

MovieRoll

Sprint 2

Objetivo da sprint

O objetivo desta sprint é o desenvolvimento de todas as telas e navegação do app com dados mockados.

Funcionalidades implementadas na sprint

Com base no protótipo feito no site (<https://marvelapp.com>), implementamos o app usando o ambiente de desenvolvimento Xcode.

Usamos no app: Navigation Controller e TabBarController.

Implementamos as telas:

1. **Tela LaunchScreen:**

- Contendo: logo do app.

2. **Tela Login:**

- Contendo os campos: e-mail e senha.
- Contendo os botões: entrar; login pelo Facebook ou Google; e recuperar senha.

3. **Lançamentos:**

- Contendo: uma collectionView dentro de uma célula customizada da TableView, onde há várias listas (collectionView) de filmes na horizontal organizadas por gêneros (Ação, Aventura, Comédia e entre outros).

4. **Detalhes de Filme:**

- Contendo: imagem e diversas descrições de um filme selecionado na CollectionView da Tela Lançamentos ou de um filme proveniente da ação do botão roletar da Tela Roleta.

5. **Roleta:**

- Contendo:
- Uma lista (Buttons) de gêneros (Ação, Aventura, Comédia e entre outros) para selecionar.
- Cinco botões de estrelas para selecionar de acordo com o nível de exigência do usuário.
- Um intervalo da data de lançamento para selecionar no UIPickerView.
- Uma lista (CollectionView) de plataformas de Streaming (Netflix, Apple TV+, Amazon Prime Video e entre outras) para selecionar.
- Um botão roletar que filtra um filme de acordo com o que o usuário selecionou nos campos acima.

6. **Perfil:**

- Contendo:
- Tela com dados do usuário (imagem, nome e e-mail) em uma TableView.
- Tela com o histórico de filmes roletados, favoritos e assistidos usando o Segmented Control e uma CollectionView.
- Tela de configurações com três botões que permitem limpar o histórico de filmes roletados, favoritos e assistidos; e um botão pro usuário desconectar da conta.
- Tela Modal de ajuda com informações sobre o app e de seus desenvolvedores.

Tarefas realizadas por cada aluno na sprint

• **Artur:**

- Implementou a tela Roleta e a CollectionView de plataformas.
- Refatorou a tela Lançamentos e os dados mocados.

• **Bruno:**

- Realizou Auto Layout da tela Detalhes de Filme e refatorou a tela Lançamentos.
- Inseriu ícones no projeto e criou alertas na tela Roleta.
- Implementou cliques nas plataformas da tela Roleta.

• **Raul:**

- Realizou Auto Layout e implementou a tela de Lançamentos.
- Implementou o UIPickerView na tela Roleta.
- Implementou a tela de Login.
- Implementou botões de limpar histórico na tela Perfil em configurações.

- **Renato:**
 - Implementou e refatorou todas as funcionalidades da tela Perfil.
 - Implementou a tela Roleta e Lançamentos.
 - Refatorou os filtros das telas: Lançamentos, Roleta e Perfil.
 - Realizou Auto Layout e Refatorou as ViewControllers, ViewModels e Service das telas Lançamentos, Roleta e Perfil.
 - Refatorou o código seguindo o padrão MVVM.
- **Vitor:**
 - Criou o repositório do projeto e atualizou o README.md no Github.
 - Refatorou a tela de Detalhes de filme.
- **Ygor:**
 - Implementou o filtro do botão roletar na tela Roleta.
 - Implementou a função de filtrar por gêneros na tela Roleta.
 - Adicionou imagens dos filmes ao projeto e fez Auto Layout da tela Detalhes de filme.

Dificuldades Encontradas

Nossa primeira dificuldade foi aplicar logo a arquitetura MVVM, onde “codamos” algumas telas que até ficaram funcionais, porém tivemos que reavaliar o código para manter na arquitetura exigida. Após refatoração, encontramos dificuldade em consertar alguns bugs do app como seleção de filtros não selecionados para a roleta. Outra dificuldade que tivemos foi de realizarmos a tratativa dinâmica que precisamos no app: não roletar o mesmo filme duas vezes (a não ser que o usuário limpe o histórico de filmes roletados com o intuito de reiniciar sua roleta).

Objetivo da próxima sprint

O objetivo da próxima sprint é aplicar API no projeto para termos mais filmes, testando mais filtros, principalmente quanto à nota do filme (filtrado por estrelas) e às plataformas das mídias, uma vez que os dados mocados nos limitam muito a aplicação de testes.

Descrição do backlog da próxima sprint

- Tarefa de API: Para a próxima Sprint, retiraremos os dados mocados e implementaremos a API. Converteremos dados para Swift. Trataremos a Interface de Programação de Aplicação de acordo com as necessidades do projeto.

- Tarefa de refatoração: Refatoração do código para melhoria estética e funcional. Adição de comentários (//) e marcadores (// MARK: -) para melhor leitura e demarcação do código.

Tarefas que serão realizadas por cada aluno na próxima sprint

Vitor, Ygor e Bruno: aplicação da API conforme informações necessárias para o app;
Renato, Artur e Raul: refatoração do código;