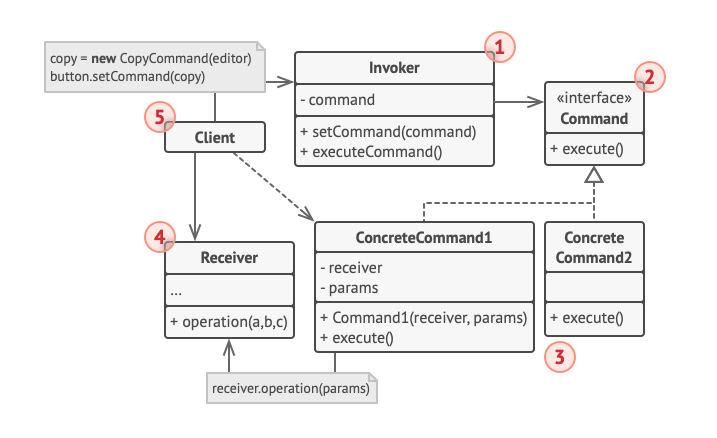
COMMAND



Comportamiento (Behavioral)

**Descripción:**  
El patrón Command encapsula una solicitud como un objeto, permitiendo parametrizar clientes con diferentes solicitudes, almacenar solicitudes en cola o registrar su historial, y soportar operaciones como deshacer y rehacer. Separa al emisor de la solicitud del objeto que la ejecuta.

**Características principales:**

* Encapsula la acción solicitada como un objeto independiente.
* Permite ejecutar, deshacer y rehacer operaciones.
* Separa el objeto que invoca una operación del que la implementa.
* Facilita la implementación de sistemas basados en acciones o tareas.

**Componentes:**

1. **Command:** Interfaz o clase abstracta que declara un método para ejecutar una operación.
2. **ConcreteCommand:** Implementaciones concretas que definen una acción específica.
3. **Receiver:** Objeto que conoce cómo realizar la operación solicitada.
4. **Invoker:** Objeto que solicita la ejecución del comando.
5. **Client:** Configura los objetos comando y asigna el receptor adecuado.

**Ventajas:**

* Fomenta el desacoplamiento entre emisor y receptor de la acción.
* Facilita el almacenamiento y procesamiento de comandos en secuencia.
* Permite implementar colas de tareas, historiales, macros y operaciones deshacer.

**Desventajas:**

* Puede aumentar el número de clases debido a la necesidad de crear una clase por cada comando.
* Puede ser complejo si no se administra bien la creación y almacenamiento de comandos.

**Aplicaciones comunes:**

* Sistemas de menú o botones configurables (por ejemplo, interfaces gráficas).
* Control remoto o interfaz de dispositivos electrónicos.
* Sistemas con operaciones programadas o ejecutadas por lotes.
* Implementación de funcionalidades deshacer/rehacer en editores o software interactivo.