, también disminuye su nivel de aceite, lo cual es una característica particular de las motocicletas.

Punto 3: Implementé el polimorfismo con las clases derivadas y sobrescribí el método **Acelerar** en las clases **Camion** y **CarroElectrico**

Para implementar el polimorfismo, sobrescribí el método **Acelerar** en las clases derivadas como **Camion** y **CarroElectrico**. Cada una de estas clases tiene una implementación diferente para el método **Acelerar**, adaptada a las características específicas de cada tipo de vehículo. Por ejemplo:

- En **Camion**, debido a su peso, la aceleración es más lenta, por lo que la cantidad de velocidad sumada es menor.
- En **CarroElectrico**, la aceleración afecta el nivel de batería, lo que es característico de los vehículos eléctricos.

Punto 4: Agregué métodos adicionales en la clase base **Vehiculo** para ser sobrescritos

Para asegurar que cada tipo de vehículo tuviera su propio comportamiento, agregué algunos métodos adicionales en la clase base **Vehiculo**. Estos métodos son **Frenar**, **Acelerar** y **InformacionVehiculo**, que son comunes a todos los vehículos, pero luego son sobrescritos en las clases derivadas. De esta manera, pude personalizar el comportamiento de cada tipo de vehículo, manteniendo la estructura común en la clase base.

Punto 5: Creación de nuevas clases derivadas (Motocicleta, Camion, CarroElectrico)

Finalmente, creé las clases derivadas **Motocicleta**, **Camion** y **CarroElectrico**, cada una con sus propias propiedades y comportamientos adicionales. Por ejemplo:

- **Motocicleta:** Tiene un nivel de aceite que disminuye con cada aceleración o frenado.
- **Camion:** Posee una capacidad de carga que afecta su aceleración.
- **CarroElectrico:** Tiene un nivel de batería que disminuye al acelerar, y también puede ser recargado con el método **cargarBateria()**.

Resumen General

Lo que hice en este proyecto fue expandir el sistema original creando nuevas clases derivadas, como **AutoDeCombustion**, **Motocicleta**, **Camion** y **CarroElectrico**, y practicar conceptos como la **encapsulación** y el **polimorfismo**.

Al derivar estas nuevas clases de la clase base **Vehiculo**, pude aprovechar propiedades y métodos comunes, mientras que las clases derivadas podían sobrescribir los métodos para adaptarlos a sus características específicas.

Lo nuevo que implementé fue:

1. Creé nuevas clases derivadas (**AutoDeCombustion**, **Motocicleta**, **Camion**, y **CarroElectrico**).
2. Practiqué la **encapsulación** de propiedades como **velocidad** en la clase base para proteger la información.
3. Implementé **polimorfismo** sobrescribiendo métodos como **Acelerar** y **Frenar** para que se comporten de manera diferente según el tipo de vehículo.
4. Me aseguré de que cada clase derivada tuviera al menos 3 propiedades, como se me solicitó.