**SP Medical Group**

Documentação

SP Medical Group

Sumário

[1. Resumo 3](#_gjdgxs)

[2. Descrição do projeto 3](#_30j0zll)

[3. Banco de dados relacional 3](#_1fob9te)

[4. Modelagem de dados 3](#_3znysh7)

[Modelo Conceitual 4](#_2et92p0)

[Modelo Lógico 4](#_tyjcwt)

[Modelo Físico 5](#_3dy6vkm)

[Cronograma](#_1t3h5sf) 6

5. [Trello](#_4d34og8) 6

6. Back-End 7

Funcionalidades 7.1

Sistema Web 7.2

Perfis de usuário 7.3

Funcionalidades 7.4

Sistema Mobile 7.5

Perfis de usuário 7.6

Funcionalidades 7.7

# Resumo

Documentação feita com o intuito de mostrar todo o desenvolvimento do projeto, envolvendo as modelagens e o resumo do projeto

# Descrição do projeto

Uma nova clínica médica chamada SP Medical Group, empresa de pequeno porte que

atua no ramo da saúde, foi criada pelo médico Fernando Strada em 2020 na região da

Paulista em São Paulo. Fernando tem uma equipe de médicos que atuam em diversas

áreas (pediatria, odontologia, gastrenterologia etc.).

Sua empresa, por ser nova, iniciou a administração dos registros de forma simples,

utilizando softwares de planilhas eletrônicas e, com o sucesso da clínica, sua gestão

passou a se tornar complicada devido à alta demanda dos pacientes.

Fernando Strada contratou para desenvolver um sistema web/mobile integrado

onde seja possível realizar a gestão da clínica de forma automatizada e ter acesso fácil

aos dados sobre as informações de seus pacientes.

# Banco de dados relacional

Banco de dados relacional serve para armazenar dados relacionados entre si, com uma importância fundamental, pois no caso se houvesse algum rompimento no sistema, seria um prejuízo enorme, pois todos os dados, desde um funcionário ao cliente seria apagado.

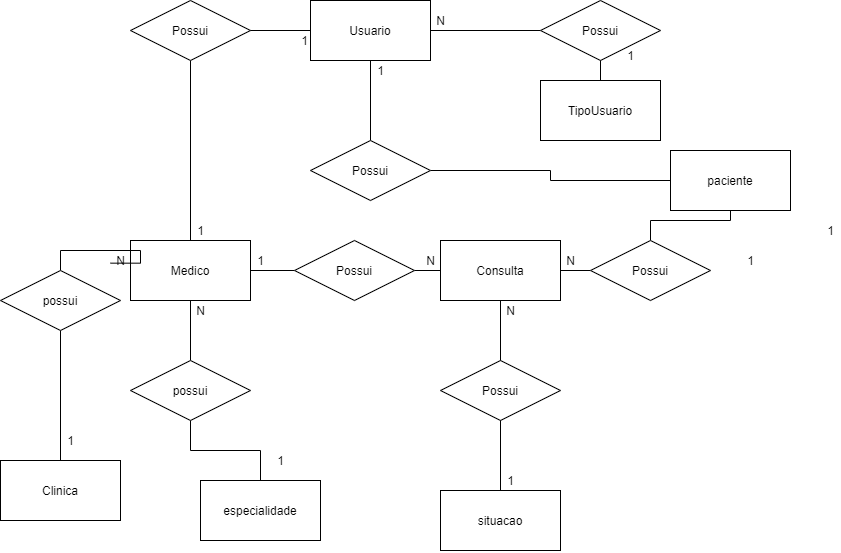
# Modelagem de dados

No caso da modelagem de dados, serve como guia, para facilitação e o compreendimento, assim fazendo com que o desenvolvedor deslanche em questão a tempo de trabalho.

## 

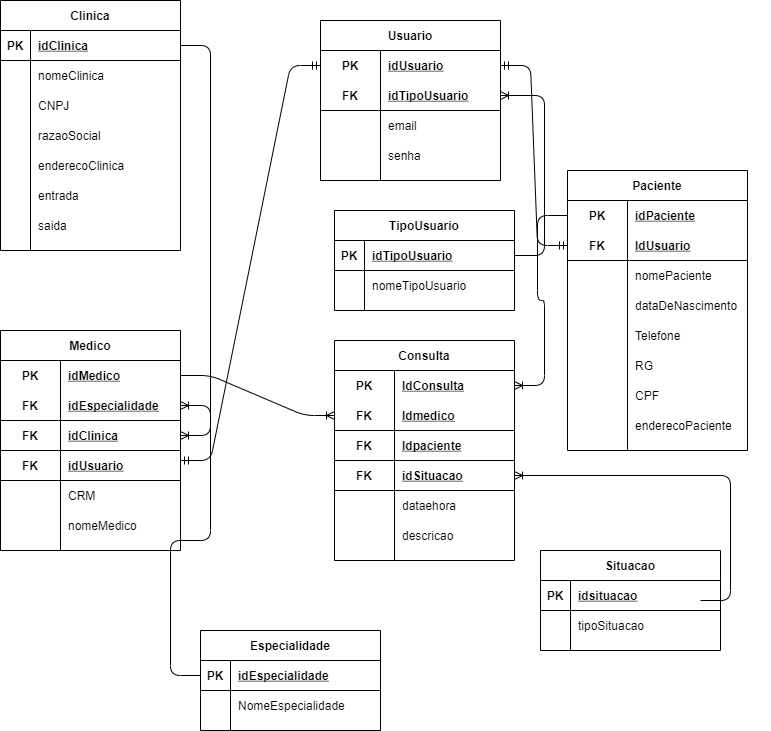
## Modelo Conceitual

Modelagem conceitual define as relações, entidades e cardinalidade em conjunto, demonstrando o raciocínio do banco de dados.



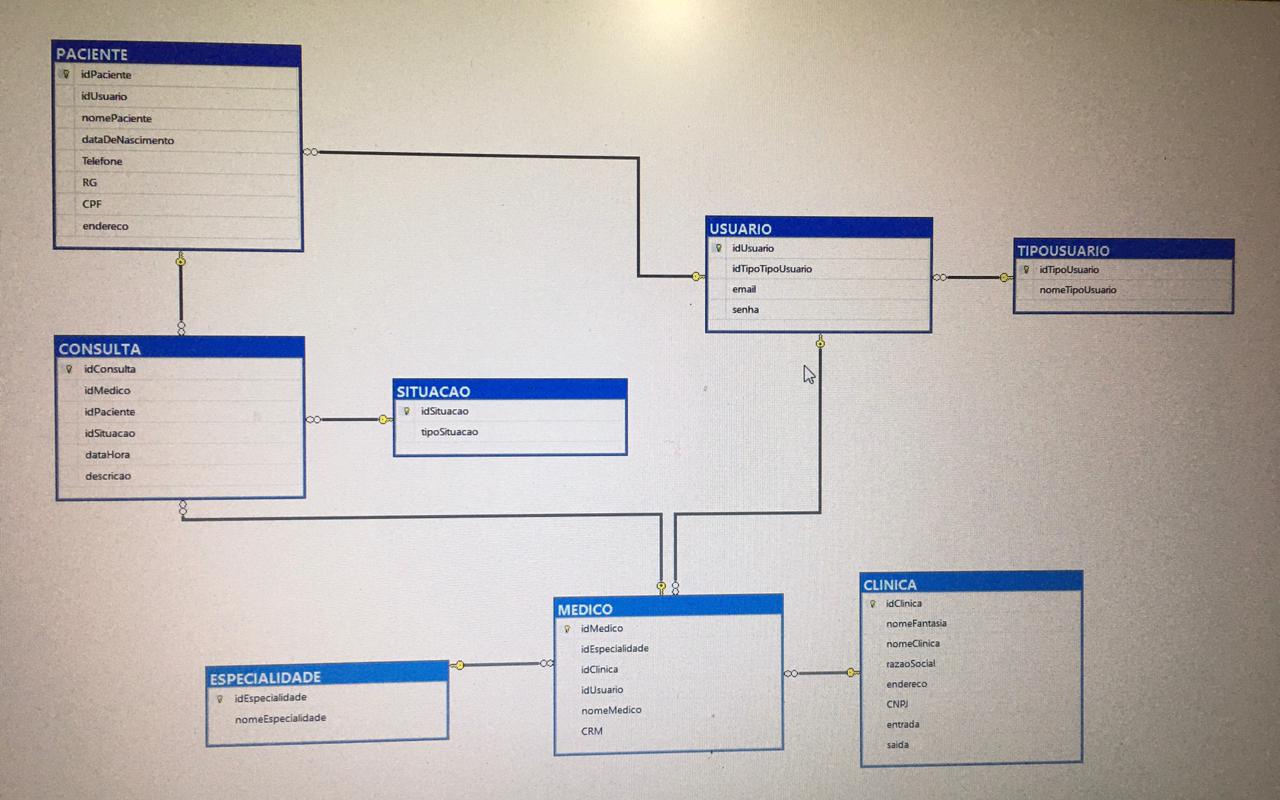
## Modelo Lógico

Com o modelo lógico utilizamos uma forma mais operacional equivalente ao banco de dados, envolvendo chaves (estrangeiras e primárias)



## Modelo Físico

O modelo físico almeja uma forma mais visual no Excel envolvendo planilhas, sendo assim facilitando a compreensão, colocando cada entidade no lugar onde é funcional.



## Cronograma

|  |  |  |  |
| --- | --- | --- | --- |
|  | Dia 1 | Dia 2 | Dia 3 |
| Modelo Conceitual | X |  |  |
| Modelo Lógico | X |  |  |
| Modelo Físico | X |  |  |
| DDL |  | X |  |
| DML |  | X |  |
| DQL |  |  | X |

### Trello

<https://trello.com/invite/b/A95yLQbS/4bbf3a33a587fbc1d543577d0b2581db/sp-medical-group>

### Back-End

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 |
| Organização | X |  |  |  |
| Criação do Projeto |  | X |  |  |
| Configurar o projeto |  | X |  |  |
| Endpoints |  |  | X |  |
| Postman |  |  |  | X |

**UTLIZAÇÃO**

**API** (Application Programming Interface) seria uma interface de programação de aplicativos.

**JSON** (JavaScript Object Notation) seria uma notação de objetos JAVASCRIPT.

**HTTP** (Hypertext Transfer Protocol) Protocolo de Transferência de Hipertexto.

REST (Representational State Transfer) – Interface de Programação de Aplicativos.

**CONCEITO DE API**

API faz parte de uma interface e que permitem a criação de plataformas de maneira mais simples e prática para desenvolvedores. A partir disto é possível criar softwares, aplicativos e programas.

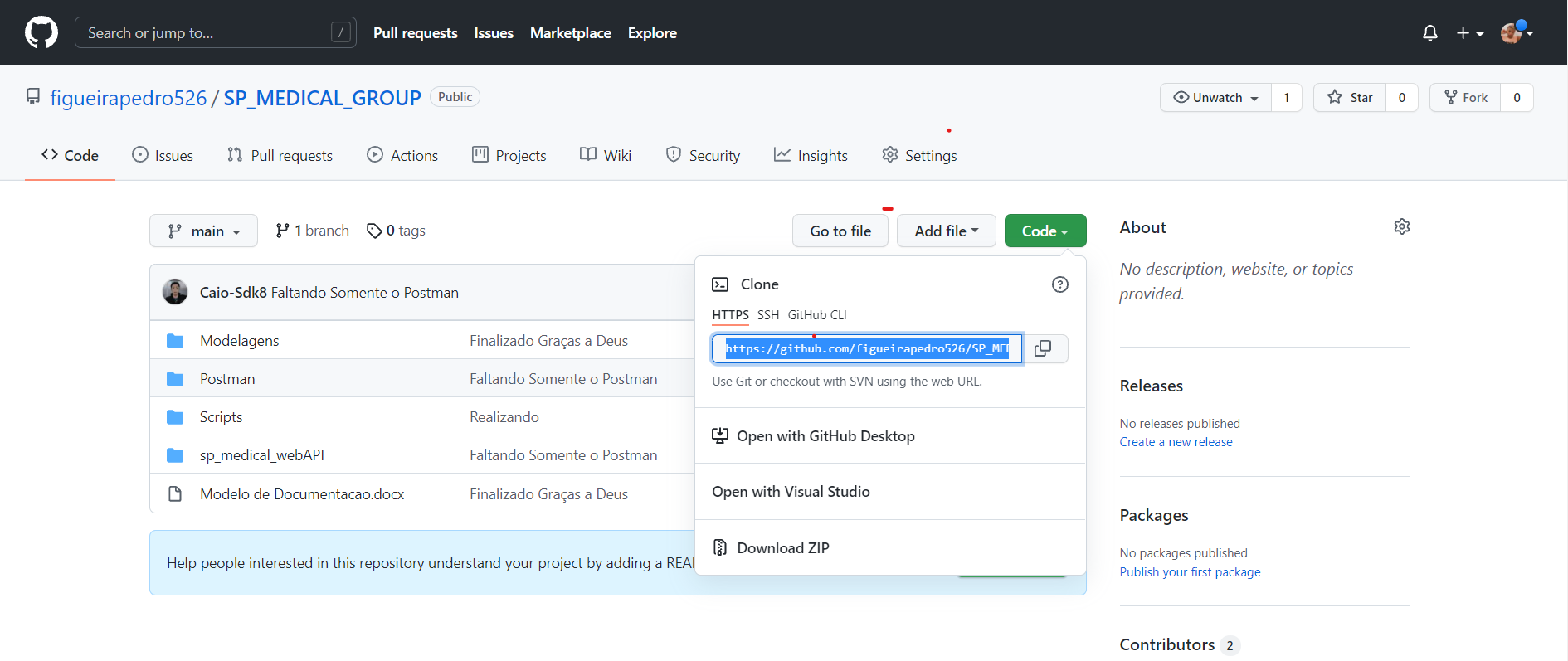
API é um conjunto de padrões e instruções estabelecidos para utilização do software, definindo as requisições e as respostas seguindo o protocolo HTTP, neste caso expresso no formato JSON, para que seja possível acessar o sistema em diversos dispositivos distintos sem a preocupação com a linguagem que será utilizada por estes.

**ARQUITETURA UTILIZADA**

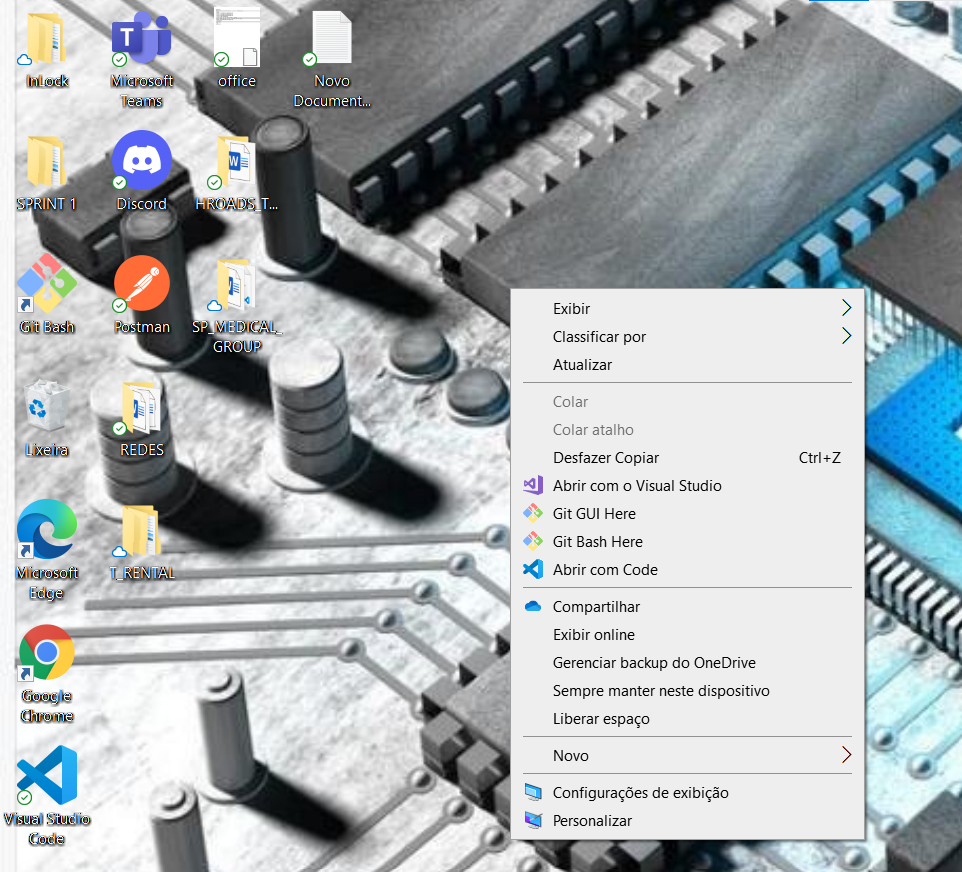
No caso do projeto abordado ao SP MEDICAL GROUP, foi incluído o Entity Framework, com ênfase ao DataBase First, ensinados em aula e efetuado com clareza.

**Passo a Passo Execução da API**

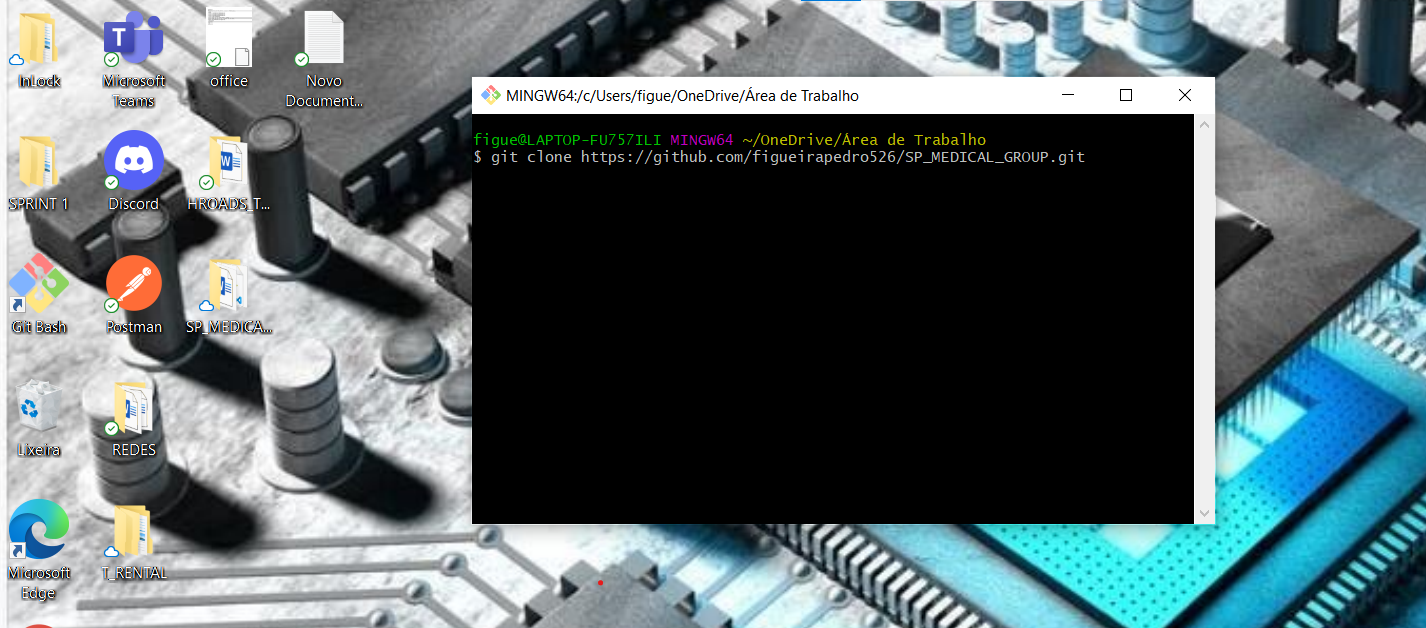
Pra dar inicio ao processo, devemos copiar o link do repositório do GitHub



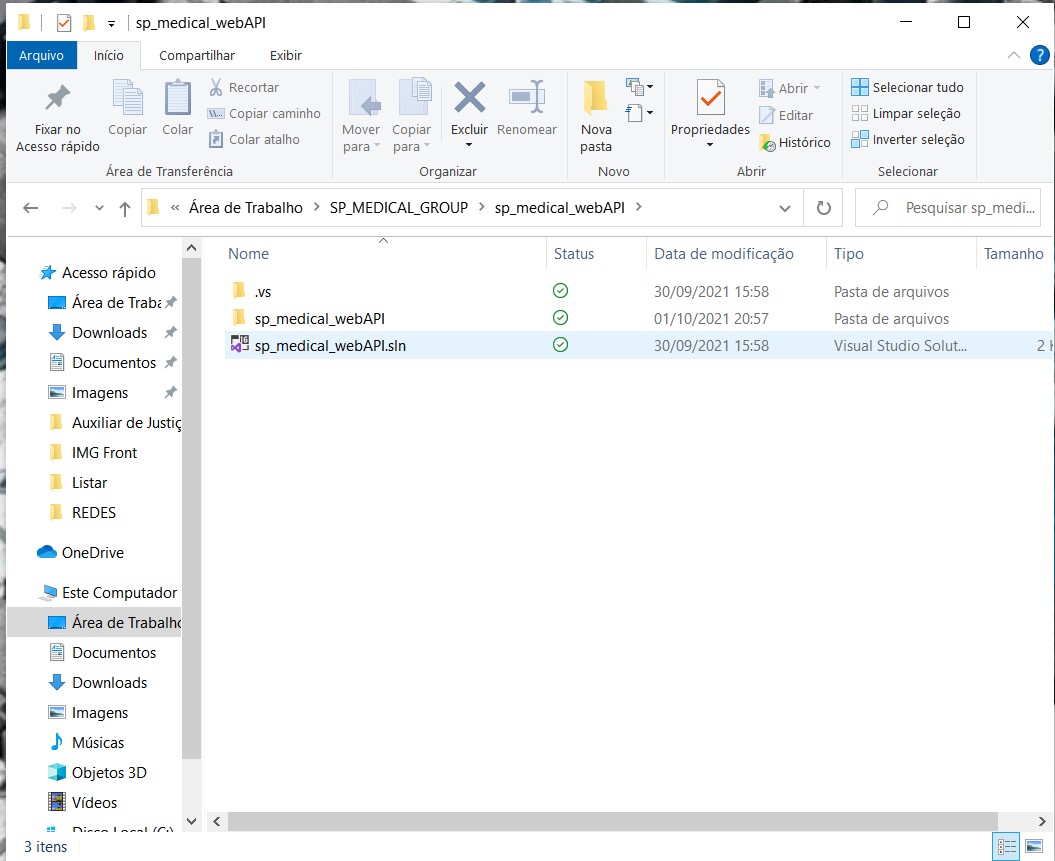
Abrir o GIT BASH HERE para colar o link do repositório



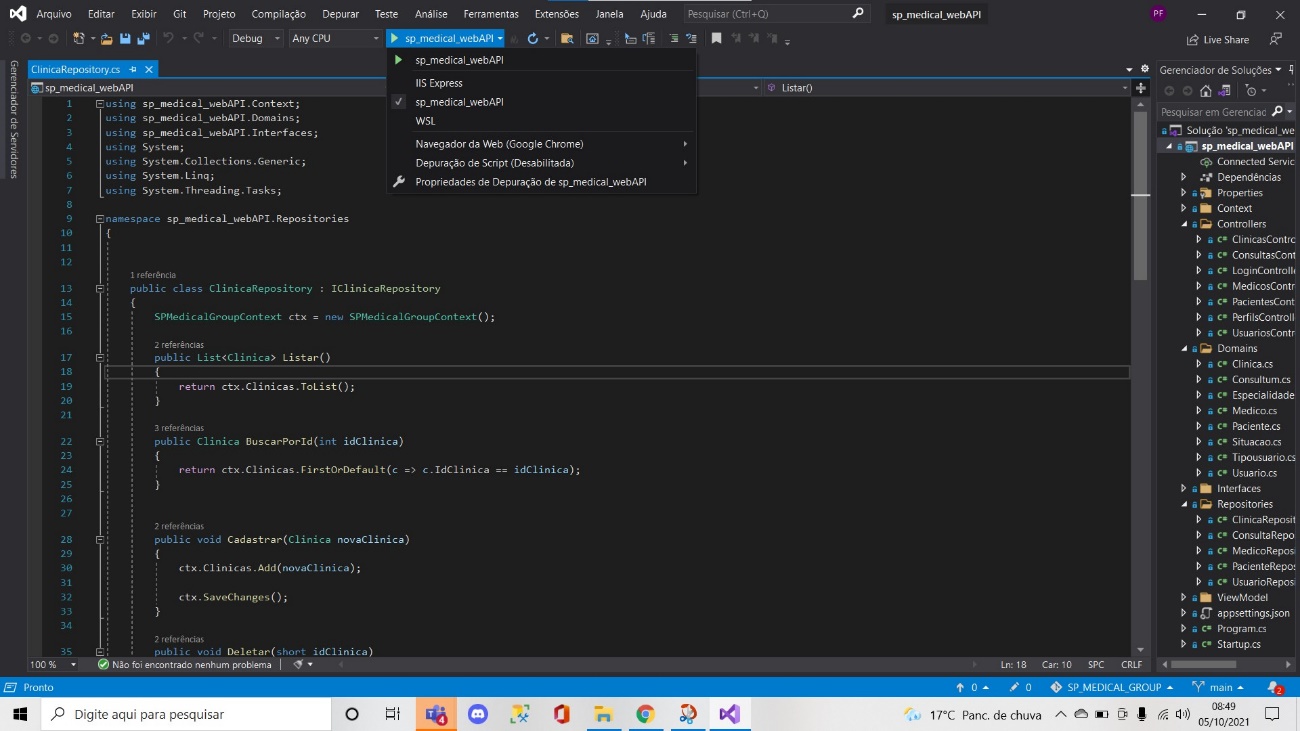
Dar o **git clone,** colando o link do repositório...... Automaticamente abrirá a pasta do projeto



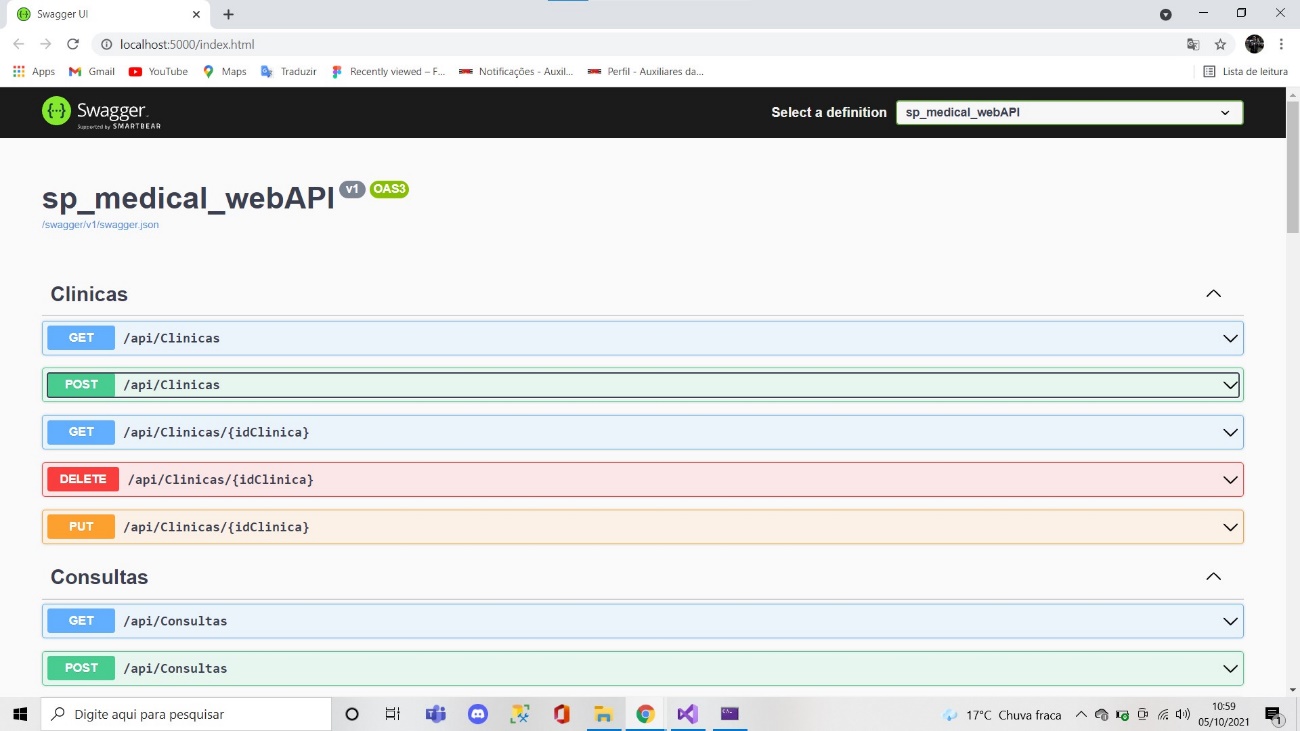
Em seguida, abrir a pasta e seguir esse caminho, ate chegar no arquivo do visual studio “sp\_medical\_group”



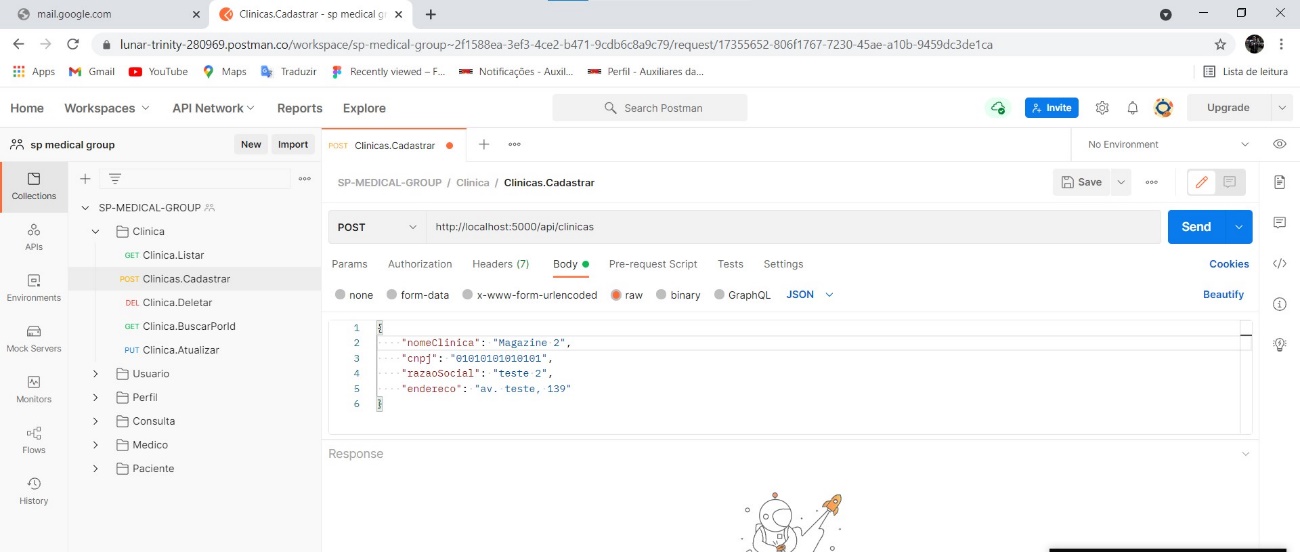
Ao abrir, você terá toda acessibilidade da API criada... Sendo assim escolher o compilador com o nome do arquivo “sp\_medical\_group” e selecionar.



Ao compilar, será redirecionado ao sweguer, onde o visualizador terá toda a visualização de todo o projeto



Posteriormente, você poderá abrir o postman, e importar o link oferecido na pasta POSTMAN, e poderá fazer cadastro, atualização, deletar...... Simbolizando um processo de um site em si. E para efetuar a operação que você fez, tem que acionar o send (em azul).



## **Funcionalidades**

Insira as funcionalidades que a API atende.

Por exemplo:

## **Sistema Web**

### **Perfis de usuário:**

1. **Administrador**: Área administrativa da escola;
2. **Comum**: Pode ser um espectador ou palestrante.

### **Funcionalidades:**

1. O **administrador** poderá cadastrar qualquer tipo de usuário;
2. O **administrador** poderá cadastrar os dados da instituição;
3. O **administrador** poderá cadastrar os tipos de eventos;
4. O **administrador** poderá cadastrar eventos livres ou restritos;
5. Qualquer usuário autenticado poderá ver todos os eventos cadastrados;
6. O usuário **comum** poderá ver os eventos que participará;
7. O usuário **comum** poderá se inscrever para assistir a um evento;
8. O **administrador** poderá aprovar a participação do espectador de um determinado evento restrito;

## **Sistema Mobile**

### **Perfis de usuário:**

1. **Administrador**: Área administrativa da escola;
2. **Comum**: Pode ser um espectador ou palestrante;

### **Funcionalidades:**

1. Qualquer usuário autenticado poderá ver todos os eventos cadastrados;
2. O usuário **comum** poderá se inscrever para assistir um evento;
3. O usuário **comum** poderá ver os eventos que participará;