

Documentação

OpFlix

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

Sumário

1. [Resumo 2](#_Toc19525538)

[Objetivos](#_Toc19525539)

1. [Descrição do projeto 2](#_Toc19525540)

[Resumo do projeto](#_Toc19525541)

1. [Modelagem de Software 3](#_Toc19525542)

[Modelo Lógico](#_Toc19525543)

[Modelo Físico](#_Toc19525544)

[Modelo Conceitual](#_Toc19525545)

[Cronograma](#_Toc19525546)

1. [Backend 6](#_Toc19525547)

[Inicialização do programa](#_Toc19525548)

[Deploy do projeto](#_Toc19525549)

[Swagger](#_Toc19525550)

[Pacotes do NuGet](#_Toc19525551)

[Arquitetura do projeto](#_Toc19525552)

[**Postman**](#_Toc19525553)

[**Ferramentas utilizadas**](#_Toc19525554)

1. [Funcionalidades 9](#_Toc19525555)

[Web](#_Toc19525556)

[Mobile](#_Toc19525557)

1. [Protótipos 10](#_Toc19525558)

[Web](#_Toc19525559)

[Mobile](#_Toc19525560)

1. [Front-End 11](#_Toc19525561)
2. [Mobile 12](#_Toc19525562)
3. [Arquitetura do Projeto 13](#_Toc19525563)
4. [Referências 14](#_Toc19525564)

[Links](#_Toc19525565)

[Livros](#_Toc19525566)

Resumo

Objetivos

O documento a seguir tem como objetivo descrever e estruturar a modelagem, cronograma e outros detalhes do *software* requisitado por Tadeu da companhia OpFlix. O documento também descreve os diversos *endpoitns* (funções) que a aplicação possui e como verifica-las. O presente arquivo será enviado junto a três fotos que dizem respeito aos modelos apresentados mais à frente do documento, três arquivos sql e a um arquivo de coleção do Postman.

Descrição do projeto

Uma nova companhia no ramo cinematográfico chamada OpFlix, empresa de pequeno porte que atua no ramo de filmes e séries, foi criada por Tadeu em 2019 na região da Paulista em São Paulo. Tadeu tem uma equipe de parceiros que desejam realizar a divulgação de novos lançamentos por ano.

Sua empresa, por ser nova, realiza a divulgação apenas pelas redes sociais, porém deseja realizar a divulgação por sua página na internet e através de um aplicativo mobile.

Tadeu nos contratou para desenvolver um sistema web/mobile integrado onde seja possível realizar a gestão de todos os lançamentos de maneira integrada e otimizada.

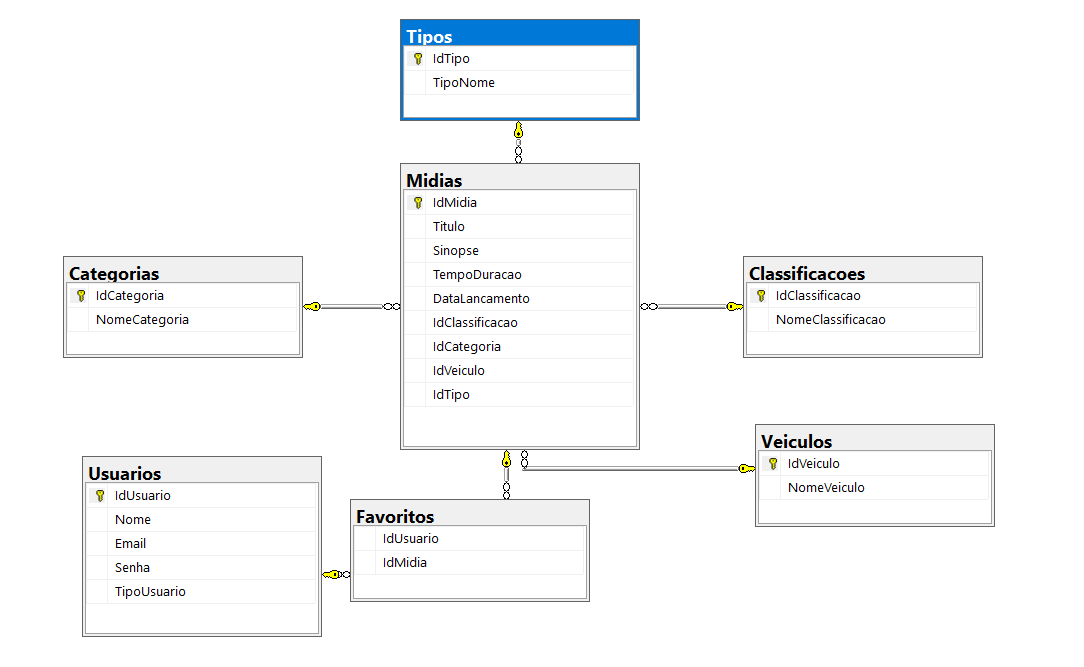
Resumo do projeto

Desenvolvimento de software com foco em divulgar os lançamentos dos cinemas, Netflix, Amazon Prime, entre outros.

Modelagem de Software

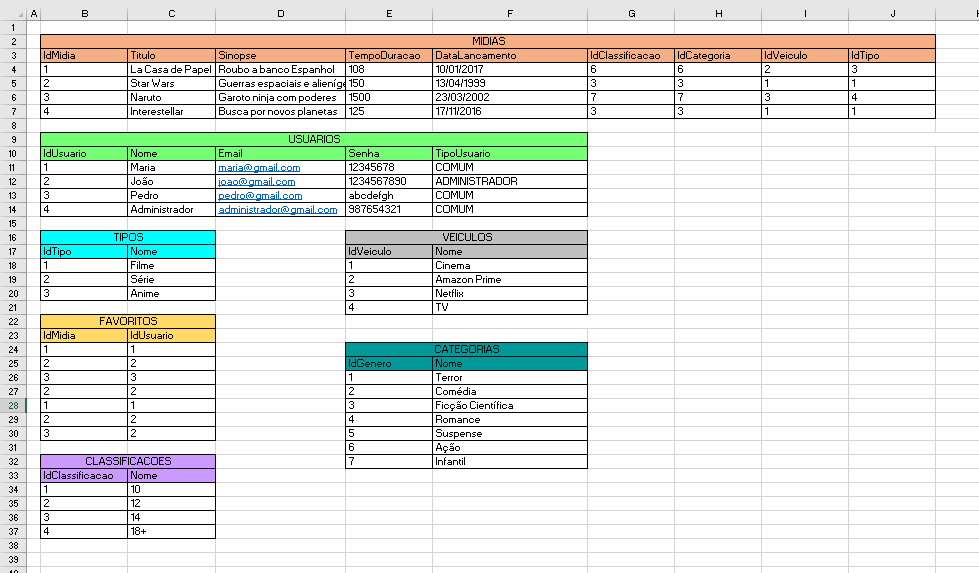
A modelagem de *software* é uma representação de como as informações da aplicação estão dispostas e organizadas. Ele facilita o entendimento e reduz os erros na criação da aplicação.

Modelo Lógico

O modelo lógico é o que melhor representa a forma na qual as informações da aplicação estão organizadas e relacionadas

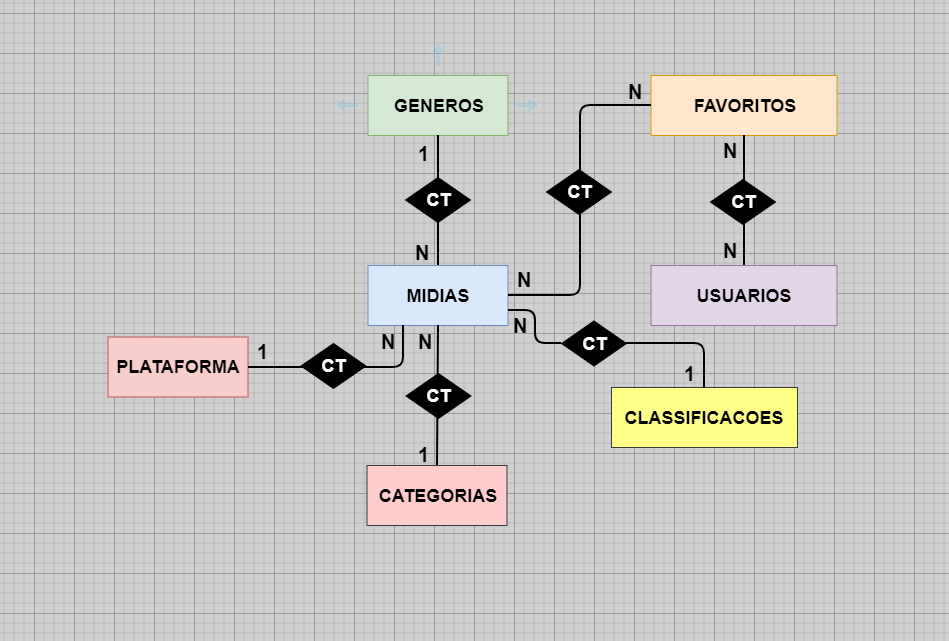
Modelo Físico

O modelo físico representa um exemplo das informações que podem estar contidas no banco de dados da aplicação e sua disposição nele. É o modelo que representa de modo mais visual o banco de dados.



Modelo Conceitual

O modelo conceitual demonstra principalmente o nome dos grupos de informação presentes no banco e como elas se relacionam.



Cronograma

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

Backend

Inicialização do programa

**Banco de Dados**

Primeiramente acesse esse link do GitHub https://github.com/CassioCancio/ 2s2019-sprint-1-bd-opflix e clone esse repositório na sua máquina. O presente arquivo será enviado junto a três fotos que dizem respeito aos modelos apresentados mais à frente do documento e a três arquivos sql. Os arquivos sql são uma espécie de roteiro de criação que deve ser reproduzido no programa para computador Microsoft SQL Server Management Studio 18. Os arquivos devem ser executados na seguinte ordem:

1° M\_01\_CassioCancio\_DDL;

2° M\_02\_CassioCancio\_DML;

3° M\_03\_CassioCancio\_DQL.

Para executar, abra o programa descrito acima e arraste o primeiro arquivo para o programa e pressione o botão do teclado “F5”, depois faça o mesmo com o segundo e com o terceiro. O banco de dados já está pronto.

**Aplicação API**

É necessário, primeiramente, que o programa Visual Studio 2017 esteja instalado na máquina. Para executar a aplicação abra a pasta 2s2019-sprint-1-bd-opflix/Senai.OpFlix.WebApi e clique duas vezes no arquivo “Senai.OpFlix.WebApi”. O arquivo será iniciado no programa descrito acima. Para iniciar as funcionalidades basta pressionar o botão do teclado “F5” ou clicar no botão que aparecerá na aplicação:



Ao executar, uma aba do Google Chrome abrirá instantaneamente com a seguinte URL “http://localhost:5000”. Desse modo, a aplicação já estará funcionando.

Deploy do projeto

Primeiramente acesse esse link do GitHub https://github.com/CassioCancio/ 2s2019-sprint-1-bd-opflix e clone esse repositório na sua máquina. Após a clonagem, acesse:

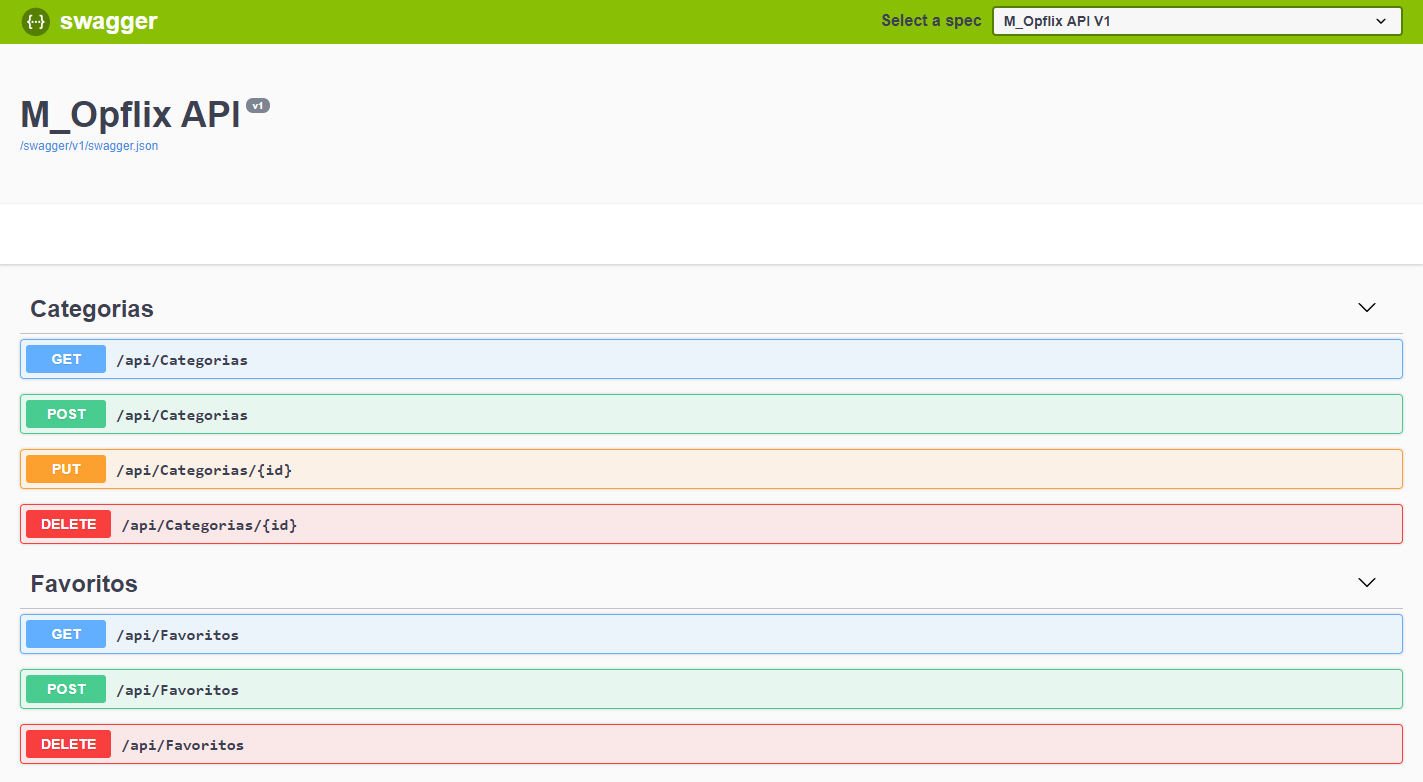


Swagger

Para acessar o Swagger é necessário executar o programa no Visual Studio 2017, apertando o seguinte botão:

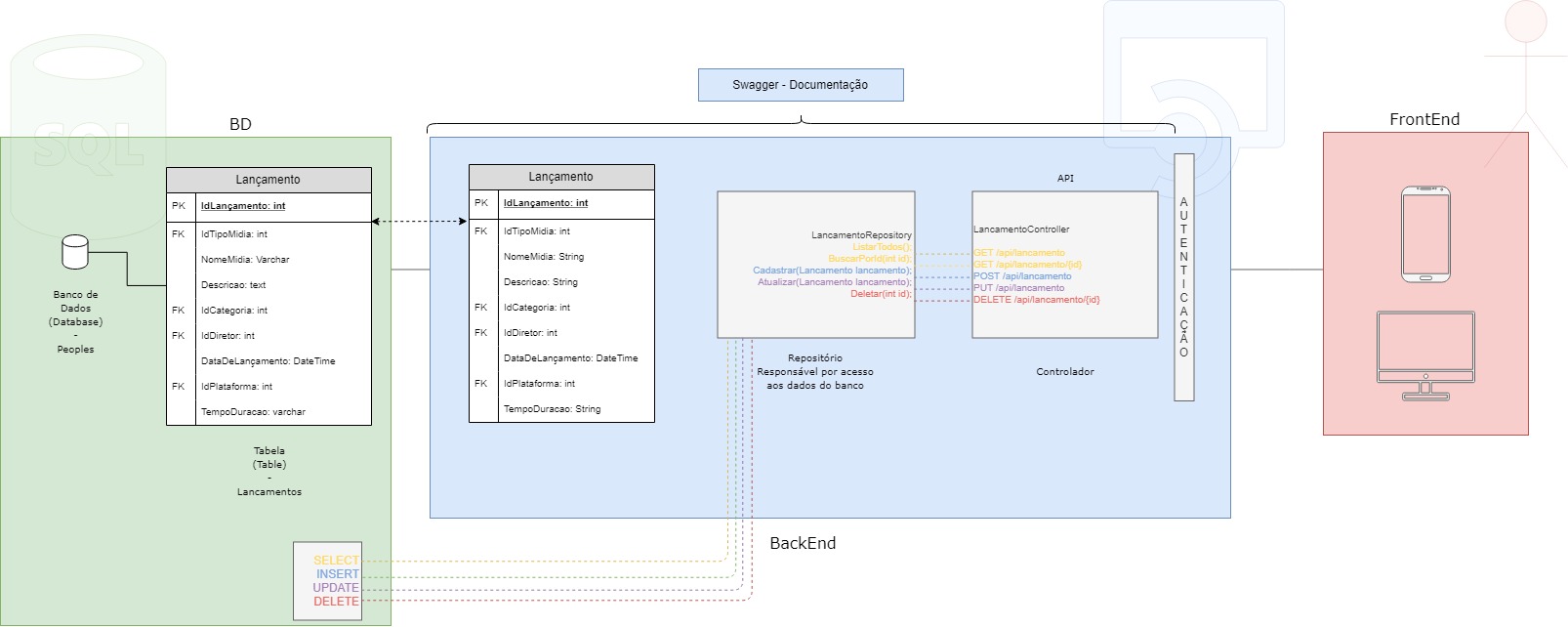


Ao executar, uma aba do Google Chrome abrirá instantaneamente com a seguinte URL “http://localhost:5000”. Para acessar o Swagger, adicione a esta URL o seguinte endereço de pesquisa “/swagger”, que direcionará a seguinte tela:



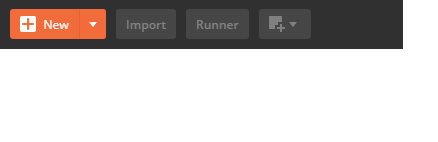
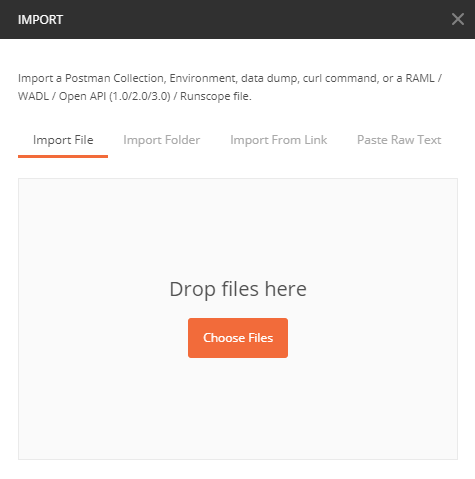
Pacotes do NuGet

* Microsoft.AspNetCore.App 2.1.1
* Microsoft.AspNetCore.Authentication.JwtBearer 2.1.1
* Microsoft.AspNetCore.Razor.Design 2.1.2
* Microsoft.EntityFrameworkCore.SqlServer 2.1.11
* Microsoft.EntityFrameworkCore.SqlServer.Design 1.1.6
* Microsoft.EntityFrameworkCore.Tools 2.1.11
* Microsoft.NETCore.App 2.1.0
* Microsoft.VisualStudio.Web.CodeGeneration.Design 2.1.9
* Swashbuckle.AspNetCore 4.0.1
* System.Data.SqlClient 4.6.1
* System.IdentityModel.Tokens.Jwt 5.5.0

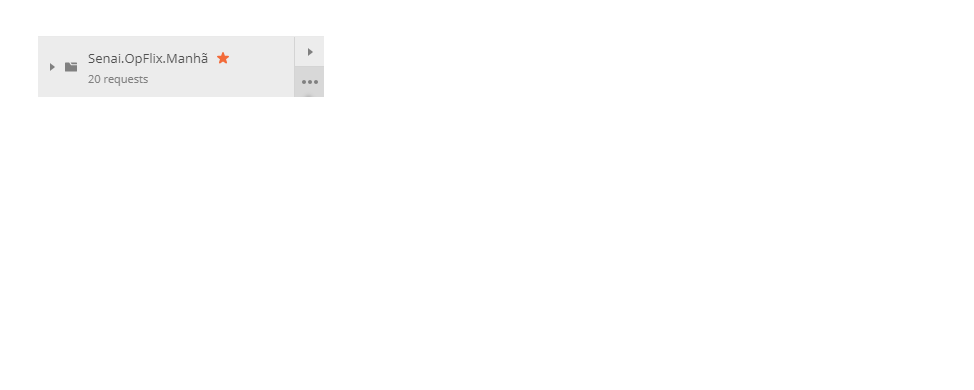
Arquitetura do projeto

## **Postman**

**Importar**

1. Abra o aplicativo “Postman” em seu desktop;
2. Clique no botão “Import” presente no menu:
3. Selecione a opção “Import File” e clique no botão “Choose Files”:
4. Selecione o arquivo “OpFlix\_Postman”, presente na pasta “2s2019-OpFlix-sprint-2-BackEnd”;
5. Todos os arquivos estarão presentes na pasta “Senai.OpFlix.Manhã”;

**Exportar:**

1. Abra o aplicativo “Postman” em seu desktop;
2. Na pasta “Senai.OpFlix.Manhã”, clique no botão com os três pontos:
3. Selecionar a opção “Export” na coleção “Collection v2.1”;
4. Colocar o nome e o local desejado para salvar;
5. Após isso, o projeto já está exportado para seus arquivos.

## **Ferramentas utilizadas**

* Microsoft Visual Studio 2017
* Microsoft SQL Server Management Studio 18
* Postman
* Navegador de Web
* Swagger
* Jwt.io
* Trello
* GitHub

Funcionalidades

Web

As funcionalidades da aplicação pra web são as seguintes:

I. O administrador poderá cadastrar qualquer tipo de usuário (administrador ou cliente), também poderá listar os usuários cadastrados;

II. O administrador poderá cadastrar qualquer tipo de usuário (administrador ou cliente), também poderá listar os usuários cadastrados;

III. O administrador poderá cadastrar plataformas (contendo nome e mídia), poderá também listar e atualizar;

IV. O administrador poderá cadastrar categoria (contendo nome), poderá também listar e atualizar;

V. O administrador poderá cadastrar o lançamento de um filme/série (contendo título, uma pequena sinopse, a categoria vinculada, tempo de duração, se é filme ou série e a data do primeiro lançamento daquele item), também poderá listar, atualizar, deletar, filtrar por data de lançamento e filtrar por nome da plataforma;

VI. O cliente poderá visualizar todos os lançamentos publicados;

Mobile

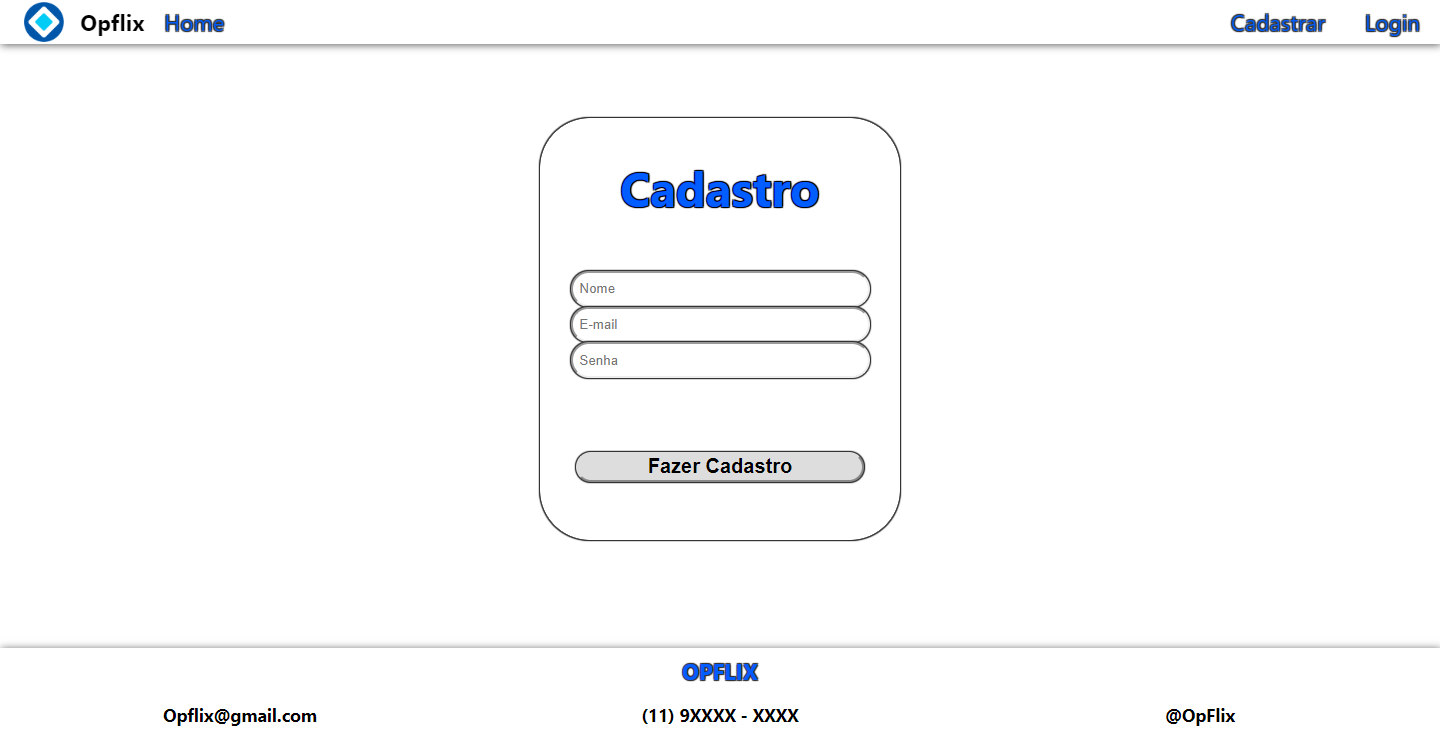
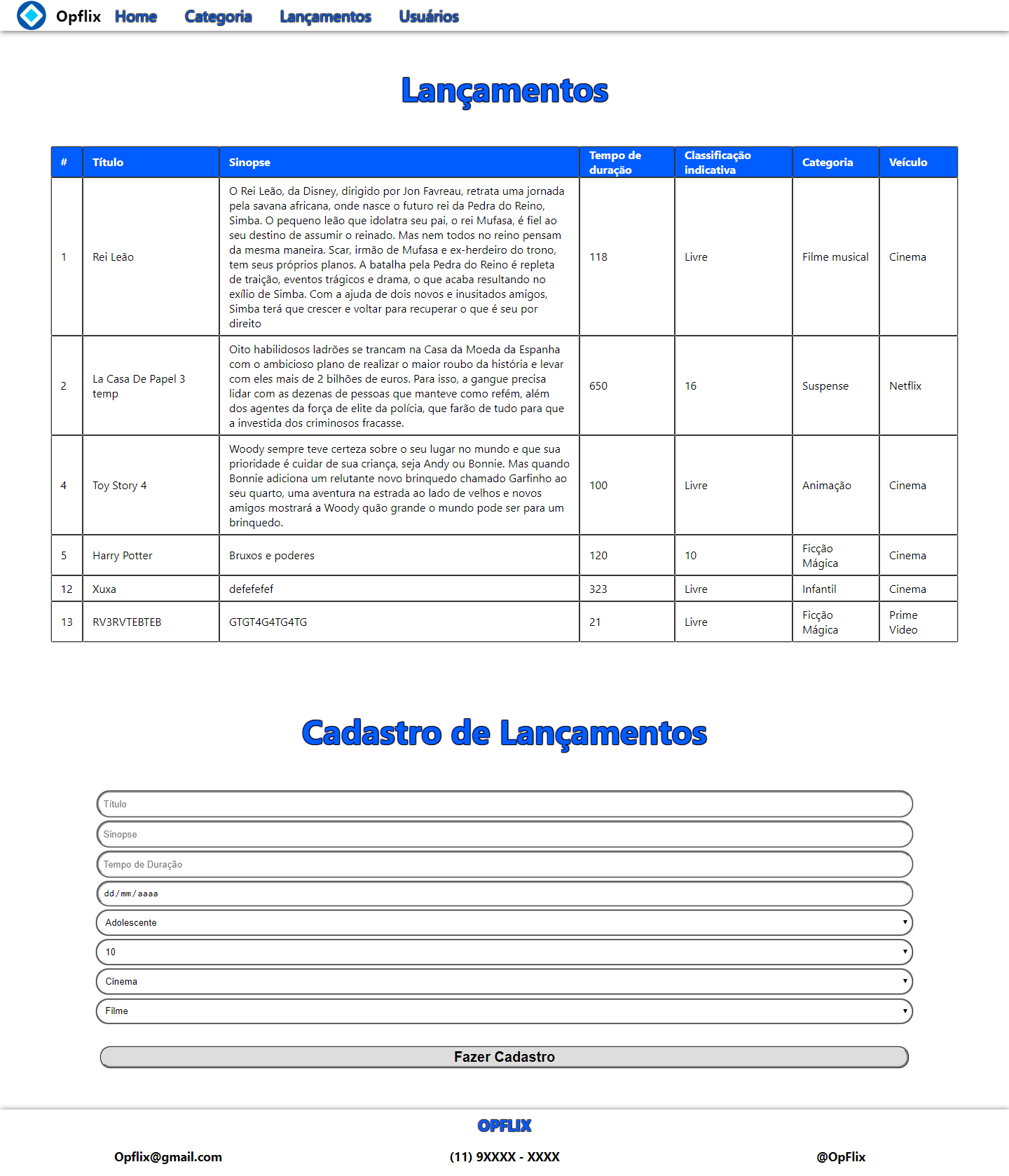
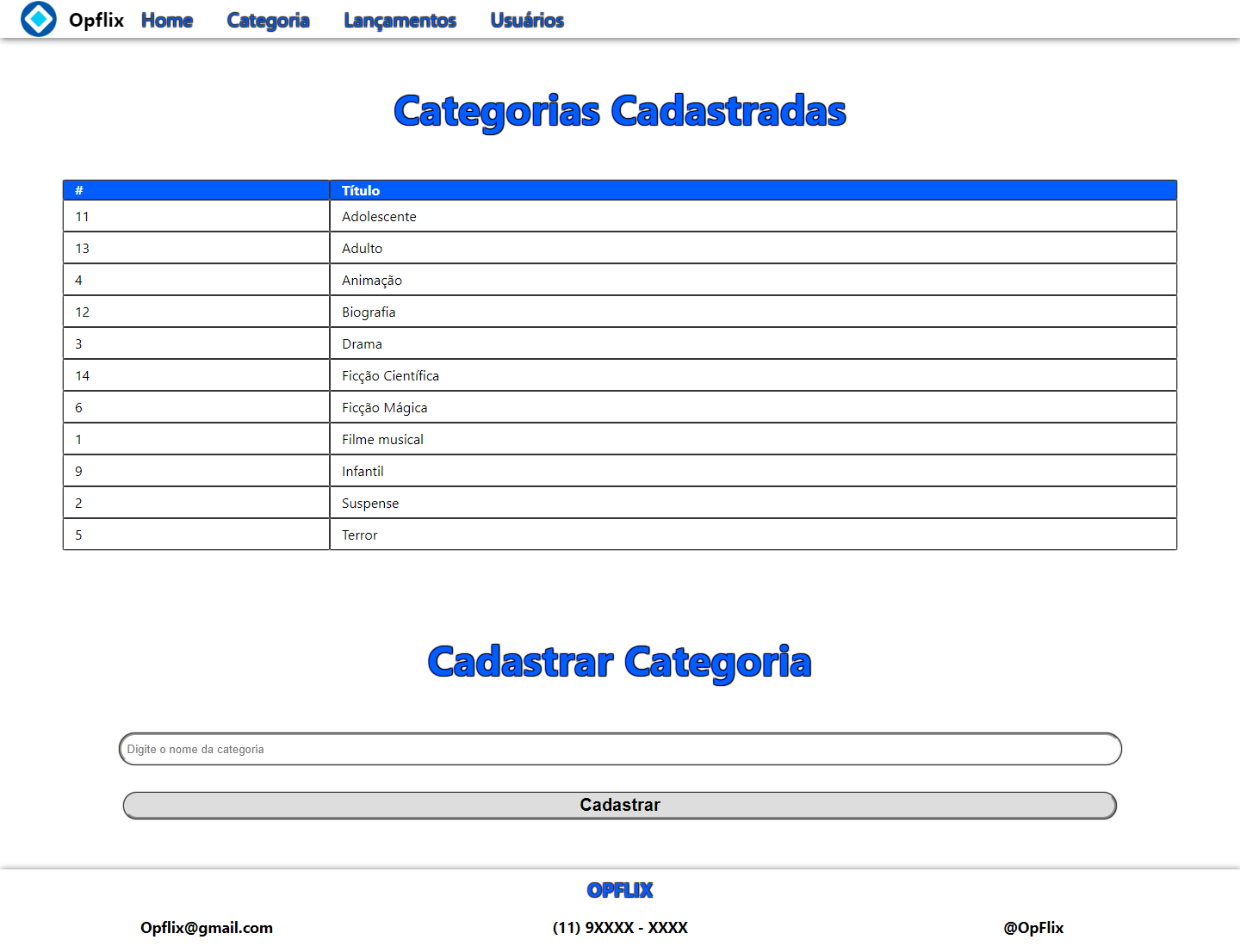
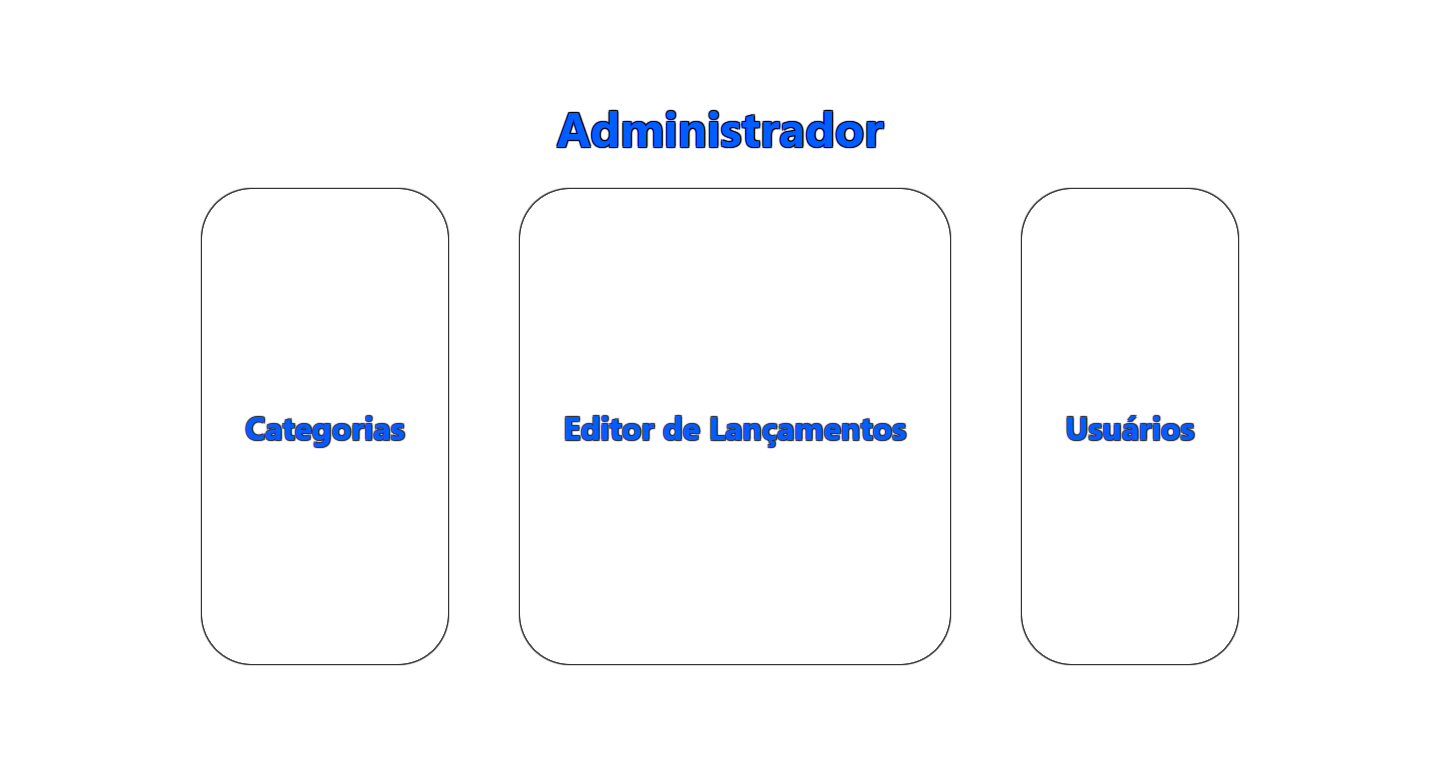
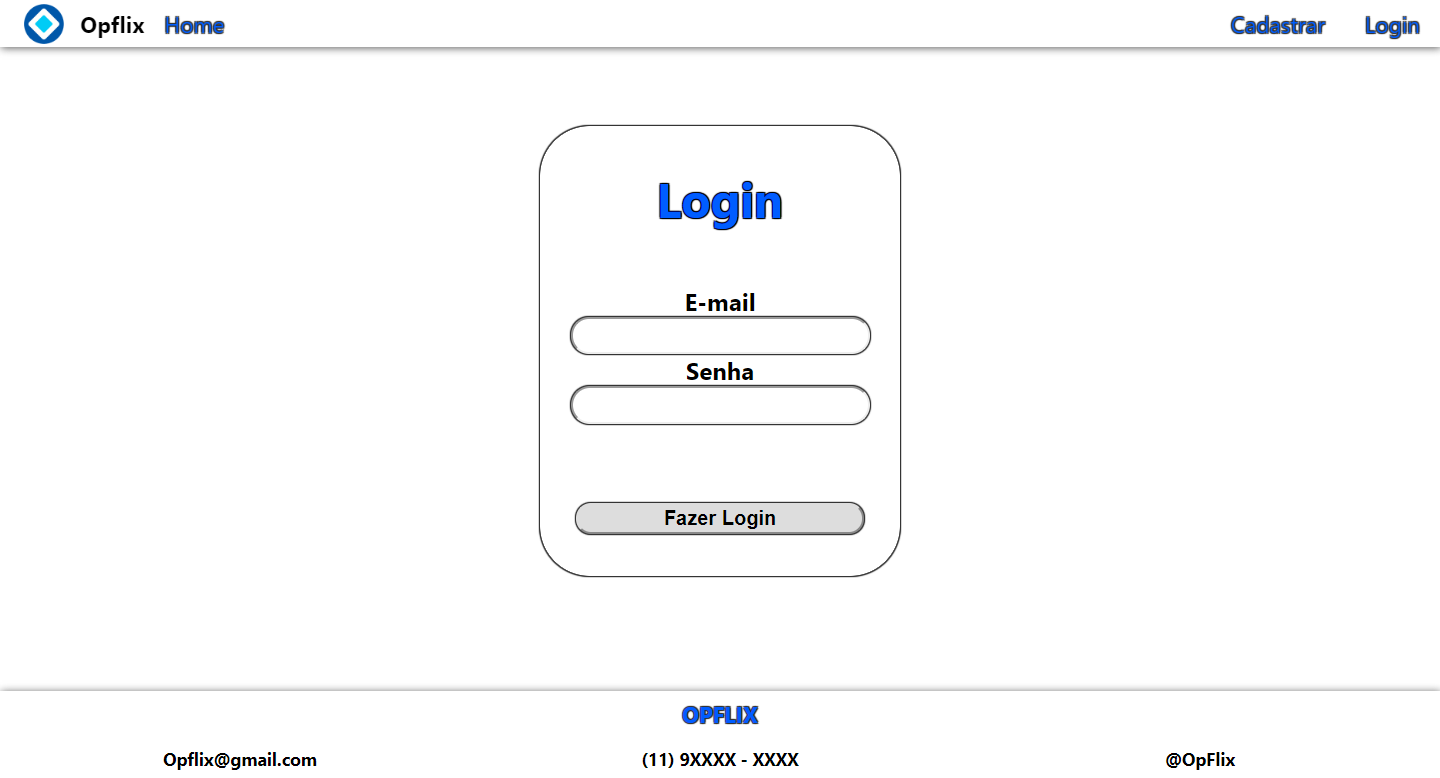
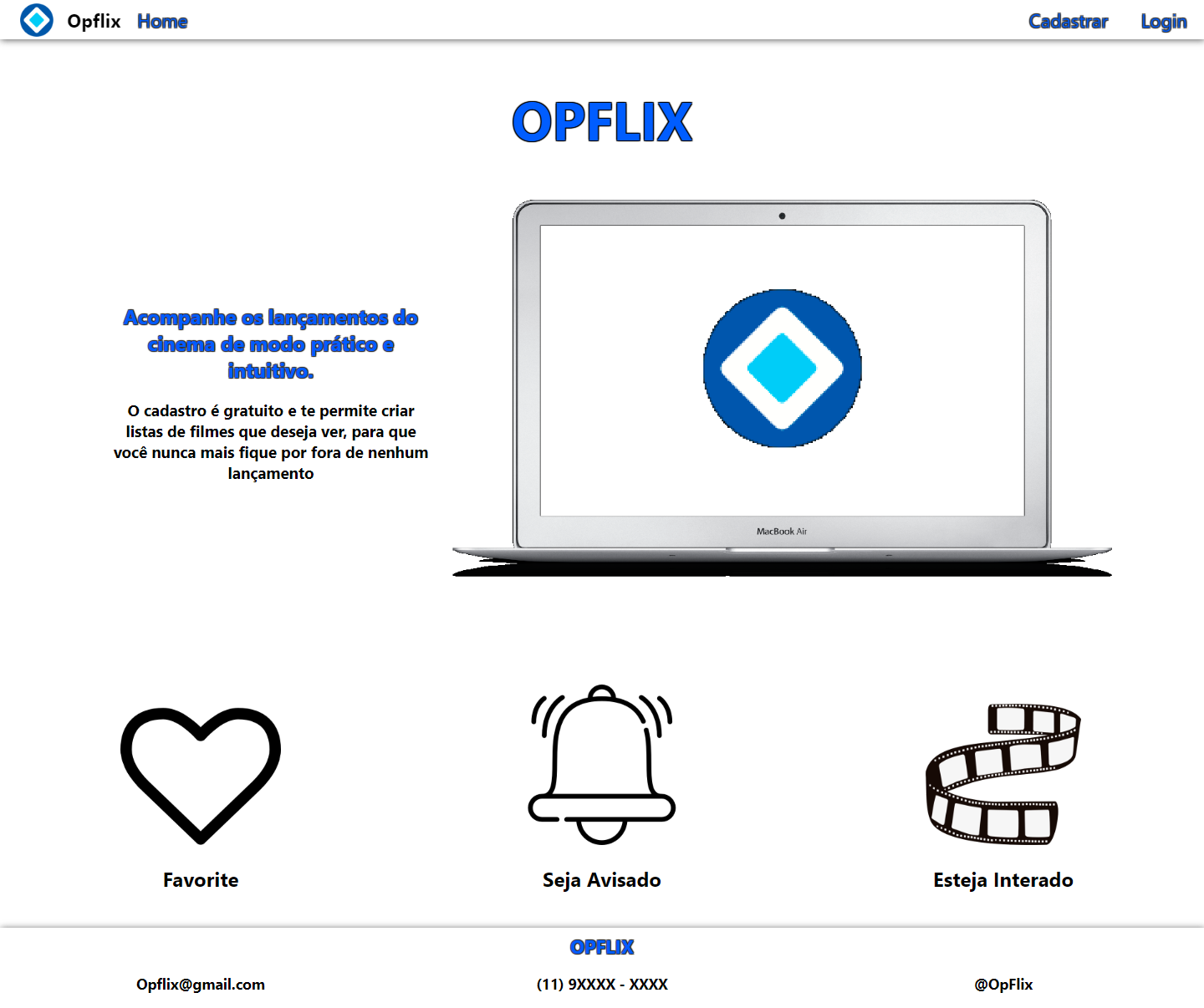
As funcionalidades da aplicação para mobile são:

I. O cliente poderá visualizar todos os lançamentos publicados;

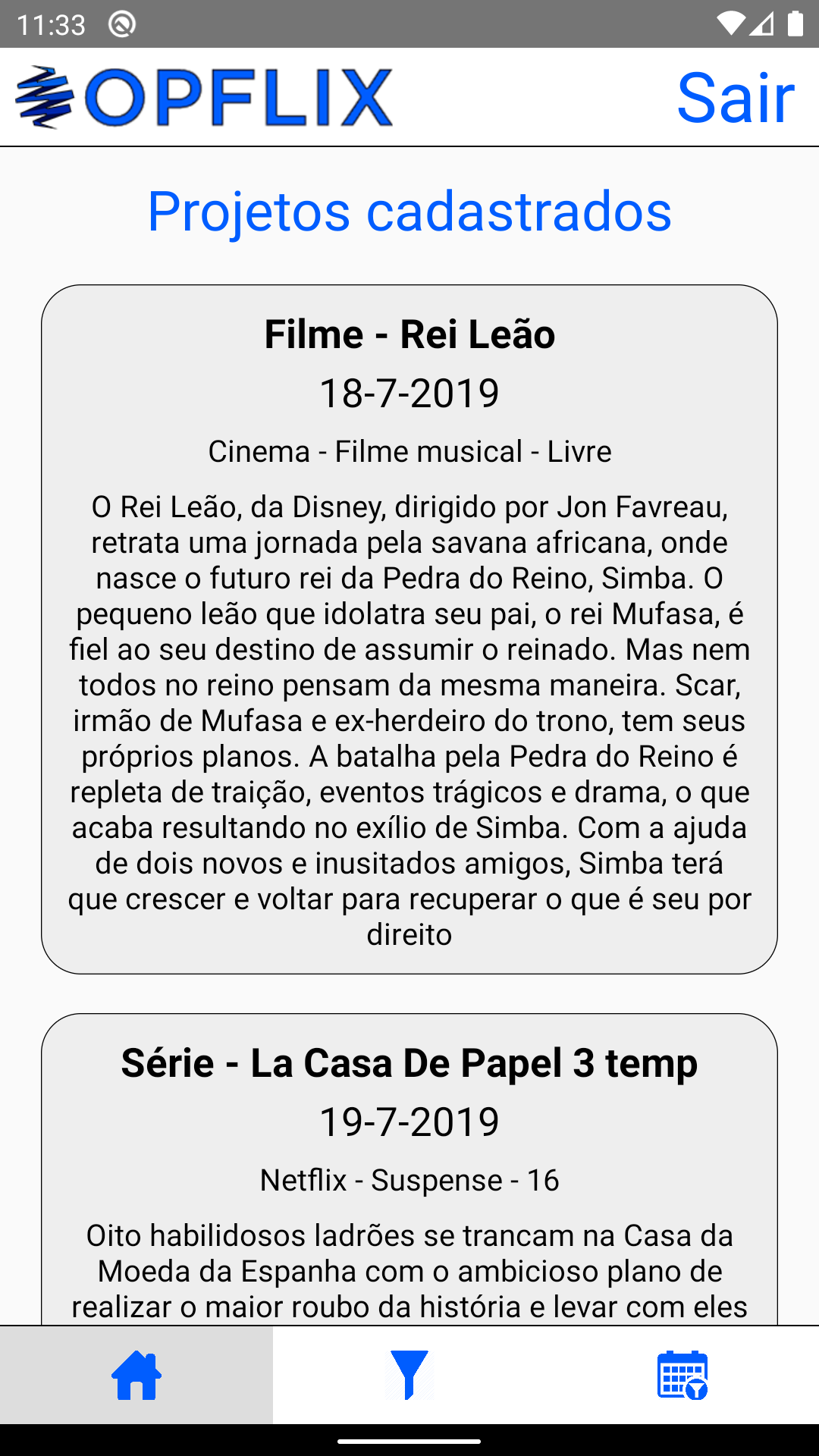
II. O cliente poderá selecionar por mês de lançamento, ou por nome da categoria;

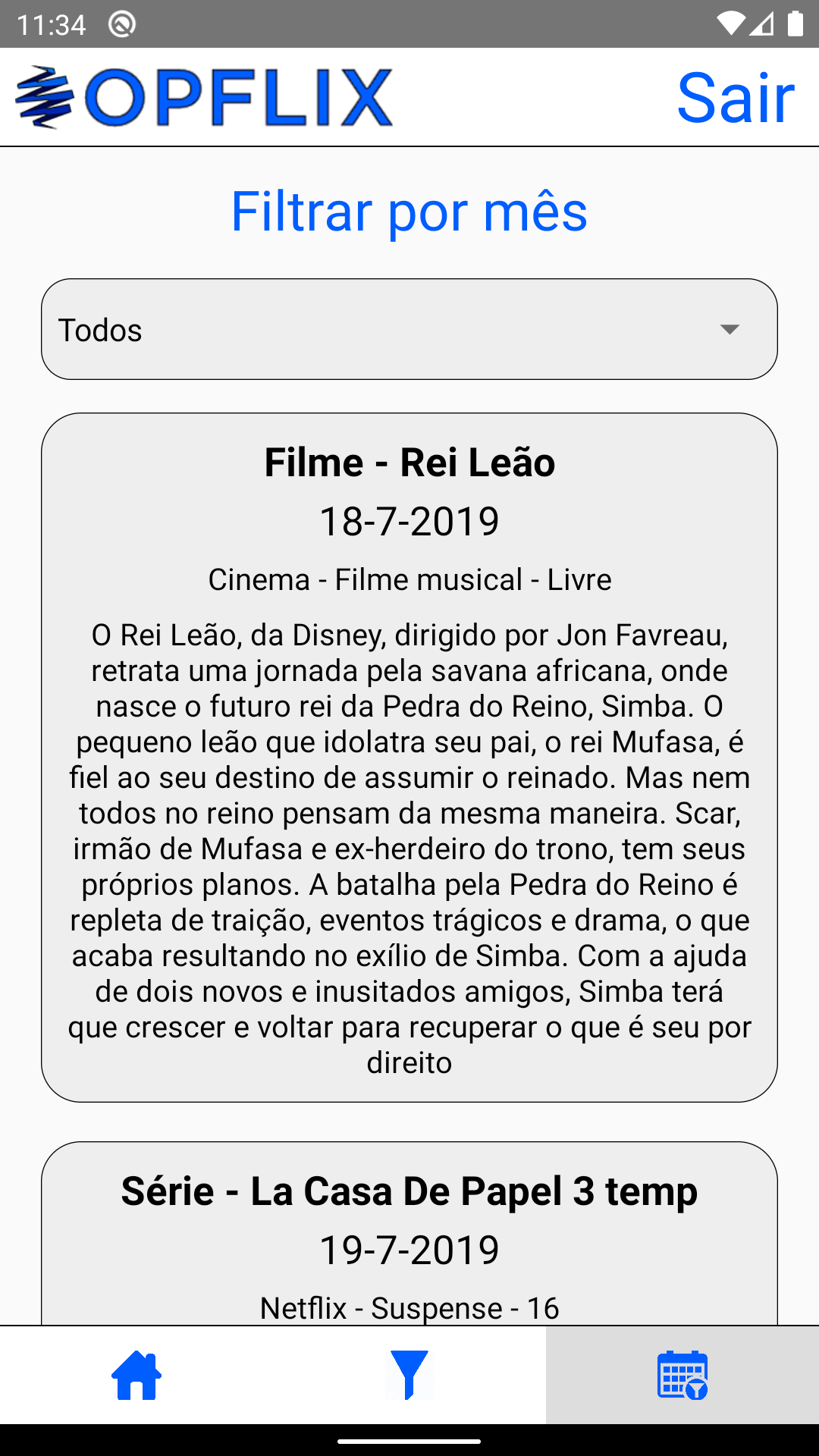
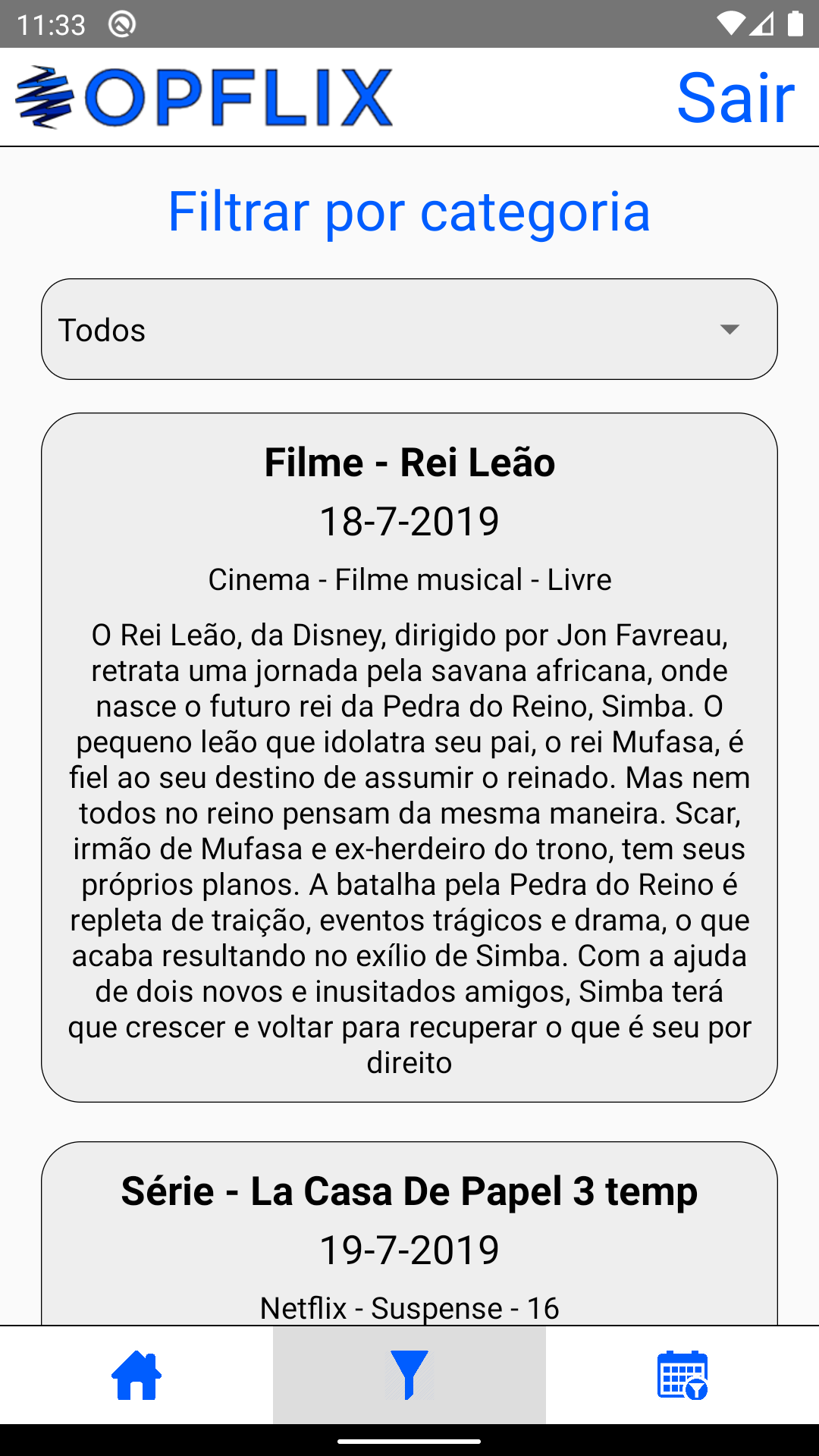
Protótipos

Web

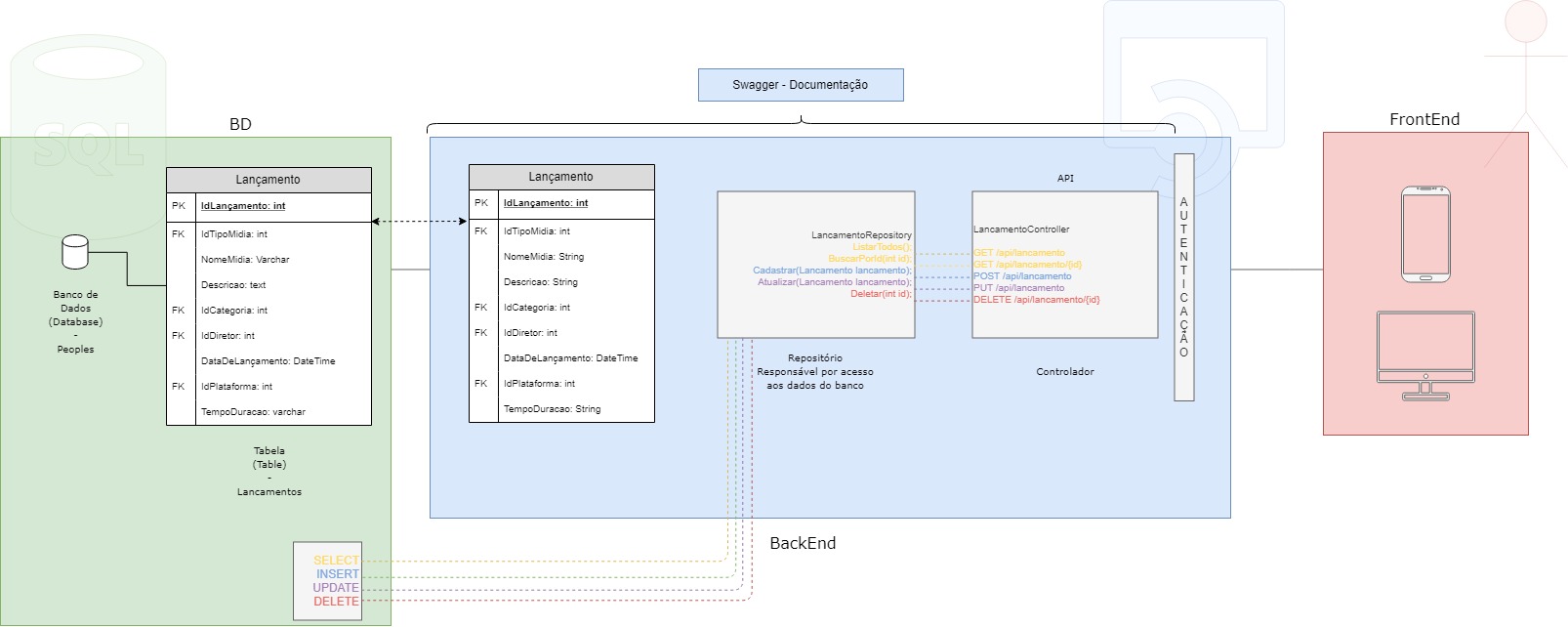


Mobile





Arquitetura do Projeto



Referências

Links

Livros