**COMMAND PATTERN**

**LOS PRINCIPIOS DE DISEÑO UTILIZADOS EN ESTE PATRÓN SON:**

* Las clases deben estar abiertas a su extensión y cerradas a su modificación
* Responsabilidad

**Los posibles casos de uso de este patrón serian.**

* Cuando en un sistema multi hilos puede existir una cola de comandos que va cambiando en tiempo de ejecución, cada hilo podría tomar un comando y ejecutarlo sin conocer detalles, al finalizar elimina el comando de la lista y toma el siguiente elemento.
* Dentro del logger de un sistema, de esa forma una clase solo se preocupa por llamar al comando Create log, pero no de la forma en la que se implementa.
* Dentro de un sistema que se encarga de ejecutar una pila de comandos y en el proceso el sistema falló, lo que provoca que la ejecución se detenga, usando una pila de comandos podríamos restaurar el sistema y reiniciar la ejecución de los comandos en el mismo orden y sin manejar procesos complicados para mantener la información de la pila.

**BENEFICIOS:**

* Desacopla el objeto que invoca la operación del que sabe cómo realizarlo.
* Este patrón ayuda en términos de extensibilidad, ya que podemos agregar nuevos comandos sin cambiar el código existente.
* Capacidad para hacer, des hacer y rehacer comandos de forma sencilla.
* Se puede encapsular y ejecutar una serie de comandos de forma sencilla.

**DESVENTAJAS:**

* El código puede volverse más complicado ya que está introduciendo una capa completamente nueva entre remitentes y receptores.