

## Documentación de Herencia y Polimorfismo

### Punto 1: Derivé la clase autoDeCombustible de la clase Vehiculo.

- Para resolver este punto, creé la clase autoDeCombustible que hereda de la clase base Vehiculo. Esto me permitió reutilizar las propiedades y métodos de la clase base, como Color, Modelo, Year, acelerar, frenar, apagar y encender.
- En el constructor de autoDeCombustible, inicialicé propiedades específicas como consumoPorKilometro, TipoDeCombustible y Kilometraje.

### Punto 2: Encapsulé la propiedad `private int velocidad = 0`; en la clase base y sobrescribí el método frenar en la clase motocicleta.

- Encapsulé la propiedad velocidad en la clase Vehiculo para que solo pueda ser modificada a través de métodos específicos como acelerar y frenar. Esto asegura que la velocidad no pueda ser manipulada directamente desde fuera de la clase.
- En la clase motocicleta, sobrescribí el método frenar para agregar un mensaje adicional que indique que la motocicleta está frenando de manera específica.

### Punto 3: Agregué propiedades específicas a cada clase derivada.

- En la clase CarroElectrico, agregué propiedades como capacidadBateria, tiempoCarga y un método para cargar la batería.
- En la clase autoDeCombustible, agregué propiedades como consumoPorKilometro, TipoDeCombustible y Kilometraje.
- En la clase motocicleta, agregué propiedades como velocidadMaxima, peso y CapacidadTanque.
- En la clase Camion, agregué propiedades como TipoDeCombustible, Kilometraje y TieneRemolque.

### Punto 4: Implementé el método InformacionVehiculo en la clase base.

- Este método muestra la información básica del vehículo (color, modelo y año). Lo reutilicé en todas las clases derivadas para evitar duplicar código.

### **Punto 5: Probé todas las clases en el archivo Program.cs.**

- Creé instancias de cada clase (CarroElectrico, autoDeCombustible, motocicleta, Camion) y probé sus métodos y propiedades.
- Separé cada sección con Console.WriteLine() para que la salida sea más legible.

### **Resumen**

En este proyecto, implementé una jerarquía de clases utilizando herencia y polimorfismo. La clase base Vehiculo contiene propiedades y métodos comunes que son reutilizados por las clases derivadas (CarroElectrico, autoDeCombustible, motocicleta, Camion). Cada clase derivada tiene propiedades y comportamientos específicos que las diferencian.

### **Cómo funciona el proyecto:**

1. La clase Vehiculo actúa como base, proporcionando métodos y propiedades comunes.
2. Las clases derivadas sobrescriben métodos como frenar y agregan propiedades específicas.
3. En el archivo Program.cs, se crean instancias de cada clase y se prueban sus funcionalidades.
4. La salida en consola muestra cómo cada tipo de vehículo realiza acciones específicas, como acelerar, frenar, o mostrar información.

Este enfoque permite reutilizar código, mantener una estructura clara y aprovechar las ventajas de la programación orientada a objetos.