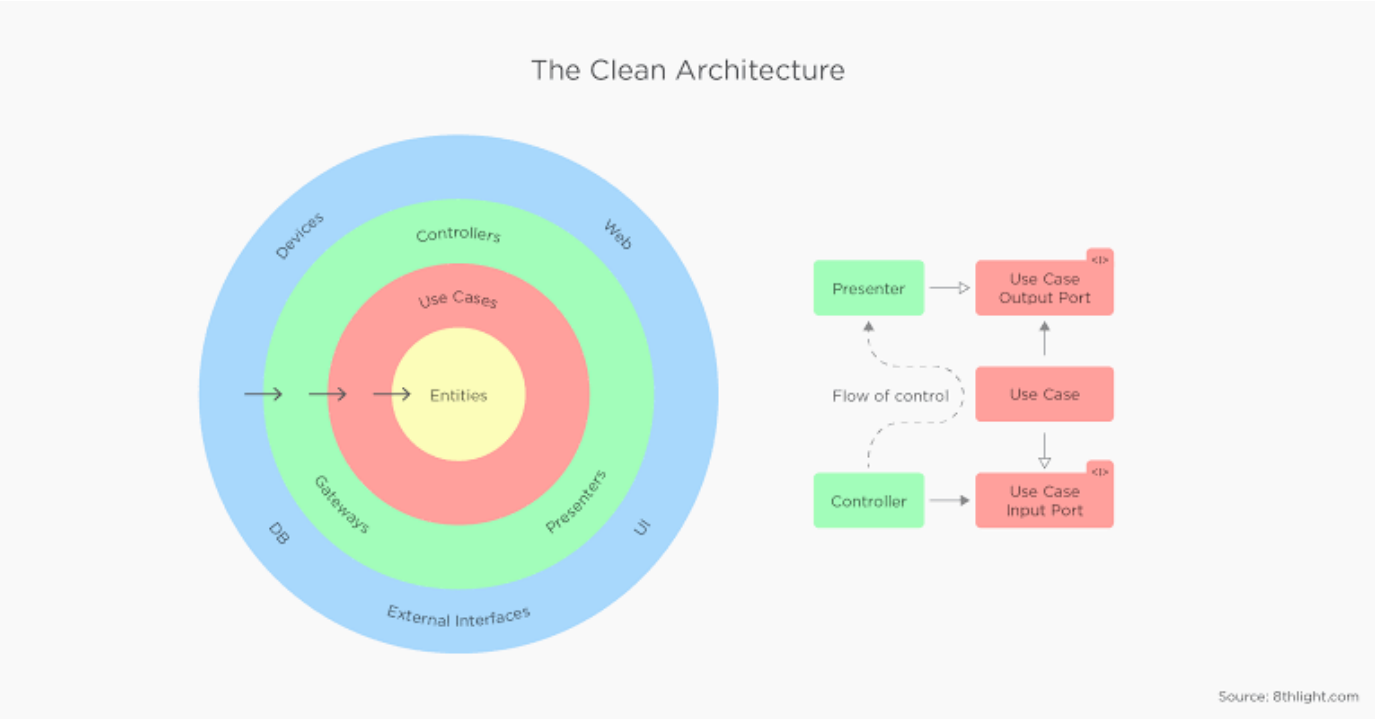
Agile | Engenheiro de Software Mobile | Cyber Security | Big Data

Clean Architecture com MVVM em aplicações Android

09-11-2020 por ederson

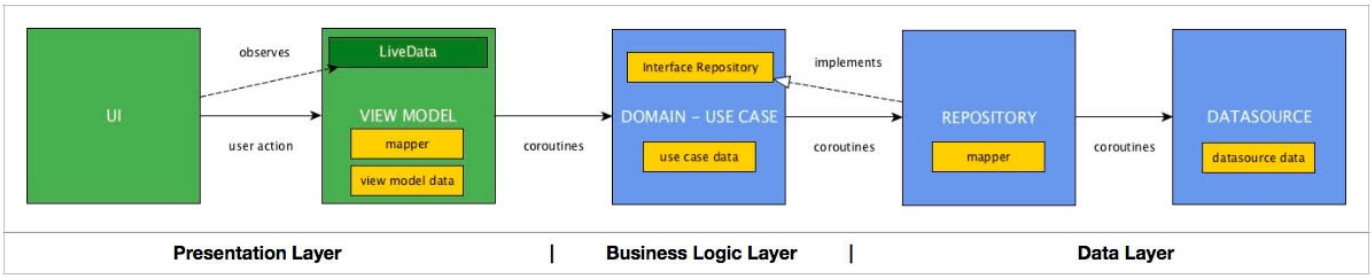
A Clean Architecture foi idealizada por Robert C. Martin, autor de um livro abordando este tema, ajudou a criar uma arquitetura em que os componentes fossem desacoplados, testáveis e de fácil manutenção.

O que é a Clean Architecture? A Clean Architecture consiste em um diagrama de camadas, em que cada um dos seus componentes possuem suas próprias responsabilidades e cada uma delas tem conhecimento apenas de camadas de dentro, ou seja, a camada de “Frameworks e Drivers” enxerga somente a de “Interface Adapters”.



Vantagens: O código é facilmente testável, se comparado a arquitetura MVVM simples; Componentes ainda mais desacoplados, a estrutura do pacote é facilmente de se navegar entre eles; Novas funcionalidades podem ser adicionadas rapidamente pelo time de desenvolvedores.

Desvantagens: Curva de aprendizado relativamente íngreme, considerando que todas as camadas funcionam juntas, exigindo um certo tempo para entender seus conceitos, principalmente desenvolvedores provenientes de padrões como MVVM e MVP simples; Pela arquitetura exigir o acréscimo de muitas classes adicionais, este modelo não é ideal para projetos de baixa complexidade. A intenção é de demonstrar as regras de dependência dentro da arquitetura.



Cada camada de MVVM usando Clean Architecture no Android e os códigos se dividem em três camadas:

Camada de Apresentação (Presentation Layer):

Nesta camada, são incluídas as “Activity”s, “Fragment”s como sendo as “Views”, e as “ViewModel”s, devem ser mais simples e intuitivo possível e ainda, não devem ser incluídas regras de negócio nas “Activity”s e “Fragment”s.

Uma “View” irá se comunicar com sua respectiva “ViewModel”, e assim, a “ViewModel” se comunicará com a camada de domínio para que sejam executadas determinadas ações. Uma “ViewModel” nunca se comunicará diretamente com a camada de dados;

Camada de Domínio (Domain Layer):

Na camada de domínio, devem conter todos os casos de uso da aplicação. Os casos de uso tem como objetivo serem mediadores entre suas “ViewModel”s e os “Repository”s.

Caso for necessário adicionar uma nova funcionalidade, tudo o que deve ser feito é adicionar um novo caso de uso e todo seu respectivo código estará completamente separado e desacoplado dos outros casos de uso. A criação de um novo caso de uso é justamente para evitar que ao adicionar novas funcionalidades, quebrar as preexistentes no código;

Camada de Dados (Data Layer):

Esta camada possui todos os repositórios que a camada de domínio tem disponíveis para utilização e “DataSource”s, que são responsáveis por decidir qual a fonte em que devem ser recuperados os dados, sendo de uma base de dados local ou servidor remoto.

Tags: Clean Architecture, Arquitetura, Android, mvc, mvp, mvvm