

EJERCICIO 1

Principios:

Utilizaremos el principio SOLID de **sustitución de Liskov** en el cual al redefinir un método en clases como “Trabajador” y “Equipo” puede cambiarse su uso siempre y cuando el tipo de dato sea el correcto para esta implementación . Además tiene uso el principio de **Inversión de dependencia** ya que las clases citadas anteriormente dependen de una clase abstracta “ElementoProyecto”. Por ultimo utilizamos el principio de **abierto/cerrado** pues el método getCotrabajadores tiene una implementación por defecto que hereda las clases que se crean a partir de ella, pudiendo las nuevas clases redefinir dicho método .

Patrones:

Como patrón de comportamiento elegimos el patrón de comportamiento de **composición** para poder crear objetos mas simples gracias a otros más complejos creados previamente . Este es el caso para crear equipos que proceden de proyectos los cuales se componen de nuestros diferentes trabajadores.

DIAGRAMA DE CLASES Y SECUENCIAL RESPECTIVAMENTE



