

## 1.INTRODUÇÃO

A ExoApi é uma empresa de desenvolvimento de software focada na integração entre diferentes sistemas, softwares ou serviços. Utiliza as tecnologias mais recentes do mercado para ter performance, controle, segurança e escalabilidade para sua empresa e seus contratantes.

Uma API consiste em um meio de possibilitar a interação entre sistemas. A API desenvolvida para este projeto vai permitir o acesso a um banco de dados para um sistema web. A API possibilita a criação de uma estrutura JSON que por sua vez vai viabilizar o tratamento do Front End para a renderização e execução das funcionalidades do sistema. A hospedagem vai ser local.

O projeto consiste no armazenamento dos projetos da empresa para a administração de seus trabalhos.

## 2.REQUISITOS DO SISTEMA

1. Ler todos os projetos (título, status, data de início, tecnologias/requisitos/área);
2. Buscar informações somente de um projeto;
3. Deletar um projeto;
4. Atualizar todas as informações de um projeto ou somente uma informação específica;
5. Disponibilizar a documentação da API;
6. Retornar os status de respostas corretos de acordo com as melhores práticas;
7. Retornar os dados em formato JSON;
8. Utilizar frameworks para a criação de API;
9. Disponibilizar o recurso de deletar um projeto, somente para usuários que possuem permissão;

## 3.FERRAMENTAS

Desenvolvimento da API com o Visual Studio 2022, banco de dados com o SQL Server 2022, interação com a API com o Insomnia e Swashbuckle.AspNetCore, pacotes Microsoft.AspNetCore.Authentication.JwtBearer e

Microsoft.EntityFrameworkCore.Sqlserver. Projeto criado com o API Web do ASP.NET Core.

#### 4.RESULTADO

As figuras 1 a 12 representam o comportamento da API de acordo a requisição feita para atender os requisitos do sistema e informações sobre o projeto.

Criando informações de projetos com usuário autenticado.

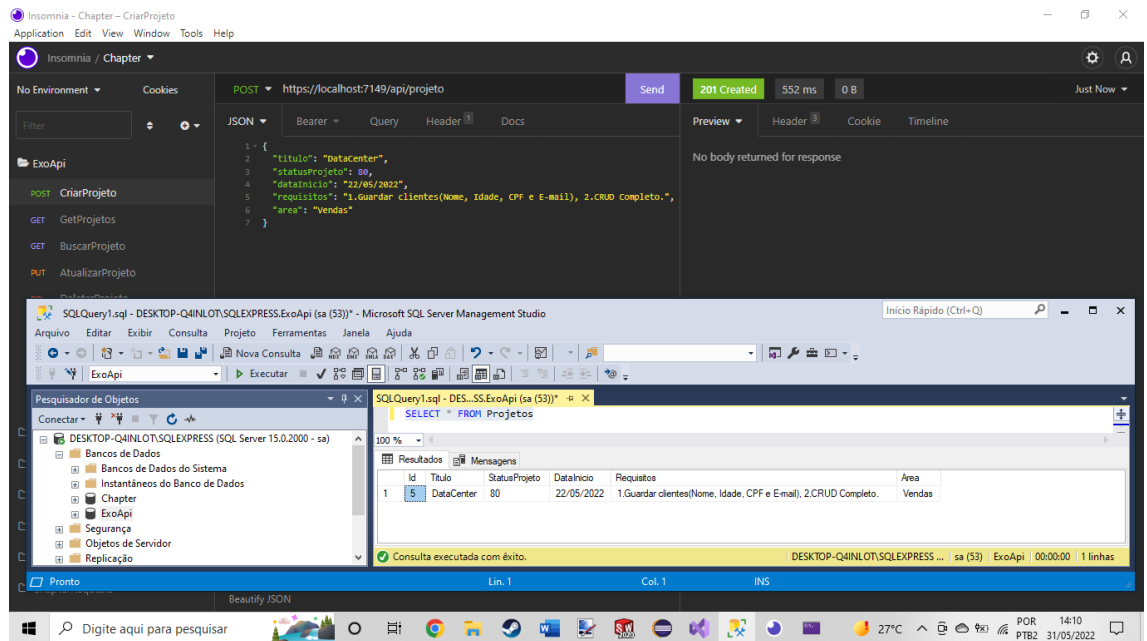


Figura 1 – requisição Post com usuário autenticado.

#### 1. Ler todos os projetos (título, status, data de início, tecnologias/requisitos/área);

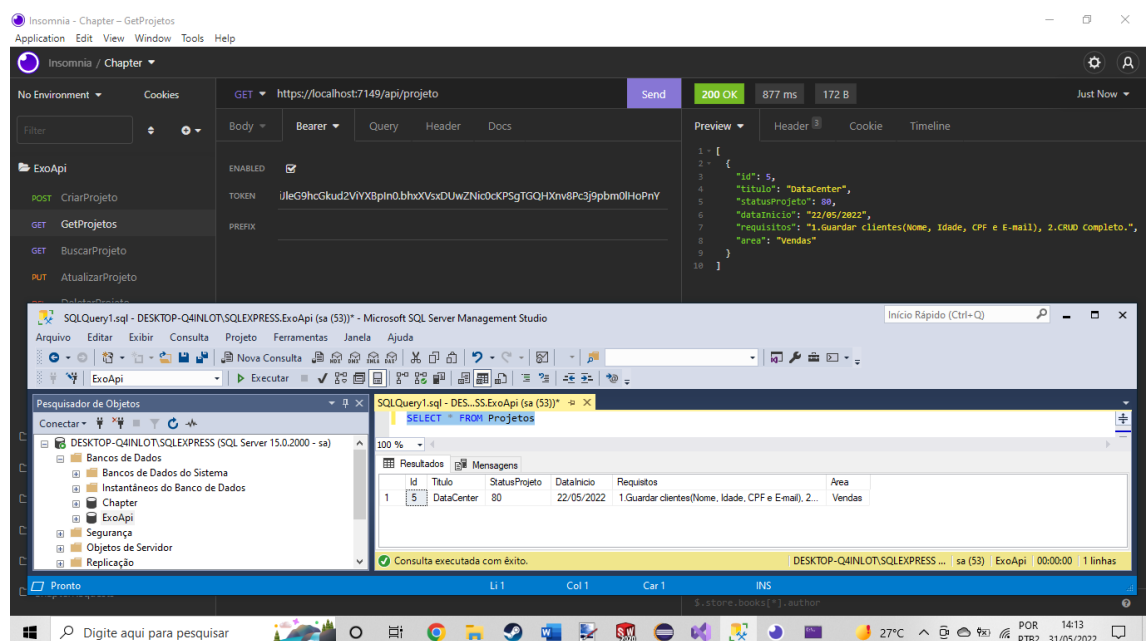


Figura 2 – requisição Get com usuário autenticado.

## 2. Buscar informações somente de um projeto;

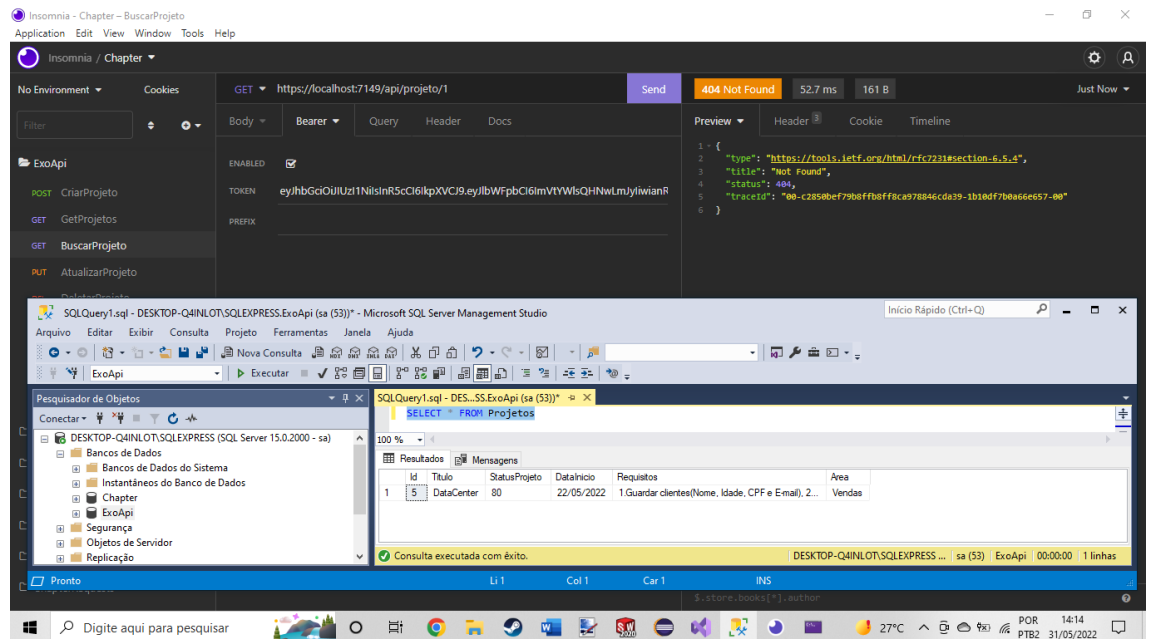


Figura 3 – requisição Get por id com usuário autenticado porém com id errado, retorno de erro tratado.

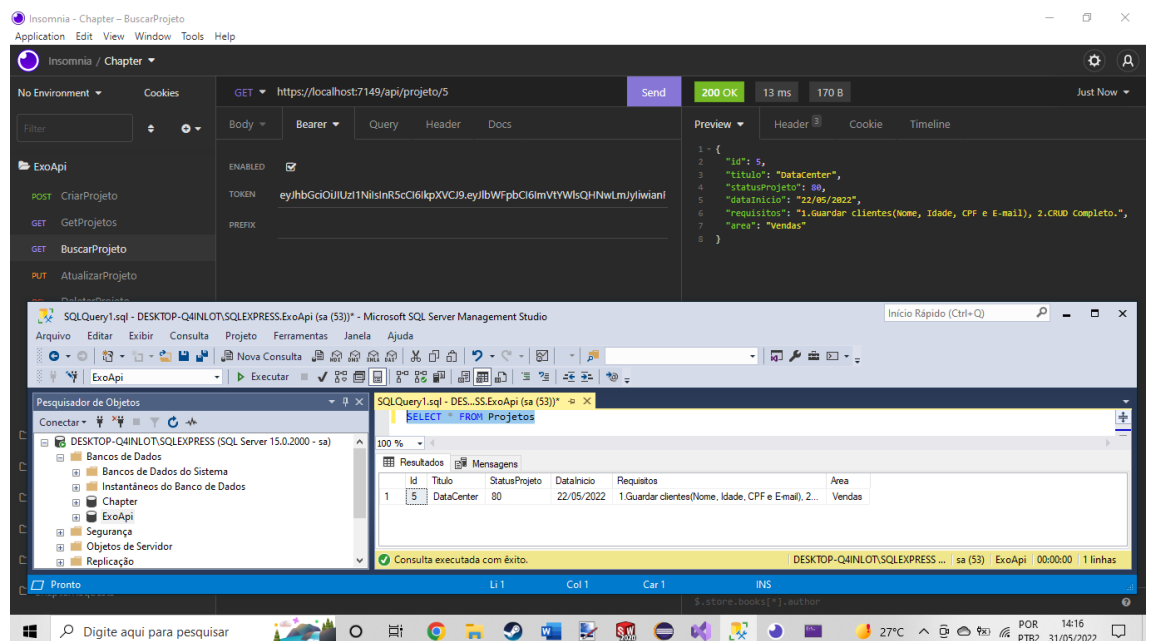


Figura 4 – requisição Get por id com usuário autenticado com id de projeto correto.

## 3. Deletar um projeto;

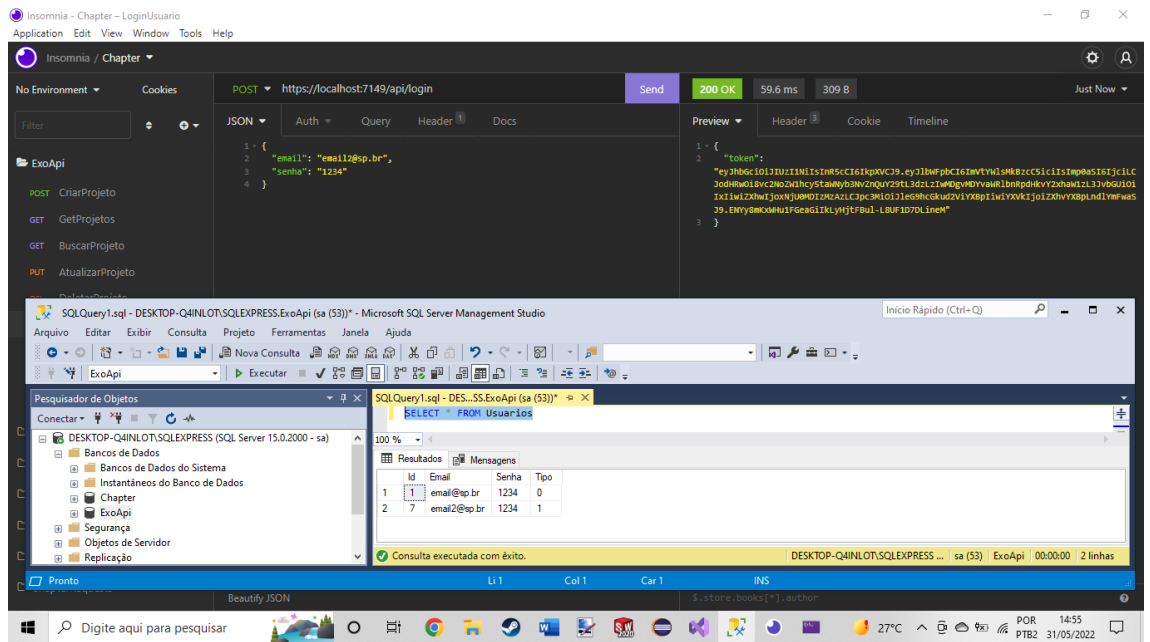


Figura 5 – autenticação de usuário com o tipo 1.

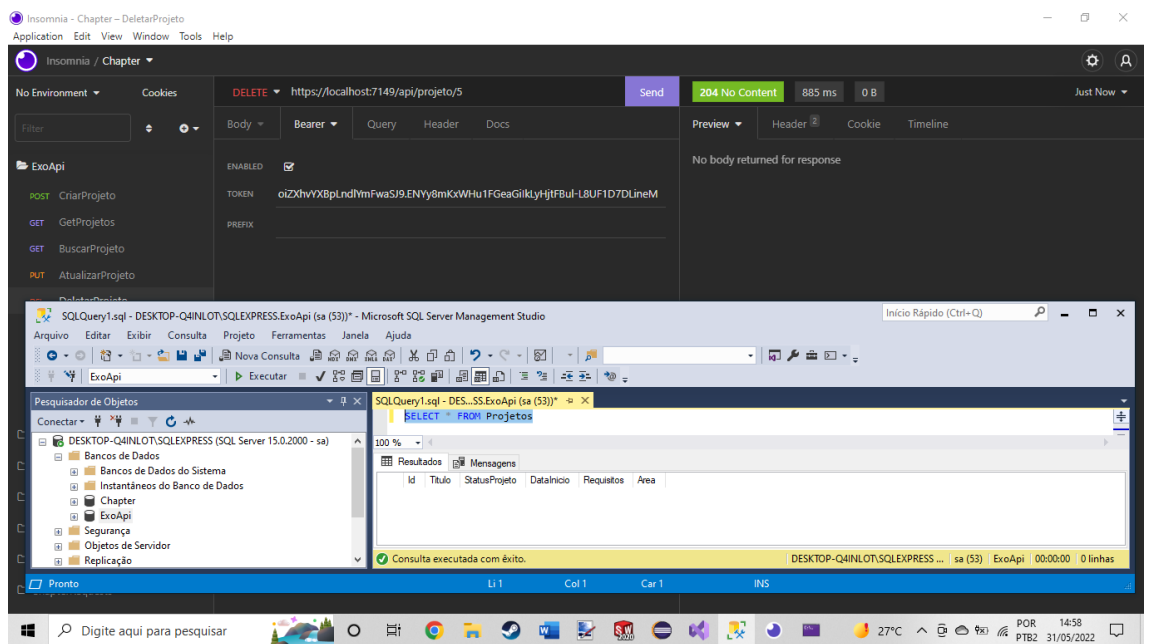


Figura 6 – Delete com usuário autenticado com o tipo 1.

- Atualizar todas as informações de um projeto ou somente uma informação específica;

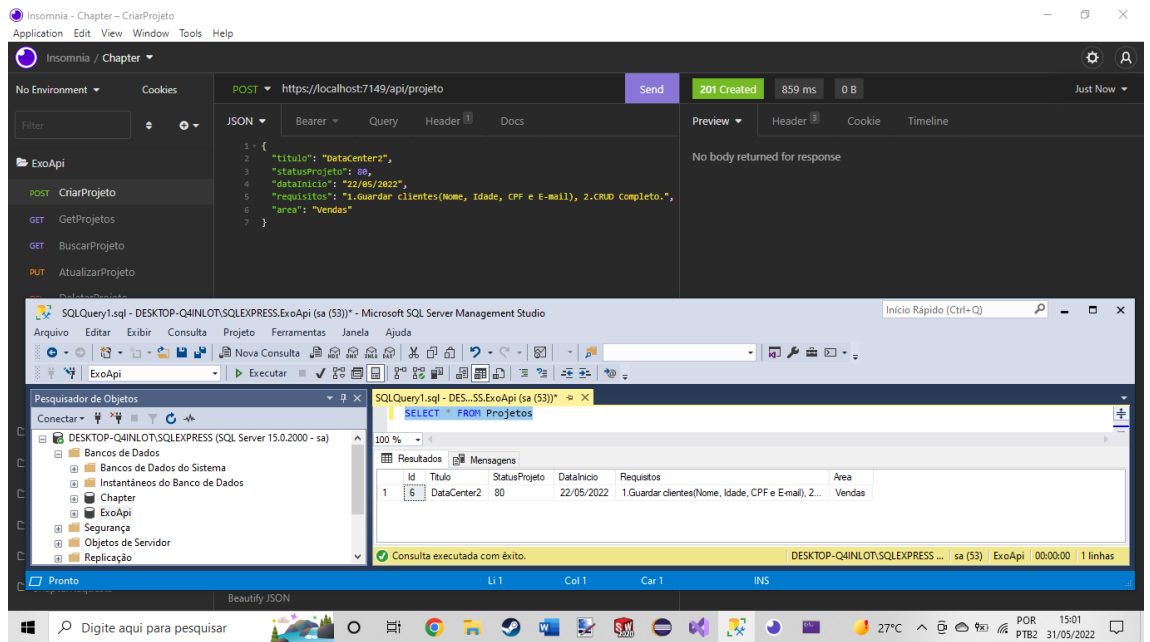


Figura 7 – projeto criado com usuário autenticado.

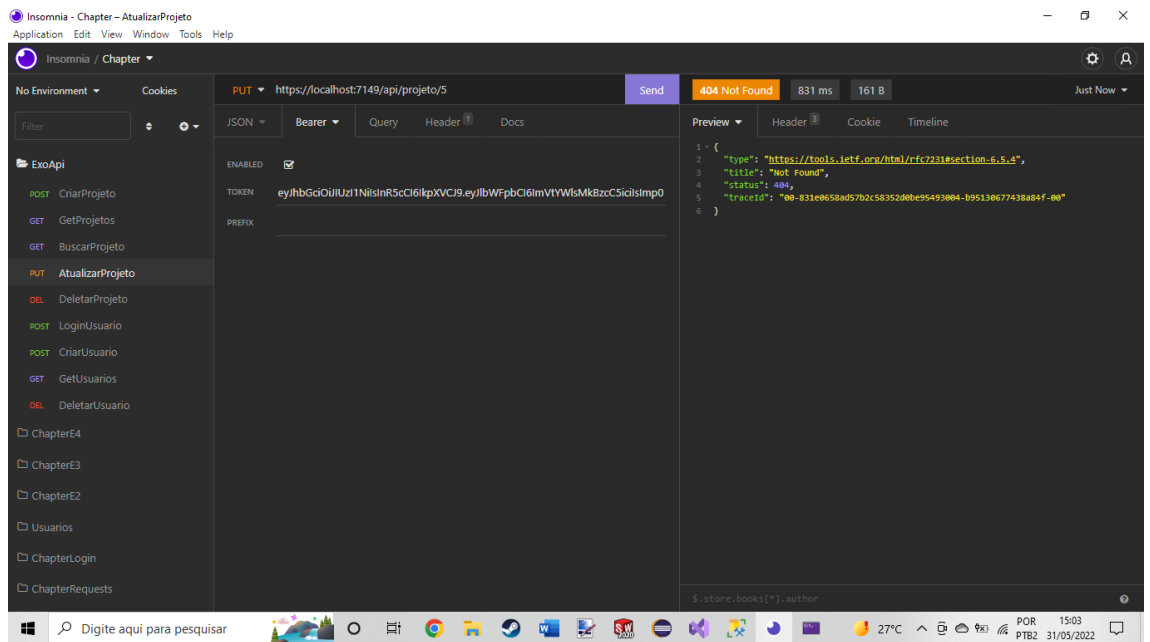


Figura 8 – requisição put com tratamento de erro por não ter o id solicitado.

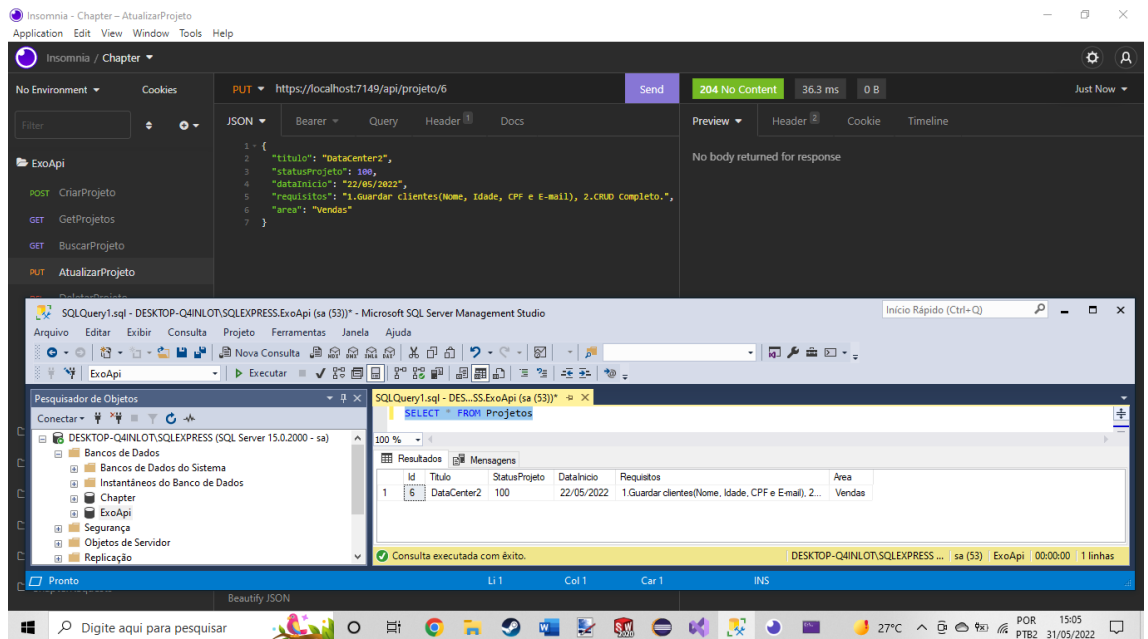


Figura 9 – requisição put com usuário autenticado atualizando apenas a informação do atributo StatusProjeto.

5. Disponibilizar a documentação da API;

GitHub: <https://github.com/Andpedroso/SENAI-UC14-AT3>

6. Retornar os status de respostas corretos de acordo com as melhores práticas;  
Ver figuras 3 e 8.
7. Retornar os dados em formato JSON;

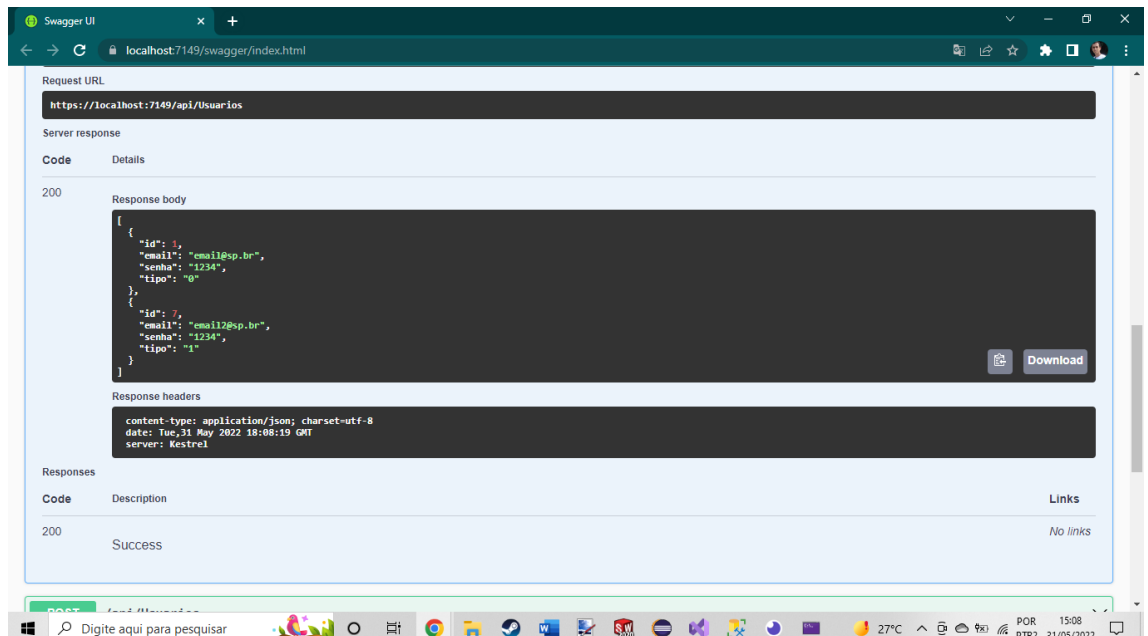


Figura 10 – retorno de dados em formato JSON.

8. Utilizar frameworks para a criação de API;

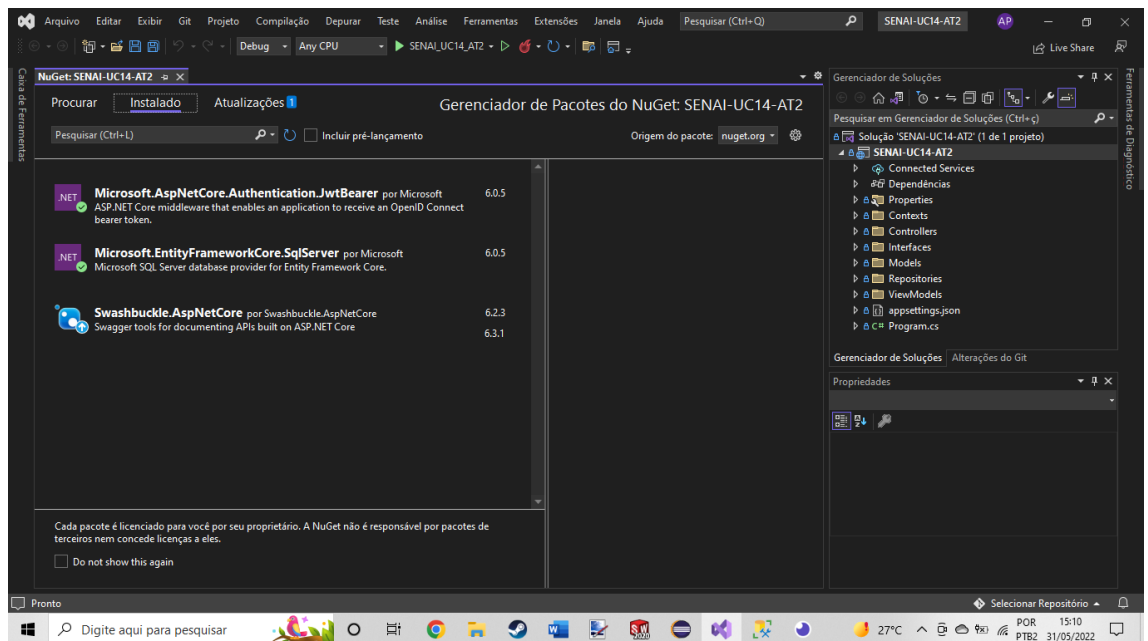


Figura 11 – API Web do ASP.NET Core e os pacotes utilizados.

9. Disponibilizar o recurso de deletar um projeto, somente para usuários que possuem permissão;

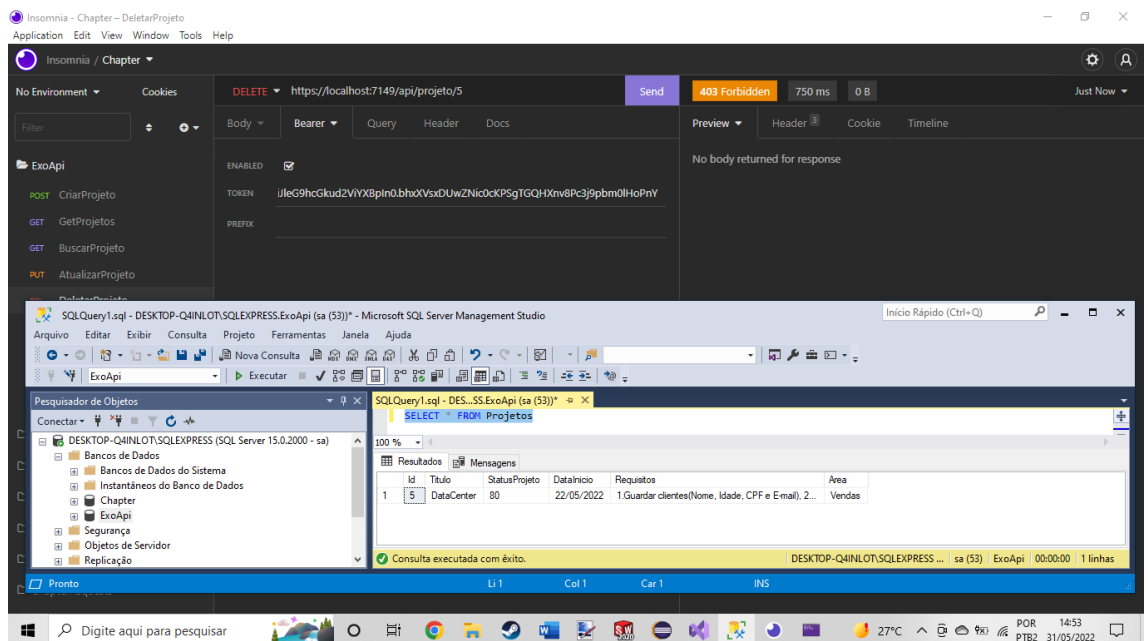


Figura 12 – tentativa de Delete com usuário autenticado do tipo 0.