

Documentação

Projeto de software: Portal da Saúde de Poá

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lucas Nogueira de Souza  Aprendiz Técnico de Desenvolvimento de Sistemas | Sesi Suzano | SESI Suzano  Última atualização: 16/09/2020 |  | Sess |

Resumo

Objetivos

Esse documento tem o objetivo de detalhar os requisitos dos usuários que deverão ser atendidos pelo sistema a ser construído nesse projeto. Ele servirá como base para a construção do sistema por parte dos programadores.

Aqui estará presente uma descrição detalhada das funções do produto, de suas interfaces e do comportamento esperado do sistema.

Descrição do projeto

O Portal da saúde tem como objetivo mostrar os locais de atendimento de saúde na cidade de Poá, localizada no estado de São Paulo.

Resumo do projeto

O Portal da Saúde de Poá pode exibir aos seus usuários todos os locais de atendimento de saúde em Poá, tais como hospitais, Unidades Básicas de Saúde, Unidades de Pronto Atendimento, entre outros. Além disso, o sistema exibirá os serviços que cada local oferece, e a situação dos mesmos. Com isso, os cidadãos da cidade saberão onde encontrar o(s) serviço(s) que procuram. O portal também exibirá as campanhas acontecendo na cidade, como campanhas de vacinação, por exemplo; e os locais onde essas ocorrem.

O sistema contará com um **sistema para todos os usuários**, onde poderão ter acesso a todas as funcionalidades citadas acima; e um **sistema de administradores**, onde estes poderão administrar as informações exibidas aos usuários.

Situação Atual

Na atual versão desse documento (Setembro de 2020), o portal da saúde está finalizado em ambiente de desenvolvimento. Sendo assim, ainda não há acesso público ao sistema.

Modelagem de Software

Escopo

|  |  |  |
| --- | --- | --- |
| **Nº** | **Módulo** | **Descrição** |
| 1 | Sistema Web | Será desenvolvido um sistema web para os usuários terem acesso às informações e para os administradores poderem editar as mesmas. |
| 2 | Web API | Será desenvolvido uma api que fará a integração entre o sistema web e o banco de dados. |
| 3 | Banco de Dados | Será desenvolvido um banco de dados responsável por armazenar todos os dados da aplicação. |

Modelagem de Dados

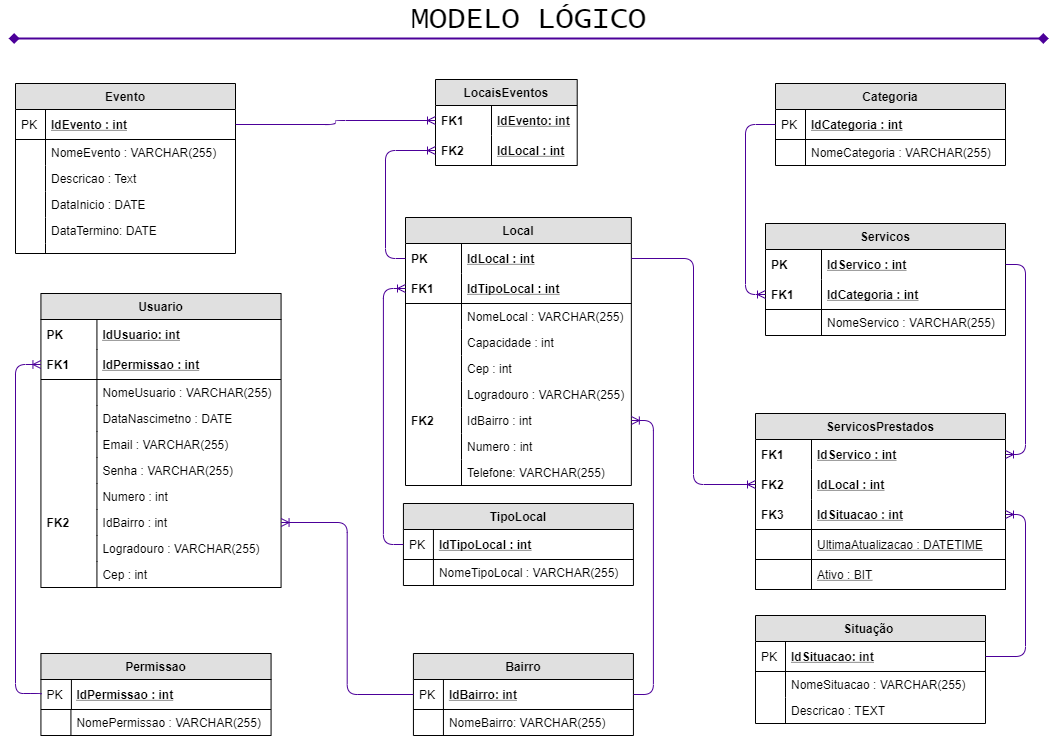
A **modelagem de dados** é uma técnica usada para a especificação das regras de negócios e as estruturas de dados de um banco de dados. Ela faz parte do ciclo de desenvolvimento de um sistema de informação e é de vital importância para o bom resultado do projeto. Modelar dados consiste em desenhar o sistema de informações, concentrando-se nas entidades lógicas e nas dependências lógicas entre essas entidades.

A modelagem de dados está dividida em:

Modelo Lógico

A modelagem lógica do banco de dados é usada para explorar os conceitos do domínio e seus relacionados. Isso pode ser feito para o escopo de um simples projeto ou para uma empresa inteira. Os modelos lógicos descrevem os tipos de entidades lógicas, tipicamente referenciadas simplesmente como tipos de entidades, os atributos de dados que descrevem essas entidades e os relacionamentos entre as entidades.

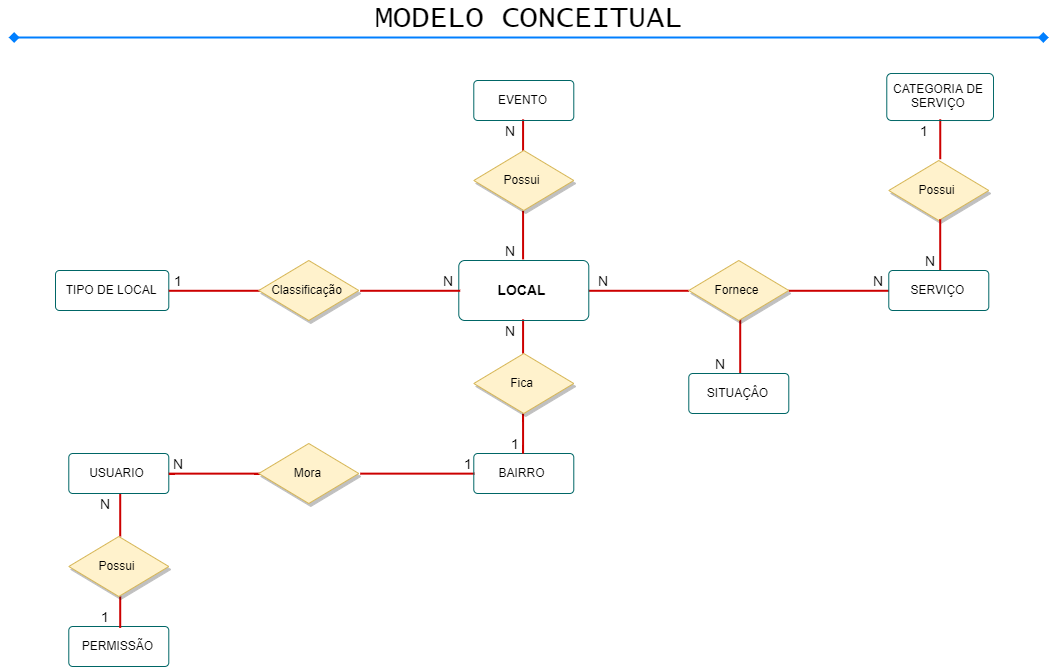
Abaixo, veja o modelo lógico do banco de dados do projeto:



Modelo Conceitual

A modelagem conceitual baseia-se no mais alto nível e deve ser usada para envolver o cliente, pois o foco aqui é discutir os aspectos do negócio do cliente e não da tecnologia. Os exemplos de modelagem de dados vistos pelo modelo conceitual são mais fáceis de compreender, já que não há limitações ou aplicação de tecnologia específica. O diagrama de dados que deve ser construído aqui é o Diagrama de Entidade e Relacionamento, onde deverão ser identificados todas as entidades e os relacionamentos entre elas. Este diagrama é a chave para a compreensão do modelo conceitual de dados.

Abaixo, veja o modelo conceitual do banco de dados do projeto:



Atores

|  |  |  |
| --- | --- | --- |
| **Nº** | **Ator** | **Definição e Privilégio de Acesso e Segurança** |
| 1 | Usuário | Usuário comum que terá acesso à exibição das informações da aplicação, como locais de atendimento, serviços,etc. |
| 2 | Administrador | Usuário administrador que terá acesso ao sistema de administradores, onde o mesmo poderá editar as informações contidas no sistema. |

Premissas

* Hospedagem do servidor Web API (ASP.NET Core 2.2);
* Hospedagem do banco de dados SQL Server;
* Hospedagem do sistema web.

Requisitos

**Requisitos são solicitações, desejos, necessidades**. Um requisito é a propriedade que um software exibe para solucionar problemas reais, é a conjuntura indispensável para satisfazer um objeto. Um requisito é uma maneira pelo qual o sistema oferecido deve fazer, ou um condicionamento no desenvolvimento do sistema. Os requisitos são separados em:

Requisitos Funcionais

Os **requisitos funcionais são todas as necessidades**, características ou funcionalidades esperadas em um processo que podem ser atendidos pelo software. De forma geral, um requisito funcional expressa uma ação que deve ser realizada através do sistema.

Abaixo, confira os requisitos funcionais a aplicação:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Descrição** |  | **Ator** | |
| RF001 | Listar Locais de Atendimento | | | Usuário e Administrador |
| RF002 | Listar Serviços de Saúde | | | Usuário e Administrador |
| RF003 | Listar Áreas de atendimento | | | Usuário e Administrador |
| RF004 | Listar Campanhas | | | Usuário e Administrador |
| RF005 | Listar serviços presentes em local específicado | | | Usuário e Administrador |
| RF006 | Listar locais que oferecem o serviço especificado | | | Usuário e Administrador |
| RF007 | Listar serviços de cada área de atendimento | | | Usuário e Administrador |
| RF008 | Listar locais participantes da campanha especificada | | | Usuário e Administrador |
| RF009 | Fazer Login | | | Administrador |
| RF010 | Fazer Logout | | | Administrador |
| RF011 | Cadastrar Local de Atendimento | | | Administrador |
| RF012 | Editar Local de Atendimento | | | Administrador |
| RF013 | Excluir Local de Atendimento | | | Administrador |
| RF014 | Cadastrar Serviço | | | Administrador |
| RF015 | Editar Serviço | | | Administrador |
| RF016 | Excluir Serviço | | | Administrador |
| RF017 | Cadastrar área de atendimento | | | Administrador |
| RF018 | Editar área de atendimento | | | Administrador |
| RF019 | Excluir área de atendimento | | | Administrador |
| RF020 | Vincular serviço a local | | | Administrador |
| RF021 | Desvincular serviço de local | | | Administrador |
| RF022 | Vincular local à determinada campanha | | | Administrador |
| RF023 | Desvincular local de campanha | | | Administrador |
| RF024 | Cadastrar novo administrador | | | Administrador |

Requisitos não funcionais

Os **requisitos não funcionais** são praticamente todas as **necessidades que não podem ser atendidas através de funcionalidades**. Geralmente mensurável, os requisitos não funcionais definem características e impõe limites do sistema como método de desenvolvimento, tempo, espaço, Sistema Operacional, dentre outros e cuja medida pode ser determinada é importante que se associe essa medida ou referência à cada requisito não funcional. Resumidamente, Um requisito funcional é **o que** o sistema deve fazer um requisito não funcional é **como** o sistema deve fazer.

Abaixo, confira os requisitos não funcionais da aplicação:

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Categoria** |
| RNF001 | Identidade Visual | Usabilidade |
| RNF002 | Telas de listagem simples | Usabilidade |
| RNF003 | Mínimo de 6 caracteres na senha | Segurança |
| RNF004 | Verificação de autenticação e autorização ao realizar qualquer ação de administrador. | Segurança |

Regras de Negócio

Regras de negócio são **premissas e restrições** aplicadas ao software para que a aplicação funcione da maneira esperada.

Abaixo, confira as Regras de Negócio da aplicação:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Nome** | | **Descrição** | |
| RN001 | | Somente administradores terão acesso ao sistema de administradores. | | Apenas usuários previamente cadastrados por outro administrador autorizado poderão ter acesso ao sistema de administradores e todas suas funções. |
| RN002 | | Apenas locais de atendimento localizados na cidade de Poá serão cadastrados no sistema. | | Apenas locais de atendimento (hospitais,UBSs,UPAs,etc.) localizados na cidade de Poá serão cadastrados e exibidos aos usuários. |
| RN003 | | Um endereço de email não poderá ser vinculado a mais de um usuário. | | O sistema não cadastrará emails que já pertencem a outros usuários, evitando assim conflitos na autenticação e autorização dos usuários. |
| RN004 | | Os nomes dos locais de atendimento deverão ser únicos. | | O sistema não cadastrará locais com um nome já existente. |
| RN005 | | Os nomes das áreas de atendimento terão um limite de caracteres. | | As áreas de atendimento(categorias) terão um limite de 255 caracteres em seu nome. |
| RN006 | | Os nomes dos locais de atendimento terão limites de caracteres. | | Os nomes dos locais de atendimento cadastrados pelo sistema deverão ter no mínimo 5 caracteres e no máximo 255 caracteres. |
| RN007 | | As áreas de atendimento, situações, permissões, tipos de locais, bairros, e locais serão únicos. | | As entidades citadas não poderão ser cadastradas mais de uma vez com o mesmo nome, não permitindo assim a duplicação das mesmas. |

Restrições de Hardware

|  |  |
| --- | --- |
| **Nº** | **Descrição** |
| 1 | Dispositivo deve conter acesso à internet. |

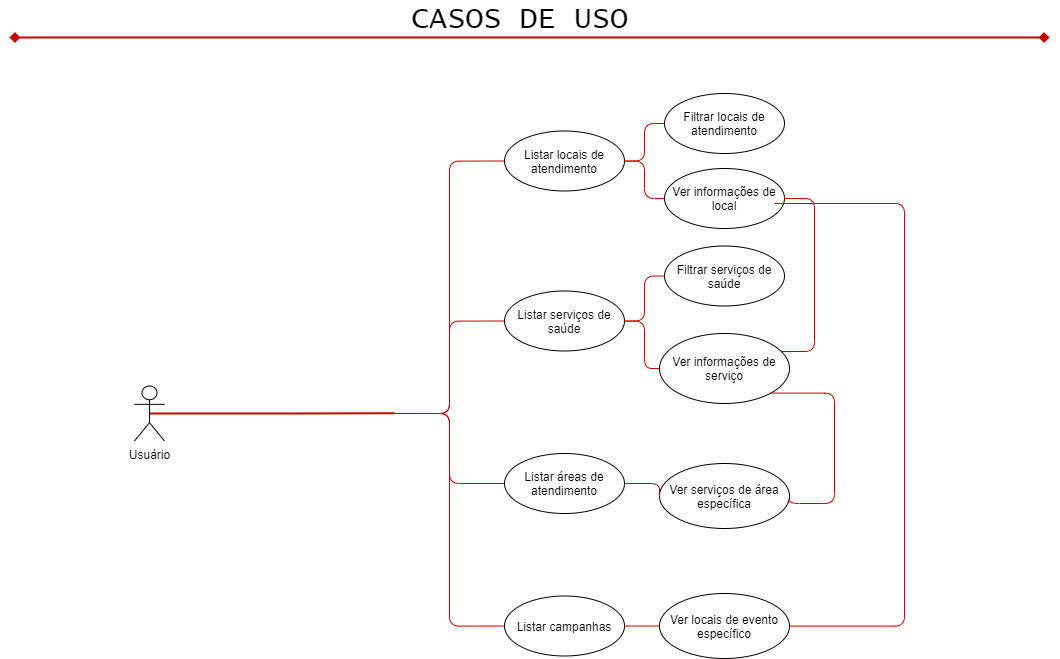
Restrições de Software

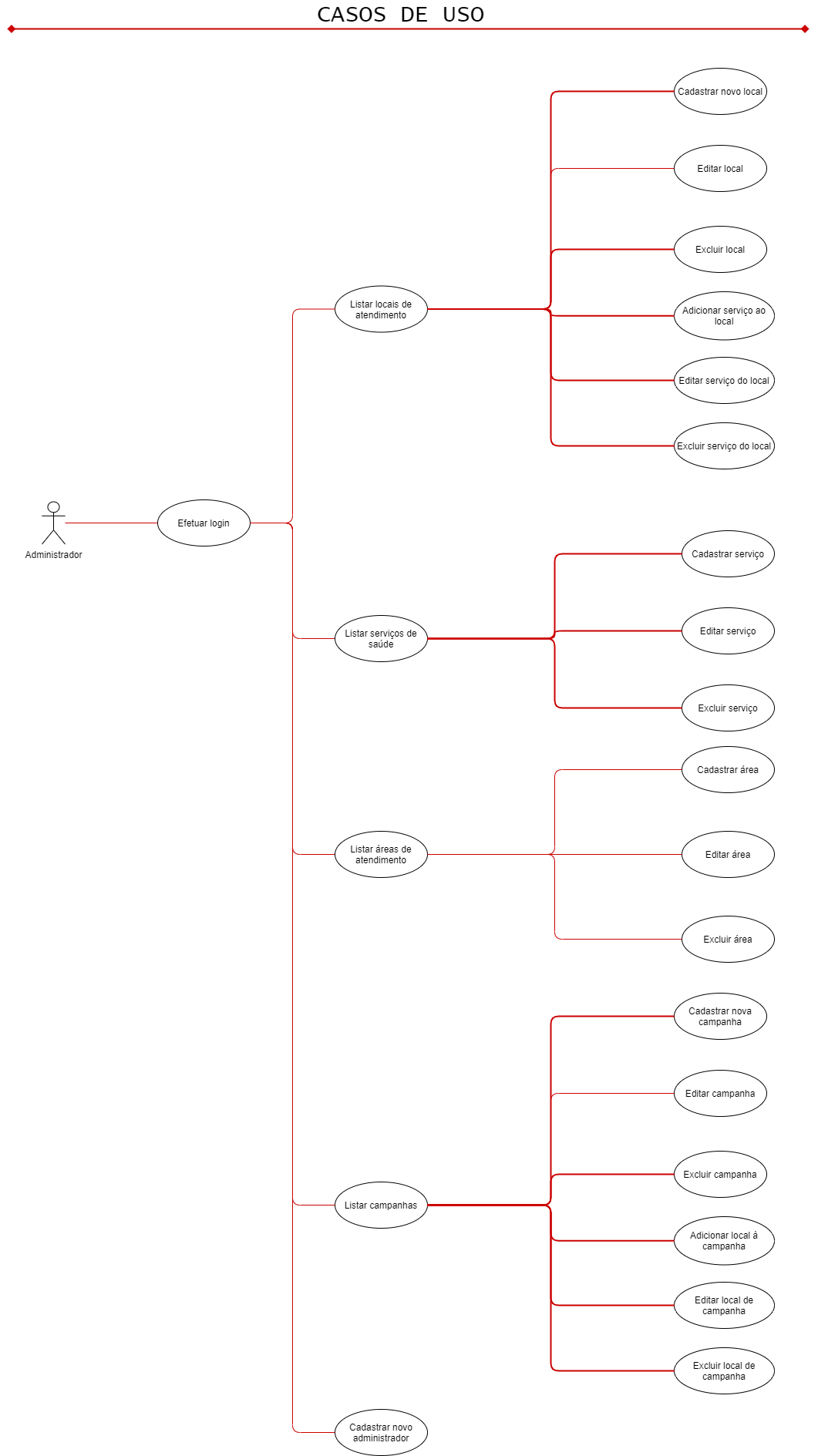
|  |  |
| --- | --- |
| **Nº** | **Descrição** |
| 1 | Dispositivo deve conter um navegador web. |

Casos de Uso

Um caso de uso narra a interação entre o sistema e os atores envolvidos, para atingir um ou mais objetivos.

Abaixo, confira os casos de uso da aplicação:





Funcionalidades

Funcionalidades Web

Usuário:

* Página Inicial;
* Locais de Atendimento
* Áreas de Atendimento;
* Campanhas;
* Sobre;
* Login.

Administrador:

* Página inicial de administradores;
* Editar Locais de Atendimento
* Editar Serviços de Saúde
* Editar Áreas de Atendimento
* Cadastrar administrador;

Banco de Dados

O banco de dados **é a organização e armazenagem de informações sobre um domínio específico.** De forma mais simples, é o agrupamento de dados que tratam do mesmo assunto, e que precisam ser armazenados para segurança ou conferência futura.

Na aplicação, foi desenvolvido um **banco de dados relacional,** que tem por definição ser uma coleção de itens de dados com relacionamentos predefinidos entre si. Esses itens são organizados como um conjunto de tabelas com colunas e linhas. As tabelas são usadas para reter informações sobre os objetos a serem representados no banco de dados. Cada coluna da tabela retém um determinado tipo de dado e um campo armazena o valor em si de um atributo. As linhas na tabela representam uma coleção de valores relacionados de um objeto ou de uma entidade. Cada linha em uma tabela pode ser marcada com um único identificador chamado de chave principal. Já as linhas entre as várias tabelas podem ser associadas usando chaves estrangeiras. Esses dados podem ser acessados de várias maneiras diferentes sem reorganizar as próprias tabelas do banco de dados.

Tecnologias usadas

O sistema de gerenciamento do banco de dados utilizado foi o **Microsoft SQL Server Express** através do **SQL Server Management Studio.**

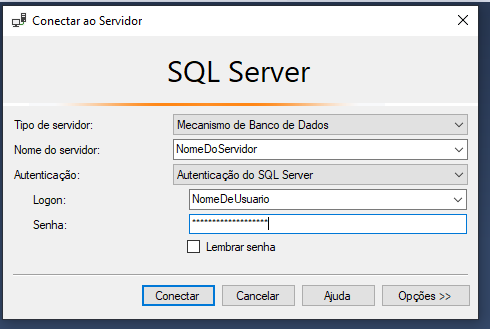
Implementação

Abaixo, confira os passos para a implementação do banco de dados:

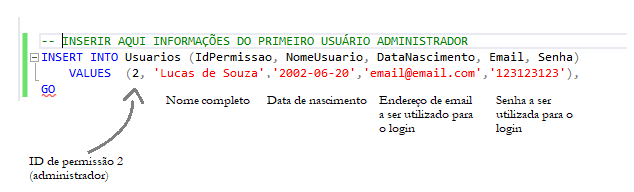
1. Obter um servidor de hospedagem SQL Server.

(**OBSERVAÇÃO**: Os servidores de hospedagem geralmente oferecem instruções de como se conectar ao banco de dados, opte sempre pela instrução da hospedagem.);

1. Abrir o SQL Server Management Studio;
2. Se conectar ao banco de dados, com as informações do servidor de hospedagem obtidos no passo 1 (ver exemplo abaixo);



1. Abrir o arquivo **DDL.sql (**website-da-saude/database/scripts/DDL.sql**);**
2. Clicar em “**Executar**” (encontrado na parte superior da tela) ou use o atalho **F5;**
3. Abrir o arquivo **DML.sql (**website-da-saude/database/scripts/DML.sql**)**;
4. **IMPORTANTE:** Na linha 122 do arquivo, inserir as informações do primeiro usuário administrador (veja exemplo abaixo);



1. Repetir o passo 5.

Back-end

Back-end, como o próprio nome sugere, vem da ideia **do que tem por trás de uma aplicação.** Essa parte do sistema trabalha fazendo a ponte entre os dados que vem do navegador rumo ao banco de dados e vice-versa, sempre aplicando as devidas regras de negócio, validações e garantias em um ambiente onde o usuário final não tenha acesso e possa manipular algo.

Na aplicação, foi desenvolvido um sistema **ASP.NET Web API**, com a linguagem de programação **C# (C Sharp)**.Uma API (Application Programming Interface)  tem o papel de conectar dois sistemas por meio de uma linguagem de programação em comum. Por exemplo, fazendo com que aplicações, base de dados e serviços consigam se comunicar. Evitando, assim, programações complexas e cansativas.

## ASP.Net Web API é um framework que facilita a construção de serviços REST HTTP que alcançam uma grande variedade de clientes incluindo Mobile, Browsers e aplicações locais.

Na API desenvolvida, os dados são recebidos e devolvidos no formato JSON (JavaScript Object Notation). A vantagem de se ter uma Web API com esse formato é a escabilidade que esta traz para o sistema, já que seria possível se conectar ao servidor através de um aplicativo mobile, por exemplo, eliminando assim a necessidade de desenvolver outro sistema Back-end para o mesmo.

Implementação

Para a implementação, será necessário um servidor de hospedagem ASP.NET. O processo de publicar o servidor back-end varia de acordo com o serviço de hospedagem escolhido. Portanto, procure saber os passos para a implementação de acordo com o serviço de hospedagem escolhido. (exemplo: Azure, AWS, Smarter ASP.NET, entre outros).

Front-end

Front-End é basicamente a interface do usuário, sendo tudo o que o usuário pode ver e interagir com toda a “frente” do software. Há então as camadas de interações do usuário, ou seja, efeitos, animações e etc.

Toda interatividade que existe em uma página, ao clicar em algo e ter um alerta, por exemplo, todo fluxo da aplicação e toda a camada visual é o Front-End.

O Front-end da aplicação foi construído utilizando tecnologias base da web, que são: **HTML**, **CSS** e **JavaScript.**

Implementação

Para a implementação, será necessário um servidor de hospedagem comum. O processo de publicar o Front-end varia de acordo com o serviço de hospedagem escolhido. Portanto, procure saber os passos para a implementação de acordo com o serviço de hospedagem escolhido.

Arquitetura do Projeto

