

Documentação

SenaiMEDGROUP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Alameda Barão de Limeira, 539 – São Paulo/SP |  |  |  | sp.senai.br |

Sumário

1. [Resumo 2](#_Toc533767843)

[Objetivos](#_Toc533767844)

1. [Descrição do projeto 2](#_Toc533767845)

[Resumo do projeto](#_Toc533767846)

1. [Modelagem de Software 3](#_Toc533767847)

[Modelo Lógico](#_Toc533767848)

[Modelo Físico](#_Toc533767849)

[Modelo Conceitual](#_Toc533767850)

[Cronograma](#_Toc533767851)

1. [Funcionalidades 5](#_Toc533767852)

[Web](#_Toc533767853)

[Mobile](#_Toc533767854)

1. [Protótipos 6](#_Toc533767855)

[Web](#_Toc533767856)

[Mobile](#_Toc533767857)

1. [Front-End 7](#_Toc533767858)
2. [Mobile 8](#_Toc533767859)
3. [Arquitetura do Projeto 9](#_Toc533767860)
4. [Referências 10](#_Toc533767861)

[Links](#_Toc533767862)

[Livros](#_Toc533767863)

Resumo

Objetivos

O seguinte documento tem como objetivo registrar a criação do projeto do Senai: “SenaiMEDGROUP”. Neste documento se apresenta as especificações técnicas, capacidades, etc do projeto acima mencionado.

Descrição do projeto

O projeto se refere a um sistema de dados para a área de saúde/clínicas/medicina para aprendizado de banco de dados, API, REACT, etc. Através dessas funcionalidades será possível cadastrar médicos, pacientes, clínicas, consultas, etc. Podendo ser acessado através da Web e programado usando as capacidades FRONT END, BACK END e MOBILE.

Resumo do projeto

Um projeto de sistema para cadastros de usuários, serviços, localidades disponíveis em vários ambientes, seja web, seja mobile.

Modelagem de Software

Modelo Lógico

Tabelas apresentadas com suas chaves primárias e estrangeiras

Chaves primárias e estrangeiras de acordo com as entidades/tabelas apresentadas.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Consulta | | | | | |  |  |
| ID | CNPJ | Endereco | Data\_Inicio | Razao\_Social | Nome\_Fantasia | Horario\_Funcionamento | Id\_Medico |
| 1 | 321321321 | Alameda Barão de Limeira, 539 - Santa Cecilia, São Paulo - SP, 01202-001 | 2019-02-11 | SP Medical Group | SP Medical Group | SP Medical Group | 7:00  As  19:00 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Usuários | | | | | | |  |  |
| ID | Nome | CPF | RG | Telefone\_Contato | Email | Senha | ID\_Tipo\_Usuario | Data\_Nascimento |
| 1 | Ricardo | 8640090230 | 8640090230 | (11) 986400230 | ricardo@email.com | 123 | 1 | 2019 |
| 2 | Roberto | 8640090230 | 8640090230 | (11) 986400230 | roberto@email.com | 123 | 1 | 2019 |
| 3 | Helena | 8640090230 | 8640090230 | (11) 986400230 | helena@email.com | 123 | 1 | 2019 |
| 4 | Ligia | 94839859000 | 435225435 | (11) 986400230 | ligia@email.com | 123 | 2 | 2019 |
| 5 | Alexandre | 73556944057 | 326543457 | (11) 986400230 | alexandre@email.com | 123 | 3 | 2019 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo\_Médicos | | | |
| ID | CRM | ID\_Area\_Clinica | Id\_Usuario |
| 1 | 54356-SP | 2 | 15 |
| 2 | 53452-SP | 17 | 16 |

|  |  |
| --- | --- |
| Tipos de Usuário | |
| ID | Tipo |
| 3 | Administrador |
| 1 | Médico |
| 2 | Paciente |

|  |  |
| --- | --- |
| Status\_Consulta | |
| ID | Status |
| 1 | Agendada |
| 3 | Cancelada |
| 2 | Realizada |

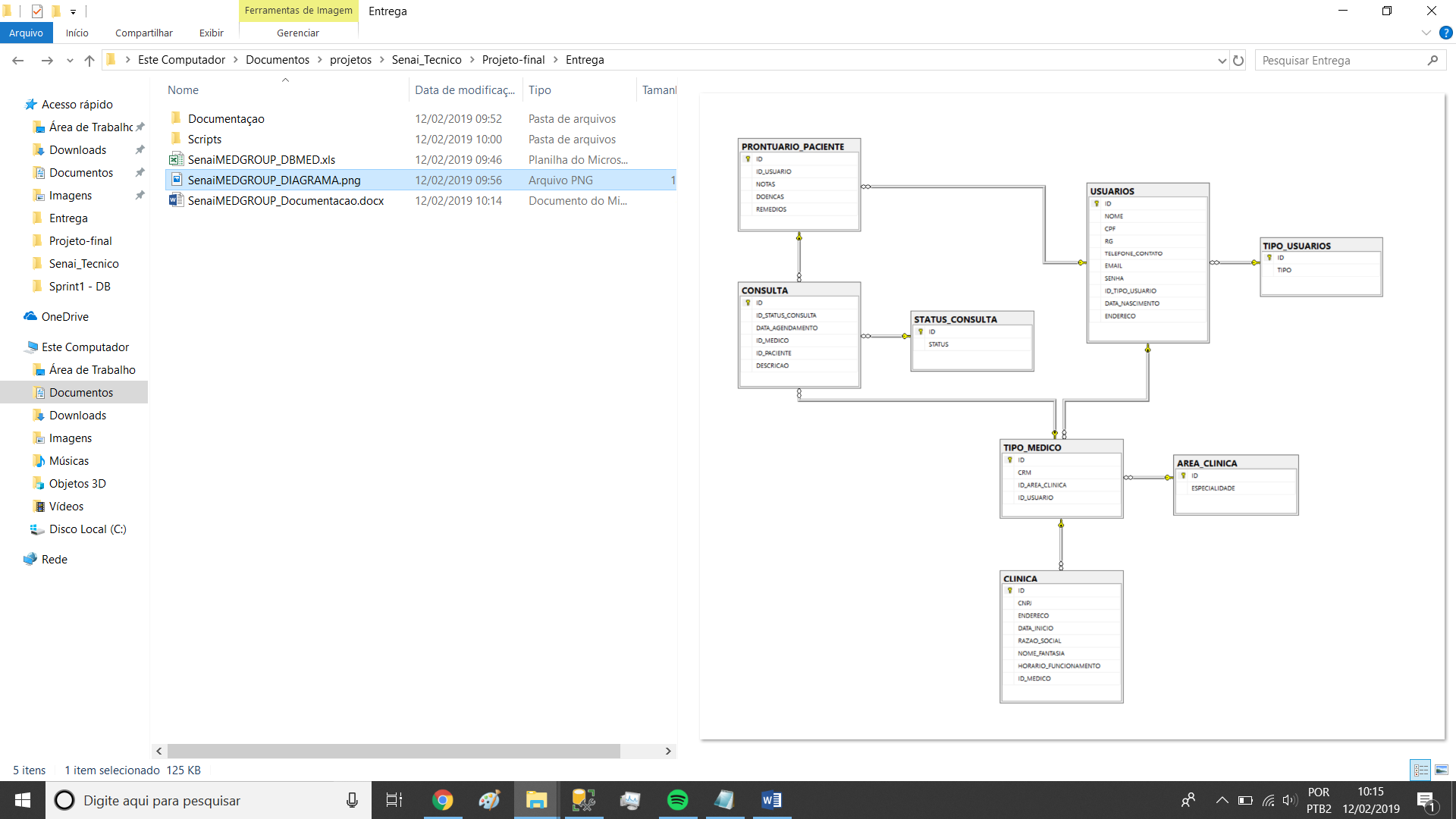
|  |  |
| --- | --- |
| Area\_Clinica | |
| ID | Nome |
| 1 | Psiquiatra |
| 2 | Cardiologista |
| 3 | Endocrinologista |
| 4 | Nutricionista |
| 5 | Pediatra |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prontuario\_Paciente | | | | | |
| ID | ID Usuário | Notas | Doencas | Remedios |
| 1 | 1 | 18 | EBOLA | CHAZINHO |
| 2 | 2 | 19 | ESTRESSE | CHAZINHO |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Consultas | | | | | |
| ID | ID Paciente | ID Médico | Data da consulta | Observações | ID Status |
| 1 | 1 | 2 | 12/04/2019 | Foi receitado medicamento x... | 2 |
| 2 | 2 | 1 | 05/06/2019 | Precisa ser medicado diariamente. | 1 |

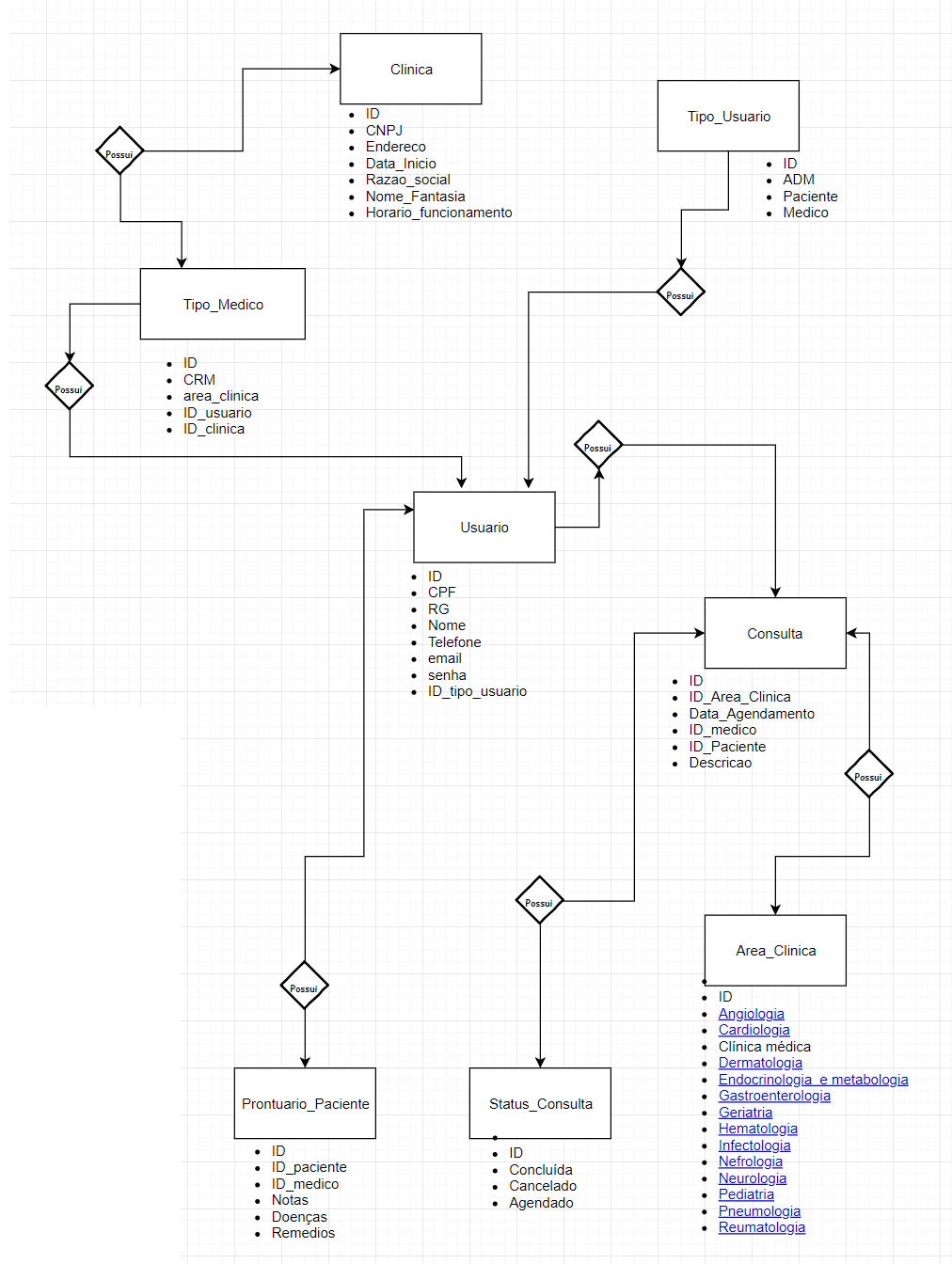
Modelo Físico

Diagrama do Banco de Dados expondo a forma com que os dados estão organizados



Modelo Conceitual

O banco de dados é composto por 8 tabelas, interligadas pela tabela de usuários que possui informações que serão utilizadas para dar base as tabelas Tipo\_Medico que especifica o usuário médico e Prontuario\_Paciente que especifica o usuário paciente. Assim as tabelas Clinica e Consulta podem ser inseridas se ligando com médico e pacientes respectivamente



Cronograma

|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 | Dia 5 | Dia 6 | Dia 7 | Dia 8 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Modelo Lógico** | x |  |  |  |  |  |  |  |
| **Modelo Físico** | x |  |  |  |  |  |  |  |
| Modelo Conceitual | x |  |  |  |  |  |  |  |

Funcionalidades

Web

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 vinculado ao paciente (prontuário);

7. O paciente poderá visualizar suas próprias consultas;

Mobile

1. O paciente poderá visualizar suas próprias consultas;

2. O paciente poderá visualizar a rota entre sua localização e o endereço da clínica;

3. O médico poderá ver os agendamentos associados a ele;

4. Notificação: No momento que o administrador cadastrar um agendamento, o médico deverá receber uma notificação pelo aplicativo;

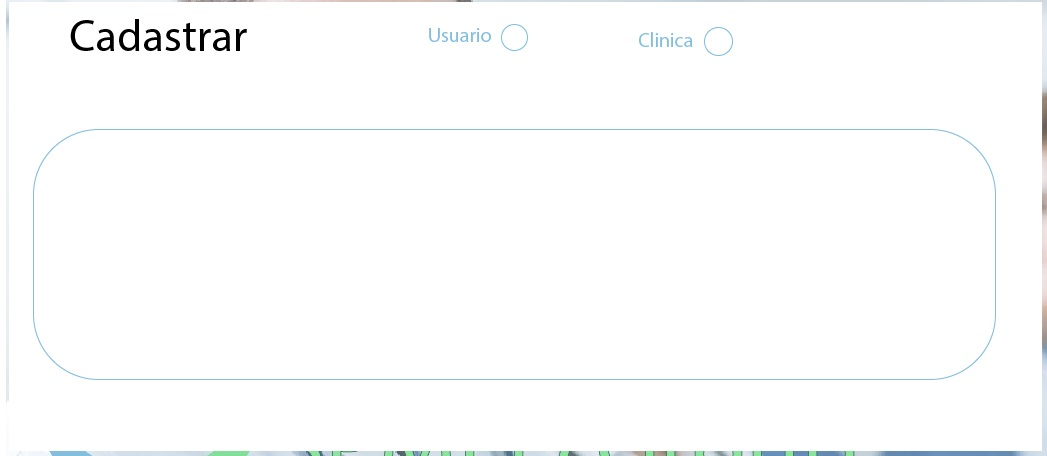
Protótipos

Web

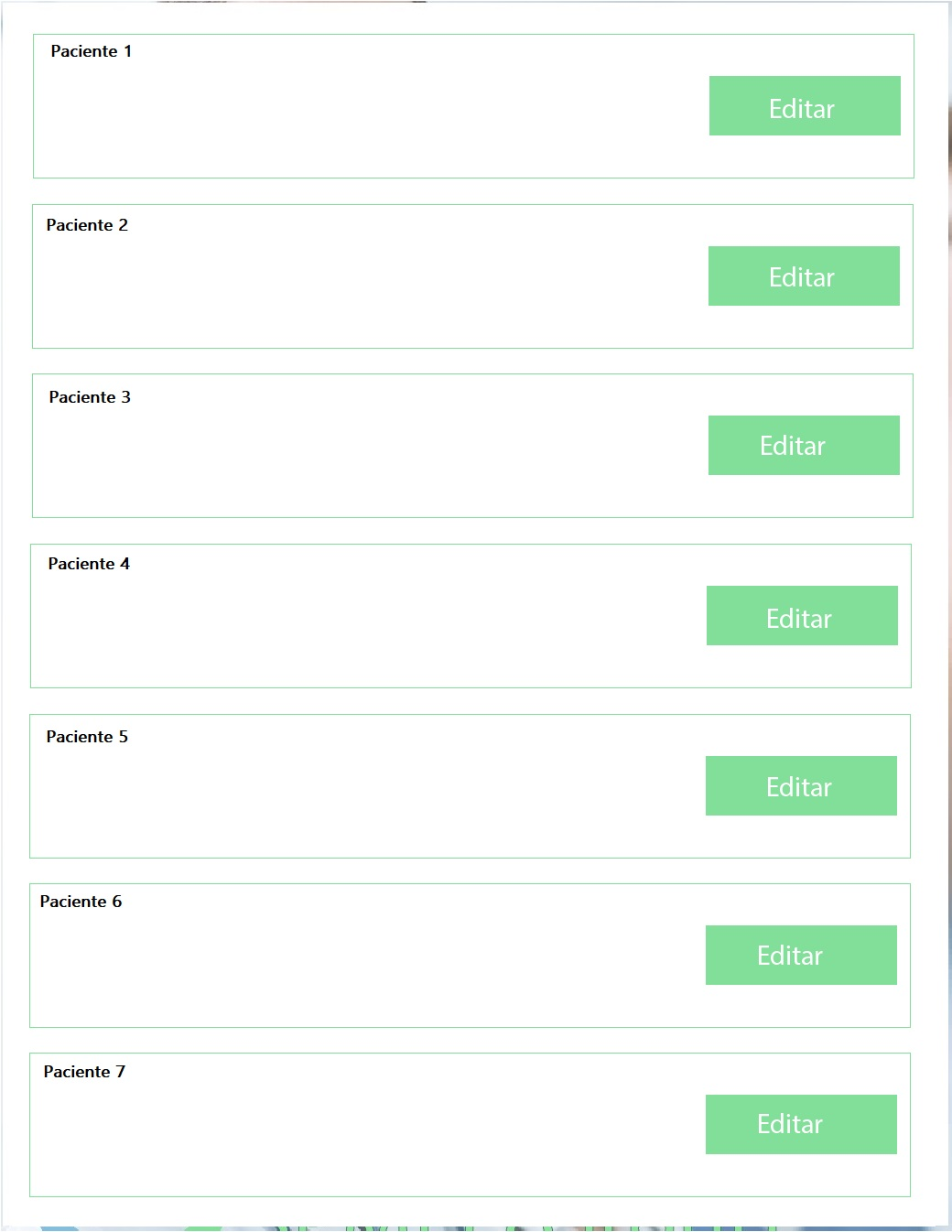
Homepage



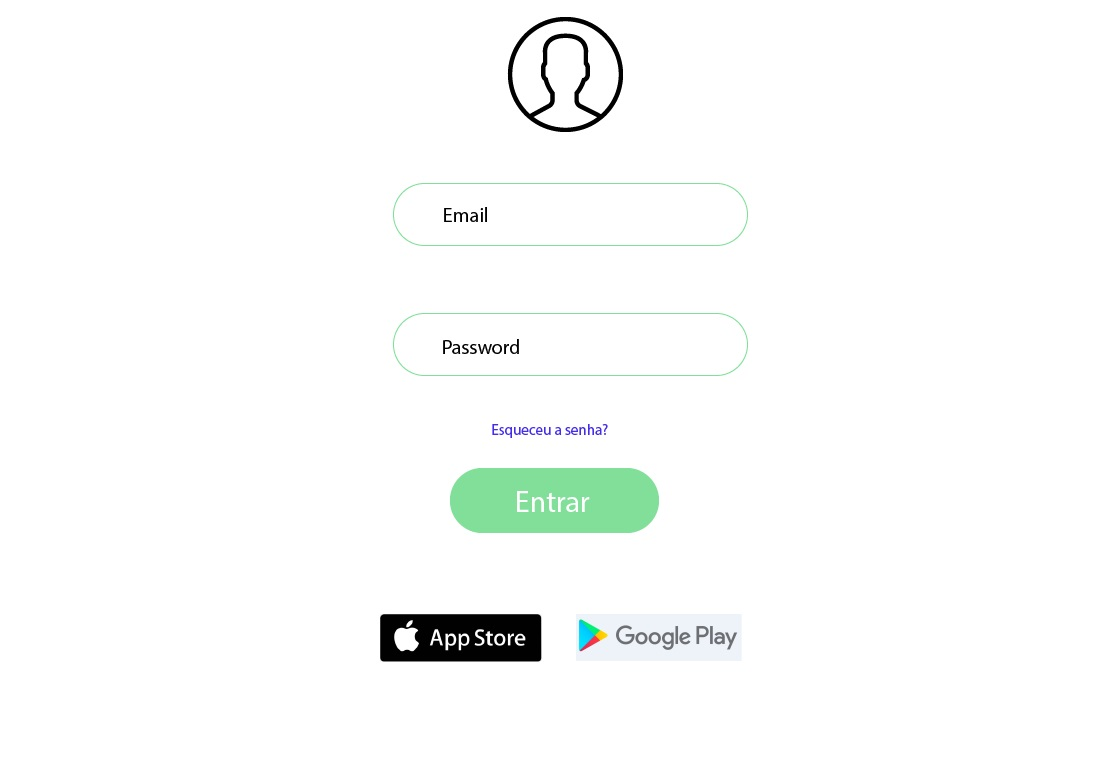
Página de Cadastro



Página de Listagem de Consultas

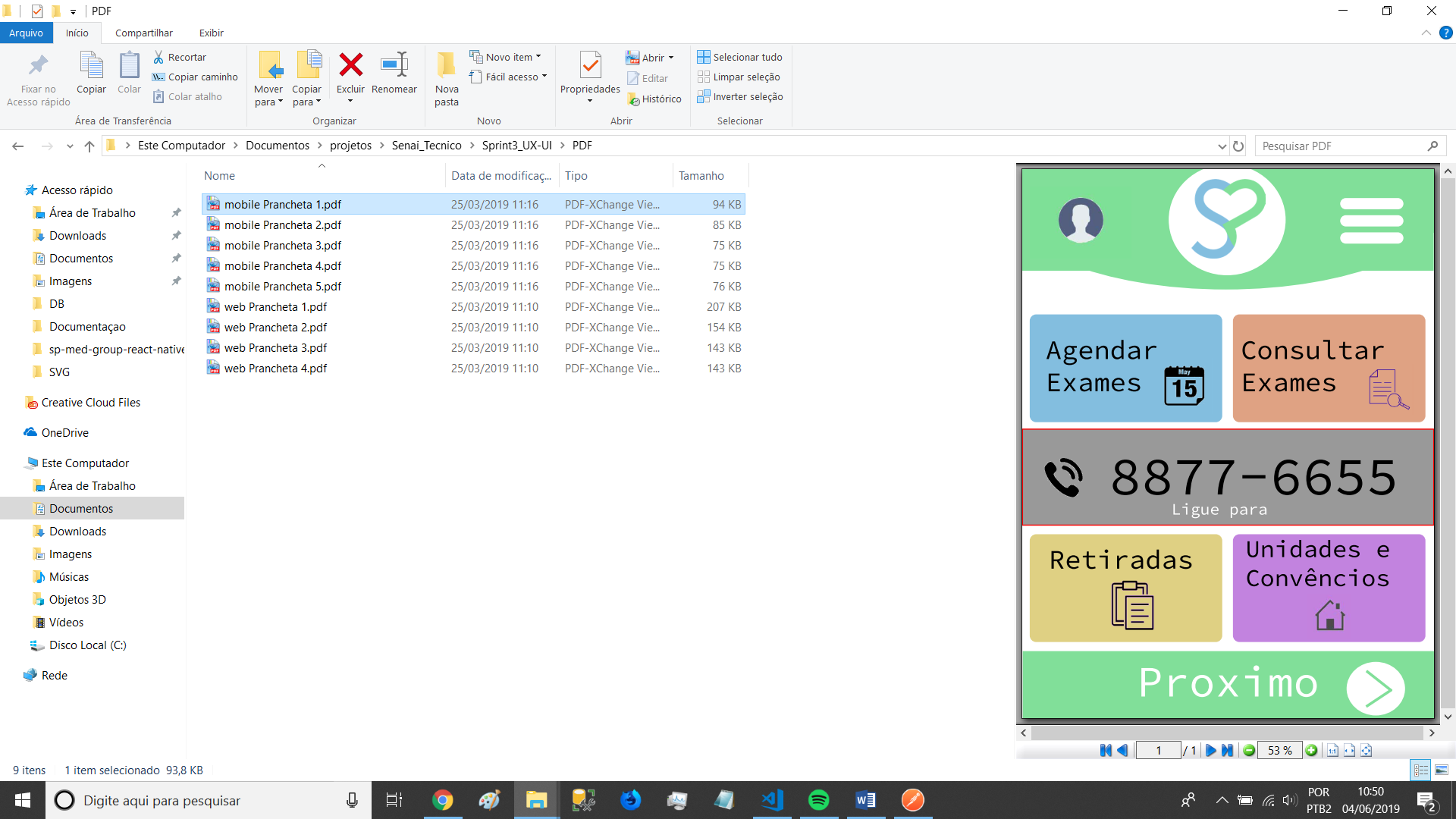


Página de Login



Mobile

Página principal



Back-End

**Tutorial (Database-first)**

**Utilizando os arquivos do Github**

Primeiramente é necessário baixar os arquivos no site: <https://github.com/ArturLOtt/Senai-Sprint2-Projeto-Final.git>

**Criação do banco de dados**

Requisito: Faça uso do Microsoft SQL Server Managements Studio ou outro programa de sua preferência que suporte a linguagem SQL

Na pasta Scripts, execute os três arquivos ali presentes, na seguinte ordem, para criar o banco de dados. Utilize a Tecla F5 ou clique na seta verde para executar os comandos.

* DDL – CRIACAO SPMEDGROUP
* DML – MANIPULACAO SPMEDGROUP
* DQL – QUERY SPMEDGROUP)

**Utilizando o banco de dados na aplicação**

Requisito: Faça uso do Visual Studio ou outro programa de sua preferência que suporte a linguagem C# e o Entity Framework

Abra o arquivo SPMEDGROUP\_MANHA.SLN no Visual Studio e prossiga com o próximo passo.

Utilizando o Gerenciador de Pacotes do Nugget, instale os seguintes arquivos:

* Microsoft.EntityFrameworkCore.Design;
* Microsoft.EntityFrameworkCore.SqlServer;
* Microsoft.EntityFrameworkCore.Tools;
* Sendgrid;
* Swashbuckle.AspNetCore;
* Swashbuckle.AspNetCore.Swagger;

Após isso, abra o “Console de Gerenciador de Pacotes”, dentro de “Ferramentas” no menu superior e execute o seguinte comando:

Scaffold-DbContext "Data Source=.\SqlExpress;Initial Catalog=SPMEDGROUP;User Id=sa;pwd=132;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Domains -ContextDir Contexts -Context SPMEDGROUPContext

Caso haja algum erro, verifique se o comando que esta tentando executar está igual ao comando apresentado. Caso o erro persistir adicione “-force” no final da linha desse comando e execute novamente, isso irá forçar o comando a ser executado.

Ao termino dessa operação você terá feito a ligação do banco de dados ao projeto.

**Fazendo requisições da sua aplicação**

Requisito: Faça uso do Postman ou outro programa de sua preferência em que seja possível fazer requisições com a internet.

Com o programa aberto, Clique em “Import” e selecione o arquivo “**SPMEDGROUP.postman\_collection.json**” para utilizar poder utilizar todas as funcionalidades.

Volte ao Visual Studio e coloque sua aplicação para funcionar. Para ver as requisições possíveis dessa API, utilize o seguinte link:

http://localhost:5000/swagger

Para fazer uso de qualquer um dos métodos, abra o respectivo arquivo, preencha as informações necessárias e clique o grande botão azul SEND. Verifique o Status da requisição, se der 200 OK, ela foi executada com sucesso, caso contrário, há algum erro e é melhor revisar os passos anteriores.

Front-End

O projeto web foi realizado utilizando a tecnologia javascript através da biblioteca React

**Dependências:**

Os passos anteriores (Scripts e BackEnd) devem ser efetuados para o correto funcionamento dessa aplicação.

Antes dessa etapa, verifique a String de conexão no arquivo do BackEnd e modifique para a que for de sua utilidade. (Há duas Strings de conexão disponíveis, uma com os serviços da Azure e outro para uso local através do Localhost)

**Tutorial**

Baixe o Node.js através deste link. Caso já possua, pule esta etapa:

<https://nodejs.org/en/>

Sugere-se que se baixe a versão mais estável recomendada (a de menor numeração)

A seguir abra o prompt de comando e digite:

npm install –g create-react-app

A seguir vá para a pasta em que o foi baixado o arquivo do Github, e dentro da pasta FrontEnd haverá duas pastas sp-med-group-react e sp-med-group-react-native. O primeiro se refere ao projeto web e o segundo se refere ao projeto mobile. Entre em sp-med-group-react e abra o command prompt dentro dessa pasta e digite o comando

npm install

Assim que terminar a instalação das dependências do node modules, digite e rode o comando abaixo para iniciar o projeto no seu browser:

npm start

Se tudo ocorrer corretamente, o programa deve estar correndo normalmente no seu browser de default automaticamente.

**Informações sobre o Projeto**

Para usuários não cadastrados é possível a navegação pelo site e realizar o Login de usuários já previamente cadastrados.

Para usuários Médicos é permitido a listagem de todas as consultas desse médico e todos os pacientes que esse médico tem consultas. Essas funções estão presentes em uma página específica própria do médico a que ele tem acesso após o login.

Para usuários Pacientes é permitido a listagem de todos as consultas desse paciente. Essas funções estão presentes em uma página específica própria do paciente a que ele tem acesso após o login.

Para usuários administradores é permitido a listagem de todos os usuários cadastrados assim como é possível cadastrar novos. Também é permitido a eles alterar dados cadastrados como descrição de consulta, etc. Essas funções estão presentes em uma página específica própria do administrador a que ele tem acesso após o login.

Mobile

O projeto mobile foi realizado utilizando React-Native, um framework para criação de aplicações híbridas (tanto Android quanto iOS)

**Dependências:**

Seguindo os passos anteriores, para o React Native não é necessário ter efetuado os passos do FrontEnd, por isso essa parte irá replicar alguns passos já efetuados no anterior. Também é necessário já ter feito os passos relacionados aos Scripts e BackEnd para o correto funcionamento dessa aplicação.

Assim como na etapa de FrontEnd, verifique a String de conexão no arquivo do BackEnd e modifique para a que for de sua utilidade. (Há duas Strings de conexão disponíveis, uma com os serviços da Azure e outro para uso local através do Localhost)

**Tutorial**

Baixe o Node.js através deste link. Caso já possua, pule esta etapa:

<https://nodejs.org/en/>

Sugere-se que se baixe a versão mais estável recomendada (a de menor numeração)

A seguir baixe o Android Studio para desenvolvimento Android, XCode para desenvolvimento iOS ou qualquer outro aplicativo de sua preferência que dê suporte ao desenvolvimento mobile E que possua um emulador incluído para executar a aplicação.

Nota: É possível utilizar somente seu aparelho celular para esta etapa.

A seguir abra o prompt de comando e digite:

npm install –g create-react-app

A seguir vá para a pasta em que o foi baixado o arquivo do Github, e dentro da pasta FrontEnd haverá duas pastas sp-med-group-react e sp-med-group-react-native. O primeiro se refere ao projeto web e o segundo se refere ao projeto mobile. Entre em sp-med-group-react e abra o command prompt dentro dessa pasta e digite o comando

npm install

Assim que terminar a instalação das dependências do node modules, digite e rode o comando abaixo para iniciar o projeto no seu browser:

npm start

Se tudo ocorrer corretamente, o programa deve estar correndo normalmente no seu browser de default automaticamente.

**Informações sobre o Projeto**

Para usuários não cadastrados é possível a navegação pelo site e realizar o Login de usuários já previamente cadastrados.

Para usuários Médicos é permitido a listagem de todas as consultas desse médico e todos os pacientes que esse médico tem consultas. Essas funções estão presentes em uma página específica própria do médico a que ele tem acesso após o login.

Para usuários Pacientes é permitido a listagem de todos as consultas desse paciente. Essas funções estão presentes em uma página específica própria do paciente a que ele tem acesso após o login.

Para usuários administradores é permitido a listagem de todos os usuários cadastrados assim como é possível cadastrar novos. Também é permitido a eles alterar dados cadastrados como descrição de consulta, etc. Essas funções estão presentes em uma página específica própria do administrador a que ele tem acesso após o login.

react-native run-android

Nota: tenha certeza de rodar esse comando no usuário da sua máquina que tem o node modules configurado corretamente para todos os usuários ou tem permissão de administrador. Em caso de falha por não reconhecimento do comando, utilizar o código: C:\Users\{NomeDoUsuario}\AppData\Roaming\npm\react-native run-android , modificando o {NomeDoUsuario} pelo nome do usuário que possui o node modules do react instalados e que está tentando efetuar o comando anterior (o usuário atual)

### Preparação

Antes de iniciar o projeto você deve fazer tudo que está nos links acima. Então você deverá :

* Você tambem devera ir para a API e setar o mesmo ip para ela no arquivo [Properties/launchSettings](https://github.com/Chingling152/WebApi-SPMedGroup/blob/master/Senai.WebApi.SpMedGroup/Properties/launchSettings.json)
* O proximo comando sera : **react-native run-android**
* O projeto ira abrir na avd ou no seu celular (depende qual você esta usando)

### Bibliotecas

Aqui está todas as bibliotecas que serão instaladas ao ser digitado o comando **npm install** na pasta do projeto (alem do react-native)

* **react-navigation** - navegação entre paginas
* **react-native-gesture-handler** - lidar com scroll e gestos de zoom e rolagem
* **@react-native-community/async-storage** - armazenar dados do usuario localmente
* **jwt-decode** - validar dados do usuario
* **react-native-maps** - gerar mapas
* **react-native-maps-direction** - exibir rota entre 2 pontos
* **react-native-google-maps-directions** - Enviar dados para o google maps

Essas bibliotecas também deverão ser linkadas ao projeto utilizando o comando:

react-native link {nome da biblioteca}

### Google API

Caso a minha API do google não esteja funcionando (porque há um limite de requisições). Você devera criar uma nova, [neste link](https://console.developers.google.com/google).  
Lá você tera que gerar uma key para as seguintes APIs.

* Directions API
* Geocoding API
* Maps SDK for Android

## Telas

Apos iniciar o projeto a primeira tela sera a de login (você precisa pelo menos ter um valor cadastrado no banco de dados ([Veja: Banco de dados- Valores inicias](https://github.com/Chingling152/SQL-SPMedGroup/blob/master/Essenciais/5-Valores_iniciais.sql)) ou insira algum usando a API/Site)  
Ao logar você poderá apenas visualizar suas consultas, ou clicar em uma delas e ver todas as suas informações.  
Todas as telas só são acessiveis depois de o usuario estar logado.

Tela de login

Caso ocorra algum problema (com a API por exemplo) será retornado apenas um erro sem muitos detalhes.  
O mesmo acontece com erro de autenticação e validação.

Ao entrar na tela de visualizar consultas, você verá todas as consultas (caso você logue como administrador você podera ver todas , mas não podera altera-las porque esta função só é disponivel no [Site da SPMedicalGroup](https://github.com/Chingling152/React-SpMedGroup))

Tela de listagem de consultas

Na tela de visualizar terá apenas alguns detalhes a mais. Como por exemplo a descrição completa (que na listagem de todas as consultas apenas mostra no maximo 114 caracteres).

Tela de listagem de consulta

Você podera ver a sua localização atual e o caminho para a clinica onde a consulta foi marcada. Para isso voce deverá permitir que o aplicativo use a função GPS do seu celular e estar com ela ligada.

## Services

## Cronograma

O projeto foi iniciado no dia **13/05/2019** e finalizado no dia **23/05/2019**.  
Aqui ficara meu cronograma do projeto com todas minhas atividades.

* **Dia 0**  
  Eu ja tinha o layout pronto do mobile que foi feito na criação do site (então deixei isso como dia 0 porque não aconteceu no dia do projeto em si, mas foi usado nele (um pouco))
* **Dia 1**
  + Iniciei a arquitetura do projeto criando todas as paginas (apenas com um Text nela com o nome da pagina).
  + Instalei quase todas as bibliotecas necessarias (AsyncStorage foi instalada no **dia 3**).
  + Iniciei a pagina de login. Fiz os primeiros testes de login.
* **Dia 2**
  + Importei meu arquivo de services padrão (Conexão com API e Validação de token).
  + Recebi o token e lidei com os erros de email/senha.
  + Criei as rotas entre as telas e fiz a o usuario ser redirecionado para a pagina de listar consultas ao logar.
  + Iniciei a pagina de visualização de consultas.
  + Componentizei a consulta assim poderia buscar os valores.
* **Dia 3**
  + Iniciei a estilização do projeto.
  + Adicionei uma Splashscreen ao projeto e um icone.
  + Estilizei a pagina de Login e o StackNavigator.
  + Instalei a biblioteca para o AsyncStorage e iniciei o login automatico.
  + Iniciei a função de deslogar o usuario (ainda não estava 100% funcional).
* **Dia 4**
  + Finalizei a visualização de consultas.
  + Finalizei a função de logout.
  + Iniciei a visualização de apenas uma consulta.
  + Adicionei um indicador de carregamento nas telas de login e listagem de consulta.
* **Dia 5**
  + Finalizei a listagem de uma consulta.
  + Iniciei a documentação colocando informações ate aqui (fui incrementando esse cronograma a cada dia)
  + Adicionei feedback aos usuarios que não tem consultas
  + Adicionei uma pagina inicial para verificar dados do usuario e API
* **Dia 6**
  + Passei o dia inteiro retocando o projeto (testando vulnerabilidades).
  + Arrumei o design do aplicativo
  + Fixei alguns problemas de UX
* **Dia 7**
  + Iniciei a parte de localização do usuario.
  + Tentei resolver problemas com permissão do usuario para acessar o GPS.
  + Finalizei os requisitos minimos do aplicativo
* **Dia 8**
  + Resolvi problemas do GPS e permissão do usuario
  + Inicei o mapa
  + Modifiquei a status bar
* **Dia 9**
  + Adicionei o mapa com a localização da consulta e do Usuario
  + Criei a API no google maps
* **Dia 10**
  + Fiz a rota da localização do usuario até a da clinica Utilizando a API do google
  + Fiz o feedback caso a API não esteja funcionando ou o local da consulta não exista
  + Não adicionei a opção de redirecionar ao google maps p

Arquitetura do Projeto

Referências

Links

Livros