

Relatório MVC: TP3

Gabriel Campello Marques
Arthur Grandão de Mello
Rafael Ferreira Leandro
Luis Pedro de Castro Alves

Data: 20/09/2022

Nome do professor: Andre
Luiz Peron Martins Lanna

Objetivo

Apresentar o padrão de projeto MVC (model-view-controller) e suas vantagens na criação e organização de uma aplicação

Introdução ao padrão MVC

A arquitetura de um sistema tem diversos elementos como: elementos utilitários, de interação, elementos que fazem parte do domínio do problema, elementos de conexão, de persistência etc. Dessa forma, na arquitetura sempre definimos os seus elementos que serão utilizados no software e como eles se conectam. Uma arquitetura complexa exige mais tempo para desenvolvimento, porém, através de geração automática de aplicações torna-se mais produtivo. Por isso algumas equipes definem um framework para uma determinada aplicação e assim podemos utilizar muitas coisas pré-prontas que facilitam o desenvolvimento. Visando maior organização e facilitar o desenvolvimento da aplicação foi criado o padrão de projeto MVC. A implementação original foi descrita no artigo "Applications Programming in Smalltalk-80: How to use Model-View-Controller", a ideia era criar um padrão cujo objetivo é separar o projeto em três camadas independentes, que são o modelo, a visão e o controlador. Essa separação de camadas ajuda na redução de acoplamento e promove o aumento de coesão nas classes do projeto. Assim, quando o modelo MVC é utilizado, pode facilitar a manutenção do código e sua reutilização em outros projetos.

Vantagens do padrão MVC

Baixo acoplamento: é o grau em que uma classe conhece a outra. Se o conhecimento da classe A sobre a classe B for através de sua interface, temos um baixo acoplamento, e isso é bom. Por outro lado, se a classe A depende de membros da classe B que não fazem parte da interface de B, então temos um alto acoplamento, o que é ruim.

Coesão: quando temos uma classe elaborada com um único e bem focado propósito, dizemos que ela tem alta coesão, e isso é bom. Quando temos uma classe com propósitos que não pertencem apenas a ela, temos uma baixa coesão, o que é ruim.

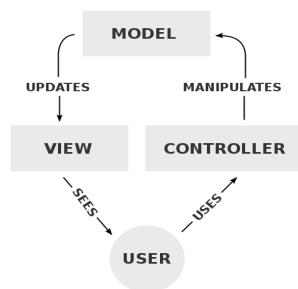
Descrição do padrão MVC

A comunicação entre interfaces e regras de negócios é definida através de um controlador, que separa as camadas. Quando um evento é executado na interface gráfica, como um clique em um botão, a interface se comunicará com o controlador, que por sua vez se comunica com as regras de negócios. Imagine uma aplicação financeira que realiza cálculos de diversos tipos, como os de juros. Você pode inserir valores para os cálculos e também escolher que tipo de cálculo será realizado. Isto tudo é feito pela interface gráfica, que para o **modelo MVC é conhecida como View**. No entanto, o sistema precisa saber que você está requisitando um cálculo, e para isso, terá um botão no sistema que quando clicado gera um evento.

Este evento pode ser uma requisição para um tipo de cálculo específico como o de juros simples ou juros compostos. Fazem parte da requisição os valores digitados no formulário e a seleção do tipo de cálculo que o usuário quer executar sobre o valor informado. O evento do botão é como um pedido a um intermediador (Controller) que prepara as informações para então enviá-las para o cálculo. O controlador é o único no sistema que conhece o responsável pela execução do cálculo, neste caso, a camada que contém as regras de negócios. Esta operação matemática será realizada pelo Model assim que ele receber um pedido do Controller.

O Model realiza a operação matemática e retorna o valor calculado para o Controller, que também é o único que possui conhecimento da existência da camada de visualização. Tendo o valor em

mãos, o intermediador o repassa para a interface gráfica que exibirá para o usuário. Caso esta operação deva ser registrada em uma base de dados, o Model se encarrega também desta tarefa. Exemplificação visual a seguir:



As camadas do padrão MVC e suas funcionalidades

A Camada View do MVC

- É a camada que exibe uma representação dos dados
- É a camada de interface com usuário (view).
- Também conhecida como cliente-side.
- Faz a exibição dos dados a serem inseridos pelo usuário
- É responsável por usar as informações modeladas para produzir interfaces de apresentação conforme a necessidade.

A Camada Model do MVC

- É a camada que contém a estrutura de dado atrás de uma parte específica da aplicação
- Responsável pela leitura, manipulação e validação de dados.
- Responsável por tratar as regras de negócio.
- Obtém os dados e os traduz em informações relevantes para serem exibidas pela View.
- Notifica a view e controller associados quando há uma mudança em seu estado.

A Camada Controller do MVC

- É a camada de controle.
- Exerce o controle de qual model deverá ser aplicado e qual view será mostrado ao usuário.
- Podemos dizer que esta camada faz uma gerência das outras duas camadas.
- O controller manipula e roteia as requisições dos usuários.
- Interpreta as requisições submetidas pelo usuário e traduz em comandos que são enviados para o (Model) e/ou para a View) .
- Valida as requisições dos usuários de acordo com as regras de autenticação e autorização definidas na model.

Referencias

[Learn to make a MVC application with Swing and Java 8 | by Sylvain Saurel | Medium](#)

[Padrão MVC | Arquitetura Model-View-Controller - Portal GSTI](#)

[Introdução ao Padrão MVC: Primeiros passos na Arquitetura MVC \(devmedia.com.br\)](#)