

Documentação

OpFlix

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

Sumário

1. [Resumo 2](#_Toc25759273)

[Objetivos](#_Toc25759274)

1. [Descrição do projeto 2](#_Toc25759275)

[Resumo do projeto](#_Toc25759276)

1. [Ferramentas utilizadas 2](#_Toc25759277)
2. [Modelagem de Software 2](#_Toc25759278)

[Modelo Lógico](#_Toc25759279)

[Modelo Físico](#_Toc25759280)

[Modelo Conceitual](#_Toc25759281)

[Cronograma](#_Toc25759282)

1. [Criando Scripts do Banco de Dados 5](#_Toc25759283)

[Notas](#_Toc25759284)

[Criando o Script](#_Toc25759285)

[Rodando o Script](#_Toc25759286)

1. [Deploy / Rodar Projeto 6](#_Toc25759287)
2. [Swagger 7](#_Toc25759288)

[Acessando](#_Toc25759289)

1. [Pacotes Nuget 7](#_Toc25759290)

[Pacotes Necessários](#_Toc25759291)

[Como instalar](#_Toc25759292)

1. [Postman 9](#_Toc25759293)

[Importando](#_Toc25759294)

1. [Front-End 12](#_Toc25759295)
2. [Mobile 15](#_Toc25759296)
3. [Arquitetura do Projeto 16](#_Toc25759297)
4. [Referências 17](#_Toc25759298)

[Links](#_Toc25759299)

[Livros](#_Toc25759300)

Resumo

Objetivos

Este documento planeja a construção da aplicação Web e Mobile opflix

Descrição do projeto

Opflix é um projeto voltado para uma aplicação de lançamentos de series e filmes de várias plataformas;

Resumo do projeto

O projeto é a construção de aplicações Web/Mobile para lançamento de filmes e séries.

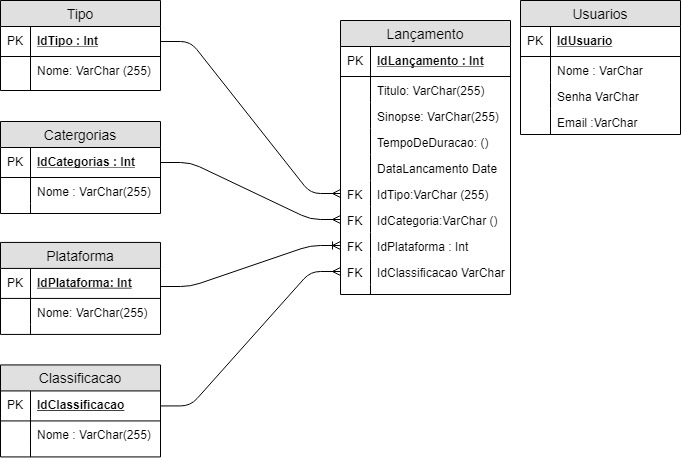
Ferramentas utilizadas

* BackEnd:
  + Microsoft Visual Studio
* Swagger:
  + Postman
* Banco de dados:
  + Microsoft SQL Server Management Studio

Modelagem de Software

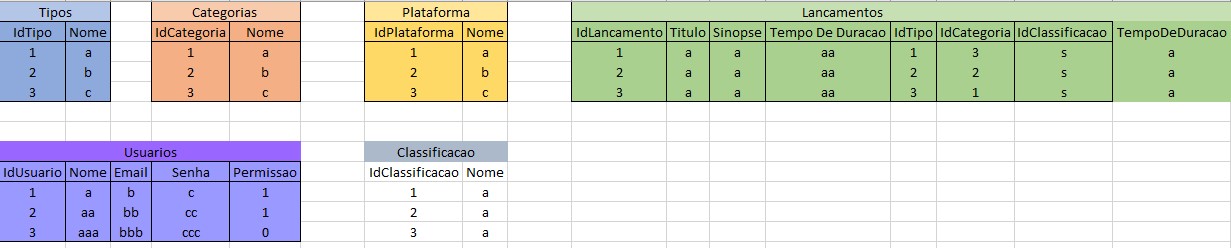
Atividade de construir modelos que expliquem as características ou o comportamento de um [software](https://pt.wikipedia.org/wiki/Software) ou de um [sistema de software](https://pt.wikipedia.org/wiki/Sistema_de_software). Na construção do software os modelos podem ser usados na identificação das características e funcionalidades que o software deverá prover ([análise de requisitos](https://pt.wikipedia.org/wiki/An%C3%A1lise_de_requisitos)), e no planejamento de sua construção[[](https://pt.wikipedia.org/wiki/Modelagem_de_software#cite_note-1)

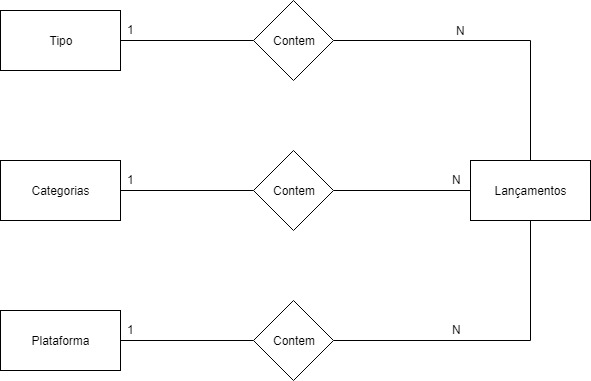
Modelo Lógico

Modelo lógico é onde possui a relação das entidades e os seus dados

Modelo Físico

Modelo Físico ou Modelo Excel, possui os dados das entidades e se assemelha ao resultado final do código.





Modelo Conceitual

O Modelo conceitual onde ligamos todas as entidades de forma resumida.

Cronograma

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

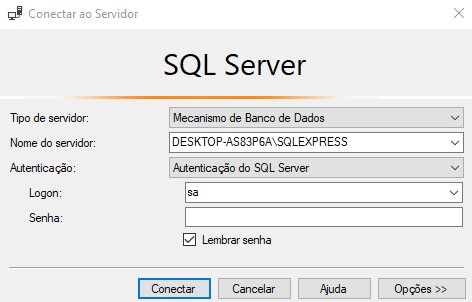
Criando Scripts do Banco de Dados

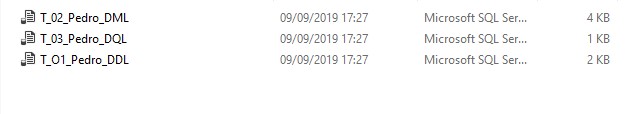
Notas

* Antes de rodar a aplicação backend é necessário ter o banco de dados criado, e sem problemas.
* É necessário ter instalado a aplicação Microsoft SQL Server Studio.

Criando o Script

**1º -** Abra a aplicação, atente-se no tipo de servidor e autenticação, o nome do servidor será diferente em cada máquina. Aperte o botão conectar;



**2º -** Ao se conectar, abra o diretório de pastas e encontre o caminho “2s2019-sprint-1-bd-opflix\Code Scripts”, antes lembre-se de clonar o repositório GitHub. Nesse diretório haverá três documentos, arraste e solte cada um deles no SQL Server

Rodando o Script

**1º** - Com a estrutura já criada, é preciso rodar o projeto. Selecione todos os arquivos com exceção do terceiro e clique em F5, então o projeto estará criado com dados salvos.

Deploy / Rodar Projeto

1º - Abra o Visual Studio;

2º - No diretório de pastas encontre o projeto “2s2019-T2-backend” e acesse a pasta: “Senai.OpFlix.WebApi”\ e acesse: Senai.OpFlix.WebApi.sln”;

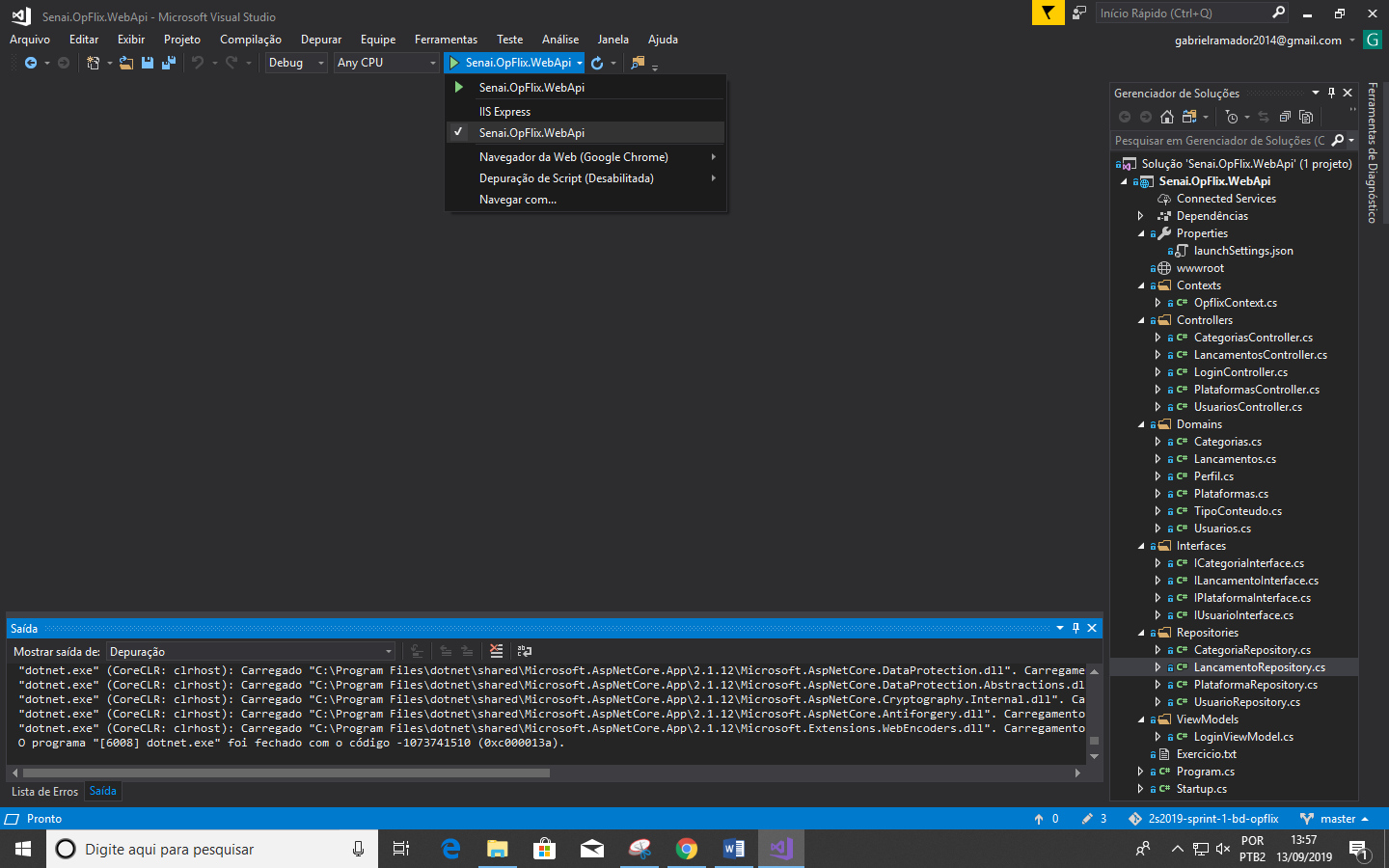
3º - Com um duplo clique no mouse abra o projeto “Senai.OpFlix.WebApi.sln”;

4º - Com o projeto aberto, pressione a tecla F5 ou F11 e o projeto é executado.

Swagger

Acessando

**1º -** Na plataforma Visual Studio (Microsoft) pressione a tecla F11 ou então com o mouse selecione a opção de execução na parte superior do ambiente de desenvolvimento.



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

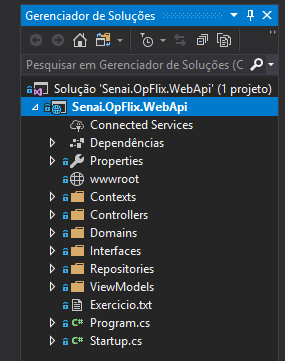
Pacotes Nuget

Pacotes Necessários

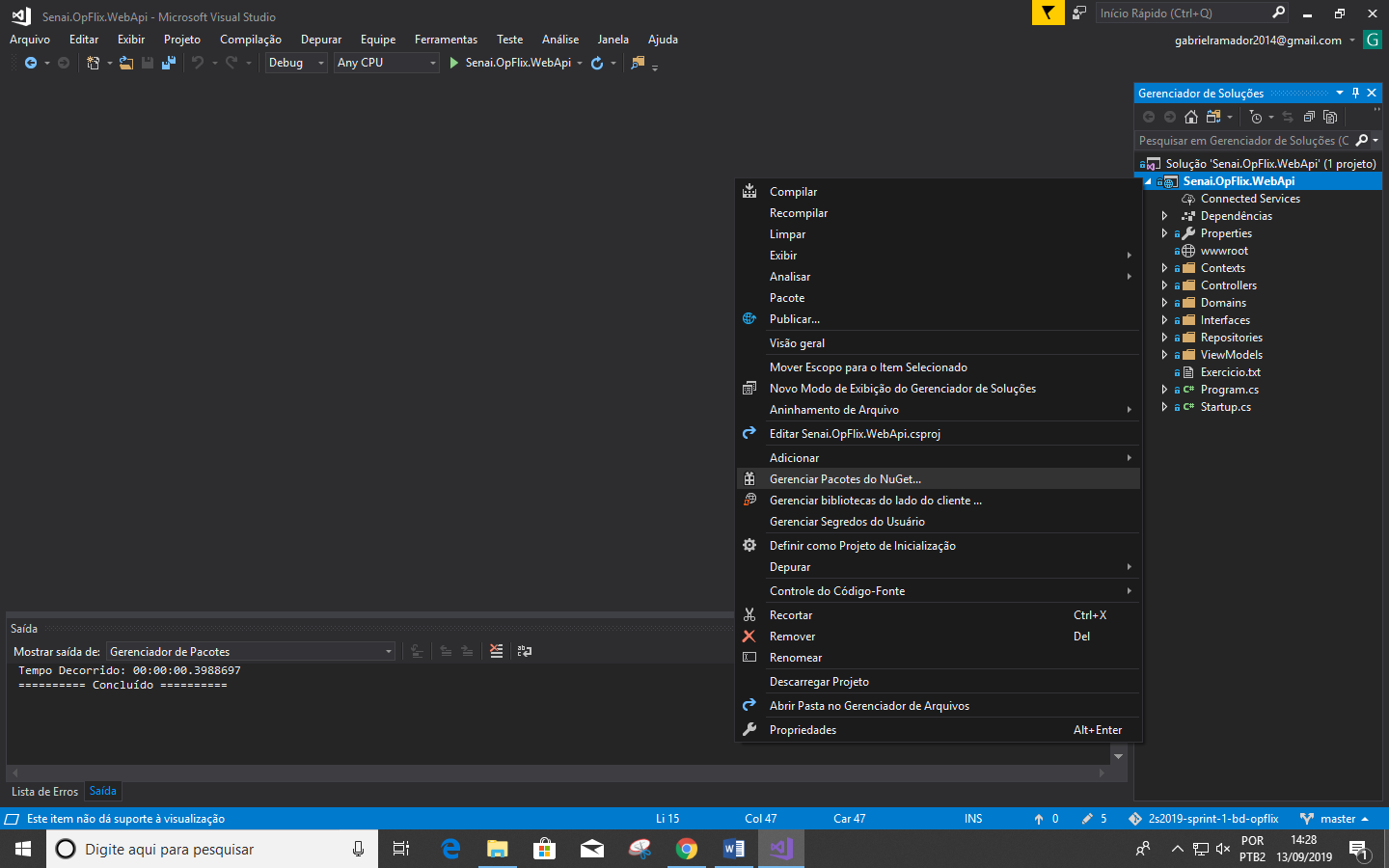
* **Uso Swagger:**
  + Swashbuckle.AspNetCore 4.0.1
* **Uso SqlClient:**
  + System.Data. SqlClient 4.6.1
* **Uso Entity Framework Core:**
  + Microsoft.EntityFrameworkCore. SqlServer 2.1.11
  + Microsoft.EntityFrameworkCore. SqlServer. Design 1.1.6
  + Microsoft.EntityFrameworkCore.Tools 2.1.11
* **Uso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

Como instalar

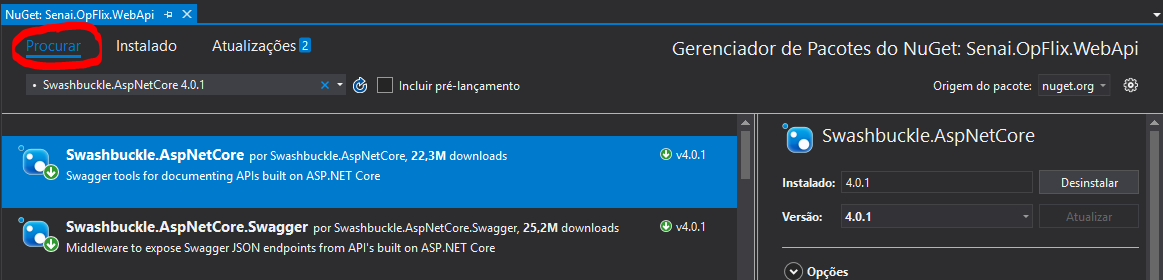
**1º -** Com o projeto aberto, ao lado direito da tela, na janela “**Gerenciador de Soluções**”, com o botão direito do mouse selecione a opção com o nome do projeto;



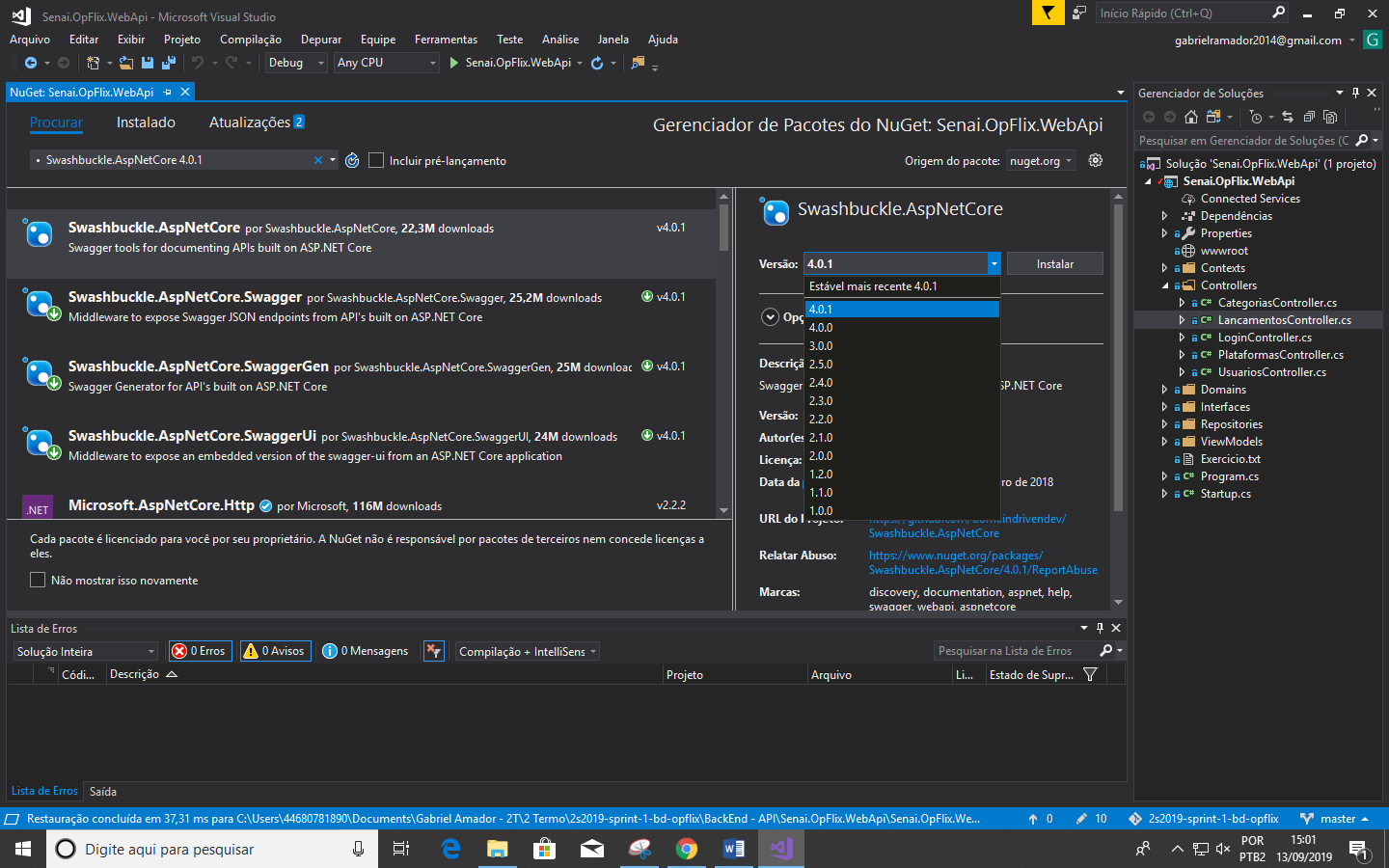
2º - Dentre as opções mostradas, escolha “**Gerenciar Pacotes do Nuget”**



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



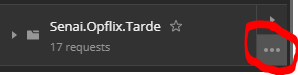
**4º -** Selecionado o pacote, escolha a versão desejada e instale-o.



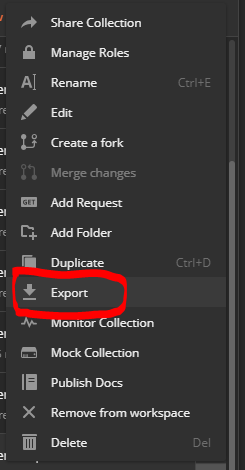
Postman

Exportando

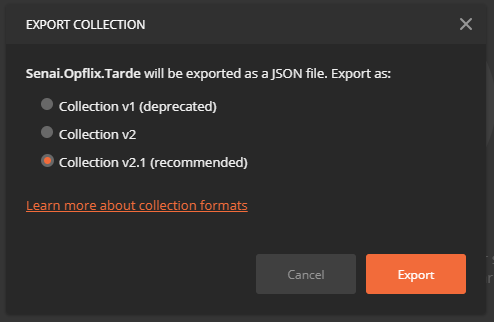
1º - Primeiramente, com o aplicativo instalado na máquina, ao lado esquerdo da tela selecione a pasta de EndPoints a ser exportada e clique no ícone circulado em vermelho;



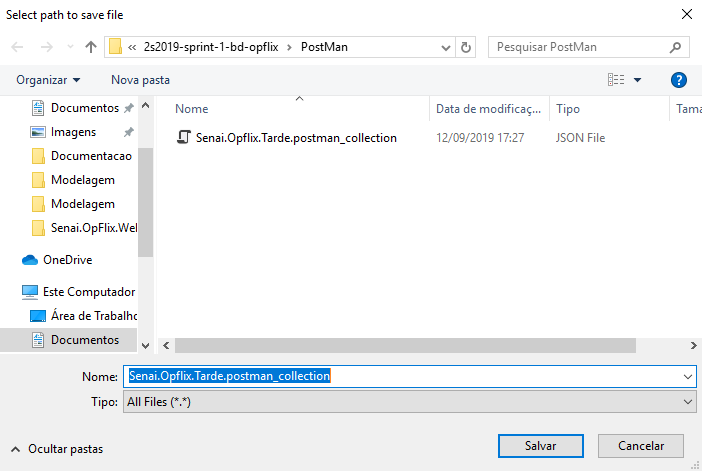
**2º -** No menu aberto, selecione a opção “Export”;



3º - Na aba de exportação, haverá três opções, ative a opção “Collection v2.1 (recommended)” e exporte o projeto;

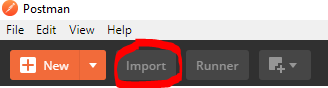


**4º -** Nesse ultimo passo, defina o nome do arquivo e o diretório onde será armazenado o projeto;

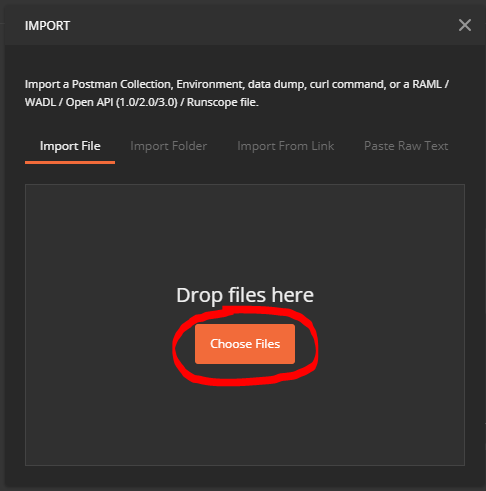


Importando

**1º -** Aberto o aplicativo Postman, selecione o botão “Import” no canto superior esquerdo da tela;

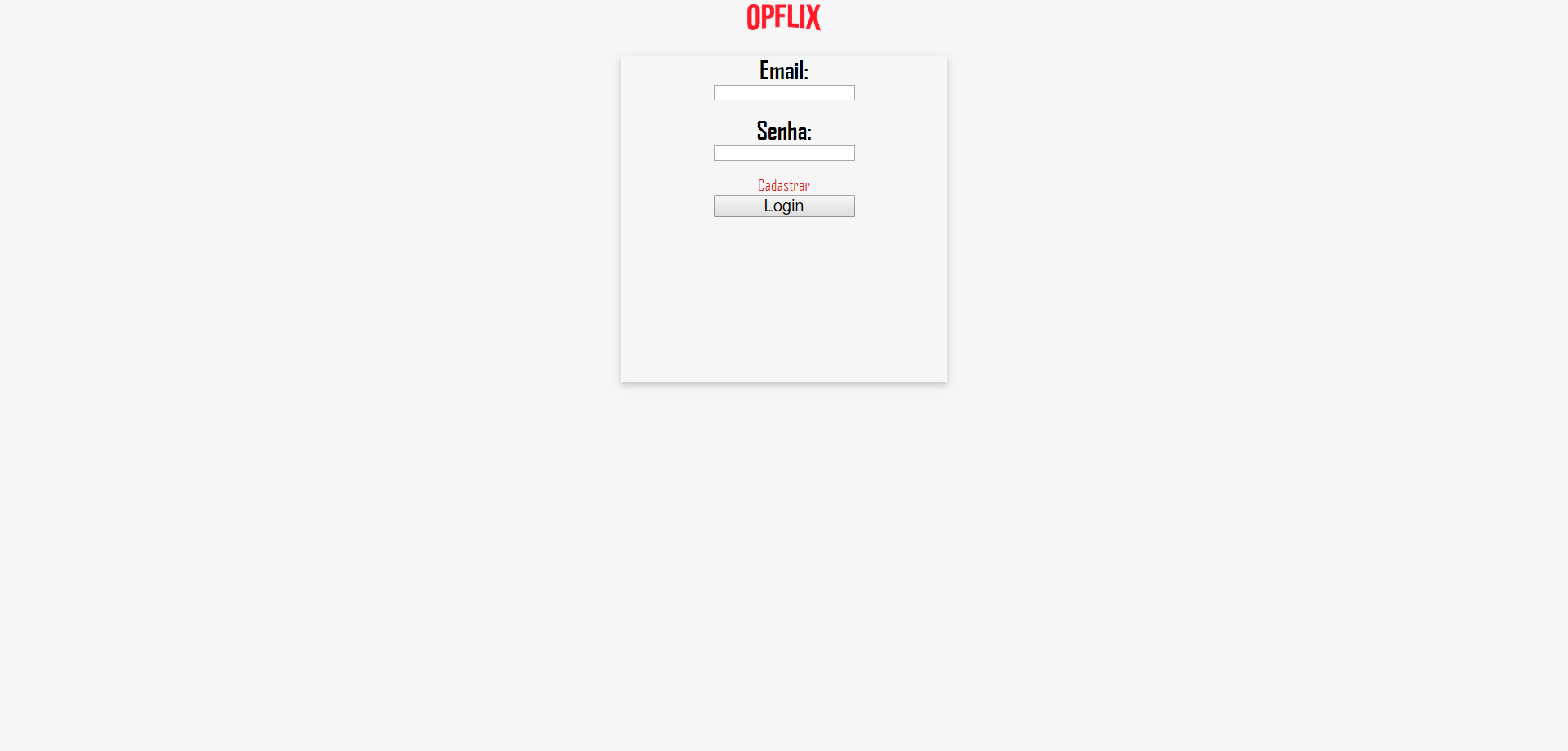


**2º -** Na nova aba, há dois caminhos ou arrastar e soltar o arquivo (arquivo de um export anterior) no local solicitado, ou selecionar o caminho do diretório. Respectivamente, pelo segundo caminho clique no botão “Choose Files”

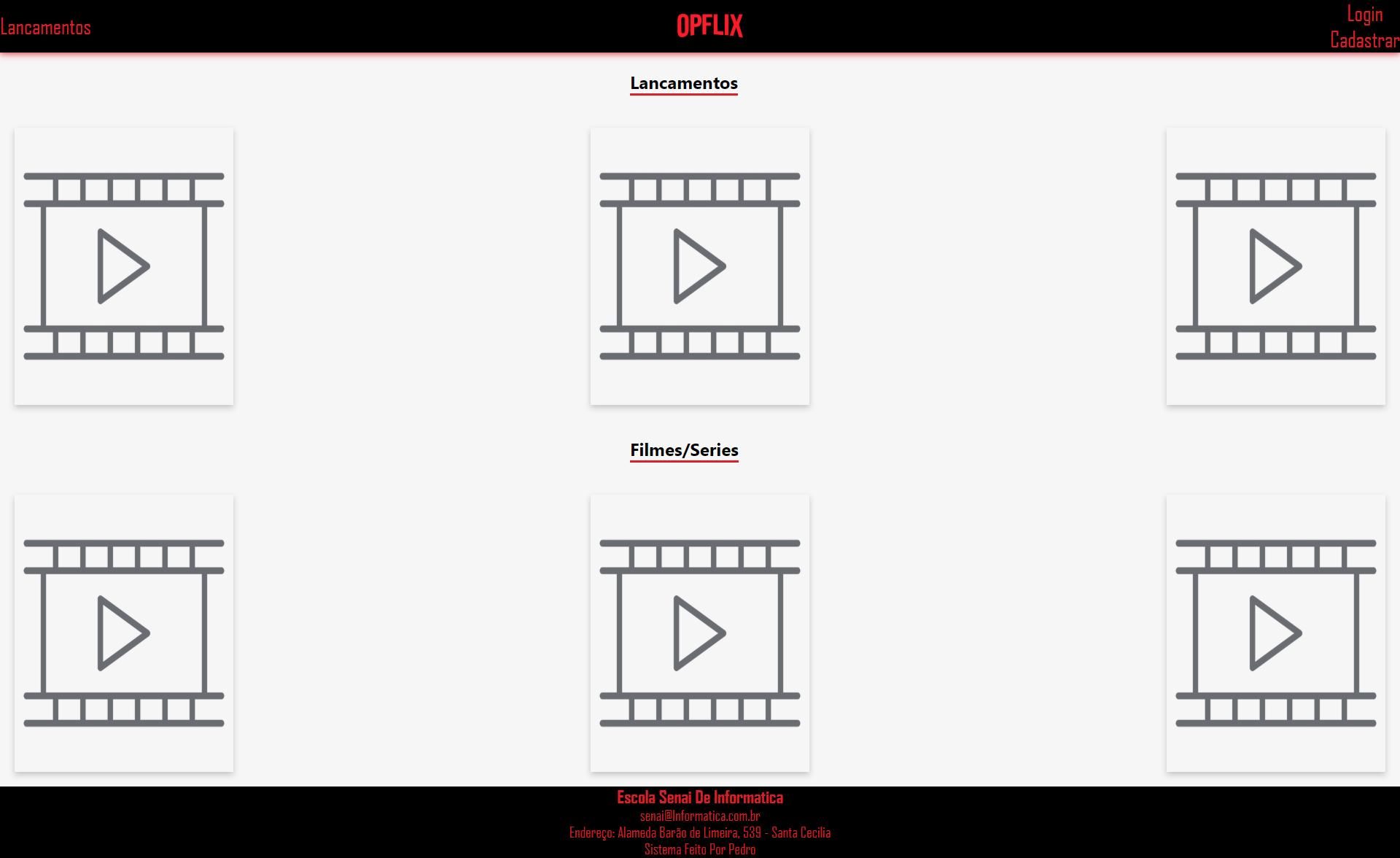


**3º -** Selecione o projeto e clique em abrir

Front-End

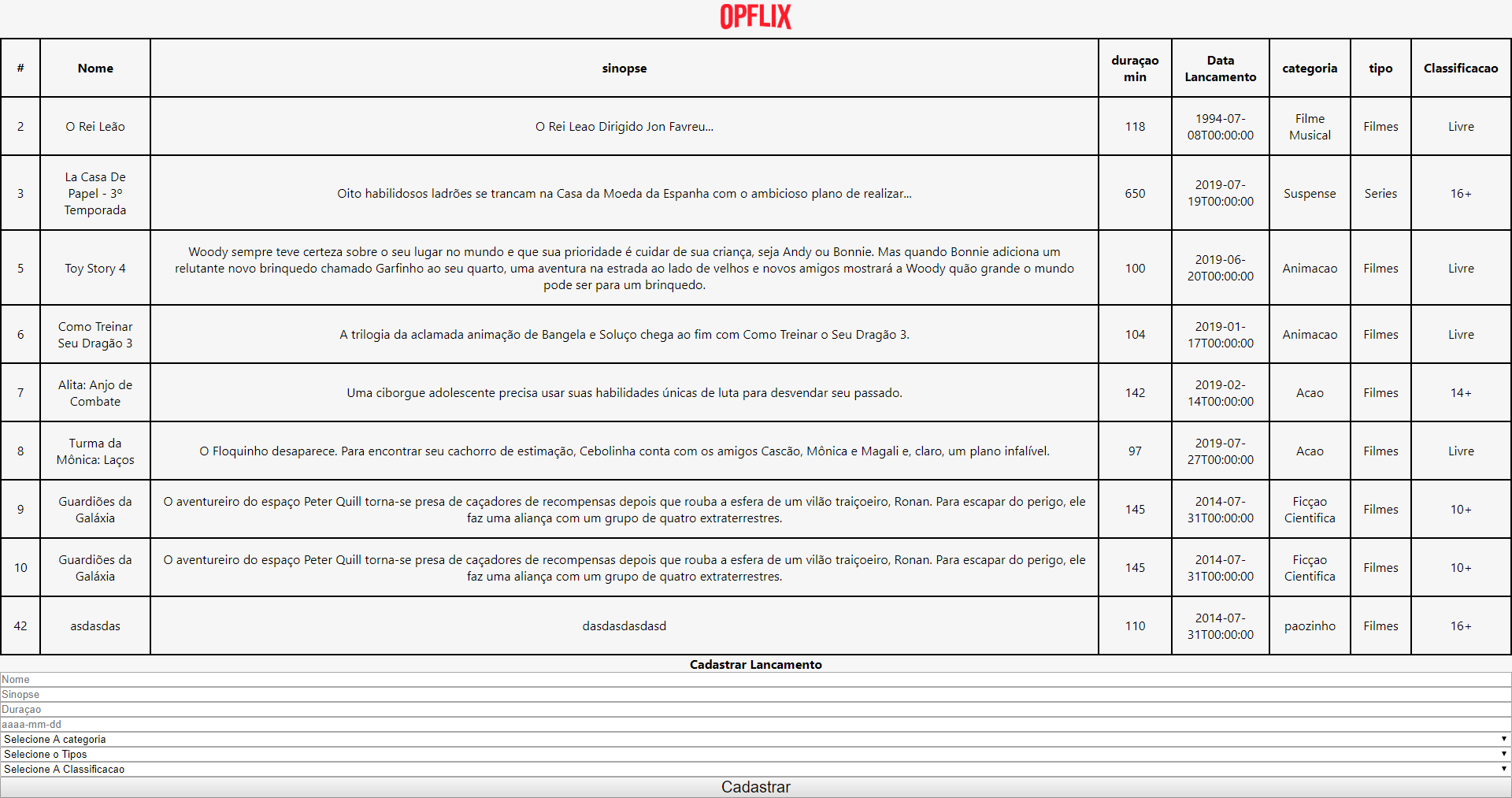


**Home:** A home ou página principal do site traz a experiência efetiva do cliente. Nesta página apenas o usuário logado pode acessa-la**.** Nela o usuário pode verificar a lista de lançamentos, filtrá-la, favorizar prediletos e realizar logout. A tipografia é uma característica realçada nesta página e cada lançamento possui informações próprias.

****

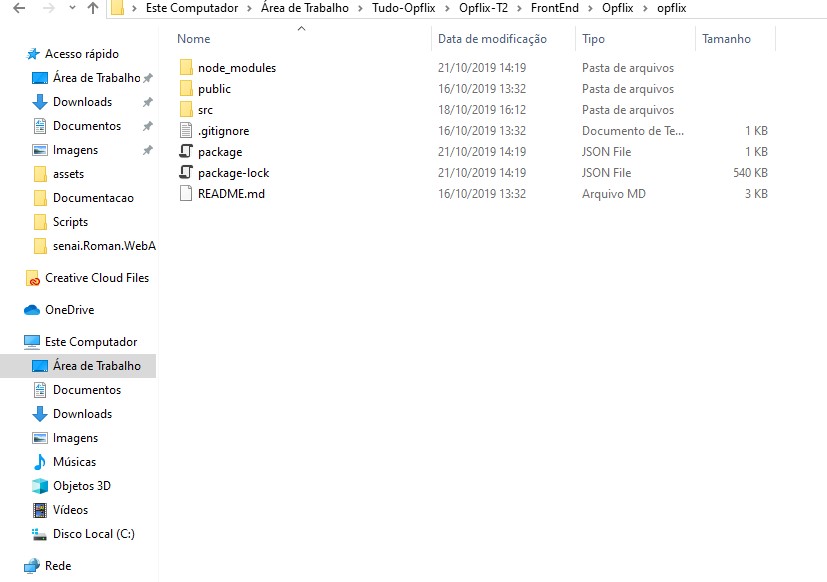
**Dashboard:**

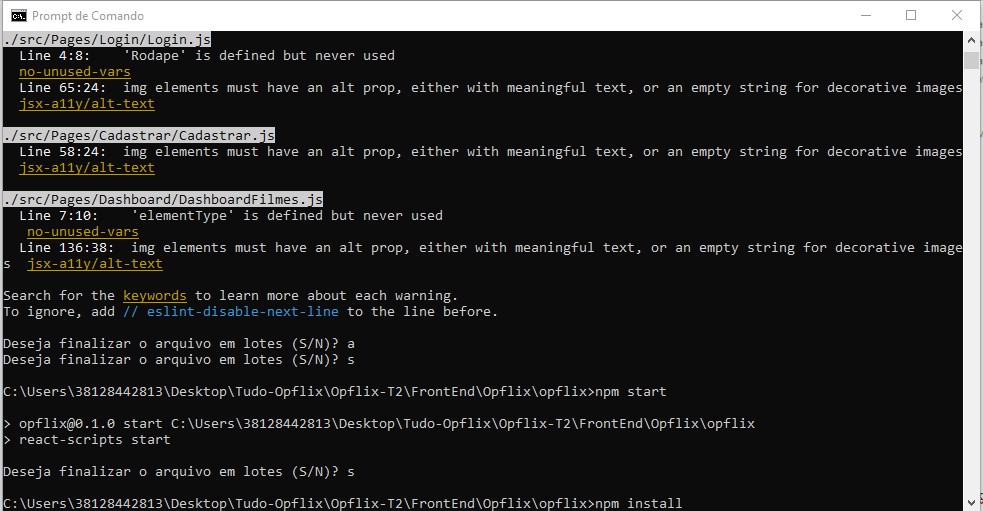
**Resumo:** Esta página é exclusiva do perfil administrador, ela é a mais extensa e com mais funcionalidades. Se baseia em cadastros, listagens e atualizações de dados.

****

**Abrir e instalar projeto.**

**1º -** Como primeiro passo acesse o link “<https://nodejs.org/en/>” e baixe a versão do node.js mais adequada com seu computador;

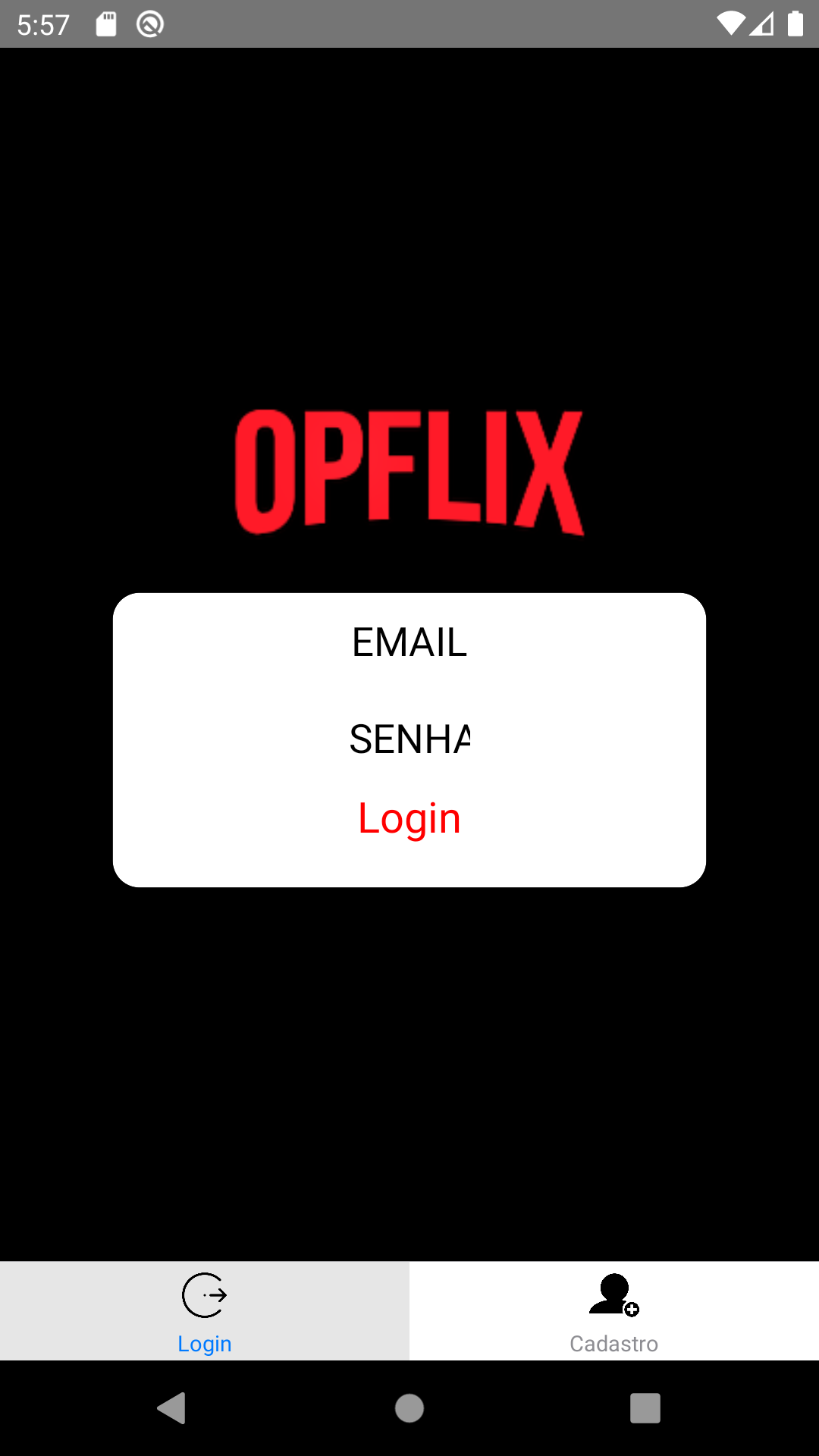
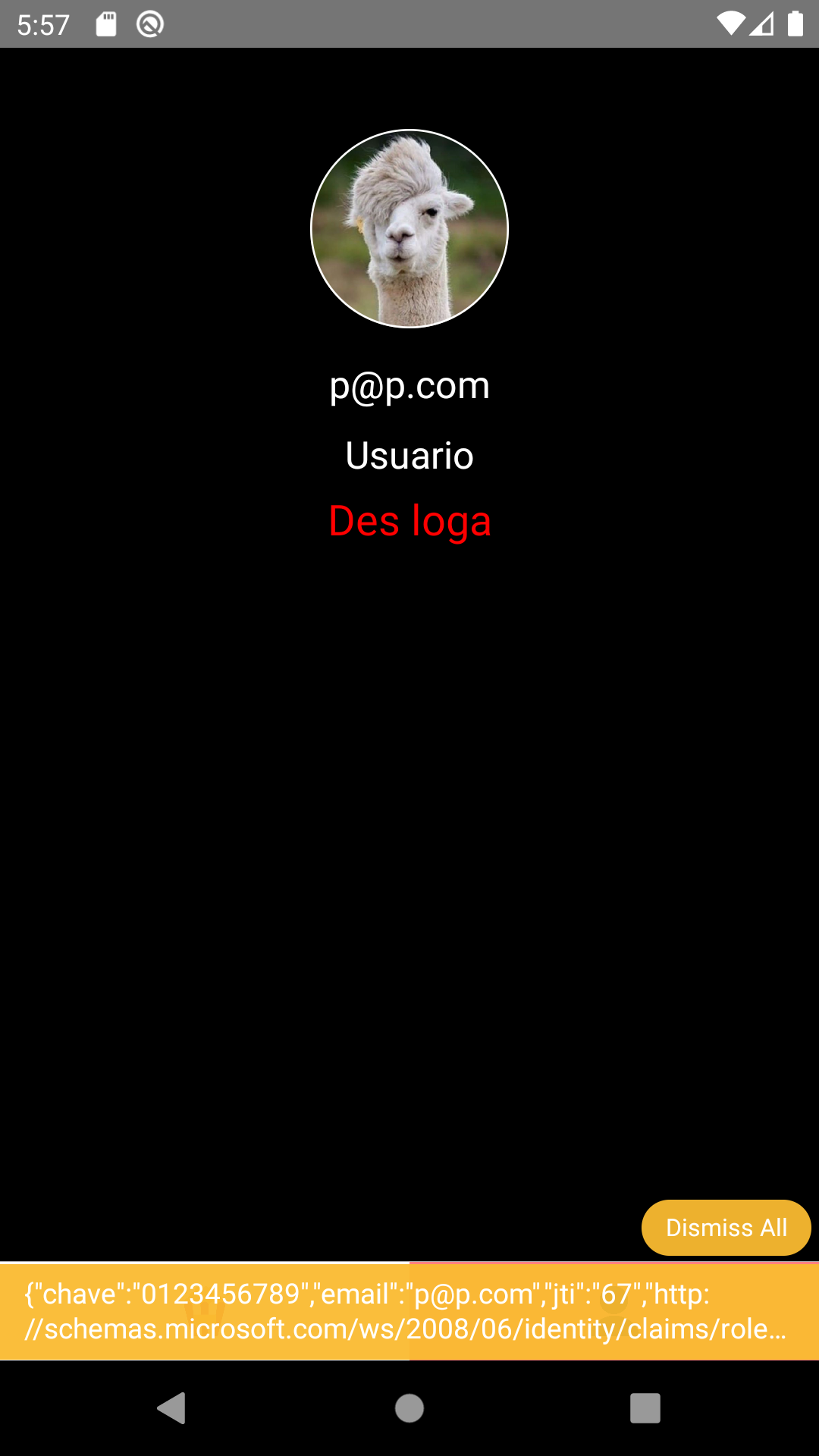
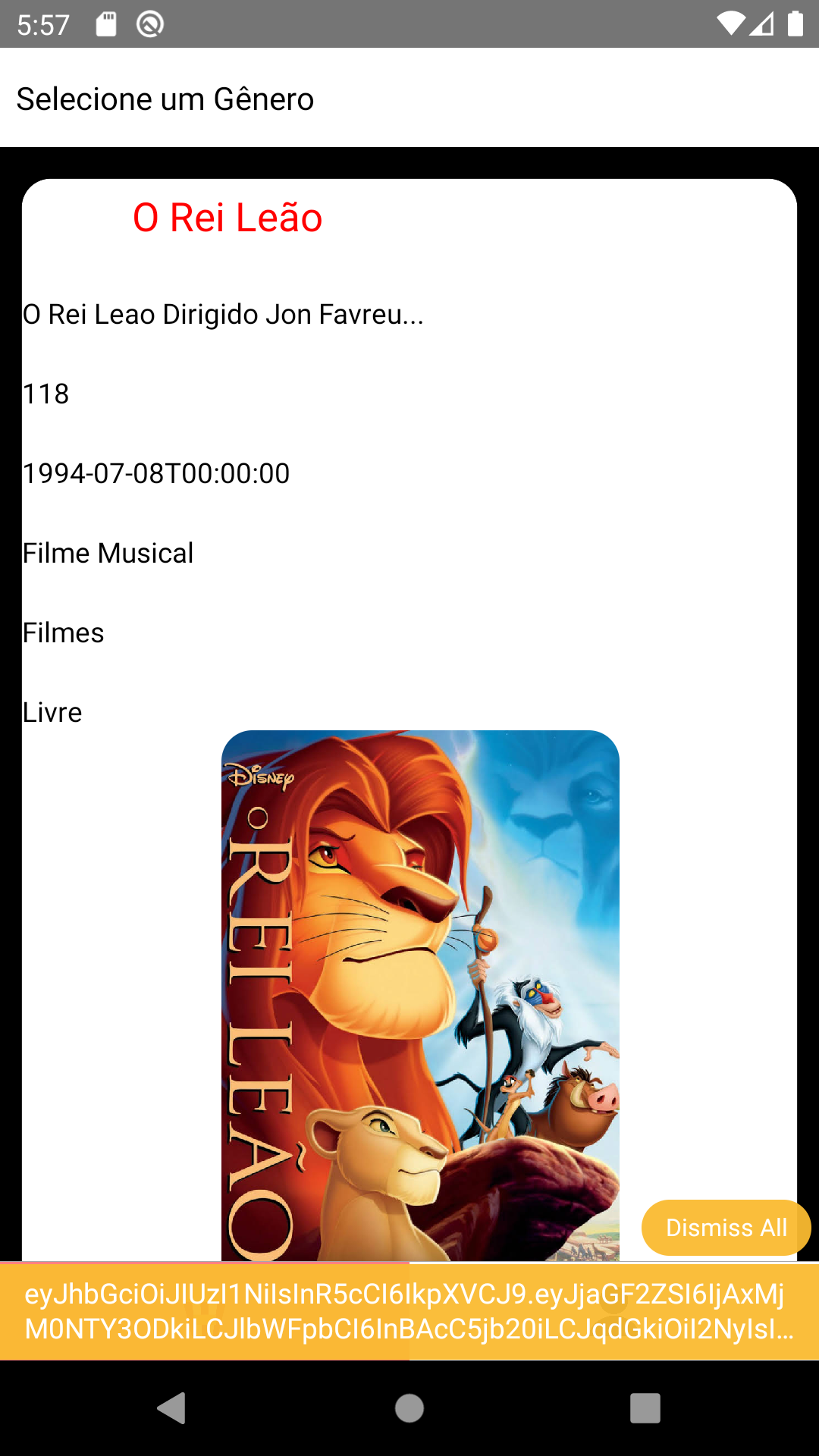
**2º -** Agora o repositório já clonado na pasta “2s2019-sprint-1-bd-opflix/FrontEnd/opflix-react/” digite como na imagem “cmd .” na barra de navegação e aperte a tecla ENTER;

****

**4º** - Por fim, digite “npm start” e espere, a partir daí o sistema estará rodando.

****

Mobile

Arquitetura do Projeto

Referências

Links

<https://github.com/Psouza-queiroz/2s2019-sprint-1-bd>

Livros