**Reflexion:**

El desarrollo del sistema “SistemaVehiculos” permite comprender de manera práctica cómo los principios de la Programación Orientada a Objetos (POO) se complementan entre sí para construir software flexible, escalable y fácil de mantener. La implementación de clases abstractas, interfaces y herencia múltiple por interfaz demuestra que un diseño bien estructurado puede adaptarse a nuevos requerimientos sin modificar la base del sistema.

El polimorfismo destaca como el principio más poderoso en este caso, ya que posibilita tratar diferentes objetos (Automovil, Bicicleta, Motocicleta) como instancias de una misma superclase (Vehiculo), ejecutando comportamientos distintos según su naturaleza. Esta capacidad de vinculación dinámica facilita la extensión del sistema: por ejemplo, añadir una nueva clase como Camion o Scooter requeriría solo su implementación concreta, sin alterar el resto del código.

Asimismo, las interfaces Combustible y Mantenimiento ilustran la importancia del diseño por contratos, donde las clases declaran lo que pueden hacer sin exponer cómo lo hacen. Esto refuerza el principio de bajo acoplamiento, que mejora la modularidad y la reutilización.

En conjunto, el proyecto evidencia que la aplicación coherente de los pilares de la POO conduce a un software más ordenado, robusto y sostenible en el tiempo, capaz de crecer sin perder legibilidad ni estabilidad.