

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 possui como objetivo demonstrar as funcionalidades e o processo de desenvolvimento da plataforma de divulgação OpFlix.

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 lhe 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

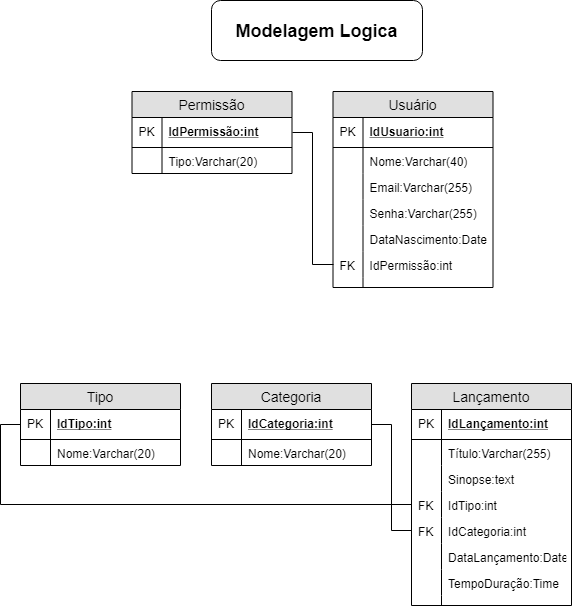
O cliente OpFlix deseja divulgar os lançamentos do universo cinematográfico em mais plataformas além de suas redes sociais. Com esse objetivo, foi solicitada a construção de uma plataforma Web e Mobile.

Modelagem de Software

Essa sessão tem como objetivo demonstrar as modelagens realizadas com o objetivo de estruturar o armazenamento dos dados.

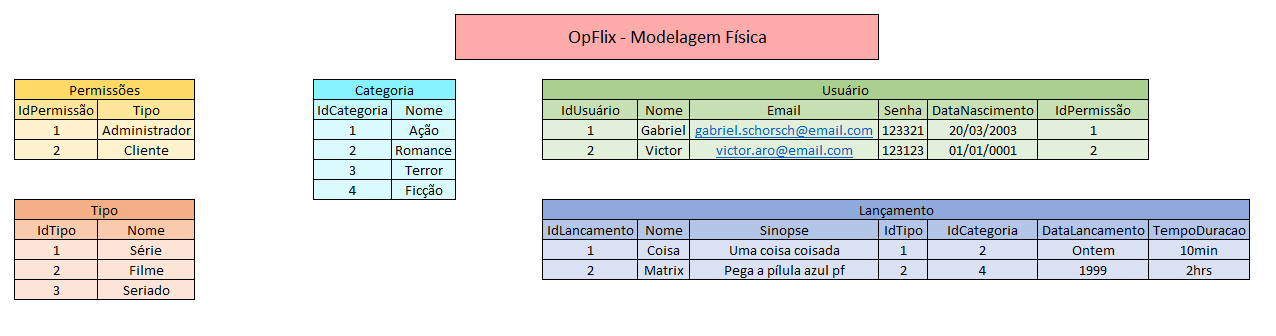
Modelo Lógico

O modelo lógico demonstra as entidades e suas relações dentro do banco de dados, contendo também nomenclaturas e informações como o tipo de dado que será armazenado.



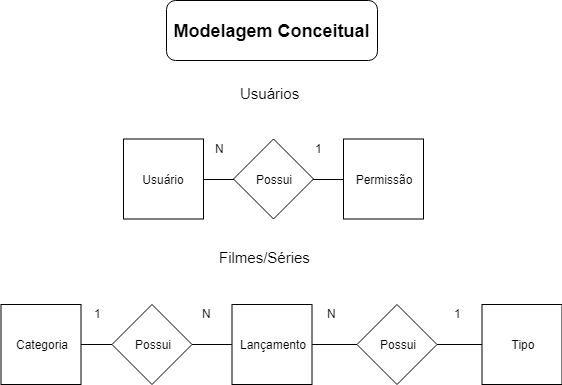
Modelo Físico

O modelo físico é um modelo que representa como as informações serão armazenadas dentro do banco de dados, contendo exemplos de dados reais.



Modelo Conceitual

O modelo conceitual é o modelo que contém as entidades de maior porte e seus relacionamentos, tendo assim o maior nível de abstração entre os modelos.



Para criação e população da tabela no banco de dados relacionais SQL Server deve-se executar os scripts na seguinte ordem:

1. T\_01\_Gabriel\_Schorsch\_DDL.sql
2. T\_02\_Gabriel\_Schorsch\_DML.sql
3. T\_03\_Gabriel\_Schorsch\_DQL.sql

Cronograma

<https://trello.com/b/dkhacl7D/topflix>

Deploy do Projeto

1º - No repositório do GitHub “[2s2019-sprint-1-bd/OpFlix/Modelagem](https://github.com/Schorsch003/2s2019-sprint-1-bd/tree/master/OpFlix)” baixe e execute respectivamente os arquivos [T\_01\_Gabriel\_Schorsch\_DDL.sql](https://github.com/Schorsch003/2s2019-sprint-1-bd/blob/master/OpFlix/Scripts/T_01_Gabriel_Schorsch_DDL.sql), [T\_02\_Gabriel\_Schorsch\_DML.sql](https://github.com/Schorsch003/2s2019-sprint-1-bd/blob/master/OpFlix/Scripts/T_02_Gabriel_Schorsch_DML.sql) e [T\_03\_Gabriel\_Schorsch\_DQL.sql](https://github.com/Schorsch003/2s2019-sprint-1-bd/blob/master/OpFlix/Scripts/T_03_Gabriel_Schorsch_DQL.sql) na plataforma SQL Server Management Studio (SSMS).

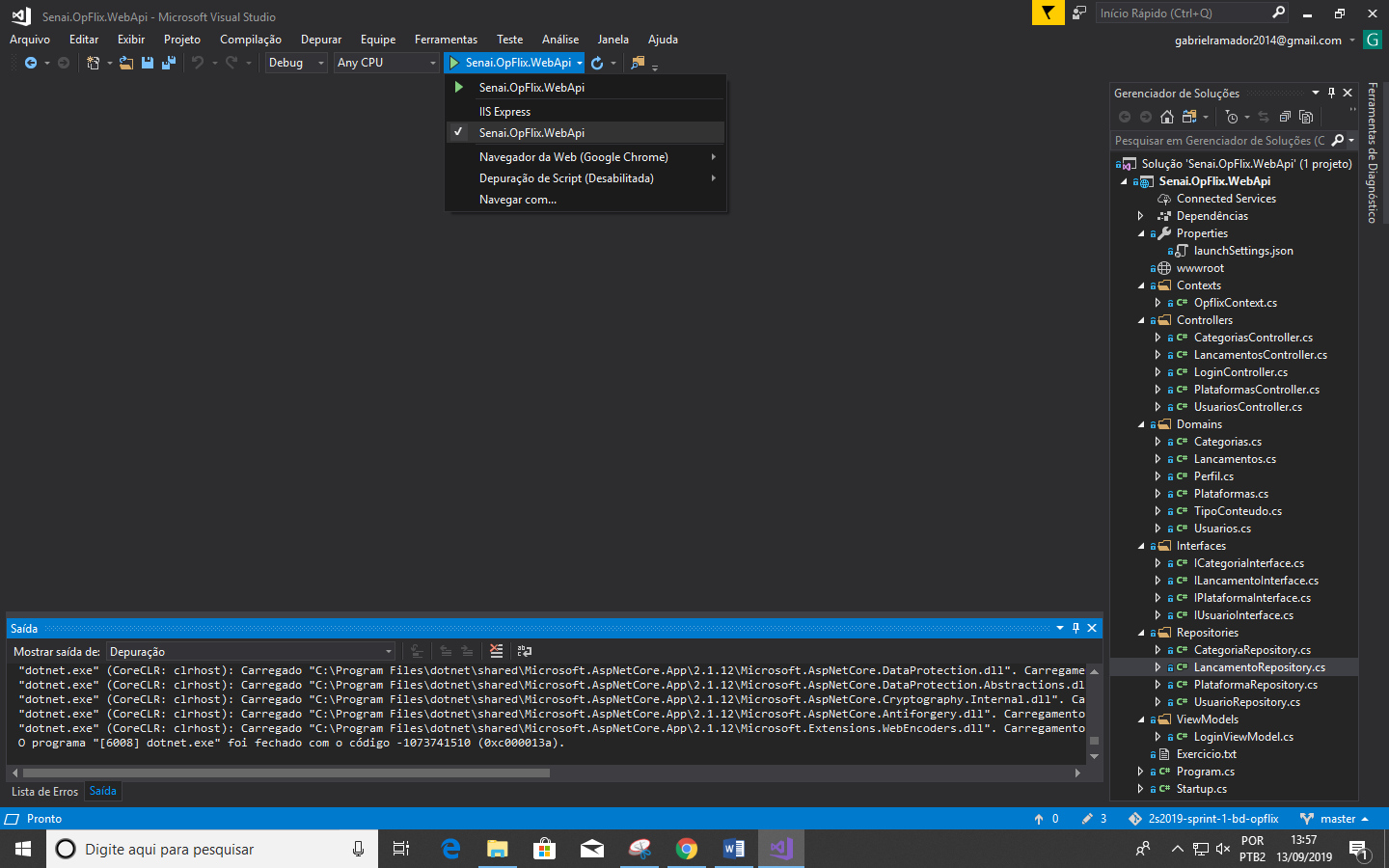
2º - Novamente abra o GitHub e acesse o repositório “[2s2019-sprint-2-backend/OpFlix](https://github.com/Schorsch003/2s2019-sprint-2-backend/tree/master/Senai.OpFlix.WebApi)” execute o arquivo “Senai.OpFlix.WebApi.sln” no Visual Studio.

3º - Com o projeto aberto, pressione a tecla F5 ou F11 e o projeto é executado.

Swagger

Acessando

**1º -** Dentro do Microsoft Visual Studio, pressione a tecla F5 ou no menu superior, selecione a opção de executar, lembrando de mudar o modo de execução para o projeto.



**2º -** Após a execução, abra um navegador de preferência e acesse a URL : “<http://localhost:5000/swagger/index.html>”;

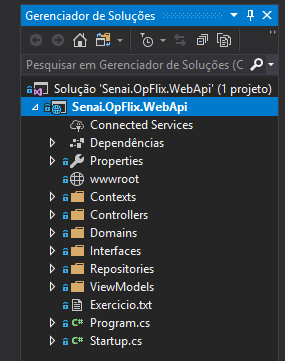
Pacotes Nuget

Segue uma lista de pacotes utilizados durante a criação do projeto

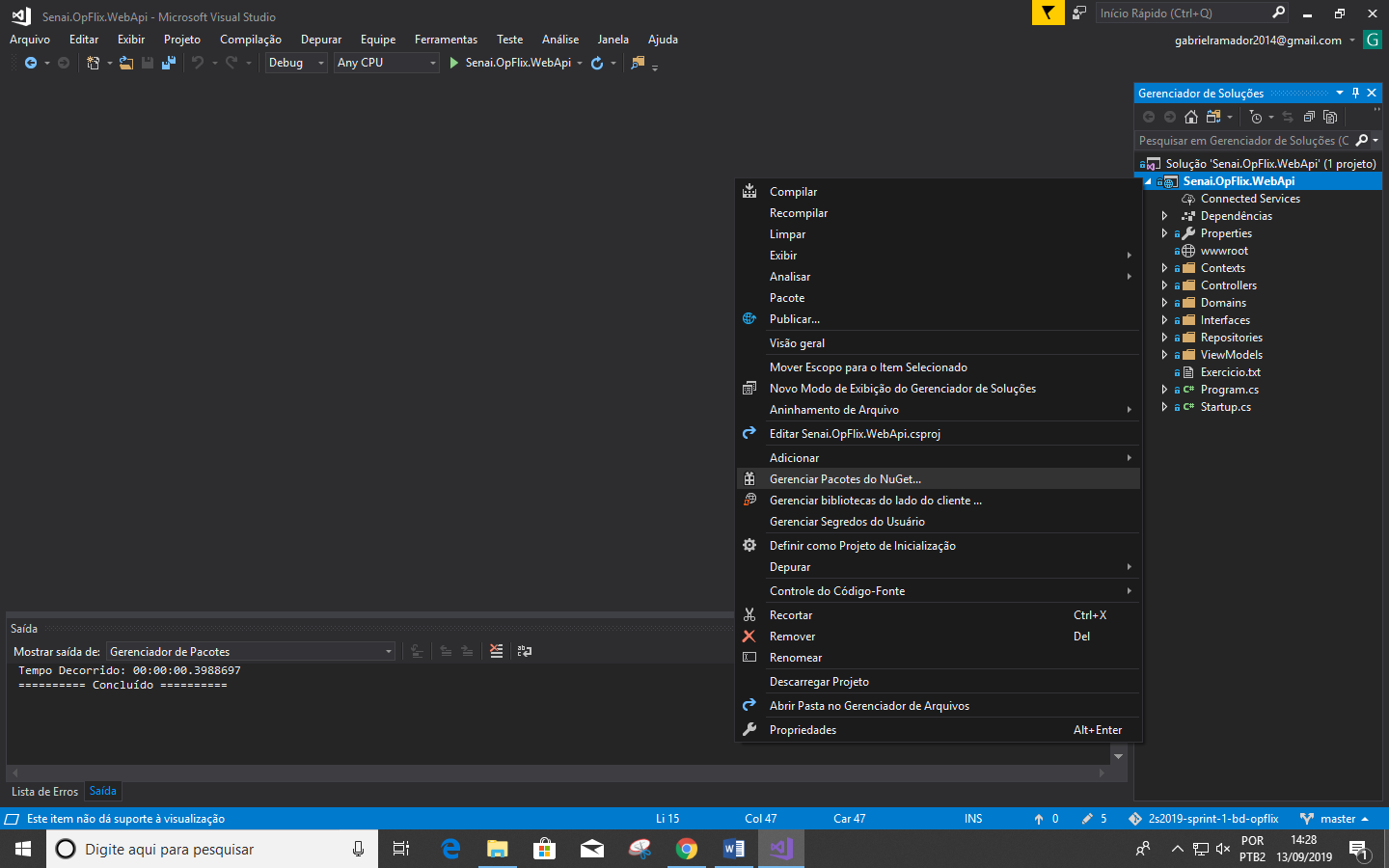
* **Swagger:**
  + Swashbuckle.AspNetCore Versão 4.0.1
* **SqlClient:**
  + System.Data.SqlClient Versão 4.6.1
* **Entity Framework Core:**
  + Microsoft.EntityFrameworkCore.SqlServer 2.1.11
  + Microsoft.EntityFrameworkCore.SqlServer.Design 1.1.6
  + Microsoft.EntityFrameworkCore.Tools 2.1.11
* **JSON Web Token:**
  + Microsoft.AspNetCore.Authentication.JwtBearer 2.1.1
  + System.IdentityModel.Tokens.Jwt 5.5.0

Como Instalar

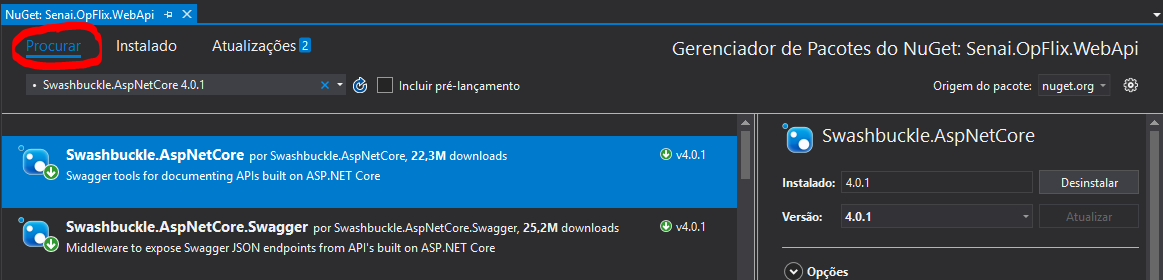
**1º -** Com o projeto aberto, ao lado direito da tela, na janela “**Gerenciador de Soluções**”, com o botão direito do mouse selecione a opção com o nome do projeto;



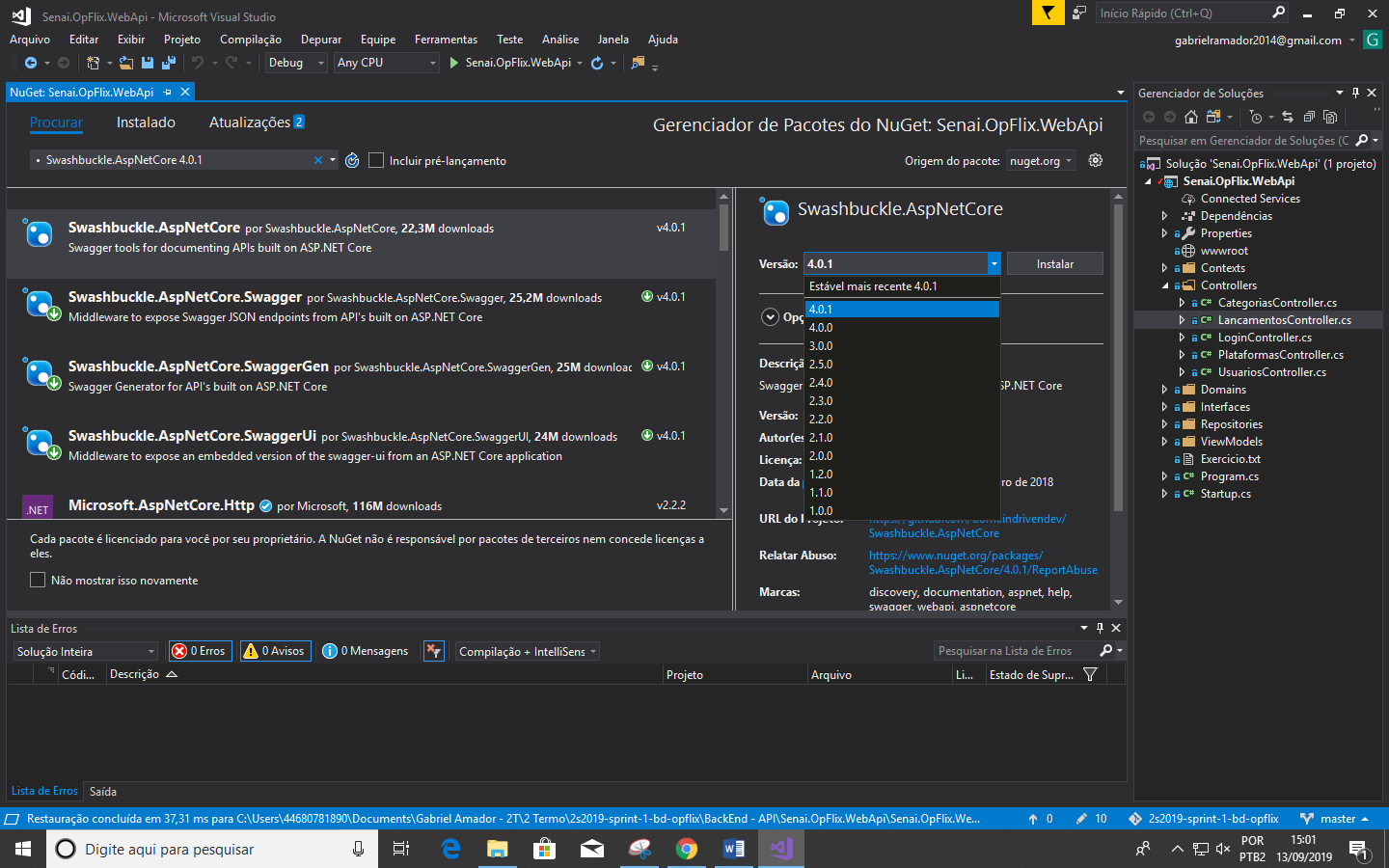
2º - Dentre as opções mostradas, escolha “**Gerenciar Pacotes do Nuget”**



**3º -** Com a nova janela aberta, acesse a opção “Procurar”, e na barra de pesquisa digite por vez, os pacotes indicados na sessão acima.;



**4º -** Selecionado o pacote, escolha a versão desejada e instale-o.



Postman

O Postman é um programa que nos permite testar o projeto enviando as requisições que serão utilizadas, trazendo o resultado em sua tela

Instalação

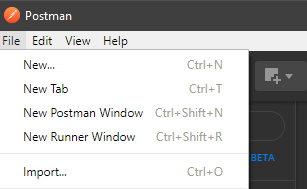
Acesse o site [www.getpostman.com/downloads](http://www.getpostman.com/downloads) e selecione a opção de download que corresponde ao seu sistema operacional.



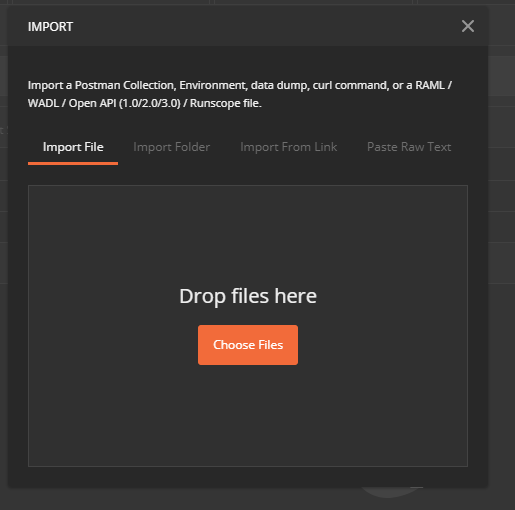
Importação

1º - Para importar as requisições utilizadas no projeto, acesse novamente o [repositório](https://github.com/Schorsch003/2s2019-sprint-2-backend/tree/master/Senai.OpFlix.WebApi/Senai.OpFlix.WebApi) e baixe o arquivo “Senai.OpFlix.postman\_collection.json”.

2º - Dentro do Postman, selecione a opção “File” e em seguida “Import” ou utilize a tecla de atalho Ctrl + O



3° - Selecione a opção “Import File” e use o navegador para encontrar o arquivo que foi trazido do repositório.



4º - Após isso aperte o botão de confirmação e todas as requisições feitas serão importadas para a sua plataforma.

Funcionalidades

Web

### Perfis de usuário:

* **Administrador**: Para o colaborador da gestão administrativa;
* **Cliente**: Clientes da empresa;

### Funcionalidades

1. O administrador poderá cadastrar qualquer tipo de **usuário (administrador ou cliente)**;
2. O administrador poderá cadastrar categorias (contendo nome);
3. O administrador poderá cadastrar o lançamento de um filme/série (contendo título, uma pequena sinopse, a categoria vinculada ~poderá ter somente uma vinculada~, tempo de duração, e se é filme ou série e a data do primeiro lançamento daquele item);
4. O cliente poderá visualizar todos os lançamentos publicados;

Mobile

### Perfis de usuário:

* **Administrador**: Para o colaborador da gestão administrativa;
* **Cliente**: Clientes da empresa;

### Funcionalidades

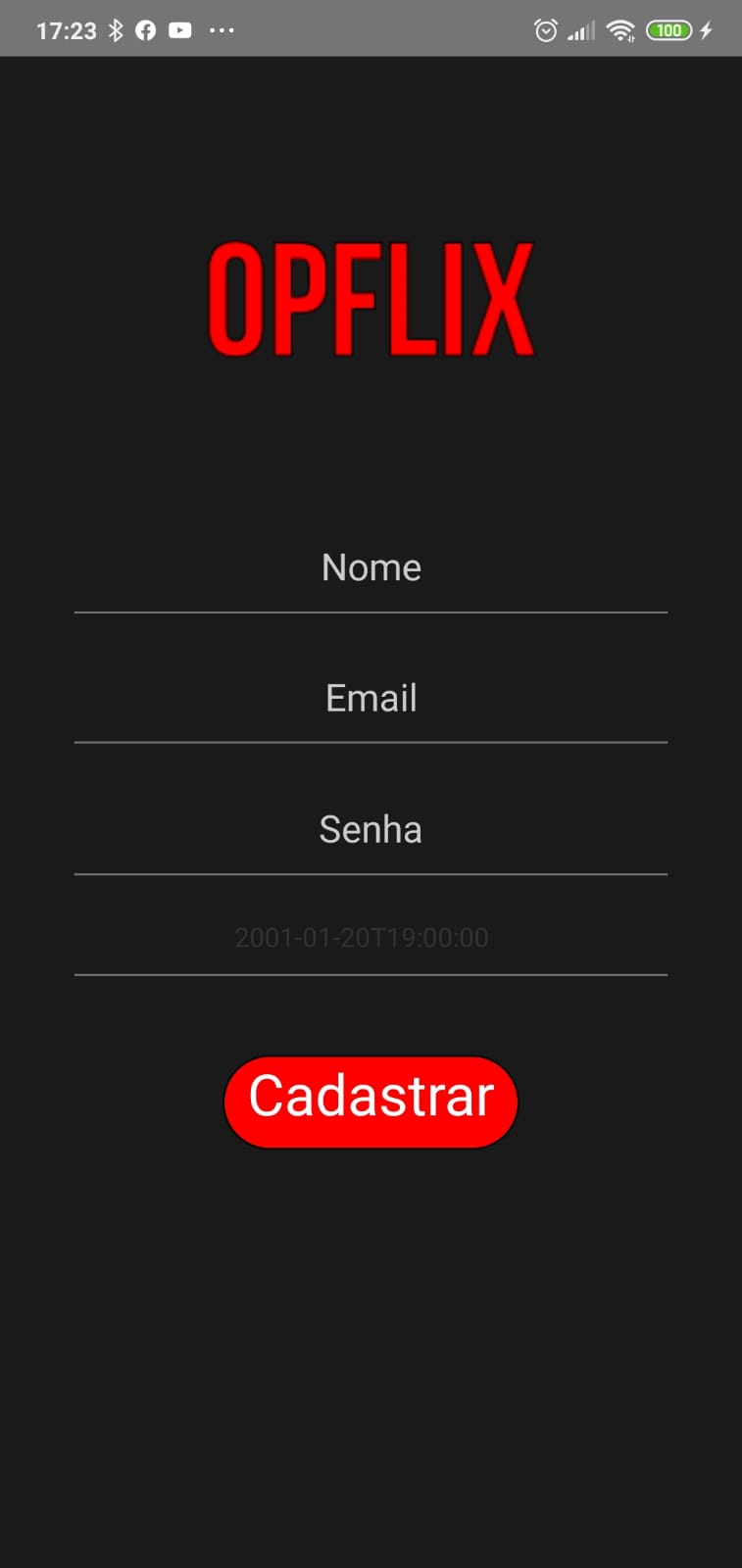
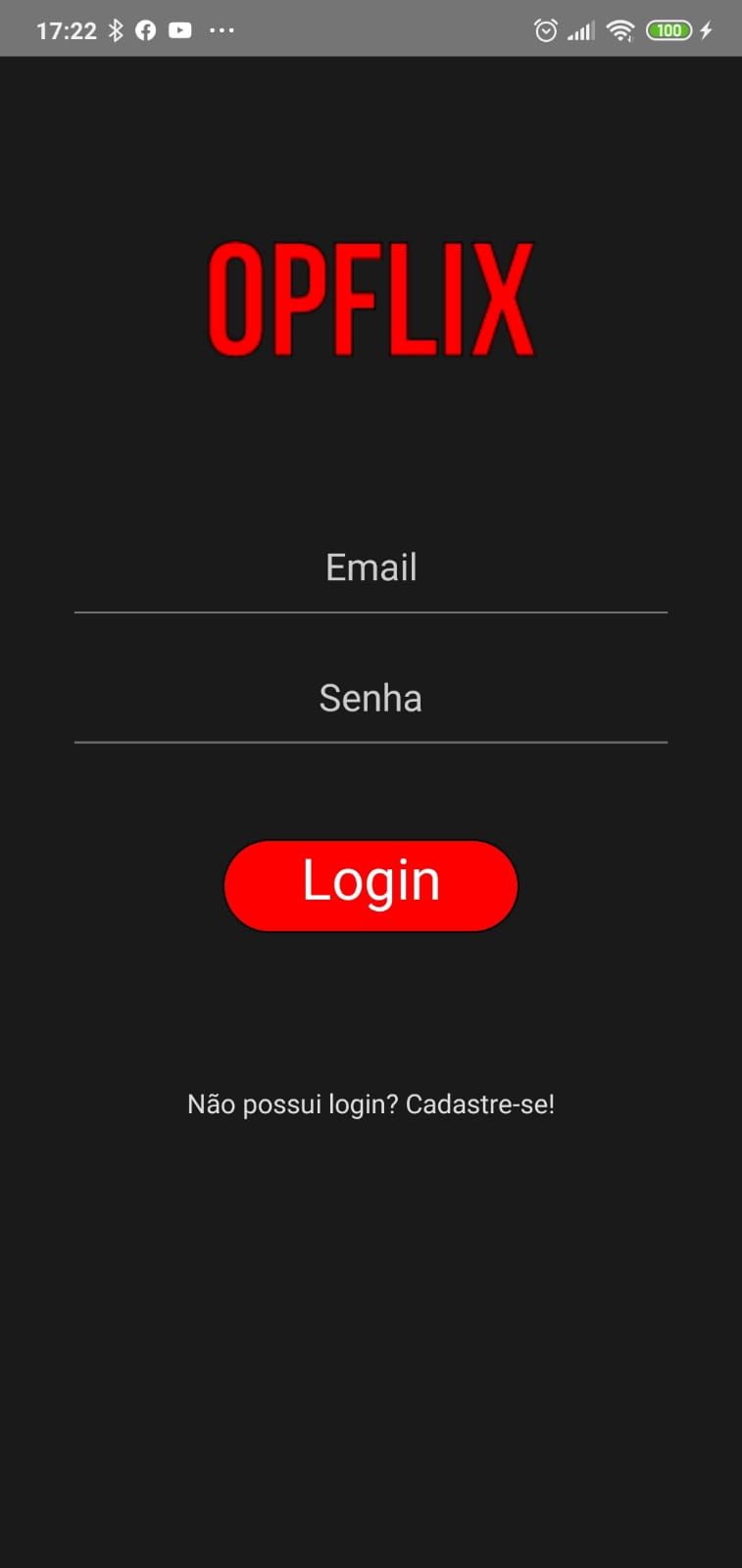
1. O cliente poderá visualizar todos os lançamentos;
2. O cliente poderá realizar a ordenação por data de lançamento e/ou categoria;

Protótipos

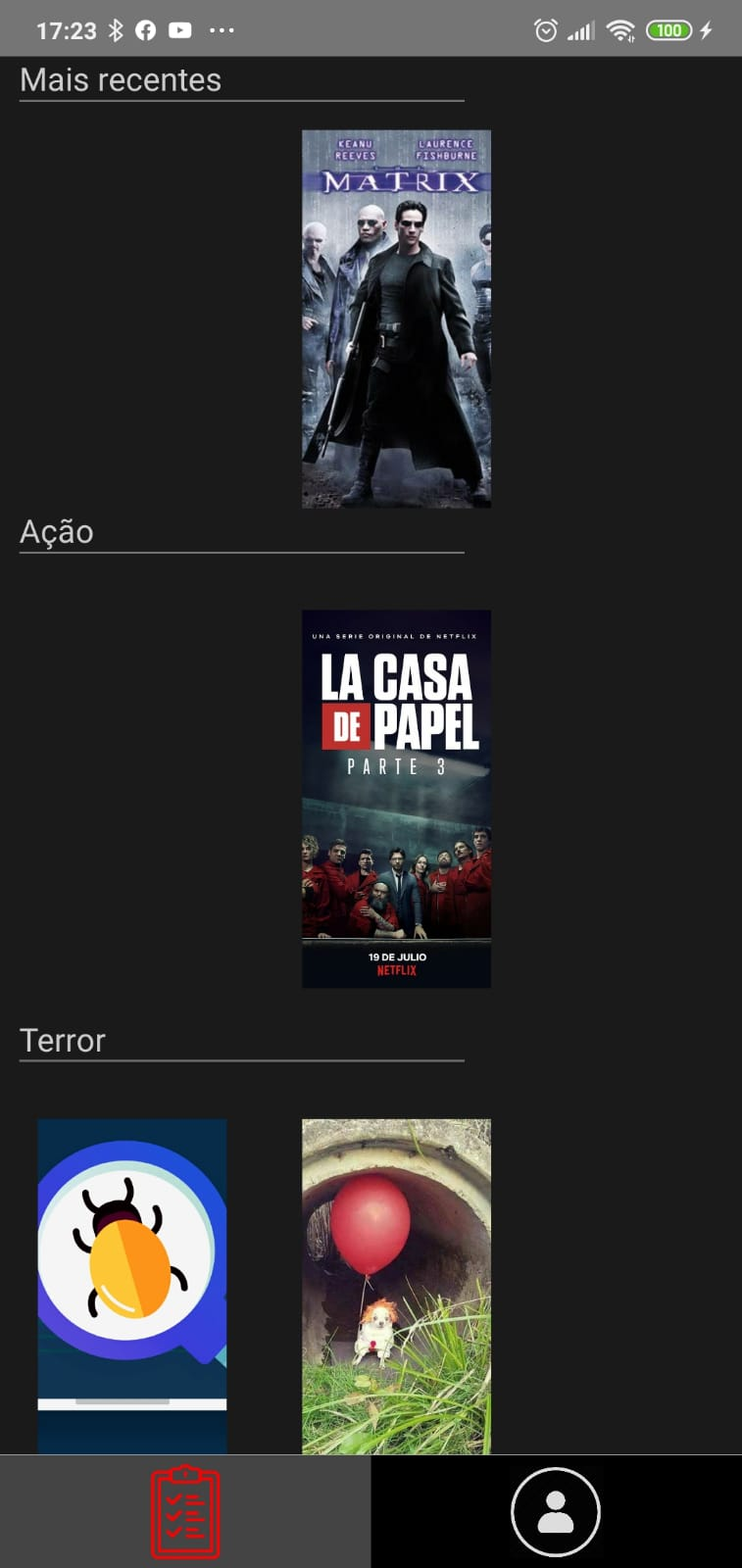
Web

Mobile

* **Login e Cadastro:**

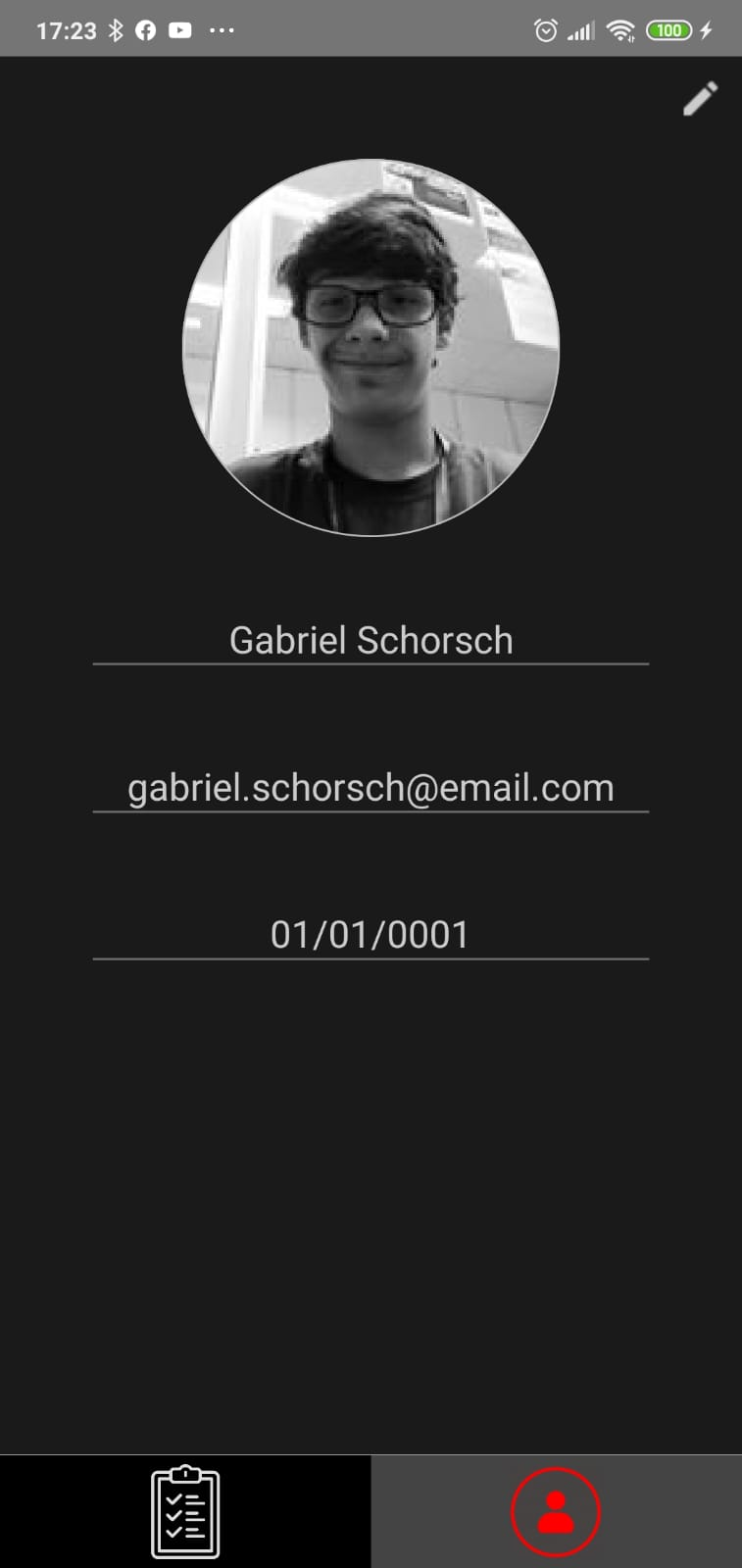


* **Home** 
  + **Listagem Lançamentos e Favoritos**



* + **Lançamento Selecionado (favorizado / não favorizado)**



* + **Tela Usuário/ Perfil**

Front-End

Mobile

Arquitetura do Projeto

Referências

Links

<https://github.com/Schorsch003/2s2019-sprint-1-bd/tree/master/OpFlix>

<https://www.w3schools.com/sql/>

<https://docs.microsoft.com/pt-br/sql/t-sql/language-reference?view=sql-server-2017>

Livros