**PARADIGMAS: PRÁCTICA 2**

**DIAGRAMA UML:**

**Diagrama

Descripción generada automáticamente**

En el código actual, se están aplicando los principios SOLID, y el único principio que podría estar en riesgo es el Principio de Sustitución de Liskov. Esto se debe a que al intentar implementar un vehículo sin matrícula, como un "scooter", estábamos obligando a la clase Vehicle a incluir una matrícula, lo que no sería necesario en este caso. Para solucionar este problema, he creado una clase Plate que hereda de Vehicle, y he eliminado el atributo plate de la clase base. Las clases que lo necesitan, como PoliceCar y Taxi, pueden heredar esta nueva clase Plate.

Al querer permitir que el coche de policía pueda utilizar distintos dispositivos de medición, como un radar o un alcoholímetro, estamos incumpliendo el Principio de Abierto/Cerrado (Open/Closed Principle). Esto se debe a que cada vez que añadimos un nuevo tipo de medidor, tendríamos que modificar la clase PoliceCar. La solución sería crear una interfaz IMeasurementDevice que permita a las clases que implementan diferentes medidores (como Radar, Alcoholímetro, etc.) interactuar con PoliceCar de manera genérica. De esta forma, PoliceCar no necesitaría ser modificado cada vez que se agregue un nuevo medidor, y podríamos asignarle cualquier dispositivo de medición que implemente esta interfaz.

REPOSITORY: https://github.com/mariaascaniom/paradigmas\_practica\_2