**SP MEDICAL GROUP**

GUSTAVO REZENDE PAZ

Documentação

Sumário

[1. Resumo 3](#_Toc80562521)

[2. Descrição do projeto 3](#_Toc80562522)

[3. Banco de dados relacional 3](#_Toc80562523)

[4. Modelagem de dados 3](#_Toc80562524)

[Modelo Conceitual ...................................................4](#_Toc80562525)

[Modelo Lógico 5](#_Toc80562526)

[Modelo Físico 6](#_Toc80562527)

[Cronograma 7](#_Toc80562528)

[Trello 7](#_Toc80562529)

Backend..................................................................................................................................................................8

Funcionalidades......................................................................................................................................................9

Frontend...............................................................................................................................................................10

# Resumo

A documentação tem como fundamento apresentar detalhadamente o desenvolvimento e organização do banco de dados da clinica médica ‘’SP Medical Group’’, destacando as etapas elaboradas do início até o término do projeto, o cronograma das atividades, o modo como foi produzido e o material com informações do sistema.

# Descrição do projeto

O cliente Fernando Strada solicitou ao desenvolvedor um sistema para realizar a gestão da clínica de forma dinâmica e precisa para fácil acesso aos dados sobre as informações de seus pacientes. Para isso, foi realizado modelagens, scripts e uma organização. Com tabelas, funções e parâmetros para facilitar a visualização dos dados. O documento com as tabelas e descrição do que foi produzido pode-se encontrar no git hub do desenvolvedor Gustavo Rezende.

# Banco de dados relacional

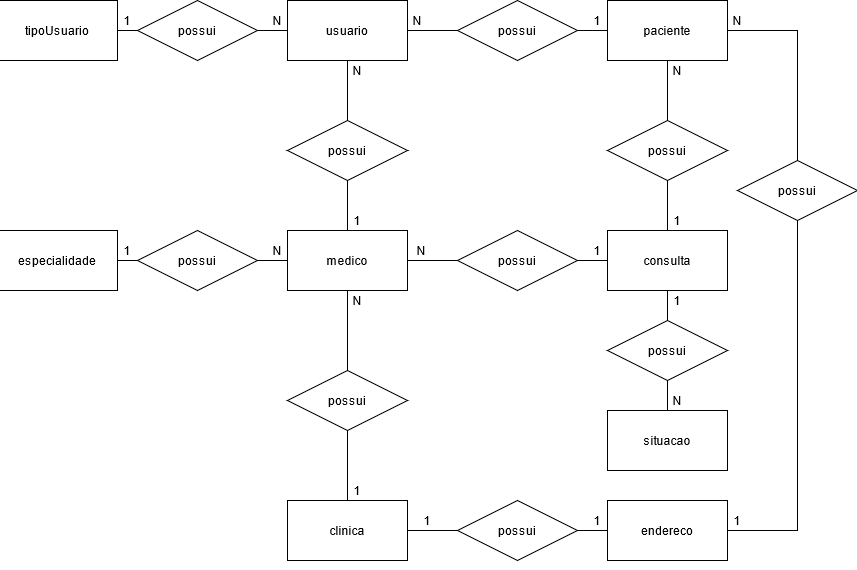
Bancos de dados são importantes para organizarmos e armazenarmos informações de maneira organizada e centralizadas, assim facilitando o acesso aos dados e tornando possível adquirir informações através de relacionamentos entre entidades/tabelas, quando houver.

Nos casos em que estes relacionamentos existem, temos bancos de dados relacionais. Nestes a modelagem é feita de uma forma que estes tenham tabelas interligadas entre si, cada registro tem seus campos e é acompanhado por uma chave primária, que pode ser usada como estrangeira em outra entidade, formando a dita relação entre tabelas, que se bem feita, além de evitar anomalias, deixa o banco sem redundância e menos lento.

# Modelagem de dados

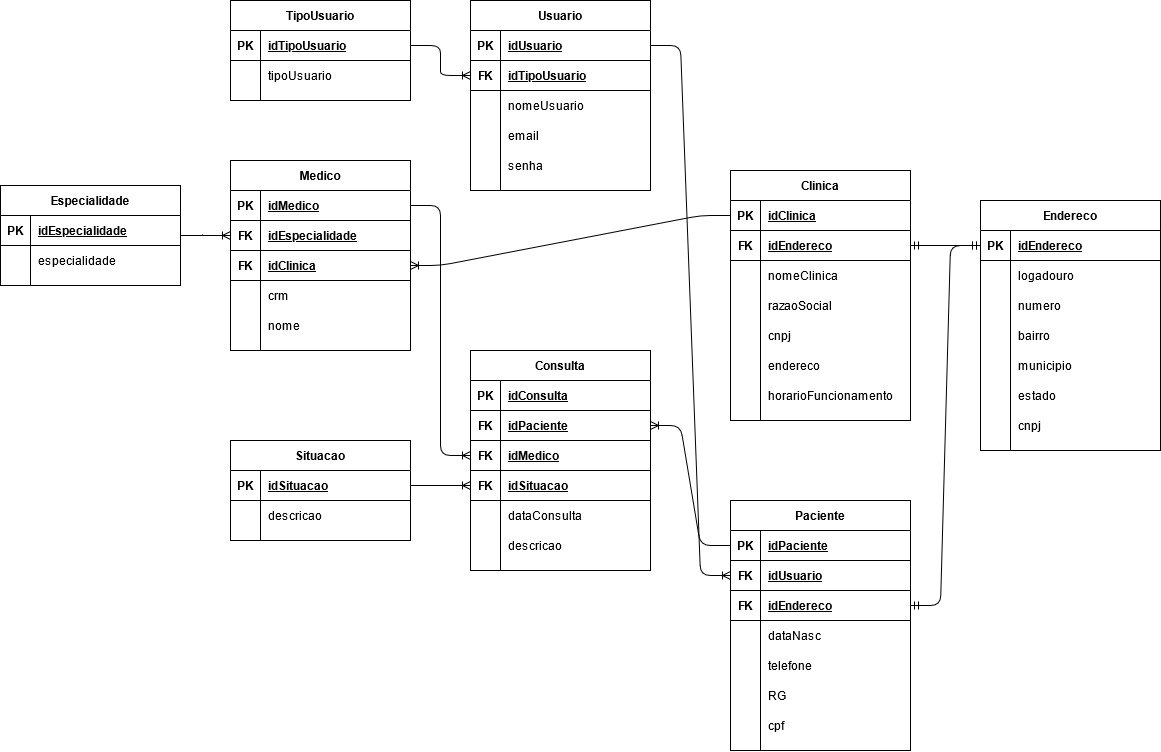
A modelagem de dados é uma etapa de extrema importância para projetos de desenvolvimento de softwares, em especial na área de bancos de dados. Nela temos tipos de relações (Cardinalidade), as entidades presentes e como o próprio nome sugere, um modelo do banco, seja esse qualquer um dos três possíveis, conceitual, lógico e físico.

## Modelo Conceitual



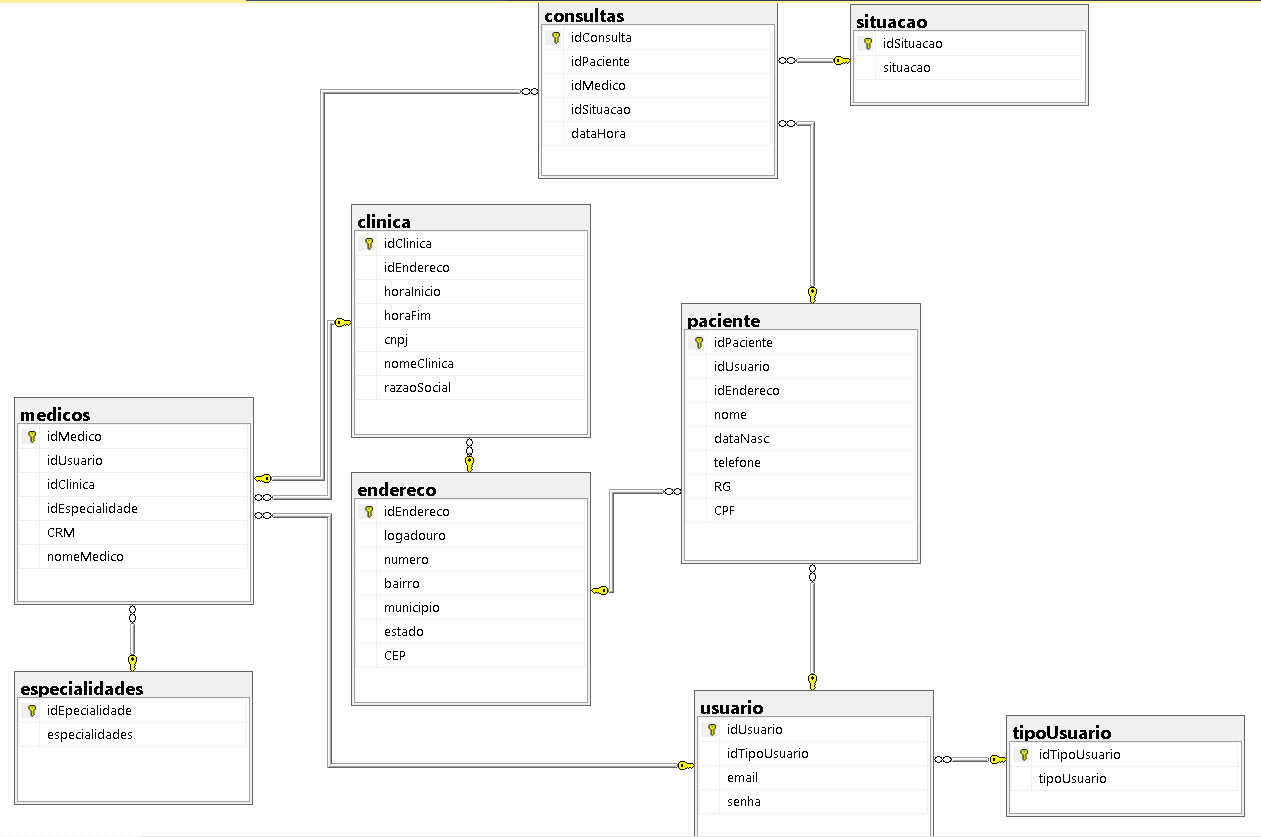
Nesse modelo conceitual do sistema, é possível traçar uma primeira organização mais simples do projeto, trazendo as classes e as respectivas cardinalidades que haverá entre elas. De acordo com o que foi solicitado foi introduzido Usuário, tipo de usuário, Paciente, Médicos, consultas, especialidades, situação da consulta, clínica e dados da clínica como endereço.

## Modelo Lógico



A modelagem lógica é um protótipo mais aprofundado e mais sólido do modelo conceitual, trazendo uma visão mais ampla das necessidades do sistema da SP Medical Group, envolvendo as cardinalidades de forma mais dinâmica, as relações entre as chaves primárias e estrangeiras, além dos atributos de cada entidade.

## Modelo Físico



A modelagem física pode ser entendida como uma esquematização mais específica do escopo, já apresentando os dados em modelo de tabela (com as linhas e colunas - registros e campos). Assim como no Excel que simula o banco de dados em uma planilha.

## Cronograma

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dias | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| Organização Trello | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Modelo Conceitual | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Modelo Lógico | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Modelo Físico | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SCRIPT DDL | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SCRIPT DML |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SCRIPT DQL |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentação |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |
| Desenvolvimento da API |  |  |  |  |  | X | X | X | X |  |  |  |  |  |  |  |  |
| Testes (postman) |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| Design dos layouts |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |
| Aplicação web frontend |  |  |  |  |  |  |  |  |  |  | X | X | X |  |  |  |  |
| Aplicação Mobile |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X |

### Trello

https://trello.com/b/351AYZMM/2dmmed-group

4. Back-End

O projeto SpMedicalGroup foi desenvolvido utilizando o SQL Server para a parte do banco de dados e o Microsoft Visual Studio para parte da API, em linguagem C#. A Application Programming Interface (API) é uma forma de integrar sistemas definindo as requisições e respostas obtendo muita vantagem, por facilitar o desenvolvimento com mais agilidade precisão e principalmente organização, seguindo protocolo HTTP.

Para a inserção e a troca de respostas é possível utilizar um formato em JSON (JavaScript Object Notation) para que o sistema possa ser acessado de várias maneiras Dispositivos diferentes sem se preocupar com a linguagem que usarão.

E para haver uma praticidade na organização a API foi feita utilizando o estilo de arquitetura REST, sigla em inglês para Representational State Transfer, ou, traduzindo, Interface de Programação de Aplicativos.

Como executar?

Para executar a API e conseguir utilizar basta entrar no link <http://senai-spmedgroup-webapi.azurewebsites.net> , uma outra opção é utilizar aplicativos para realizar as requisições como o

Postman, escolha qual tipo de requisição que deseja realizar, ou seja GET, POST, HEAD, OPTIONS, PUT, DELETE e TRACE. Copiando o caminho e passando {link/api/campo}, por exemplo:

<http://senai-spmedgroup-webapi.azurewebsites.net/api/consultas/minhasMed> 

Clonando o repositório <https://github.com/gustrpaz?tab=repositories> , seguindo o caminho:

\2021-2S-2M\2S-Projetos\SP-Medical-Group\Backend, abra a solução [Senai\_SPMedGroup\_webAPI.sln](https://github.com/gustrpaz/2021-2S-2M/blob/main/2S-Projetos/SP-Medical-Group/Backend/Senai_SPMedGroup_webAPI/Senai_SPMedGroup_webAPI.sln) . Utilizando o Microsoft Visual Code. Dentro do software, execute a aplicação e faça requisições pelo domínio <http://localhost:5000>



Funcionalidades

O projeto (API) possui diversas funcionalidades que o usuário pode utilizar, como por exemplo:

Funções do CRUD (Criar, Ler, Atualizar e Deletar)

**Sistema Web**

**Perfis de usuário**:

1. Administrador: Área administrativa das clínicas;

2. Paciente: Usuário do paciente das clínicas;

3. Médico: Usuário do médico das clínicas.

**Funcionalidades**:

1. O administrador poderá cadastrar qualquer tipo de usuário;

2. O administrador poderá cadastrar os dados da clínica;

3. O administrador poderá cadastrar os médicos de cada clínica;

4. O administrador poderá agendar consultas;

5. O administrador poderá listar todas as consultas;

6. O administrador poderá cancelar as consultas;

7. O administrador poderá remover uma consulta do sistema;

8. O administrador poderá remover uma clínica;

9. O administrador poderá atualizar os dados de uma clínica

10. O administrador poderá listar todas as clínicas;

11. O administrador poderá listar os médicos de cada clínica;

12. O administrador poderá listar os pacientes de cada clínica

13. O administrador poderá cadastrar os pacientes de cada clínica;

14. O administrador poderá listar os pacientes de cada clínica;

15. O administrador poderá listar um paciente pelo Id;

16. O administrador poderá cadastrar uma imagem para um paciente;

17. O médico e o paciente poderão listar as consultas onde estão envolvidos

# Sistema Web

PASSO A PASSO DE COMO EXECUTAR O FRONT END COM REACT (OBS: TENHA A API E BANCO RODANDO NO SEU SISTEMA)

1. Já tendo o repositório clonado, scripts do banco e api rodando, siga o caminho:  
SP\_MedicalGroup -> SP\_MedicalGroup -> FrontEnd -> spmedUi

vá ao terminal e

digite o comando “npm i” para fazer a instalação de todas as dependências usadas no aplicativo, sejam outros pacotes ou arquivos necessários para que a aplicação funcione de forma correta.

2. Em seguida, quando a instalação for finalizada digite npm start para rodar os scripts

3. Pronto, agora você pode começar a navegar pelo site

**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 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;

7. Sistema Mobile

PASSO A PASSO DE COMO EXECUTAR O MOBILE COM REACT NATIVE (OBS: TENHA A API E BANCO RODANDO NO SEU SISTEMA)

1. Troque a url da aplicação para seu Ip em vez de localhost:5000, para isso vá ao cmd da sua máquina e digite ipconfig e selecione o endereço ipv4
2. Coloque seu ip no applicationURL e comente o do localhost

3. Rode a API e siga o caminho: SP-Medical-Group -> Banco -> Mobile->

SpMedMobile, abra o cmd e digite npm i:

4. Agora digite npx react-native run-android e use o aplicativo.

1. Banco não relacional

Foi criado um banco não relacional utilizando o MongoDB para que seja feita a Data analytics, o armazenamento das localizações dos pacientes, foi utilizado a API do google para o auxilio dessa função, na qual é exibido um mapa com os marcadores nos respectivos endereços.

Diferença do banco relacional para o banco não relacional:

**SQL NoSQL**

Database Database

Tables Collections

Rows Documents (BSON) = JSON porém em binário

Columns Fields

Comandos utilizados no MongoDB:

Use SPMEDGROUP

Db. createCollection(“mapas”)

Instalação de um pacote no backend “MongoDB.Driver”

Alterações feitas na API:

Startup:

options.AddPolicy("CorsPolicy",

builder =>

{

builder.WithOrigins("\*")

.AllowAnyHeader()

.AllowAnyMethod();

});