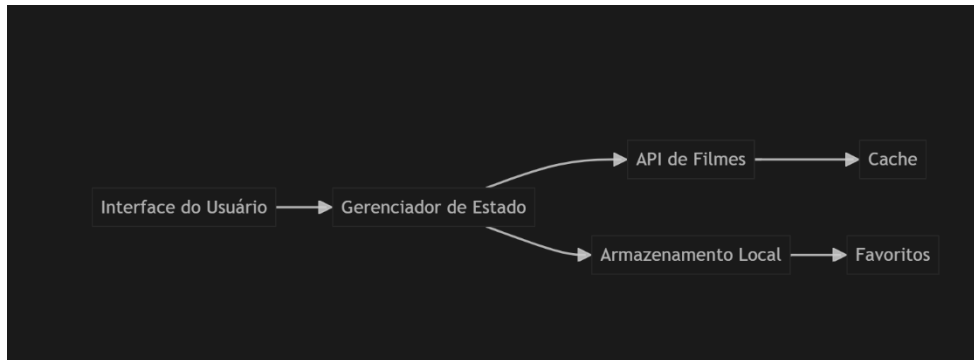


# Documentação EasyMovie

## Visão Geral

EasyMovie é um aplicativo móvel desenvolvido em React Native usando o framework Expo. É uma plataforma de streaming que permite aos usuários explorar, pesquisar e favoritar filmes, oferecendo uma interface moderna e intuitiva.

## Estrutura do Projeto



## Tecnologias Principais

- React Native: Framework principal para desenvolvimento mobile
- Expo: Plataforma de desenvolvimento
- React Navigation: Sistema de navegação
- React Native Paper: Biblioteca de componentes UI
- Axios: Cliente HTTP para requisições à API
- AsyncStorage: Armazenamento local
- Expo Linear Gradient: Efeitos de gradiente
- React Native Responsive Dimensions: Dimensionamento responsivo

## Funcionalidades Principais

### 1. Autenticação

- Sistema completo de login e registro
- Proteção de rotas
- Gerenciamento de sessão
- Logout seguro

### 2. Navegação

O aplicativo utiliza um sistema de navegação híbrido com:

Stack Navigator: Para navegação principal entre telas

Tab Navigator: Para navegação entre as principais seções do app

### 3. Telas Principais

- Home Screen
  - Exibição de filmes populares
  - Filmes mais bem avaliados
  - Interface com cards responsivos
  - Acesso rápido aos detalhes dos filmes.
- Favorites Screen
  - Lista de filmes favoritos do usuário
  - Layout em grid com 3 colunas
  - Gerenciamento de favoritos (adicionar/remover)
  - Estado vazio personalizado
- Search Screen
  - Busca de filmes
  - Resultados em tempo real
  - Interface otimizada para pesquisa
  - Movie Detail Screen
  - Informações detalhadas do filme
  - Opção de favoritar/desfavoritar
  - Navegação integrada

### 4. Gerenciamento de Estado

O aplicativo utiliza diversos contextos React para gerenciamento de estado:

AuthContext: Gerenciamento de autenticação

FavoritesContext: Controle de filmes favoritos

ThemeContext: Personalização de tema

### 5. UI/UX

- Design moderno e responsivo
- Tema escuro predominante
- Componentes personalizados
- Feedback visual para interações
- Suporte a diferentes tamanhos de tela
- 6. Segurança
- Rotas protegidas
- Validação de formulários
- Tratamento de erros
- Sessão segura

## Configurações Técnicas

## Dependências Principais

- "@react-native-async-storage/async-storage": "^2.1.2",
- "@react-navigation/bottom-tabs": "^7.3.13",
- "@react-navigation/native": "^7.1.9",
- "@react-navigation/stack": "^7.3.2",
- "axios": "^1.9.0",
- "expo": "~53.0.9",
- "react-native-paper": "^5.14.5"

## Considerações de Desenvolvimento

- O projeto segue uma arquitetura baseada em componentes
- Utiliza hooks do React para gerenciamento de estado
- Implementa context API para estado global
- Possui sistema de temas personalizável
- Utiliza dimensões responsivas para adaptação a diferentes dispositivos

## Boas Práticas Implementadas

1. **Componentização:** Componentes reutilizáveis e modulares
2. **Contextos:** Gerenciamento eficiente de estado global
3. **Proteção de Rotas:** Sistema de autenticação robusto
4. **Responsividade:** Adaptação a diferentes tamanhos de tela
5. **Feedback:** Respostas visuais para ações do usuário
6. **Performance:** Otimizações de renderização
7. **Manutenibilidade:** Código organizado e bem estruturado

## Documentações e Fontes Essenciais

### 1. Documentações Oficiais Base

#### React Native

- Site Oficial: <https://reactnative.dev/>
- Documentação: <https://reactnative.dev/docs/getting-started>
- Componentes nativos
- APIs do React Native
- Ciclo de vida dos componentes
- Performance e otimização

## Expo

- Site Oficial: <https://expo.dev/>
- Documentação: <https://docs.expo.dev/>
- Configuração inicial
- Gerenciamento de assets
- APIs nativas
- Build e deploy

## **2. Navegação**

### React Navigation

- Site Oficial: <https://reactnavigation.org/>
- Documentação: <https://reactnavigation.org/docs/getting-started>
- Stack Navigation
- Tab Navigation
- Gestão de rotas
- Navegação aninhada

## **3. UI/UX**

### React Native Paper

- Site Oficial: <https://callstack.github.io/react-native-paper/>
- Documentação: <https://callstack.github.io/react-native-paper/docs/guides/getting-started>
- Componentes pré-construídos
- Tematização
- Consistência visual
- React Native Responsive Dimensions
- NPM: <https://www.npmjs.com/package/react-native-responsive-dimensions>
- Por que usar: Ferramenta para:
- Dimensionamento responsivo
- Adaptação a diferentes tamanhos de tela

## **4. API de Filmes**

### TMDB (The Movie Database)

- Site Oficial: <https://www.themoviedb.org/>
- Documentação API: <https://developers.themoviedb.org/3>
- Catálogo de filmes
- Informações detalhadas
- Imagens e posters
- Busca e filtros

## **5. Gerenciamento de Requisições**

## Axios

- Site Oficial: <https://axios-http.com/>
- Documentação: <https://axios-http.com/docs/intro>
- Requisições à API
- Interceptadores
- Tratamento de erros

## **6. Armazenamento Local**

### AsyncStorage

- Documentação: <https://react-native-async-storage.github.io/async-storage/>
- Dados de usuário
- Favoritos
- Cache

## **7. Efeitos Visuais**

### Expo Linear Gradient

- Documentação: <https://docs.expo.dev/versions/latest/sdk/linear-gradient/>
- Gradientes
- Efeitos visuais

## **8. Recursos de Aprendizado Adicionais**

- React Native Community: <https://reactnative.dev/community/overview>
- Expo Blog: <https://blog.expo.dev/>
- React Navigation Blog: <https://reactnavigation.org/blog/>
- React Native Debugger: <https://github.com/jhen0409/react-native-debugger>
- Flipper: <https://fbflipper.com/>
- Material Design Guidelines: <https://m3.material.io/>
- <https://callstack.github.io/react-native-paper/docs/showcase>.