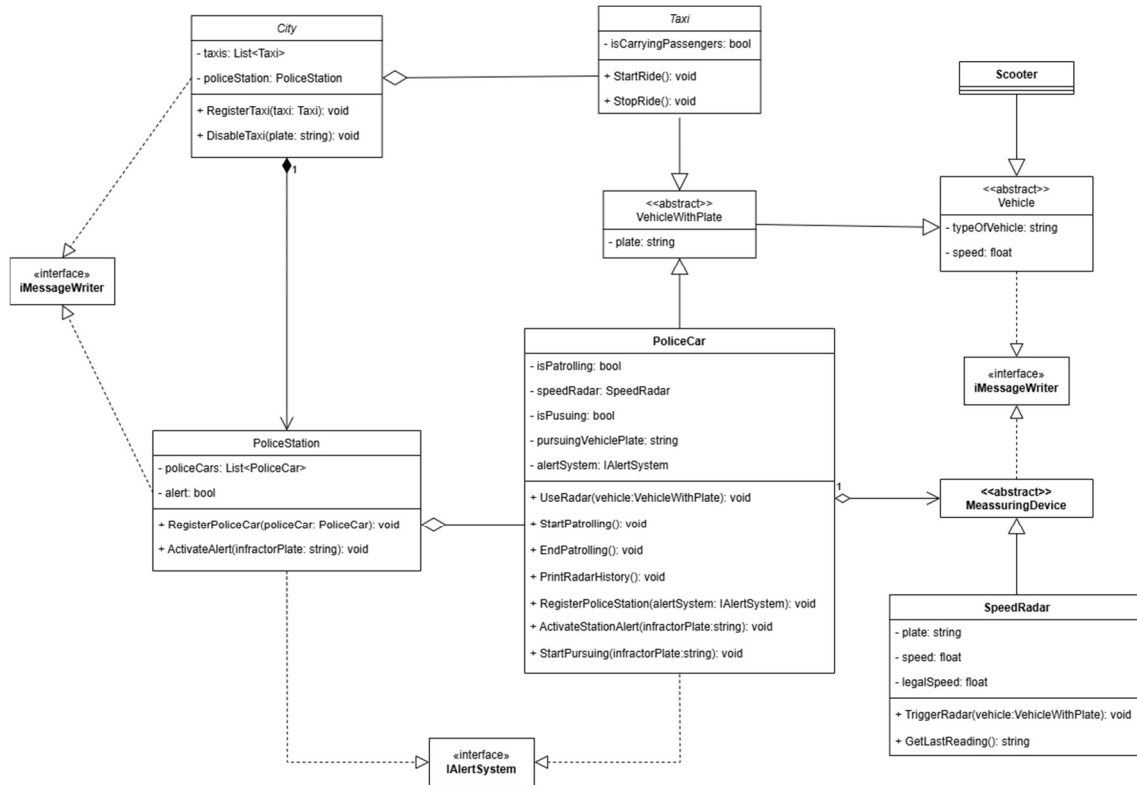


Practica 2 - Arquitectura software

Javier Ahumada

Git: <https://github.com/javierahumada4/ProgrammingParadigmsAndTechniques.git>

UML



Se puede ver que la arquitectura no cumple del todo los principios SOLID. En primer lugar, la ciudad rompe el Single Responsibility Principle puesto que se encarga tanto de los taxis como de la comisaría. El Open/Closed Principle parece cumplirse gracias a las clases abstractas de los vehículos y los dispositivos de medida, que permiten modificar las clases heredadas sin modificar la clase base. También se cumple el Liskov Substitution Principle gracias a las clases abstractas. El Interface Segregation Principle se cumple gracias a IMessageWriter, que permite evitar crear métodos innecesarios, podría haber mejora si utilizáramos una interfaz para la matrícula de los vehículos en lugar de la clase VehicleWithPlate. El Dependency Inversion Principle se cumple en general.

En el UML se incluye la clase abstracta MeasuringDevice, de la que hereda el velocímetro y podría heredar el alcoholímetro sin violar SOLID. En el código violamos el SOLID puesto que el coche de policía y el velocímetro dependen el uno del otro.