PRINCIPIOS SOLID

- ## SINGLE RESPONSIBILITY PRINCIPLE (SRP)

Este principio se cumple en el código de la práctica, pues, por ejemplo, la clase Vehicle contiene las funciones generales mínimas que comparten todos los vehículos y cada clase que hereda a Vehicle se encarga de su funcionalidad específica, teniendo cada clase solo sus atributos necesarios. Además, las funcionalidades de estas clases están divididas en labores atómicas y funciones más generales (como UseRadar) que combinan estas labores atómicas para hacer procesos más complejos.

- ## OPEN/CLOSE PRINCIPLE (OCP)

Se ha cumplido mediante hacer privadas todas las variables locales de cada clase y solo dando acceso a modo de getter a las propiedades que otras clases necesiten acceder. De esta forma, mediante la herencia de clases y mediante los getters se pueden amplificar y especificar la funcionalidad de las clases, pero los parámetros privados de cada instancia no son modificables.

- ## LISKOV SUBSTITUTION PRINCIPLE (LSP)

Se cumple ya que ninguna clase derivada modifica la clase base. Todas las clases derivadas de Vehicle especifican la funcionalidad del vehículo, heredando las propiedades base que comparten todos los vehículos, pero las funcionalidades comunes entre todos los tipos de vehículo se comparten. Además, se ha creado la clase intermedia PlatedVehicle para la gestión de vehículos sin matrícula sin necesidad de comprobar cada vez que si la matrícula es o no nula.

- ## INTERFACE SEGREGATION PRINCIPLE (ISP)

Se cumple, debido a que la única interfaz existente (IMessageWriter) solo consta de la función WriteMessage, la cual se usa en su totalidad cada vez que se invoca.

- ## DEPENDENCY INVERSION PRINCIPLE (DIP)

Se cumple debido a que gracias a la implementación de la clase intermedia PlatedVehicle se han eliminado las dependencias entre las clases.

PREGUNTA 7

Se incumpliría el Single Responsibility Principle. Para solucionarlo, se podría hacer una función general que compruebe qué tipo de medidor tenga y llame a su función correspondiente, y así hacer una función específica para cada medidor y poder controlar aparte la implementación de cada medidor.

DIAGRAMA UML