

Ambientes Visuais

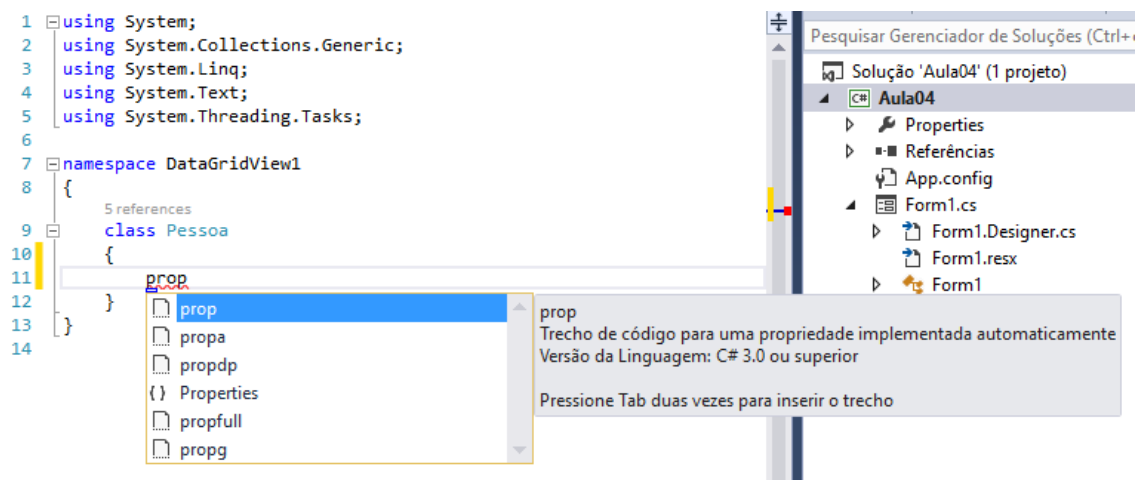
Objetivos:

- Definir classes e objetos para representar os dados de uma Pessoa (cliente),
- Utilizar o componente DataGridView,
- Utilização a coleção List<T>,
- Utilizar os componentes: DateTimePicker, MaskedTextBox

1. Crie um novo projeto no Visual Studio chamado **DataGridView**
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 as propriedades (ou atributos) da classe Pessoa digite o atalho **prop** e a tecla Tab.



- Declare as seguintes propriedades para a classe Pessoa.

```
7 namespace DataGridView
8 {
9     class Pessoa
10     {
11         public string Nome { get; set; }
12         public string DataNascimento { get; set; }
13         public char Sexo { get; set; }
14         public bool NecessidadeEspeciais { get; set; }
15         public string RG { get; set; }
16         public string CPF { get; set; }
17
18     }
19 }
20
21
22
```

- Monte o formulário do projeto e configure a DataGridView de acordo com as propriedades específicas para as colunas.

Form1

Cadastro de Pessoas

Nome:

Data de Nascimento:

Sexo:

Necessidades Especiais: ☐ Sim ☐ Não

RG:

CPF:

DataGridView

	Nome	Data de Nascimento	Sexo	CPF	RG	Necessidades Especiais
*						

Callout Boxes:

- TextBox**
Name: txtNome
- DateTimePicker**
Format: Short
Name: dtpDataNascimento
- ComboBox**
Name: cmbSexo
Items: Masculino, Feminino.
- RadioButton**
Name: rbSim, rbNao
Text: Sim, Não
- MaskedTextBox**
Name: mtxtRG, mtxtCPF
Mask: RG (99.999.999.9)
CPF (999.999.999-99)
- Button**
Name: btnCadastrar
Text: Cadastrar

Propriedades da DataGridView

- Coluna 0 – HeaderText: Nome, Name: colName
- Coluna 1 – HeaderText: Data de Nascimento, Name: colDataNascimento
- Coluna 2 – HeaderText: Sexo, Name: colSexo
- Coluna 3 – HeaderText: CPF, Name: colCPF
- Coluna 4 – HeaderText: RG, Name: colRG
- Coluna 5 – HeaderText: Necessidades Especiais, Name: colNecEspeciais

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

```

10
11 namespace DataGridView
12 {
13     public partial class Form1 : Form
14     {
15         //coleção de objetos da classe Pessoa
16         List<Pessoa> lista = new List<Pessoa>();
17
18         public Form1()
19         {
20             InitializeComponent();
21         }
22     }
23

```

- Crie e implemente um método que será responsável por preencher a DataGridView com a lista de objetos.

```

26 private void preencherDataGridView()
27 {
28
29     //limpa DataGridView
30     dgvClientes.Rows.Clear();
31
32     for (int i = 0; i < lista.Count; i++)
33     {
34
35         string[] nova_linha = new string[] {
36             lista[i].Nome,
37             lista[i].DataNascimento,
38             lista[i].Sexo == 'F' ? "Feminino" : "Masculino",
39             lista[i].CPF, lista[i].RG,
40             lista[i].NecessidadeEspeciais == false ? "Não" : "Sim"
41         };
42
43
44         dgvClientes.Rows.Add(nova_linha);
45     }
46 }
47

```

- Crie um evento Click no botão Cadastrar e implemente da seguinte forma...

```

49 private void btnCadastrar_Click(object sender, EventArgs e)
50 {
51
52     Pessoa pessoa = new Pessoa();
53     pessoa.Nome = txtNome.Text;
54
55     // 0 - Masculino; 1 - Feminino
56     if (cmbSexo.SelectedIndex == 0)
57     {
58         pessoa.Sexo = 'M';
59     }
60     else
61     {
62         pessoa.Sexo = 'F';
63     }
64
65     pessoa.DataNascimento = dtpDataNascimento.Text;
66
67     if (rbSim.Checked)
68     {
69         pessoa.NecessidadeEspeciais = true;
70     }else
71     {
72         pessoa.NecessidadeEspeciais = false;
73     }
74
75     pessoa.RG = mtxtRG.Text;
76     pessoa.CPF = mtxtCPF.Text;
77
78     //adiciona na lista de objetos
79     lista.Add(pessoa);
80
81     //chama o método
82     preencherDataGridView();
83
84
85 }

```

- Execute aplicação e clique no botão Cadastrar.

Cadastro de Pessoas

Nome: Carlos Eduardo

Data de Nascimento: 03/06/1986

Sexo: Masculino

Necessidades Especiais: ☐ Sim ☒ Não

RG: 99.999.999-9

CPF: 888.888.888-88

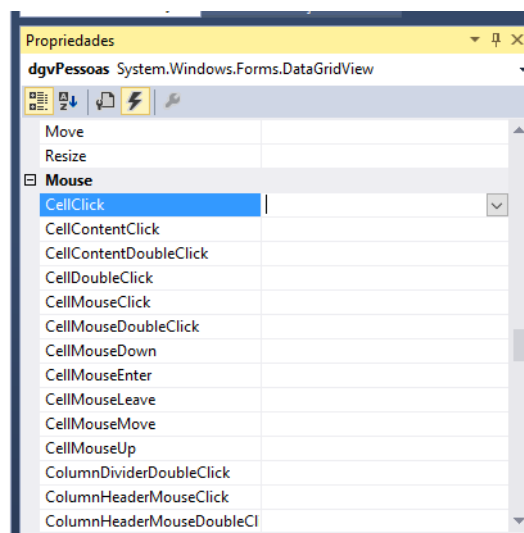
Cadastrar

	Nome	Data de Nascimento	Sexo	CPF	RG	Necessidades Especiais
»	Carlos Eduardo	03/06/1986	Masculino	888.888.888-88	99.999.999-9	Não

- Para melhorar a dinâmica na interface, vamos criar um método que limpa os campos de texto após a inserção na grid. Chame este método após a chamada do método **preencherDataGridView**.

```
85     public void limpaCampos()
86     {
87         //limpa campo de texto
88         txtNome.Clear();
89         //volta para data atual do sistema
90         dtpDataNascimento.Value = DateTime.Now;
91         //limpa combobox
92         cmbSexo.Text = "";
93         //retira o checked
94         if (rbSim.Checked)
95         {
96             rbSim.Checked = false;
97
98         }else
99         {
100             rbNao.Checked = false;
101         }
102         //limpa campos de texto
103         mtxtCPF.Clear();
104         mtxtRG.Clear();
105
106     }
```


- Para permitirmos que o usuário selecione uma linha da grid e exclua o registro, vamos implementar um evento na DataGridView chamado **CellClick**.



- Adicione um botão Excluir no formulário.

Cadastro de Pessoas

Nome:

Data de Nascimento: 

Sexo:

Necessidades Especiais: ☐ Sim ☐ Não

RG:

CPF:

	Nome	Data de Nascimento	Sexo	CPF	RG	Necessidades Especiais
*						

- Implemente o evento como o exemplo a seguir.

```

147 private void btnExcluir_Click(object sender, EventArgs e)
148 {
149     //recupera o indice da linha selecionada
150     int indiceLinha = dgvClientes.CurrentRow.Index;
151
152     //remove da lista
153     lista.RemoveAt(indiceLinha);
154
155     //chama o método limpar campos
156     limpaCampos();
157
158     //limpa a grid
159     dgvClientes.Rows.Clear();
160
161     //atualiza a grid
162     preencherDataGridView();
163 }

```

7. Execute e teste a exclusão de um registro.

Exercícios:

1. Acrescente um MessageBox para confirmar a exclusão do registro pelo usuário. Se o usuário confirmar a exclusão, este fato de fato ocorrer, caso contrario a operação não é realizada.
2. Crie um método que valide o preenchimento dos campos de texto, caso o usuário não preencha exiba uma mensagem de orientação ao usuário.
3. Adicione uma outra classe no projeto chamada Usuario com os atributos (ou propriedades) login e senha. Implemente nesta classe um método chamado autenticar que retorna true caso o login e a senha fornecidos pelo usuário corresponderem a "joao", "123" . Caso contrario, o método deve retornar false. Monte um formulário para entrada destes dados, se o usuário autenticar com sucesso o formulário de cadastro deve ser exibido e o de login fechado.