



# ARQUITETURA DE APLICATIVOS

CONCEITO, MVC, MVVM E CLEAN ARCHITECTURE



# ARQUITETURA DE SOFTWARE

## DEFINIÇÃO

- A ARQUITETURA DE SOFTWARE É A ESTRUTURA FUNDAMENTAL DE UM SISTEMA, INCLUINDO A ORGANIZAÇÃO DE SEUS COMPONENTES, SUAS RELAÇÕES E SEUS PRINCÍPIOS DE DESIGN.

## CONCEITOS-CHAVE

- **SEPARAÇÃO DE RESPONSABILIDADES:** OS DIFERENTES COMPONENTES DO APLICATIVO DEVEM TER RESPONSABILIDADES CLARAS E BEM DEFINIDAS.
- **COESÃO:** OS ELEMENTOS RELACIONADOS DEVEM ESTAR AGRUPADOS DE FORMA LÓGICA E COESA.
- **ACOPLAMENTO:** OS ELEMENTOS DEVEM SER INTERDEPENDENTES NA MEDIDA CERTA, EVITANDO DEPENDÊNCIAS EXCESSIVAS.
- **LEGIBILIDADE DO CÓDIGO:** UM CÓDIGO LEGÍVEL FACILITA A MANUTENÇÃO E A COLABORAÇÃO NO DESENVOLVIMENTO DO APLICATIVO.

# ARQUITETURA DE SOFTWARE

## VANTAGENS

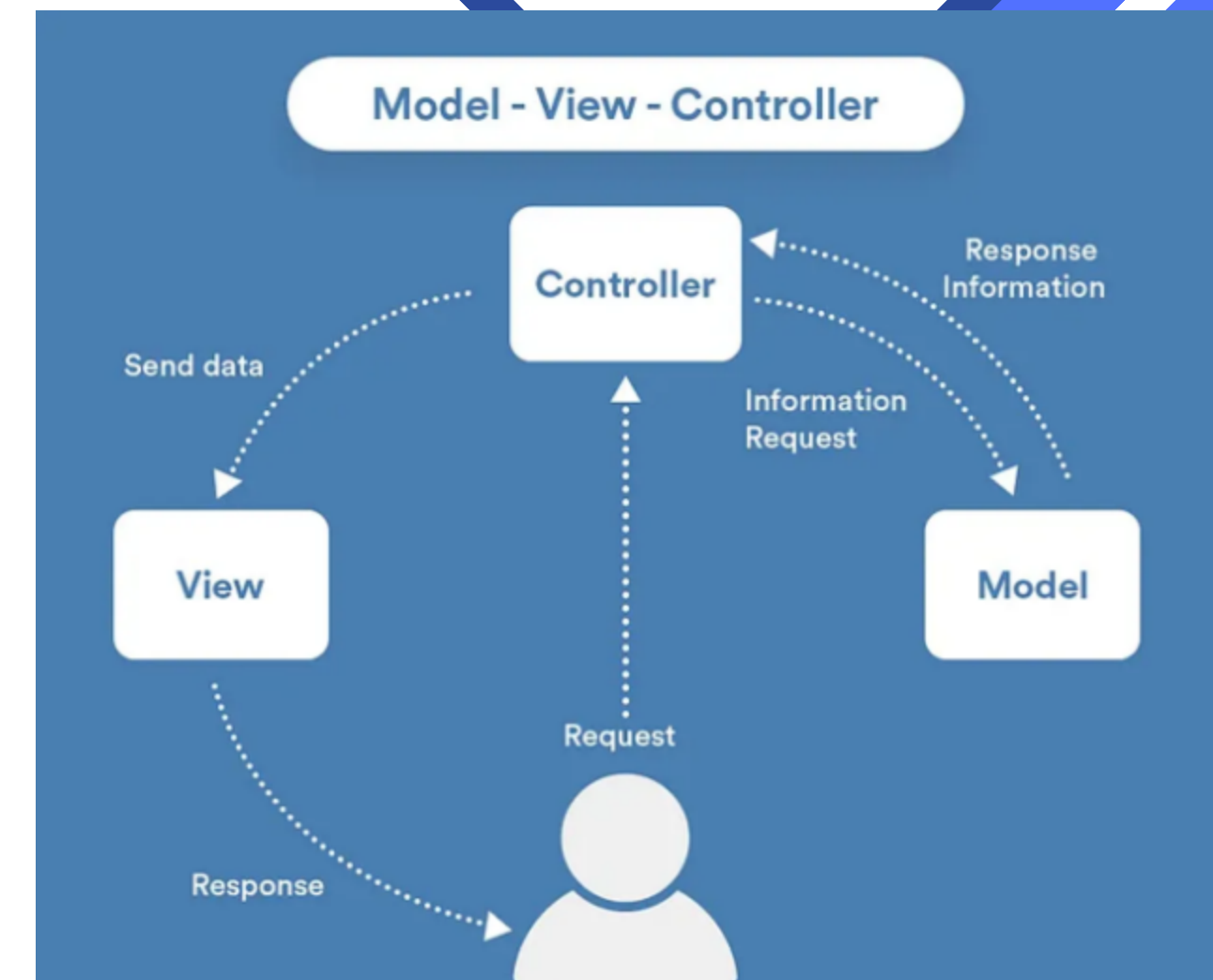
- UMA ARQUITETURA BEM DEFINIDA TRAZ DIVERSOS BENEFÍCIOS, COMO MODULARIDADE, ESCALABILIDADE E FACILIDADE DE MANUTENÇÃO.
- UMA BOA ARQUITETURA PERMITE QUE O APLICATIVO CRESÇA DE FORMA ORGANIZADA E SEJA MAIS FÁCIL DE MANTER E ATUALIZAR.

## TIPOS DE ARQUITETURAS

- MVC ( MODEL-VIEW-CONTROLLER)
- MVVM (MODEL-VIEW-VIEW-MODEL)
- CLEAN ARCHITECTURE

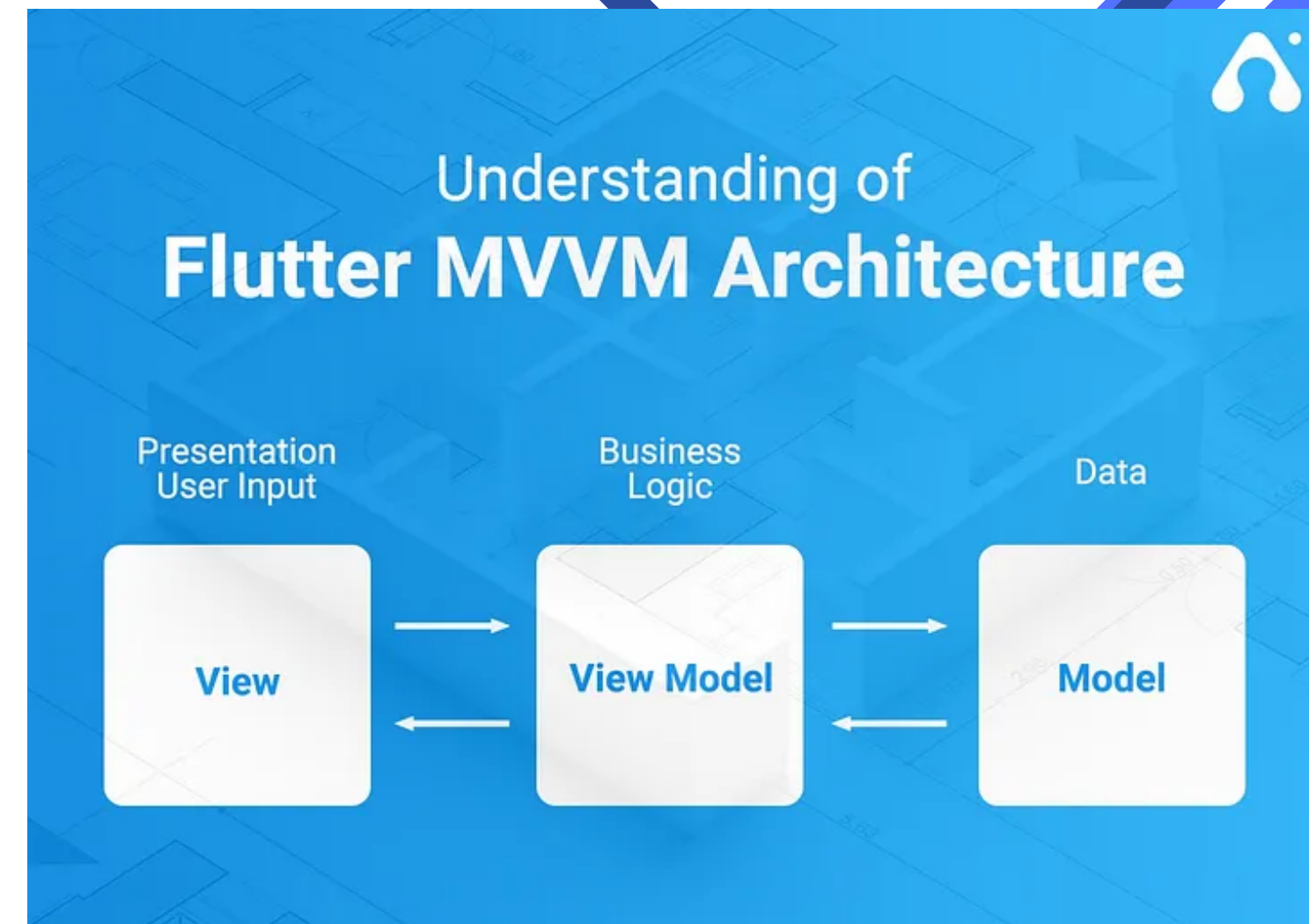
# ARQUITETURA MVC

- DIVIDE O APLICATIVO EM TRÊS COMPONENTES PRINCIPAIS - MODELO, VISUALIZAÇÃO E CONTROLADOR
- **MODEL:** FLUXO DE DADOS PRINCIPAL NESTA ARQUITETURA. ELE TAMBÉM PODE CONTER LÓGICA DE NEGÓCIOS.
- **VIEW:** RECEBE A ENTRADA DO USUÁRIO E MOSTRAR A INTERFACE DO USUÁRIO PARA O USUÁRIO..
- **CONTROLLER:** CONTÉM LÓGICA DE NEGÓCIOS COM MANIPULAÇÃO DE DADOS DE ENTRADA.



# ARQUITETURA MVVM

- MVVM É UM PADRÃO DE ARQUITETURA MUITO FAMOSO E ESTABELECIDO EM DESENVOLVIMENTO. AO APLICAR O MVVM, NOSSO CÓDIGO SERÁ REUTILIZÁVEL, SUSTENTÁVEL, FACILMENTE TESTÁVEL E EXTENSÍVEL.
- **MODEL:** CONTEM LÓGICA DE NEGÓCIOS, VALIDAÇÃO DE CÓDIGO E CONSULTAS RELACIONADAS AO BANCO DE DADOS INTERAGINDO COM VIEWMODEL.
- **VIEWMODEL:** SOLICITA DADOS PARA O MODELO E OS ENVIA PARA A VIEW. O VIEWMODEL É O MEDIADOR ENTRE O VIEW E O MODEL. PRIMEIRO, VIEWMODEL LIDA COM INTERAÇÕES DO USUÁRIO, EM SEGUIDA, ELE SOLICITA DADOS DO MODEL E OS ENVIA PARA A VIEW.
- **VIEW:** É PRINCIPALMENTE A INTERFACE DO USUÁRIO DE UM APLICATIVO ONDE UM USUÁRIO INTERAGE COM WIDGETS.



# ARQUITETURA CLEAN ARCHITECTURE

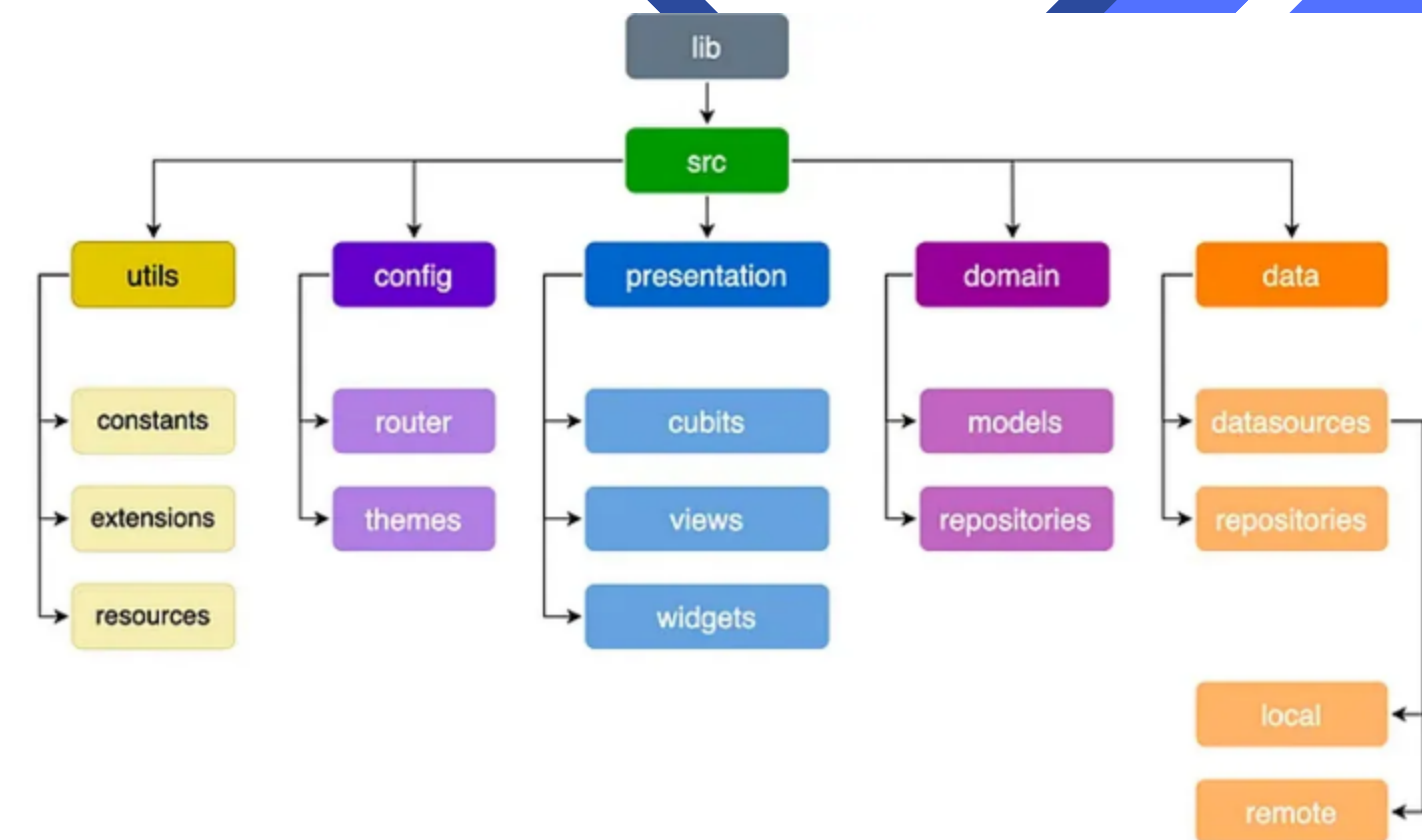
- A ARQUITETURA LIMPA É A ARQUITETURA MAIS PODEROSA QUE PROMOVE A SEPARAÇÃO DE PREOCUPAÇÕES E A CRIAÇÃO DE SISTEMAS DE SOFTWARE ALTAMENTE SUSTENTÁVEIS E FLEXÍVEIS.

- **PRÓS:**

- ORGANIZAÇÃO
- INDEPENDÊNCIA DE ESTRUTURAS EXTERNAS
- ESTABILIDADE
- FLEXIBILIDADE

- **CONTRAS:**

- SOBRECARGA
- COMPLEXIDADE
- CURVA DE APRENDIZADO
- DEMORADO



# ATIVIDADE

1

## ESCOLHA DA ARQUITETURA

*Os alunos devem escolher uma das opções de arquitetura (MVC, MVVM ou Clean Architecture) para implementar no aplicativo.*

2

## ORGANIZAÇÃO DO PROJETO

*Os alunos devem organizar o projeto em camadas, de acordo com a arquitetura escolhida.*

3

## DOCUMENTAÇÃO E APRESENTAÇÃO

*Os alunos devem documentar o processo de implementação da arquitetura escolhida e preparar uma apresentação para compartilhar suas experiências e desafios enfrentados.*



# CONCLUSÃO

- A arquitetura de aplicativos desempenha um papel fundamental no desenvolvimento de aplicativos móveis
- Os conceitos de arquitetura de software, como separação de responsabilidades, coesão, acoplamento e legibilidade do código, são fundamentais para criar uma estrutura sólida e bem organizada.
- O Flutter oferece suporte a várias arquiteturas, como MVC, MVVM e Clean Architecture, que podem ser escolhidas de acordo com as necessidades do projeto