**NOMEPROJETO**

Documentação

Sumário

[1. Resumo 1](#_gjdgxs)

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

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

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

[Modelo Conceitual 1](#_2et92p0)

[Modelo Lógico 2](#_tyjcwt)

[Modelo Físico 3](#_3dy6vkm)

[Cronograma 4](#_1t3h5sf)

[Trello](#_4d34og8) 5

[4. Back-End 6](https://docs.google.com/document/d/1U0ZO7ctiBpBsnkXKMAebZ9XFEyirneHH/edit#heading=h.4d34og8)

[Funcionalidades 1](https://docs.google.com/document/d/1U0ZO7ctiBpBsnkXKMAebZ9XFEyirneHH/edit#heading=h.2s8eyo1)

[Sistema Web 2](https://docs.google.com/document/d/1U0ZO7ctiBpBsnkXKMAebZ9XFEyirneHH/edit#heading=h.17dp8vu)

[Perfis de usuário: 3](https://docs.google.com/document/d/1U0ZO7ctiBpBsnkXKMAebZ9XFEyirneHH/edit#heading=h.3rdcrjn)

[Funcionalidades: 4](https://docs.google.com/document/d/1U0ZO7ctiBpBsnkXKMAebZ9XFEyirneHH/edit#heading=h.26in1rg)

[Sistema Mobile 5](https://docs.google.com/document/d/1U0ZO7ctiBpBsnkXKMAebZ9XFEyirneHH/edit#heading=h.lnxbz9)

[Perfis de usuário: 6](https://docs.google.com/document/d/1U0ZO7ctiBpBsnkXKMAebZ9XFEyirneHH/edit#heading=h.35nkun2)

[Funcionalidades: 7](https://docs.google.com/document/d/1U0ZO7ctiBpBsnkXKMAebZ9XFEyirneHH/edit#heading=h.1ksv4uv)

# Resumo

Esse documento faz parte dos requisitos do projeto que o cliente pediu, nele irá conter os dados do projeto e algumas explicações.

# Descrição do projeto

O Projeto é uma ideia proposta pelo cliente Fernando Strada, nele temos que fazer um software que possa cadastrar Administradores(Cadastra os outros), Médicos e Pacientes, e que ele seja um sistema para os pacientes marcarem consultas e acompanhar a situação de suas consultas.

# Banco de dados relacional

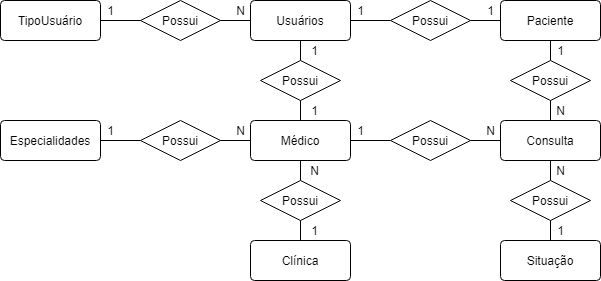
Um banco de dados relacional é aquele cujo a organização e sua estrutura é feita com tabelas relacionadas entre si. E banco de dados é muito importante para softwares, ele é a melhor forma de armazenar dados recebidos dos usuários.

# Modelagem de dados

Modelagem é uma etapa muito importante antes de construir um banco de dados, é nela que organizamos a base de dados que o banco vai receber.

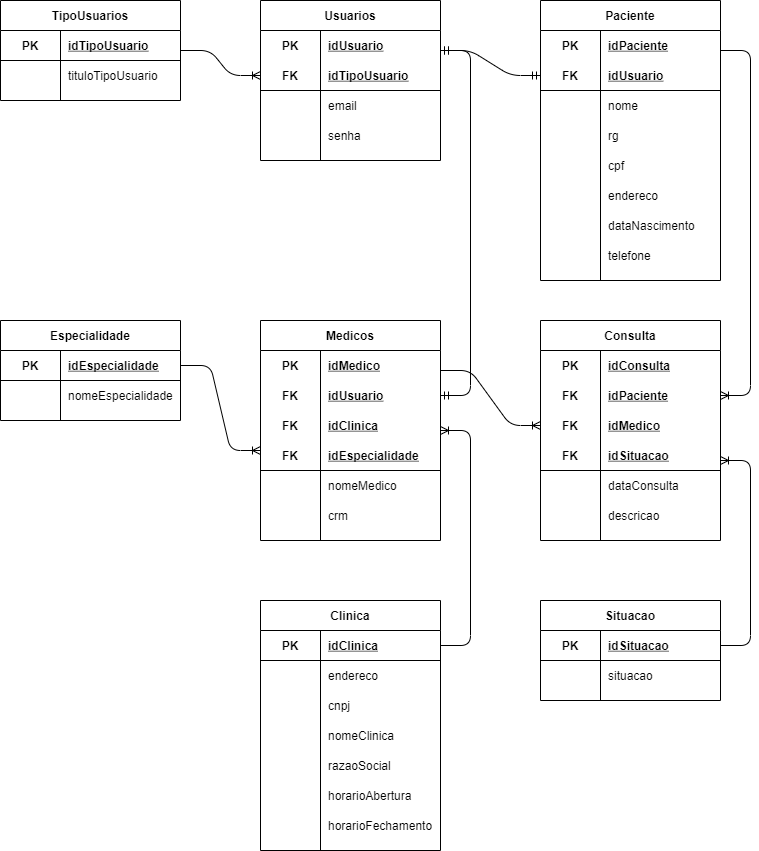
## Modelo Conceitual

Modelo Conceitual é o primeiro modelo a ser feito nas modelagens, é nele que vamos definir as entidades(tabelas) e o tipo de relação estabelecida entre elas.



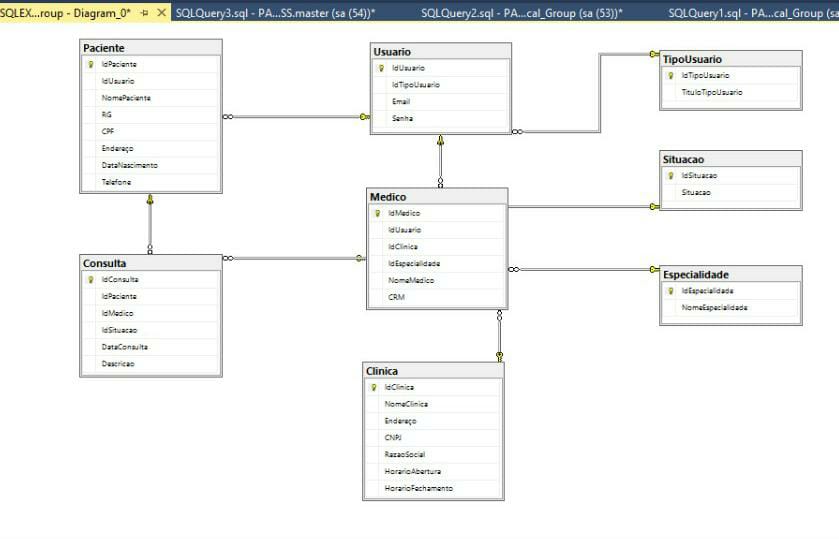
## Modelo Lógico

O Modelo Lógico é onde definimos as chaves primarias e estrangeiras das tabelas e seus campos de maneira mais clara.



## Modelo Físico

O Modelo Físico é o modelo mais esclarecedor entre os três, ele é a base de dados, onde vai ficar os dados dos usuários antes de construir o banco.



## Cronograma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 |
| Modelagens | X |  |  |  |
| Scripts |  | X |  |  |
| Documentação |  |  | X |  |
| Trello |  | X | X | X |
| Backend |  | X | X |  |
| Postman |  |  | X |  |
| Documentação 2.0 |  |  |  | X |

### Trello

<https://trello.com/b/8IPTLPqw/sp-medical-group>

# Back-End

**Arquiteturas Utilizadas:**

A arquitetura utilizada nesse projeto foi a **API,** nela usamos a linguagem **C#** para escrever as linhas de código, foi trabalhado em cima do protocolo HTTP e para ser executada foram utilizados: a linguagem **JSON** para teste de sistema e forma de entrada, saída e consultas no sistema. Além disso, foi utilizada a arquitetura **REST**.

**Significados:**

**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.

**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.

**Como Executar e Testar a API:**

Para executar a API é preciso abrir ela com o programa Visual Studio, depois de abrir a API é só clicar em “Executar” no parte superior da tela, quando esse executar vai abrir um terminal de comando ou cmd, nele vai estar o link ou o caminho para acessar a API, por exemplo: “http://localhost/5000”. Então o próximo passo é utilizar o programa Postman para testar as funcionalidades da API, lá os comandos são feitos em JSON, um exemplo disso é a função “Login” do Método POST , onde você vai inserir os dados de um Usuario do sistema, por exemplo o emai e a senha desse usuário e clica em “SEND” para ver como o sistema processa essa Funcionaidade.

## **Funcionalidades**

Sistema Web:

Perfis de usuário:

1. Administrador: Para o colaborador da área administrativa da clínica;

2. Médico: Colaboradores que atuam na área da saúde;

3. Paciente: Clientes da clínica;

Funcionalidades:

1. O administrador poderá cadastrar qualquer tipo de usuário (administrador,

paciente ou médico);

2. O administrador poderá agendar uma consulta, onde será informado o paciente,

data do agendamento e qual médico irá atender a consulta (o médico possuirá

sua determinada especialidade);

3. O administrador poderá cancelar o agendamento;

4. O administrador deverá informar os dados da clínica (como endereço, horário

de funcionamento, CNPJ, nome fantasia e razão social);

5. O médico poderá ver os agendamentos (consultas) associados a ele;

6. O médico poderá incluir a descrição da consulta que estará vinculada ao paciente

(prontuário);

7. O paciente poderá visualizar suas próprias consultas;