**Principais mudanças**

A principal e mais notória mudança que eu faria ao implementar o clean Architecture é a modularização das classes principalmente na Main Activity e Details Activity. Nelas existe muita lógica que deveria ser implementada na viewModel e seguindo a mesma lógica de modularização, separar a viewModel que recebe duas chamadas de APIs distintas, em duas viewModels diferentes. E talvez a que mais impacta no tempo de execução da UI é a API gêneros chamada desnecessariamente várias vezes e as respostas do servidor serem as mesmas, podem ser salvas tranquilamente em uma const ou em um SQL.

Arquitetura atual: MVVM + Clean Architecture:

Tela de computador

Descrição gerada automaticamente ↓data

↓local

LocalDataSource(room\_films)

↓remote

RemoteDataSource (client)

↓Repository

Todos os repositorys

↓domain

→ utils

↓useCases

UseCaseFilms

↓model

FilmsResponse

DetailsResponse

...

↓presentation

↓ui

Activities/Fragments/Adapters

↓viewmodel

ViewModels

**Possiveis Implementações jetpack**

Acho que a principal implementação seria a room, pois o tempo de resposta da UI para o usuário é menor se comparada com uma chamda de API.

Outra implementação interessante é o Paging para otimização de dados da recyclerView.

O AppSearch + AutoFill para otimizar e melhorar a experiencia do usuário ao pesquisar um filme específico no banco de dados.

E particularmente gostaria de implementar o novíssimo Material3 e toda a gama de widgets interessantes que ele proporciona para a UI