

Documentação

OpFlix

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

Sumário

1. [Resumo 2](#_Toc19633422)

[Objetivos](#_Toc19633423)

1. [Descrição do projeto 2](#_Toc19633424)

[Resumo do projeto](#_Toc19633425)

1. [Modelagem de Software 3](#_Toc19633426)

[Modelo Lógico](#_Toc19633427)

[Modelo Físico](#_Toc19633428)

[Modelo Conceitual](#_Toc19633429)

[Cronograma](#_Toc19633430)

1. [Funcionalidades 5](#_Toc19633431)

[Web](#_Toc19633432)

[Mobile](#_Toc19633433)

1. [Protótipos 6](#_Toc19633434)

[Web](#_Toc19633435)

[Mobile](#_Toc19633436)

1. [Back-End 7](#_Toc19633437)
2. [Swagger 8](#_Toc19633438)
3. [Deploy 9](#_Toc19633439)
4. [Postman/Importar 11](#_Toc19633440)
5. [Postman/Exportar 12](#_Toc19633441)
6. [NuGet 13](#_Toc19633442)
7. [Como criar Banco de Dados e Rodar no BackEnd 14](#_Toc19633443)
8. [Ferramentas Utilizadas 15](#_Toc19633444)
9. [Arquitetura do Projeto 16](#_Toc19633445)
10. [Referências 17](#_Toc19633446)

[Links](#_Toc19633447)

Resumo

Objetivos

Este documento tem como objetivo demonstrar todas as funcionalidades da plataforma OpFlix, que visa demonstrar os eventos que estarão ocorrendo na Escola SENAI de Informática.

Descrição do projeto

O OpFlix tem como objetivo a divulgação de novos eventos dentro da Escola SENAI de Informática.

Resumo do projeto

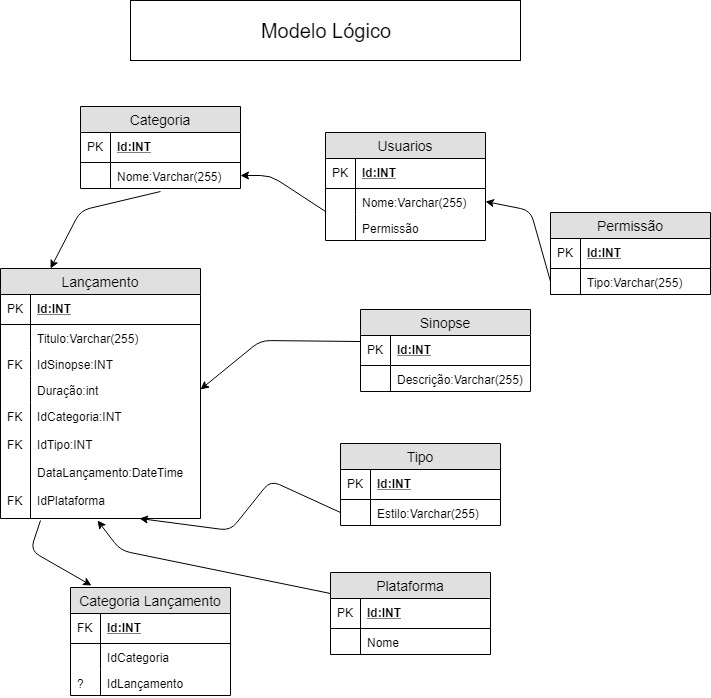
O OpFlix tem a disponibilização de eventos da Escola e suas determinadas categorias. Assim, cada aluno poderá realizar o filtro daqueles eventos que tem interesse, dentro da plataforma. E como item extra, poderá adicionar os que tem interesse, dentro de uma lista.

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 o modelo que mais se aproxima

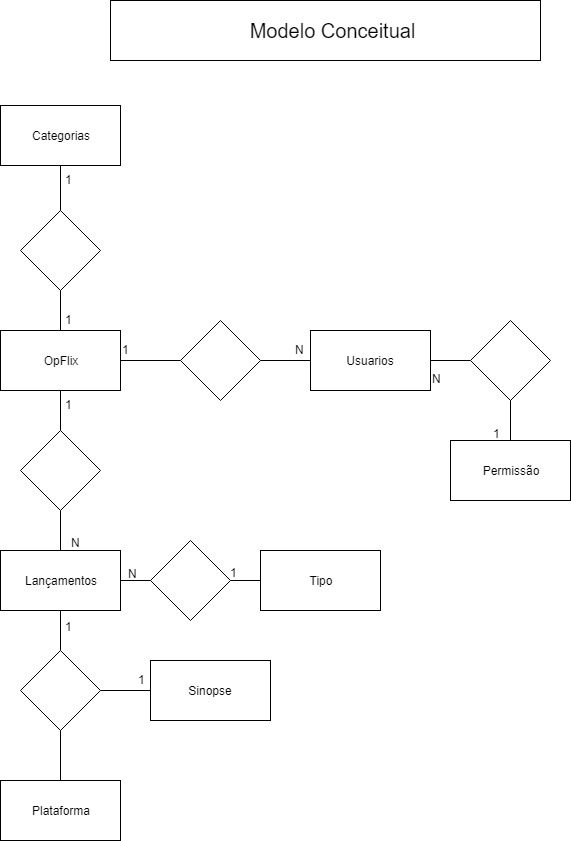


Modelo Físico

Modelo físico é o modelo onde são adicionados valores para validação da modelagem de são

Modelo Conceitual

O modelo conceitual é uma visualização marco das entidades e relacionamentos



Cronograma

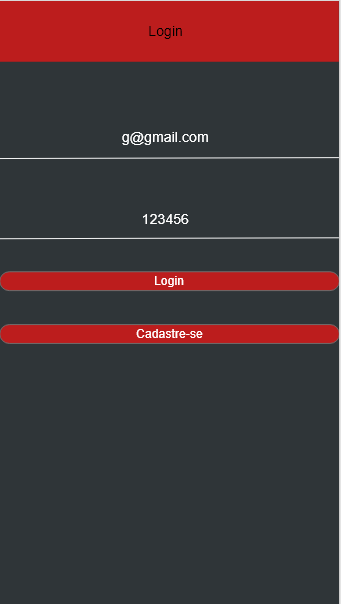
|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 | Dia 5 | Dia 6 | Dia 7 | Dia 8 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Modelo Lógico** |  |  |  |  |  |  |  |  |
| **Modelo Físico** |  |  |  |  |  |  |  |  |
| Modelo Conceitual |  |  |  |  |  |  |  |  |

Funcionalidades

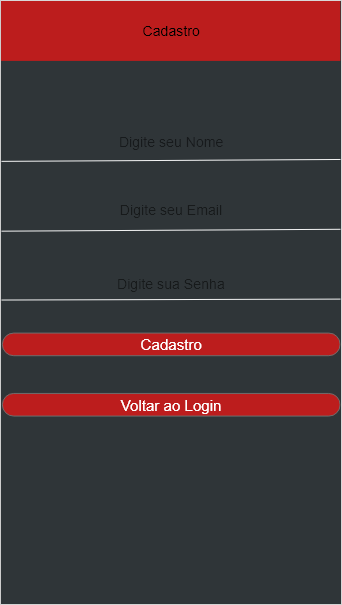
Web

Mobile

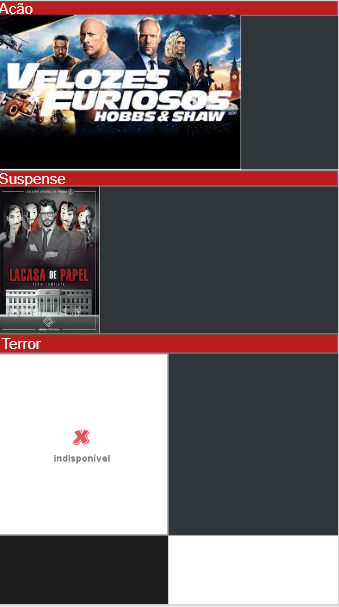
Tela de Login



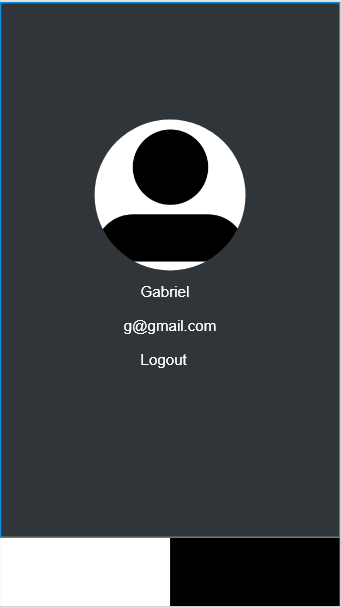
Tela de Cadastro



Tela de Lancamentos



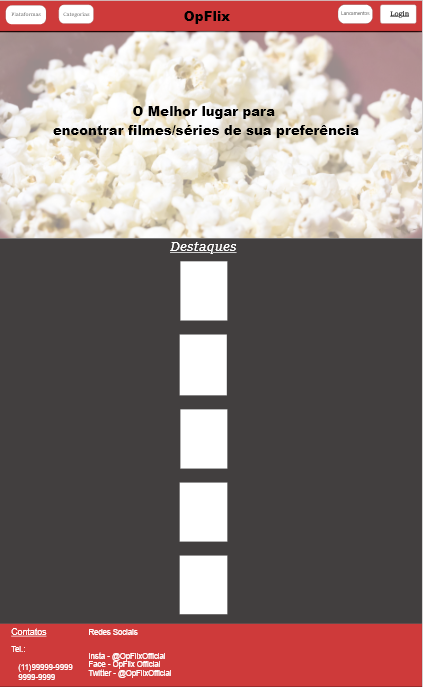
Tela de Perfil



Protótipos

Web

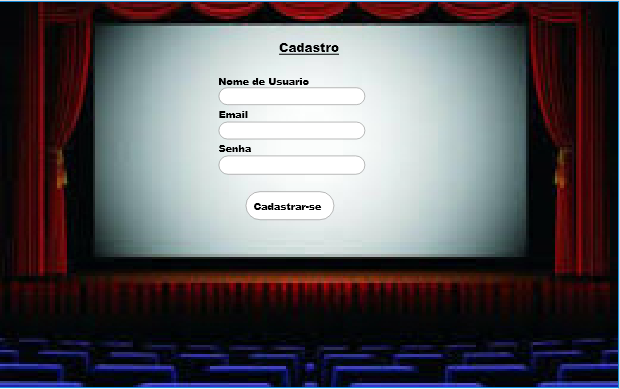
Tela de Destaques/Home



Tela de Login



Tela de Cadastro



Tela de Categorias



Tela de Plataformas

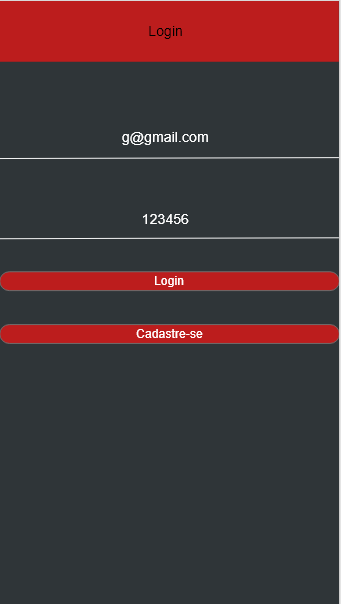


Tela de Lançamentos

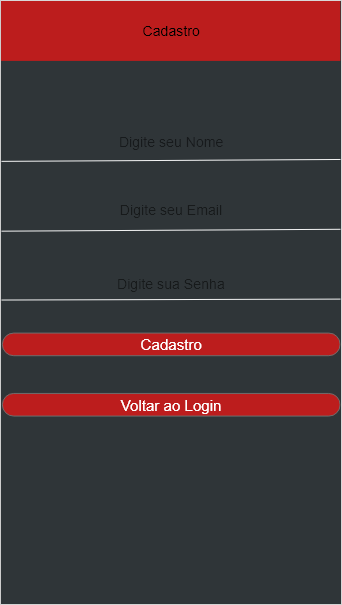


Mobile

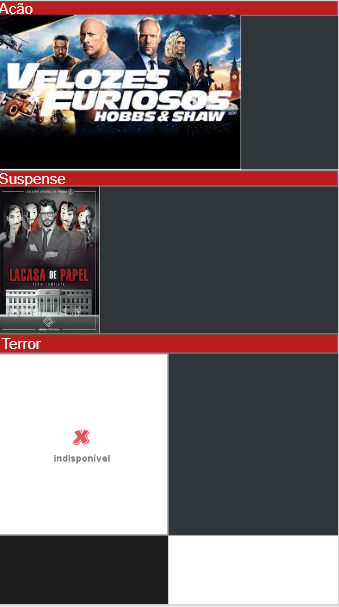
Tela de Login



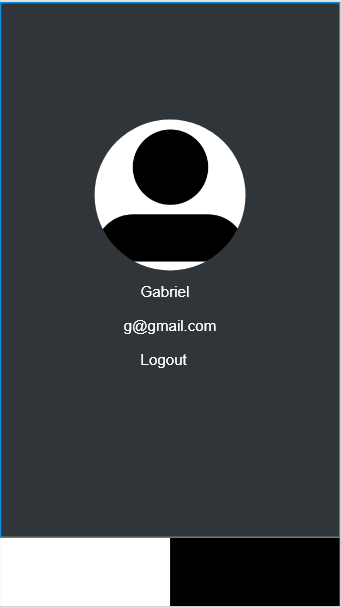
Tela de Cadastro



Tela de Lançamentos



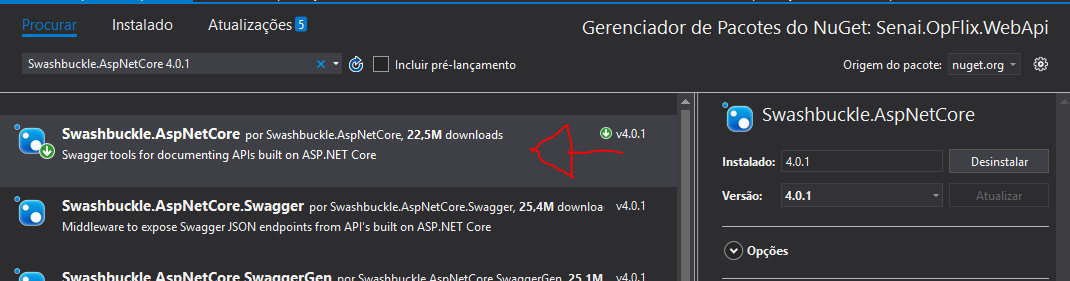
Tela de Perfil



Back-End

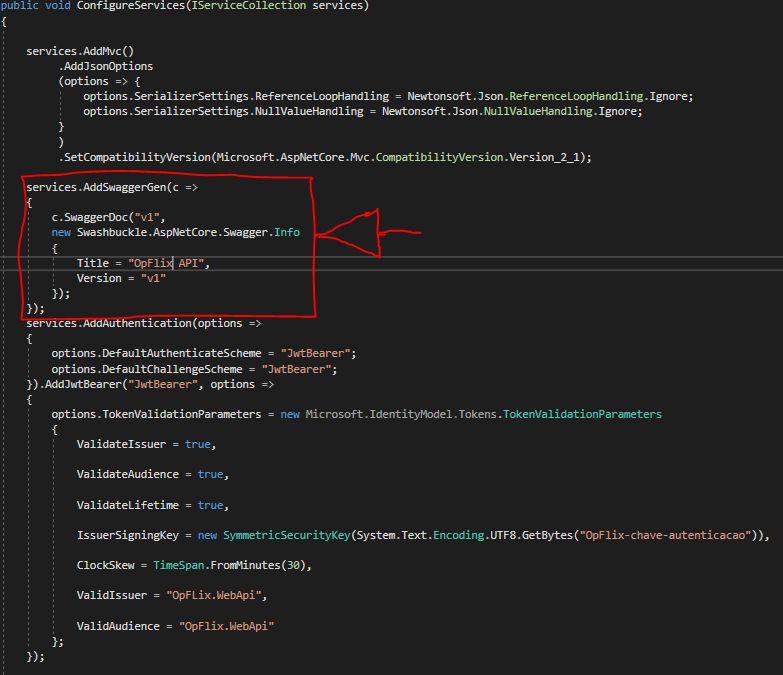
Swagger

Baixar O Pacote do NuGet, pesquisar por Swashbuckle.AspNetCore 4.0.1



Inserir o código para que o Swagger funcione

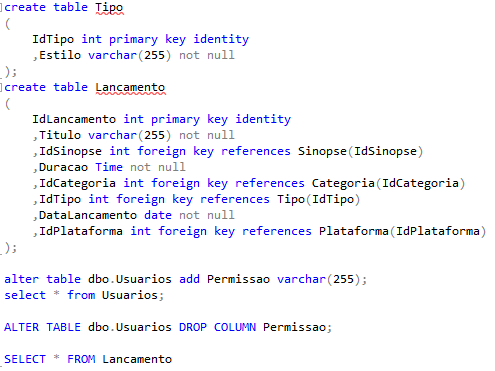
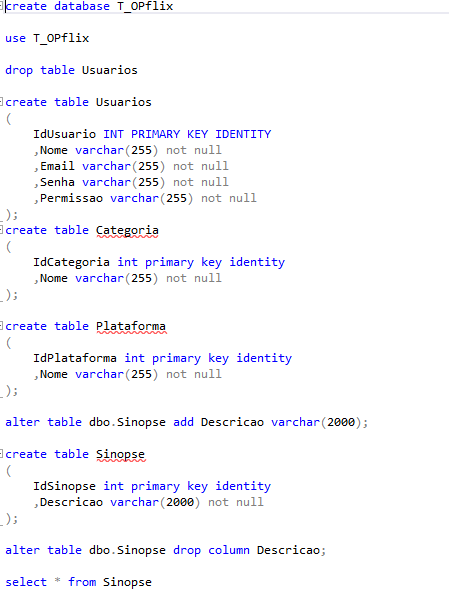
Startup.cs\*



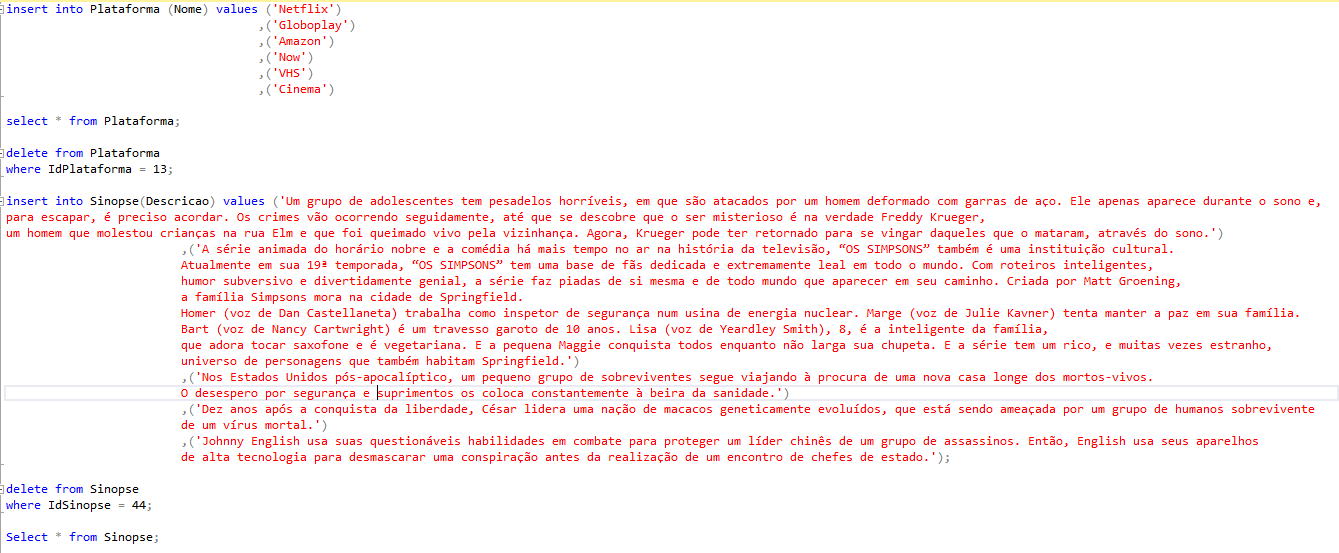
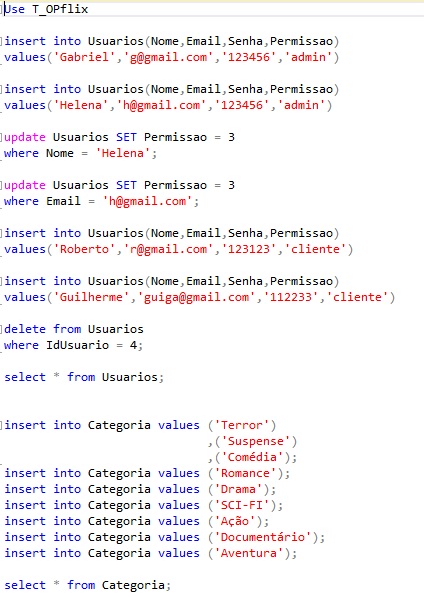
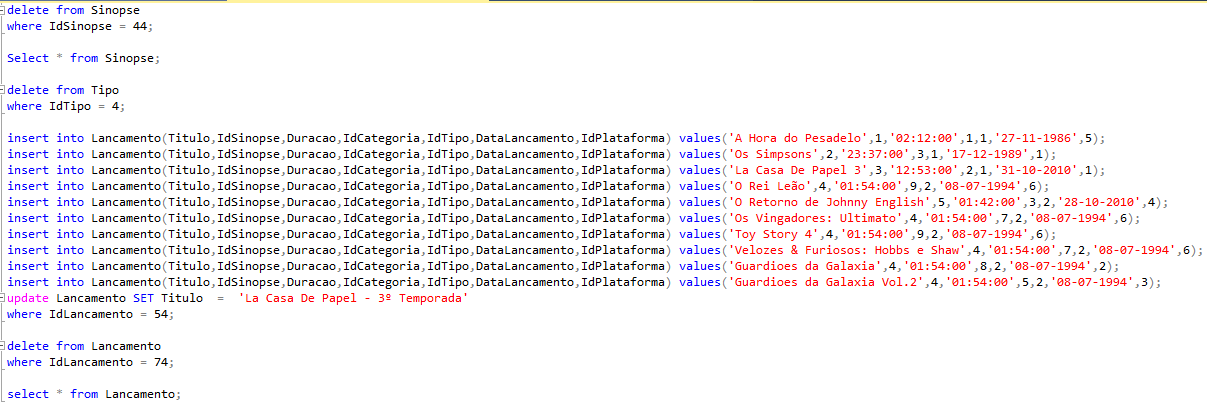


Deploy

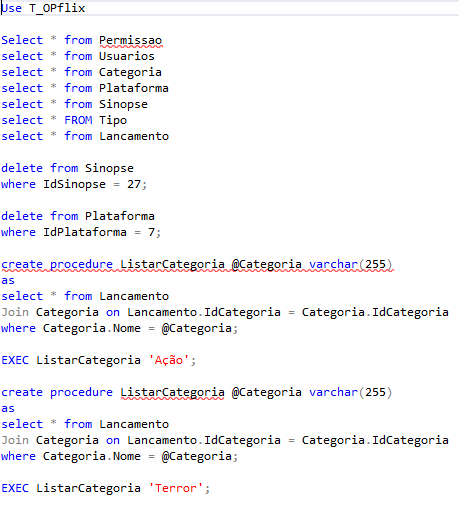
Banco De Dados DDL



Banco de Dados DML

Banco de Dados DQL

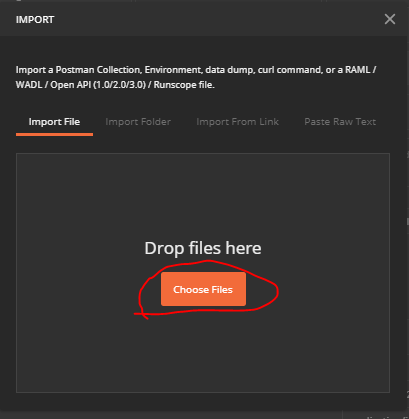


Postman/Importar

**Para importar você deve apertar no botão Import**

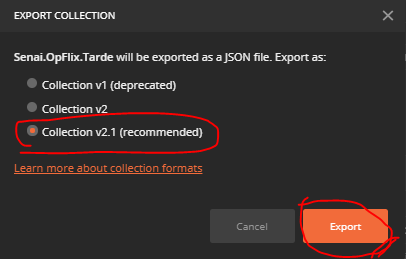
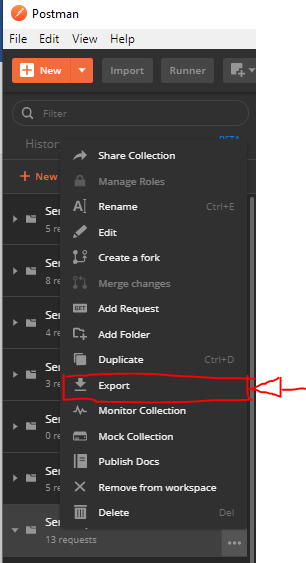
****

**Quando apertar aparecera essa janela e você clicara em “Choose File” para escolher o arquivo desejado, ou então clique em “Import Folder” para importar uma pasta.**

****

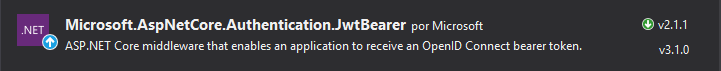
Postman/Exportar

Para exportar você deve apertar nas 3 bolinhas ao lado do arquivo e depois apertar em “Export”. Logo após isso aparecera outra janela na qual você vai deixa na opção “Collection v2.1” e depois clicara em “Export”.

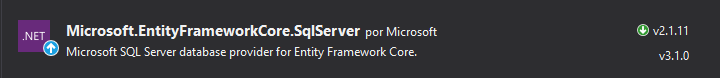


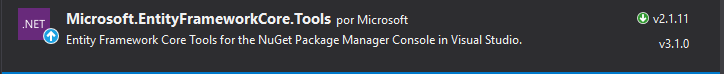
NuGet

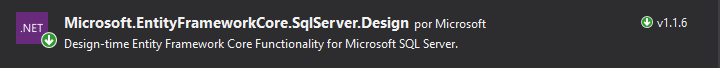
Pesquise por Microsoft.AspNetCore.Authentication.JwtBearer v.2.1.1 e baixe este pacote para realizar a autenticação no Visual Studio.



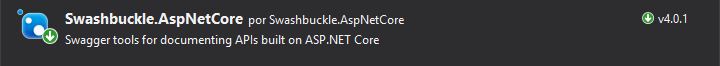
Pesquise por Microsoft.EntityFrameworkCore.SqlServer v.2.1.11, baixe esse pacote para poder implementar as informações do Microsoft Sql Server no Visual Studio.



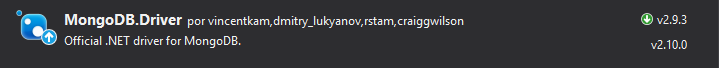




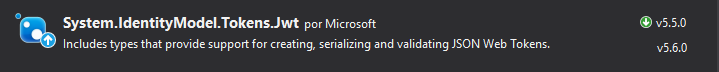
Pesquise por Swashbuckle.AspNetCore v.4.0.1, baixe este pacote para fazer a implementação de informações no Swagger.



Pesquise por MongoDB.Driver v.2.9.3, baixe este pacote para fazer a implementação de informações no MongoDB.



Pesquise por System.IdentityModel.Tokens.Jwt v.5.5.0 e baixe este pacote para quando o usuário fizer a autenticação o sistema buscar o Token da autenticação.



Como criar Banco de Dados e Rodar no BackEnd

Ferramentas Utilizadas

Arquitetura do Projeto

Referências

Links