Command

wagnerfusca@gmail.com

- TV – on, off
- SAT – on, off
- DVD – on, off
- AUD – on, off

Command

- Encapsula uma solicitação como um objeto, o que lhe permite parametrizar outros objetos com diferente solicitações, enfileirar ou registrar solicitações e implementar recursos de cancelamento de operações.

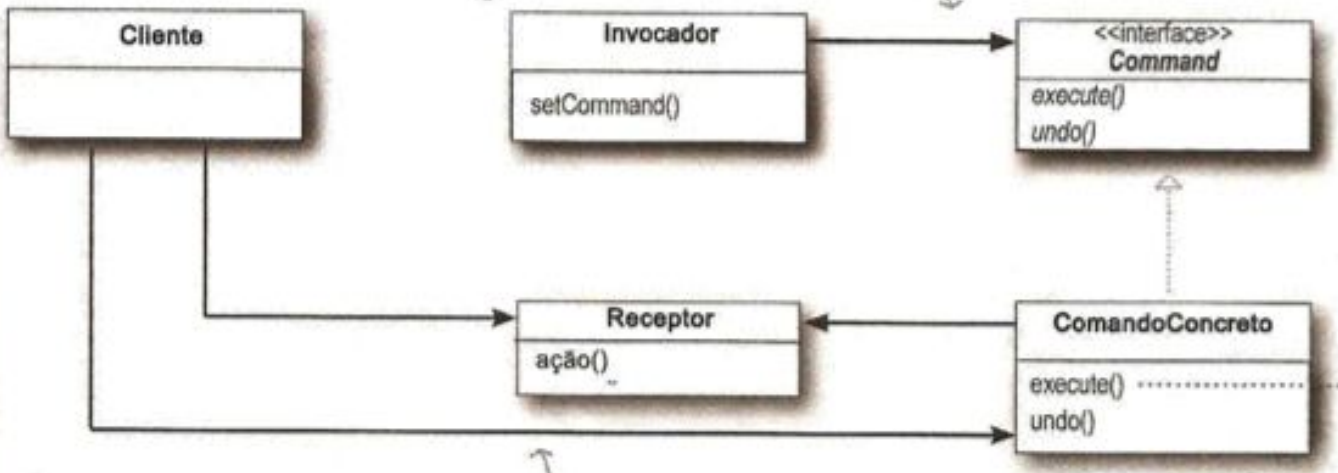
Command

- Parametrizar objetos por uma ação a ser executada. Os Commands são uma substituição orientada a objetos para callbacks;
- Especificar, enfileirar e executar solicitações em tempos diferentes.
- Suportar desfazer operações. A operação Execute, de Command, pode armazenar estados para reverter seus efeitos no próprio comando.

O cliente é responsável pela criação de um ComandoConcreto e pela definição do seu Receptor.

O Invocador contém um comando e, em algum momento, pede ao comando para atender uma solicitação chamando o seu método execute().

Command declara uma interface para todos os comandos. Como vimos anteriormente, um comando é invocado através do seu método execute(), que pede a um receptor para executar uma ação. Como você pode perceber, essa interface também possui um método undo(), que será discutido mais adiante neste capítulo.



O método execute() invoca uma ou mais ações no receptor que são necessárias para atender a solicitação.

```
public void execute() {
    receiver.action()
}
```

O Receptor sabe como executar as tarefas necessárias para atender a solicitação. Qualquer classe pode atuar como um Receptor.

O ComandoConcreto define um vínculo entre uma ação e um Receptor. O Invocador faz uma solicitação chamando execute() e o ComandoConcreto executa essa solicitação chamando uma ou mais ações no Receptor.

SQL???