**SP MEDICAL GROUP**

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)

# Resumo

Essa 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

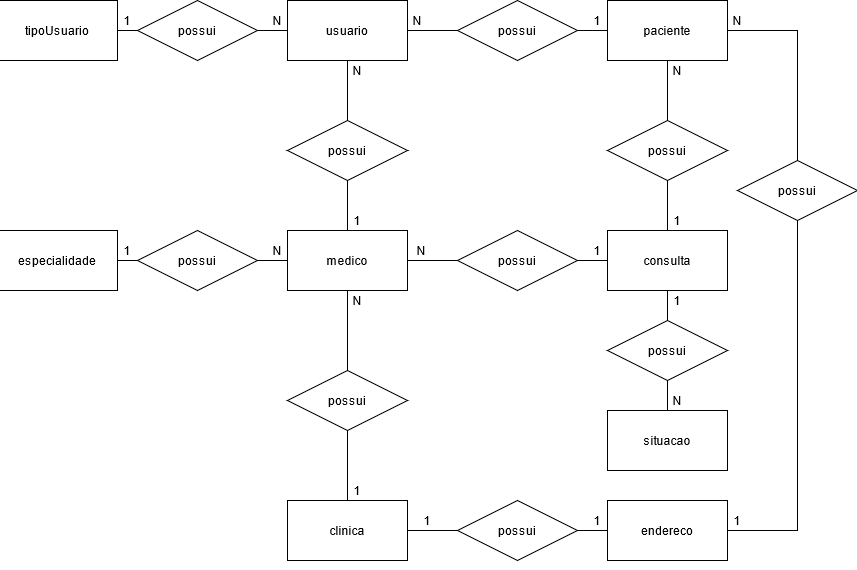
Bancos de dados são importantes para organizarmos e armazenarmos informações de maneira organizada e centralizadas, assim facilitando o acesso aos dados e tornando possível adquirir informações através de relacionamentos entre entidades/tabelas, quando houver.

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

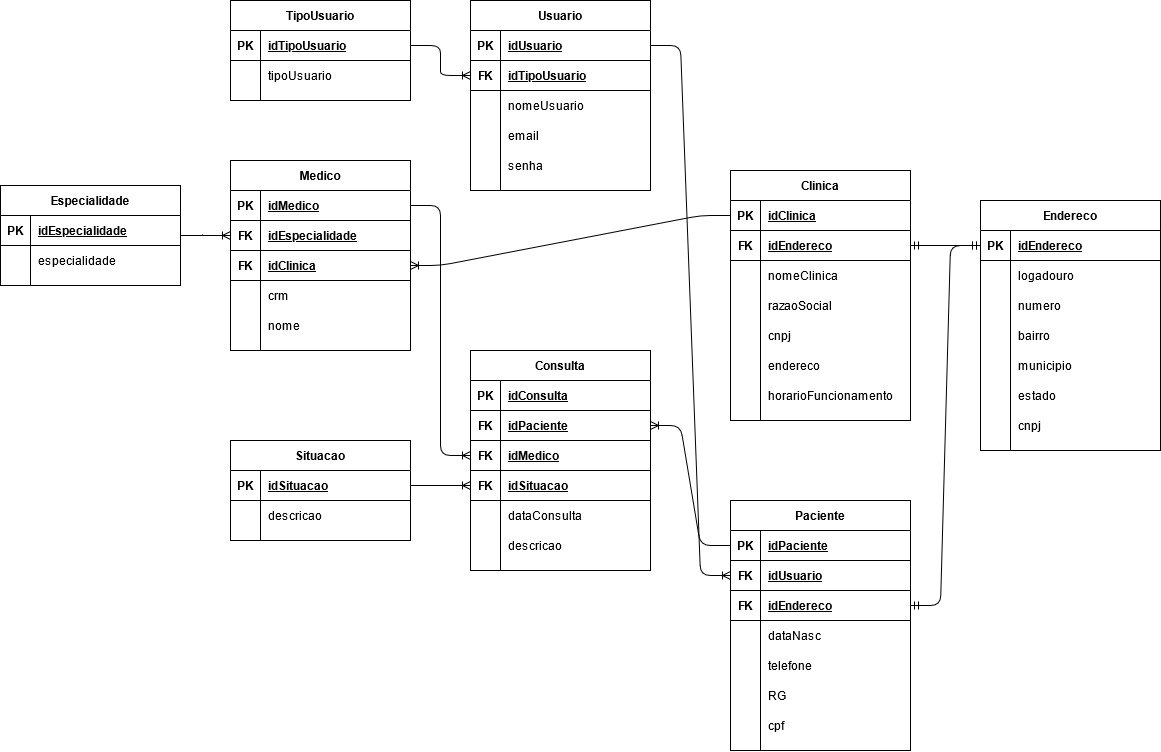
A modelagem de dados é uma etapa de extrema importância para projetos de desenvolvimento de softwares, em especial na área de bancos de dados. Nela temos tipos de relações (Cardinalidade), as entidades presentes e como o próprio nome sugere, um modelo do banco, seja esse qualquer um dos três possíveis,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

## Modelo Conceitual



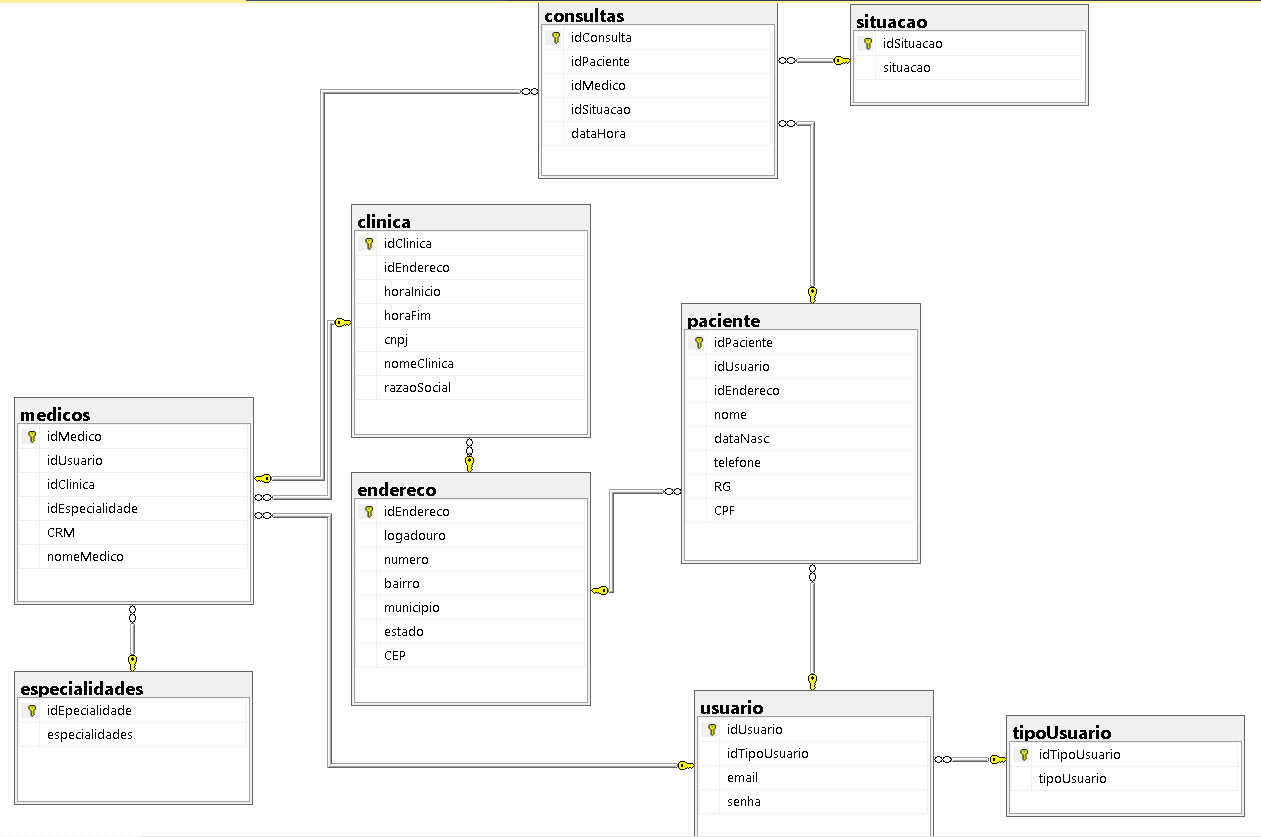
Nesse modelo conceitual do sistema, é possível traçar uma primeira organização mais simples do projeto, trazendo as classes e as respectivas cardinalidades que haverá entre elas. De acordo com o que foi solicitado foi introduzido Usuário, tipo de usuário, Paciente, Médicos, consultas, especialidades,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

|  |  |  |  |
| --- | --- | --- | --- |
|  | Dia 1 | Dia 2 | Dia 3 |
| Organização Trello | X |  |  |
| Modelo Conceitual | X |  |  |
| Modelo Lógico | X |  |  |
| Modelo Físico | X |  |  |
| SCRIPT DDL | X |  |  |
| SCRIPT DML |  | X |  |
| SCRIPT DQL |  |  | X |
| Documentação |  |  | X |

### Trello

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

4. 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.

Insira também um passo a passo de como executar a API criada e testar as requisições

no Postman.

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;

A API desenvolvida para o projeto SP Medical Group foi desenvolvida na linguagem de programação C#, utilizando o software Microsoft Visual Studio. API é um conjunto de padrões pré-estabelecidos para a utilização do software, fazendo requisições e recebendo respostas seguindo protocolo HTTP.

HTTP é uma sigla em inglês, para Hypertext Transfer Protocol, traduzindo, Protocolo de Transferência de Hipertexto. A API recebe e envia informações em formato JSON, JSON significa JavaScript Object Notation, traduzindo, Notação de Objetos JavaScript.

Finalizando, a API foi feita utilizando o estilo de arquitetura REST, sigla em inglês para Representational State Transfer, ou, traduzindo, Interface de Programação de Aplicativos.

Para utilizar a API hospedada, acesse o site [https://senai-spmedicalgroup-webapi-murillo.azurewebsites.net](https://senai-spmedicalgroup-webapi-murillo.azurewebsites.net/index.html), ou utilize um software terceiro, com esse domínio, para fazer requisições.

Para utilizar a API de forma local, clone o repositório <https://github.com/MurilloAssis/2021-2S-2D>, acessando o endereço de pastas \sprint2-api\Projetos\SP\_MEDICAL\_GROUP\BackEnd, abra a solução senai\_spmedicalgroup\_webapi.sln. Para executar é necessário o software Microsoft Visual Code. Dentro do software, execute a aplicação e faça requisições pelo domínio <http://localhost:5000>.

A API conta com diversas funcionalidades que o usuário pode utilizar, como por exemplo:

**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;