**SP MEDICAL GROUP**

Documentação

Sumário

[1. Resumo 3](#_gjdgxs)

[2. Descrição do projeto 3](#_30j0zll)

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

[4. Modelagem de dados 3](#_3znysh7)

[Modelo Conceitual 3](#_2et92p0)

[Modelo Lógico 3](#_tyjcwt)

[Modelo Físico 3](#_3dy6vkm)

[Cronograma 3](#_1t3h5sf)

[Trello 3](#_4d34og8)

# Resumo

Documento feito para o armazenamento das modelagens e um resumo da descrição do projeto

# 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.

# Banco de dados relacional

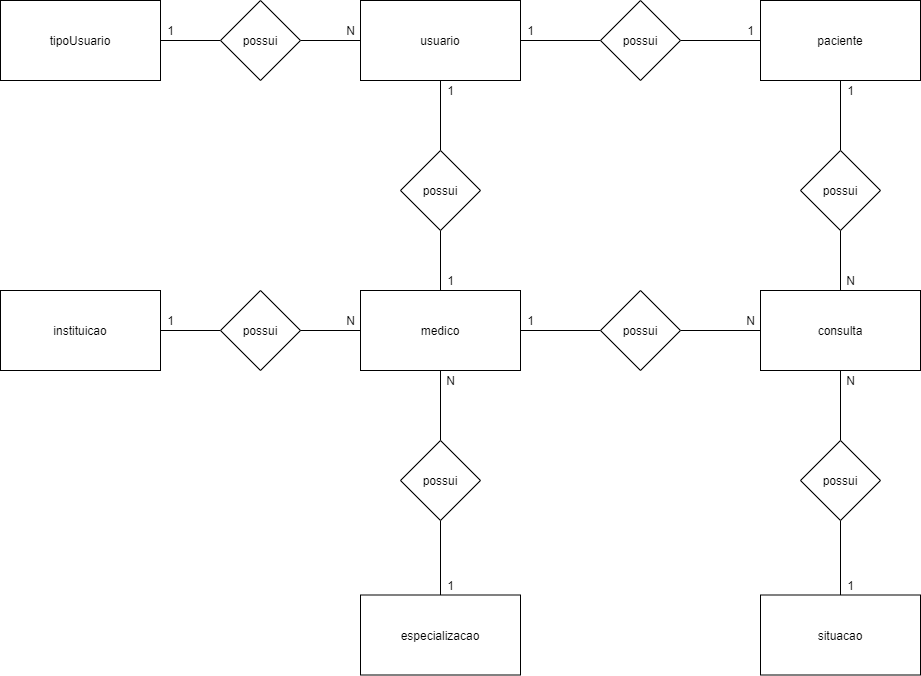
Banco de dados relacional é aquele que ordena os dados para que eles sejam vistos como tabelas ou relações. Ele é importante pois nos permite ordenar nossos dados de forma mais intendivel

# Modelagem de dados

Modelagem de dados é o método de criar um modelo do sistema para explicar suas características e comportamento para facilitar o entendimento do projeto, para evitar erros em seu desenvolvimento.

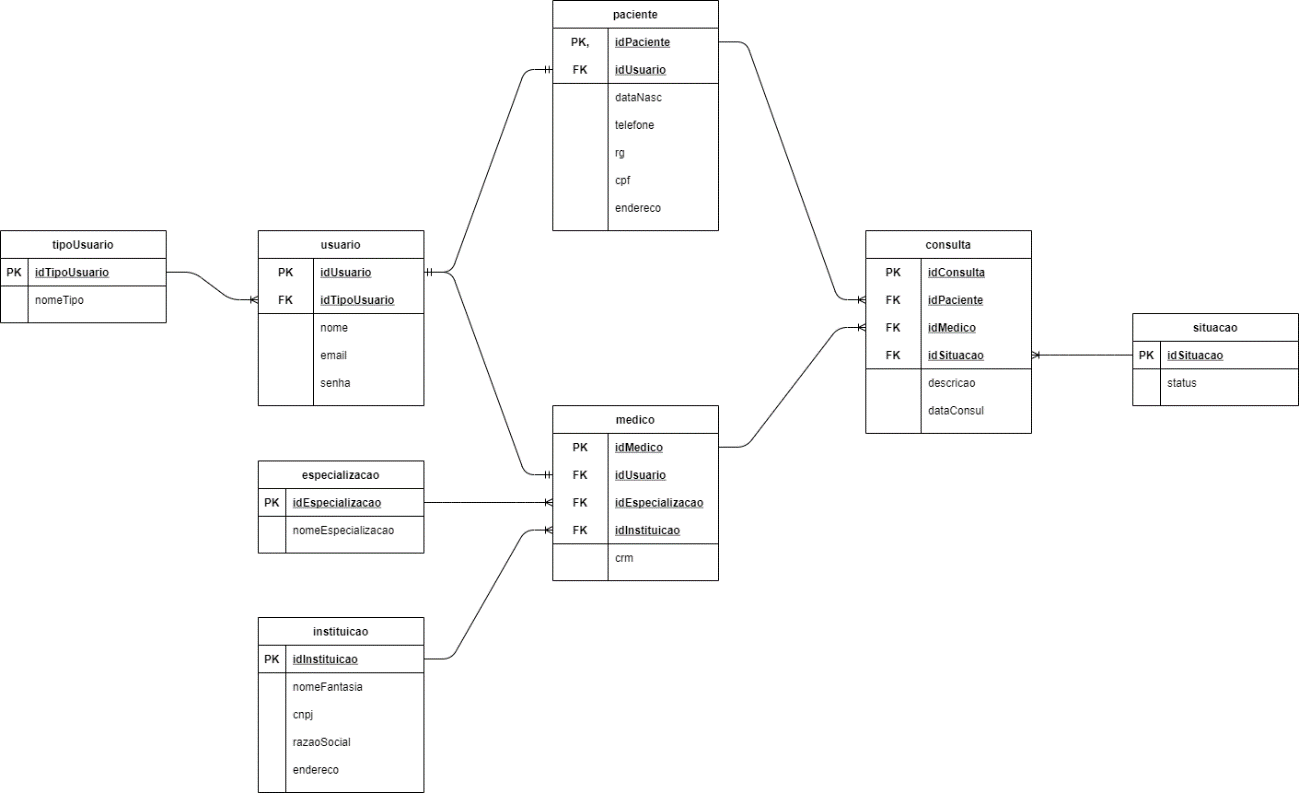
## Modelo Conceitual

Esse modelo mostra as entidades de forma mais simples e estabelece as relações de cardinalidade.



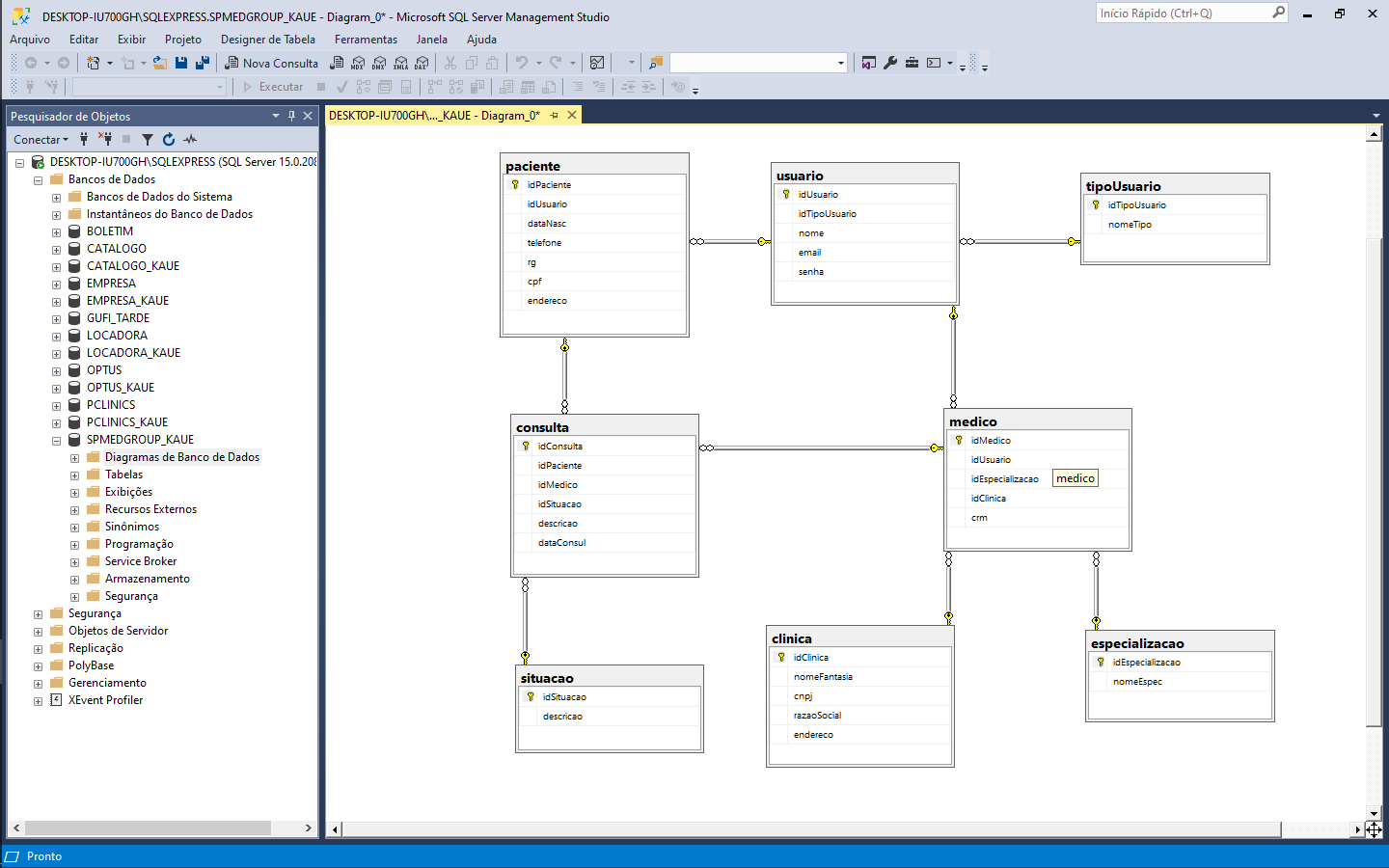
## Modelo Lógico

Esse modelo mostra as entidades de forma mais complexa já contendo seus campos e também apresenta cardinalidade.



## 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 |
| Modelo Conceitual | X |  |  |  |  |
| Modelo Lógico |  | X | X |  |  |
| Modelo Físico |  |  |  | X | X |

### Trello

<https://trello.com/b/Y8MEzgB4/spmedgroup>

# **5 . Back-End**

Para este projeto optamos por desenvolver a nossa aplicação no formato de uma API, ela foi desenvolvida

Por exemplo:

O código do sistema foi desenvolvido no formato de 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** é um conjunto de tecnologias no ADO.NET que dão suporte ao desenvolvimento de aplicativos de software orientado a dados. O Entity Framework permite que os desenvolvedores trabalhem com dados na forma de objetos e propriedades específicos de domínio, como clientes e endereços de clientes, sem ter que se preocupar com as tabelas e colunas de banco de dados subjacentes em que esses data são armazenados.

**Swagger** é uma ferramenta usada para documentar os endpoints da nossa API

**Data Base First** é o método de construção da API onde se usa as tabelas preexististentes no banco de dados, e os transforma em classes dentro da API

**JWT**  é o método de autenticação usado, onde a autenticação é por meio de tokens

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

* **Clone o repositório**
* **Abra a coleção do postman**
* **Faça a requisição**

# **6 . Web**

A parte web do projeto resolvemos desenvolver em JavaScript mais especificamente usando a biblioteca ReactJs.

Foram desenvolvidas 4 telas:

* Login: Página que gera um token e redireciona para as outras páginas baseado no token gerado
* Administrador: Página onde os administradores podem cadastrar e listar todas as consultas
* Médico: Página onde os médicos podem listar suas próprias consultas e adicionar um prontuário as suas consultas
* Paciente: Página onde os pacientes podem ver suas próprias consultas