

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 informar aos usuários sobre os lançamentos de filmes e séries em diferentes plataformas.

Descrição do projeto

O OpFlix tem como objetivo informar aos usuários sobre os lançamentos de filmes e séries em diferentes plataformas.

Resumo do projeto

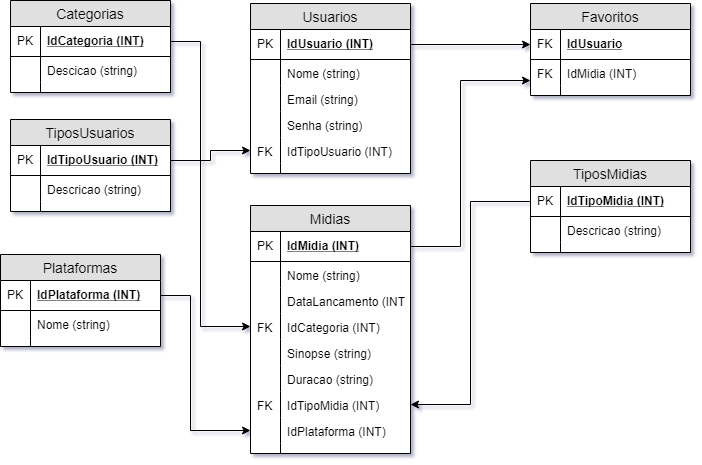
O OpFlix tem a disponibilização de filmes, séries e suas determinadas categorias. Assim, cada usuário poderá realizar o filtro das mídias que tem interesse, dentro da plataforma. E como item extra, poderá adicionar os que tem interesse, dentro de uma lista.

Modelagem de Software

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

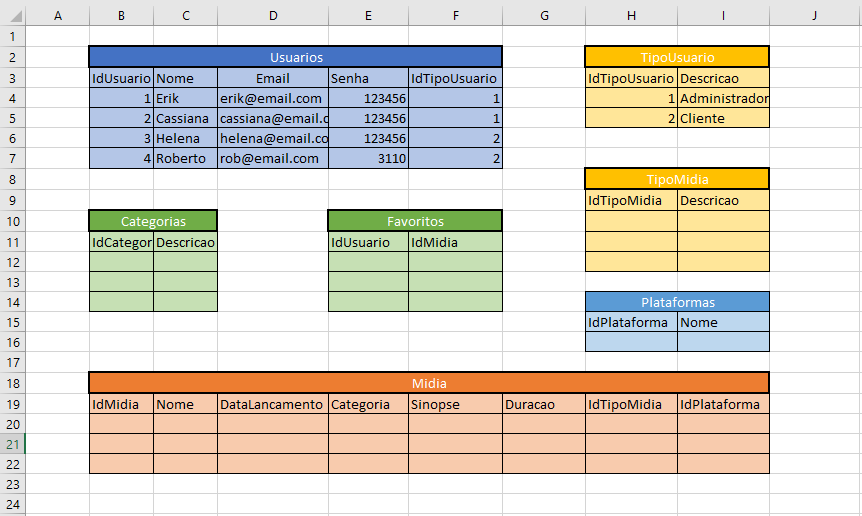
Modelo Lógico

O modelo lógico é modelo que mais aproxima-se da estrutura de um banco de dados.



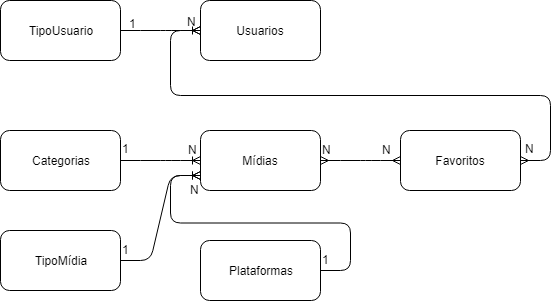
Modelo Físico

O modelo físico é um teste para validar os valores que serão inseridos.



Modelo Conceitual

O modelo conceitual é uma visualização macro das entidades e dos relacionamentos.



Cronograma

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

BACKEND

Como acessar o Swagger?

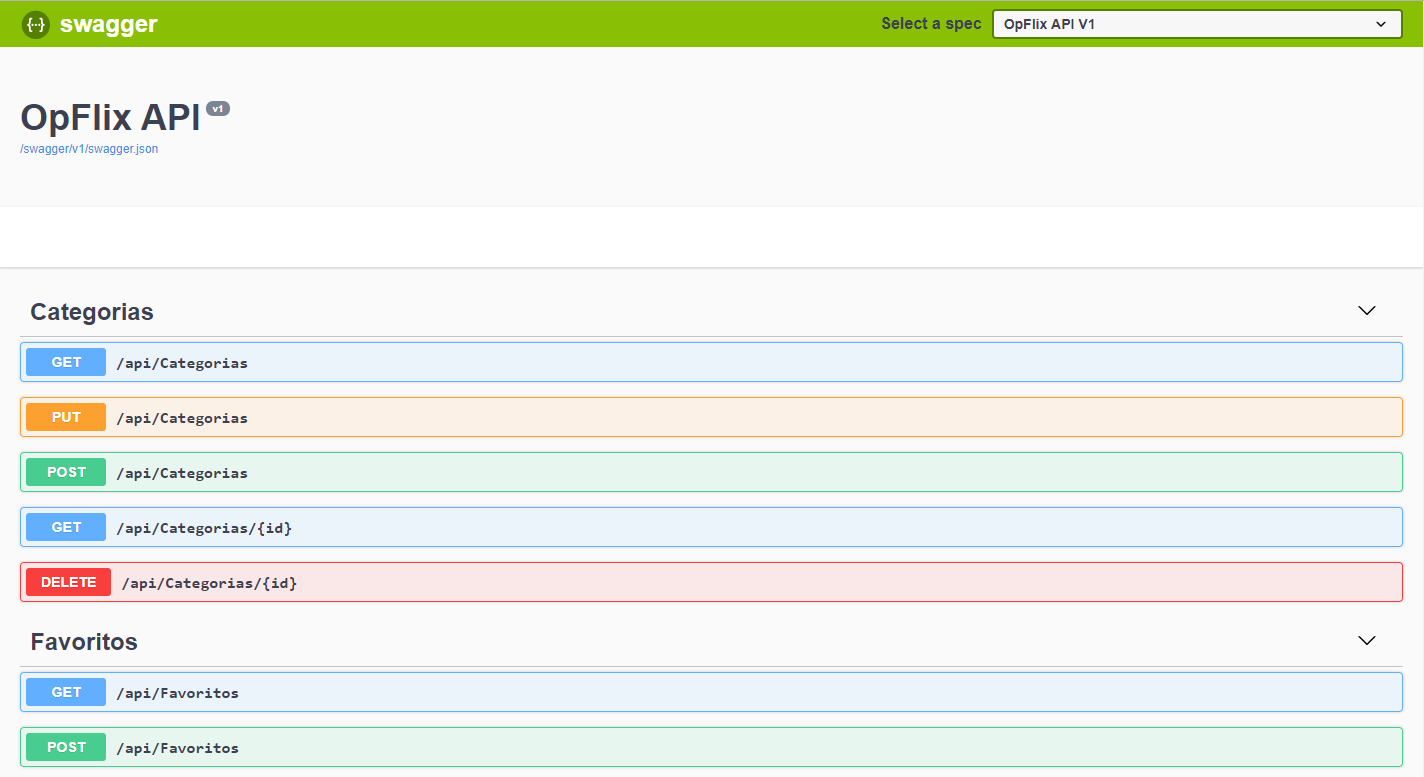
1. Abrir o projeto no Visual Studio 2017
2. Para rodar a aplicação, clique neste botão, localizado na parte superior da IDE:



1. Após a compilação, abra seu navegador e insira o seguinte endereço na barra de pesquisa:



1. Após abrir a página do Swagger, você conseguira visualizar todos os métodos do programa, suas funcionalidades e como testá-los.



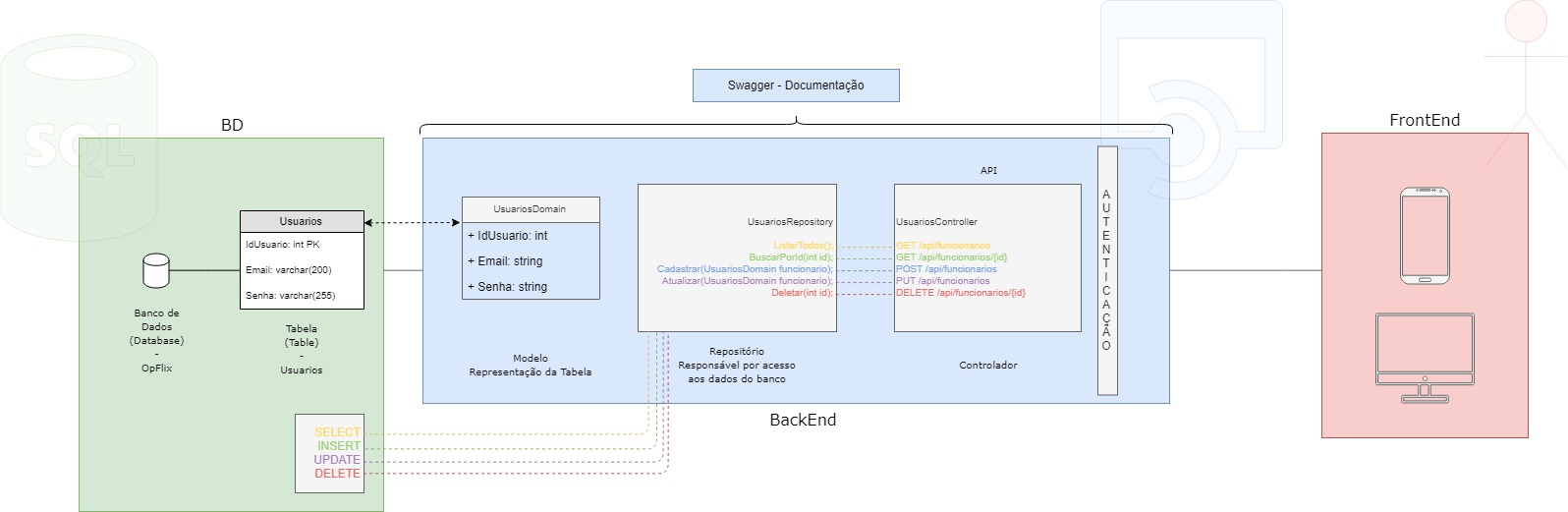
Deploy do projeto:



Pacotes NuGet que foram utilizados no projeto:

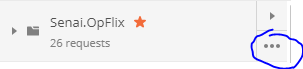
* Swashbuckle.AspNetCore 4.0.1
* System.Data.SqlClient 4.6.1
* Microsoft.EntityFrameworkCore.SqlServer 2.1.11
* Microsoft.EntityFrameworkCore.SqlServer.Design 1.1.6
* Microsoft.EntityFrameworkCore.Tools 2.1.11
* Microsoft.AspNetCore.Authentication.JwtBearer 2.1.1
* System.IdentityModel.Tokens.Jwt 5.5.0

Arquitetura do projeto



Exportação do Postman:

Para exportar um arquivo do postman, clique nos 3 pontos na lateral da coleção a ser exportada:



Após isso, clique em “Export”:



Importação do Postman:

No canto superior esquerdo do programa, clique em “Import”:



Ferramentas utilizadas no projeto:

* Visual Studio 2017
* Microsoft SQL Server Management Studio 18
* Postman

Funcionalidades

Web

Mobile

Protótipos

Web

Mobile

Front-End

Mobile

Arquitetura do Projeto

Referências

Links

Livros