**SP MEDICAL GROUP**

Documentação

Sumário

[1. Resumo 3](#_Toc80486665)

[2. Descrição do projeto 3](#_Toc80486666)

[3. Banco de dados relacional 3](#_Toc80486667)

[4. Modelagem de dados 3](#_Toc80486668)

[Modelo Conceitual 3](#_Toc80486669)

[Modelo Lógico 4](#_Toc80486670)

[Modelo Físico 5](#_Toc80486671)

[Cronograma 5](#_Toc80486672)

[Trello 5](#_Toc80486673)

5. Back-End 6

Arquiteturas Utilizadas 6

Significados 6

Como Executar e Testar a API 6

Funcionalidades 7

Sistema Web: 7

# Resumo

Este documento relata a modelagem de um banco de dados para a atividadeSP MEDICAL GROUP, proposta pelos professores da escola Senai Informática. Tal atividade visa colocar em prática habilidades de modelagem de banco de dados adquiridas durante aulas.

# Descrição do projeto

Uma nova clínica médica chamada SP Medical Group, atua no ramo da saúde, empresa de pequeno porte que 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 se tornou complicada devido à alta demanda dos pacientes.

# Banco de dados relacional

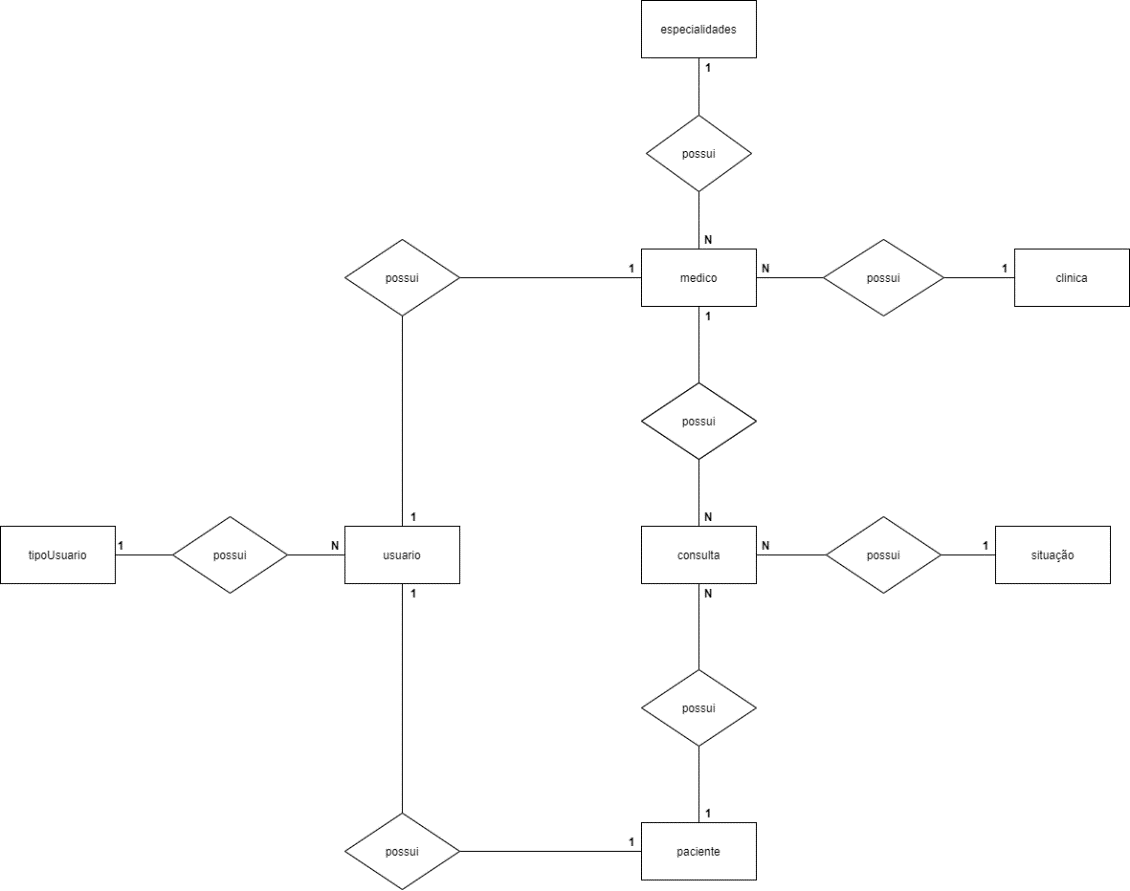
Um banco de dados relacional é um banco de dados que classifica os dados para que sejam tratados como tabelas ou relações. Isso é importante porque nos permite classificar os dados de maneira mais organizada.

# Modelagem de dados

A modelagem de dados é a primeira etapa na construção de um banco de dados, no qual as entidades que o banco de dados terá, os campos das entidades e os relacionamentos entre eles são definidos. Isso, por sua vez, torna mais fácil construir um banco de dados, porque já conhecemos a estrutura básica inicial do banco de dados. Existem 3 tipos de modelagem: conceitual, lógica e física.

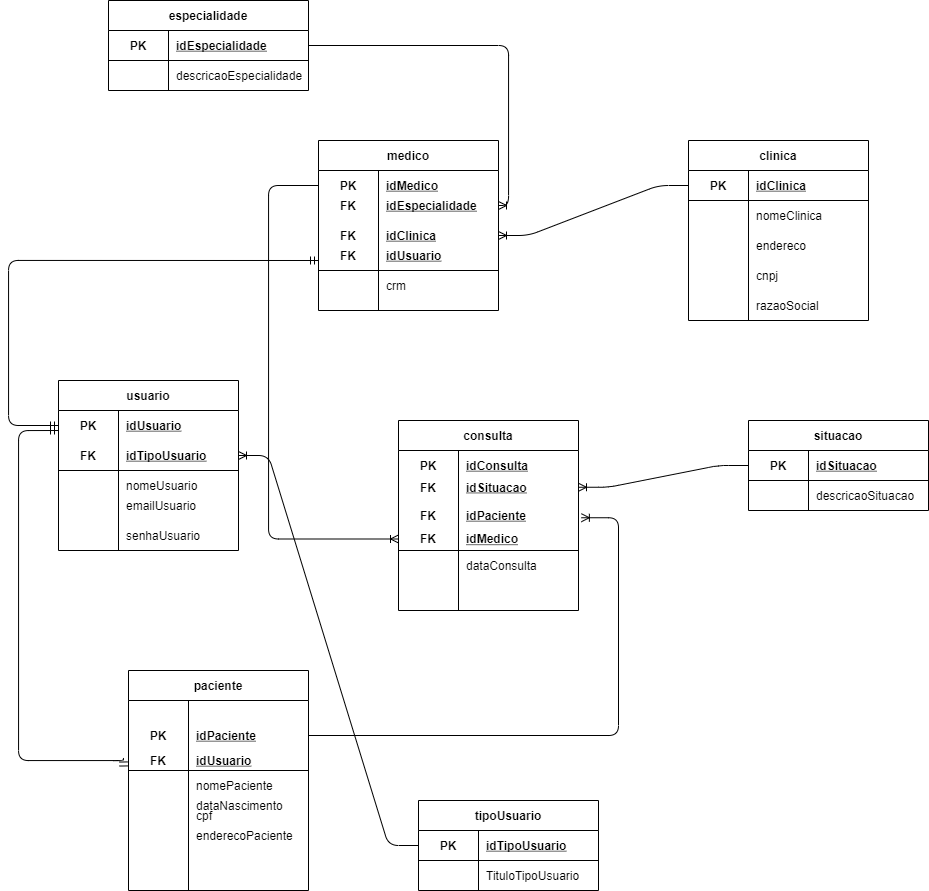
## Modelo Conceitual

Na modelagem conceitual definiu-se as entidades, suas relações e a cardinalidade entre elas. Essa modelagem ajuda a compreender as relações entre as entidades e auxilia na construção da modelagem lógica.



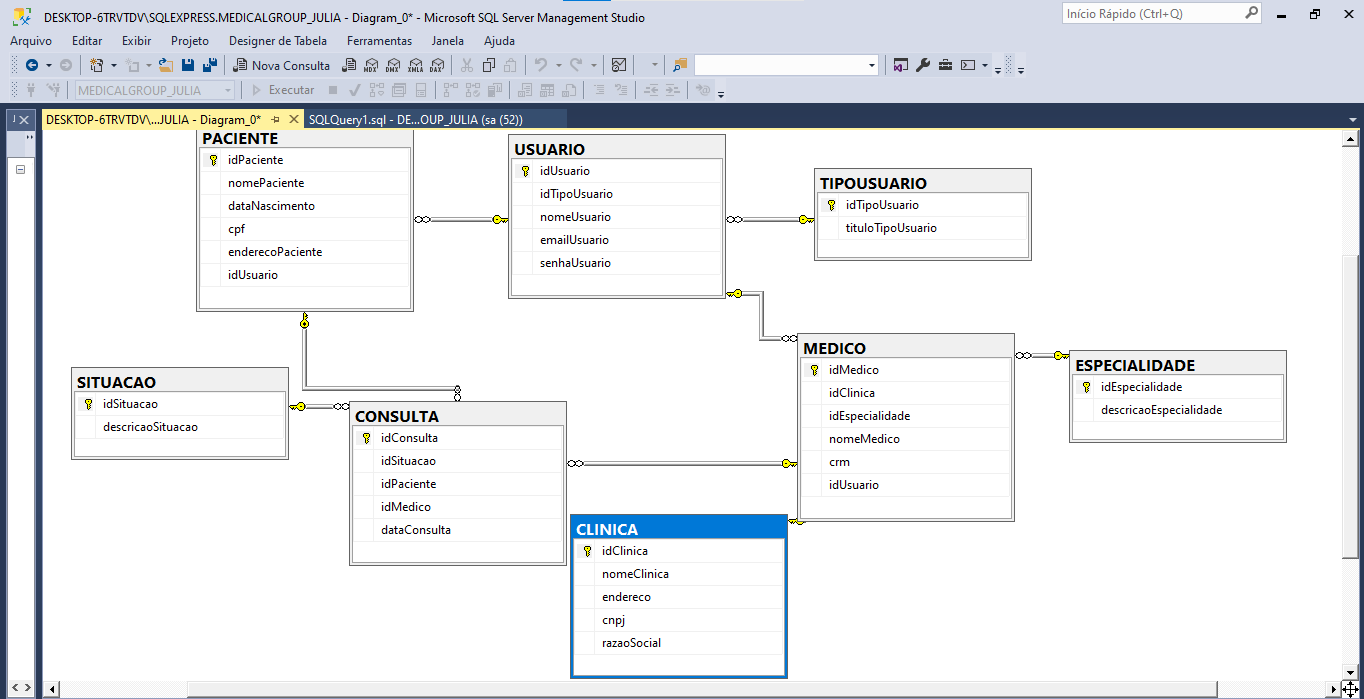
## Modelo Lógico

Na modelagem lógica, as entidades passaram a ser representadas por pequenas tabelas. Esse modelo mostra as entidades de forma mais complexa já contendo seus campos e também apresenta cardinalidades. Esse tipo de modelagem ajuda a construir modelagem física e de banco de dados.



## Modelo Físico

Esse modelo representa de forma visual o banco de dados contendo as entidades, os campos e os dados armazenados nesses campos.



## 

## Cronograma

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 | Dia 5 | Dia 6 | Dia 7 |
| Modelagens | X | X |  |  |  |  |  |
| Scripts |  | X | X |  |  |  |  |
| Trello | X |  |  | X |  |  |  |
| Backend |  |  |  |  | X | X | X |
| Postman |  |  |  |  |  |  | X |

### Trello

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

# Back-End

**Arquiteturas Utilizadas:**

Neste projeto foi utilizada a **API,** sendo usada 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, com algum projeto feito, depois de abrir a API é só clicar em “Executar” na 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 email e a senha desse usuário e clica em “SEND” para ver como o sistema processa essa Funcionalidade.

## **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;