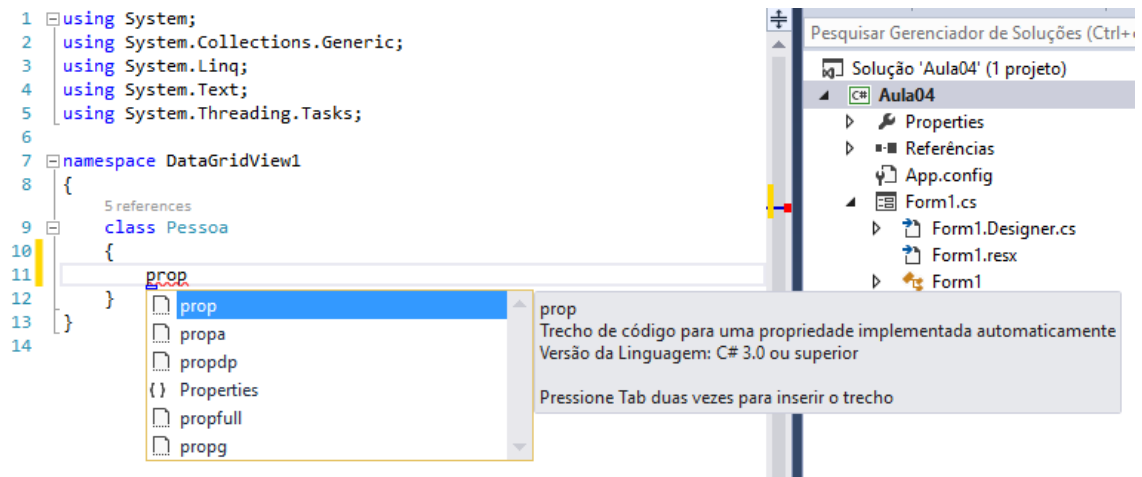


Aula 04 - Ambientes Visuais

Objetivo:

Definir classes e objetos para representar os dados, apresentar os dados com o componente DataGridView, utilização a coleção `List<T>` para acumular valores durante a execução do programa.

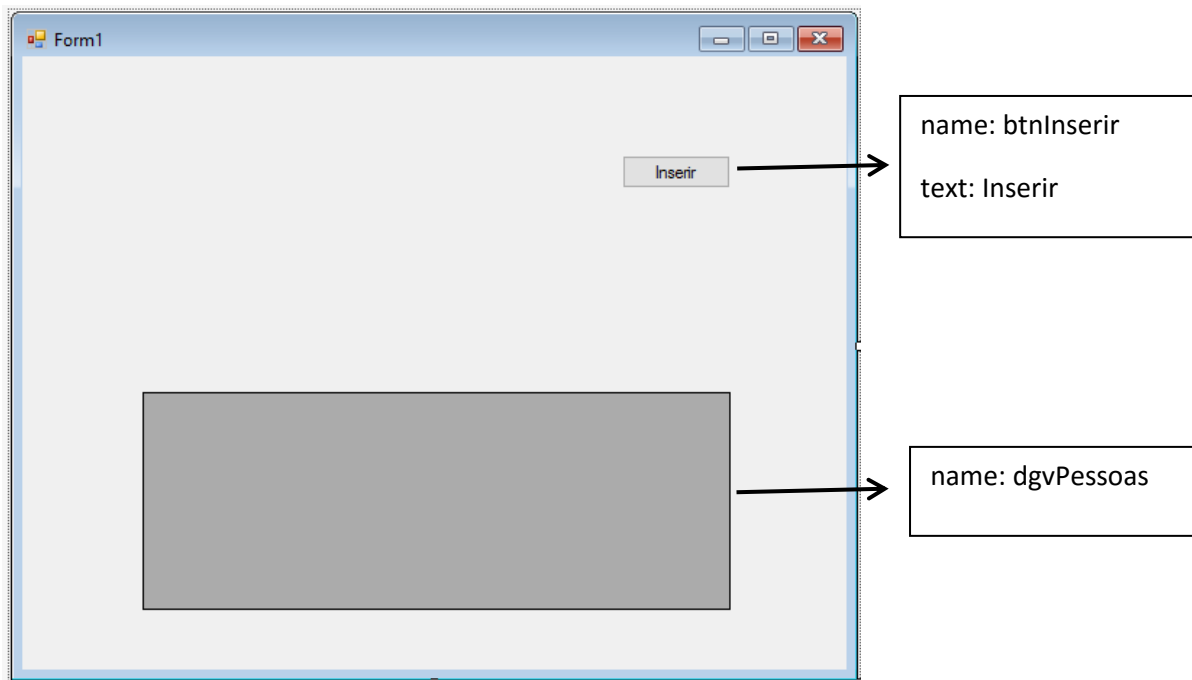
1. Crie um novo projeto no Visual Studio chamado Aula04
2. Clique com o botão direito do mouse sobre o nome do projeto e adicione ao projeto uma Classe.cs. Menu Adicionar > Classe, de o nome de Pessoa.cs
3. Para adicionar os campos (ou propriedade) da classe Pessoa digite o atalho **prop** e a tecla Tap.



- 3.1. Declare as seguintes propriedades para a classe Pessoa.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Aula04
8 {
9     class Pessoa
10     {
11         public string nome { get; set; }
12
13         public string dataNascimento { get; set; }
14
15         public string cpf { get; set; }
16
17         public string rg { get; set; }
18     }
19 }
20
```

4. Monte no formulário do projeto, adicione o componente DataGridView e um botão, como o exemplo abaixo.



5. Adicione um event Click no botão Inserir. Neste evento vamos inicialmente criar uma coleção (lista) de pessoas e exibir na datagridview.
- 5.1. Declare como atributo da classe Form1, um List para armazenar objetos da classe Pessoa.

```
10
11 namespace Aula04
12 {
13     3 references
14     public partial class Form1 : Form
15     {
16         //declaracao lista de objetos da classe pessoa
17         List<Pessoa> lista = new List<Pessoa>();
18
19         1 reference
20         public Form1()
21         {
22             InitializeComponent();
23         }
24     }
25 }
```

- 5.2. Crie um evento Clique no botão Inserir e implemente da seguinte forma.

```

23 private void btnInserir_Click(object sender, EventArgs e)
24 {
25
26     //cria 3 objetos da classe Pessoa
27     Pessoa p1 = new Pessoa();
28     p1.nome = "Maria da Silva";
29     p1.dataNascimento = "12/11/1982";
30     p1.cpf = "333.333.333-33";
31     p1.rg = "45.369.895-96";
32
33     Pessoa p2 = new Pessoa();
34     p2.nome = "Pedro Algustos";
35     p2.dataNascimento = "06/07/1972";
36     p2.cpf = "555.555.555-55";
37     p2.rg = "22.222.222-22";
38
39     Pessoa p3 = new Pessoa();
40     p3.nome = "Mario Henrique";
41     p3.dataNascimento = "09/02/2000";
42     p3.cpf = "444.444.444-44";
43     p3.rg = "77.777.777-77";
44
45     //adiciona os objetos na lista
46     lista.Add(p1);
47     lista.Add(p2);
48     lista.Add(p3);
49
50
51     //exibe a lista na datagridview
52     dgvPessoas.DataSource = lista;
53
54 }

```

5.3. Execute aplicação e clique no botão Inserir.

The screenshot shows a Windows application window titled "Form1". In the top right corner, there is a button labeled "Inserir". Below the button is a DataGridView control displaying a table with the following data:

	nome	dataNascimento	cpf	rg
▶	Maria da Silva	12/11/1982	333.333.333-33	45.369.895-96
	Pedro Algustos	06/07/1972	555.555.555-55	22.222.222-22
	Mario Henrique	09/02/2000	444.444.444-44	77.777.777-77

Below the table, there is a horizontal scrollbar.

5.4. Adicione os componentes de texto ao Form, para entrega de dados pelo usuário.

The screenshot shows a Windows Form titled 'Form1'. It contains four text input fields with labels: 'Nome:', 'Data de Nascimento:', 'RG:', and 'CPF:'. To the right of each input field is a label indicating its name: 'name: txtNome', 'name: txtDataNascimento', 'name: txtRG', and 'name: txtCPF'. Below the input fields is a button labeled 'Inserir'. At the bottom of the form is a large, empty rectangular area, which is intended for a DataGridView control.

5.5. Altere o evento Click do botão inserir para recuperar os valores dos campos de texto, criar um objeto da classe Pessoa e exibir a lista no datagridview.

```
23 1 reference  
24 private void btnInserir_Click(object sender, EventArgs e)  
25 {  
26  
27     //recupera os valores dos campos de texto e atribui ao objeto  
28     Pessoa pessoa = new Pessoa();  
29     pessoa.nome = txtNome.Text;  
30     pessoa.dataNascimento = txtDataNascimento.Text;  
31     pessoa.cpf = txtCPF.Text;  
32     pessoa.rg = txtRG.Text;  
33  
34  
35     //adiciona o objeto na lista  
36     lista.Add(pessoa);  
37  
38     //exibe a lista na datadridview  
39     dgvPessoas.DataSource = lista;  
40  
41 }  
42 }  
43 }  
44
```

5.6. Execute e teste a inserção dos dados

Form1

Nome:

Data de Nascimento:

RG:

CPF:

	nome	dataNascimento	cpf	rg
▶	Marilda da Silva	20/04/1975	777.777.777-77	22.222.222-99

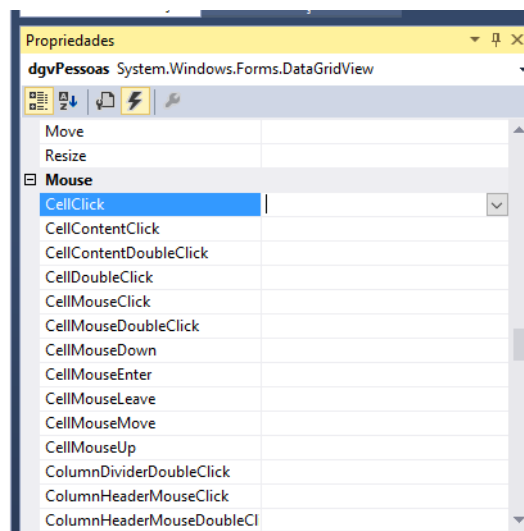
5.7. Após o primeiro registro, tente inserir um outro. Você vai perceber que a datagridview não foi atualizada. Para resolver este problema vamos limpar os registros da datagridview e exibir a listagem atual a cada clique do botão.

```
23 1 reference
24 private void btnInserir_Click(object sender, EventArgs e)
25 {
26
27     //recupera os valores dos campos de texto e atribui ao objeto
28     Pessoa pessoa = new Pessoa();
29     pessoa.nome = txtNome.Text;
30     pessoa.dataNascimento = txtDataNascimento.Text;
31     pessoa.cpf = txtCPF.Text;
32     pessoa.rg = txtRG.Text;
33
34
35     //adiciona o objeto na lista
36     lista.Add(pessoa);
37
38
39     //nula a referencia da fonte de dados (limpa as colunas)
40     dgvPessoas.DataSource = null;
41
42     //exibe a lista na datadridview
43     dgvPessoas.DataSource = lista;
44
45 }
```

5.8. Para melhorar a dinâmica na interface, vamos criar um método que limpa os campos de texto após a inserção na grid.

```
23 1 reference
24 private void btnInserir_Click(object sender, EventArgs e)
25 {
26     //recupera os valores dos campos de texto e atribui ao objeto
27     Pessoa pessoa = new Pessoa();
28     pessoa.nome = txtNome.Text;
29     pessoa.dataNascimento = txtDataNascimento.Text;
30     pessoa.cpf = txtCPF.Text;
31     pessoa.rg = txtRG.Text;
32
33     //adiciona o objeto na lista
34     lista.Add(pessoa);
35
36     //nula a referencia da fonte de dados (limpa as colunas)
37     dgvPessoas.DataSource = null;
38
39     //exibe a lista na datagridview
40     dgvPessoas.DataSource = lista;
41
42     //chama o método de limpar
43     limpaCamposTexto();
44 }
45 1 reference
46 public void limpaCamposTexto()
47 {
48     txtNome.Text = "";
49     txtDataNascimento.Text = "";
50     txtRG.Text = "";
51     txtCPF.Text = "";
52 }
53 }
```

5.9. Para permitirmos que o usuário selecione uma linha da grid e exclua o registro, vamos implementar um evento do datagridview chamado CellClick.



6. Neste evento, vamos recuperar os dados da linha selecionada pelo usuário e preencher as caixas de texto.

```
54 private void dgvPessoas_CellClick(object sender, DataGridViewCellEventArgs e)
55 {
56     /*
57     * CurrentRow - recupera a linha selecionada
58     *
59     * Cells[i] - indica qual célula da grid a partir de 0
60     *
61     * Value - Recuperar o valor da célula
62     *
63     * ToString - Converte para string o valor
64     */
65
66     txtNome.Text = dgvPessoas.CurrentRow.Cells[0].Value.ToString();
67     txtDataNascimento.Text = dgvPessoas.CurrentRow.Cells[1].Value.ToString();
68     txtCPF.Text = dgvPessoas.CurrentRow.Cells[2].Value.ToString();
69     txtRG.Text = dgvPessoas.CurrentRow.Cells[3].Value.ToString();
70 }
71
```

7. Execute a aplicação e teste a seleção de um registro

Nome: Maria

Data de Nascimento: 12/12/2005

RG: 555

CPF: 5555

Inserir

	nome	dataNascimento	cpf	rg
▶	Maria	12/12/2005	5555	555
	Joao	09/6/222	555	444

8. Vamos adicionar uma opção de excluir o registro da lista e da grid, recuperando o índice da linha selecionada, removendo da lista e exibindo a lista atualizada na grid.
- 8.1. Adicione um botão Excluir no formulário.

Exercícios:

1. Crie um novo método no responsável por atualizar e exibir a lista na datagridview e assim evitar a duplicação das linhas abaixo nos dois eventos.

```
//nula a referencia da fonte de dados (limpa as colunas)
dgvPessoas.DataSource = null;

//exibe a lista na datagridview
dgvPessoas.DataSource = lista;
```

2. Acrescente um MessageBox para confirmar a exclusão do registro pelo usuário. Se o usuário confirmar a exclusão, este fato ocorrer, caso contrário a operação não é realizada.
3. Crie um método que valide o preenchimento dos campos de texto, caso o usuário não preencha exiba uma mensagem de orientação ao usuário.