**NOMEPROJETO**

Documentação

Sumário

[1. Resumo 3](#_Toc71092221)

[2. Descrição do projeto 3](#_Toc71092222)

[3. Modelagem de Dados 3](#_Toc71092223)

[Modelo Conceitual 3](#_Toc71092224)

[Modelo Lógico 3](#_Toc71092225)

[Modelo Físico 3](#_Toc71092226)

[Cronograma 3](#_Toc71092227)

[Trello 4](#_Toc71092228)

[4. Back-End 4](#_Toc71092229)

[Funcionalidades 5](#_Toc71092230)

[Sistema Web 5](#_Toc71092231)

[Perfis de usuário: 5](#_Toc71092232)

[Funcionalidades: 5](#_Toc71092233)

[Sistema Mobile 5](#_Toc71092234)

[Perfis de usuário: 5](#_Toc71092235)

[Funcionalidades: 5](#_Toc71092236)

# Resumo

Bom nesse projeto, tive que desenvolver o Projeto-SP-Med-Group, contendo modelagem, Banco de dados , Api e o html/css. Ele é sobre uma clinica medica e nela é possível organizar, os dados dos médicos, dos paciente etc...

# 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.

# Modelagem de Dados

A modelagem de dados é a parte da ciência da computação que se encarrega de construir estruturas que permitam armazenar e recuperar informações em contextos específicos.

## Modelo Conceitual

## O **modelo conceitual** concentra-se no mais alto nível de abstração e não leva em conta o banco de dados em si, mas a forma como as estruturas serão criadas para armazenar os dados

## 

## 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](https://www.luis.blog.br/chave%20primaria-chave-estrangeira-e-candidata.aspx), 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.





## Cronograma

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 | Dia 5 |
| Modelo Conceitual | X |  |  |  |  |
| Modelo Lógico | X |  |  |  |  |
| Modelo Físico | x |  |  |  |  |

### Trello

https://trello.com/b/v3bxdsnZ/projeto-sp-med-group

# Back-End

Insira aqui uma descrição sobre a arquitetura utilizada para a criação do back-end.

Por exemplo:

O código do sistema foi desenvolvido através de uma API utilizando o Microsoft Visual Studio.

**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.

Além disso, foi utilizado o estilo de arquitetura REST.

**API** – Application Programming Interface – Interface de Programação de Aplicativos.

**HTTP** – Hypertext Transfer Protocol – Protocolo de Transferência de Hipertexto.

**JSON** – JavaScript Object Notation – Notação de Objetos JavaScript.

**REST** – Representational State Transfer – Interface de Programação de Aplicativos.

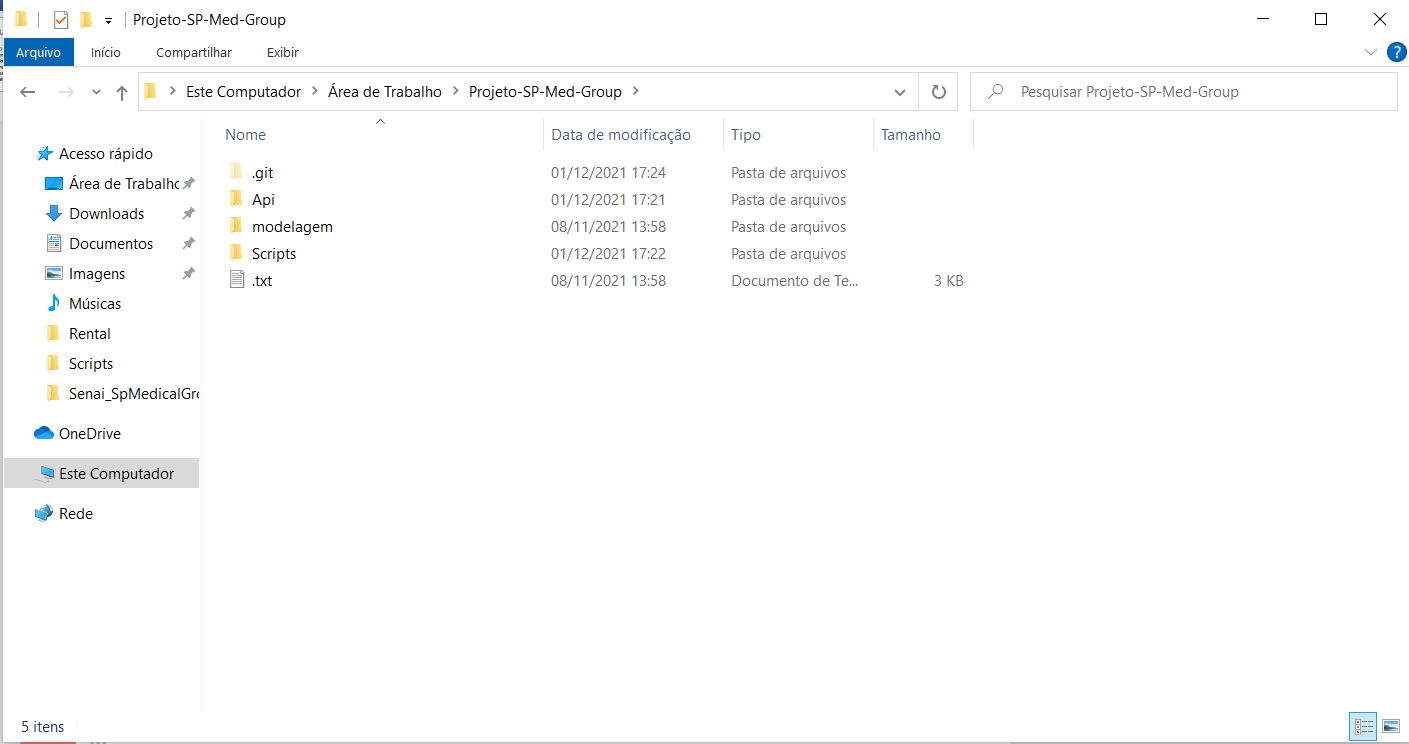
**Entity Framework**: o Entity Framework é um conjunto de tecnologias no ADO.NET que dão suporte ao desenvolvimento de aplicativos de software orientado a dados. Os arquitetos e desenvolvedores de aplicativos orientados a dados lutam com a necessidade de realizar dois objetivos muito diferentes. Precisam modelar as entidades, as relações e a lógica dos problemas de negócios que estão solucionando e também precisam trabalhar com os mecanismos de dados usados para armazenar e recuperar os dados.

-**Swagger** : Usado para adquirir uma documentação dos endpoints.

-**DataBase First**: É um tipo de estratégia para começar a sua API, mais para usar ela precisa ter um banco de dados já existente, para que ela consiga gerar classes (Domain, Context).

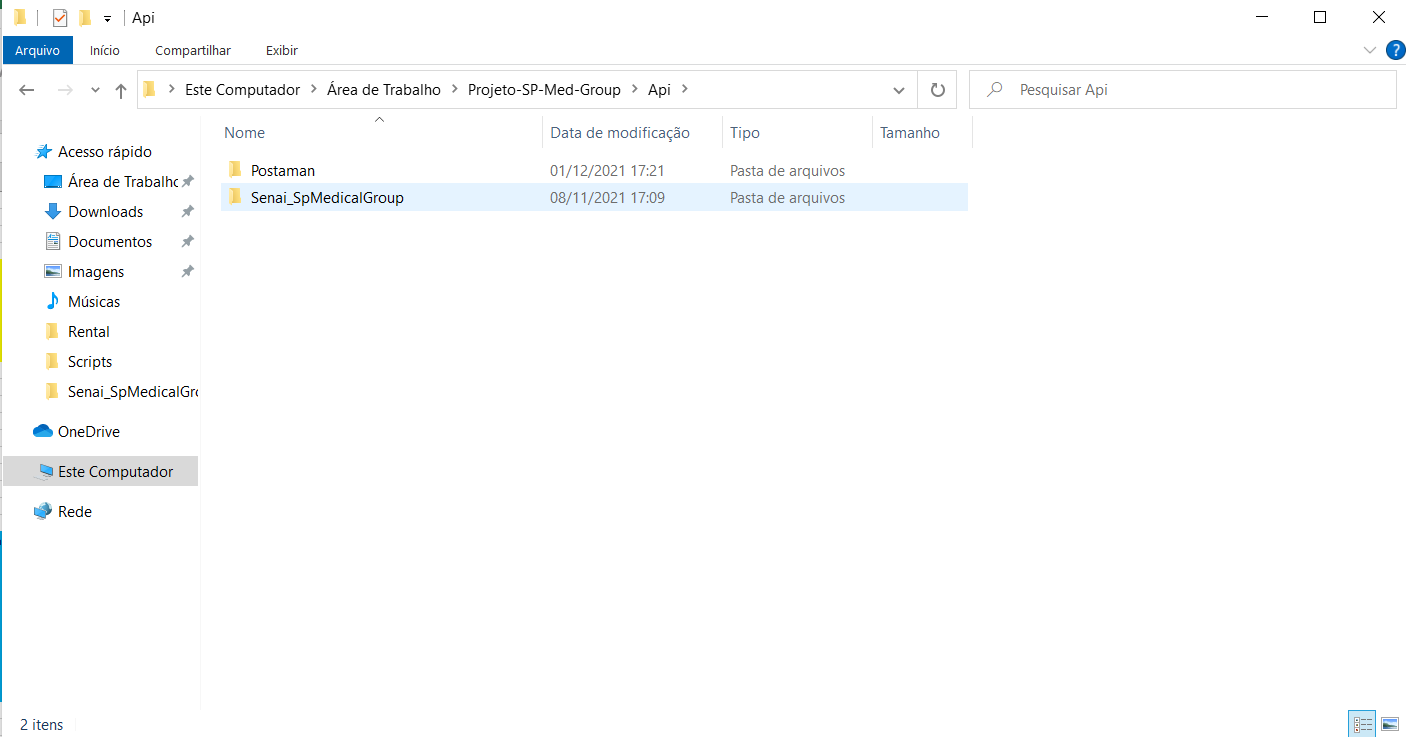
-**JWT** : É um método que realiza uma autenticação e autorização, por meio de um token gerado.

Insira também um passo a passo de como executar a API criada e testar as requisições no Postman.

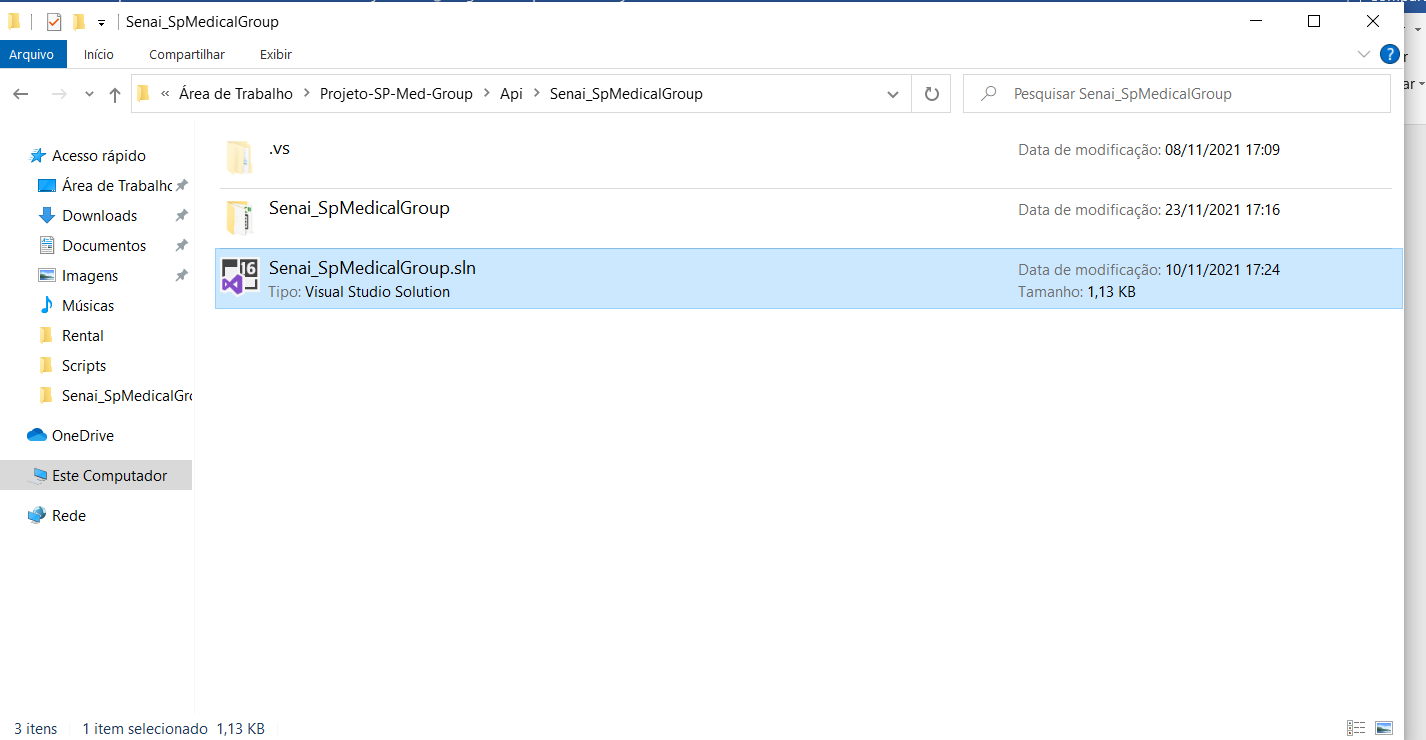


- primeiro clone o repositório.

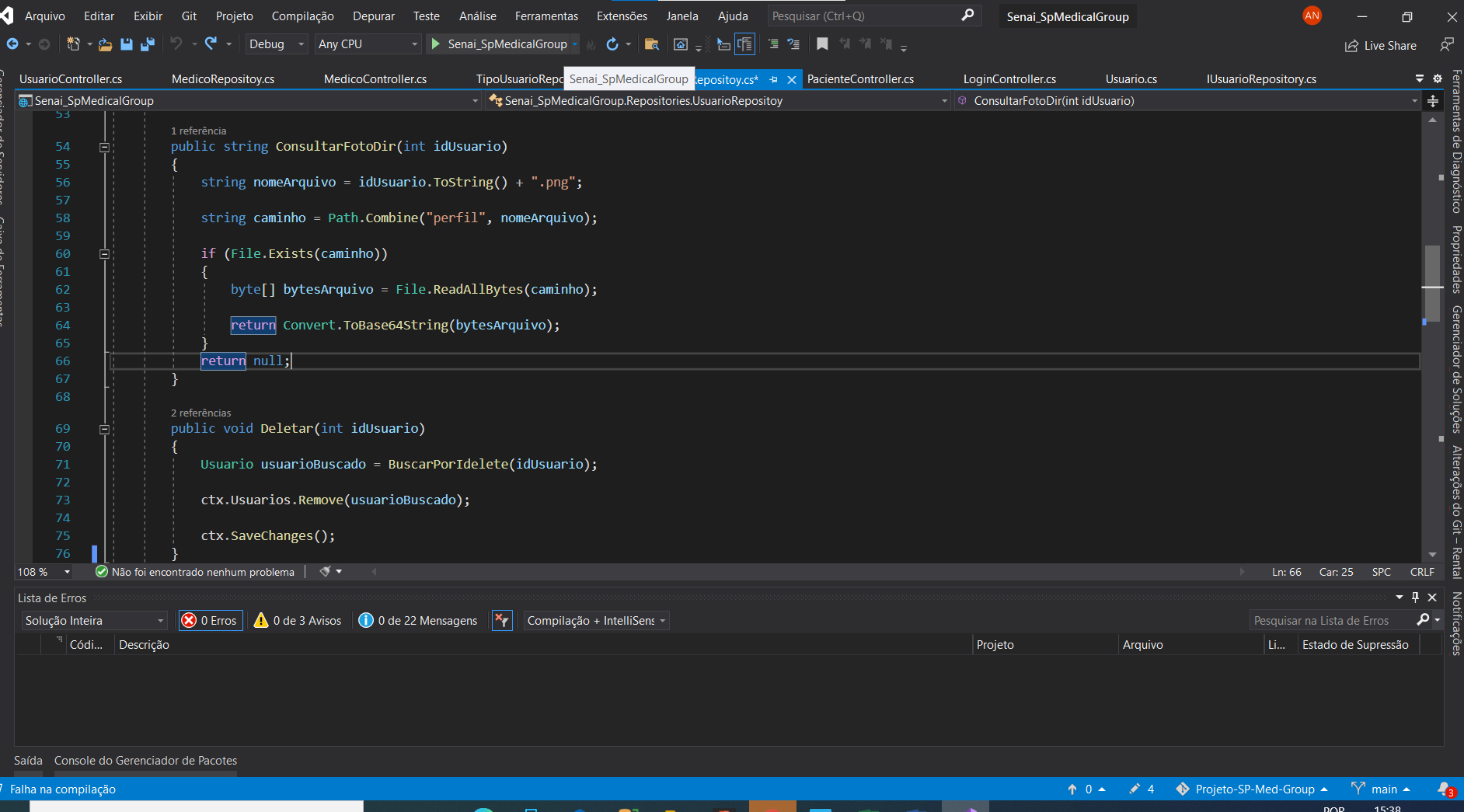
- logo em seguida entre na pasta chamada “Api”



- assim que abrir a pasta “api”, abra a pasta chamado “Senai\_SpMedicalGroup”



- abra o documento do visual studio



Após rodar a API, ela abrira o navegador padrão com a documentação em forma de Swagger.

-Abra o Postman, vá em “Import”,busque o link do Postman através dos seus arquivos, e teste.

## Funcionalidades

1. O **paciente** poderá visualizar suas próprias consultas;

2. O **médico** poderá ver as consultas (os agendamentos) associados a ele;

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;