

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. [Back-End 5](#_Toc533767852)

Swagger

Deploy

Pacotes Nuget

Ferramentas Usadas

Arquitetura do Projeto

Criando o Banco de Dados

Importando o PostMan

1. [Front-End 7](#_Toc533767858)
2. [Mobile 8](#_Toc533767859)
3. [Referências 10](#_Toc533767861)

[Links](#_Toc533767862)

[Livros](#_Toc533767863)

Resumo

Objetivos

Descrição do projeto

Criar um site e um aplicativo para uma empresa chamada OpFlix que gerencia lançamentos de filmes e séries em diversas plataformas.

Resumo 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. 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.

**Acessando o Banco de Dados**

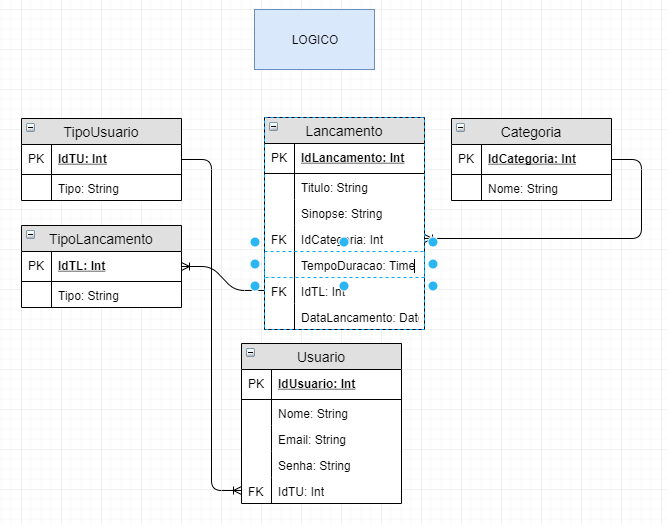
Acesse o link do Github abaixo, entre na pasta BD, e lá estará os 3 arquivos do Banco de Dados (DDL, DML, DQL)

<https://github.com/S0L4/2s2019-sprint-1-bd-OpFlix>

Modelagem de Software

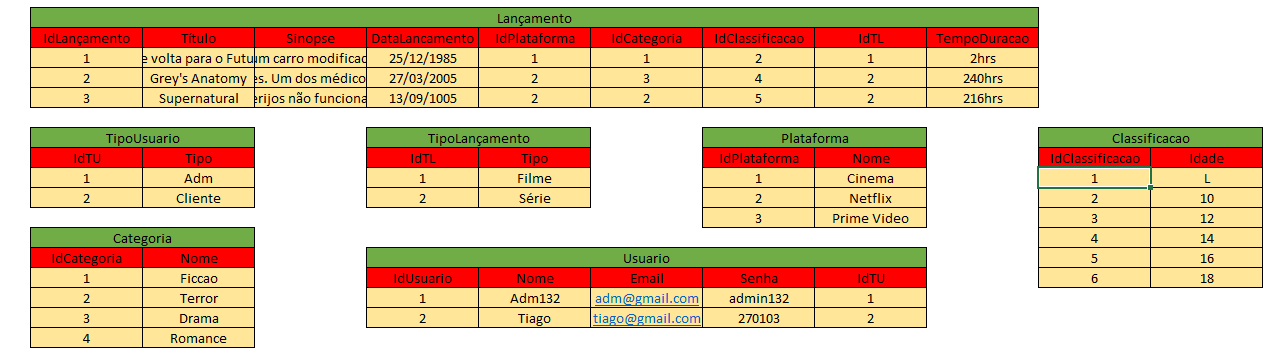
A **Modelagem de Dados** é a criação de um modelo físico que explique a lógica por traz do sistema, com ele você é capaz de explicar as características de funcionamento e comportamento de um software. A **modelagem de dados**é a base de criação do [Banco de dados](http://aprendaplsql.com/oracle/o-que-e-um-banco-de-dados/) e parte essencial para a qualidade do software.

Modelo Lógico

O modelo lógico irá mostrar os tipos de dados das informações que você adicionara no banco de dados.

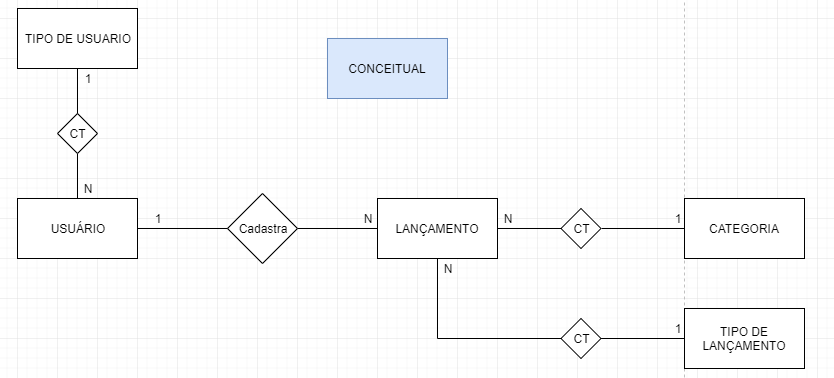
Modelo Físico

O modelo físico é o que mais se aproxima do banco de dados em si, replicando as tabelas e suas informações.



Modelo Conceitual

O **objetivo do modelo conceitual** é **criar um sistema coerente de objetos, propriedades e relações**claramente mapeados para o domínio da tarefa do usuário.



Cronograma

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

Back-End

**Swagger**

O Swagger é uma forma de trazer uma melhor organização para que o Cliente fique ciente do que o programador está fazendo e os passos que são tomados.

[**http://localhost:5000/swagger/index.html**](http://localhost:5000/swagger/index.html)

**Deploy**

Após a elaboração do projeto, criação, avaliação e testes terem sido realizados, temos a questão da publicação e liberação dos projetos para o cliente, onde precisamos ter maneiras simples de implantação do lado do usuário para que o mesmo possa fazer uso da aplicação, seja ela web, mobile ou desktop.



**Pacotes Nugget**

**Swagger** - Swashbuckle.AspNetCore 4.0.1

**SqlClient** - System.Data.SqlClient 4.6.1

**EFCore** - Microsoft.EntityFrameworkCore.SqlServer 2.1.11

Microsoft.EntityFrameworkCore.SqlServer.Design 1.1.6

Microsoft.EntityFrameworkCore.Tools 2.1.11

**JWT** - Microsoft.AspNetCore.Authentication.JwtBearer 2.1.1

System.IdentityModel.Tokens.Jwt 5.5.0

**Ferramentas Usadas**

**1** - Visual Studio 2017

**2** - Postman

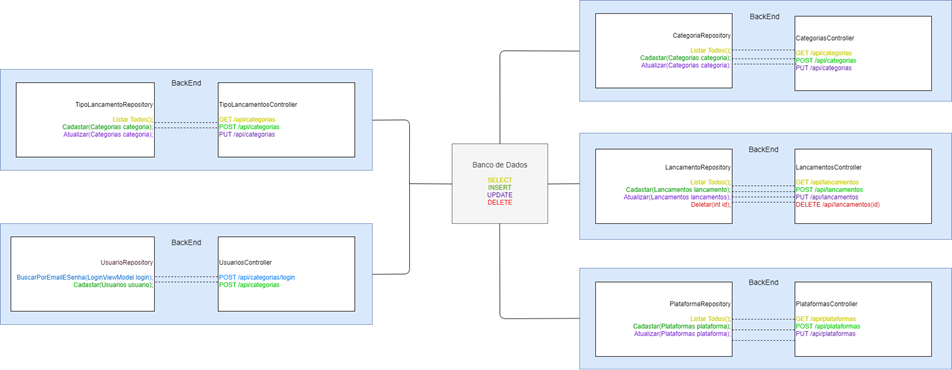
**3** - Swagger

**4** - SqlServer

**5** - Trello

**6** - GitHub

**Arquitetura do Projeto**



**Criando o Banco de Dados**

Acesse o link do Github abaixo, baixe a pasta BD, e lá estará os 3 arquivos do Banco de Dados (DDL, DML, DQL). Depois de completar o download, acesse o arquivo: [T\_01\_TiagoSola\_DDL.sql](https://github.com/S0L4/2s2019-sprint-1-bd-OpFlix/blob/master/BD/T_01_TiagoSola_DDL.sql) e execute os comando e assim sucessivamente..

<https://github.com/S0L4/2s2019-sprint-1-bd-OpFlix>

**Importando o PostMan**

O Postman é um aplicativo com a função de testar e desenvolver APIs em uma interface bastante simples e intuitiva. Ele nos permite simular requisições HTTP de forma rápida, armazenando-as para que possamos usá-las posteriormente.



Front-End

Mobile

Arquitetura do Projeto

Referências

Links

Livros