

Usuário do GitHub: alandvgarcia

Pop Movies

Descrição

O PopMovies vai permitir que você possa consultar análises e classificação dos filmes. Caso esteja procurando um filme para assistir o PopMovies irá lhe informar as tendências que estão passando nas telas dos cinemas.

Público-Alvo/Intended User

Para o público que assiste séries e filmes em geral.

Funcionalidades/Features

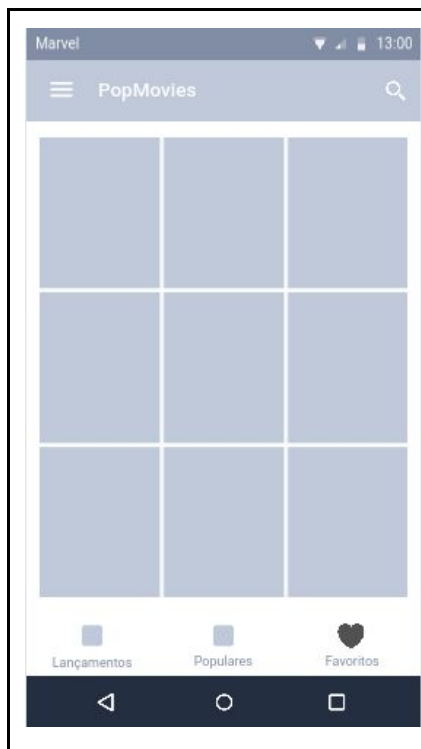
Liste as principais funcionalidades do seu app. Por exemplo:

- Salvar os favoritos do usuário.
- Compartilhar conteúdo.

Protótipo de Interfaces do Usuário

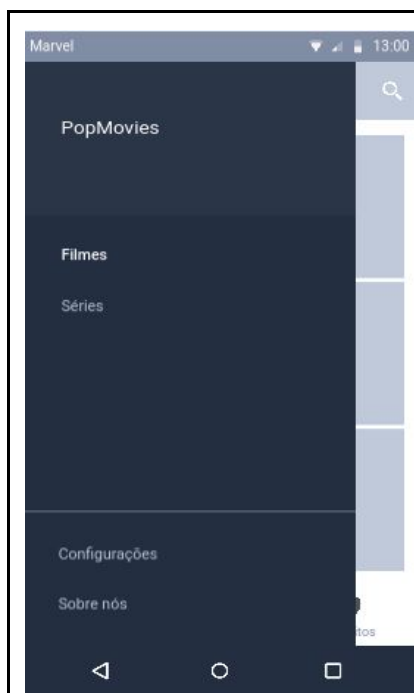
Elas podem ser feitas a mão (tire uma foto dos seus desenhos e os insira neste fluxo), ou usando um programa como o Photoshop ou Balsamiq.

Tela 1 - Tela Principal



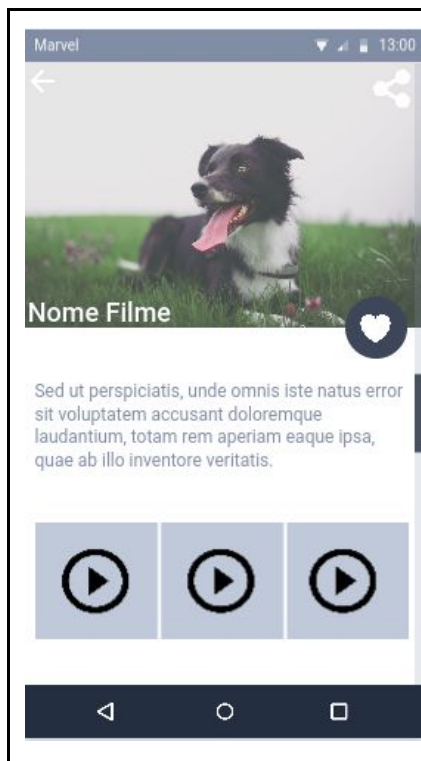
Essa será a tela principal do aplicativo, onde inicialmente o aplicativo irá iniciar mostrando os filmes que estão em lançamento. A tela terá funcionalidades como, buscar os filmes, selecionar abaixo a categoria por Lançamentos, Populares e Favoritos. Haverá um “hamburguer button” onde irá abrir um menu para selecionar o gênero do filme desejado.

Tela 1 - Hamburger Button



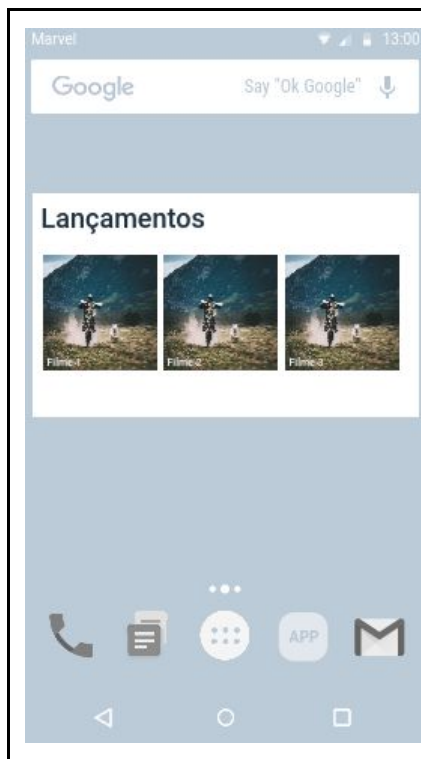
Ainda na tela principal, no hamburger button terá gêneros para que o usuário possa filtrar. Haverá também um menu para configurações onde pretendo colocar algo que seja personalizado para o usuário.

Tela 3 - Detalhes do filme



Aqui é onde terá os detalhes do filmes, penso em colocar mais recursos além da descrição como a avaliação, compartilhar e favoritar.

Widget



O Widget irá mostrar a miniatura dos filmes que serão o os 3 com maior pontuação na categoria selecionada, sendo as lançamentos, populares e favoritos, isso será da preferência do usuário.

Essas são apenas as telas principais, mas sem mudar a essência do aplicativo poderão ser alteradas conforme o desenvolvimento e implementação de novas funcionalidades conforme a necessidade.

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

Terá persistência de dados local com SQLite, mas pretendo também implementar login para usuário utilizando o firebase, onde futuramente eu irei salvar os dados e os filmes favoritos do usuário, como irei utilizar MVVM o Model serão as entidades que serão criadas na persistência.

A entidade principal será:

Movie

ID
ORIGINAL_LANGUAGE
ORIGINAL_TITLE
TITLE
OVERVIEW
RELEASE_DATE
STATUS
VOTE_AVERAGE
VOTE_COUNT
POPULARITY

Acredito que conforme a necessidade e implementação de novas funcionalidades terei novas entidades, mas para o essencial da aplicação é a entidade *movie*.

Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX).

Ao entrar no aplicativo, a tela inicial será os filmes onde o usuário pode selecionar as opções no tab inferior da tela, onde ele pode visualizar os lançamentos, populares e os favoritos.

Ao selecionar um filme vai abrir a tela de detalhes do filme, onde o usuário vai poder ver a descrição, trailer e avaliação e também favoritar.

Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.

<https://github.com/koush/ion> - Para requisitar dados da API, consumindo JSON e imagens.

<https://github.com/ReactiveX/RxAndroid> - RxAndroid: Reactive Extensions for Android

<https://developer.android.com/topic/libraries/data-binding/> - Data binding

<https://developer.android.com/topic/libraries/architecture/viewmodel> - ViewModel

As bibliotecas acima irei utilizar pois pretendo usar a arquitetura MVVM.

Descreva como você implementará o Google Play Services.

Irei utilizar o firebase para salvar os filmes que o usuário favoritar, irá persistir local no dispositivo, mas também pretendo utilizar o firebase para que o usuário tenha acesso aos seus dados de qualquer dispositivo que ele logar.

Também irei implementar o google maps onde irei mostrar os cinemas que estão próximo ao usuário.

Tarefa 1: Configuração do Projeto/Project Setup

Inicial irei configurar as bibliotecas e vou definir as packages para que eu comece implementando o MVVM.

Tarefa 2: Configuração API

Nessa etapa irei configurar as classes para consumo da API.

Tarefa 3: Models e ViewModels

Irei escrever os models da aplicação que serão consumidos pelo ViewModel.

Tarefa 4: Views/UI

Implementação da interface onde irei configurar o Data Binding com a ViewModel.

