

# Práctica 1

Bartolucci Nicolás Marcelo - 317520527  
Decuir Fuentes Rodrigo André - 320022711  
Demetrio Torres Yanahi - [n.cuenta](#)

**Equipo:** Tukú Wana Min Nakú

1. Menciona los principios de diseño esenciales del patrón Strategy y Observer.

**Observer** es un patrón de diseño de comportamiento que te permite definir un mecanismo de suscripción para notificar a varios objetos sobre cualquier evento que le suceda al objeto que están observando.

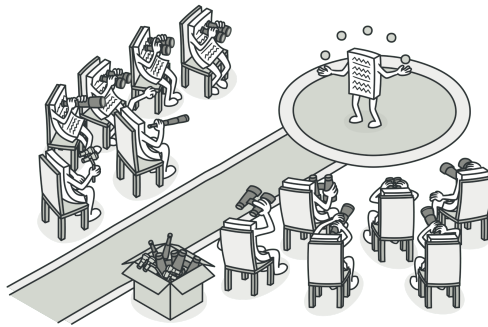


Figura 1: Observer

**Strategy** es un patrón de diseño de comportamiento que te permite definir una familia de algoritmos, colocar cada uno de ellos en una clase separada y hacer sus objetos intercambiables.



Figura 2: Strategy

2. Menciona una desventaja de cada patrón.

## Observer

- ☞ Los suscriptores son notificados en un orden aleatorio.
- ☞ Se podrían producir actualizaciones en cascada muy ineficientes.
- ☞ Produce un aumento de complejidad en el código, y aumento del número de clases necesarias para el código.

## = Strategy

- ☞ Menor eficiencia. Aumenta el número de objetos creados.
- ☞ Los clientes deben tener un cierto conocimiento de cada estrategia, para así poder elegir en cada situación cuál es la más apropiada.
- ☞ Dado que todas las estrategias comparten una interfaz común, si las diferencias entre ellas es grande, es probable que mucha de la información que se les pasa no sea de utilidad más que a las más complejas.

### 3. Anotaciones acerca de la práctica.

...

### 4. Instrucciones de ejecución.

Entra a la carpeta “ src ” y compila el código con:

```
javac *.java
```

Desde la misma carpeta ejecuta el código con el siguiente comando:

```
java SuperSmashBrus
```