

DOCUMENTAÇÃO DE SOFTWARE

GCC

(Gerenciador de Cadastro de Clientes)

- Grupo SP Imóvel

Visão Geral

Sobre

- ***Proposta principal***
- ***Processo de desenvolvimento***

Funcionalidades

- ***Parar os usuários***
- ***Aplicabilidades***
- ***Para os desenvolvedores***
- ***Classes***
- ***Métodos***

Proposta principal

Esse software foi desenvolvido com base no teste proposto pelo “Grupo SP Imóvel”, sendo ele parte do processo seletivo da empresa para a vaga de emprego como desenvolvedor web.

Segue o teste:



Criar Projeto MVC C# (.NET Framework) | JQUERY | CSS | SQL SERVER (SQL Server enviar o .BAK ou script)

1. Banco de Dados SQL Server
 - DataBase “dbTeste”.
 - Criar tabela Cliente(ClientId, Cliente, TipoCliente, NomeContato, TelefoneContato, Cidade, Bairro, Logradouro, DataCadastro, DataAtualizacao) *salvando script SQL do mesmo.
 - Criar Procedures (Consulta, Insert, Update, Delete) *salvando script SQL do mesmo.
 - Gerar 5.000,00 registro na tabela Imoveis *salvando script SQL do mesmo.

Ex: Cliente ('Teste1','TipoCliente1','1111-1111','850000,00','Cidade1','Bairro1','Rua Teste1','10/03/2015','13/04/2015')
Cliente ('Teste2', TipoCliente2,'2222-2222','850000,00','Cidade2','Bairro2','Rua Teste2','11/03/2015','14/04/2015')
2. Projeto MVC: Em Camadas com MVC C# + CSS + JQUERY
 - Criar tela de Gerenciamento de dados: Cadastro(Modal), Consulta em Grid, Edição(Modal), Exclusão e Paginação via SQL Server. (Utilizar JQUERY)
 - Criar input de texto para filtrar todos os campos da tabela. (Utilizar JQUERY)

16:02

Modelo do teste apresentado.

Processo de desenvolvimento

O projeto do site será entregue dia 28/09/2009 (depois do prazo estabelecido inicialmente), o modelo está incompleto no que se refere a edição, exclusão, e cadastro dos registros via modal pelo JQuery. No decorrer do desenvolvimento foi possível notar um ponto forte a ser melhorado nesse caso os conhecimentos relacionados a tecnologias Javascript, tendo em vista as várias tentativas falhas na implementação de recursos visuais (downloads de bibliotecas, referências e métodos que não surtiram efeito).

ystem Home

Sistema Gerenciador de Cadastros

10	resultados por página							Pesquisar		
Id	Cliente	Contato	Tipo	Número Celular	Cidade	Bairro	Logradouro	Cadastro	Atualização	+
1	Teste1	NomeContato1	Tipo1	19 924349582	Cidade1	Bairro1	RuaTeste1	24/09/2022 21:36:42		<div>EditDelete</div>
2	Teste2	NomeContato2	Tipo2	19 99337131	Cidade2	Bairro2	RuaTeste2	24/09/2022 21:36:42		<div>EditDelete</div>
3	Teste3	NomeContato3	Tipo3	19 959021950	Cidade3	Bairro3	RuaTeste3	24/09/2022 21:36:42		<div>EditDelete</div>
4	Teste4	NomeContato4	Tipo4	19 932941141	Cidade4	Bairro4	RuaTeste4	24/09/2022 21:36:42		<div>EditDelete</div>
5	Teste5	NomeContato5	Tipo5	19 982981965	Cidade5	Bairro5	RuaTeste5	24/09/2022 21:36:42		<div>EditDelete</div>
6	Teste6	NomeContato6	Tipo6	19 995728174	Cidade6	Bairro6	RuaTeste6	24/09/2022 21:36:42		<div>EditDelete</div>
7	Teste7	NomeContato7	Tipo7	19 941453871	Cidade7	Bairro7	RuaTeste7	24/09/2022 21:36:42		<div>EditDelete</div>
8	Teste8	NomeContato8	Tipo8	19 932437394	Cidade8	Bairro8	RuaTeste8	24/09/2022 21:36:42		<div>EditDelete</div>
9	Teste9	NomeContato9	Tipo9	19 999427653	Cidade9	Bairro9	RuaTeste9	24/09/2022 21:36:42		<div>EditDelete</div>

Funcionalidades

Para usuários:

No sistema gerenciador de cadastros é possível filtrar, pesquisar por item e “paginar” os 5.000 registros extraídos do banco de dados SQL Server.

Sistema Gerenciador de Cadastros

10 resultados por página

Pesquisar

Id	Cliente	Contato	Tipo	Número Celular	Cidade	Bairro	Logradouro	Cadastro	Atualização	+
1	Teste1	NomeContato1	Tipo1	19 924349582	Cidade1	Bairro1	RuaTeste1	24/09/2022 21:36:42		<div>EditDelete</div>
2	Teste2	NomeContato2	Tipo2	19 99337131	Cidade2	Bairro2	RuaTeste2	24/09/2022 21:36:42		<div>EditDelete</div>
3	Teste3	NomeContato3	Tipo3	19 959021950	Cidade3	Bairro3	RuaTeste3	24/09/2022 21:36:42		<div>EditDelete</div>
4	Teste4	NomeContato4	Tipo4	19 932941141	Cidade4	Bairro4	RuaTeste4	24/09/2022 21:36:42		<div>EditDelete</div>
5	Teste5	NomeContato5	Tipo5	19 982981965	Cidade5	Bairro5	RuaTeste5	24/09/2022 21:36:42		<div>EditDelete</div>
6	Teste6	NomeContato6	Tipo6	19 995728174	Cidade6	Bairro6	RuaTeste6	24/09/2022 21:36:42		<div>EditDelete</div>
7	Teste7	NomeContato7	Tipo7	19 941453871	Cidade7	Bairro7	RuaTeste7	24/09/2022 21:36:42		<div>EditDelete</div>
8	Teste8	NomeContato8	Tipo8	19 932437394	Cidade8	Bairro8	RuaTeste8	24/09/2022 21:36:42		<div>EditDelete</div>
9	Teste9	NomeContato9	Tipo9	19 999427653	Cidade9	Bairro9	RuaTeste9	24/09/2022 21:36:42		<div>EditDelete</div>

1	Teste1	NomeContato1	Tipo1	19 924349582	Cidade1	Bairro1	RuaTeste1	24/09/2022 21:36:42		<div>EditDelete</div>
2	Teste2	NomeContato2	Tipo2	19 99337131	Cidade2	Bairro2	RuaTeste2	24/09/2022 21:36:42		<div>EditDelete</div>
3	Teste3	NomeContato3	Tipo3	19 959021950	Cidade3	Bairro3	RuaTeste3	24/09/2022 21:36:42		<div>EditDelete</div>
4	Teste4	NomeContato4	Tipo4	19 932941141	Cidade4	Bairro4	RuaTeste4	24/09/2022 21:36:42		<div>EditDelete</div>
5	Teste5	NomeContato5	Tipo5	19 982981965	Cidade5	Bairro5	RuaTeste5	24/09/2022 21:36:42		<div>EditDelete</div>
6	Teste6	NomeContato6	Tipo6	19 995728174	Cidade6	Bairro6	RuaTeste6	24/09/2022 21:36:42		<div>EditDelete</div>
7	Teste7	NomeContato7	Tipo7	19 941453871	Cidade7	Bairro7	RuaTeste7	24/09/2022 21:36:42		<div>EditDelete</div>
8	Teste8	NomeContato8	Tipo8	19 932437394	Cidade8	Bairro8	RuaTeste8	24/09/2022 21:36:42		<div>EditDelete</div>
9	Teste9	NomeContato9	Tipo9	19 999427653	Cidade9	Bairro9	RuaTeste9	24/09/2022 21:36:42		<div>EditDelete</div>
10	Teste10	NomeContato10	Tipo10	19 999270608	Cidade10	Bairro10	RuaTeste10	24/09/2022 21:36:42		<div>EditDelete</div>

Mostrando de 1 até 10 de 5.000 registros

Anterior

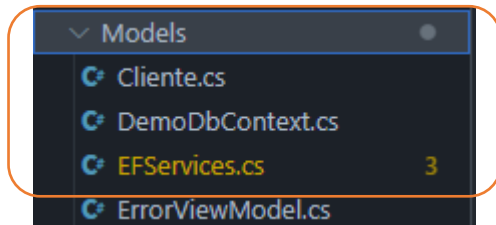
12345...500

Próximo

Para os desenvolvedores

- Classes e Métodos

No total são 3 classes principais:



A classe **Cliente** modula os objetos “clientes”, foi criada a partir de uma tabela com o mesmo nome, presente no banco de dados.

```
12 references
6  public partial class Cliente
7
8  //Model criado a partir do modelo de tabela do Banco de dados (Database First/ Engenharia Reversa EF)
9
10 public int ClienteId { get; set; }
11 public string Cliente1 { get; set; } = null!;
12 public string TipoCliente { get; set; } = null!;
13 public string NomeContato { get; set; } = null!;
14 public string TelefoneContato { get; set; } = null!;
15 public string Cidade { get; set; } = null!;
16 public string Bairro { get; set; } = null!;
17 public string Logradouro { get; set; } = null!;
18 public DateTime DataCadastro { get; set; }
19 public DateTime? DataAtualizacao { get; set; }
20 }
```

Propriedades

dbo.Cliente	
Columns	
Clienteld	(int, not null)
Cliente	(nvarchar(80), not null)
TipoCliente	(nvarchar(60), not null)
NomeContato	(nvarchar(80), not null)
TelefoneContato	(nvarchar(13), not null)
Cidade	(nvarchar(40), not null)
Bairro	(nvarchar(40), not null)
Logradouro	(nvarchar(80), not null)
DataCadastro	(datetime, not null)
DataAtualizacao	(datetime, null)

Já a classe **DemoDbContext** é responsável por gerenciar as operações no banco de dados através do Entity Framework

```
using System;
using System.Collections.Generic;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata;
//criado a partir dos comandos do Entity Framework
namespace Gerenciamento_Clientes.Models
{
    5 references
    public partial class DemoDbContext : DbContext
    {
        3 references
        public DemoDbContext()
        {
        }

        0 references
        public DemoDbContext(DbContextOptions<DemoDbContext> options)
            : base(options)
        {
        }

        3 references
        public virtual DbSet<Cliente> Clientes { get; set; } = null!;
```

```
3 references
public virtual DbSet<Cliente> Clientes { get; set; } = null!;

0 references
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    if (!optionsBuilder.IsConfigured)
    {
        warning To protect potentially sensitive information in your connection string, you should move it out of source
        optionsBuilder.UseSqlServer("Server=localhost,1433;DataBase=dbTeste;Uid=sa;Pwd=1q2w3e4r@#$$");
    }
}
```

```

0 references
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Cliente>(entity =>
    {
        entity.HasNoKey();

        entity.ToTable("Cliente");

        entity.Property(e => e.Bairro).HasMaxLength(40);

        entity.Property(e => e.Cidade).HasMaxLength(40);

        entity.Property(e => e.Cliente1)
            .HasMaxLength(80)
            .HasColumnName("Cliente");

        entity.Property(e => e.ClienteId).ValueGeneratedOnAdd();

        entity.Property(e => e.DataAtualizacao).HasColumnType("datetime");

        entity.Property(e => e.DataAtualizacao).HasColumnType("datetime");

        entity.Property(e => e.DataCadastro)
            .HasColumnType("datetime")
            .HasDefaultValueSql("(getdate())");

        entity.Property(e => e.Logradouro).HasMaxLength(80);

        entity.Property(e => e.NomeContato).HasMaxLength(80);

        entity.Property(e => e.TelefoneContato).HasMaxLength(13);

        entity.Property(e => e.TipoCliente).HasMaxLength(60);
    });

    OnModelCreatingPartial(modelBuilder);
}

1 reference
partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

```

A classe **EFServices** foi utilizada na tentativa de buscar um objeto cliente por id, porém sem êxito.


```
using System.Linq;
namespace Gerenciamiento_Clientes.Models
{
    0 references
    public class EFServices
    {
        0 references
        public Cliente acharCliente(int id)
        {
            Cliente clienteEncontrado = null;
            using (var context = new DemoDbContext())
            {
                clienteEncontrado = context.Clientes.Where(s => s.ClienteId == id).FirstOrDefault<Cliente>();
            }
            return clienteEncontrado;
        }
    }
}
```