

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

O presente documento tem a finalidade de apresentar todas as funcionalidades da plataforma OpFlix. Esta que por sua vez visa apresentar os próximos lançamentos de series, filmes e outras mídias de entretenimento.

Descrição do projeto

O OpFlix tem como principal funcionalidade e objetivo a divulgação dos mais diversos conteúdos de entretenimento, a plataforma apresenta diversos dados dos lançamentos desses conteúdos, que incluem nome do lançamento, categorias vinculadas ao lançamento, data de estreia, plataformas, mídias, sinopse e duração.

Resumo do projeto

O OpFlix é uma plataforma capaz de listar lançamentos de filmes e séries com seus respectivos dados. Os usuários da plataforma são capazes de visualizar estes lançamentos previamente cadastrados.

O projeto possui alguns perfis de usuários, e, todos os clientes se encaixarão em um dos disponíveis. Sendo eles:

### **Perfis de usuário:**

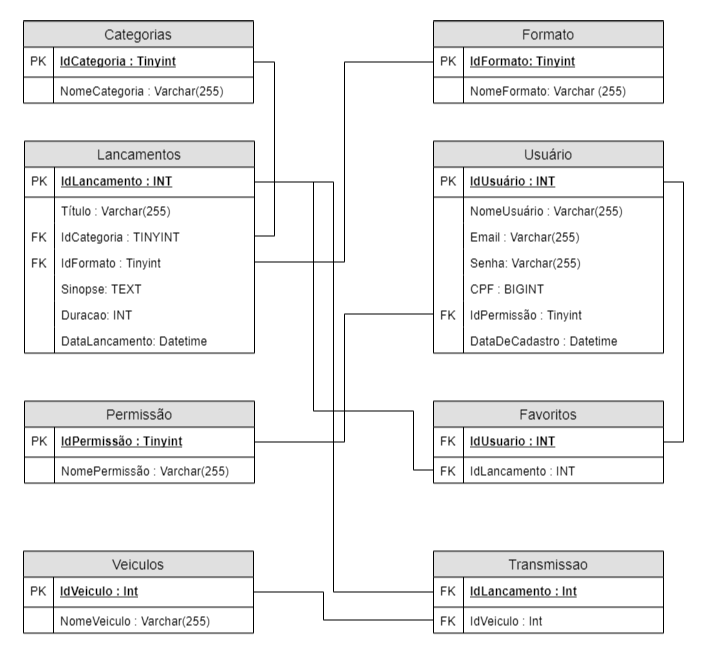
* **Administrador**: Para o colaborador da gestão administrativa;
* **Cliente**: Clientes da empresa;

Modelagem de Software

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

Modelo Lógico

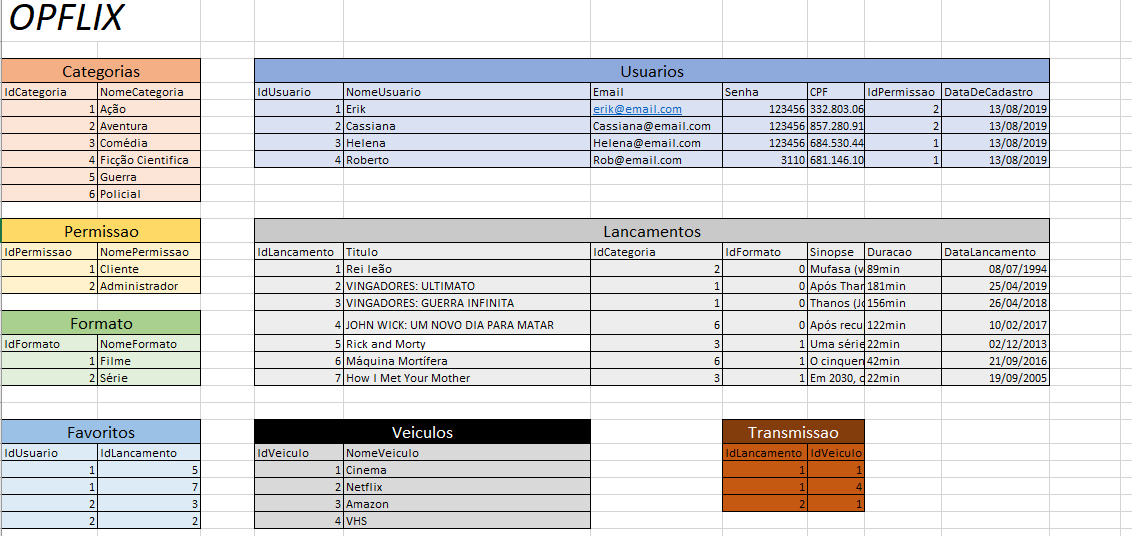
O modelo lógico é o responsável por levar em conta algumas limitações e implementar recursos como adequação de padrão e nomenclatura, define as [chaves primárias e estrangeiras](https://www.luis.blog.br/chave%20primaria-chave-estrangeira-e-candidata.aspx), normalização, integridade referencial, entre outras.



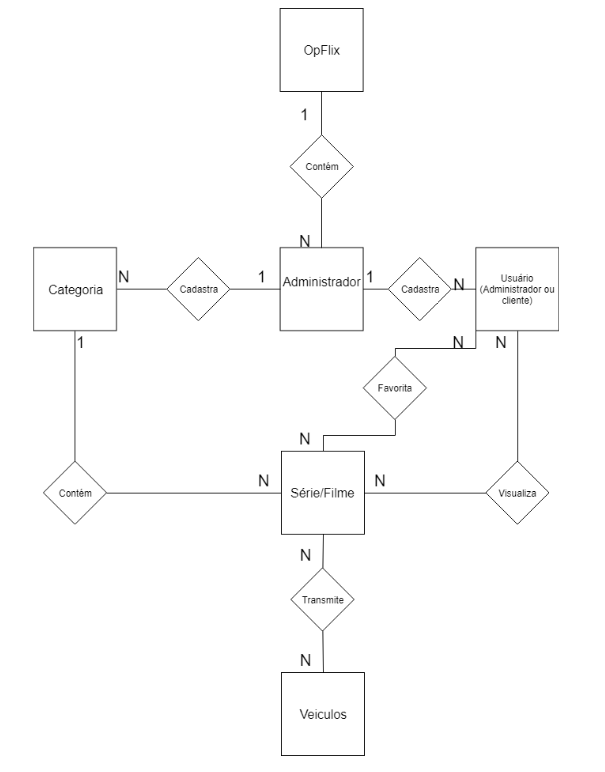
Modelo Físico

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

No modelo físico fazemos a modelagem física do modelo de banco de dados. Neste caso leva-se em conta as limitações impostas pelo SGBD escolhido e deve ser criado sempre com base nos exemplos de modelagem de dados produzidos no item anterior, modelo lógico.



Modelo Conceitual

O modelo conceitual é uma visualização dos “caminhos” que os dados percorrem, assim como as suas cardinalidades. A modelagem conceitual baseia-se no mais alto nível e deve ser usada para envolver o cliente, pois o foco aqui é discutir os aspectos do negócio do cliente e não da tecnologia. 

Cronograma

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

Execução do Script

1º Instalar o Microsoft SQL Server Management Studio e o Visual Studio 2017

2º Abrir o Documento M\_01\_MuriloMilani\_DDL.sql e executar (F5).

3º Abrir o Documento M\_02\_MuriloMilani\_DML.sql e executar (F5).

4ºAbrir o Documento M\_03\_MuriloMilani\_DQL.sql e executar (F5).

5°Abrir o Arquivo Senai.OpFlix.WebApi.sln e executar (F5).

Back-End

A plataforma em **API** do **OpFlix** foi produzida em **C#**. Durante a elaboração deste Back-End Foram utilizadas algumas ferramentas que auxiliam tanto o desenvolvedor como o usuário.

Como por exemplo o **JSON** que funciona como uma formatação leve de troca de dados.

O **JWT** que nada mais é que um padrão ([RFC-7519](https://tools.ietf.org/html/rfc7519)) de mercado que define como transmitir e armazenar objetos **JSON** de forma compacta e segura entre diferentes aplicações. A Utilização do **JWT** também fornece uma maior segurança da aplicação, trabalhando junto com **Interfaces** e sistemas de **Hashing** a segurança se intensifica ainda mais pois o acesso ao banco de dados é restringido.

O **Entity Framework (EF)** que é um O/RM (Mapeador Relacional de Objeto) que permite que os desenvolvedores de .NET trabalhem com um banco de dados usando objetos .NET,

O **Postman** que basicamente é uma plataforma colaboração para o desenvolvimento de **API**. Os seus recursos simplificam cada etapa da criação de uma **API** e agilizam a colaboração para que você possa criar **APIs** melhores e mais rapidamente. Para importar uma coleção Postman em sua máquina basta ir até a opção “File” (ou “Arquivo” na versão traduzida) e selecionar “Import”( ou “Importar” na versão traduzida) e selecionar o diretório onde o arquivo está presente. Para exportar uma coleção do Postman você precisa clicar com o botão alternativo em cima da sua Coleção e selecionar a opção “Export” (ou “Exportar ” na versão traduzida), selecionar a versão que você deseja para a sua coleção e selecionar onde o seu arquivo será salvo.

E por fim o **Swagger** que assim como o **Postman** é uma plataforma para auxiliar em seu projeto, ele simplifica o desenvolvimento da **API** para usuários, equipes e empresas. Sua utilização é bem simples, pois basta incluir ao final da URL do projeto funcional o “/Swagger” para visualizar End-Points que servem como testes em tempo real e de fácil acesso.

Graças ao uso da API em seu desenvolvimento, o OpFlix pode ser utilizado em múltiplas plataformas seja ela Mobile, ou Desktop.

Além disso, durante o desenvolvimento foram utilizados NuGet’s que são basicamente um gerenciador de pacotes do .NET. Esses pacotes são extremamente importantes para a execução correta da aplicação. Os NuGet’s utilizados nesta Aplicação são:

* 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

Funcionalidades

Web

Mobile

Protótipos

Web

Mobile

Front-End

Mobile

Arquitetura do Projeto

Referências

Links

Livros