

Documentação

OpFlix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Alameda Barão de Limeira, 539 – São Paulo/SP |  |  |  | sp.senai.br |

Sumário

1. [Resumo 2](#_Toc533767843)

[Objetivos](#_Toc533767844)

1. [Descrição do projeto 2](#_Toc533767845)

[Resumo do projeto](#_Toc533767846)

1. [Modelagem de Software 3](#_Toc533767847)

[Modelo Lógico](#_Toc533767848)

[Modelo Físico](#_Toc533767849)

[Modelo Conceitual](#_Toc533767850)

[Cronograma](#_Toc533767851)

1. [Funcionalidades 5](#_Toc533767852)

[Web](#_Toc533767853)

[Mobile](#_Toc533767854)

1. [Protótipos 6](#_Toc533767855)

[Web](#_Toc533767856)

[Mobile](#_Toc533767857)

1. [Front-End 7](#_Toc533767858)
2. [Mobile 8](#_Toc533767859)
3. [Arquitetura do Projeto 9](#_Toc533767860)
4. [Referências 10](#_Toc533767861)

[Links](#_Toc533767862)

[Livros](#_Toc533767863)

Resumo

Objetivos

Este documento tem como objetivo demonstrar todas as funcionalidades da plataforma OpFlix, que visa demonstrar como funciona um app de Lançamentos.

Descrição do projeto

O OpFlix tem como objetivo mostrar os maiores lançamentos de 2019 e os melhores filmes/séries disponíveis.

Resumo do projeto

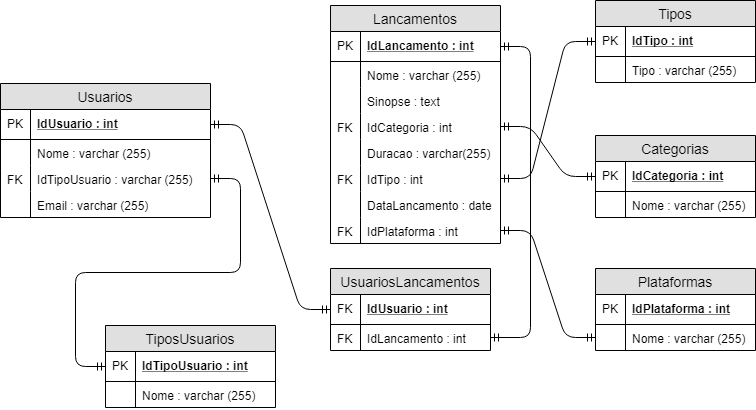
O OpFlix tem objetivo de mostrar quais são os filmes/séries disponíveis para o entretenimento de nossos clientes, mostrando quais seus gêneros/categorias, as plataformas em que estão disponíveis e suas datas de lançamento.

Modelagem de Software

Esta sessão tende a demonstrar o que será desenvolvido no banco de dados e quais são as suas representações.

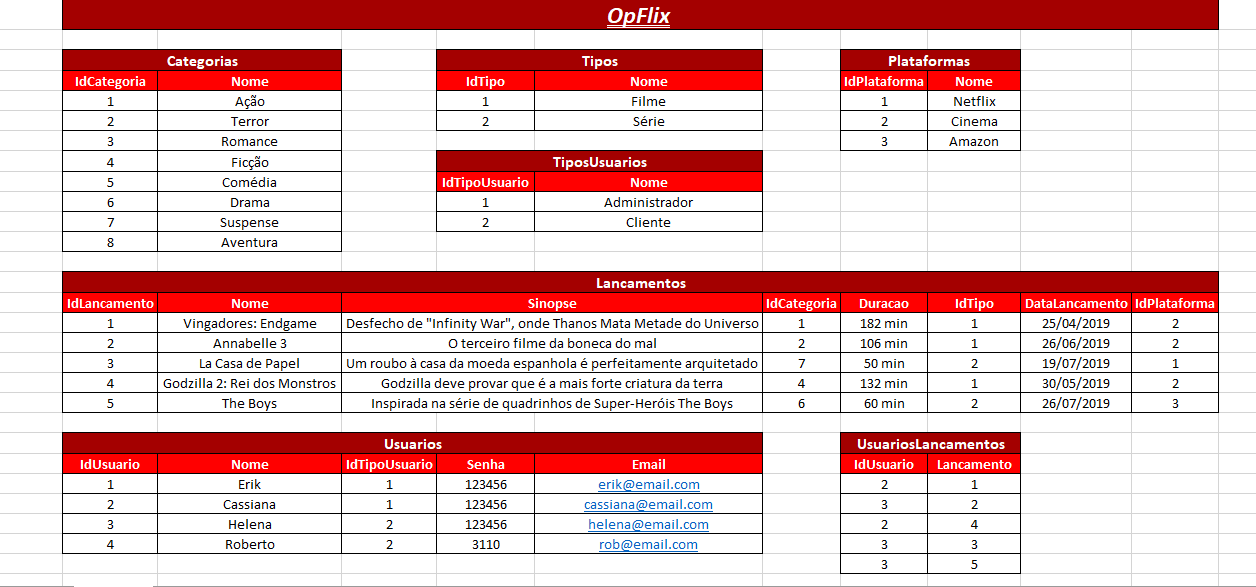
Modelo Lógico

O modelo lógico já leva em conta algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define as [chaves primárias e estrangeiras](https://www.luis.blog.br/chave%20primaria-chave-estrangeira-e-candidata.aspx), normalização, integridade referencial, entre outras. Para o modelo lógico deve ser criado levando em conta os exemplos de modelagem de dados criados no modelo conceitual.



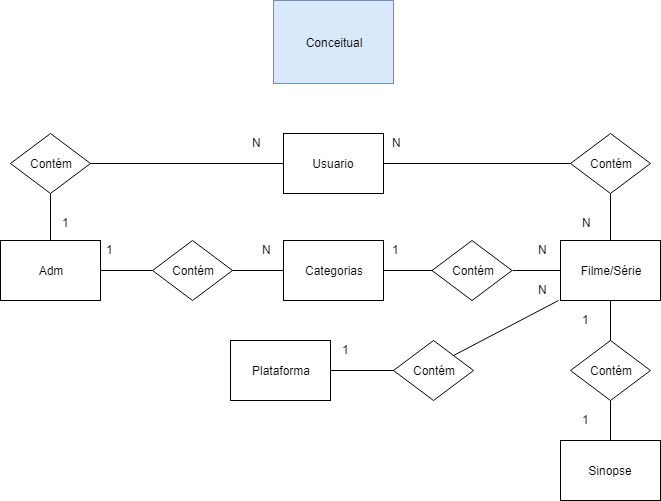
Modelo Físico

No modelo físico fazemos a modelagem física do modelo de banco de dados. Neste caso leva-se em conta as limitações impostas pelo SGBD escolhido e deve ser criado sempre com base nos exemplos de modelagem de dados produzidos no item anterior, modelo lógico.



Modelo Conceitual

A modelagem conceitual baseia-se no mais alto nível e deve ser usada para envolver o cliente, pois o foco aqui é discutir os aspectos do negócio do cliente e não da tecnologia. Os exemplos de modelagem de dados vistos pelo modelo conceitual são mais fáceis de compreender, já que não há limitações ou aplicação de tecnologia específica. O diagrama de dados que deve ser construído aqui é o [Diagrama de Entidade e Relacionamento](https://www.luis.blog.br/modelo-de-entidade-e-relacionamento-mer/), onde deverão ser identificados todas as entidades e os relacionamentos entre elas. Este diagrama é a chave para a compreensão do modelo conceitual de dados.



Cronograma

|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 | Dia 5 | Dia 6 | Dia 7 | Dia 8 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Modelo Lógico** |  |  |  |  |  |  |  |  |
| **Modelo Físico** |  |  |  |  |  |  |  |  |
| Modelo Conceitual |  |  |  |  |  |  |  |  |

Funcionalidades

Web

O administrador poderá cadastrar qualquer tipo de usuário (administrador ou cliente);

O administrador poderá cadastrar categorias (contendo nome);

O administrador poderá cadastrar o lançamento de um filme/série (contendo título, uma pequena sinopse, a categoria vinculada, tempo de duração, e se é filme ou série e a data de lançamento);

O cliente poderá visualizar todos os lançamentos publicados;

Público

Login;

Cadastro de novos usuários (pode deixar um endpoint público para cadastro de novos clientes e um outro endpoint aonde administradores cadastram outros administradores);

Administrador

CRUD Lançamentos;

Cadastrar/Listar/Atualizar Categorias;

Cadastrar/Listar/Atualizar Plataformas/Mídias;

Foram usados também nesse projeto o Postman, um aplicativo que te permite visualizar seus endpoints e como se da o funcionamento dos verbos (Rest). O Postman foi baixado

Mobile

Administrador: Para o colaborador da gestão administrativa;

Cliente: Clientes da empresa;

Funcionalidades

O cliente poderá visualizar todos os lançamentos;

O cliente poderá realizar a ordenação por data de lançamento e/ou categoria;

Cliente

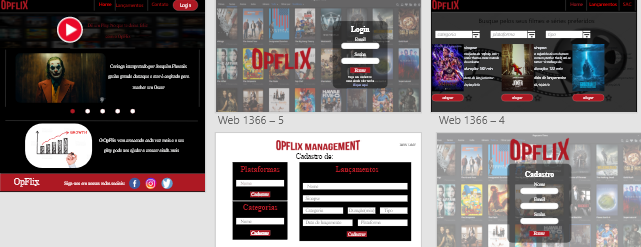
Realizar Login;

Listagem de lançamentos;

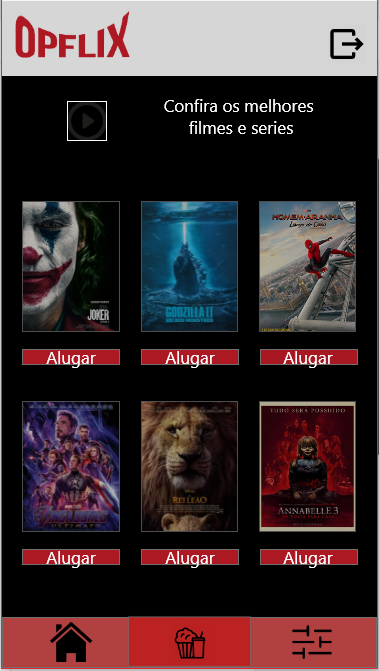
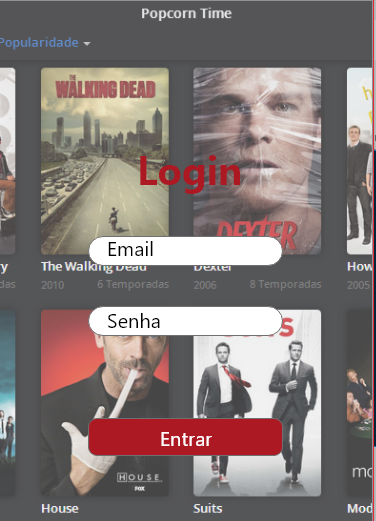
Filtros;

Protótipos

Web

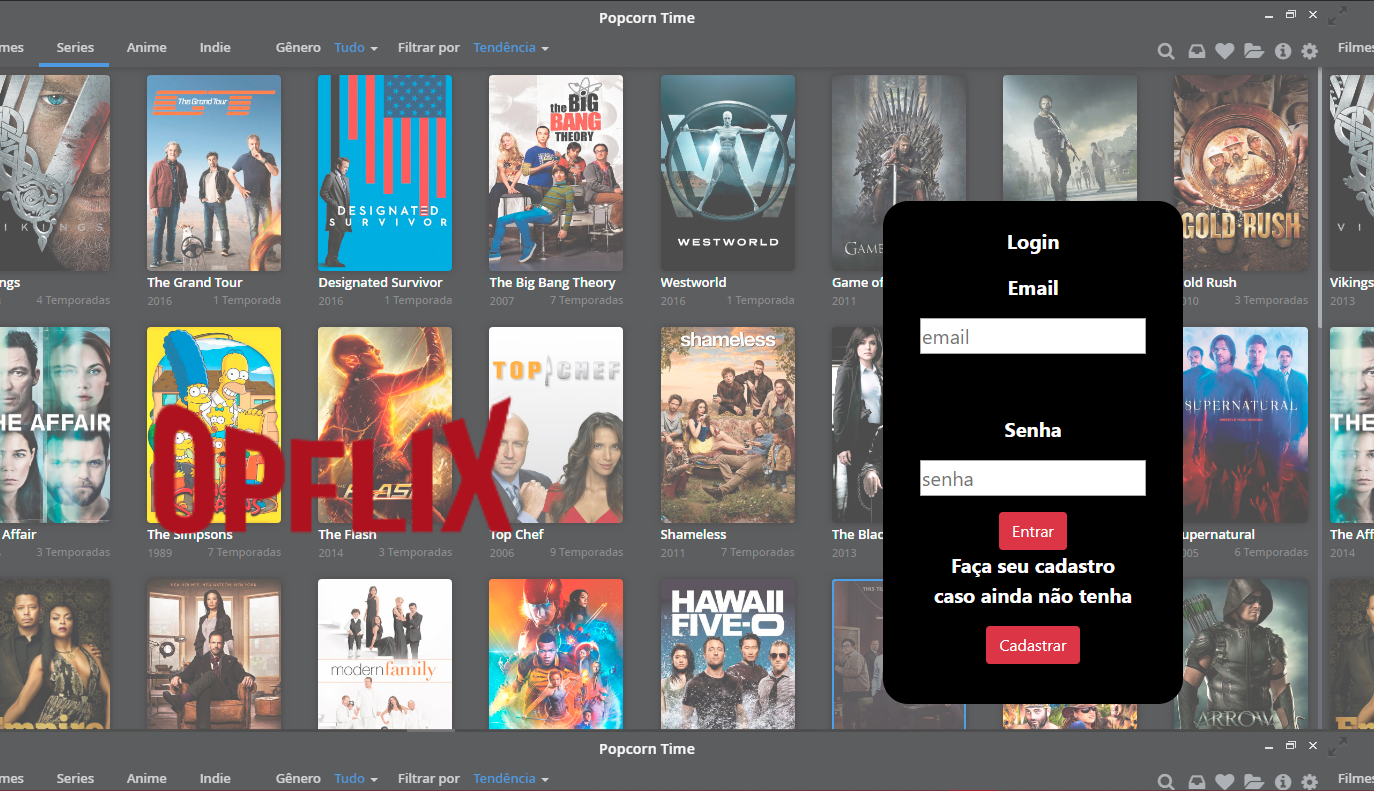


Mobile

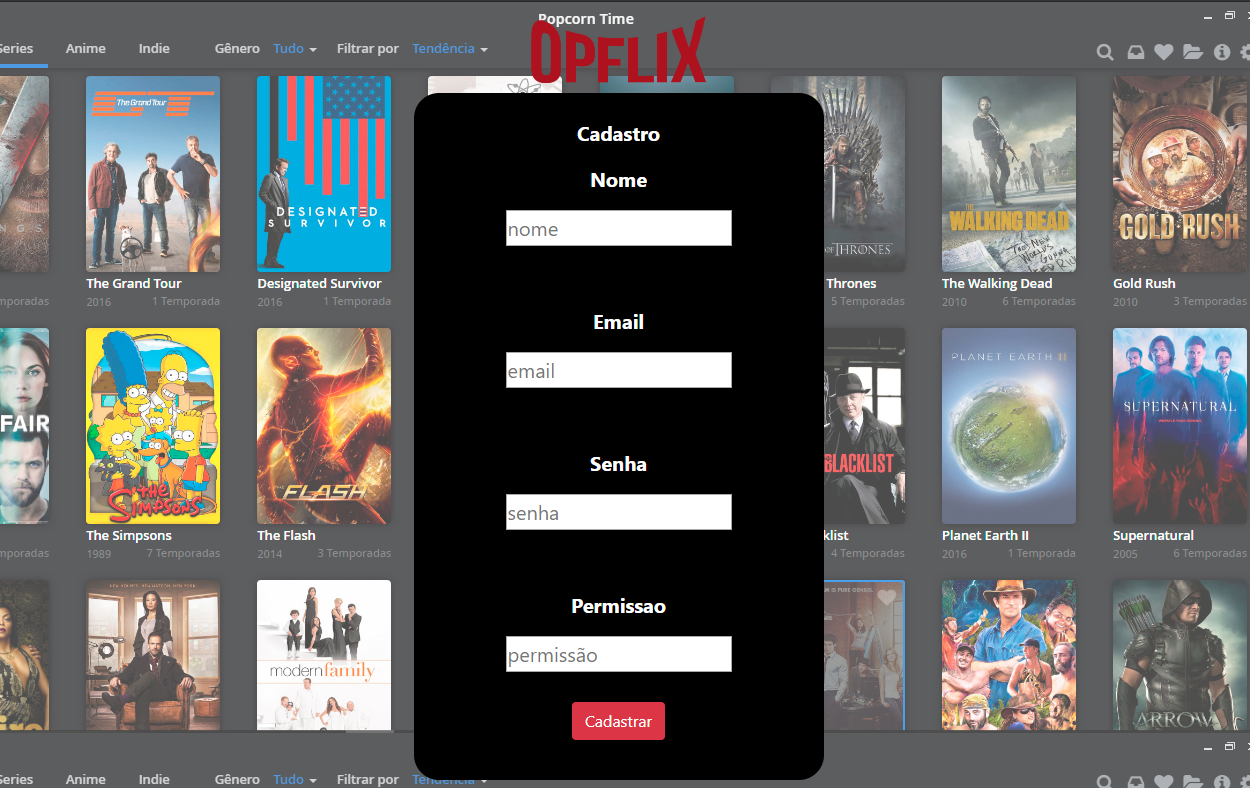
  

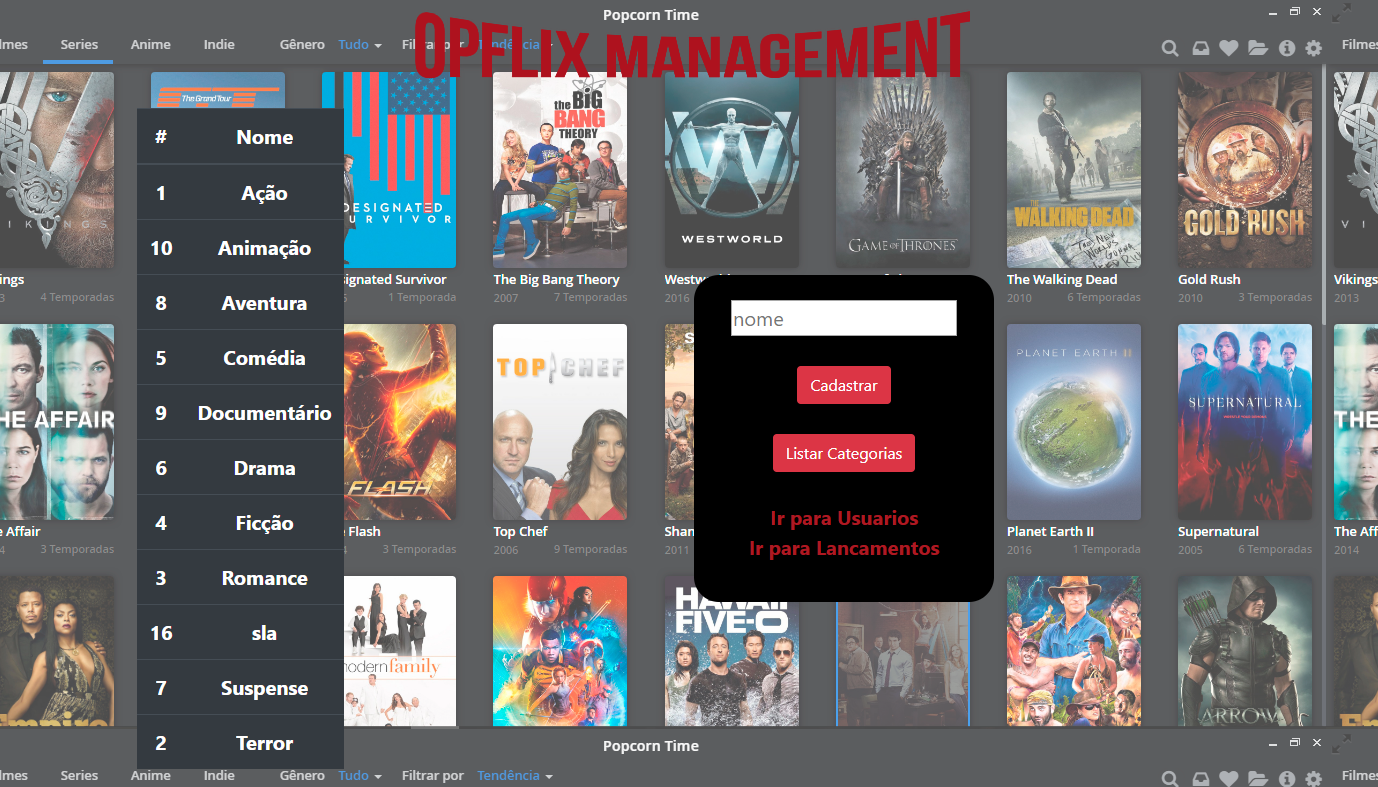
Front-End

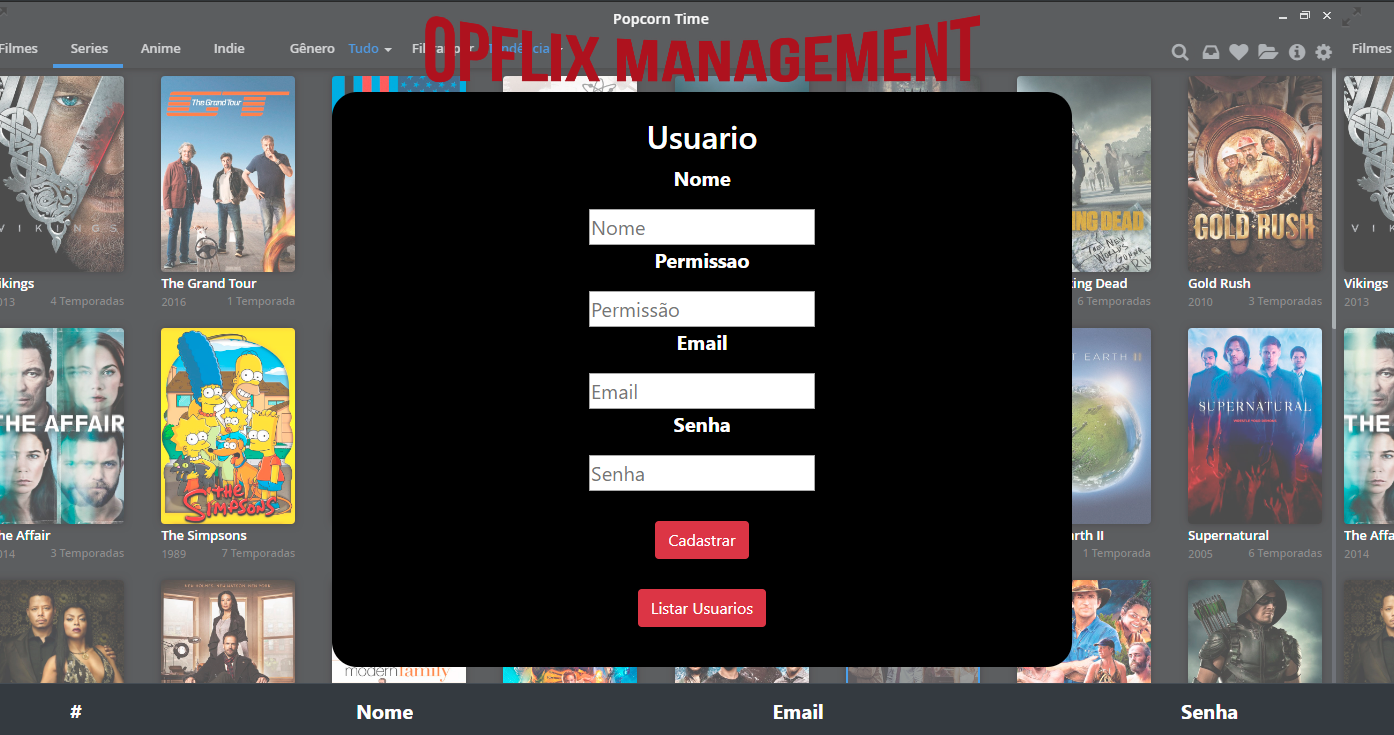












O aplicativo web foi feito utilizando React

Para que fosse feito este projeto, foi preciso a instalação de algumas tecnologias utilizando o npm install. Foram as tecnologias:

"auth": "0.0.9",

"axios": "^0.19.0",

"google-maps-react": "^2.0.2",

"react": "^16.10.2",

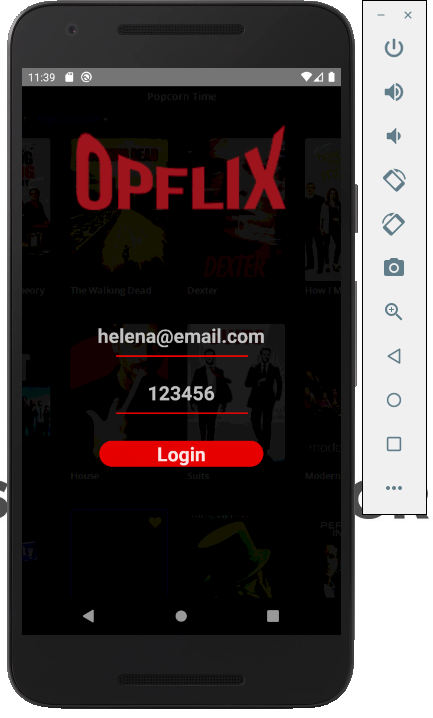
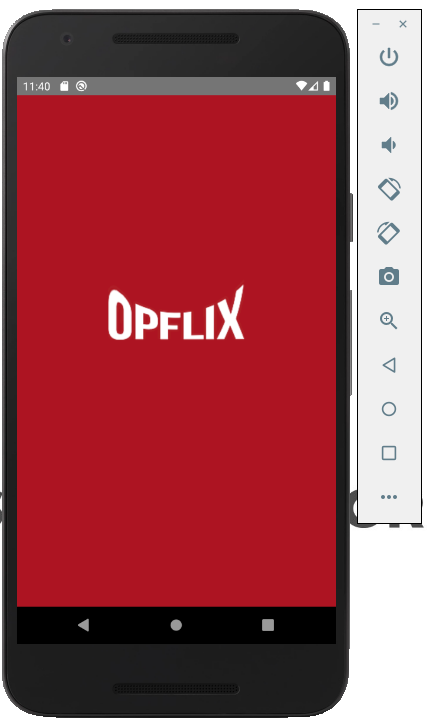
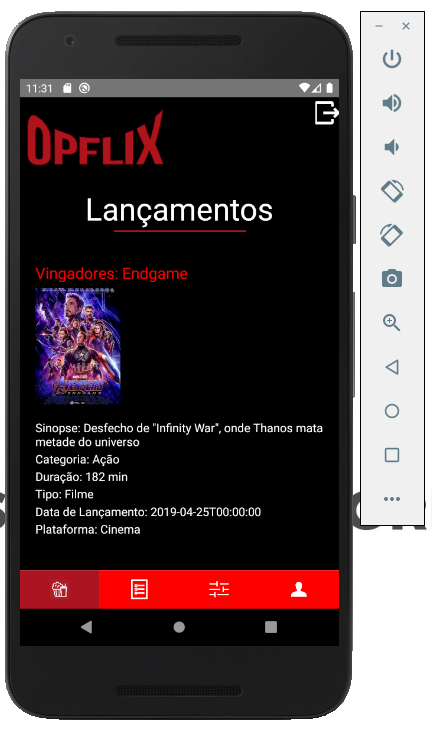
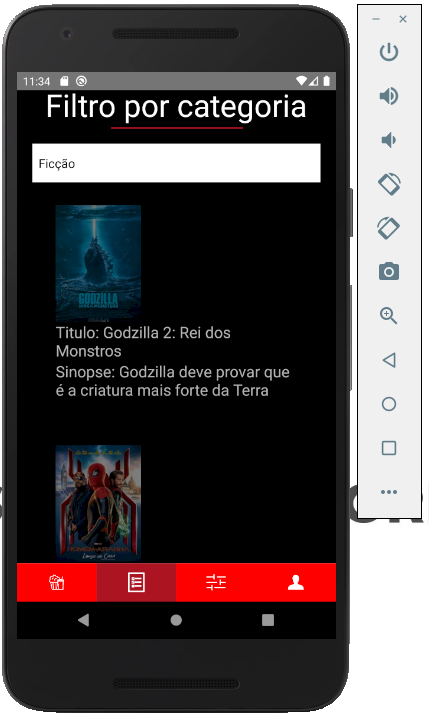
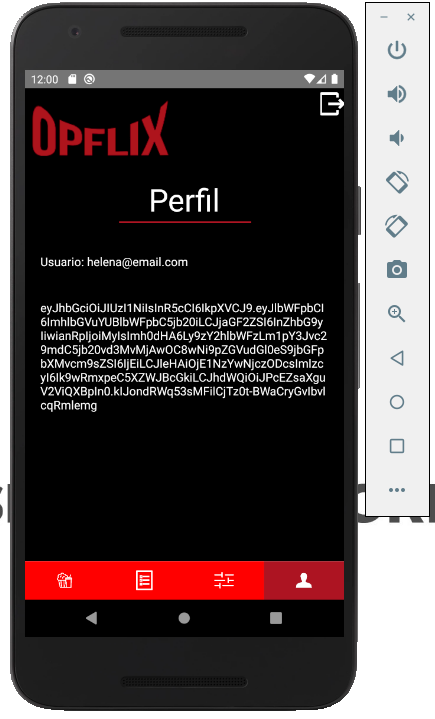
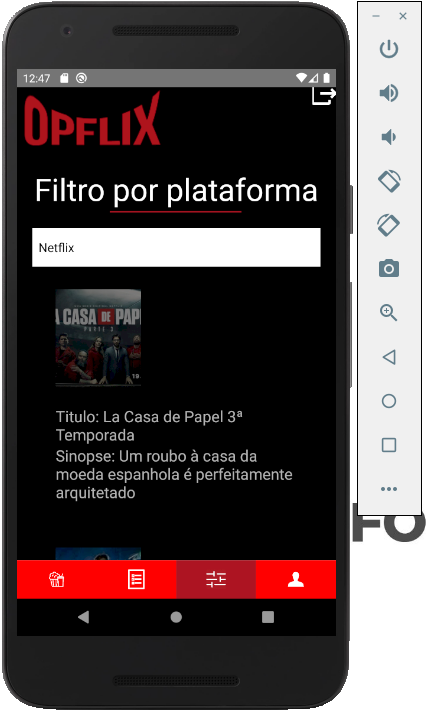
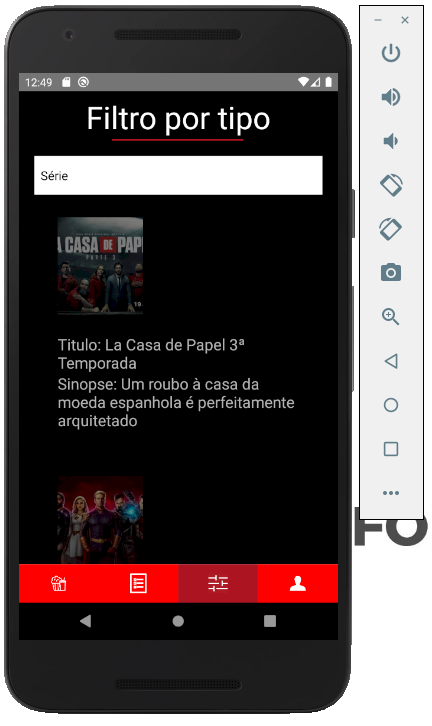
"react-dom": "^16.10.2",

"react-router": "^5.1.2",

"react-router-dom": "^5.1.2",

"react-scripts": "3.2.0"

Mobile

O app mobile foi feito a partir do React-Native, utilizando o emulador do Android Studio

Para que fosse concluído o projeto, forma instaladas as seguintes tecnologias utilizando o npm intstall:

"jwt-decode": "^2.2.0",

"react": "16.8.6",

"react-native": "0.60.6",

"react-native-gesture-handler": "^1.5.0",

"react-native-picker-select": "^6.3.3",

"react-native-reanimated": "^1.4.0",

"react-native-screens": "^2.0.0-alpha.8",

"react-native-search-filter": "^0.1.5",

"react-native-splash-screen": "^3.2.0",

"react-navigation": "^4.0.10",

"react-navigation-stack": "^1.10.3",

"react-navigation-tabs": "^2.5.6"

Arquitetura do Projeto

Referências

Links

Livros