Carlos Henrique Brasileiro
Eduardo Epiphânrio Moreira
Felipe Linhares Afonso
Juliana Carvalho de Souza
Lucca Bessa

Frameworks MVC com foco no Spring MVC

Belo Horizonte

Julho de 2018

Carlos Henrique Brasileiro
Eduardo Epiphânrio Moreira
Felipe Linhares Afonso
Juliana Carvalho de Souza
Lucca Bessa

Frameworks MVC com foco no Spring MVC

Pesquisa acadêmica apresentada ao curso técnico de Informática do Centro Federal de Educação Tecnológica de Minas Gerais como atividade avaliativa da disciplina Linguagem de Programação II.

Professor: Prof. Cristiano Amaral Maffort

Belo Horizonte
Julho de 2018

Resumo

Este trabalho visa tratar acerca de frameworks do padrão arquitetural MVC, dando ênfase especialmente no Spring MVC. Traz-se nessa pesquisa acadêmica uma explicação sobre a organização de aplicações que se valem de tais ferramentas, bem como sobre o funcionamento de seus respectivos fluxos de dados. Além disso, expõe-se sua utilidade em alguns projetos específicos, tais como os de desenvolvimento WEB, a fim de demonstrar e justificar seu amplo uso atualmente.

Palavras-chave: Frameworks MVC. Spring MVC. MVC. WEB. Spring.

Abstract

This paper aims to address frameworks following the architectural pattern MVC with focus on the Spring MVC framework. This academic research brings an explanation about the organization of applications that use such tools as well as the operation of their respective data flows. In addition, its usefulness is exposed in some specific projects like WEB development, in order to demonstrate and to justify its ample use at the moment.

Keywords: Frameworks MVC. Spring MVC. MVC. WEB. Spring.

Sumário

1 Introdução

Por um grande período o desenvolvimento de software passou por grande instabilidade, de modo que diversos aplicativos eram entregues fora do prazo ou simplesmente não cumpriam corretamente o que fora requisitado pelo cliente. Com isto, foram necessárias diversas reformas na ideia, no modo de se programar, em formas de cumprir com o prazo e seguir padrões de qualidade.

Foi neste contexto que que os padrões de projeto como o MVC tornaram-se uma solução elegante para diversos desses problemas, pois tornava possível a reutilização de grandes partes do código graças ao modo como é divido em camadas. Assim, possibilita uma melhor análise e entendimento de cada funcionalidade da aplicação, uma vez que cada camada cumpre com uma responsabilidade, além de trazer outras vantagens. Juntamente a isso, a utilização de frameworks foi uma forma de evitar a repetição de código, criando assim soluções remodeláveis para aplicações que cumprissem funções em comum.

O Spring MVC acabou sendo um framework que veio a substituir o Struts 1 que foi uma das principais soluções em framework MVC para Java que visava a reutilização do controlador, pois era um grande problema ter que reimplementar o padrão MVC a cada projeto. Além disso, o Spring apresentava as mais novas funcionalidades e recursos da linguagem Java, deste modo, se apresentando de forma mais moderna e proporcionando a tecnologia do container Spring.

O conteúdo do presente estudo está organizado em três seções, onde na primeira se apresenta o padrão arquitetural MVC, na segunda trata-se sobre frameworks de desenvolvimento que implementam este padrão e na terceira foca-se no uso do framework Spring MVC.

2 MVC

2.1 O padrão MVC

O padrão arquitetural MVC, Model-View-Controller - em tradução para o português brasileiro Modelo-Visão-Controlador ou ainda Modelo-Apresentação-Controlador - criado em 1970, é um modelo de estruturação de aplicações baseado em três camadas e nos relacionamentos entre elas. Este padrão surgiu como forma de findar a forte dependência entre os componentes de um sistema que dificultava não somente a manutenção do código fonte como também sua reusabilidade.

O modelo MVC consiste em separar a lógica de negócios da camada de apresentação. Nesse sentido, a camada de visão ou apresentação é responsável por exibir ao usuário a interface pela qual irá interagir com o sistema, permitindo que este entre com dados a serem processados e também veja os resultados das consultas realizadas no sistema. Já a camada de modelo é responsável por realizar toda a lógica de negócios, como cálculos e consultas a banco de dados. Por fim, a camada de controle realiza o intermédio entre essas duas camadas, sendo acionada por eventos disparados pela visão e buscando a resposta desejada por meio de invocação à camada de modelo que processa as informações dadas e retorna o resultado correspondente.

O modelo arquitetural vem sendo bastante implementado em vários tipos de aplicações tais como desktop, mobile e web, devido às suas vantagens. Especialmente no contexto de aplicações web, o padrão pode sofrer algumas modificações em função do comportamento requisição/resposta, tais como tornar inviável atualizações automáticas na camada de visão.

Para auxiliar na implementação do MVC, existem vários frameworks de desenvolvimento para cada uma das três camadas citadas anteriormente, que disponibilizam componentes para serem aproveitados, permitindo que o desenvolvedor realize um menor trabalho na infraestrutura do sistema.

3 Frameworks MVC

3.1 O que é um Framework?

Um framework,ou arcabouço em português brasileiro, é um conjunto abstrato de códigos genéricos, geralmente classes, que visa proporcionar ao desenvolvedor funcionalidades de infraestrutura da aplicação que se tornam repetitivas ao decorrer do processo de desenvolvimento e que são comuns a um grande número de projetos, funcionando como uma solução geral - devendo por isso ser bem documentado.

Um framework pode ser constituído por vários módulos, podendo ao programador optar por usar somente os de seu interesse, funcionando assim como uma espécie de esqueleto do sistema. Adicionalmente, um framework pode permitir a edição de algumas partes de seu código fonte ou ainda pode ser imutável em outras.

A principal diferença entre um framework e uma biblioteca é o fato de que o primeiro conta com thread de controle própria que chamará os métodos da aplicação, sendo assim uma ferramenta ativa, enquanto o segundo é uma ferramenta passiva uma vez que os métodos dos componentes são chamados por meio de threads da aplicação. Assim, o framework possibilita a inversão de controle que é basicamente a chamada do código da aplicação pelo framework e não o contrário, como costuma ser com bibliotecas.

3.2 Frameworks MVC

Os frameworks MVC surgiram após a constatação de que a implantação do modelo MVC trazia certo trabalho repetitivo em diversos projetos.

Para citar um exemplo da origem dessa ferramenta, tomemos a história das aplicações Java web. Como muitas empresas adotavam o padrão MVC, surgiu a necessidade de criar frameworks que possibilitavam a reutilização de código entre projetos das mesmas, levando assim a criação de muitas versões simples de frameworks MVC por essas empresas de forma a tentar solucionar os problemas da reutilização. Contudo, percebeu-se que o retrabalho era bastante grande entre projetos, levando assim a criação, no ano 2000, de uma ferramenta mais poderosa do que os primeiros frameworks, chamada Struts MVC, que disponibilizava funcionalidades para a camada de controle.

Com o passar do tempo, a linguagem Java foi ganhando novas facilidades enquanto o Struts permaneceu com a versão mais antiga de Java, tornando-se assim obsoleto por não oferecê-las. Outros frameworks surgiram então para substituir/competir com o Struts. Um exemplo é o Struts 2, criado também pela comunidade Struts, mas que é incompatível

com o Struts original. Outro exemplo é o Spring que vem se tornando muito famoso por ser um container leve que serve a aplicação com recursos tais como gerenciamento de objetos. Um de seus componentes é o Spring MVC, uma ferramenta moderna e poderosa que se usa do container mencionado.

Ainda no contexto de uma aplicação Java web, podemos citar alguns outros frameworks das camadas de visão, controle e modelo.

Visão: JSP, Velocity, Freemarker Sitemesh

Controle: Struts Action, VRaptor, JSF, Spring MVC

Modelo: Hibernate

4 Spring MVC

4.1 Framework Spring

O framework Spring, criado por Rod Johnson, começou a ser desenvolvido em 2002, com o propósito de simplificar a programação java, não tendo foco para o suporte ao desenvolvimento web. Entretanto, com o passar do tempo o Spring passou por diversas mudanças que o tornou uma estrutura Java perfeita para aplicações web. Atualmente, a última versão deste sistema é a versão 5, lançada em 2017.

O Framework é composto ao todo por sete módulos - a saber: Spring Core, Spring DAO, Spring Context, Spring Web MVC, Spring Web, Spring ORM e Spring AOP - que oferecem vários serviços empregados mais frequentemente nas aplicações, como o acesso à banco de dados. O Spring permite ao desenvolvedor escolher os que melhor se adéquem as necessidades da sua aplicação, tornando-se uma das principais vantagens desse recurso, visto que não existe a necessidade de se arrastar todas as ferramentas do framework para criar uma aplicação simples. A ferramenta foi construída baseando-se em conceitos de programação orientada a objetos, empregando-se, dentre outros, os objetos simples POJO (Plain Old Java Object), que tornam a aplicação eficiente para a realização de serviços empresariais complexos.

Os módulos do framework, também conhecidos como containers, se encarregam das classes da aplicação java e de definir suas dependências por meio de um arquivo XML. Dessa maneira, cria-se um baixo acoplamento entre as classes, sendo esta solução conhecida como injeção de dependência, o que se associa a ideia de inversão de controle(IoC) onde a sequência de chamadas de métodos não é determinada pelo programador.

Segundo José Ronaldo Júnior (JúNIOR, 2007), padrões de projeto constituem o cerne do projeto do Framework Spring, buscando a resolução de problemas com soluções existentes no mercado, facilitando e otimizando a criação de componentes com o objetivo de servir de base para reutilizações futuras.

4.2 Spring MVC

O Spring Framework é um framework que usa o padrão de arquitetura MVC e fornece um modelo abrangente de programação e configuração para aplicativos corporativos modernos baseados em Java - em qualquer tipo de plataforma de implementação, como aplicações web, permitindo a construção de aplicativos robustos e flexíveis.

Um elemento-chave do Spring é o suporte de infra-estrutura no nível do aplicativo:

o Spring concentra-se na junção de aplicativos corporativos para que as equipes possam se concentrar na lógica de negócios no nível do aplicativo, sem vínculos desnecessários com ambientes de implementação específicos.

4.2.1 Fluxo de uma Requisição Spring MVC

O fluxo de uma requisição dentro do funcionamento do framework pode ser descrito como:

I. O acesso a uma URL no browser envia uma requisicao HTTP para o servidor que roda a aplicacao web com Spring MVC. O Controlador fica responsável pela recepção da requisição;

II. O controlador procura a classe responsavel por tratar essa requisicao, entregando a ela os dados enviados pelo browser. Essa classe faz o papel do controller;

III. O controller passa os dados para o model, que executa todas as regras de negocio, como calculos, validacoes e acesso ao banco de dados;

IV. O resultado das operacoes realizadas pelo model e retornado ao controller;

V. O controller retorna o nome da view, junto com os dados que ela precisa para renderizar a pagina;

VI. O Framework encontra a view que processa os dados, transformando o resultado em um HTML.

VII. O HTML e retornado ao browser do usuario.

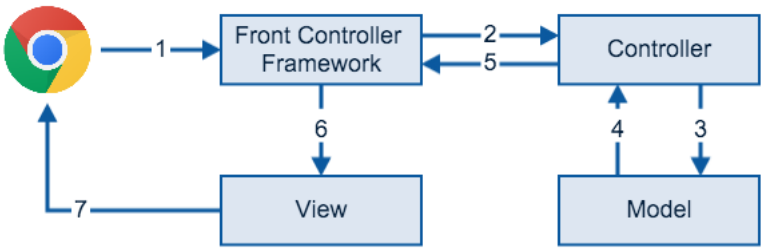

Figura 1: Fluxo de uma requisição no Spring MVC

4.2.2 Por quê usar Spring MVC?

Nos dia atuais, aplicações Web estão amplamente presentes e o surgimento do Spring MVC veio para facilitar o trabalho do programador web, ao fazer com que o desenvolvedor não se preocupe com detalhes da implementação e que ao invés disso possa focar somente na lógica de negócios.

Utilizando o modelo de programação MVC, um projeto se torna mais manutenível, uma vez que os itens são agrupados conforme as semelhanças entre as funções desempenhadas, o que permite uma maior compreensão acerca do funcionamento da aplicação.

Além disso, fazendo o uso de frameworks MVC, o desenvolvimento torna-se padronizado, já que o programador deve seguir o conjunto de classes e funções do Spring MVC, o que garante a segurança da aplicação. Adicionalmente, o Spring MVC é uma ferramenta muito valorizada no mercado, sendo testada e mantida por uma empresa experiente, a Pivotal Software.

Entre algumas outras vantagens do Spring MVC podemos citar uma integração simples com outros frameworks, bem como a navegação por meio de anotações que possibilita uma maior clareza acerca da responsabilidade de um trecho do código e da sua forma de se relacionar com os demais.

Entretanto, o framework Spring MVC pode trazer algumas desvantagens, cabendo ao desenvolvedor ciente optar ou não por utilizá-lo. Entre as dificuldades trazidas pela ferramenta estão o uso e abuso de arquivos XML's, e a dificuldade em tratar erros, como o 404.

5 Conclusões

Por meio de leitura de materiais que tratam sobre o tema do presente estudo, buscou-se neste trabalho introduzir o conceito de padrão arquitetural MVC bem como o significado de framework antes de apresentar o tema principal - Frameworks MVC com foco no framework Spring MVC - que tal pesquisa busca tratar, para que se torne didático a medida do possível.

Adicionalmente, neste trabalho buscou-se mostrar em específico a utilização dos frameworks em aplicações Java web, projetos hoje em dia tão presentes no mercado devido às facilidades que a linguagem Java possibilita, passando inclusive pela história dos frameworks MVC para tal tipo de aplicação e focando-se no Spring MVC como ferramenta auxiliadora na implementação do padrão. Ainda, foi dada uma breve introdução no framework Spring, um container leve, mas não por isso menos poderoso, que visa servir a aplicação web com seus diversos componentes.

Diante do que foi apresentado, pode-se constatar o papel fundamental dos frameworks MVC em diminuir as dificuldades trazidas ao se implementar o padrão arquitetural mencionado, evitando retrabalho do programador em funções repetitivas da infraestrutura do projeto e permitindo que o desenvolvedor foque apenas nas regras de negócio da aplicação. Além disso, os frameworks MVC trazem também as vantagens do padrão arquitetural MVC, diminuindo o acoplamento e aumentando a coesão entre as classes e componentes do sistema, o que permite a não só a reutilização de código como também torna o projeto manutenível. Assim, atrasos na entrega do projeto podem ser evitados, já que se utilizando de tais frameworks o programador pode ganhar maior produtividade e facilidade em solucionar erros.

Para futuros trabalhos seria interessante explorar em detalhes a implementação do framework MVC, incluindo suas configurações, mapeamento e uso na aplicação, como a facilidade de popular objetos bean em Java.

Referências

CAELUM. *Spring MVC*. 2004. Último acesso em 31/07/2018. Disponível em: <https://www.caelum.com.br/apostila-java-web/spring-mvc/>. Nenhuma citação no texto.

EDUARDO. *Java Spring MVC: Criando Aplicações Web em Java*. 2014. Último acesso em 31/07/2018. Disponível em: <https://www.devmedia.com.br/java-spring-mvc-criando-aplicacoes-web-em-java/31521>. Nenhuma citação no texto.

JúNIOR, J. R. N. F. *Estudo de caso usando o Spring framework para a criação de aplicações WEB com J2EE*. Tese (Doutorado) — Universidade Federal de Santa Maria, 2007. Citado na página 15.