**Gerenciamento de consultórios**

**Objetivo**

Criar uma solução para gerenciar um consultório médico com as seguintes propostas abaixo:

* Deve ter um cadastro de médicos com as seguintes informações do médico:

- ○ CRM (Texto, 10 dígitos);

- ○ Nome do médico (Texto, 100 dígitos);

- ○ Telefone (Texto, 20 dígitos);

- ○ Valor da consulta (Decimal com duas casas após a vírgula).

* Deve ter um cadastro de consultórios com as seguintes informações:

- ○ Nome do consultório (Texto, 100 dígitos);

- ○ Endereço do consultório (Texto, 200 dígitos);

- ○ Telefone (Texto, 20 dígitos).

* Deve ter um meio para realizar o vínculo Médico/Consultório: é uma forma de ligar um consultório a vários médicos, pode ser uma rotina/página separada ou pode ser um recurso embutido no cadastro de consultórios.
* Deve ter uma tela inicial com menu lateral ou superior para acesso às rotinas da aplicação;
* Um médico pode estar vinculado a no máximo 2 consultórios;
* Não é permitido cadastrar consultórios sem endereço e telefone;
* Não é permitido cadastrar médicos sem CRM;
* A aplicação precisa persistir os dados em um banco de dados MYSQL.

**Infraestrutura banco de dados**

Precisa existir uma tabela com os dados de usuário com permissões para interagir com o sistema, não será feito nesse projeto, mas poderia ser implementado junto um controle de acesso com base no tipo do usuário. A tabela CadUsuarios irá conter os seguintes campos:

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Tamanho** |
| Pk | int primary key | 8 |
| Nome | varchar | 100 |
| Usuario | varchar | 30 |
| Senha | varchar | 15 |
| TipoUsuario | Int | 8 |

Será criado uma tabela com as informações do consultório cujo nome será: CadConsultorio e nela irá conter os seguintes campos:

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Tamanho** |
| Pk | int primary key | 8 |
| Nome | varchar | 100 |
| Endereco | varchar | 200 |
| Telefone | varchar | 20 |

Também será criado uma tabela com as informações dos médicos e seu nome será: CadMedicos e nela irá conter os seguintes campos:

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Tamanho** |
| Pk | int primary key | 8 |
| CRM | varchar | 10 |
| Nome | varchar | 100 |
| Telefone | varchar | 20 |
| ValorConsulta | numeric | 18,2 |

Precisa ser gerenciado a quais consultórios os médicos estão vinculados e cada médico pode estar em no máximo dois consultórios, para isso será criado a tabela com o nome: MovMedicos com os seguintes campos

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Tamanho** |
| Pk | int primary key | 8 |
| FkCadMedicos | Int foreign key [CadMedicos] | 8 |
| FkCadConsultorio | Int foreign key [CadConsultorio] | 8 |

Pode ser criado uma constraint que verifica a condição acima da quantidade vínculos que médico pode ter, isso é uma proteção a nível banco de dados.

**Referências**

Foram utilizados os seguintes pacotes nugets, demais referências são nativas do .NET

* MySql.Data
* Newtonsoft.Json

Será criado uma API que irá conter os seguintes endpoints, esta deverá ter uma hospedagem que forneça suporte a .NET.

* Consultórios

- Criar consultório [POST]

- Listar [GET]

- Remover [DELETE]

- Alterar [PUT]

* Médicos

- Criar consultório [POST]

- Listar [GET]

- Remover [DELETE]

- Alterar [PUT]

- DesvinculaMedicoConsultorio[DELETE]