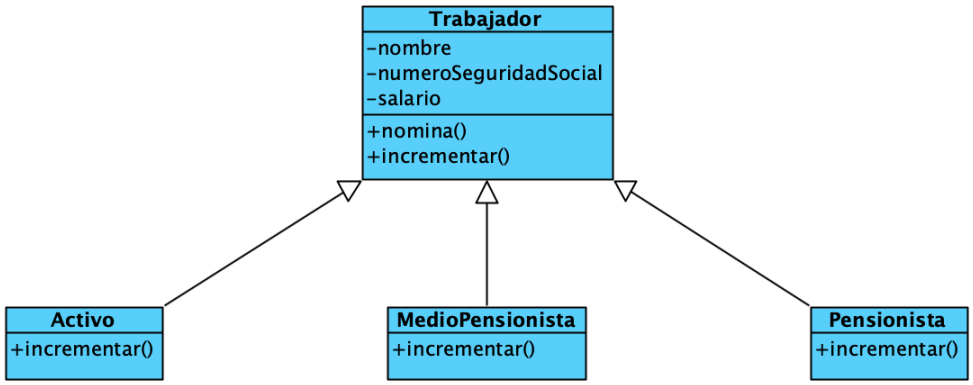
2.
a)Justificar por qué las clases descritas no pueden ser implementadas directamente en Java.

El atributo salario al que quiere acceder la clase MedioPensionista es un atributo que pertenece a la clase Activo y Pensionista.
En nuestro diagrama, vemos que la clase MedioPensionista hereda de Activo y Pensionista, accediendo de esta manera a ambos atributos salario. El problema es que en java existe una restricción de herencia única, por lo que la clase MedioPensionista solo puede heredar de una de las dos clases y por lo tanto, acceder solo a uno de los dos atributos salario.

b)Discutir y desarrollar una solución que permita resolver la situación descrita. Como es lógico, la solución propuesta debe mantener la funcionalidad actualmente existente en las tres subclases de Trabajadores y asegurar la consistencia de los atributos y la reutilización de los métodos de las clases Trabajador, Activo y Pensionista.

Vemos en el diagrama que en el caso del trabajador activo, su nuevo salario se calcula a partir de la fórmula salario@pre*1.02, mientras que en el caso del pensionista, el nuevo salario se calcula como salario@pre*1.04. Esto quiere decir que el salario nuevo de un medio pensionista es, como dice en el enunciado
la suma del salario como activo y pensionista, es decir, salario@pre*1.02 + salario@pre*1.04. Esta nueva fórmula nos permitiría que MedioPensionista herede directamente
de la clase Trabajador, evitando la doble herencia.
Resumiendo, los cambios serían los siguientes:



Teniendo el método incrementar de MedioPensionista como post condición que salario = salario@pre*1.02 + salario@pre*1.04