

Desenvolvimento Web II

Aula 02 - Padrões de projeto

Prof. Fabricio Bizotto

Instituto Federal Catarinense

fabricio.bizotto@ifc.edu.br

Ciência da Computação

9 de janeiro de 2024

1 Padrões de Projeto para Web

- Conceitos
- Padrões de Projeto para Web
 - MVC
- Padrões de Projeto para Web
 - MVP
- Padrões de Projeto para Web
 - MVVM
- Material Complementar

2 Referências

Conceitos

Os **padrões de projeto para web** são soluções reutilizáveis para problemas comuns de design de software que surgem no desenvolvimento de aplicativos web. Eles fornecem **diretrizes e estruturas** para organizar o código, melhorar a escalabilidade, a manutenibilidade e a eficiência do desenvolvimento. Os mais comuns são:

- **Modelo-Visão-Controlador (MVC)**
- **Modelo-Visão-Presenter (MVP)**
- **Modelo-Visão-ViewModel (MVVM)**

Padrões de Projeto

MVC

Model-View-Controller

Padrões de Projeto para Web

MVC - Model-View-Controller

Definição

É um dos padrões de arquitetura mais conhecidos e adotados pela indústria de software. Foi introduzido pela primeira vez no final da década de 1970 por Trygve Reenskaug, um cientista da computação norueguês, e desde então se tornou um elemento básico na arquitetura de aplicativos. O padrão facilita a separação de interesses dividindo o aplicativo em três componentes principais.



Padrões de Projeto para Web

MVC - Model-View-Controller

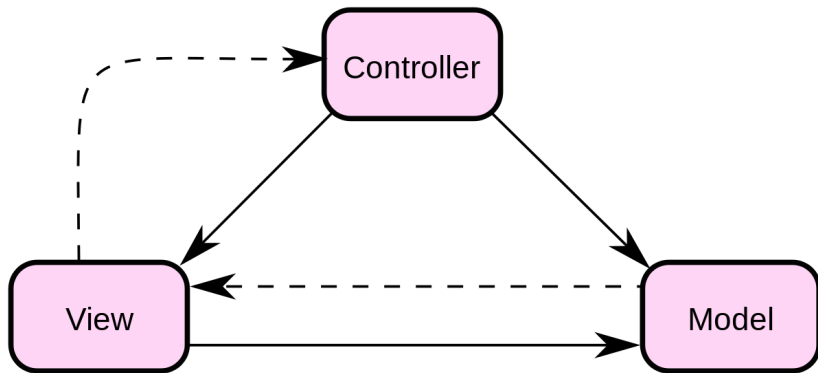


Figura: Estrutura do MVC.

Explicando

- **Model:** define quais dados o aplicativo deve conter. Se o estado desses dados mudar, o modelo geralmente notificará a visualização (*View*) (para que a exibição possa mudar conforme necessário) e, às vezes, o controlador (se for necessária uma lógica diferente para controlar a visualização atualizada).
- **View:** define como os dados do aplicativo devem ser exibidos (Interface com o Usuário). A visualização é responsável por receber a entrada do usuário e encaminhá-la para o controlador.
- **Controller:** atua como intermediário entre o modelo e a visualização. O controlador é responsável por receber a entrada do usuário da visualização e atualizar o modelo conforme necessário.

Padrões de Projeto

MVP

Model-View-Presenter

Padrões de Projeto para Web

MVP - Model-View-Presenter

Definição

Aborda algumas das desvantagens da abordagem MVC tradicional. Originou-se no início da década de 1990 na Taligent, uma joint venture entre Apple, IBM e Hewlett-Packard. Foi ainda mais popularizado pelo Dolphin Smalltalk em 1998 e, em 2006, a Microsoft adotou o MVP para programação de interface de usuário no framework .NET.



Padrões de Projeto para Web

MVP - Model-View-Presenter

No MVP quem manda é o View. Cada View chama seu Presenter ou possui alguns eventos que o Presenter escuta.

Exemplo

Quando o usuário clica no botão “Salvar”, o manipulador de eventos na View delega ao método “OnSave” do Presenter. O Presenter fará a lógica necessária e qualquer comunicação necessária com o Modelo e, em seguida, chamará de volta a Visualização por meio de sua interface para que a Visualização possa exibir que o salvamento foi concluído.

Padrões de Projeto para Web

MVP - Model-View-Presenter

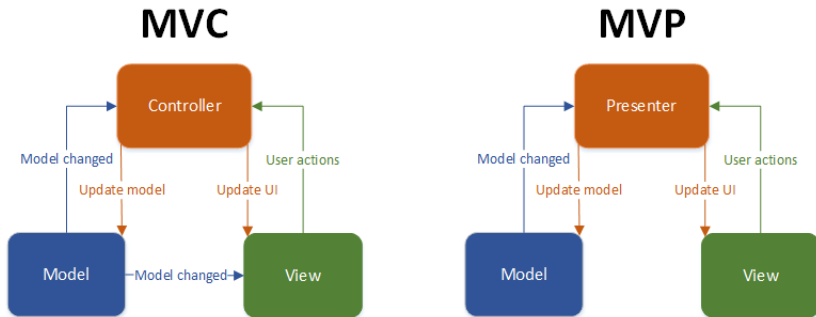


Figura: Estrutura do MVP.

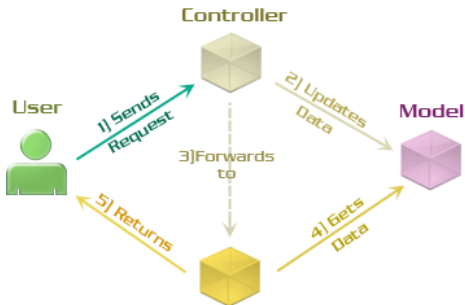
MVC vs MVP

- O **MVC** não coloca o View no comando, os Views atuam como escravos que o Controlador pode gerenciar e direcionar.
- No **MVC**, as visualizações são sem estado, ao contrário das visualizações no MVP, onde são com estado e podem mudar com o tempo.
- No **MVP**, as Views não têm lógica e devemos mantê-las o mais burras possível. Por outro lado, Views em MVC podem ter algum tipo de lógica.
- No **MVP**, o Presenter é desacoplado da View e se comunica com ela através de uma interface. Isso permite zombar do View em testes unitários, ou seja, podemos testar o Presenter sem o View.
- No **MVP**, as visualizações são completamente isoladas do modelo. No entanto, no MVC, as Views podem se comunicar com o Modelo para mantê-lo atualizado com os dados mais atualizados.

Padrões de Projeto para Web

MVC vs MVP

Model View Controller



Model View Presenter



Padrões de Projeto

MVVM

Model-View-ViewModel

Padrões de Projeto para Web

MVVM - Model-View-ViewModel

Definição

Aborda algumas das desvantagens da abordagem MVC tradicional. Originou-se no início da década de 1990 na Taligent, uma joint venture entre Apple, IBM e Hewlett-Packard. Foi ainda mais popularizado pelo Dolphin Smalltalk em 1998 e, em 2006, a Microsoft adotou o MVP para programação de interface de usuário no framework .NET.



Padrões de Projeto para Web

MVVM - Model-View-ViewModel

O MVVM foi criado pelo arquiteto de software do WPF (Windows Presentation Foundation) e Silverlight da Microsoft, John Grossman em 2005. Desde então, ele vem sendo usado principalmente no desenvolvimento mobile. Ele foi criado para ser usado em aplicativos WPF e usava XAML (uma linguagem declarativa para objetos e suas propriedades) a fim de separar a interface do usuário da lógica de negócios, aproveitando o data binding que é a vinculação de dados. Na prática, a camada Model não se comunica com a View nem a View se comunica com a Model.

Onde é usado?

O MVVM é particularmente adequado para aplicativos de **UI complexos**, onde é necessária uma extensa ligação de dados, e para projetos que usam estruturas como WPF, UWP, Angular e Xamarin. Com seu forte foco no desenvolvimento de UI, o MVVM se tornou popular no mundo do **desenvolvimento móvel**.

Model-View-ViewModel

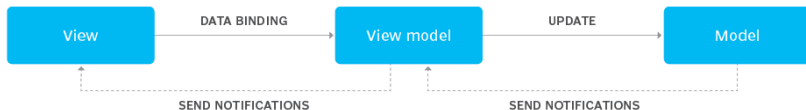


Figura: Estrutura do MVP.

Material Complementar

Vídeos, Podcasts, Livros, etc

- **MVVM (Model View ViewModel - A Arquitetura de Apps Mobile) // Dicionário do Programador.**
Canal **Código Fonte TV**.
- **Arquitetura de Software: Monolítica x SOA x Microserviços.**
Canal **Marcos Dósea**.
- **Microservices // Dicionário do Programador.**
Canal **Código Fonte TV**.
- **Microservices na prática.**
Canal **Full Cycle**.
- **Microservices na prática.**
Podcast Hipsters Ponto Tech **Monolitos**.

Referências