



Instrutor: Ralf Lima

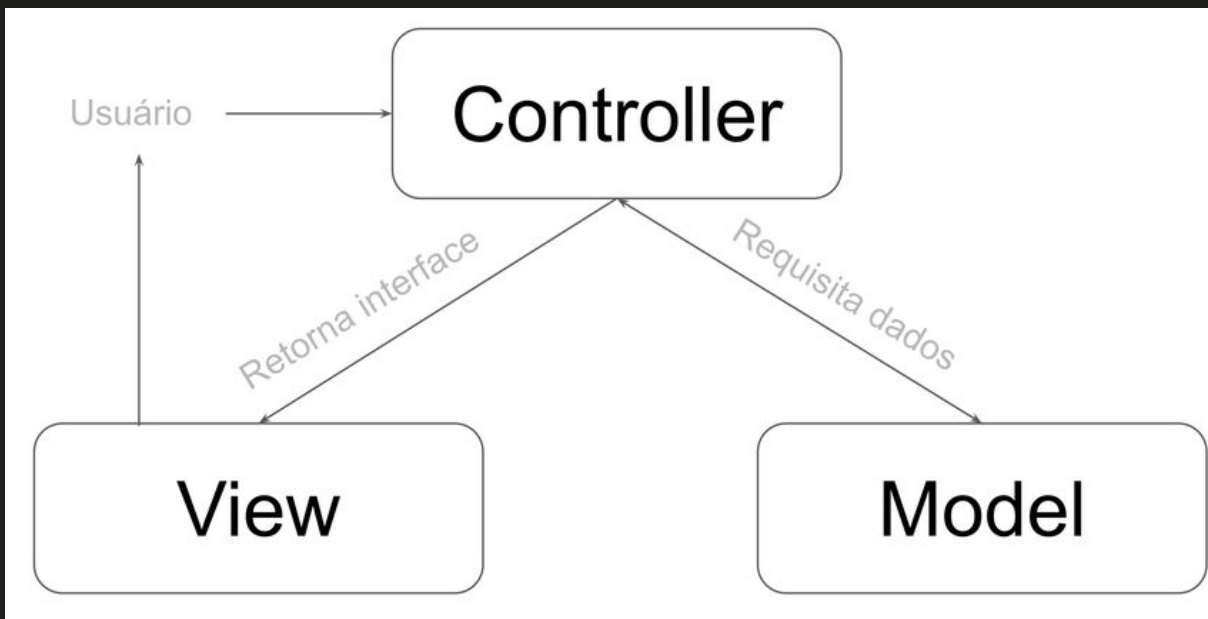
Tópicos da aula

- Padrões de projetos*;
- Responsividade;
- APIs;
- Tasks (Tarefas).

Padrões de projetos

Vamos explicar os três principais padrões de projetos, além de disponibilizar materiais para estudar outros padrões, considerados mais avançados.

MVC



MVC

Vantagens

- Projeto organizado em camadas;
- Facilidade na compreensão do projeto;
- Reutilização.

MVC

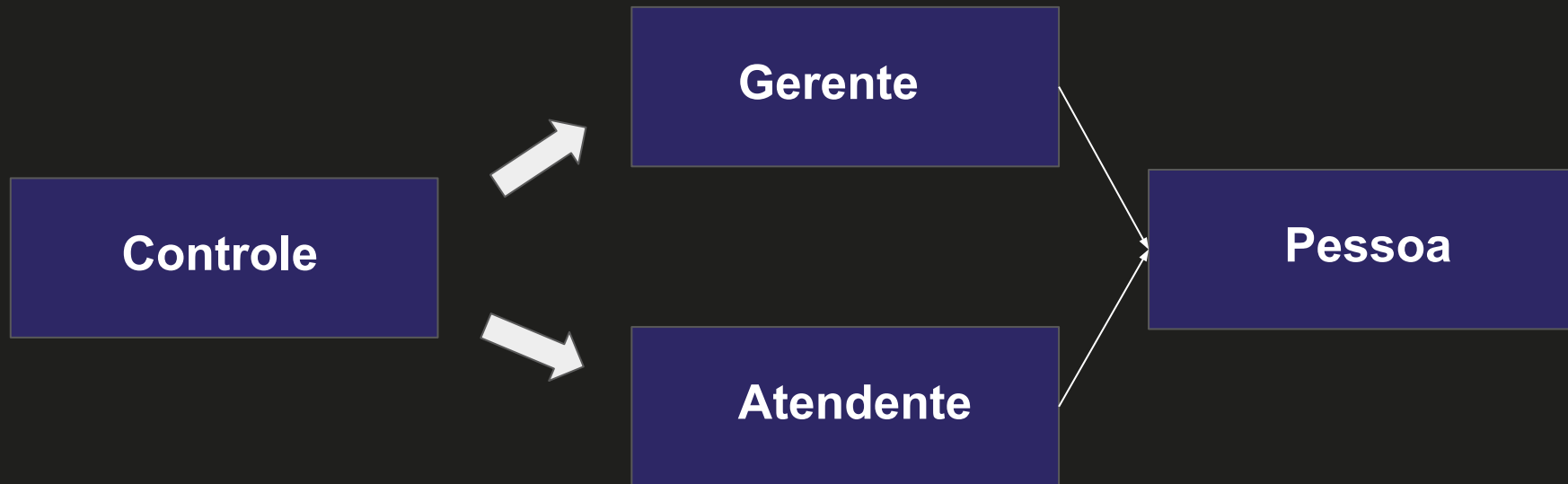
Desvantagens

- Curva de aprendizagem elevada;
- Dispõe de limitação para projetos maiores;
- Nível alto de acoplamento.

Acoplamento?



Acoplamento?



Acoplamento?

Controle

Goiânia

Porto Alegre

Gerente

Atendente

Financeiro

Gerente

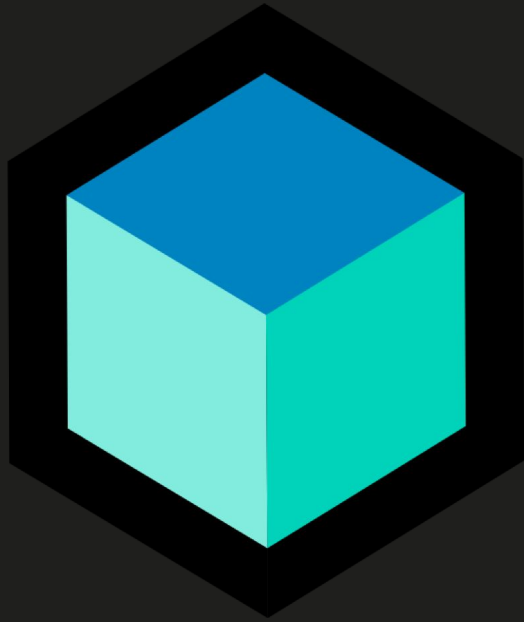
Atendente

Pessoa

MVC



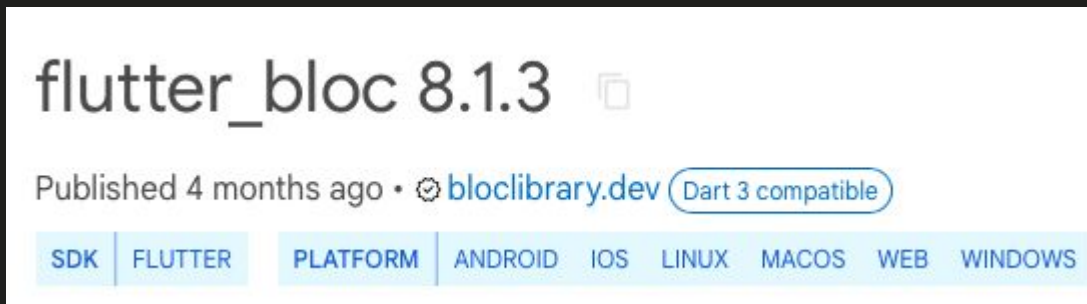
BLOC



bloc

BLOC

É um outro tipo de padrão de projetos, para que ele funcione, será necessário baixar a dependência `flutter_bloc`:



https://pub.dev/packages/flutter_bloc/install

BLOC

- Bloc: Regra de negócios;
- Event: Cria os eventos;
- State: Estado da aplicação/objeto.

BLOC

Vantagens

- Trabalha com padrão reativo;
- Projeto bem organizado;
- Reutilização das regras de negócios.

BLOC

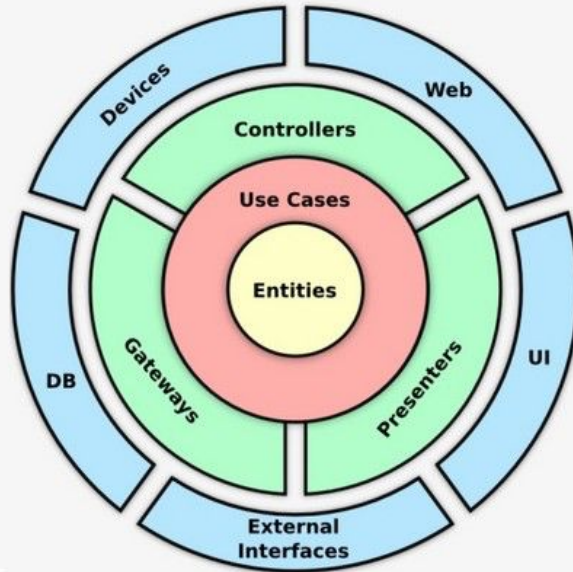
Desvantagens

- Curva de aprendizagem elevada;
- Não é recomendado para projetos de pequeno porte.

BLOC



CLEAN ARCHITECTURE



CLEAN ARCHITECTURE

Clean Architecture é uma abordagem arquitetural que define como organizar o código em camadas e componentes para garantir que as regras de negócios sejam isoladas.

CLEAN ARCHITECTURE

- Model: Modelo;
- Repository: Repositório;
- Data Repository: Implementação dos métodos na camada de repositório.

CLEAN ARCHITECTURE

Vantagens

- Projetos bem estruturados;
- Reutilização;
- Ideal para trabalhar com projetos em diversos portes: pequenos, médias ou grandes.

CLEAN ARCHITECTURE

Desvantagens

- Curva de aprendizagem elevada;
- Necessita de muito planeamento;
- Overhead de Abstração: A criação de muitas camadas e abstrações pode introduzir um overhead de abstração.

CLEAN ARCHITECTURE



Outras arquiteturas

- Redux;
- MVVM.

Referências

- <https://medium.com/follow-flutter/flutter-mvc-at-last-275a0dc1e730>
- <https://rohaaan.hashnode.dev/mastering-the-mvc-pattern-in-flutter-a-guide-to-better-code-organization>
- <https://medium.flutterdevs.com/design-patterns-in-flutter-part-1-c32a3ddb00e2>
- <https://bloclibrary.dev/>
- <https://medium.com/flutter-community/flutter-bloc-for-beginners-839e22adb9f5>
- <https://www.solutelabs.com/blog/flutter-bloc-tutorial>
- <https://dev.to/marwamejri/flutter-clean-architecture-1-an-overview-project-structure-4bhf>
- <https://medium.com/ruangguru/an-introduction-to-flutter-clean-architecture-ae00154001b0>
- <https://mobileappcircular.com/clean-architecture-in-flutter-a-quick-beginners-guide-844e89174655>

Responsividade

- MediaQuery;
- LayoutBuilder;
- Flutter Screen Util (https://pub.dev/packages/flutter_screenutil)

Responsividade - LayoutBuilder

```
LayoutBuilder(  
  builder: (BuildContext context, BoxConstraints constraints) {  
    if (constraints.maxWidth > 600) {  
      // Faça algo para telas maiores  
    } else {  
      // Faça algo para telas menores  
    }  
  },  
)
```

Links para estudos

- MediaQuery

- <https://www.boltuix.com/2023/05/flutter-media-query-different-examples.html>
- <https://www.section.io/engineering-education/responsive-flutter-design-using-mediaquery-class/>
- <https://www.geeksforgeeks.org/flutter-managing-the-mediaquery-object/>

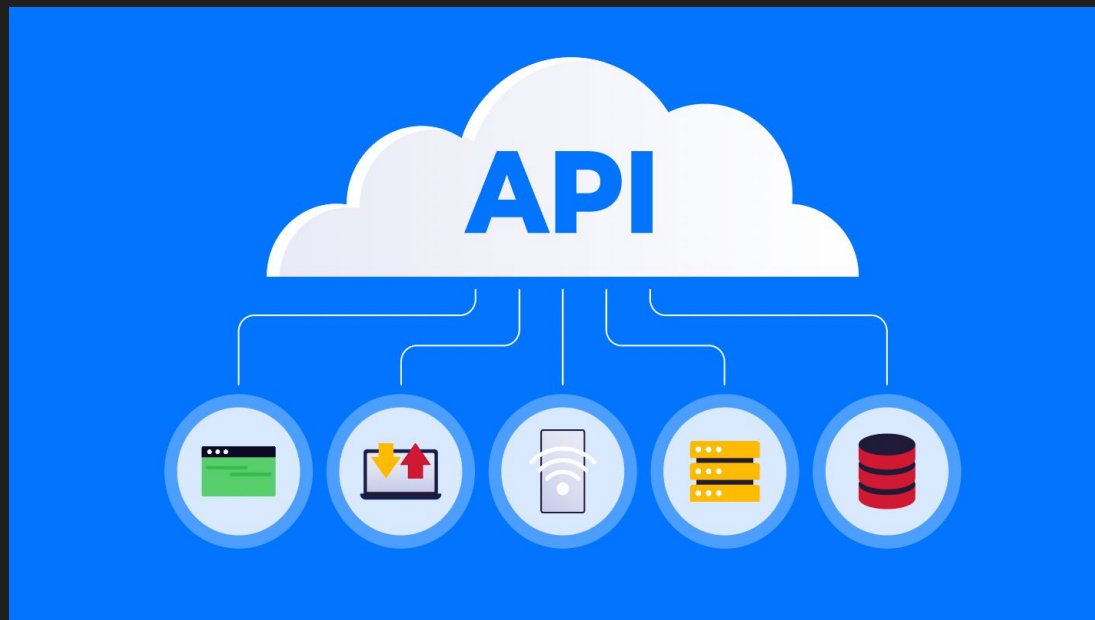
- LayoutBuilder

- <https://medium.com/@lucased78/como-usar-o-layoutbuilder-para-ajudar-na-responsividade-3ba3d8a95608>
- <https://www.geeksforgeeks.org/flutter-layoutbuilder-widget/>
- <https://www.kindacode.com/article/flutter-layoutbuilder/>

- Screen Util:

- <https://medium.com/@zhuoyuan93/flutter-screenutil-flutter-screen-adaptation-font-adaptation-get-screen-information-a51684f3346d>

APIs



Links para estudos sobre APIs

- <https://www.javatpoint.com/flutter-rest-api>
- https://www.tutorialspoint.com/flutter/flutter_accessing_rest_api.htm
- <https://blog.codemagic.io/rest-api-in-flutter/>
- <https://www.geeksforgeeks.org/implementing-rest-api-in-flutter/>
- <https://hackernoon.com/flutter-tutorial-how-to-integrate-apis-with-a-state-management-approach>
- <https://medium.com/google-developer-experts/working-with-rest-apis-flutter-f3c8c673f1dc>

Tasks (Tarefas)



Utilizem o SCRUM

- Dividam os projetos (sprints);
- Utilizem o Trello (ou outro software) para fazer a gestão;
- Reuniões periódicas (a famosa daily);
- Definam prazos de entrega para cada sprint;
- Ao término da sprint, documente e envie para toda equipe;
- Antes de continuar a fazer as sprints pendentes, ajude os colegas.

SCRUM

