

Documentação

OpFlix

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

Rafael Pieri

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

Este documento tem como objetivo demonstrar todas as funcionalidades da plataforma OpFlix, que visa cadastrar os lançamentos, e usuários assim como suas propriedades. Para isto, foi definido que eles possuiriam categorias, títulos, sinopse...

Objetivos

Descrição do projeto

O OpFlix tem a principal função de cadastrar usuários e lançamentos.

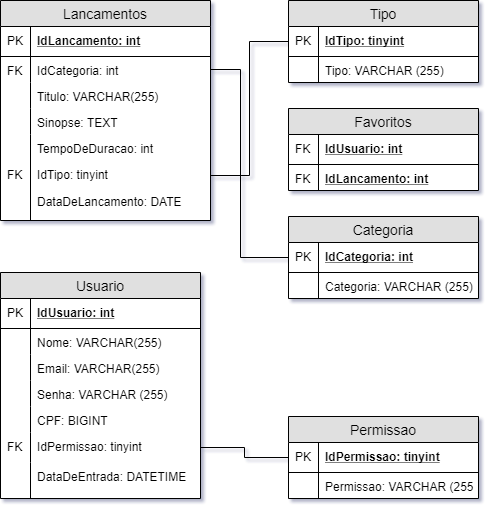
Resumo do projeto

O projeto permite adicionar lançamentos , assim como suas características, sendo categorias, sinopse, tempo de duração, tipo, data de lançamento e veículo. Já nos usuários será possível cadastrar nome, email, senha, CPF, permissão e data de entrada. Assim podendo exibi-los de diversas formas como o cliente desejar.

Modelagem de Software

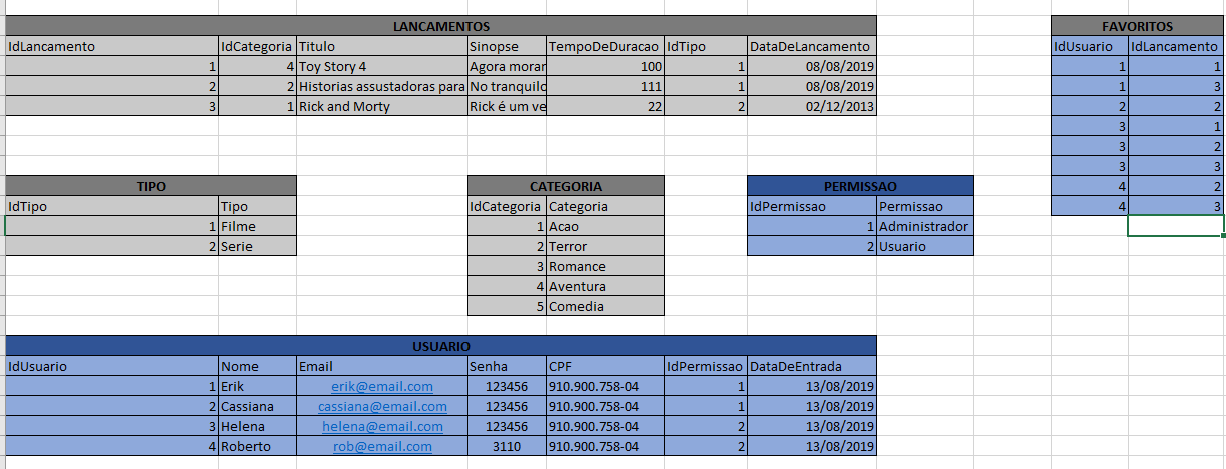
Depois que selecionamos um SGBD, iremos mapear o modelo conceitual para ele. O projetista relaciona as características e restrições do modelo conceitual com as do modelo selecionado para implementação. O modelo lógico constitui uma representação específica de um modelo interno, utilizando as estruturas de BD suportada pelo banco escolhido

Modelo Lógico



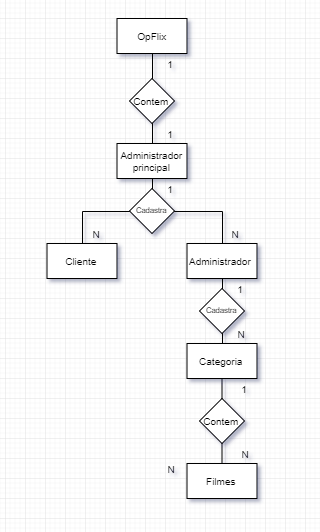
Modelo Físico

Aqui é trabalhado no nível mais baixo de abstração, descrevendo o modo como os dados são salvos em meios de armazenamentos, como discos e fitas, sendo exigido a definição tanto dos dispositivos de armazenamento físico como dos métodos de acesso (físico) necessários para se chegar aos dados nesse dispositivos, o que o torna dependente tanto de software como de hardware.



Modelo Conceitual

O objetivo aqui é criar um modelo conceitual de forma gráfica, sendo este chamado de Diagrama Entidade e Relacionamento (DER), que identificará todas as entidades e relacionamentos de uma forma global. Aqui é evitado qualquer detalhamento específico do modelo de BD.



Cronograma

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

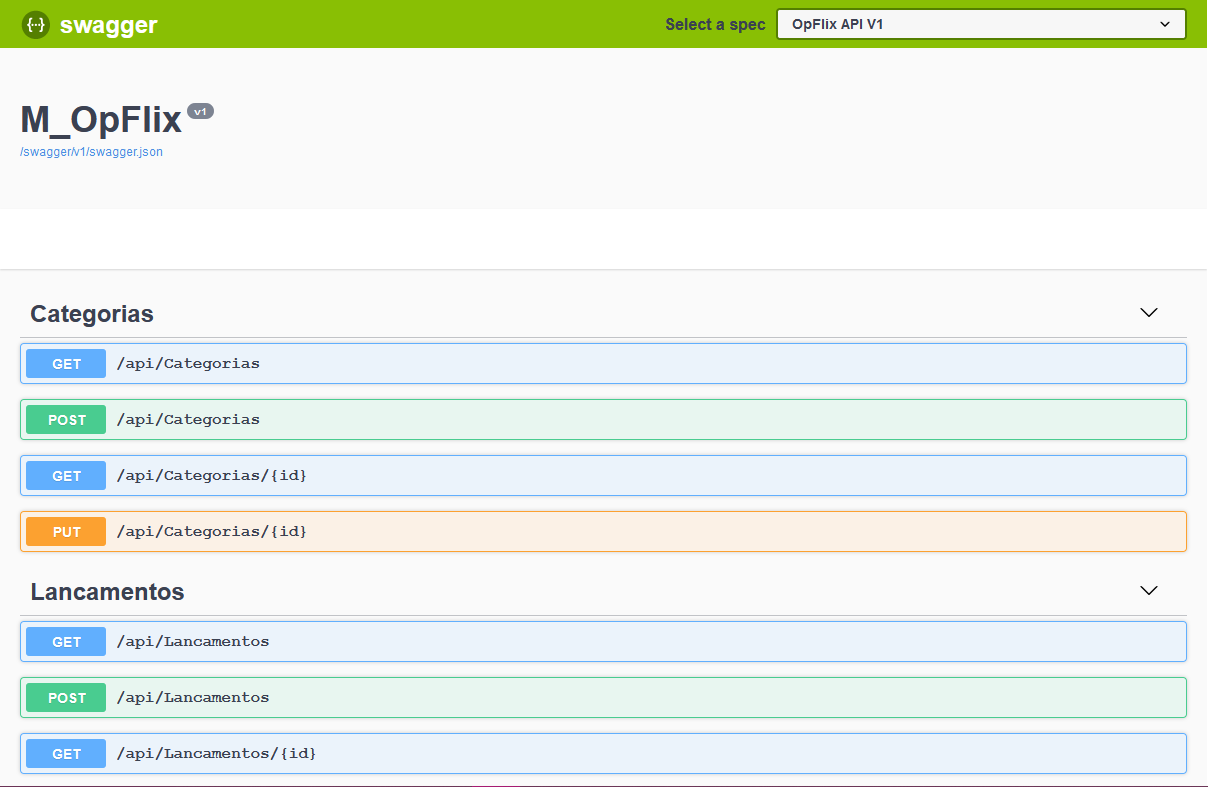
Funcionalidades

Web

Mobile

Swagger

Para acessar o Swagger basta adicionar um “/swagger/index.html” à sua url após a execução do projeto, assim disponibilizando todos os endpoints que o projeto contém.



Deploy

O projeto está disponibilizado na plataforma GitHub.

Importação do Postman

É necessário possuir o Postman em seu computador, então executar ele, quando já estiver aberto existirá um botão no canto superior direito com o nome de “Import”, clique nele e selecione o arquivo do Postman com o nome de “Senai.OpFlix.Manha.postman\_collection”, assim os endpoints já estarão em sua máquina.

Nugets utilizados

Microsoft.EntityFrameworkCore.SqlServer V 2.1.11

Microsoft.EntityFrameworkCore.SqlServer.Design V1.1.6

Microsoft.EntityFrameworkCore.Tools V2.1.11

Microsoft.VisualStudio.Web.CodeGeneration.Design V2.1.9

Swashbuckle.AspNetCore v4.0.1

Swashbuckle.AspNetCore.Swagger V4.0.1

Arquitetura do projeto

WebApi

Executando o projeto

É preciso ter o Microsoft SQL Server Management Studio 18 e o Visual Studio 2017 instalados em sua máquina, então deverá abrir o projeto do SQL e executar as ações, começando com a “CREATE DATABASE M\_OpFlix” assim o banco já será criado, depois executar a “USE M\_OpFlix” para selecionar esse banco, então, as outras ações já poderão ser excutadas de uma vez, seguindo a numeração dos arquivos.



Depois dessa etapa, o último passo é executar o projeto no Visual Studio 2017, e selecionar o botão de executar (que também pode ser feito com o uso da tecla F5).



Ferramentas Utilizadas

Visual Studio 2017

Microsoft SQL Server Management Studio 18

Swagger

JwtBearer

Postman

GitHub

Protótipos

Web

Mobile

Front-End

Mobile

Arquitetura do Projeto

Referências

Links

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

Livros