

## Tarea ED04\_1 – Sistema de Gestión para un Concesionario de Vehículos

### A. Preguntas de reflexión :

- I. ¿Por qué es útil definir la interfaz DescuentoAplicable para los clientes?**
  - Es útil porque representa un conjunto de métodos respecto a diferentes clases para un uso específico
- II. ¿Qué ventajas aporta el uso de clases abstractas (Vehiculo y Cliente) para modelar jerarquías?**
  - La ventaja principal de las clases abstractas es que evitas la reutilización de código, la estandarización, el polimorfismo.
- III. Justifica la relación empleada entre Venta y Vehiculo/Cliente (asociación, composición, agregación...).**
  - La relación es asociación y se justifica simplemente porque en el diagrama se entiende que la clase cliente se relaciona con vehiculo y venta porque se hace “una venta de un vehiculo a un cliente”.
- IV. ¿Cómo el uso de interfaces y herencia mejora la extensibilidad del sistema?**
  - Lo mejora para evitar repetición de código en clases y facilita la agregación de nuevos métodos sin tener que modificar clases antes creadas
- V. ¿Qué ventajas ofrece dividir la clase Concesionario en varios componentes especializados?**
  - Que se evita la sobre carga innecesaria SRP, que al evitar eso permite un diseño más limpio y más sencillo de entender
- VI. ¿Cómo garantizarías que el sistema sea extensible para incluir nuevos tipos de vehículos o clientes?**
  -
- VII. ¿Qué principios SOLID están mejor representados en este diseño y cómo?**
  - Principio SRP (Responsabilidad Unica)
  - Principio ISP (Segregación de Interfaces)
- VIII. ¿Es recomendable o deseable representar los constructores y getters/setters de las clases en el diagrama de clases? ¿Hay casos donde esa representación tenga más sentido? Justifica si, en tu caso, has decidido incluirlos en el diagrama o no.**
  - No es necesario usar los getters y los setters en este diagrama por lo sencillo que es, sería esencial en caso que el diseño se orientado a tener una ejecución más lógica