**6. Por qué se cumplen los principios SOLID:**

S: las clases y métodos solo realizan una cosa

O: ideas abstractas para tener código reutilizable, abiertas para extensión, pero no para modificarlas

L: Una clase derivada no modifica el comportamiento de la base

I: una clase con interfaz no implementa métodos que no usa

D: clases de alto nivel no dependen de las de bajo nivel.

**7. ¿Qué principio SOLID se incumple?**

Estaríamos violando el segundo principio: “Open-Closed”. Porque la clase policía no está preparada para extenderse, tendríamos que modificarla añadiendo un nuevo elemento que sea un medidor de alcohol.

Lo que podemos proponer es una clase abstracta que sea medidor ya que el coche de policía solo puede tener un único aparato de medida. Y esto se puede definir como una interfaz que solo hereden los coches policía (no nos interesa que lo hereden todos los coches). Aunque heredarían dos interfaces. Tal vez podríamos hacer una interfaz solo para esos aparatos de medida y que lo hereden específicamente las clases del aparato de medida. La interfaz de los aparatos puede heredar la interfaz del mensaje para no tener que hacer que una clase herede dos interfaces a la vez.