**SP MEDICAL GROUP**

Documentação

Sumário

[1. Resumo 3](#_gjdgxs)

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

[3. Modelagem de Dados 3](#_1fob9te)

[Modelo Conceitual 3](#_3znysh7)

[Modelo Lógico 3](#_2et92p0)

[Modelo Físico 3](#_tyjcwt)

[Cronograma 3](#_3dy6vkm)

[Trello 4](#_1t3h5sf)

[4. Back-End 4](#_4d34og8)

[Funcionalidades 5](#_2s8eyo1)

[Sistema Web 5](#_17dp8vu)

[Perfis de usuário: 5](#_3rdcrjn)

[Funcionalidades: 5](#_26in1rg)

[Sistema Mobile 5](#_lnxbz9)

[Perfis de usuário: 5](#_35nkun2)

[Funcionalidades: 5](#_1ksv4uv)

# Resumo

Este documento trata-se de um projeto, cujo respectivo nome é SP Medical Group. Neste projeto foram elaboradas diversas partes do processo, envolvendo Banco de Dados, Back-End, UI-UX, Front-End e Mobile.

# Descrição do projeto

Este projeto trata-se de uma nova clínica, 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. O serviço foi solicitado para desenvolver um sistema web/mobile integrado, onde seja possível pesquisar dados sobre as informações de seus pacientes.

# Modelagem de Dados

Modelagem de Dados é o processo em que levantamos, analisamos, separamos e organizamos todos os dados e tipos de informações que serão utilizados.

## Sobre o 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.

Sistema Web

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;

Sistema Mobile

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;

### Trello

<https://trello.com/b/dh81A5Kz/sp-medical-group>

# **Banco de Dados**

## Modelo Conceitual

O modelo conceitual é um modelo mais simples, que serve para o começo da nossa organização, mostrando as entidades e os relacionamentos, facilitando a compreensão do usuário.

Diagrama

Descrição gerada automaticamente

## Modelo Lógico

O modelo lógico é aquele onde nós organizamos as chaves primárias e estrangeiras e os valores de cada tabela.

Diagrama

Descrição gerada automaticamente

## Modelo Físico

O Modelo Físico é o modelo que nos mostra o Banco de Dados de maneira física com alguns dados de exemplo para testar os tipos de relação etc.

Diagrama

Descrição gerada automaticamente

## Cronograma

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

# **Back-End**

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.

**JWT** – JSON Web Token – Validação do Usuário.

**ORM** – Object Relational Mapper – Mapeamento de Objeto Relacional.

**SWAGGER** – Interface para Documentação dos Métodos da API.

Passo a passo de como executar a API criada e testar as requisições no Postman:

Para conseguirmos executar a API, devemos abrir a solução senai.SpMedicalGroup.webApi, em seguida troque na seta de  para , então aperte botão de executar, com isso sua API estará sendo executada.

Em seguida, abra o aplicativo Postman, crie uma coleção, dentro dessa coleção crie pastas com os respectivos nomes das Domains e dentro de cada pasta, crie requisições.

Depois de criar uma requisição a renomeie, após renomear, insira a URL http://localhost:5000/api/(Nome da Controller).

Para realizar o método, selecione ao lado o verbo que deseja:

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

Os principais métodos utilizados para o CRUD são:

**GET** - Listar

**POST** - Postar

**PUT** – Atualizar

**PATCH** – Atualizar apenas um atributo do registro

**DELETE** – Deletar

Escolha o verbo e execute (Existem métodos que os Ids podem ser passados direto na URL, métodos que necessitam de corpo e existem os que possuem ambos), se for necessário autenticação ou autorização para executar o método, insira o token, que é obtido através do método Login.

## Funcionalidades

Nós utilizamos a API (Application Programming Interface), com o modelo JWT (JSON Web Token) para realizar a autenticação no sistema, utilizamos o Entity Framework para a realização dos CRUDS (Create, Read, Update e Delete) de todas as entidades, também utilizamos os pacotes do Nuget para realizar a aplicação.

## Sistema Web

### 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;

# **Front-End**

A aplicação Web foi desenvolvida com o intuito de ser a interface em que o usuário fará interação com o banco de dados e API.

Foi utilizada a biblioteca ReactJs para desenvolver a parte do front-end e realizar a integração com a API desenvolvida anteriormente. Esta ontendo Login, Cadastro de Consultas, Listagem de Consultas conforme o usuário logado, logout e responsivo(iPad).

Utilizamos os pacotes:

Passo a passo:

1. Primeiramente rode o comando no terminal para criar um projeto front-end,: npx create-react-app nomeprojeto
2. Crie as pastas pages, services...
3. Crie os arquivos dentro das pages.
4. Rodar comando para iniciar o projeto: npm start
5. Continuar desenvolvimento.

Telas

Administrador / Médico / Paciente

1. Login;

Administrador

1. Cadastrar nova consulta;

2. Listagem de consultas;

Médico

1. Listagem de consultas;

2. Inserir uma nova descrição na consulta;

Paciente

1. Listagem de consultas;

# **Mobile**

A aplicação Mobile foi desenvolvida com o intuito de melhorar a usabilidade da plataforma, focando no paciente e em quais informações ele precisa visualizar.

O aplicativo mobile é integrado com a API que foi desenvolvida anteriormente, feito com React Native e possui as seguintes funcionalidades:

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 os agendamentos associados a ele;

Persistência de dados

Os dados referentes ao usuário logado na aplicação mobile deverão ser salvos

localmente.

Telas

Médico / Paciente

1. Login

Médico

1. Visualizar as suas consultas

Paciente

1. Visualizar as suas consultas

Principais comandos:

Para criar:

npx react-native init novoProjeto

Para executar:

npx react-native run-android