12

Para saber mais: Design Pattern Command

Já falamos muito sobre o MVC, que é um padrão arquitetural. Ou seja, um padrão que define uma parte da arquitetura da aplicação (no nosso caso, define as camadas). O interessante é que também usamos um *Design Pattern* para construir o MVC. Qual?

Vamos lá! Repare que cada ação possui somente um método, que se chama `executa` (*execute*). Veja o exemplo:

```
public class NovaEmpresa implements Acao {

    public String executa(HttpServletRequest request, HttpServl
        throws ServletException, IOException {
    }
}
```

COPIAR CÓDIGO

Também poderíamos ter chamado esse método de `rode` (*run*) ou *call*, não faria diferença, mas sempre o objetivo desse método é encapsular a execução da ação.

A nossa interface `Acao` padronizou a assinatura do método para todas as ações seguirem o mesmo comportamento.

Repare também que, só pelo nome do método, não sabemos o que está sendo executado. Para tal, precisamos olhar o nome da classe. Isso também não importa para o controlador, o que importa é ter o método `executa`. Seguindo esse padrão, conseguimos delegar as chamadas do controlador central para as ações. Ótimo, mas qual é o padrão no final?

Todas as nossas classes seguem o padrão **Command**, ou seja, são comandos seguindo a nomenclatura do famoso livro sobre *Design Patterns*:

- ***Design Patterns: Elements of Reusable Object-Oriented Software***

Os padrões de projeto são um tópico importante no desenvolvimento de software, pois permitem resolver um problema de maneira estruturada e organizada. Sabendo dessa importância existem [vários cursos (https://cursos.alura.com.br/career/expert-orientacao-a-objetos)](https://cursos.alura.com.br/career/expert-orientacao-a-objetos) na Alura sobre padrões de projeto.