

1. O Spring MVC

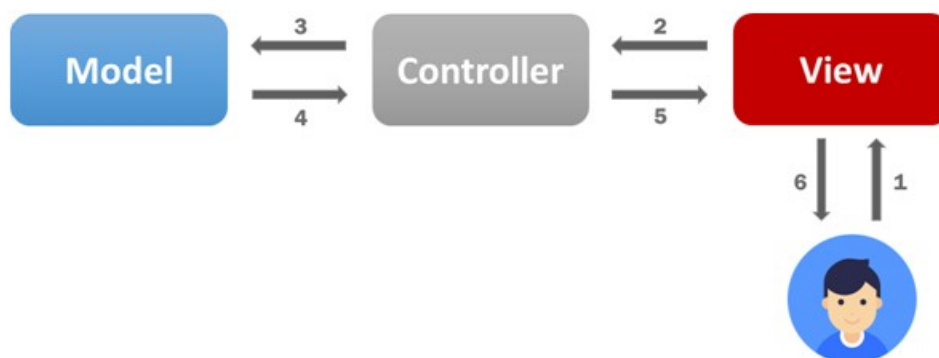
O Spring MVC é um dos frameworks para desenvolvimento Web mais utilizados hoje em dia. Com ele, temos à nossa disposição uma implementação do padrão MVC em conjunto com os principais recursos do Spring.

1.1 O que é MVC?

MVC é um padrão criado com o objetivo de organizar a arquitetura da nossa aplicação, a dividindo, basicamente, em três camadas com responsabilidades bem definidas. Por esse motivo dizemos que o MVC é um padrão arquitetural. Todo sistema desenvolvido sobre esse padrão terá, basicamente, as seguintes camadas:

- **Model:** Local onde devem ser criadas as classes que representam o domínio da aplicação, implementada a persistência de dados, a validação e as regras de negócio;
- **View:** Local onde devemos implementar a interface da nossa aplicação, para que os usuários possam acessar as funcionalidades e consumir seus dados, como uma interface gráfica construída com tecnologias como JSF, HTML, CSS, JavaScript;
- **Controller:** Camada que tem como responsabilidade integrar e intermediar a comunicação entre as camadas Model e View.

A partir disso, quando o usuário clica em um link em uma aplicação web, a View interpretará essa ação e solicitará ao Controller algo que atenda àquele pedido. O Controller, por sua vez, tratará essa requisição e fará a solicitação ao Model. No Model, a lógica é executada, o banco de dados, por exemplo, é consultado, e os dados são repassados ao Controller, que, por fim, envia esses dados para a View, que os exibirá conforme os elementos visuais que ela possui .



1.2 Como o Spring MVC funciona?

Ao receber uma requisição (1), o Front Controller, implementado pelo Spring MVC, e que também é chamado de Dispatcher Servlet, irá identificar o controller mais adequado para tratá-la. Então, essa solicitação é passada para ele junto com os dados que possam vir com a requisição (2).

Logo após, o controller identificará o método responsável por atender esse tipo de requisição. Nesse método, o model será chamado (3) para realizar o processamento, por exemplo, obter a lista de carros.

Com essa lista disponível (4), o controller, montará a resposta, um objeto, a ser enviado para o Front Controller (5). Esse objeto é formado, basicamente, por esses dados e mais uma string, que define a página que deve ser exibida.

O Front Controller, então, vai enviar esses dados para a View (6), que identificará a página adequada, incluirá nessa página os dados obtidos pelo model e irá gerar o HTML com esses dados. Feito isso, o View envia essa página ao Front Controller (7) que, por fim, a envia para o browser (8).

