

# Práctica 01

Noe E. Amador González — 419004815

31 de Julio, 2020

1. Menciona los principios de diseño esenciales de los patrones Observer y Strategy.
  - Favorecer la composición sobre la herencia.
  - Programar a una interfaz (o clases abstractas), no a una implementación.
  - "Programar a una interfaz" realmente significa "Programar a un supertipo"
  - Identificar los aspectos de que varían encapsulándolos y separarlos de los que no cambian.
  - Esforzarse por obtener diseños débilmente acoplados entre objetos que interactúen.
2. Menciona una desventaja de cada patrón.
  - **Strategy**
    - Una desventaja que yo veo es que al tener tantos comportamientos (estrategias) podemos llegar a confundirlos si no definimos diferencias claras entre todas las estrategias.
    - Podríamos implementar tantas estrategias/comportamientos que se podrían convertir en un problema cuando el programa escale.
  - **Observer**
    - Si no se implementa bien una desventaja es que si el sujeto tiene muchos observadores entonces a la hora de actualizar estados del sujeto este puede llegar a ocupar una complejidad enorme.

## README

- La implementación de la práctica se ha llevado a cabo con la versión 8 de Java.
- Para correr la simulación hay que compilar todos los archivos \*.java y después utilizar la clase *Main* como la clase que ejecutará la simulación.

- Es importante destacar que el diagrama del patron *Strategy* y *Observer* los he juntado en un solo diagrama. En este diagrama no especifiqué metodos constructores por cuestiones de estetica y tiempo.
- Soy consciente que no tengo las mejores practicas de lenguaje (incluyendo documentacion); en mi defensa puedo decir que no tengo experiencia en Java (estudio Matematicas Aplicadas). Sin embargo tengo como meta mejorar estas practicas durante el curso y me comprometo a ir mejorando mis scripts en las siguientes tareas/practicas/proyectos.