

# Padrões de Arquitetura de Software

Leonardo Rocha

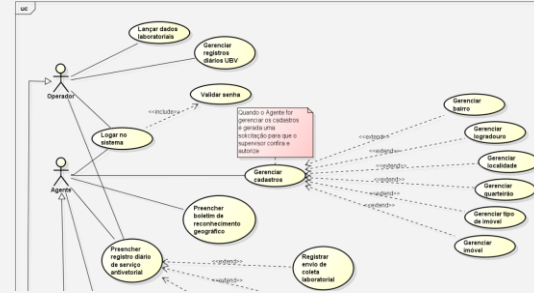




## Requisitos do sistema



## Arquitetura



## Implementação



# Arquitetura

“

A arquitetura define os elementos de software e como eles interagem entre si, além das suas conexões e de sua persistência.

Os padrões de arquitetura facilitam a implementação e resolução de problemas por meio de soluções conhecidas.

”

(ZENKER, 2019)

# Introdução

O que são padrões de arquitetura de software?



- **Facilidade de manutenção e evolução**

- \* Separação das preocupações do negócio
- \* Modularidade

- **Reusabilidade**

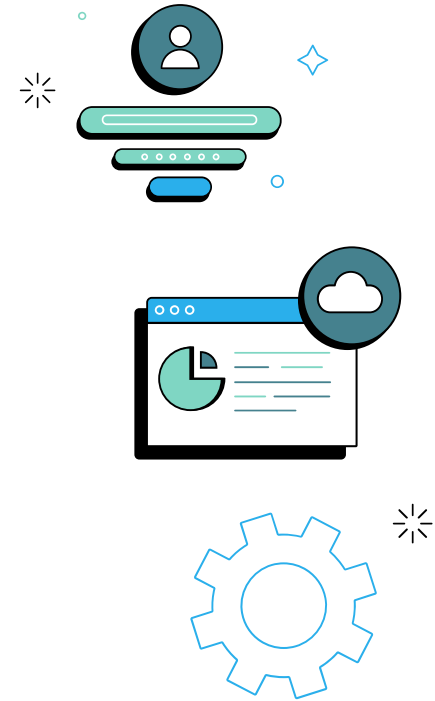
- \* Componentes reutilizáveis
- \* Bibliotecas e frameworks (Laravel, Node.js)

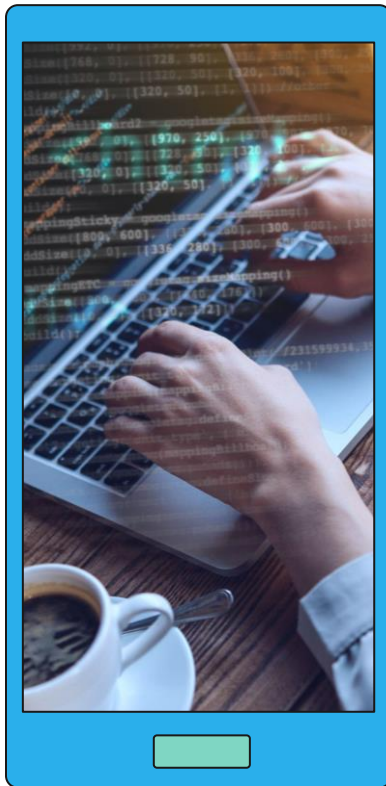
- **Testabilidade**

- \* Isolamento de componentes
- \* Automação de testes

# Objetivo

- Compreender os padrões **MVC**, MVP e MVVM;
- Identificar características e diferenças entre os padrões;
- Saber escolher um dos padrões;

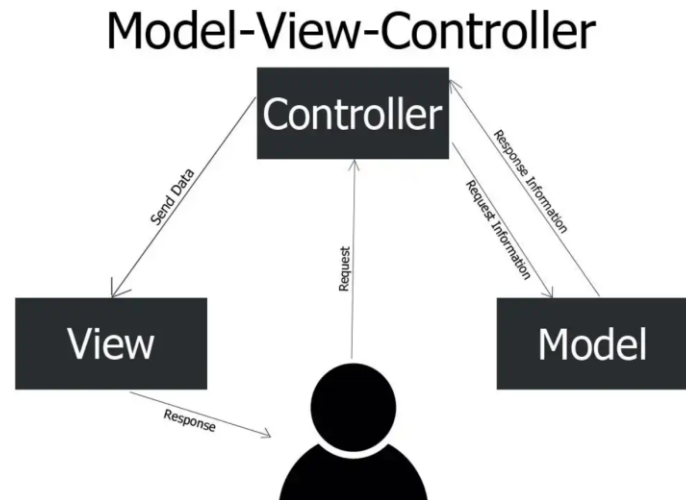




# MVC

<b>Model</b>	Gerencia dados e lógica de negócio
<b>View</b>	Exibe dados ao usuário
<b>Controller</b>	Interpreta entradas do usuário, atualiza o Model e a View

(GAMA et. al., 2007)



# Vantagens e Desvantagens



## - Vantagens

- \* Separação das preocupações
- \* Testabilidade
- \* Reusabilidade

## - Desvantagens

- \* Complexidade
- \* Dificuldade em gerenciar grandes aplicativos



# MVC - Benefícios

Aumento de produtividade;

Padronização na estrutura do software;

Facilidade para manter a aplicação;

Redução da complexidade do código;

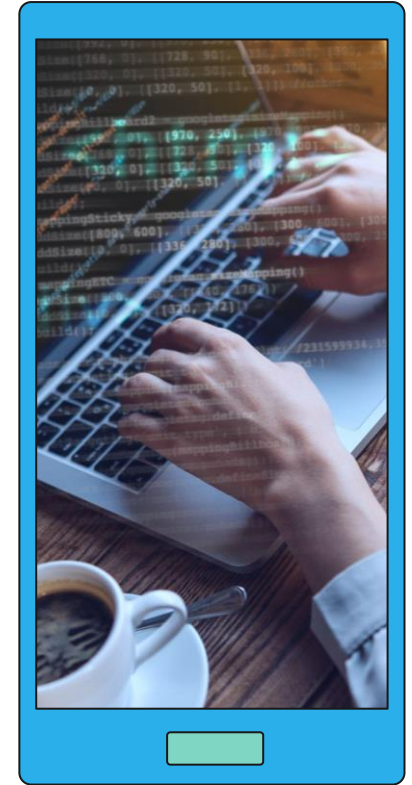
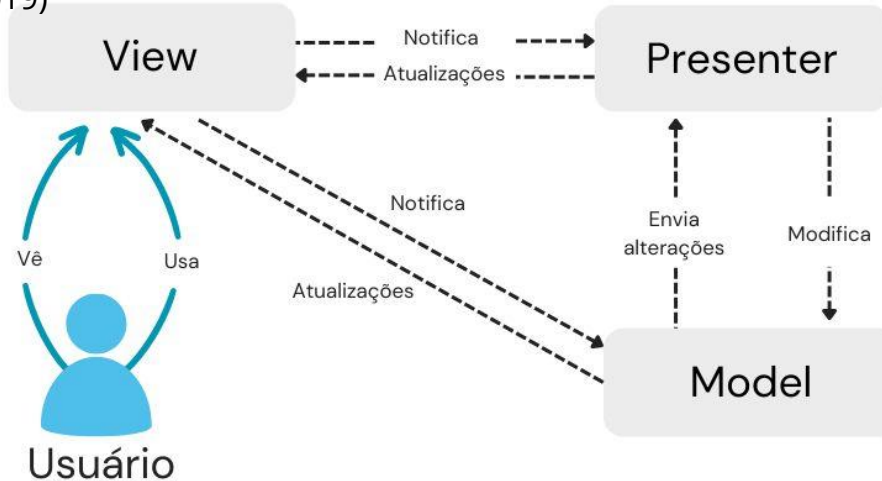
Documentação implementada ou facilitada;

Vocabulário comum de projeto entre desenvolvedores;

# MVP

<b>Model</b>	Gerencia dados e lógica de negócio
<b>View</b>	Interface com o usuário
<b>Presenter</b>	Atua como intermediário, manipula o Model e atualiza a View

(ZENKER et. al., 2019)



# Vantagens e Desvantagens



## - Vantagens

- \* Melhor testabilidade
- \* Separação clara de responsabilidades

## - Desvantagens

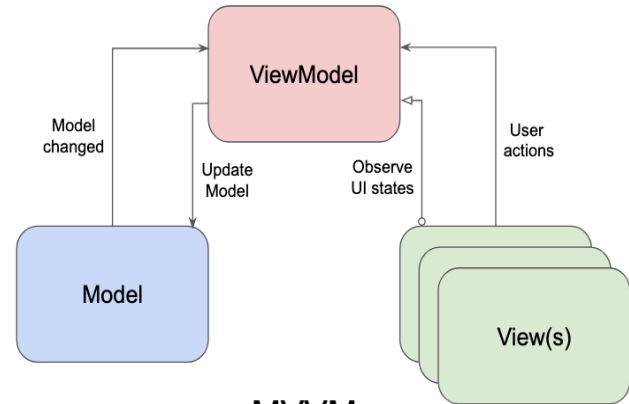
- \* Pode induzir a complexidade adicional



# MVVM

<b>Model</b>	Gerencia dados e lógica de negócio
<b>View</b>	Interface com o usuário
<b>ViewModel</b>	Manipula o Model e expõe dados de forma que a View possa consumir.

(ZENKER et. al., 2019)



MVVM

# Vantagens e Desvantagens



## - Vantagens

- \* Excelente para ***data binding***
- \* Facilita a separação de UI e lógica de negócio

## - Desvantagens

- \* Alta curva de aprendizado
- \* Pode ser difícil implementar corretamente

# • Comparação

## MVC

Melhor para web  
Controle centralizado

## MVP

Melhor para desktop e mobile  
Testabilidade

## MVVM

Ideal para data binding  
Frameworks como WPF e Xamarim

Aplicações web tradicionais

Aplicações com lógica de interface complexa

Aplicações com forte necessidade de data binding

**Quando usar cada padrão**



# Saiba mais...

## **Introdução ao padrão MVC**

Disponível em: <https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>. Acesso em: 11 jun. 2024.

## **Padrão MVC**

Disponível em: <https://www.devmedia.com.br/padrao-mvc-java-magazine/21995>. Acesso em: 11 jun. 2024.

# Referências

FOWLER, Martin. Padrões de arquitetura de aplicações corporativas. Porto Alegre: Bookman, 2007.

GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; et al. Padrões de projetos: soluções reutilizáveis de software orientados a objetos. Porto Alegre: Bookman, 2007.

ZENKER, Aline M.; SANTOS, Jailson Costa dos; COUTO, Júlia M C.; et al. Arquitetura de sistemas. Porto Aletre: SAGAH, 2019.





lkr15tecnologia



leonardossrocha



leonardossrocha

