**SP MEDICAL GROUP**

Documentação

Sumário

[1. Resumo 3](#_Toc71092221)

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

3. Banco de dados relacional................................................................................................3

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

[Modelo Conceitual 3](#_Toc71092224)

[Modelo Lógico 3](#_Toc71092225)

[Modelo Físico 3](#_Toc71092226)

[Cronograma 3](#_Toc71092227)

[Trello 4](#_Toc71092228)

[5. 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

Documento baseado no projeto, onde foram feitos as 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 cuja estrutura dos dados possuem tabelas relacionadas (Diferentemente do Banco de dados Não-Relacional, onde toda informação é agrupada e guardada no mesmo registro), onde traz para o usuário uma facilidade quanto ao entendimento.

# 4. Modelagem de Dados

Modelagem de dados é uma estratégia onde é feito uma estrutura do sistema visual para mostrar as relações das tabelas por meio dos dados, facilitando o entendimento.

## Modelo Conceitual

Essa modelagem tem a função de mostrar as entidades de maneira objetiva, simples onde exibe também as cardinalidades.

Diagrama

Descrição gerada automaticamente com confiança baixa

## Modelo Lógico

Essa modelagem tem a função de mostrar as entidades de maneira mais detalhada, com seus campos onde também exibe as cardinalidades.

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

## Modelo Físico

Essa modelagem tem a função de mostrar as entidades de maneira detalhada, com seus campos onde também exibe os registros com os dados.

Diagrama

Descrição gerada automaticamente

## Cronograma

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 | Dia 5 |
| Modelo Conceitual | X |  |  |  |  |
| Modelo Lógico | X |  |  |  |  |
| Modelo Físico | X |  |  |  |  |

### Trello

https://trello.com/b/AG3rlvvC/projetospmedgroup

# Back-End

O Back-End foi feito com uma API, com a IDE 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.

Foram utilizados também:

**ENTITY FRAMEWORK** – O Entity Framework é uma ferramenta presente na plataforma .NET, parte integrante do pacote de tecnologias do ADO.NET.

**JWT** – Json Web Token é um padrão aberto, documentado pelo RFC 7519. Com ele conseguimos transmitir informações, garantindo a sua autenticidade, podendo ser usado na autenticação de APIs, sistemas ou em ações mais específicas, como recuperar a senha de um usuário, por exemplo. Ele é usado para realizar autenticação entre duas partes por meio de um token assinado que autentica uma requisição web.

**SWAGGER** – é utilizado para a documentação dos endpoints da API.

Como executar e testar a API passo a passo:

* Clone o repositório do GIT
* Abra a pasta API
* Abra a pasta senai.sp\_med\_group.webApi
* Abra solução senai.sp\_med\_group.webApi.sln (irá abrir o visual studio)
* Abra a pasta “Context“ na aba da direita

Tela de computador

Descrição gerada automaticamente

* Dentro da pasta, abra a SpMedContext.cs, e altere a string de conexão (onde está escrito “optionsBuilder”), colocando os dados correspondentes à sua máquina.

Tela de computador

Descrição gerada automaticamente

* Inicie a API na aba superior

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

* Abra o Postman, clique em “Import”, selecione o arquivo do postman na pasta “Postman” e teste as funcionalidades nas requests da collection.

# Front-End

Interface gráfica do sistema, desenvolvida com a biblioteca **React.Js.**

Tem como função permitir a interação e visualização com o sistema web.

## Funcionalidades

### Perfis de usuário:

1. **Administrador**: Para o colaborador da área administrativa da clínica;
2. **Paciente:** Clientes da clínica;
3. **Médico:** Colaboradores que atuam na área da saúde;

### Funcionalidades:

1. O **administrador** poderá cadastrar qualquer tipo de usuário (administrador,

paciente ou médico);

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);

1. O **administrador** poderá cancelar o agendamento;
2. 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);

# O **paciente** poderá visualizar suas próprias consultas;

# Mobile

Aplicação mobile desenvolvida em JavaScript, a partir do framework híbrido **React Native.**

### Perfis de usuário:

1. **Médico**: Colaboradores que atuam na área da saúde;
2. **Paciente**: Clientes da clínica;

### Funcionalidades:

1. O **paciente** poderá visualizar suas próprias consultas;

2. O **médico** poderá ver as consultas (os agendamentos) associados a ele;