

Documentação

Desenvolvido por:

Gabriele Viveiros de Brito

Sumário

1.	Resumo.....	3
2.	Descrição do projeto	3
2.1.	Sistema Web	3
2.1.1.	Perfis de Usuário	3
2.1.2.	Funcionalidades	3
2.2.	Sistema Mobile.....	4
2.2.1.	Perfis de Usuário	4
2.2.2.	Funcionalidades	4
3.	Modelagem de software	5
3.1.	Modelo Conceitual	5
3.2.	Modelo Lógico.....	6
3.3.	Modelo Físico	6
3.4.	Cronograma.....	7
4.	Back-End.....	7
4.1.	Como executar e testar a API?	7
4.1.1.	Passo-a-passo.....	7
4.2.	Cronograma.....	8
5.	UI / UX	8
5.1.1.	Imagens do layout.....	8
5.2.	Cronograma.....	9
6.	Front-end	10
6.1.1.	Imagens da aplicação na web	Erro! Indicador não definido.
6.2.	Cronograma.....	10
7.	Mobile	11
7.1.1.	Imagens da aplicação mobile	Erro! Indicador não definido.
7.2.	Cronograma.....	11
8.	Referências.....	12

1. Resumo

Uma nova clínica médica chamada SP Medical deseja criar um sistema web/mobile integrado onde seja possível realizar a gestão da clínica de forma automatizada e ter acesso fácil aos dados sobre as informações de seus pacientes. Partindo dessa premissa, o presente documento tem como objetivo expor como foi desenvolvido o banco de dados desta clínica médica.

2. Descrição do projeto

A clínica médica SP Medical Group, é uma empresa de pequeno porte que atua no ramo da saúde, e 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, gastroenterologia etc.). Por ser uma empresa nova, iniciou a administração da clínica de forma simples utilizando softwares de planilhas eletrônicas e, com o sucesso da clínica, sua gestão passou a se tornar complicada devida a alta demanda dos pacientes.

Portanto, Fernando deseja que seja desenvolvido um sistema web/mobile integrado onde seja possível realizar a gestão da clínica de forma automatizada e ter acesso fácil aos dados sobre as informações de seus pacientes.

Com os seguintes requisitos:

2.1. Sistema Web

2.1.1. Perfis de Usuário

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

2.1.2. 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.

2.2. Sistema Mobile

2.2.1. Perfis de Usuário

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

2.2.2. Funcionalidades

1. O paciente poderá visualizar suas próprias consultas;
2. O médico poderá ver as consultas (os agendamentos) associados a ele.

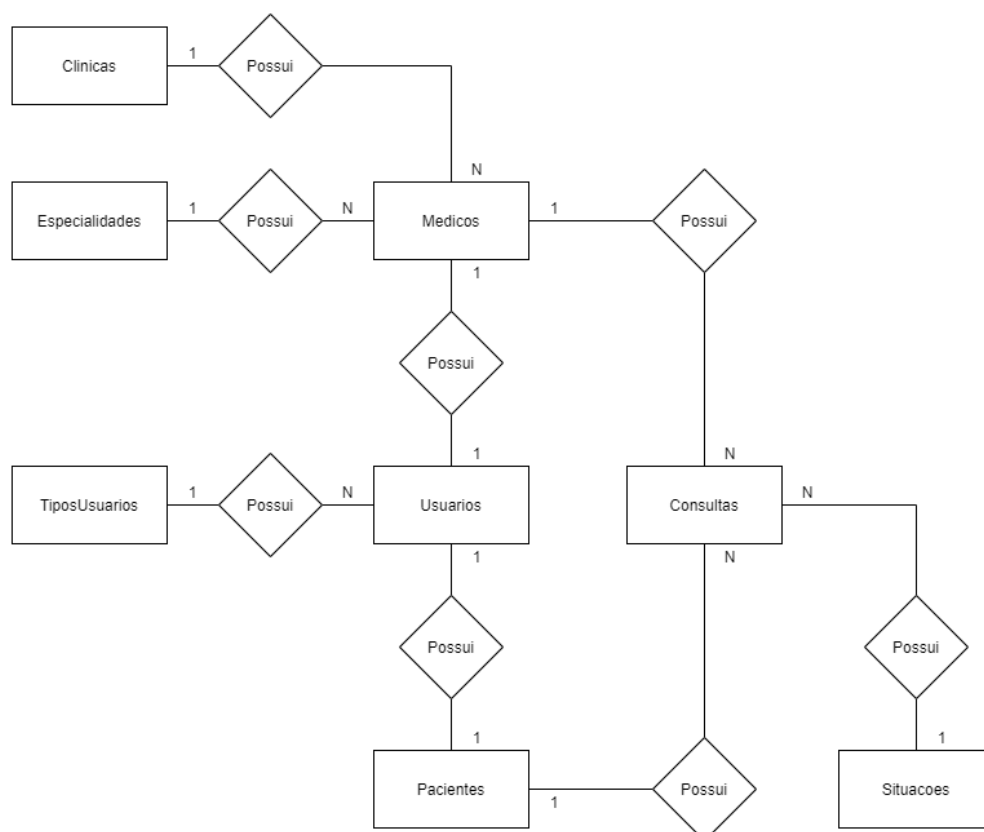
3. Modelagem de software

O banco de dados relacional é uma coleção de relações, ou seja, são tabelas bidimensionais, onde os dados são armazenados. Como exemplo, podemos querer armazenar dados sobre os clientes de uma loja. Para isso, criamos tabelas que guardam diferentes conjuntos de dados relacionados a esses clientes, como dados pessoais, dados de compras, dados de crédito, etc. Cada uma dessas tabelas é uma relação do banco.

A utilização de um banco de dados em uma melhora a organização, e facilita a gestão e o processamento dos dados. Isso acontece porque os dados são organizados de modo a modelar aspectos o mundo real, gerando informações relevantes para os usuários a partir desses dados.

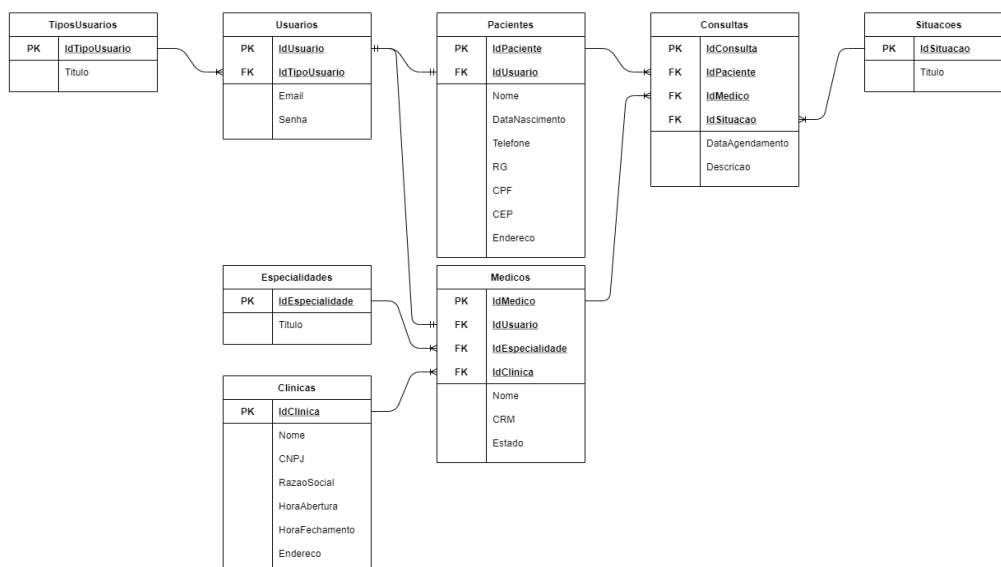
3.1. Modelo Conceitual

O modelo conceitual é um conjunto de suposições baseadas no mundo real que indicarão as regras de negócio de um sistema. Portanto, modelo conceitual é a descrição do sistema proposto na forma de um conjunto de ideias, por isso a participação do usuário final, ou cliente, é fundamental.



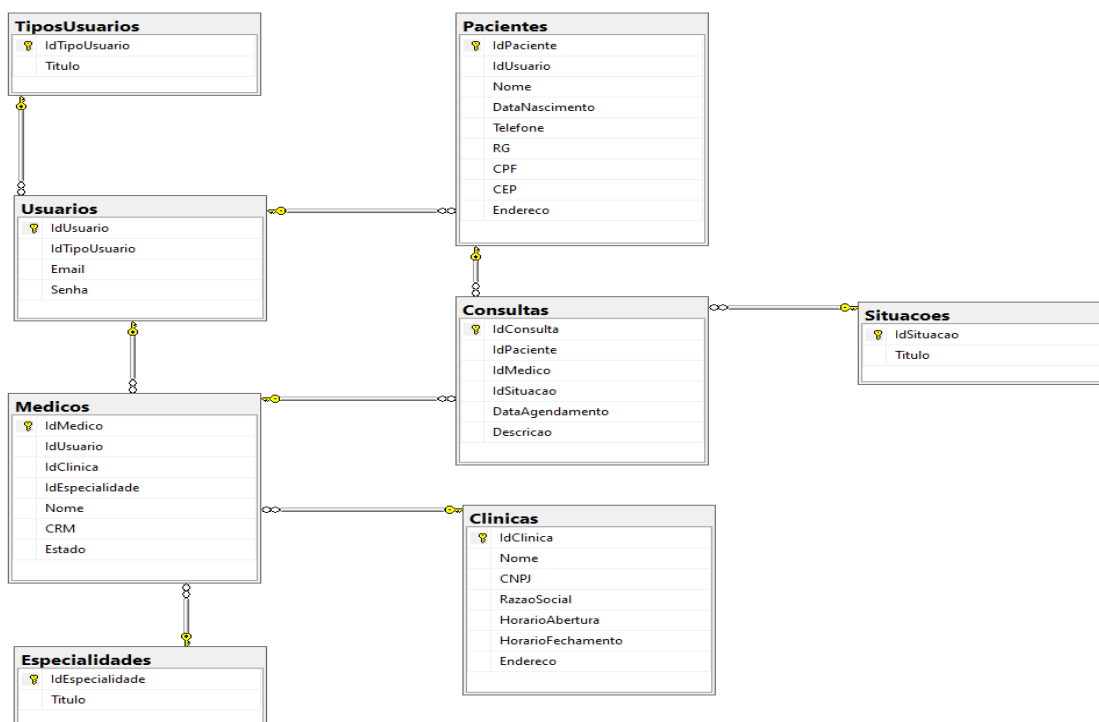
3.2. Modelo Lógico

Este modelo adequa o modelo conceitual para o tipo de banco de dados que será implementado. É desenvolvido na fase de projeto e não é de interesse dos usuários, somente da equipe de desenvolvimento. Nesse modelo que os atributos e as relações entre ficam mais evidentes.



3.3. Modelo Físico

Este modelo é o projeto físico para implementação do banco de dados. Tem todas as preocupações com desempenho, com a capacidade dos equipamentos etc. Pode-se utilizar o Excel para simular a estrutura real do banco de dados com base no modelo lógico elaborado anteriormente.



3.4. Cronograma

	Dia 1	Dia 2	Dia 3
Trello	X		
Modelo Conceitual	X		
Modelo Lógico	X		
Modelo Físico	X		
Criação do banco de dados		X	
DDL		X	
DML		X	
DQL			X
Documentação			X

Link de acesso ao Trello: “<https://trello.com/b/k0tH6ygk/spmedicalgroupmodelagem>”

4. Back-End

O Back-End da aplicação, ou seja, o código do sistema, foi desenvolvido no Microsoft Visual Studio 2019 na linguagem C#. É uma API (Application Programming Interface) com arquitetura REST (Representational State Transfer), que é um conjunto de normas que possibilita a comunicação entre plataformas através de uma série de padrões e protocolos. As requisições e respostas seguem o protocolo HTTP (HyperText Transfer Protocol) utilizando formato JSON (JavaScript Object Notation) para intercâmbio de dados, possibilitando o acesso do sistema em diversos dispositivos.

4.1. Como executar e testar a API?

São necessários os seguintes programas:

- Microsoft SQL Server;
- Microsoft SQL Management Studio;
- Microsoft Visual Studio;
- Postman.

4.1.1. Passo-a-passo

1. Executar os Scripts no *Microsoft SQL Management Studio* para que as entidades sejam criadas no banco de dados;
2. Abrir a solução (SPMedicalGroup_WebAPI.sln) no *Microsoft Visual Studio*;
3. Dentro da pasta ‘Contexts’ no arquivo ‘SPMedicalGroupContext.cs’ alterar as informações de acesso ao banco de dados na linha 35 de acordo com suas credenciais de acesso;
4. Importar a coleção ‘Senai_SPMedicalGroup_WebAPI.postman_collection’ no *Postman*;

5. Executar a API pelo *Microsoft Visual Studio*;
6. Testar as requisições e respostas.

4.2. Cronograma

	Dia 1	Dia 2	Dia 3
Trello	X		
Criar o projeto da API	X		
Pacotes NuGet	X		
Domains e Context	X		
Configuração Startup	X		
Interfaces	X		
Repositories		X	
Controllers		X	
Coleção Postman			X
Documentação			X

Link de acesso ao Trello: “<https://trello.com/b/lfd9wV1X/spmedicalgroupapi>”

5. UI / UX

O Layout da aplicação, tanto para web quanto para dispositivos móveis, foi desenvolvido no Figma. Para a criação do layout da aplicação foram feitos wireframes, tanto de baixa fidelidade quanto de alta fidelidade, e prototipação de todas as páginas, para saber se a interface e a experiência do usuário tivessem qualidade.

5.1.1. Imagens do layout

Web



Adm.png



Cadastrar.png



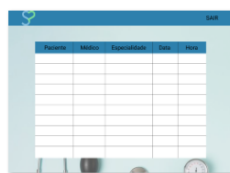
Login.png



Login_Alta.png



Médico.png



Paciente.png

5.2. Cronograma

	Dia 1	Dia 2	Dia 3
Trello	X		
Baixa fidelidade (web)	X		
Alta fidelidade (web)		X	
Alta fidelidade (web)		X	
Baixa fidelidade (mobile)			X
Alta fidelidade (mobile)			X
Documentação			X

Link de acesso ao Trello: “<https://trello.com/b/WVbL4rwv/spmedicalgroupuiux>”

6. Front-end

O Front-end da aplicação, ou seja, toda a interface utilizada pelo usuário final, foi desenvolvido no Microsoft Visual Studio Code 2019 utilizando a biblioteca React Js que utiliza o conceito de SPA (Single Page Application). Isso significa que a aplicação toda tem apenas uma página, e apenas os elementos exibidos na tela (componentes) são alterados, trazendo muito mais performance e uma melhor experiência para o usuário final.

6.1. Cronograma

	Dia 1	Dia 2	Dia 3
Trello	X		
Página de Login	X		
Página ADM	X	X	
Página Cadastrar		X	
Página Médico		X	X
Página Paciente			X

Link de acesso ao Trello: “<https://trello.com/b/AI8OH3S/spmedicalgroupfront>”

7. Mobile

O Front-End da aplicação para dispositivos móveis, ou seja, toda a interface utilizada pelo usuário final no celular, foi desenvolvido no Microsoft Visual Studio Code 2019 utilizando o framework React Native, que possibilita criar aplicações nativas tanto para Android quanto para iOS, trazendo facilidade no desenvolvimento e uma boa experiência ao usuário final.

7.1. Cronograma

	Dia 1	Dia 2	Dia 3
Trello	X		
Navegação	X		
Tela Login	X		
Tela Histórico de consultas	X		
Conexão com a API		X	X
Documentação			X

Link de acesso ao Trello: “<https://trello.com/b/NWfgIF1o/spmedicalgroupmobile>”

8. Referências

Disponível em: <https://www.fiveacts.com.br/modelagem-de-dados/>. Acesso em: 17 ago. 2021.

Disponível em: <https://www.oracle.com/br/database/what-is-a-relational-database/> . Acesso em: 27 ago. 2021.

Disponível em: <https://www.techtudo.com.br/listas/2020/06/o-que-e-api-e-para-que-serve-cinco-perguntas-e-respostas.ghtml> . Acesso em: 15 set. 2021.

Disponível em: <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces> . Acesso em: 24 set. 2021.