Documentação App Minhas Férias

Rafael Leal

19/12/2023

Sumário

Capítulo 1

Objetivo

O App foi criado com o propósito de registrar eventos agendados nas férias para organização. Ele pode ser encontrado nesse link: github

Funcionalidades:

1. Registrar eventos com nome, endereço, dia e hora **(Em andamento)**
2. Registrar companheiros de viagem **(A fazer)**
3. Registrar controle de gastos e classificar os tipos **(A fazer)**
4. Criar linha do tempo dos eventos **(A fazer)**

Capítulo 2

Arquitetura

Para construção do projeto foram usados os princípios de arquitetura limpa seguindo principalmente as referências [1] e [2].

2.1 SOLID

Também se usam os princípios do acrônimo **SOLID** [3].

- **S**ingle Responsibility Principle
- **O**pen-Closed Principle
- **L**iskov Substitution Principle
- **I**nterface Segregation Principle
- **D**ependency Inversion Principle

2.1.1 Single Responsibility Principle

2.1.2 Open-Closed Principle

2.1.3 Liskov Substitution Principle

2.1.4 Interface Segregation Principle

2.1.5 Dependency Inversion Principle

2.2 Módulos

Foram criados os módulos para melhor organizar cada componente de forma mais desacoplada possível facilitando a manutenção e a testabilidade.

Módulos:

- app

- domain
- data-repository
- data-local
- presentation-commom
- presentation-registered-events

2.2.1 app

Módulo principal do App que depende de todos os demais.

Injeções:

Nesse módulo reside a classe MyApplication: Application() necessára para usar a injeção de dependência com Hilt.

Contem a injeção do UseCase.Configuration usado para definir o Dispatcher do UseCase.

Essa injeção é necessária pois facilitar controlar o contexto de corrotinas nos testes de unidade

Testes:

Contem os seguintes testes:

- de interface
- migração do banco de dados

2.2.2 domain

Esse é o módulo mais estável de todo projeto. Nele é feita a parte de lógica essencial para o funcionamento do aplicativo e portanto não deve sofrer alterações a menos que seja extremamente necessário, seguindo o princípio do **SOLID** Open-Closed Principle 2.1.2

2.2.3 data-repository

2.2.4 data-local

2.2.5 presentation-commom

2.2.6 presentation-registered-events

Capítulo 3

Bibliotecas Externas

Para melhorar a performance são usadas algumas bibliotecas externas.

Elas são importadas no arquivo build.gradle do projeto e usadas como referência para cada módulo. Dessa forma não precisamos atualizar as versões em cada módulo manualmente.

3.1 Injeção

Para injeção de dependência usamos

- Hilt.

Referências Bibliográficas

[1] Alexandru Dumbravan and Ed Price. *Clean Android Architecture: Take a layered approach to writing clean, testable, and decoupled Android applications.* Packt Publishing Ltd, 2022.

[2] Eran Boudjnah. *Clean Architecture for Android: Implement Expert-led Design Patterns to Build Scalable, Maintainable, and Testable Android Apps (English Edition).* BPB Publications, 2022.

[3] C Robert. Clean architecture: A craftsman's guide to software structure and design (robert c. martin series), 2019.