



DOCUMENTACIÓN DEL PROCESO DE CREACION DE TAREA: HERENCIA Y POLIMORFISMO

Punto 1: Derivaciones:

- `AutoDeCombustible`, `Motocicleta` y `Camión` se heredan de `Vehículo` y en esta se añadieron propiedades únicas. Por ejemplo: en `AutoDeCombustion` sería la propiedad de `TipoCombustible`, en `Motocicleta` sería `TieneCasco`, y en `Camión` sería `Ejes`.

Punto 2: Creación de las clases que se derivaron.

- `AutoDeCombustible` está representa un auto que utiliza gasolina o diésel.
- `Motocicleta` está clase representa una moto con características específicas.
- `Camión` este representa un camión con capacidades de carga.

Punto 3: Encapsulados:

Utilice el modificador `private` para poder proteger o resguardar los datos de cada clase que se creó.

- `AutoDeCombustible` → `private int NivelCombustible;`
- `Motocicleta` → `private bool TieneCasco;`
- `Camión` → `private bool TieneRemolque;`

Y para poder tener acceso a estos atributos, agregue el método public de la siguiente manera → `public void CargarCombustible(int cantidad) {NivelCombustible += cantidad; }`

Para la sobreescritura de Métodos:

Añadí y sobrescribí los métodos `Acelerar ()`; y `Frenar ()`; en cada clase para que tengan procedimientos específicos; como:

- `AutoDeCombustible`:
 - `Acelerar (int cuanto)`; este es para poder aumentar la velocidad y el poder reducir el combustible.
 - `Frenar ()`; este disminuye la velocidad y gasta menos combustible en el proceso.
 - `CargarCombustible(int cantidad)`; este es para agregar combustible.

➤ **Motocicleta:**

- `Acelerar (int cuanto)`; este es para poder aumentar la velocidad de manera más rápida que un carro.
- `Frenar ()`; este reduce la velocidad más rápido que otros vehículos.
- `UsarCasco()`; este activa una simulación de seguridad, de si se colocó el casco o no.

➤ **Camión:**

- `Acelerar(int cuanto)`; para aumentar la velocidad de manera más lenta que un carro.
- `Frenar()`; este nos indica que el camión necesita más espacio para poder frenar.
- `EngancharRemolque()`; este simula la acción de conectar un remolque.

Resumen del Proyecto.

Este proyecto sirve para demostrar del como funcionan las propiedades de Herencia, Polimorfismo y Encapsulados en la programación orientada a objetos (POO) en C#. También muestra del cómo se utilizan las clases; Y este proyecto tiene un diseño con una estructura de modularización y es reutilizable y así nos permite el poder expandir la funcionalidad de la clase base `Vehículo` con nuevas clases y métodos que se agregaron en ella de forma ordenada. Y de igual manera la información relevante se imprime en consola para poder tener vista del cómo cada vehículo responde a las acciones proporcionadas en cada clase, tomando en cuenta que la clase base `vehículo` no se modifica al ejecutarse.

CAPTURA

```
Microsoft Visual Studio Debug Console
Color Amarillo:
Modelo BYD:
Año 2026
El nivel de batería es: 51
vas a 20 KMS / Hora
Auto frenando... Reducción de velocidad y consumo mínimo de combustible
Combustible cargado. Nivel actual: 108
vas a 30 KMS / Hora
Motocicleta frenando... Baja velocidad más rápido
Casco colocado. Seguridad incrementada.
vas a 5 KMS / Hora
Camión frenando... Necesita más distancia para detenerse
Remolque enganchado. Capacidad de carga aumentada.

C:\Users\USUARIO\Documents\PROYECTO herencia_polimorfismo_proyect\herencia_polimorfismo_proyect\herencia_polimorfismo_pr
oyect\bin\Debug\net8.0\herencia_polimorfismo_proyect.exe (process 7164) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```