**Projeto SP Med. Group**

Documentação

Sumário

[1. Resumo 3](#_Toc71092221)

[2. Descrição do projeto 3](#_Toc71092222)

[3. Modelagem de Dados 3](#_Toc71092223)

[Modelo Conceitual 3](#_Toc71092224)

[Modelo Lógico 3](#_Toc71092225)

[Modelo Físico 3](#_Toc71092226)

[Cronograma 3](#_Toc71092227)

[Trello 4](#_Toc71092228)

[4. Back-End 4](#_Toc71092229)

[Funcionalidades 5](#_Toc71092230)

[Sistema Web 5](#_Toc71092231)

[Perfis de usuário: 5](#_Toc71092232)

[Funcionalidades: 5](#_Toc71092233)

[Sistema Mobile 5](#_Toc71092234)

[Perfis de usuário: 5](#_Toc71092235)

[Funcionalidades: 5](#_Toc71092236)

# Resumo

Neste documento mostrarei brevemente a documentação do Projeto Medicals, seu conceito e banco de dados e API.

# Descrição do projeto

O projeto foi realizado com base em uma solicitação de um cliente, onde ele pediu um sistema completo para a clinica. Onde foi feita as modelagens que são o modelo conceitual, modelo lógico e modelo físico (documento Excel), foi feito um banco de dados e a API

# Modelagem de Dados

A modelagem de dados refere–se à criação de um modelo que explica as características e o comportamento de software criando com base nas principais características que evitam erros de programação, projeto e operação, assim auxiliando no entendimento do projeto.

## Modelo Conceitual

## C:\Users\Maria\Desktop\senai_sprint_1bd\sprint-1-bd\projeto_medicals\modelagens\projeto_medicals-conseitual.png

Modelagem conceitual ajuda a destacar conexões importantes em processos e sistemas no mundo real. Ele nos dá uma base para o modelo lógico.

## Modelo Lógico

## C:\Users\Maria\Desktop\senai_sprint_1bd\sprint-1-bd\projeto_medicals\modelagens\projeto_medicals-logico.png

O modelo lógico se adéqua ao modelo conceitual para o tipo de banco de dados que será implementado. Essa fase é necessária para adicionar os atributos nas entidades corretas.

## Modelo Físico

## 

## Este modelo é o projeto físico para implementação de banco de dados.

## Cronograma

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 | Dia 5 |
| Modelo Conceitual | X |  |  |  |  |
| Modelo Lógico | X |  |  |  |  |
| Modelo Físico | X |  |  |  |  |
| Banco De  Dados |  | X | X |  |  |
| API |  |  |  | X | X |

### Trello

https://trello.com/invite/b/bZmmyH5h/9458bf5ac4157f968d792094731e6926/projeto-medicals

# Back-End

Insira aqui uma descrição sobre a arquitetura utilizada para a criação do back-end.

Por exemplo:

O código desse projeto foi desenvolvido através de uma API utilizando o Microsoft Visual Studio. E essa API foi desenvolvida no formato HTTP, no modelo JSON e o no 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

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;

Persistência de dados

Deve ser realizada no banco de dados criado anteriormente na fase de modelagem do sistema.

Autenticação

Deve utilizar o modelo JWT (JSON Web Token) para realizar a autenticação no sistema.

Autorização

Os endpoints devem ter autorização baseada nas funcionalidades descritas acima. Onde

a funcionalidade deve ser somente feita pelo ator estabelecido nela.

Exemplo: O **administrador** poderá cancelar o agendamento.

Administrador: Ator

Cancelar o agendamento: Funcionalidade do endpoint.