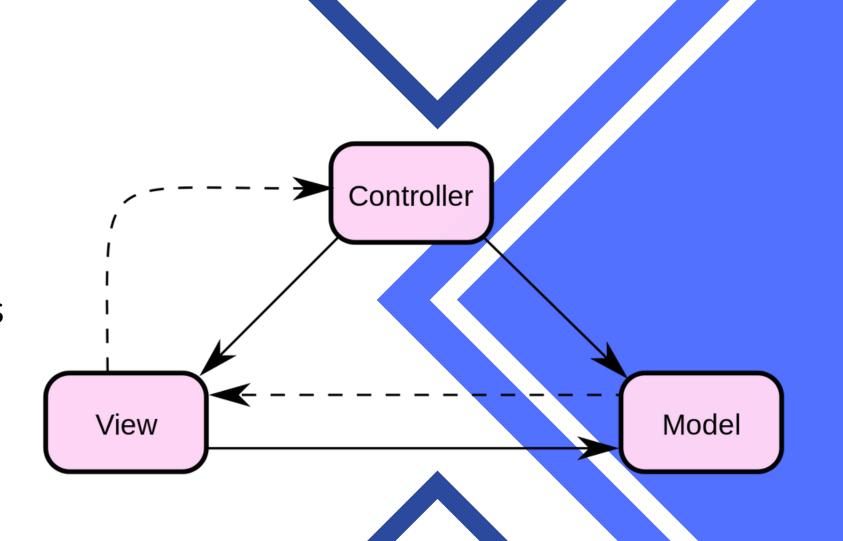
ARQUITETURA FLUTTER

INTRODUÇÃO ÀS ARQUITETURAS DO FLUTTER

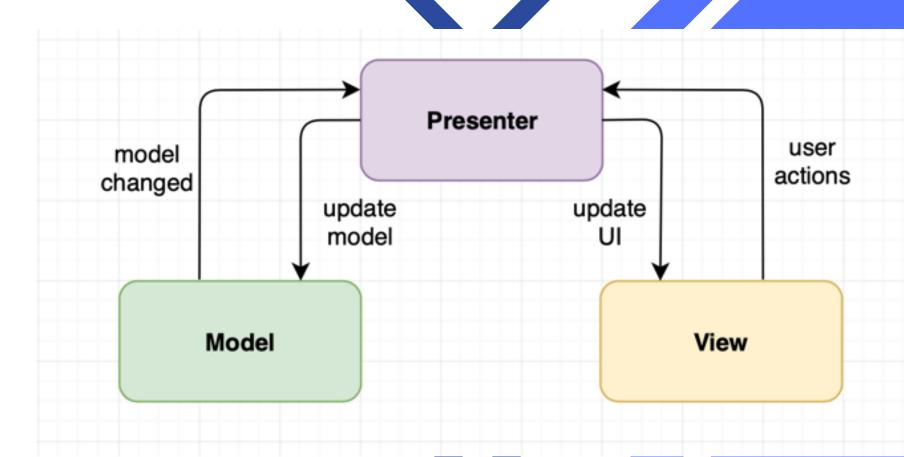
MVC

- Arquitetura recomendada para o desenvolvimento de aplicativos Flutter é baseada no padrão de arquitetura MVC (Model-View-Controller)
- Separar "as áreas de trabalho" e "as áreas de responsabilidade" ao desenvolver software
- Isso significa separar "a interface" da "funcionalidade" dos "dados" que compõem o aplicativo



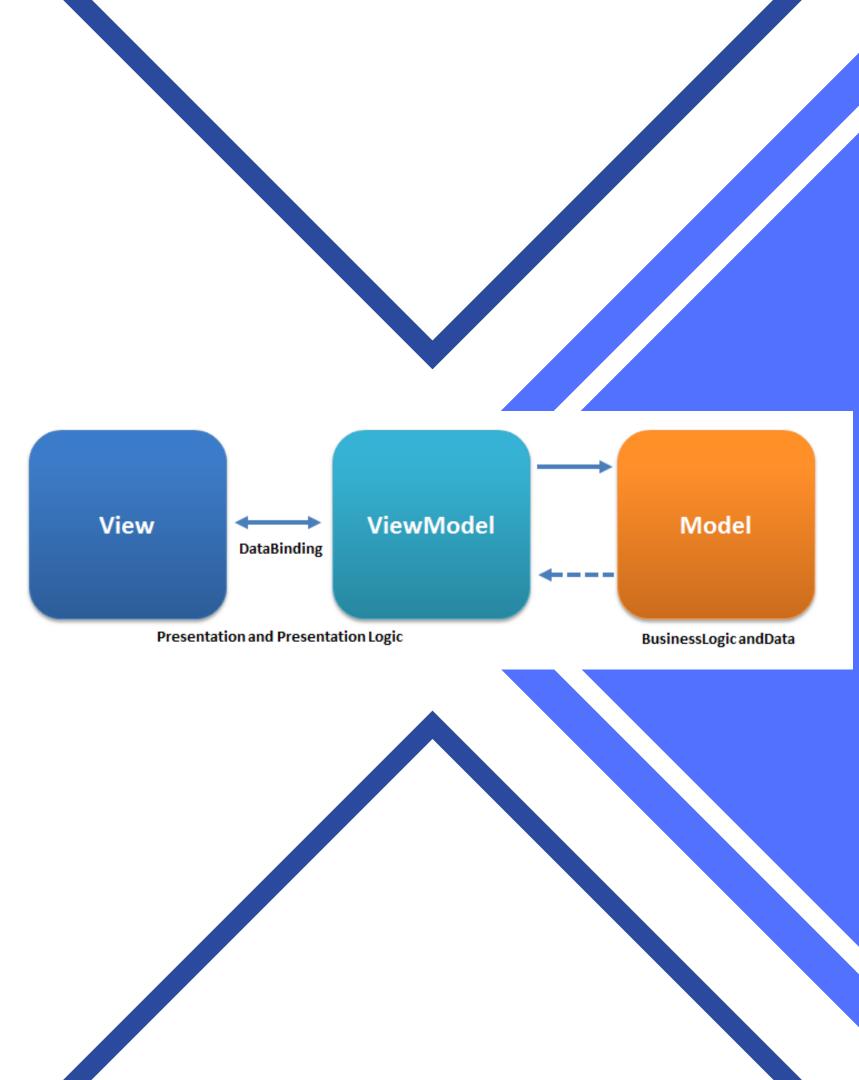
MVP

- A arquitetura MVP (Model-View-Presenter) é um padrão de design que tem como objetivo separar a lógica de negócios da interface do usuário
- O modelo uma interface que define os dados a serem exibidos ou manipulados na interface do usuário
- A visão é uma interface passiva que exibe dados (o modelo) e roteia comandos do usuário (eventos) para o apresentador agir sobre esses dados
- O apresentador recupera dados dos repositórios, o modelo e os formata para exibição na visão



MVVM

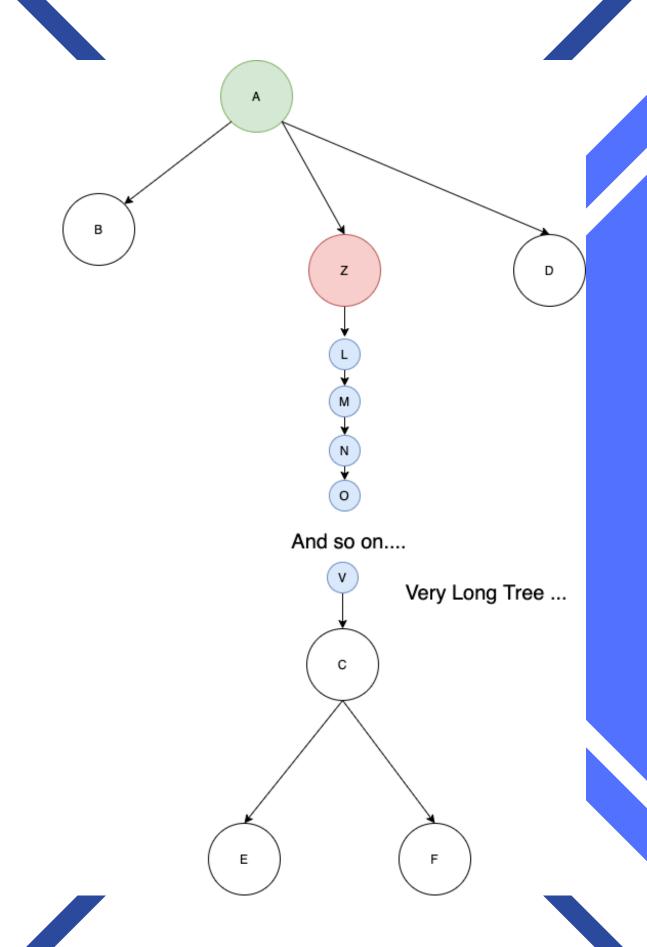
- Model-View-ViewModel (MVVM), separa o modelo de dados de um aplicativo, a interface do usuário e a lógica do aplicativo em camadas distintas.
- O modelo de dados representa os dados e a lógica de negócios do aplicativo
- A visualização (view) é a interface do usuário (IU) do aplicativo
- O ViewModel é responsável por conectar o Model e a View, fornecendo dados para a exibição e comunica as interações do usuário ao modelo



OUTRAS ARQUITETURAS

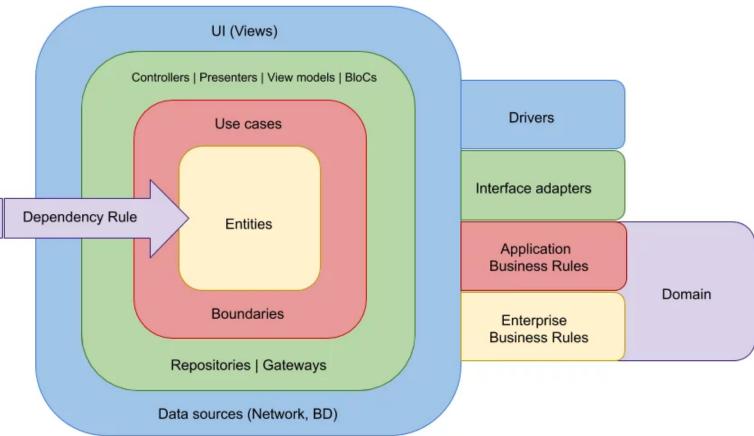
PROVIDER

- Fácil de aprender e possui muito menos código repetitivo em comparação com outros padrões, como a Arquitetura BLoC, que usa o Provider internamente
- Responsável por observar um objeto e avisar para todos os componentes interessados se esse objeto mudou, para que eles se atualizem
- Isso é conhecido como "Single Source of Truth" ou "Fonte Única da Verdade"



BLOC

- A arquitetura BLoC (Business Logic Component) permite uma separação da lógica de negócio da sua aplicação e da UI através do uso de Streams
- É construída toda a lógica de negócios
 com essas streams, o que nos permite o uso de bibliotecas Reactive como a RxDart
- É possível fazer streams que se originam da arquitetura BLoC e podemos ter streams que se conectam da UI para a BLoC



MOBX

- MobX torna o gerenciamento de estado simples e escalável, aplicando transparentemente a programação reativa funcional
- Escrever código minimalista e sem boilerplate que capture sua intenção
- O MobX é descomprometido e permite que você gerencie o estado do seu aplicativo fora de qualquer estrutura de IU, tornando seu código desacoplado, portátil e, acima de tudo, facilmente testável

CONCLUSÃO

- INTRODUÇÃO ÀS ARQUITETURAS DO FLUTTER
- IMPLEMENTAÇÃO DE PADRÕES DE ARQUITETURAS ESPECÍFICOS

