**Documentação do Teste a Resoluti**

**Backend**

Linguagem C#, Asp net core web api

Para o projeto de Backend não usei camadas por ser um projecto simples, porém criei simplesmente pastas baseando o meu projeto na estrutura do Clean Arqutecture.

Em que tenho as seguintes pastas no projecto:

**Domain**: onde estão as minhas entidades que são mapeadas no banco de dados.

**Application**: é o onde encontramos os DTOs, as interfaces, o Mapping, os services onde estão a lógica do projecto,

**Infrastructure**: é responsável por ter acesso ao banco de dados, toda manipulação do banco de dados é tratado no Infrastruture.

**Controllers:** onde estão as controllers da minha aplicação

**Photo**: responsável por armazenar todas as fotos do projeto

**Migration**: onde são gerada as migrations do projeto

**Constantes**: é a pasta onde geralmente coloco as contantes e as mensagens retornada pela api etc.

O meu projeto é composto por 4 tabelas que são:

**User**: é a tabela que armazena usuários do sistema e onde é feita a consulta para o login.

**Person**: é a tabela que armazena as pessoas que serão registadas no sistema

**Address**: é a tabela responsável por armazenar os endereços da pessoa.

**Contat**: é a tabela responsável por armazenar os contatos da pessoa.

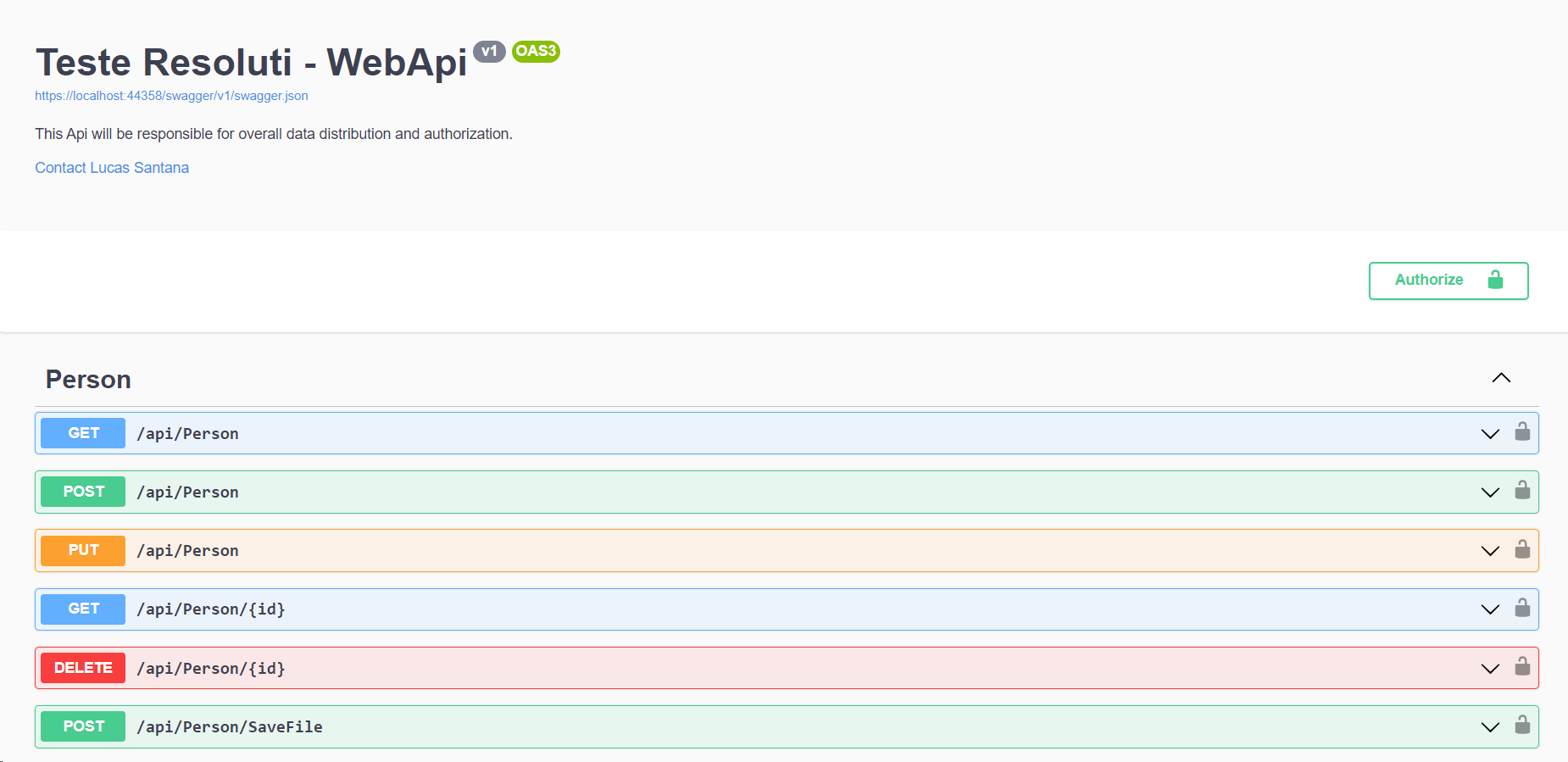
Não achei necessidade em ter uma tabela para armazenar os Estados e dos tipos de contato, como são todos conhecidos simplesmente adicionei num select no meu front end.

Para aparte de login não achei necessidade em usar o Identity, criei um login simples, configurei autenticação e autorização nas rotas excepto a de login.

Usei o AutoMapper para mapear os meus DTos para as minhas entidades.

As variáveis e os métodos foram escritos em Inglês, geralmente é um padrão adotado por mim, e as mensagens estão em Português.

No projeto tem a camada de testes (**Resoluti.UnitTests**) onde faço teste simplesmente do método de Login, para demostrar o conhecimento que tenho sobre testes unitários e de integração embora não seja o foco do teste.



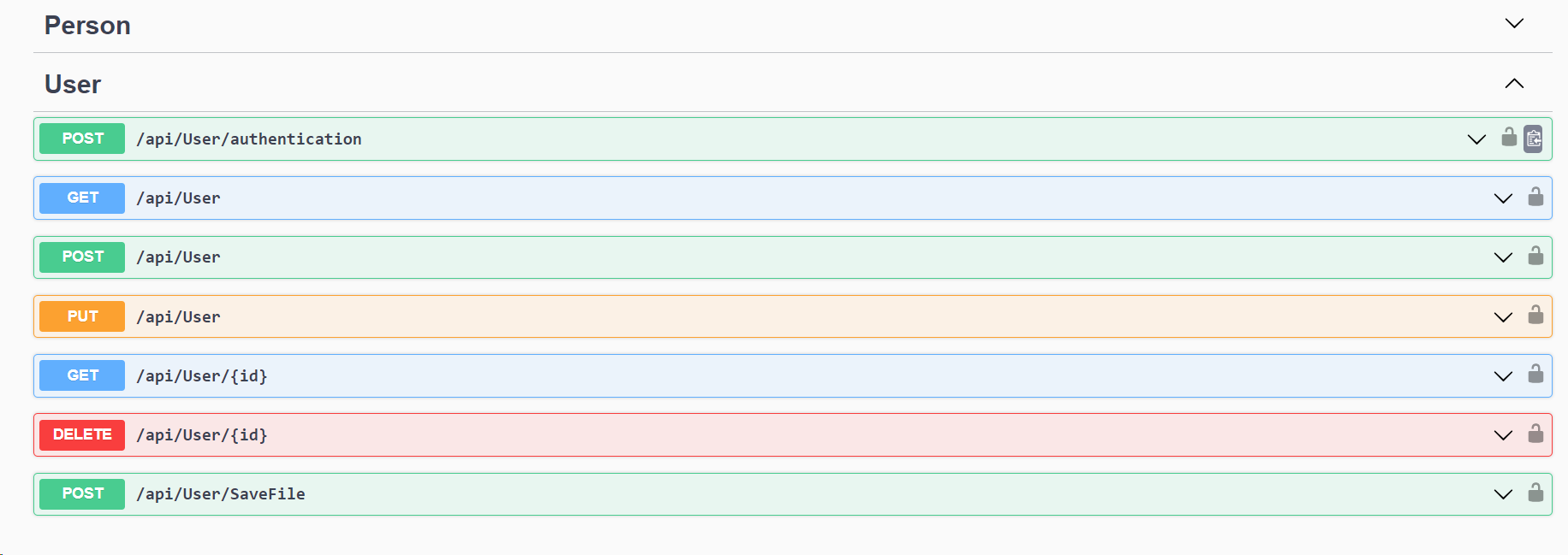


Figura-1

Imagem da api do projeto

**Front-End**

Angular, angular material

Para o projeto de front end usei o angular

Criei as telas de cadastro de pessoas, listar pessoas, cadastro de usuário, listar usuário, editar, eliminar, tela de login e também a funcionalidade de pesquisar.

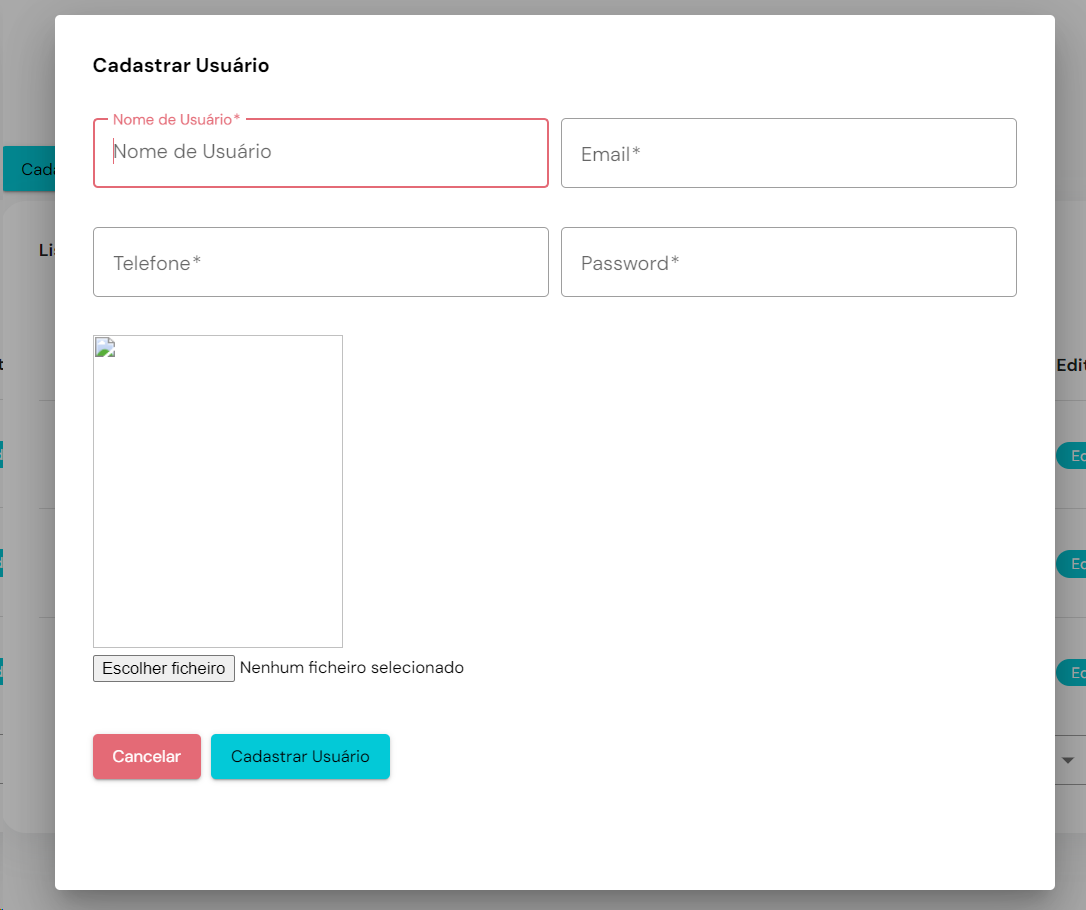


Figura -2

Tela de cadastro de usuário

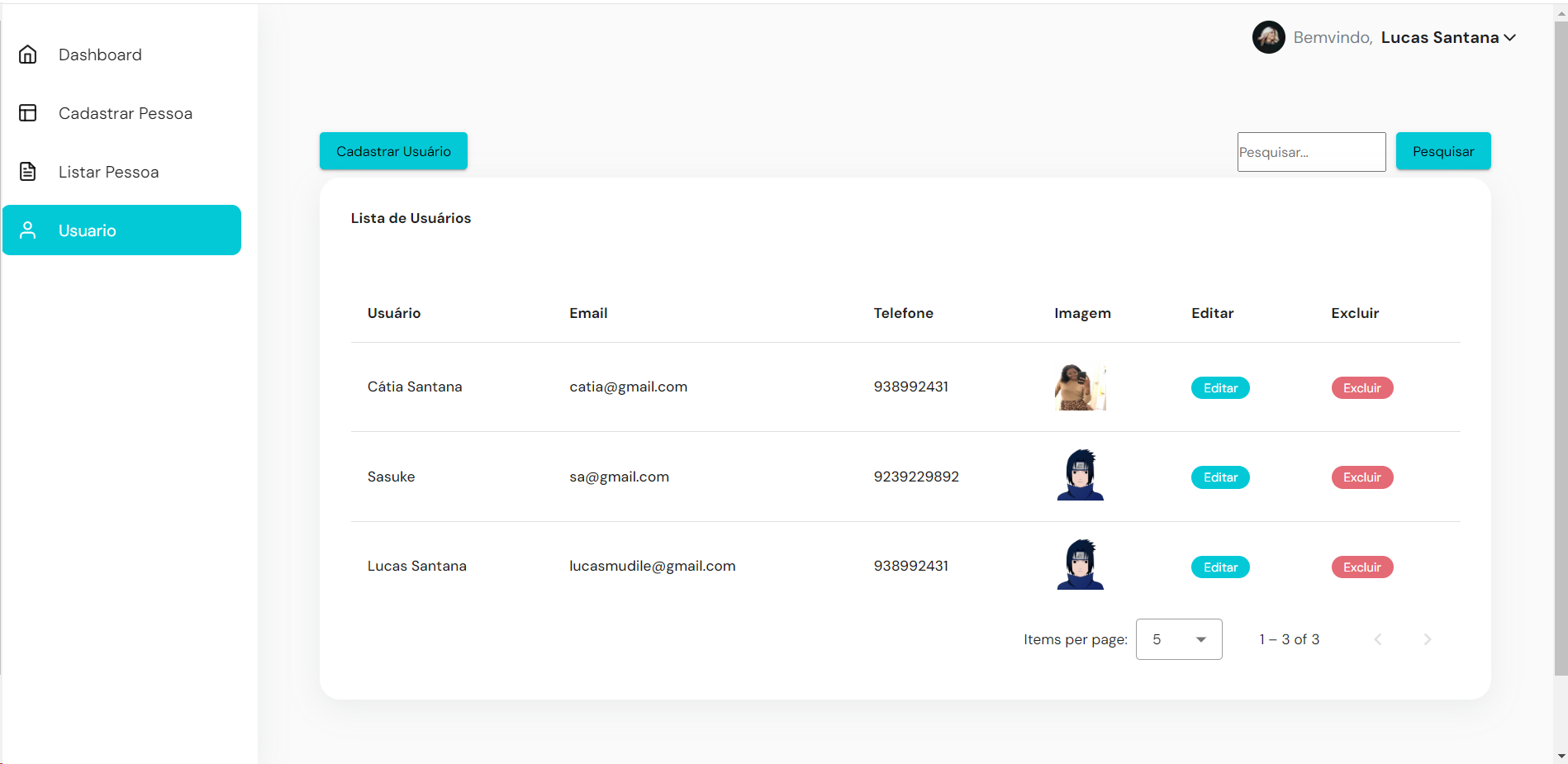


Figura -2

Tela de lista de usuários

Para o login tem dois usuários semeados / criados quando se gera a migration que são:

Email: [admin@gmail.com](mailto:admin@gmail.com)

Senha: 123

Email: [lucas@gmail.com](mailto:lucas@gmail.com)

Senha:123

Obs: Como é apenas um teste não achei necessidade de encriptar a senha.

O banco de dados usado no projeto é o **Sql Server, versão do Sql Server Manangemnet studio é a 19**

As tabelas do projeto e a base de dados

