Examen Práctico: Aplicación de los Principios SOLID en un Sistema de Reportes de Vehículos

Objetivo:
Diseñar e implementar en Python un sistema orientado a objetos que genere reportes sobre distintos tipos de vehículos, aplicando correctamente los **principios SOLID**.

Enunciado:
Una empresa de logística desea un sistema para generar **reportes personalizados** sobre su flota de vehículos, que incluye **autos**, **camiones** y **motocicletas**. Cada tipo de vehículo tiene información distinta (peso máximo, consumo, tipo de carga, etc.), y cada uno debe poder mostrar sus datos en **formatos distintos de reporte**: por pantalla, PDF o archivo de texto (simulado).
El sistema debe ser **extensible**, permitir **agregar nuevos tipos de vehículos** y **nuevos tipos de reportes sin modificar las clases existentes**.

Requisitos del sistema:

1. **Registrar vehículos** con información específica según su tipo.
2. **Generar un reporte** del vehículo en uno de los siguientes formatos:
    - Consola
    - Archivo de texto (simulado)
    - PDF (simulado)
3. Deben poder agregarse nuevos formatos de reporte en el futuro sin alterar los vehículos.
4. Aplicar de manera explícita los cinco principios **SOLID**:
    - Responsabilidad única
    - Abierto/cerrado
    - Sustitución de Liskov
    - Segregación de interfaces
    - Inversión de dependencias

Tareas:

1. Crear clases base para Vehiculo y GeneradorReporte.
2. Implementar clases hijas: Auto, Camion, Moto.
3. Implementar generadores de reporte para consola, archivo de texto y PDF (simulado con print()).
4. Mostrar reportes de diferentes vehículos utilizando los distintos formatos.
5. Asegúrate de que el sistema no tenga dependencias fuertes entre clases.

Sugerencias:

- Usa abc.ABC y @abstractmethod para interfaces de reporte.
- Usa **composición** para delegar la generación de reportes.
- Cada clase debe tener **una sola responsabilidad** (por ejemplo, una clase no debe imprimir y manejar lógica de vehículo a la vez).
- Usa **inyección de dependencias** para asignar el generador de reporte a cada vehículo.

Evaluación (20 puntos):

Criterio	Puntos
Aplicación del principio de responsabilidad única	4 pts
Aplicación del principio abierto/cerrado	4 pts
Aplicación del principio de sustitución de Liskov	3 pts
Aplicación del principio de segregación de interfaces	3 pts
Aplicación del principio de inversión de dependencias	4 pts
Documentación y legibilidad del código	2 pts
