**Senac - Projeto Integrador**

**Sistema Mobile**

**TI-27**

Ezequiel

Senac Botucatu

Sumário

[Introdução 2](#_Toc21621771)

[Tela TLogin 4](#_Toc21621772)

[Tela de Menu 9](#_Toc21621773)

[**TCadastroEmpregados** 10](#_Toc21621774)

[TComodos 15](#_Toc21621775)

[Mock Flow 17](#_Toc21621776)

# Introdução

Esse projeto tem o intuito de alcançar as medidas básicas exigidas pelo Senac Botucatu. Meu sistema é um sistema de limpeza de casa.

Utilizamos da linguagem de Banco de dados SQLite, linguagem de programação C# e XAML .

Após isso, será colocado todos os Requisitos funcionais e não funcionais do sistema organizados a baixo por Tela, Classe, Método e Tabela.

# Tela TLogin

|  |  |
| --- | --- |
| **MainPage** | |
| **Identificador** | 20190911-Login |
| **Nome** | TLogin |
| **Módulo** | Programa de Limpeza |
| **Data de criação** | 11/08/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A tela é responsável por realizar o Acesso do usuário, para que ele possa acessar outra parte do sistema como Menu e as telas de cadastro (ID:)  entryNome: (Não pode ficar vazio)  entrySenha:(Não pode ficar vazio)  buttonLogar: (Inicia o evento que faz comparação das informações preenchidas)  buttonCriarConta: (Abre o Form de cadastro de usuário (ID:)) |

|  |  |
| --- | --- |
| **TCadastroUsuario** | |
| **Identificador** | 20190911-Login |
| **Nome** | TLogin |
| **Módulo** | Programa de Limpeza |
| **Data de criação** | 11/08/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A tela é responsável por realizar o Acesso do usuário, para que ele possa acessar outra parte do sistema como Menu e as telas de cadastro (ID:)  entryNomePessoa: (Não pode ficar vazio)  entryNomeUsuario:(Não pode ficar vazio)  entryEmail:(Não pode ficar vazio)  entrySenha:(Não pode ficar vazio)  buttonCadastrar: cadastra o usuario |

|  |  |
| --- | --- |
| **Classe - CUsuario | METODO –** CadastrarUsuario | |
| **Identificador** | 20191010-CadastrarUsuario |
| **Nome** | Cadastrar Usuario |
| **Módulo** | Programa de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A classe é responsável por comparar as informações das contas dos usuarios, para a verificação se os dados estão corretos, sua conexão é com a tabela (ID:20191010-Login) |
| **Exemplo de uso** | **Exemplo de uso**  Classe.CUsuario CU = new Classe.CUsuario();  bool Resultado = CU.CadastrarUsuario(entryNomePessoa.Text, entryNomeUsuario.Text, entryEmail.Text, entrySenha.Text, entryCEP.Text);    **Exemplo de retorno com êxito**  Navigation.PushAsync(new MainPage());  RETURN TRUE;  **Exemplo de retorno com erro**  DisplayAlert("Erro", "°Não deixe os campos vazios !!!", "OK");  RETURN FALSE; |

|  |  |
| --- | --- |
| **Classe - CUsuario | METODO – Login** | |
| **Identificador** | 20191010-Login |
| **Nome** | Login |
| **Módulo** | Programa de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A classe é responsável por comparar as informações das contas dos usuarios, para a verificação se os dados estão corretos, sua conexão é com a tabela (ID:20190911-Login) |
| **Exemplo de uso** | **Exemplo de uso**  Classe.CUsuario CU = new Classe.CUsuario();  bool Logar = CU.Login(entryNome.Text, entrySenha.Text);  **Exemplo de retorno com êxito**  Navigation.PushAsync(new MenuStrip());  RETURN TRUE;  **Exemplo de retorno com erro**  DisplayAlert("Erro", "Ocorreu um erro no App", "OK");  RETURN FALSE; |

|  |  |
| --- | --- |
| **Login** | |
| **Identificador** | 20191010-Login |
| **Nome** | Login |
| **Módulo** | Programa de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A tabela é responsável por realizar o armazenamento das contas dos usuários do sistema.  Seus campos são:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **NOME** | **TIPO** | **NULO** | **PK / FK** | **DESCRIÇÃO** | | ID | INTEGER | NÃO | Pk | Identificação | | Nome | TEXT(300) | NÃO | - | Nome do Usuário | | Usuario | TEXT(500) | NÃO | - | Senha do usuário | | Email | TEXT(300) | NÃO | - | Email do usuario | | Senha | TEXT(255) | NÃO | - | Senha do cliente | | CEP | TEXT(8) | NÃO | - | CEP do usuario | |

# Tela de Menu

|  |  |
| --- | --- |
| **MenuStrip** | |
| **Identificador** | 20191010-MenuStrip |
| **Nome** | MenuStrip |
| **Módulo** | Programa de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A tela é responsável por realizar o acesso ao usuário a todas as opções do app, MenuStrip está iniciando por cima do TMenu   * Limpar * Serviço * Pedido * Conta |
| **Exemplo de uso** | **Exemplo de uso**  Detail = new NavigationPage(new MainPage());  IsPresented = false; |

# **TCadastroEmpregados**

|  |  |
| --- | --- |
| **TCadastroEmpregados** | |
| **Identificador** | 20191010- TCadastroEmpregados |
| **Nome** | TCadastroEmpregados |
| **Módulo** | Sistema de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A tela é responsável por realizar o acesso a opção de cadastrar e apagar as informações dos Funcionários. Está referenciada com a tabela (ID:20191010-Funcionarios).  entryID: (Não vazio)  entryCPF: (Não vazio e Apenas números)  entryRG: (Não vazio e Apenas números)  entryCep: (Não vazio e Apenas números) |

|  |  |
| --- | --- |
| **Classe - CFuncionario | METODO - Cadastrar** | |
| **Identificador** | 20190911-Cadastrar |
| **Nome** | Cadastrar |
| **Módulo** | Sistema de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A classe é responsável por inserir os dados e iniciar a conexão com o Banco (ID:20191010-tbl\_Cliente)  Ele irar armazena nas seguintes variáveis para a passagem da informação   * string CPF * string CEP * string RG * string Observacao |
| **Exemplo de uso** | **Exemplo de uso**  Bool Inserir = CF.Cadastrar( entryCPF.Text, entryCep.Text, entryRG.Text, entryInformacoes.Text);  **Exemplo de retorno com êxito**  DisplayAlert("Certo", "Funcionario Cadastrado", "OK");  Se a variável receber True:  **Exemplo de retorno com erro**  Se a variável receber False:  DisplayAlert("Erro", "Ocorreu um erro no App", "OK"); |

|  |  |
| --- | --- |
| **Classe - CFuncionario | METODO - Excluir** | |
| **Identificador** | 20191010-Excluir |
| **Nome** | Excluir |
| **Módulo** | Programa de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atua.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A classe é responsável por inserir os dados Excluir e iniciar a conexão com o Banco SQL(ID:20191010-tbl\_Cliente)  Ele irar atualizar as informações das seguintes variáveis presentes:   * int ID |
| **Exemplo de uso** | **Exemplo de uso**  **Exemplo de retorno com êxito**  Se a variável receber True:    **Exemplo de retorno com erro**  Se a variável receber False; |

|  |  |
| --- | --- |
| **Classe - ModelFuncionario |METODO - ModelFuncionario** | |
| **Identificador** | 20191010-**ModelFuncionario** |
| **Nome** | **ModelFuncionario** |
| **Módulo** | Programa de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | O método é responsável por iniciar a conexão binding com o Banco (ID: ID:20191010-Funcionario).  public int ID { get; set; }  public string CPF { get; set; }  public string RG { get; set; }  public string CEP { get; set; }  public string Observacao { get; set; } |

|  |  |
| --- | --- |
| **Funcionario** | |
| **Identificador** | 20191010- Funcionario |
| **Nome** | Funcionario |
| **Módulo** | Programa de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A tabela é responsável por realizar o armazenamento das informações dos funcionarios  Seus campos são:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **NOME** | **TIPO** | **NULO** | **PK / FK** | **DESCRIÇÃO** | | ID | INTENGER | NÃO | Pk | Identificação | | CPF | TEXT | NÃO | - | CPF do Funcionário | | RG | TEXT | NÃO | - | RG do Funcionário | | CEP | TEXT | NÃO | - | CEP do Funcionário | | Observação | TEXT | NÃO | - | Observação do Funcionario | |

# TComodos

|  |  |
| --- | --- |
| TComodos | |
| **Identificador** | 20191010-TComodos |
| **Nome** | TComodos |
| **Módulo** | Programa de Limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A tela é responsável por realizar o acesso cadastro de informações dos fornecedores. Sua tabela referencial do SQL é (ID:)  PickeHora: hora  pickeFuncionario: funcionario  PickeDia: Dia a limpeza |

|  |  |
| --- | --- |
| Pedido | |
| **Identificador** | 20191010-Pedido |
| **Nome** | Pedido |
| **Módulo** | Sistema de limpeza |
| **Data de criação** | 10/10/2019 |
| **Autor** | Ezequiel Pereira da Silva |
| **Data da última atual.** |  |
| **Autor** |  |
| **Versão** | 1.0 |
| **Descrição** | A tabela é responsável por armazenar os editoras  Seus campos são:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **NOME** | **TIPO** | **NULO** | **PK / FK** | **DESCRIÇÃO** | | ID | INTEGER | NÃO | Pk | Identificação | | Banheiro | INTEGER | NÃO | - | Quantidade de banheiros | | Cozinha | INTEGER | NÃO | - | Quantidade de cozinha | | Sala | INTEGER | NÃO | - | Quantidade de Sala | | Quarto | INTEGER | Não | - | Quantidade de quarto | | Hora | TEXT | NÃO | - | HORA | | DATA | TEXT | NÃO | - | DATA | | EMPREGADO | TEXT | NÃO | - | funcionário | | Total | INTEGER | NÃO | - | TOTAL DE CUSTO | |

# Mock Flow

https://wireframepro.mockflow.com/view/M295495c8660dc2a80b5b58663df754881569289046309