

Documentação

OpFlix

Alameda Barão de limeira,

539, São Paulo, SP

Sumário

**Resumo3**

Objetivos

**Descrição do projeto3**

Resumo do Projeto

Sistema Web

Sistema Mobile

**Modelagem de software5**

Modelo Lógico

Modelo Físico

Modelo Conceitual

**Funcionalidades8**

API

Como acessar o swagger

Como fazer deploy

Pacotes NuGet

Arquitetura do projeto

Criação de todo do banco e rodar o projeto de backend

Quais ferramentas foram utilizadas

Web

Mobile

**Protótipos15**

Web

Mobile

**Front-End16**

**Mobile17**

**Arquitetura do Projeto**1**8**

**Referências**1**9**

Links

Livros

Resumo

**Objetivos**

**Este documento tem como objetivo demonstrar todas as funcionalidades da plataforma OPFlix, que visa oferecer um guia e a divulgação de novos lançamentos de filmes e séries por ano.**

Descrição do projeto

O OPFlix tem como objetivo a disponibilização de

**Resumo do projeto**

O cliente Tadeu quer um sistema web/mobile integrado onde seja possível realizar a gestão de todos os lançamentos de maneira integrada e otimizada.

Tadeu levantou os seguintes requisitos:

Sistema Web

**Perfis de usuário:**

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

**Funcionalidades**

1. O administrador poderá cadastrar qualquer tipo de **usuário (administrador ou cliente)**;
2. O administrador poderá cadastrar categorias (contendo nome);
3. O administrador poderá cadastrar o lançamento de um filme/série (contendo título, uma pequena sinopse, a categoria vinculada -poderá ter somente uma vinculada, tempo de duração, e se é filme ou série e a data do primeiro lançamento daquele item);
4. O cliente poderá visualizar todos os lançamentos publicados;

Sistema mobile

**Perfis de usuário:**

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

**Funcionalidades**

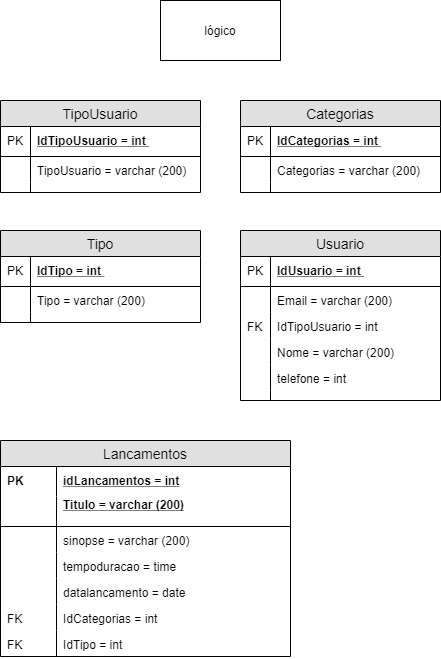
1. O cliente poderá visualizar todos os lançamentos;
2. O cliente poderá visualizar os lançamentos pesquisando por ID

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 já leva em conta algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define as chaves primárias e estrangeiras, normalização, integridade referencial, entre outras. Para o modelo lógico deve ser criado levando em conta os exemplos de modelagem de dados criados no modelo conceitual.

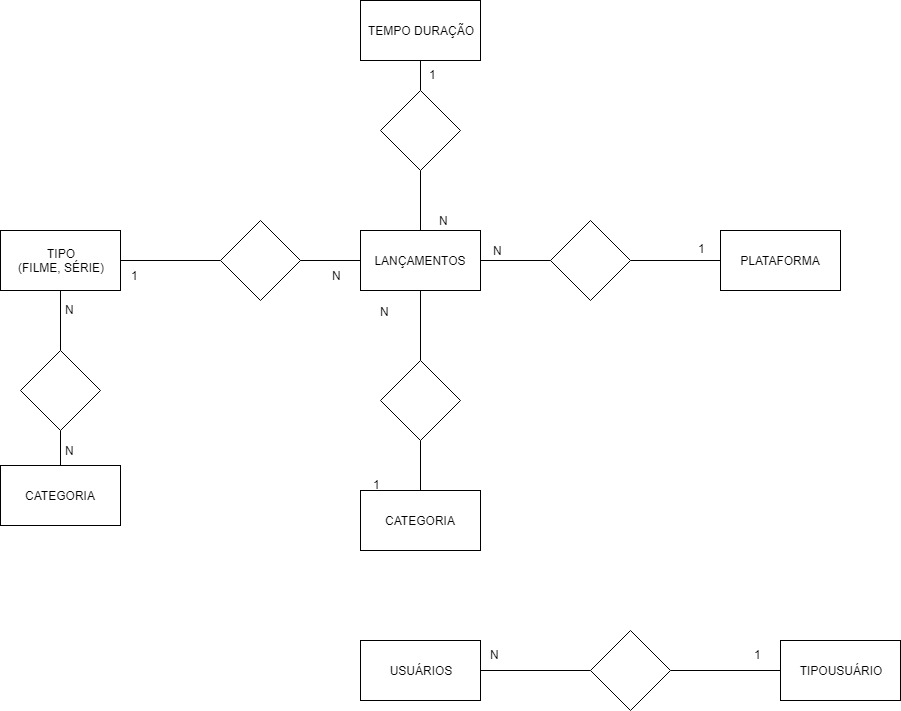


**Modelo Físico**

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**

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. Os exemplos de modelagem de dados vistos pelo modelo conceitual são mais fáceis de compreender, já que não há limitações ou aplicação de tecnologia específica. O diagrama de dados que deve ser construído aqui é o Diagrama de Entidade e Relacionamento, onde deverão ser identificados todas as entidades e os relacionamentos entre elas. Este diagrama é a chave para a compreensão do modelo conceitual de dados.



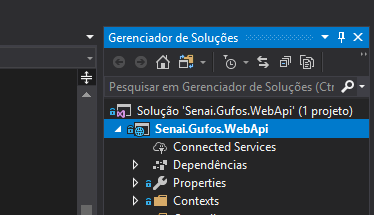
Funcionalidades

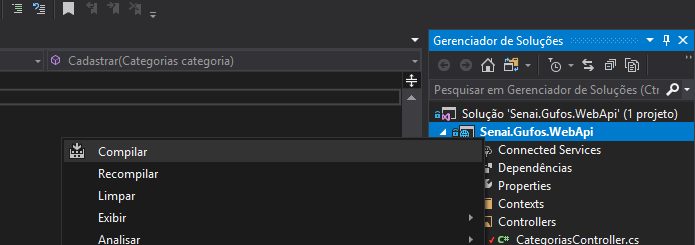
**Api**

API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web. A sigla API refere-se ao termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos".

**Como fazer deploy?**

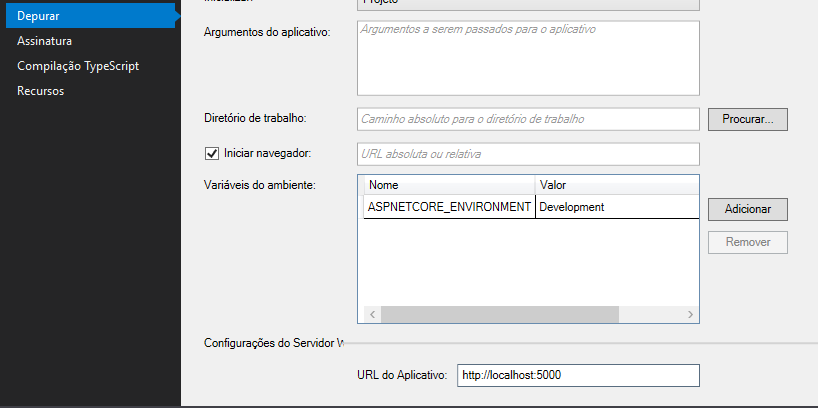
Depois de acessar o projeto, clicar na solução do projeto e depois clicar com o botão direito do mouse onde está escrito em negrito o nome do projeto, e então clicar em “Compilar”, o projeto começará a rodar.



****

**Como acessar o Swagger?**

Depois de fazer deploy, clicar em Properties>Depurar, e checar qual será a porta usada em “URL do Aplicativo”.

****

Depois de checar, colar o endpoint no seu navegador e colocar “/Swagger”.

**Pacotes NuGet usados**

Microsoft.AspNetCore.App (já vem quando cria projeto em ASP.NET CORE 2.1)

Microsoft.AspNetCore.Razor.Design (já vem quando cria projeto em ASP.NET CORE 2.1)

Microsoft.NETCore.App (já vem quando cria projeto em ASP.NET CORE 2.1)

Swashbuckle.AspNetCore 4.0.1 (Swagger)

System.Data.SqlClient 4.6.1 (SqlClient)

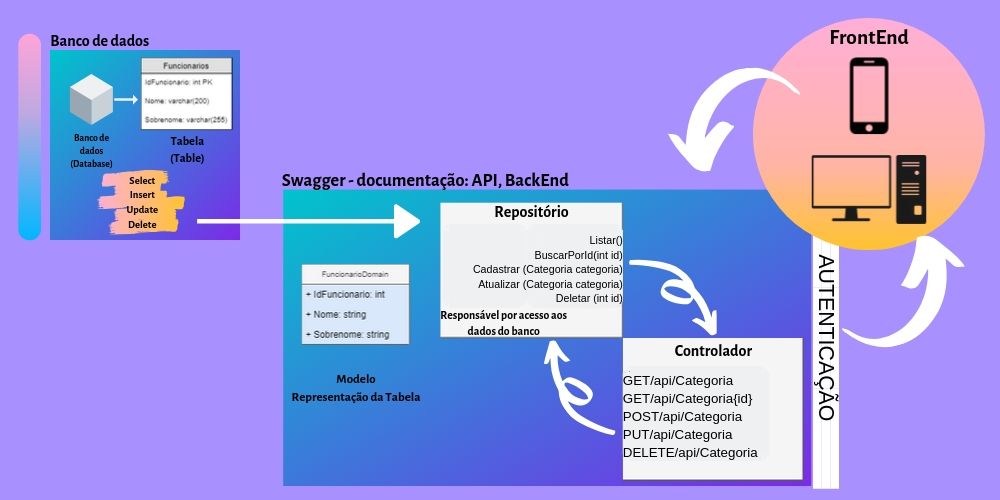
Microsoft.EntityFrameworkCore.SqlServer 2.1.11 (EntityFrameworkCore)

Microsoft.EntityFrameworkCore.SqlServer.Design 1.1.6 (EntityFrameworkCore)

Microsoft.EntityFrameworkCore.Tools 2.1.11 (EntityFrameworkCore)

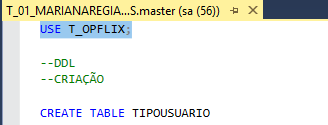
Microsoft.AspNetCore.Authentication.JwtBearer 2.1.1 (JWT)

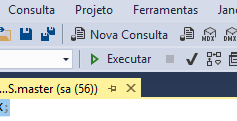
System.IdentityModel.Tokens.Jwt 5.5.0 (JWT)

**Arquitetura** **do** **projeto**

**Criação do banco e rodar o projeto de backend**

Depois de clicar em algum dos três arquivos de Banco de Dados (Sql), é só selecionar “Use nome\_projeto” e clicar em “Executar”.

****

****

Depois de colocar o banco para rodar, faça deploy.

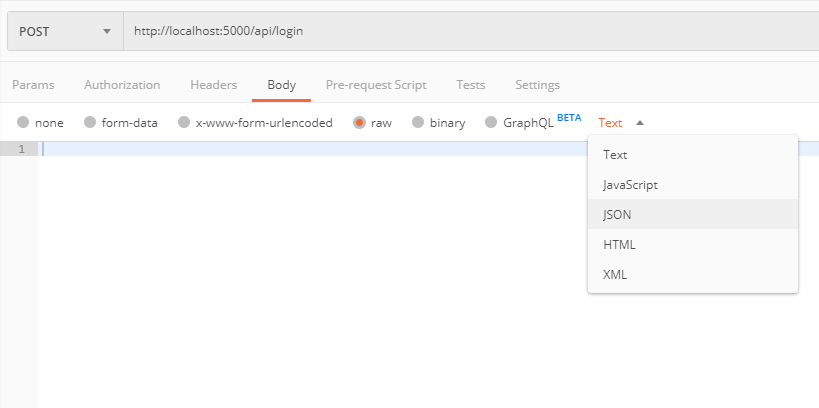
Para simular um usuário, a ferramenta é o Postman.

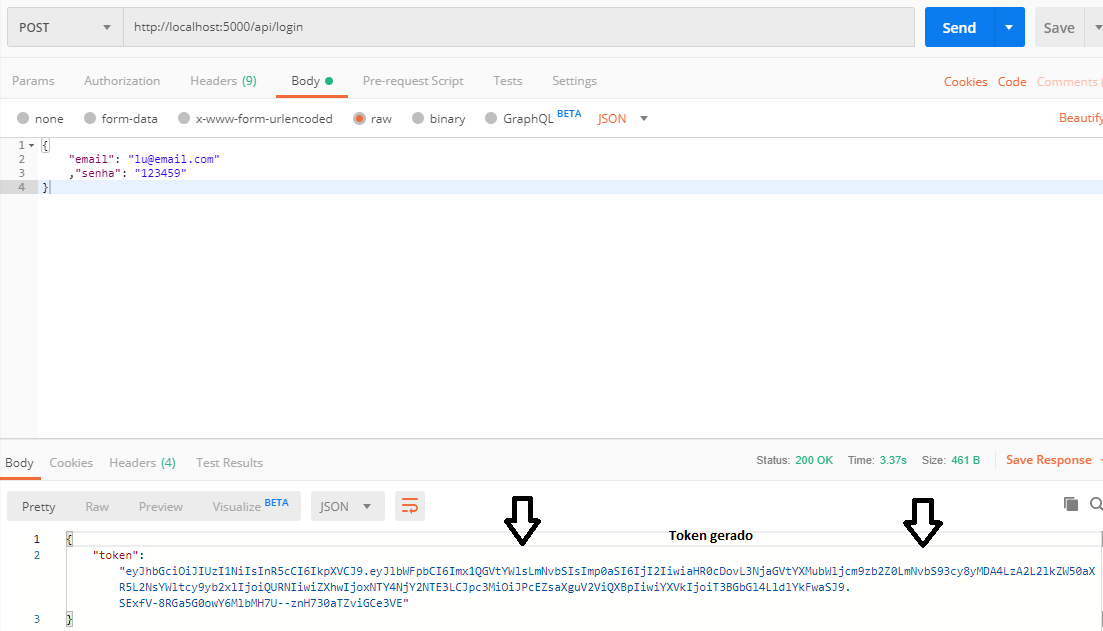
Abra o postman, cole o endpoint na aba de pesquisa dependendo da ação (GET, POST, PUT, DELETE), será necessário mudar no ícone ao lado da aba.

**Caso haja** autorização especial (administrador e comum), será necessário gerar token: colocar endpoint correspondente ao login, mudar ícone para POST,

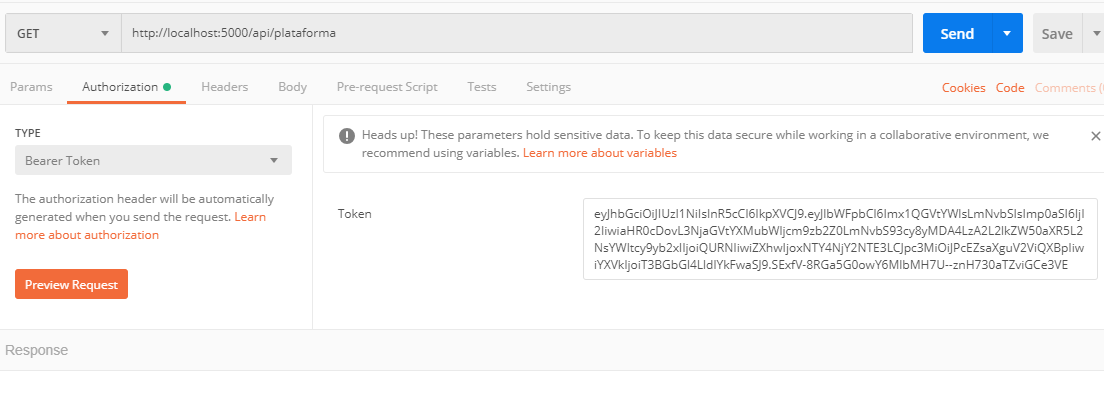
Então

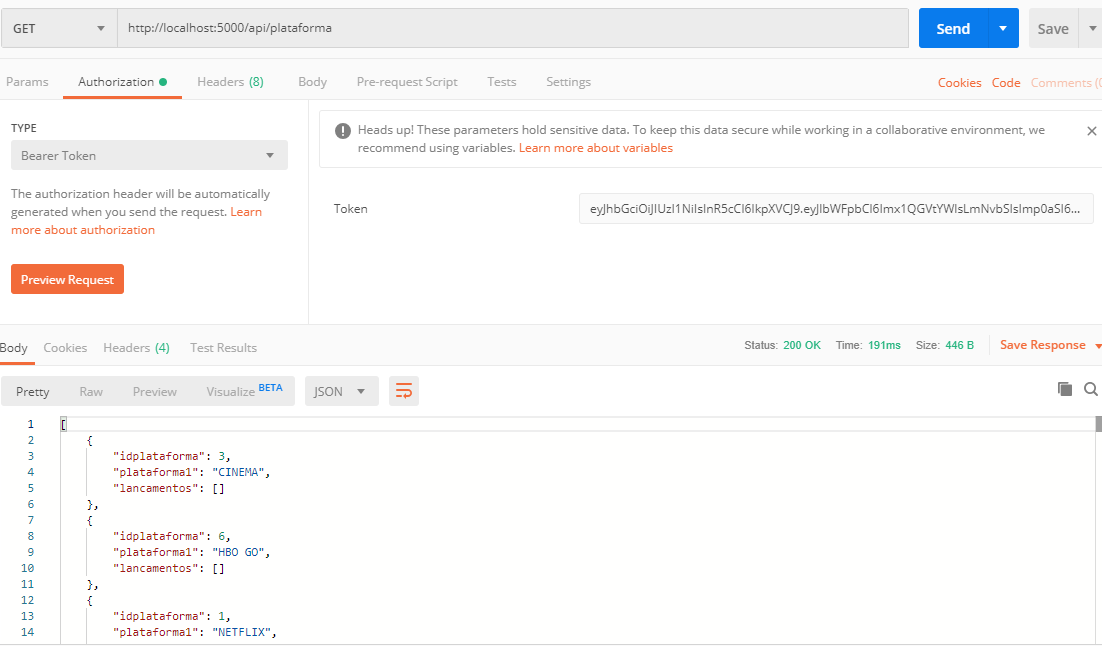
Body>raw> JSON



 e inserir email e senha em modelo JSON, ele gerará um token;

depois, inserir o endpoint da ação desejada, mudar o ícone de acordo com a ação e

Authorization>Type: Bearer Token>Colar token gerado ali>Send



**Quais ferramentas foram utilizadas**

Visual Studio 2017 versão 15.9.14

Swagger.Io

Postman versão 7.7.2

Microsoft SQL Server Management Studio versão 15.0.18131.0

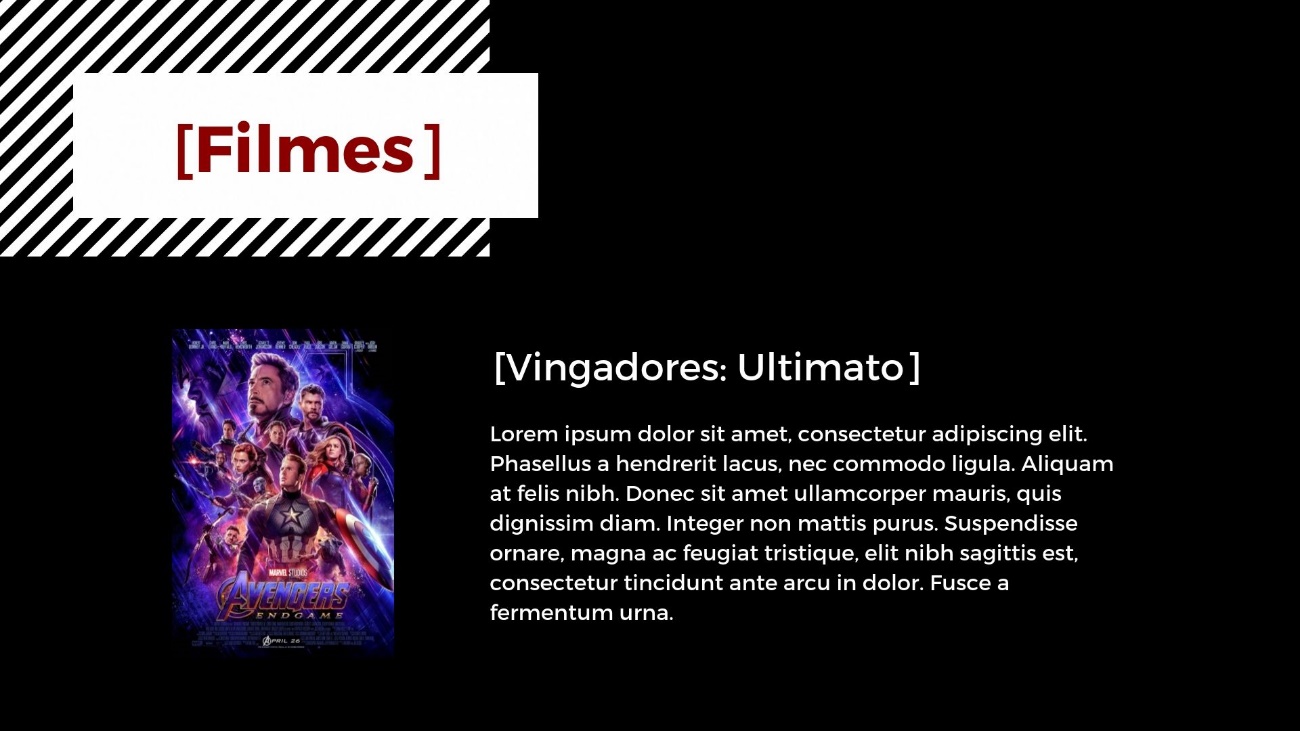
**Web**

**Mobile**

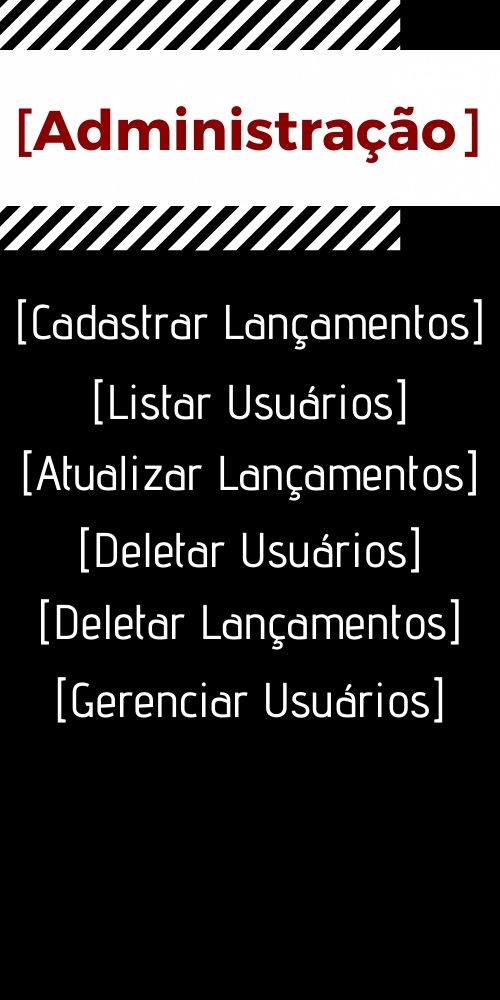
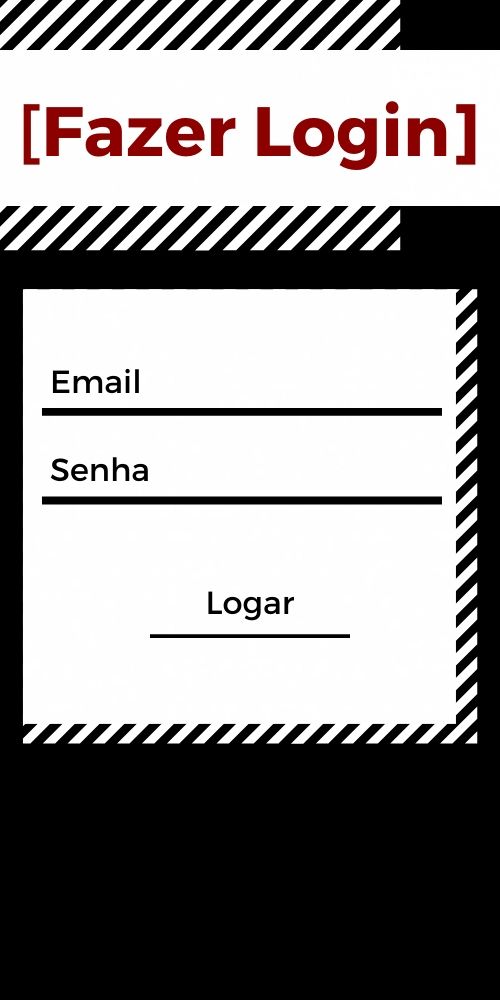
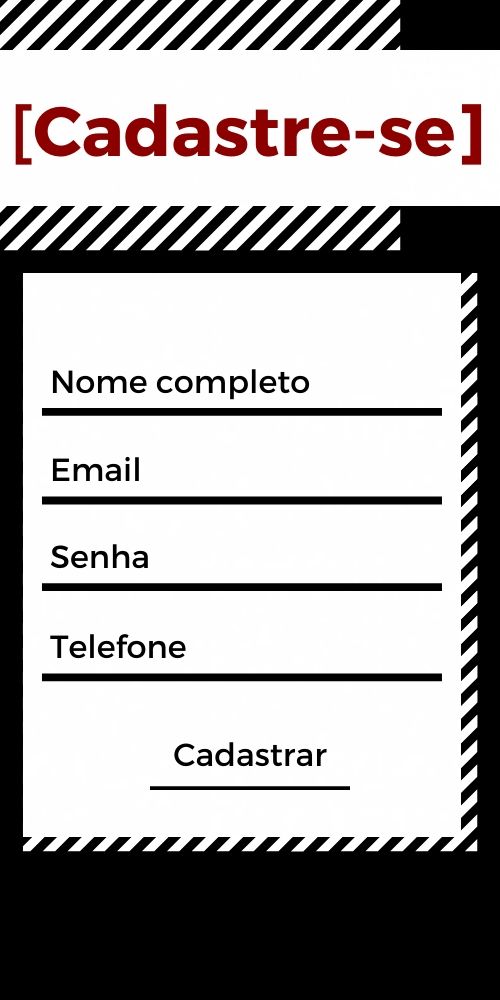
Protótipos

**Web**

****

****

**Mobile**

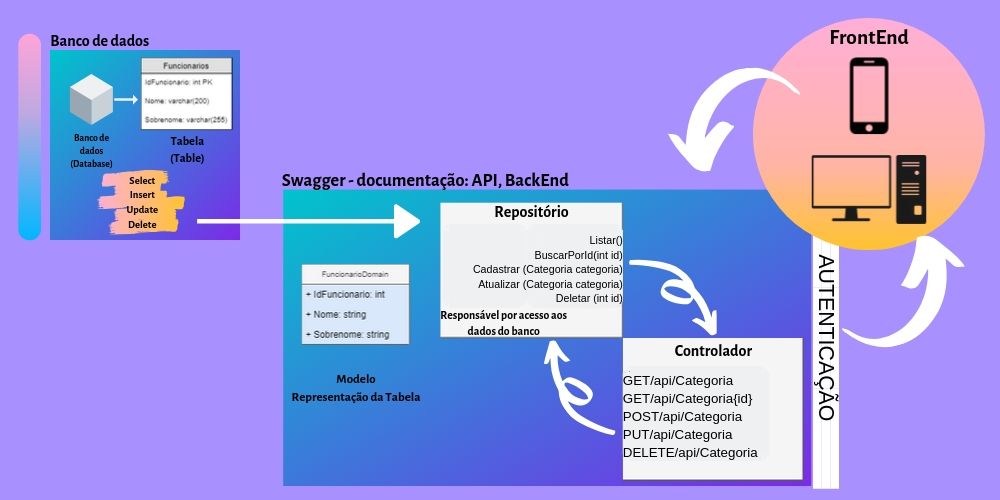
****

FrontEnd



Mobile

Arquitetura do Projeto



Referências

Links

Livros