

Documentação

Sumário

1. Resumo	3
2. Descrição do projeto	3
3. Banco de Dados relacional.....	3
4. Modelagem de Dados	3
Modelo Conceitual	3
Modelo Lógico	4
Modelo Físico	5
Cronograma.....	5
5. Trello.....	6
6. Back-End	6
Cronograma.....	6
Funcionalidades.....	6
7. UX-UI.....	7
8. Front-end.....	8
9. Mobile.....	9
10.NoSQL.....	9
11. Sistema Web	9
Perfis de usuário.....	9
Funcionalidades.	10
Teste do Sistema.....	10
12. Sistema Mobile.....	19
Perfis de usuário.....	19
Funcionalidades.	19
Teste do Sistema.....	20

1. Resumo

Documentação sobre o projeto SP Medical Group, contendo os detalhes do processo de criação e desenvolvimento da aplicação.

2. Descrição do projeto

Projeto criado afim de auxiliar nova clínica médica chamada SP Medical Group, empresa de pequeno porte que atua no ramo da saúde, criada pelo médico Fernando Strada em 2020 na região da Avenida Paulista em São Paulo. Fernando tem uma equipe de médicos que atuam em diversas áreas (pediatria, odontologia, gastroenterologia etc.).

Será criado um sistema que ajudará a clínica a cadastrar e organizar os dados dos pacientes, médicos, consultas, etc.

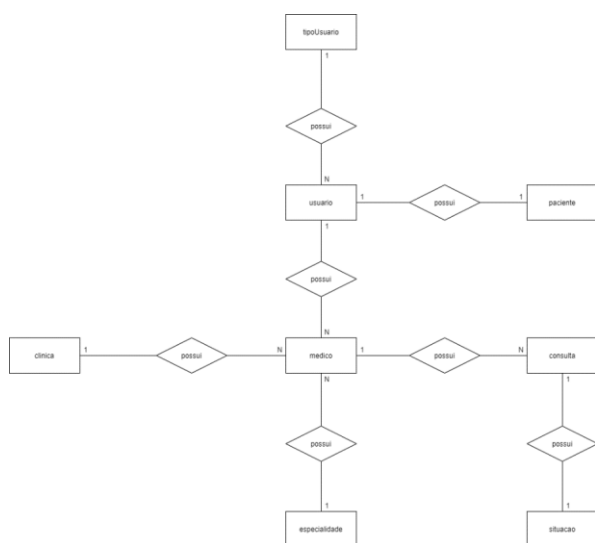
3. Banco de dados relacional

Um banco de dados relacional é um tipo de banco onde pontos de dados se relacionam entre si, através do modelo de relação é possível criar e representar bancos de dados de forma intuitiva e direta. Em um banco como esse cada registro da tabela precisa de um ID, as famosas chaves, sendo primária ou estrangeira. Graças a esse modelo se tem uma padronização de representar diversos dados de forma eficiente e de fácil acesso.

4. Modelagem de dados

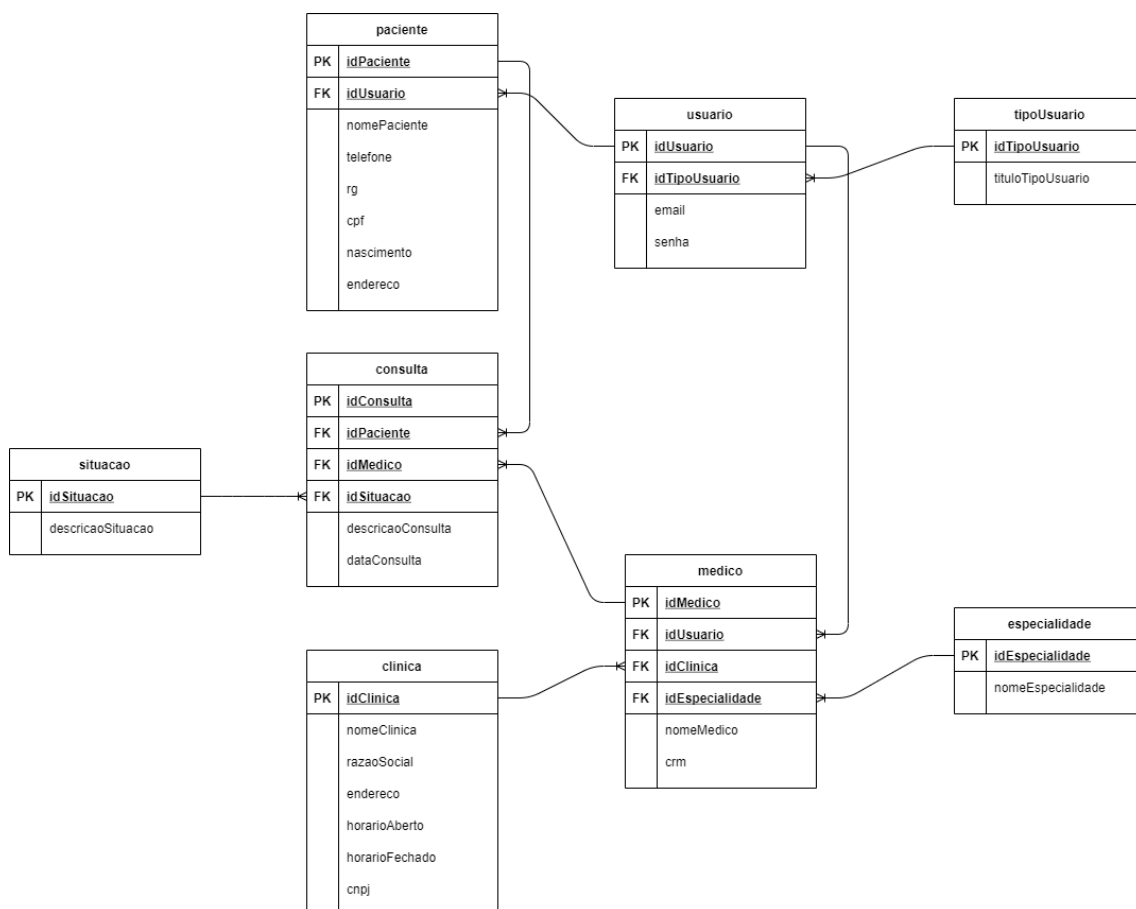
A modelagem de dados é a parte de desenvolvimento de sistemas que se encarrega de construir estruturas que permitam armazenar e recuperar informações em contextos específicos.

Modelo Conceitual



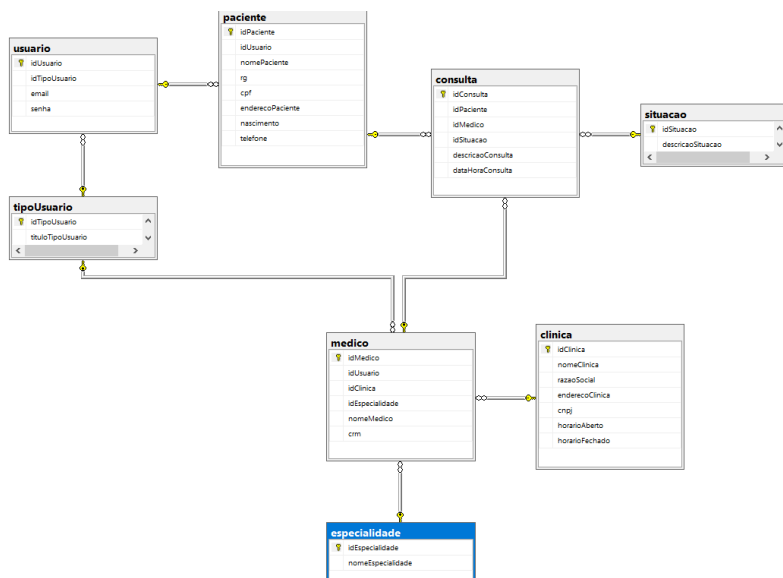
Modelagem Conceitual: Modelo mais simples de identificar as entidades e suas relações entre as tabelas, exemplo: 1:1, 1:N, N:1, N:N

Modelo Lógico



Modelagem lógica: Modelo mais complexo para identificar chaves Primárias e Estrangeiras e saber seus ID e atributos

Modelo Físico



tipoUsuario		usuario					especialidade	
idTipoUsuario	tituloTipoUsuario	idUsuario	idTipoUsuario	email	senha		idEspecialidade	nomeEspecialidade
1	Administrador	1	2	ricardo.lemos@spmedicalgroup.com.br	123		1	Acupuntura
2	Paciente	2	2	roberto.possarile@spmedicalgroup.com.br	1234		2	Anestesiologia
3	Médico	3	2	helenasouza@spmedicalgroup.com.br	12345		3	Angiologia
		4	2	ligia@gmail.com	123456		4	Cardiologia
		5	2	alexandre@gmail.com	1234567		5	Cirurgia Cardiovascular
		6	2	fernando@gmail.com	12345678		6	Cirurgia da Mão
		7	2	henrique@gmail.com	abc		7	Cirurgia do Aparelho Digestivo
		8	2	joao@hotmail.com	abcd		8	Cirurgia Geral
		9	2	bruno@gmail.com	abcde		9	Cirurgia Pediátrica
		10	2	mariana@outlook.com	acbde		10	Cirurgia Plástica
		11	1	saulo@gmail.com		C# melhor que python	11	Cirurgia Torácica
							12	Cirurgia Vascular
							13	Dermatologia
							14	Radioterapia
							15	Urologia
							16	Pediatria
							17	Psiquiatria

clinica						
idClinica	nomeClinica	razaoSocial	endereco	cnpj	horarioAberto	horarioFechado
1	Clinica Possarile	SP Medical Group	Av. Barão Limeira, 532, São Paulo, SP	86.400.902/0001-30	07:00	23:00
2	Clinica Portuguesa	Clinica Portugal	Av. Portugal, 1676, Santo André, SP	99.999.999/9999-99	08:00	22:00

paciente							
idPaciente	idUsuario	nomePaciente	rg	cpf	endereco	nascimento	telefone
1	4	Ligia	43522543-5	94839859000	Rua Estado de Israel 240, São Paulo, Estado de São Paulo, 04022-000	10/13/1983	11 3456-7654
2	5	Alexandre	32654345-7	73556944057	Av. Paulista, 1578 - Bela Vista, São Paulo - SP, 01310-200	7/23/2001	11 98765-6543
3	6	Fernando	54636525-3	16839338002	Av. Ibirapuera - Indianópolis, 2927, São Paulo - SP, 04029-200	10/10/1978	11 97208-4453
4	7	Henrique	54366362-5	14332654765	R. Vitória, 120 - Vila São Jorge, Barueri - SP, 06402-030	10/13/1985	11 3456-6543
5	8	João	53254444-1	91305348010	R. Ver. Geraldo de Camargo, 66 - Santa Luzia, Ribeirão Preto - SP, 09405-380	8/27/1975	11 7656-6377
6	9	Bruno	54566266-7	79799299004	Alameda dos Arapuanes, 945 - Indianópolis, São Paulo - SP, 04534-001	3/21/1972	11 99436-8769
7	10	Mariana	54566266-8	13771913039	R. São Antonio, 232 - Vila Universal, Barueri - SP, 06407-140	03/05/2018	NULO

medico				
idMedico	idUsuario	idClinica	idEspecialidade	nomeMedico
1	1	1	2	Ricardo Lemos
2	2	1	17	Roberto Possarile

Modelo Físico: Modelo que exemplifica com os dados em seus respectivos lugares da tabela, simulando um banco de dados

Cronograma

	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5
Modelo Conceitual	X				
Modelo Lógico	X				
Modelo Físico	X				
Scripts: DDL,DML e DQL		X			
Documentação		X			

5. Trello

Ambiente utilizado para organizar o projeto de acordo com tarefas, etapas, links úteis e a data de realização de cada processo.

[SP MEDICAL GROUP | Trello](#)

6. Back-End

Descrição sobre a arquitetura utilizada para a criação do back-end.

A API (Application Programming Interface) foi desenvolvida através do Visual Studio 2019 Community, famosa IDE da Microsoft que possibilita o desenvolvimento web.

A linguagem utilizada para criação do projeto foi o C#, uma linguagem desenvolvida pela própria Microsoft como parte da plataforma .NET. Como já dito a arquitetura utilizada foi a API, sendo um modelo prático e mais fácil de ser compreendido.

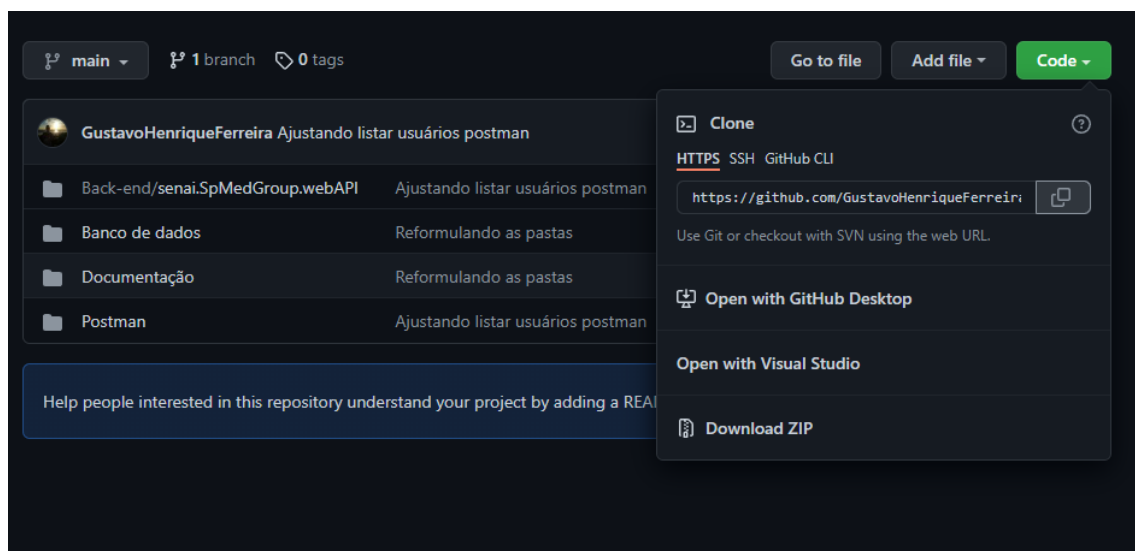
Para as requisições foi utilizado o protocolo HTTP (Hypertext Transfer Protocol), em formato JSON (JavaScript Object Notation), dessa forma é possível transferir dados de forma leve e compacta, não sendo necessário pegar todos os dados disponível para carregar páginas.

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

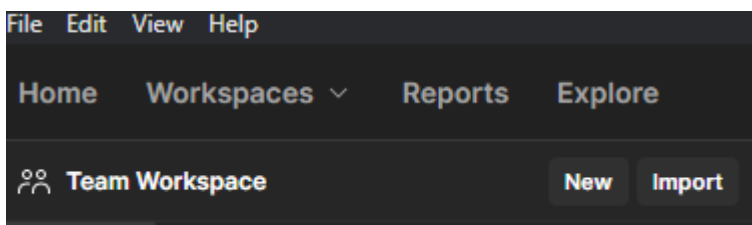
The image shows the Postman website on the left and the Postman desktop application on the right. The website has a navigation bar with links like Product, Pricing, Enterprise, Resources and support, and Explore. Below the navigation bar is a section titled 'Download Postman' with a sub-header 'Download the app to quickly get started using the Postman API Platform. Or, if you prefer a browser experience, you can try the new web version of Postman.' The desktop application interface shows a sidebar with 'Collections' and 'Environments'. The main area displays a 'Single Tweet' collection with a 'GET' request to 'https://api.twitter.com/2/tweets/16'. The 'Params' tab is active, showing a table with columns 'Key', 'Value', and 'description'. The 'Body' tab is also visible, showing a JSON response.

Key	Value	description
tweet.fields	attachments,author_id,context_annotations,conversation_id,media.fields,place.fields,reply_fields,source	Comma-separated list of fields to expand. Expansions e...
expansions	media.fields,place.fields,reply_fields,source	Comma-separated list of fields for the poll object. Exp...
media.fields	duration,ms,highlighted_media_key,non_public_metrics,organ...	Comma-separated list of fields for the poll object. Exp...
place.fields	place.fields	Comma-separated list of fields for the place object. Exp...
reply_fields	reply_fields	Comma-separated list of fields for the user object. Exp...

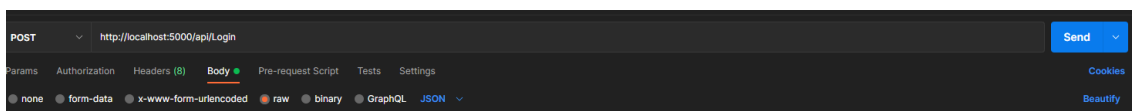
Primeiro Passo: Primeiramente instale a versão desktop do Postman, onde você irá testar as requisições.



Segundo Passo: Logo após o download, use o comando git clone para clonar o repositório, conseguindo assim usar a coleção do Postman.



Terceiro passo: Depois disso clique no botão Import e faça o upload da coleção, que trará junto dela as pastas com as requisições.



Quarto passo: Por fim cliquei no botão send e teste todas as requisições.

Cronograma

	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5
Domains	X				
Interfaces	X				
Repositories		X			
Controllers				X	
ViewModel			X		
Swagger e Startup	X				
Postman					X

7. UX-UI

Descrição sobre o processo de criação de layouts:

O que é UX-UI?

UX (User Experience) é lado do design que se relaciona com as emoções e experiências dos usuários, sendo exclusivo na relação do usuário com o produto ou serviço. O objetivo é oferecer um serviço que garanta uma boa experiência, gerando satisfação e evitando frustrações no usuário, fidelizando o cliente.

UI Design (User Interface) é área do Design que efetua a elaboração e criação do meio que o usuário interage e controla um determinado dispositivo. Esse dispositivo pode ser um sistema, software, produto, serviço, aplicativo, etc. O objetivo principal dessa área é conseguir elaborar uma interface onde o usuário consiga realizar o que ele pode, quer e espera realizar. Tudo isso de modo simples, fácil e intuitivo.

Figma

O Figma foi o software utilizado para a construção das interfaces do projeto, tanto os layouts web e mobile, sendo focado no desenvolvimento de sistemas de design gráfico, prototipagem de interface gráfica de usuário e desenvolvimento de UI/UX.

[SpMedGroup – Figma](#)

8. Front-end

O que é Front-end?

O Front-end é a parte da programação relativa à interface de uma aplicação, através dessas interfaces é possível a interação do usuário com o sistema de forma simples, fácil e interativa. Basicamente é a parte visual do site, aparência visível pelos seus visitantes.

Tecnologias e linguagens

HTML: É uma linguagem de marcação cuja sigla significa Hypertext Markup Language. Basicamente a estrutura do site, um exemplo que se assemelha seria os ossos para um ser humano.

CSS: Para a estilização do site se utiliza o CSS (Cascading Style Sheets), uma linguagem que serve para apenas estilização.

JavaScript: Para o dinamismo e ser possível a interação do usuário com a página se usa o JavaScript. Sem essa linguagem as páginas seriam apenas estáticas e não seria possível se ter alguma interação.

React: É uma biblioteca JavaScript para a criação de interfaces de usuário, possibilitando a criação de UIs interativas de forma fácil e rápida.

9. Mobile

O que é Mobile?

Uma aplicação móvel ou aplicativo mobile é um software desenvolvido para ser instalado em smartphones e iPads.

Na aplicação do SpMedicalGroup foi desenvolvido um sistema com o React Native, uma biblioteca Javascript que auxilia o desenvolvimento de aplicativos para os sistemas Android e iOS de forma nativa.

Android Studio: Ambiente de desenvolvimento integrado (IDE) utilizado na aplicação para desenvolver o aplicativo Android.

10. NoSQL

O que é NoSQL?

Originalmente NoSQL se referia a "não SQL" ou "não relacional", porém atualmente o termo é usado para se referir a Not Only SQL. Resumidamente o NoSQL representa os bancos de dados não relacionais, um banco de dados que fornece o armazenamento e recuperação de dados que são modelados de formas diferentes das relações de tabelas e linhas usadas nos bancos de dados relacionais.

Vantagens do Banco não relacional: flexibilidade, escalabilidade, alta performance e altamente funcional.

Para a aplicação foi utilizado o MongoDB, um software de banco de dados orientado a documentos livre e usa documentos BSON.

11. Sistema Web

Perfis de usuário:

1. **Administrador:** 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, sendo outro administrador, paciente ou médico;
2. O **administrador** poderá cadastrar os dados de uma consulta, como por exemplo a data do agendamento, médico, etc;
3. O **administrador** poderá alterar a situação da consulta, podendo cancelar ela, além disso tem a permissão para mudar a consulta, sendo realizada ou apenas agendada;
4. O **administrador** poderá cadastrar os dados das clínicas, como por exemplo endereço, nome, CNPJ, etc;
5. Qualquer usuário autenticado poderá ver todas as consultas relacionados a ele, sendo paciente ou médico;
6. O médico poderá inserir a descrição da consulta que está associada a um paciente;

Teste do Sistema:

Banco de dados (Banco Relacional)

1ºPasso:

```
senai_spmедgroup_01.S.master (sa (63)) -> X
CREATE DATABASE SENAI_SPMEDGROUP;
GO

USE SENAI_SPMEDGROUP;
GO

CREATE TABLE tipoUsuario(
    idTipoUsuario INT PRIMARY KEY IDENTITY(1,1),
    tituloTipoUsuario VARCHAR(35) NOT NULL,
);
GO

CREATE TABLE usuario(
    idUsuario INT PRIMARY KEY IDENTITY(1,1),
    idTipoUsuario INT FOREIGN KEY REFERENCES tipoUsuario(idTipoUsuario),
    email VARCHAR(100) NOT NULL UNIQUE,
    senha CHAR(20) NOT NULL,
);
GO

CREATE TABLE clinica(
    idClinica INT PRIMARY KEY IDENTITY,
    nomeClinica VARCHAR(100) NOT NULL UNIQUE,
    razaoSocial VARCHAR(200) NOT NULL UNIQUE,
    enderecoClinica VARCHAR(300) NOT NULL UNIQUE,
    cnpj CHAR(18) NOT NULL UNIQUE,
    horarioAberto TIME NOT NULL,
    horarioFechado TIME NOT NULL,
);
GO

CREATE TABLE paciente(
    idPaciente INT PRIMARY KEY IDENTITY,
    idUsuario INT FOREIGN KEY REFERENCES usuario(idUsuario),
    nomePaciente VARCHAR(250) NOT NULL UNIQUE,
    rg CHAR(10) NOT NULL UNIQUE,
    cpf CHAR(11) NOT NULL UNIQUE,
```

Primeiramente é preciso ter instalado o SQL Server Management Studio (SSMS). Depois de instalado e configurado será preciso criar o banco de dados chamado SENAI_SPMEDGROUP, para fazer isso abra o arquivo senai_spmедgroup_01_DDL que está na pasta de banco de dados, depois selecione o comando create database e aperte F5.

2ºPasso:

```

senai_spmedgroup_0...S.master (sa (63))
CREATE DATABASE SENAI_SPMEDGROUP;
GO

USE SENAI_SPMEDGROUP;
GO

CREATE TABLE tipoUsuario(
    idTipoUsuario INT PRIMARY KEY IDENTITY(1,1),
    tituloTipoUsuario VARCHAR(35) NOT NULL,
);
GO

CREATE TABLE usuario(
    idUsuario INT PRIMARY KEY IDENTITY(1,1),
    idTipoUsuario INT FOREIGN KEY REFERENCES tipoUsuario(idTipoUsuario),
    email VARCHAR(100) NOT NULL UNIQUE,
    senha CHAR(20) NOT NULL,
);
GO

CREATE TABLE clinica(
    idClinica INT PRIMARY KEY IDENTITY,
    nomeClinica VARCHAR(100) NOT NULL UNIQUE,
    razaoSocial VARCHAR(200) NOT NULL UNIQUE,
    enderecoClinica VARCHAR(300) NOT NULL UNIQUE,
    cnpj CHAR(18) NOT NULL UNIQUE,
    horarioAberto TIME NOT NULL,
    horarioFechado TIME NOT NULL,
);
GO

CREATE TABLE paciente(
    idPaciente INT PRIMARY KEY IDENTITY,
    idUsuario INT FOREIGN KEY REFERENCES usuario(idUsuario),
    nomePaciente VARCHAR(250) NOT NULL UNIQUE,
    rg CHAR(10) NOT NULL UNIQUE,
    cpf CHAR(11) NOT NULL UNIQUE,
);

```

Selecione o comando use para usar o banco criado.

3ºPasso:

```

senai_spmedgroup_0...S.master (sa (63))
CREATE DATABASE SENAI_SPMEDGROUP;
GO

USE SENAI_SPMEDGROUP;
GO

CREATE TABLE tipoUsuario(
    idTipoUsuario INT PRIMARY KEY IDENTITY(1,1),
    tituloTipoUsuario VARCHAR(35) NOT NULL,
);
GO

CREATE TABLE usuario(
    idUsuario INT PRIMARY KEY IDENTITY(1,1),
    idTipoUsuario INT FOREIGN KEY REFERENCES tipoUsuario(idTipoUsuario),
    email VARCHAR(100) NOT NULL UNIQUE,
    senha CHAR(20) NOT NULL,
);
GO

CREATE TABLE clinica(
    idClinica INT PRIMARY KEY IDENTITY,
    nomeClinica VARCHAR(100) NOT NULL UNIQUE,
    razaoSocial VARCHAR(200) NOT NULL UNIQUE,
    enderecoClinica VARCHAR(300) NOT NULL UNIQUE,
    cnpj CHAR(18) NOT NULL UNIQUE,
    horarioAberto TIME NOT NULL,
    horarioFechado TIME NOT NULL,
);
GO

CREATE TABLE paciente(
    idPaciente INT PRIMARY KEY IDENTITY,
    idUsuario INT FOREIGN KEY REFERENCES usuario(idUsuario),
    nomePaciente VARCHAR(250) NOT NULL UNIQUE,
    rg CHAR(10) NOT NULL UNIQUE,
    cpf CHAR(11) NOT NULL UNIQUE,
);

```

Crie todas as tabelas do mesmo jeito que foi criado o banco de dados e o use. Selecione e crie uma tabela de cada vez para não gerar nenhum conflito.

4º Passo:

```
senai_spmedgroup_0...S.master (sa (52))  senai_spmedgroup_0...S.master (sa (63))
USE SENAI_SPMEDGROUP;
GO

INSERT INTO tipoUsuario(tituloTipoUsuario)
VALUES ('Administrador'),('Paciente'),('Médico');
GO

INSERT INTO usuario(idTipoUsuario, email, senha)
VALUES (3,'ricardo.lemos@spmedicalgroup.com.br','321'),
(3,'roberto.possarle@spmedicalgroup.com.br','1234'),
(3,'helena.souza@spmedicalgroup.com.br','12345'),
(2,'ligia@gmail.com','123456'),
(2,'alexandre@gmail.com','1234567'),
(2,'fernando@gmail.com','12345678'),
(2,'henrique@gmail.com','abcd'),
(2,'joao@hotmail.com','abcd'),
(2,'bruno@gmail.com','abcde'),
(2,'mariana@outlook.com','abcdef'),
(1,'saulo@gmail.com','C# melhor que python');
GO

INSERT INTO clinica(nomeClinica, razaoSocial, enderecoClinica, cnpj, HorarioAberto, HorarioFechado)
VALUES ('Clínica Possarle', 'SP Medical Group', 'Av. Barão Limeira, 532, São Paulo, SP', '86.400.902/0001-30', '07:00:00', '23:00:00'),
('Clínica Portuguesa', 'Clínica Portugal', 'Av. Portugal, 1676, Santo André, SP', '99.999.999/9999-99', '08:00:00', '22:00:00');
GO

INSERT INTO paciente(idUsuario, nomePaciente, rg, cpf, enderecoPaciente, nascimento, telefone)
VALUES (4, 'Ligia', '43522543-5', '94839859000', 'Rua Estado de Israel 240, São Paulo, Estado de São Paulo', '10/12/1983', '11 3456-7654'),
(5, 'Alexandre', '32654345-7', '73556944057', 'Av. Paulista, 1578 - Bela Vista, São Paulo - SP', '7/12/2001', '11 98765-6543'),
(6, 'Fernando', '54636525-3', '16839338002', 'Av. Ibirapuera - Indianópolis, 2927, São Paulo - SP', '10/10/1978', '11 97208-4453'),
(7, 'Henrique', '54366362-5', '14332654765', 'R. Vitória, 120 - Vila São Jorge, Barueri - SP', '10/12/1985', '11 3456-6543'),
(8, 'João', '53254444-1', '91305348010', 'R. Ver. Geraldo de Camargo, 66 - Santa Luzia, Ribeirão Pires - SP', '8/9/1975', '11 7656-6377'),
(9, 'Bruno', '54566266-7', '79799299004', 'Alameda dos Arapuanês, 945 - Indianópolis, São Paulo - SP', '3/11/1972', '11 95436-8769'),
(10, 'Mariana', '54566266-8', '13771913039', 'R. São Antonio, 232 - Vila Universal, Barueri - SP', '3/5/2018', '');
GO

INSERT INTO especialidade(nomeEspecialidade)
```

Abra o arquivo DML que está na mesma pasta e selecione o use.

5º Passo:

```
senai_spmedgroup_0...S.master (sa (52))  senai_spmedgroup_0...S.master (sa (63))
USE SENAI_SPMEDGROUP;
GO

INSERT INTO tipoUsuario(tituloTipoUsuario)
VALUES ('Administrador'),('Paciente'),('Médico');
GO

INSERT INTO usuario(idTipoUsuario, email, senha)
VALUES (3,'ricardo.lemos@spmedicalgroup.com.br','321'),
(3,'roberto.possarle@spmedicalgroup.com.br','1234'),
(3,'helena.souza@spmedicalgroup.com.br','12345'),
(2,'ligia@gmail.com','123456'),
(2,'alexandre@gmail.com','1234567'),
(2,'fernando@gmail.com','12345678'),
(2,'henrique@gmail.com','abcd'),
(2,'joao@hotmail.com','abcd'),
(2,'bruno@gmail.com','abcde'),
(2,'mariana@outlook.com','abcdef'),
(1,'saulo@gmail.com','C# melhor que python');
GO

INSERT INTO clinica(nomeClinica, razaoSocial, enderecoClinica, cnpj, HorarioAberto, HorarioFechado)
VALUES ('Clínica Possarle', 'SP Medical Group', 'Av. Barão Limeira, 532, São Paulo, SP', '86.400.902/0001-30', '07:00:00', '23:00:00'),
('Clínica Portuguesa', 'Clínica Portugal', 'Av. Portugal, 1676, Santo André, SP', '99.999.999/9999-99', '08:00:00', '22:00:00');
GO

INSERT INTO paciente(idUsuario, nomePaciente, rg, cpf, enderecoPaciente, nascimento, telefone)
VALUES (4, 'Ligia', '43522543-5', '94839859000', 'Rua Estado de Israel 240, São Paulo, Estado de São Paulo', '10/12/1983', '11 3456-7654'),
(5, 'Alexandre', '32654345-7', '73556944057', 'Av. Paulista, 1578 - Bela Vista, São Paulo - SP', '7/12/2001', '11 98765-6543'),
(6, 'Fernando', '54636525-3', '16839338002', 'Av. Ibirapuera - Indianópolis, 2927, São Paulo - SP', '10/10/1978', '11 97208-4453'),
(7, 'Henrique', '54366362-5', '14332654765', 'R. Vitória, 120 - Vila São Jorge, Barueri - SP', '10/12/1985', '11 3456-6543'),
(8, 'João', '53254444-1', '91305348010', 'R. Ver. Geraldo de Camargo, 66 - Santa Luzia, Ribeirão Pires - SP', '8/9/1975', '11 7656-6377'),
(9, 'Bruno', '54566266-7', '79799299004', 'Alameda dos Arapuanês, 945 - Indianópolis, São Paulo - SP', '3/11/1972', '11 95436-8769'),
(10, 'Mariana', '54566266-8', '13771913039', 'R. São Antonio, 232 - Vila Universal, Barueri - SP', '3/5/2018', '');
GO

INSERT INTO especialidade(nomeEspecialidade)
```

Selecione o comando insert que insira os dados na tabela, faça igual as tabelas e realize um insert de cada vez.

6º Passo:

```

USE SENAI_SPMEDGROUP;
GO

-- Listar todas as entidades
SELECT * FROM tipoUsuario;
GO

SELECT * FROM usuario;
GO

SELECT * FROM clinica;
GO

SELECT * FROM paciente;
GO

SELECT * FROM especialidade;
GO

```

	idUsuario	idTipoUsuario	email	senha
1	1	3	ricardo.lemos@spmedicalgroup.com.br	321
2	2	3	roberto.possarle@spmedicalgroup.com.br	1234
3	3	3	helenasouza@spmedicalgroup.com.br	12345
4	4	2	ligia@gmail.com	123456
5	5	2	alexandre@gmail.com	1234567
6	6	2	fernando@gmail.com	12345678
7	7	2	henrique@gmail.com	abc
8	8	2	joao@hotmail.com	abcd
9	9	2	bruno@gmail.com	abcde
10	10	2	mariana@outlook.com	abcdef
11	11	1	saulo@gmail.com	C# melhor que python

Abra o arquivo DQL, use o banco SENAI_SPMEDGROUP e use o comando select a fim de ter certeza de que as tabelas estão criadas com os dados corretos.

Banco de Dados (NoSQL)

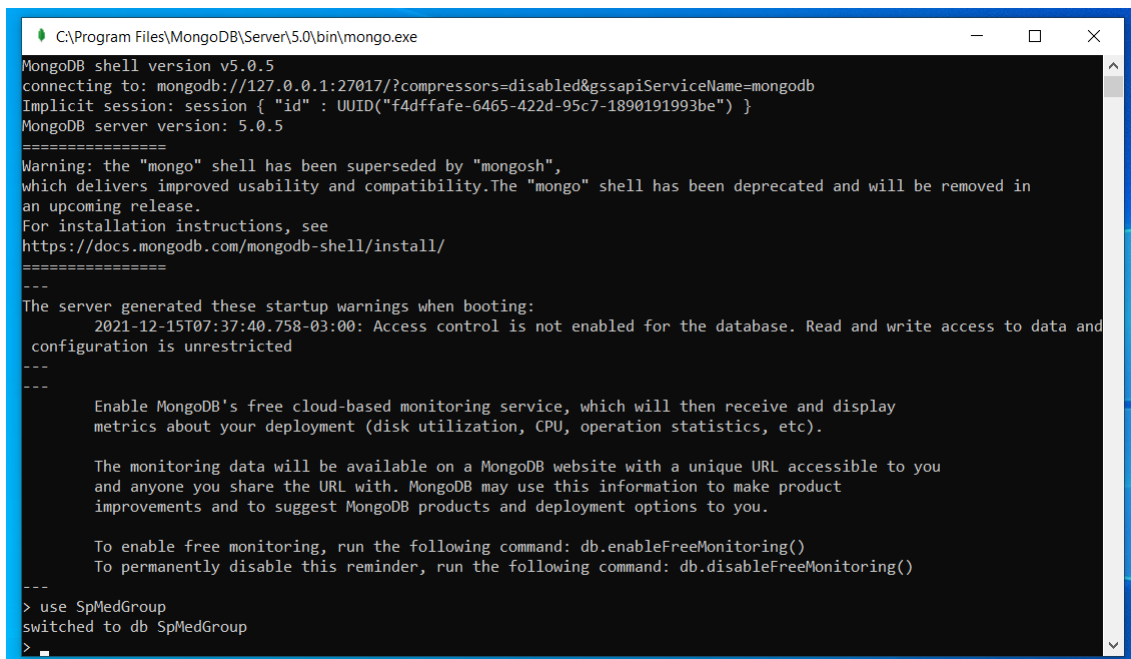
1º Passo:

Disco Local (C:) > Arquivos de Programas > MongoDB > Server > 5.0 > bin

Nome	Data de modificação	Tipo	Tamanho
InstallCompass.ps1	02/12/2021 15:35	Script do Window...	2 KB
mongo.exe	02/12/2021 17:07	Aplicativo	21.714 KB
mongod.cfg	09/12/2021 16:27	Arquivo CFG	1 KB
mongod.exe	02/12/2021 17:07	Aplicativo	46.241 KB
mongod.pdb	02/12/2021 17:07	Program Debug D...	525.068 KB
mongos.exe	02/12/2021 16:26	Aplicativo	29.185 KB
mongos.pdb	02/12/2021 16:26	Program Debug D...	306.892 KB

Necessário a instalação do MongoDB e MongoDB Community Server. Após instalado abra o mongo.exe na pasta Program Files\MongoDB\Server\5.0\bin.

2º Passo:



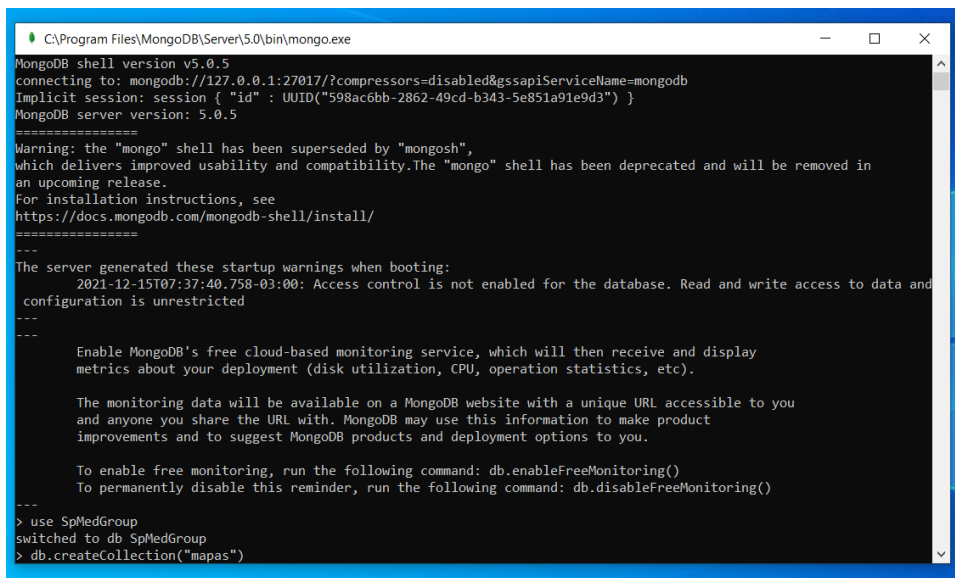
```
C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
MongoDB shell version v5.0.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("f4dffa9e-6465-422d-95c7-1890191993be") }
MongoDB server version: 5.0.5
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2021-12-15T07:37:40.758-03:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use SpMedGroup
switched to db SpMedGroup
>
```

Depois de aberto o terminal, digite use SpMedGroup no terminal para criar o banco de dados.

3º Passo:



```
C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
MongoDB shell version v5.0.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("598ac6bb-2862-49cd-b343-5e851a91e9d3") }
MongoDB server version: 5.0.5
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2021-12-15T07:37:40.758-03:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

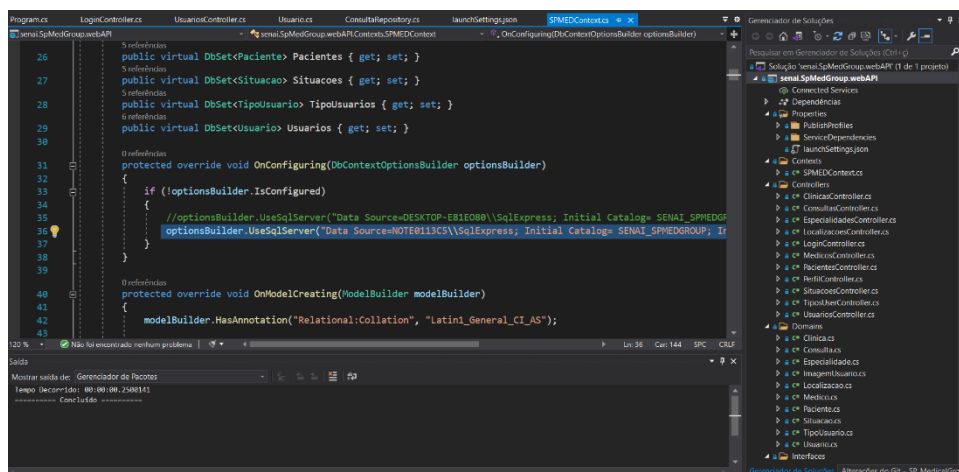
  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use SpMedGroup
switched to db SpMedGroup
> db.createCollection("mapas")
```

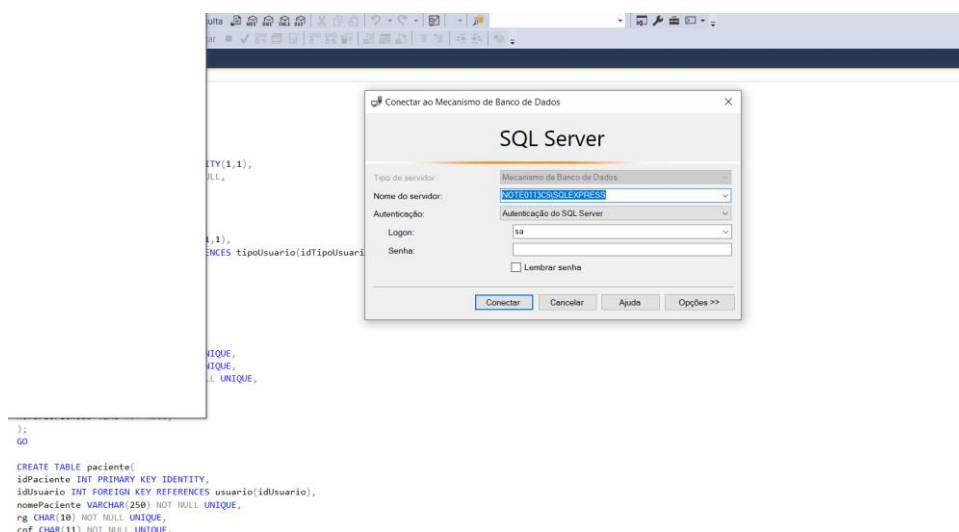
Digite o comando `db.createCollection("Mapas")`, para criar de uma coleção no banco.

API

1ºPasso:

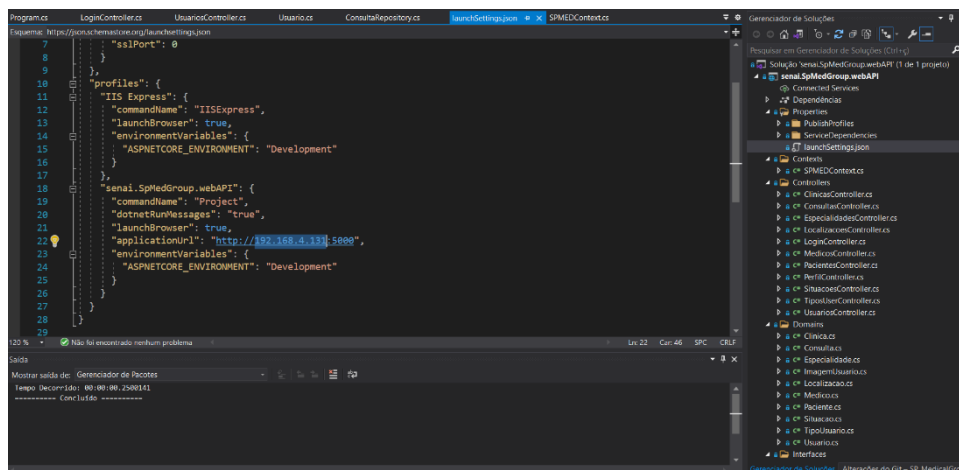


É necessário se ter instalado Microsot Visual Studio Community. Depois de instalado e configurado, abra o arquivo senai.SpMedGroup.webAPI.sln que está na pasta Back-end\senai.SpMedGroup.webAPI. No Visual Studio abra o arquivo SPMEDcontext.cs localizado na pasta contexts e mude o Data Source na optionsBuilder.

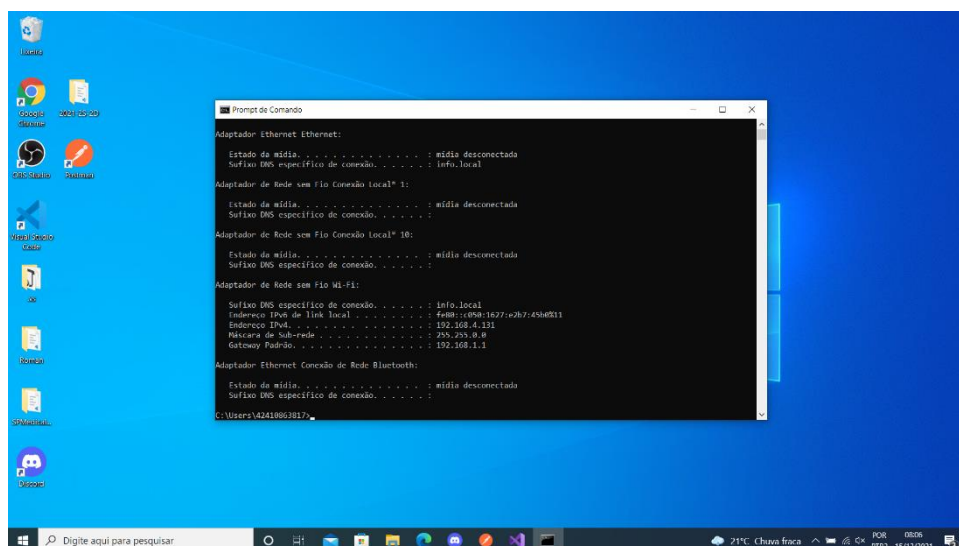


É possível ver qual é o Data Source no SSMS, assim que se abre o aplicativo há uma tela para se conecta, o nome do servidor é o Data Source. Não esqueça de colocar duas barras no Visual Studio antes do SQLEXPRESS, pois no SSMS há apenas uma barra.

2º Passo:

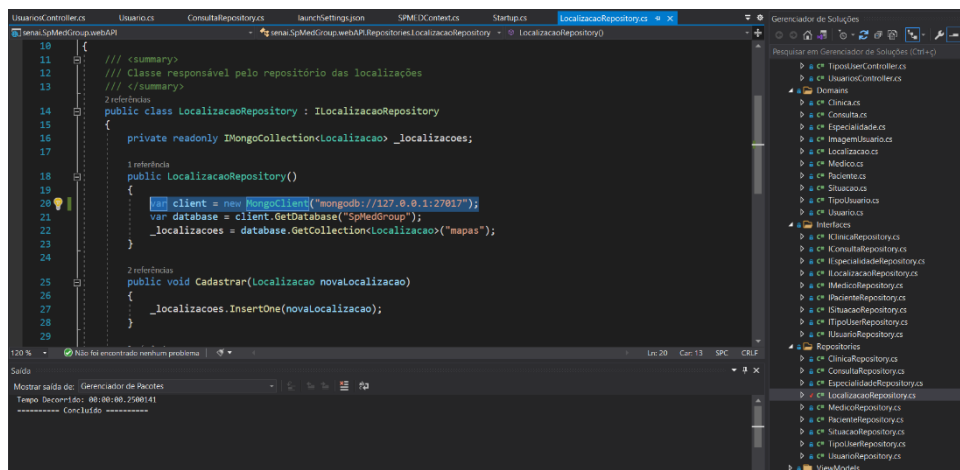


Abra o arquivo launchSettings na pasta Properties, na linha 22 mude o valor depois do http://, sendo que esse valor é IP da sua máquina.

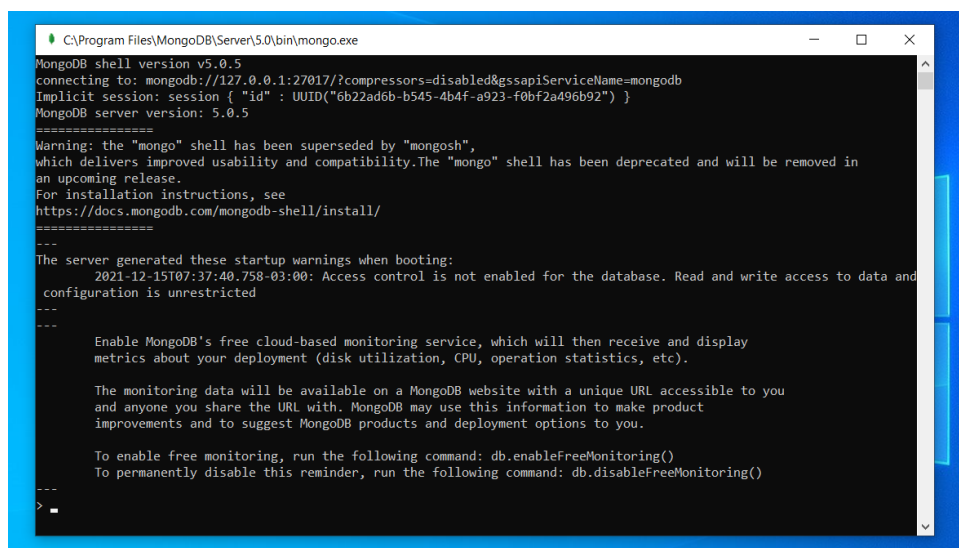
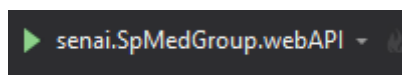


Para verificar o Ip abra o prompt de comando e digite ipconfig, o Ipv4 será o valor utilizado no launchSettings.

3º Passo:



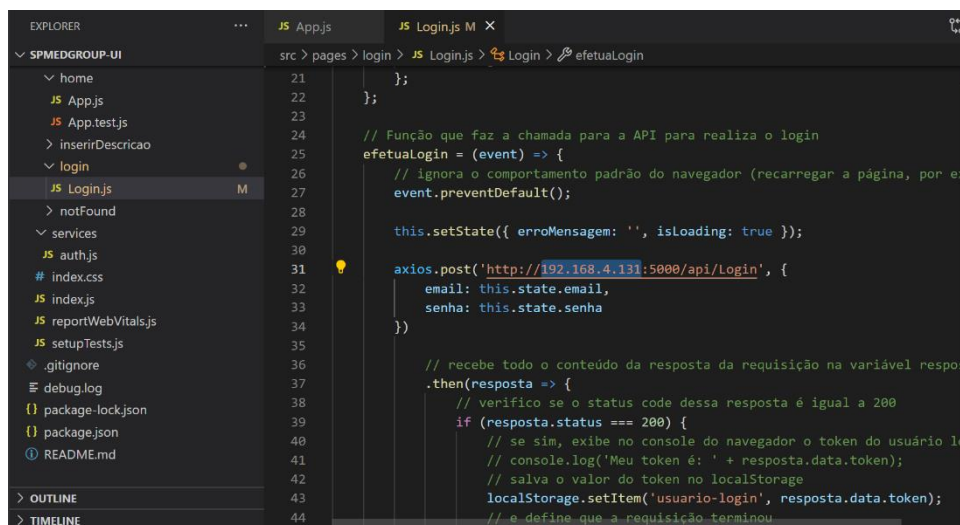
Por fim verifique se o servidor do mongodb, database e a coleção estão corretas e iguais a sua na linha 20 a 23, do arquivo LocalizacaoRepository na pasta Repositories. Depois disso inicie a API, clicando no botão verde igual a imagem abaixo.



É possível verificar o servidor no terminal do MongoDB na segunda linha.

Front-End

1º Passo:

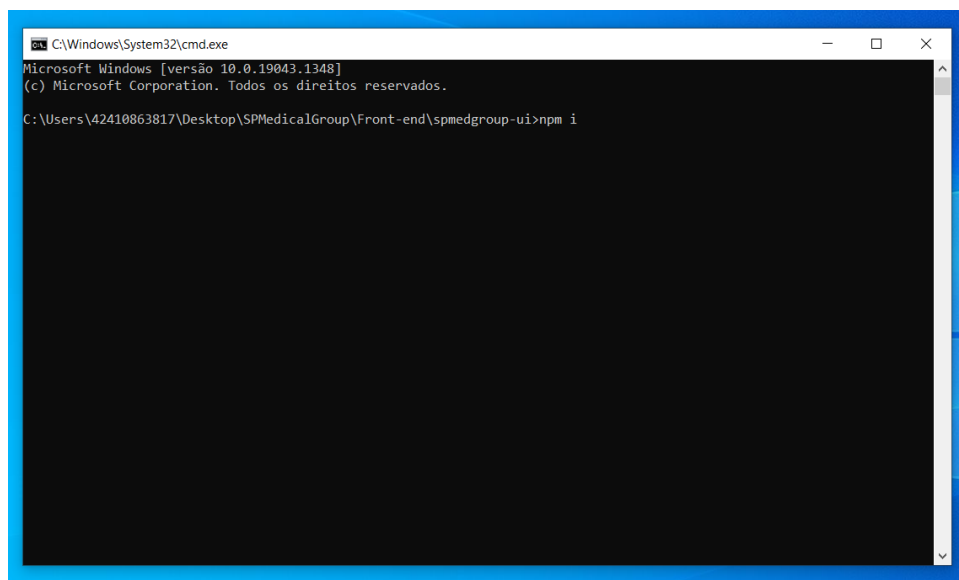


```

21  };
22  };
23
24  // Função que faz a chamada para a API para realiza o login
25  efetuaLogin = (event) => {
26    // ignora o comportamento padrão do navegador (recarregar a página, por ex
27    event.preventDefault();
28
29    this.setState({ erroMensagem: '', isLoading: true });
30
31    axios.post('http://192.168.4.131:5000/api/Login', {
32      email: this.state.email,
33      senha: this.state.senha
34    })
35
36    // recebe todo o conteúdo da resposta da requisição na variável respos
37    .then(resposta => {
38      // verifico se o status code dessa resposta é igual a 200
39      if (resposta.status === 200) {
40        // se sim, exibe no console do navegador o token do usuário lc
41        // console.log('Meu token é: ' + resposta.data.token);
42        // salva o valor do token no localStorage
43        localStorage.setItem('usuario-login', resposta.data.token);
44        // e define que a requisição terminou
  
```

Abra a pasta SPMEDGROUP-UI no front-end e logo após abra o login.js, localizado no spmedgroup-ui\src\pages\login. Depois disso mudei a coloque o IP da sua máquina igual na API. É preciso mudar o IP em todas as requisições que queira testar nas outras páginas também.

2º Passo:



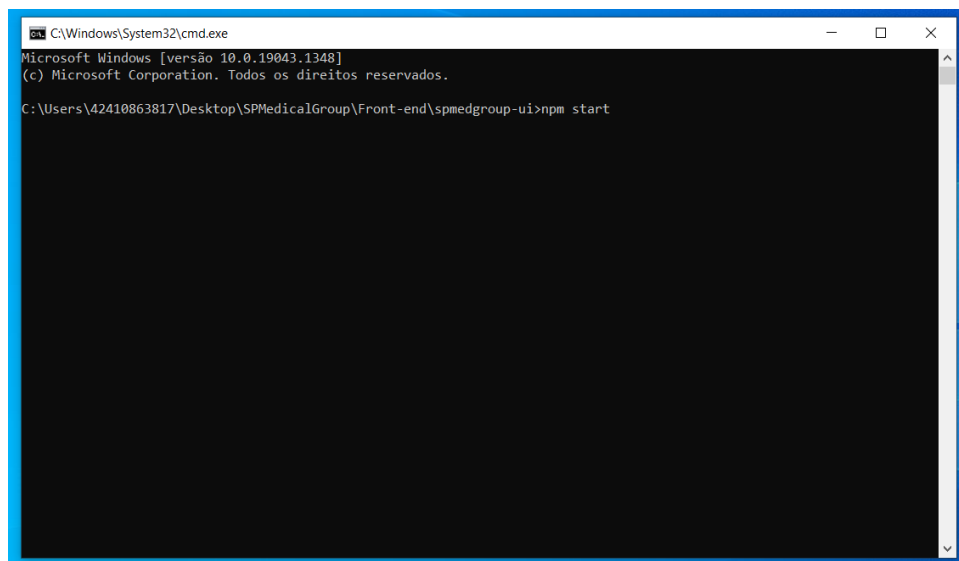
```

C:\Windows\System32\cmd.exe
Microsoft Windows [versão 10.0.19043.1348]
(c) Microsoft Corporation. Todos os direitos reservados.

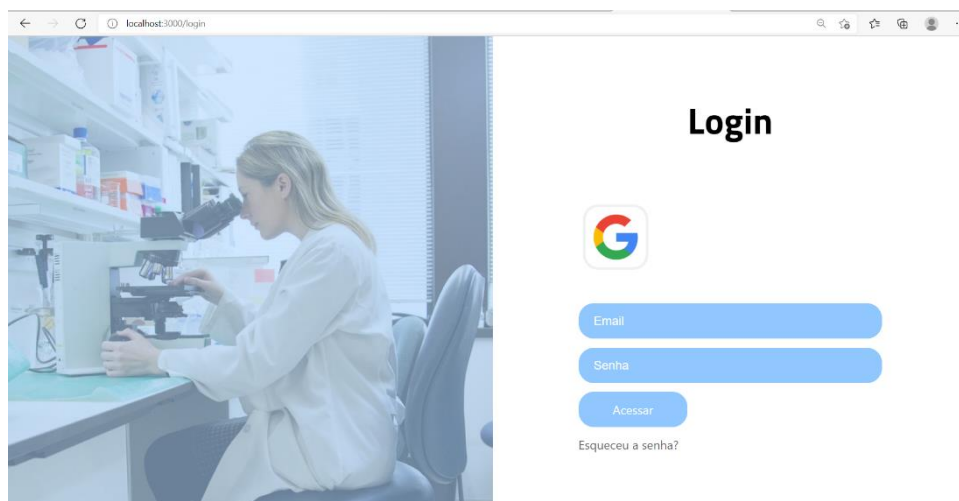
C:\Users\42410863817\Desktop\SPMedicalGroup\Front-end\spmedgroup-ui>npm i
  
```

Abra o prompt de comando na pasta ou o terminal do Visual Studio e digite npm i para instalação dos pacotes necessários.

3º Passo:



Digite `npm start` logo após a conclusão da instalação dos pacotes, o projeto será iniciado!



OBSERVAÇÃO: NECESSÁRIO SE TER O BANCO DE DADOS NA MÁQUINA E A API EM FUNCIONAMENTO.

12. Sistema Mobile

Perfis de usuário:

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

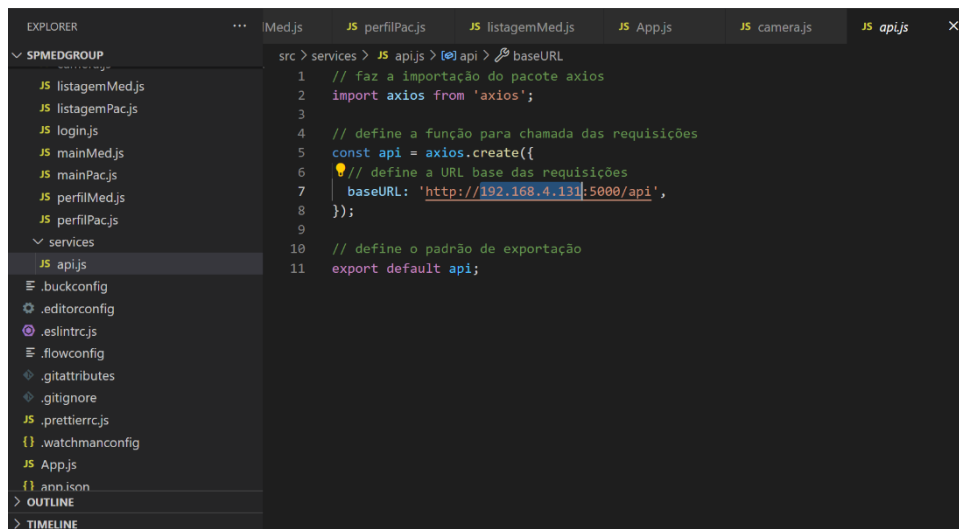
Funcionalidades:

1. Qualquer usuário autenticado poderá ver todas as clínicas;
2. Somente o **administrador** poderá agendar a consulta para o paciente;

3. O usuário **paciente** poderá ver as consultas associados a ele;
4. O usuário **médico** poderá ver as consultas associados a ele;

Teste do sistema:

1ºPasso:

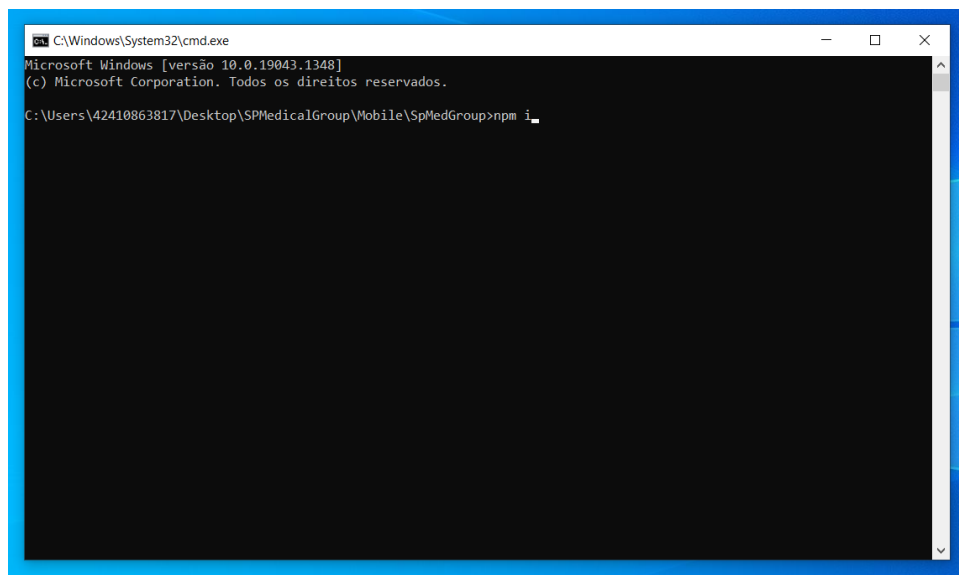


The screenshot shows a code editor with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with the following files: JS listagemMed.js, JS listagemPac.js, JS login.js, JS mainMed.js, JS mainPac.js, JS perfilMed.js, JS perfilPac.js, JS api.js, .buckconfig, .editorconfig, .eslintrc.js, .flowconfig, .gitattributes, .gitignore, .prettierrc.js, .watchmanconfig, App.js, .ann.json, OUTLINE, and TIMELINE. The code editor shows the content of api.js:

```
src > services > JS apijs > [0] api > baseUrl
1 // faz a importação do pacote axios
2 import axios from 'axios';
3
4 // define a função para chamada das requisições
5 const api = axios.create({
6   // define a URL base das requisições
7   baseUrl: 'http://192.168.4.131:5000/api',
8 });
9
10 // define o padrão de exportação
11 export default api;
```

Abra o arquivo api.js localizado no SPMedicalGroup\Mobile\SpMedGroup\src\services e coloque o IP da sua máquina. Caso você não saiba o IP leia o 2ºpasso da API no Sistema Web explicando como é possível ver o IP.

2ºPasso:



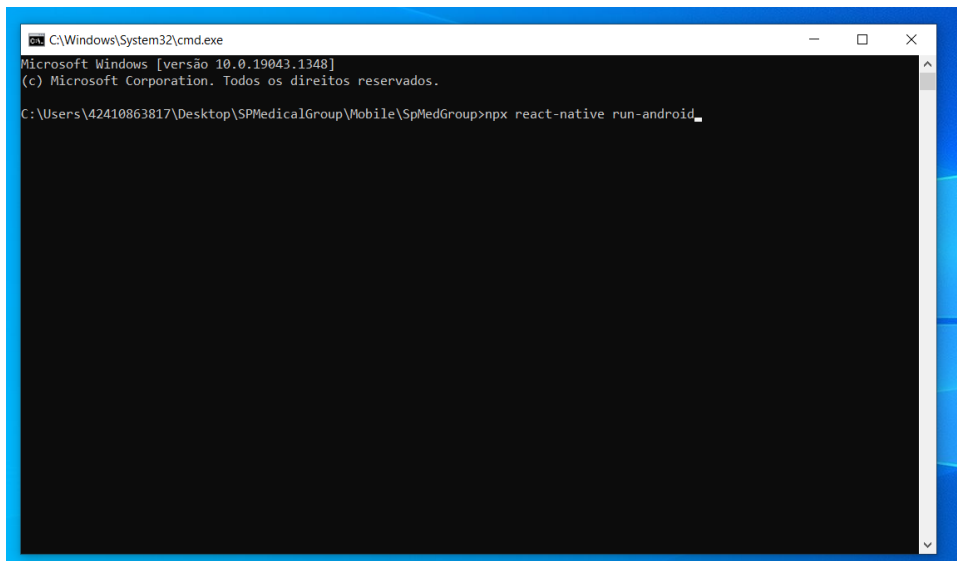
The screenshot shows a Windows command prompt window with the following text:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [versão 10.0.19043.1348]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\42410863817\Desktop\SPMedicalGroup\Mobile\SpMedGroup>npm i
```

Abra o prompt de comando na pasta e digite npm i para instalação dos pacotes necessários.

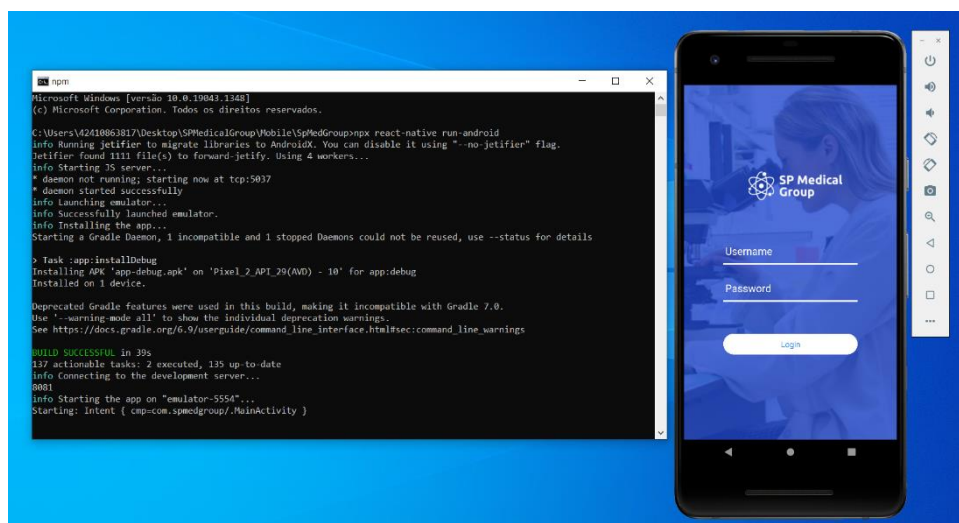
3º Passo:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [versão 10.0.19043.1348]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\42410863817\Desktop\SPMedicalGroup\Mobile\SpMedGroup>npx react-native run-android_
```

Digite `npx react-native run-android` logo após a conclusão da instalação dos pacotes, o projeto será iniciado!



OBSERVAÇÃO: NECESSÁRIO SE TER O BANCO DE DADOS NA MÁQUINA E A API EM FUNCIONAMENTO.