Relatório do trabalho da disciplina de Desenvolvimento de Jogos para Plataformas Móveis

W4tchList

Nicolae Malai - 23495

Engenharia em Desenvolvimento de Jogos Digitais

Janeiro de 2023



# Índice

[Índice 2](#_Toc124393654)

[Índice de Figuras 3](#_Toc124393655)

[Estrutura do projeto 4](#_Toc124393656)

[Lista de funcionalidade da aplicação 5](#_Toc124393657)

[Desenhos, esquemas e protótipos da aplicação 6](#_Toc124393658)

[Modelo de dados 7](#_Toc124393659)

[Implementação do projeto 9](#_Toc124393660)

[Tecnologias usadas 10](#_Toc124393661)

[Dificuldades 11](#_Toc124393662)

[Conclusão 12](#_Toc124393663)

# Índice de Figuras

[Figura 1 - Data Class e Database Organization 6](#_Toc124394501)

[Figura 2 - Layout Sketches and Todo 6](#_Toc124394502)

[Figura 3 - User Data Class 7](#_Toc124394503)

[Figura 4 - MovieList Data Class 7](#_Toc124394504)

[Figura 5 - Movie Data Class 8](#_Toc124394505)

[Figura 6 - MovieApiService 9](#_Toc124394506)

[Figura 7 - MovieInterface 10](#_Toc124394507)

# Estrutura do projeto

O objetivo inovador por trás desta aplicação é dar a oportunidade ao utilizador de criar uma conta, criar listas de filmes e adicionar os mesmos e tendo ainda a oportunidade de partilhar essa lista editável com outras pessoas.

Como primeira interação com o projeto pensei em precisar de muito menos layouts e trabalho do que o que acabei por ter. Para comecar, os layouts necessarios seriam de login, register, homescreen e pesquisa.

Em projetos prévios tive interesse em experimentar utilizar a API da TMDB mas acabei por desistir da ideia após algum tempo investido sem sucesso. Quando estava a pensar em uma ideia para este projeto deparei-me com um vídeo que me demonstrava exatamente o suficiente para conseguir aprender o básico da API e ter as bases para o resto do projeto então optei por dar mais uma chance.

Para tratar do armazenamento de dados utilizei o Firebase realtime em função síncrona com o Firebase authenticator para realizar o Login e Registo de utilizadores, assim como guardar o seu email, username, foto de perfil, password e também para fornecer com o UID para a base de dados.

Para além desta ligação entre o utilizador com a base de dados, também guardo valores para as listas de todos os utilizadores, dentro destas estão também guardadas as informações que achei necessárias de cada filme, para evitar fazer pedidos desnecessários à API e ter um melhor tempo de resposta para o utilizador.

# Lista de funcionalidade da aplicação

* Registo
* Login
* Logout
* Home page com trending movies
* Search page para pesquisa de filme por título
* Genre page para pesquisa de filme por género
* Account page com ações possíveis ao utilizador, como mudar nome, email, foto etc.
* List page com a opção de criar novas listas e adicionar filmes, assim como partilhar com outro utilizador

# Desenhos, esquemas e protótipos da aplicação

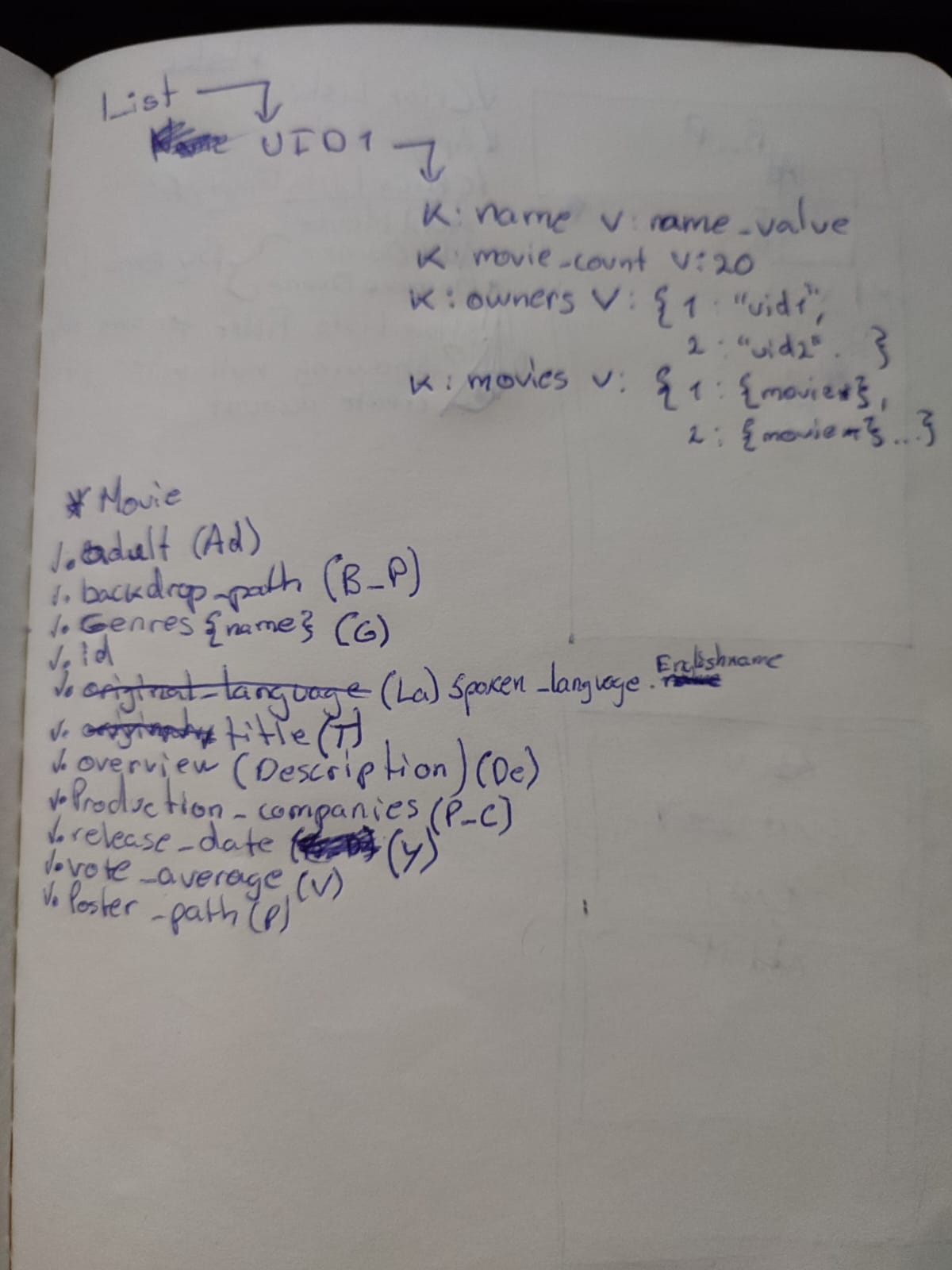


Figura 1 - Data Class e Database Organization

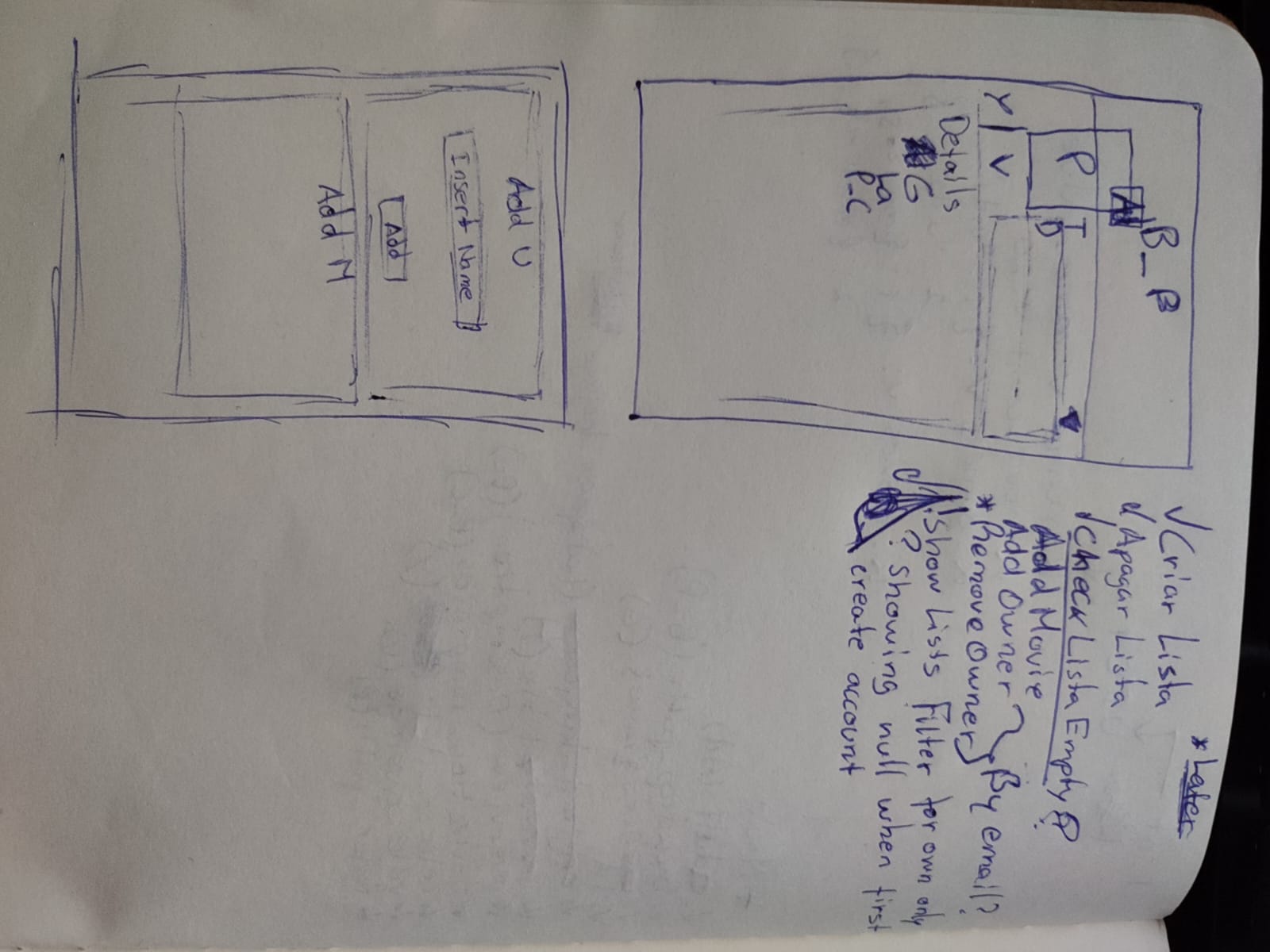


Figura 2 - Layout Sketches and Todo

# Modelo de dados

Para conseguir fazer a ligação entre o authenticator e a base de dados, criei uma classe para o User, esta é constituida por um campo email, uma List<String> com os id’s das listas que o utilizador tem e um boolean adult, utilizado para filtrar conteúdo adulto.

Texto

Descrição gerada automaticamente

Figura 3 - User Data Class

Para guardar as listas, criei uma classe que é constituída pelo nome, movie\_count e uma List<Movie>

Texto

Descrição gerada automaticamente

Figura 4 - MovieList Data Class

Para cada filme, o movie data class faz o trabalho todo, todos os campos estão filtrados para conseguir pegar os valores do resultado da API.

Texto

Descrição gerada automaticamente

Figura - Movie Data Class

# Implementação do projeto

Texto

Descrição gerada automaticamente

Figura - MovieApiService

Texto

Descrição gerada automaticamente

Figura - MovieInterface

# Tecnologias usadas

* Android Studio: IDE utilizado para o desenvolvimento.
* TMDB (<https://www.themoviedb.org/>): API utilizada.
* YouTube (<https://www.youtube.com>): Utilizado para pesquisa e ajuda visual.
* Stackoverflow (<https://stackoverflow.com/>): Utilizado para pesquisa e ajuda escrita.
* OpenAI ChatGPT (<https://openai.com/blog/chatgpt/>): AI utilizado para ajuda em diversas etapas.
* Github: Utilizado para Version Control

# Dificuldades

A maior dificuldade foi perceber as várias componentes mais complexas de kotlin e melhorar o conhecimento neste tipo de programação.

No início do projeto a ideia era utilizar a API para fazer todas as transações das listas, porém após varios dias perdidos a tentar encontrar qualquer workaround e resolver todos os problemas para conseguir implementar essa função, fui forçado a tomar outro rumo devido à falta de tempo e de não funcionamento da scheme.

Este problema obrigou-me a aumentar a complexidade da minha app a níveis muito maiores do que os que eu tinha previsto. Mesmo com todas as dificuldades consegui atingir o objetivo que tinha em mente e estou muito satisfeito com o resultado.

# Conclusão

Este projeto fez-me perceber o meu gosto por desenvolvimento de app’s assim como aumentar o meu conhecimento na àrea em grande quantidade.

Após a conclusão do mesmo, sinto-me preparado e capaz de fazer muitas aplicações com muita mais velocidade, eficácia e código melhor.