Arquitetura de Front-End

PADRÃO ARQUITETÔNICO MODEL-VIEW-PRESENTER

A plataforma Android não advoga por um tipo de arquitetura ou organização específica para desenvolver um aplicativo. O desenvolvedor é livre para utilizar padrões como **MVC** (*Model-View-Controller*), **MVVM** (*Model-View-ViewModel*) ou **MVP**(*Model-View-Presenter*) ou até mesmo não utilizar nenhuma arquitetura.

Nesse contexto, utilizar o padrão MVP significa seguir princípios que permitem a separação entre a camada de apresentação e a camada de dados da aplicação, proporcionando o código ser reutilizável e testável de maneira simples. O MVP divide os componentes da arquitetura baseado em papéis, seguindo o princípio da **separação de conceitos**(

https://en.wikipedia.org/wiki/Separation_of_concerns).

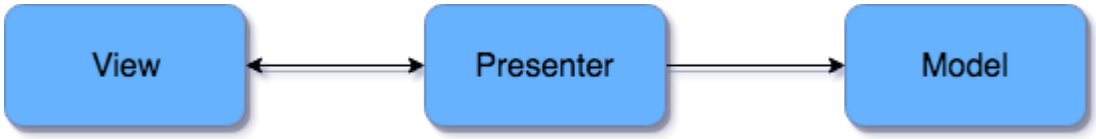
O padrão MVP divide a aplicação em 3 componentes básicos:

1. **Model**: responsável por lidar com a camada de dados da sua aplicação.
2. **View**: responsável por apresentar os componentes de interface.
3. **Presenter**: responsável por fazer a conexão entre **Model** e **View** (o *middle-man* entre os dois).

O padrão MVP também define algumas regras a serem seguidas:

1. A única responsabilidade da **View** é mostrar os componentes de interface com seus respectivos valores na tela, de acordo com os dados enviados pelo **Presenter**. A **View** é portanto, totalmente passiva.
2. A **View** delega todas as ações provenientes de interações do usuário para o **Presenter**.
3. A **View** não se comunica diretamente com o **Model** e vice-versa.
4. Toda a comunicação entre **Presenter** e **View** é feita através de interfaces.
5. O **Presenter** é responsável por delegar requisições da **View** para o **Model** e também por instruir a **View** com ações para eventos específicos.
6. O **Model** é responsável por lidar com requisições de dados de servidor, banco de dados e sistema de arquivos. O **Model**pode inclusive ser uma interface que se comunica com outros módulos que realizam essas funções.

A figura a seguir mostra as dependências entre cada componente do padrão MVP:



Links úteis (sobre implementação do MVP no Android):

- https://blog.mindorks.com/essential-guide-for-designing-your-android-app-architecture-mvp-part-1-74efaf1cda40
- https://medium.com/@cervonefrancesco/model-view-presenter-android-guidelines-94970b430ddf
- https://antonioleiva.com/mvp-android/
- http://www.tinmegali.com/en/model-view-presenter-android-part-1/

ANDROID BÁSICO

- **RecyclerView**
  - https://guides.codepath.com/android/using-the-recyclerview
  - https://antonioleiva.com/recyclerview/
  - https://developer.android.com/guide/topics/ui/layout/recyclerview.html
- **Intents**
  - http://www.vogella.com/tutorials/AndroidIntent/article.html
  - https://developer.android.com/guide/components/intents-filters.html
- **Criação de interfaces de usuário no Android:**
  - https://br.udacity.com/course/android-development-for-beginners--ud837 (Curso gratuito)

CHAMANDO A API RESTFUL

Utilizar a biblioteca FastAndroidNetworking. Alguns exemplos de uso estão disponíveis neste link.

Revision #2
Created Wed, Oct 11, 2017 2:45 PM by Marcus Sales
Updated Wed, Oct 11, 2017 7:51 PM by Isaac Pereira