



Habilitação Profissional Técnica de Nível Médio de **TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

DESENVOLVIMENTO DE SISTEMAS

TabControl

TabControl

A função do **tabcontrol**, é permitir a exibição de vários controles utilizando o mesmo espaço. Em cada **Tab (aba)** podemos incluir qualquer tipo de componente disponível no ambiente do Visual Studio. Por exemplo, em um cadastro poderíamos dividir as informações em: Dados Pessoais, Endereços, Filiação, Telefones, etc..., neste caso utilizamos este componente com várias abas, cada uma com suas informações específicas.

Abas (Tab)

Área de inserção de
componentes



TabControl

Os componentes inseridos á uma Tab (aba), quando clicamos em outra Tab (aba) não serão exibidos os componentes da anterior, como mostra a figura abaixo:

tabPage1 tabPage2

Código:

Nome:

Adicionar Inserir

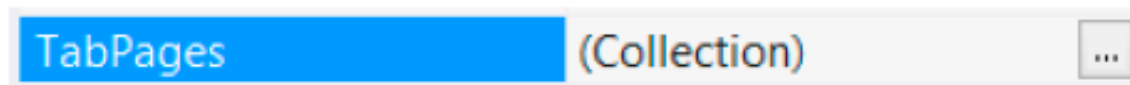
tabPage1 tabPage2

Código	Nome
--------	------

Eliminar

TabControl

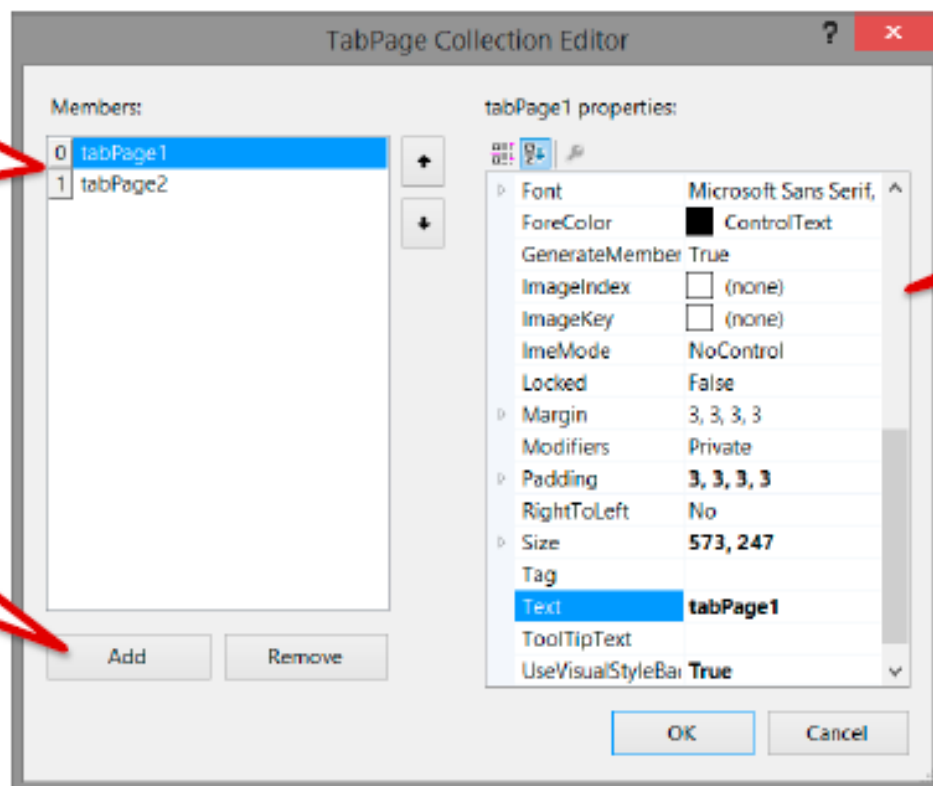
Para definir as Tabs (abas) do tabcontrol devemos utilizar a propriedade **TabPage**, na caixa de propriedades.



Após clicar no botão ... aparecerá a janela abaixo:

Lista as Tabs (abas) do TabControl.

Adicionar uma Tab (aba) ao TabControl.



Lista as propriedades de cada Tab (aba) selecionada.

TabControl – Propriedades

Text → Definir o texto que será exibido na etiqueta da Tab (aba).

Name → Definir o nome do componente.

UseVisualStyleBackColor → Se estiver false, usará a cor selecionada na propriedade BackColor, caso contrário, usará a cor Branca de fundo por definição da linguagem.

BorderStyle → Definir a borda de uma Tab (aba).

Para inserir os campos dentro de uma Tab (aba), selecione o componente na ToolBox, arraste até a área da Tab (aba) desejada, ou selecione o componente desejado e mova-o até a Tab (aba).

DataGridView



DataGridView

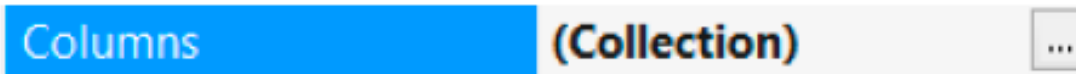
A função do **datagridview**, é permitir a exibição de dados em formato tabular, ou seja, linhas e colunas de informação. Esta classe substitui a classe `DataGrid`, presente no `Windows Forms 1.0`, com a adição de novas funcionalidades.


É possível usar um controle `DataGridView` para exibir dados com ou sem uma fonte de dados (data source). Sem a fonte de dados, podemos criar linhas e colunas que contém dados e adicioná-las diretamente ao `DataGridView`. Alternativamente, podemos definir valores para as propriedades `DataSource` e `DataMember`. Isso liga o `DataGridView` a uma fonte de dados e automaticamente já preenche suas linhas e colunas com os dados desejados.

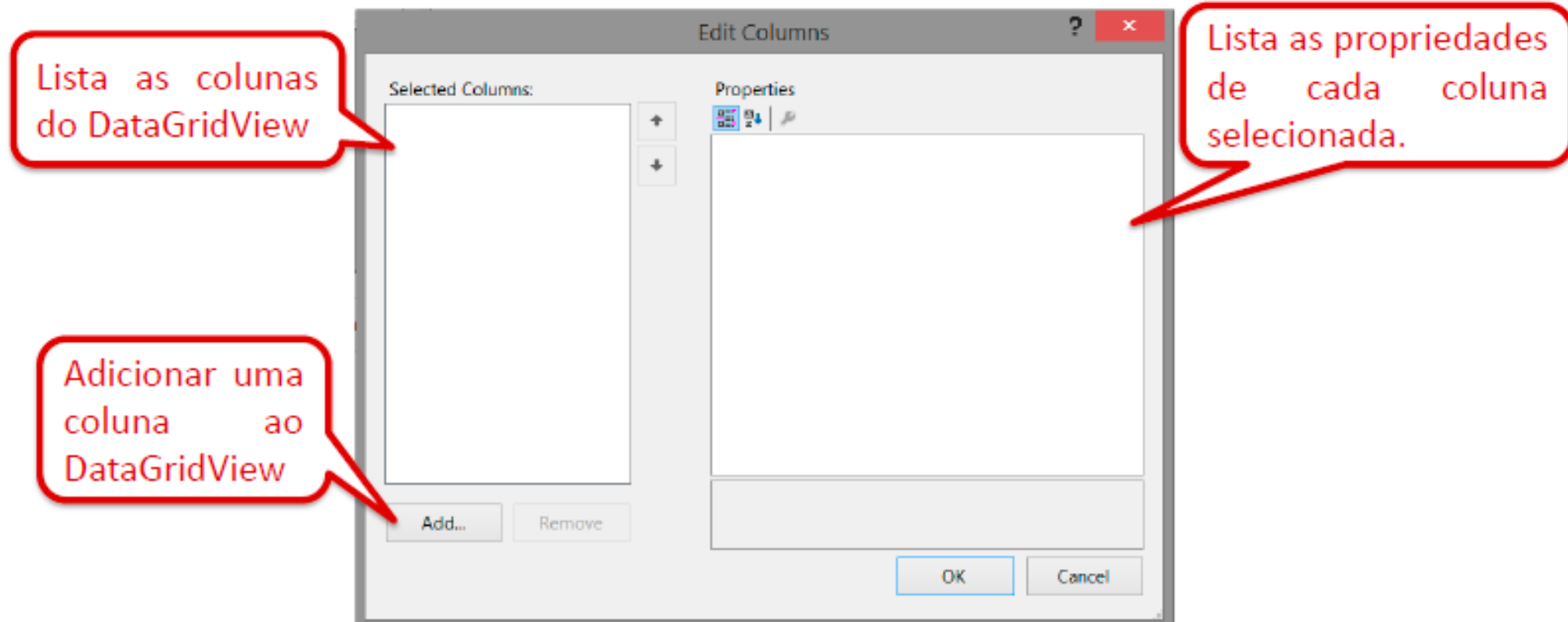


DataGridView

Para definir as colunas do datagridview devemos utilizar a propriedade **Columns**, na caixa de propriedades.



Após clicar no botão  aparecerá a janela abaixo:



DataGridView

Após clicar no botão **Add**, a seguinte tela aparecerá:

Add Column ? x

☐ Databound column

Columns in the DataSource

☒ Unbound column

Name:

Type:

Header text:

☒ Visible ☐ Read Only ☐ Frozen

Nome da coluna a ser inserida no DataGridView.

Tipos de colunas

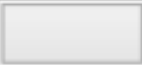
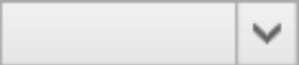

Descrição do cabeçalho da coluna.

- DataGridViewTextBoxColumn
- DataGridViewButtonColumn
- DataGridViewCheckBoxColumn
- DataGridViewComboBoxColumn
- DataGridViewImageColumn
- DataGridViewLinkColumn
- DataGridViewTextBoxColumn

DataGridView

Tipos de Colunas do DataGridView

1. **DataGridViewButtonColumn** → Inserir um Button (Botão) no grid.
2. **DataGridViewCheckBoxColumn** → Inserir um CheckBox no grid.
3. **DataGridViewComboBoxColumn** → Inserir um ComboBox no grid.
4. **DataGridViewImageColumn** → Inserir uma Image (Imagem) no grid.
5. **DataGridViewLinkColumn** → Inserir uma Label Link no grid.
6. **DataGridViewTextBoxColumn** → Inserir uma caixa de texto (TextBox) no grid.

	1	2	3	4	5	6
	Botão	CheckBox	ComboBox	Imagem	Link	TextBox
*		<input type="checkbox"/>				

DataGridView - Propriedades

AllowUserToAddRows → Permitir o usuário adicionar linhas no DataGridView.

AllowUserToDeleteRows → Permitir o usuário deletar (apagar) linhas no DataGridView.

AllowUserToOrderRows → Permitir o usuário ordenar as colunas em ordem alfabética ou numérica.

AllowUserToResizeColumns → Permitir o usuário redimensionar as colunas do DataGridView.

AllowUserToResizeRows → Permitir o usuário redimensionar as linhas do DataGridView.

RowHeaderVisible → Permitir exibir ou inibir a primeira coluna do DataGridView.

Para saber mais sobre as propriedades do DataGridView acesse a página:

<http://www.microsoft.com/brasil/msdn/Tecnologias/vs2005/DataGridView.aspx>

DataGridView – Adicionando Linha(s)

Adicionando linha(s) através da programação

```
dataGridView1.Rows.Add(<coluna1>,<coluna2>, ... , <colunaN>);
```

Nome (**Propriedade Name**) do componente.

Método para adicionar uma linha dentro do componente.

<coluna...> → A quantidade de parâmetros (colunas) vai depender da quantidade de colunas definidas no componente (Design).

Exemplo:

```
dataGridView1.Rows.Add("1", "Computador");  
dataGridView1.Rows.Add("2", "Impressora");  
dataGridView1.Rows.Add("3", "Mouse");  
dataGridView1.Rows.Add("4", "Monitor");
```

DataGridView – Inserindo Linha(s)

Inserindo linha através da programação

```
dataGridView1.Rows.Insert(<linha>,<coluna1>,<coluna2>, ... , <colunaN>);
```

Método para inserir uma linha dentro do componente.

<linha> → Indicar o número da linha na qual irá inserir as novas informações.

<coluna...> → A quantidade de parâmetros (colunas) vai depender da quantidade de colunas definidas no componente (Design).

Exemplo:

```
dataGridView1.Rows.Insert(2,"999", "Scanner");  
dataGridView1.Rows.Insert(5,"23", "Teclado");  
dataGridView1.Rows.Insert(7,"57", "Toner");  
dataGridView1.Rows.Insert(9,"11", "Multifuncional");
```

DataGridView – Removendo Linha

Removendo linhas através da programação

```
dataGridView1.Rows.RemoveAt(<indice-da-linha>);
```

Método para remover
uma linha do
componente.

<indice-da-linha> → Índice
da linha que será
eliminada.

DataGridView

Podemos manipular as linhas através das seguintes propriedades:

- ✓ **dataGridView1.CurrentRow.Cells[0].RowIndex** → Retorna a linha selecionada.
- ✓ **dataGridView1.CurrentRow.Selected** → Retorna true/false, verdadeiro/falso, se a linha está selecionada.
- ✓ **dataGridView1.CurrentRow.Cells[0].Value** → Retorna o valor (conteúdo) da célula, da coluna 0 (Zero) da linha selecionada.

Caso precise ordenar uma coluna clique sobre o cabeçalho da coluna desejada. Lembrando que para isso acontecer, a propriedade **AllowUserToOrderRows** deve estar **true**.

O evento default para o DataGridView é o **CellContentClick**, ou seja, quando selecionamos uma linha, será executado o código de programação que estiver dentro.

```
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{

}
```

Exemplo – DataGridView e TabControl

The screenshot shows a Windows application window titled "DataGridView e TabControl". The window has a standard Windows title bar with minimize, maximize, and close buttons. Inside the window, there is a TabControl with two tabs: "Entrada de Dados" (selected) and "Consultar Dados". The "Entrada de Dados" tab contains a dashed rectangular border enclosing the input fields and buttons. Inside this border, there are two text labels, "Código:" and "Nome:", each followed by a text input box. Below the input boxes, there are two buttons: "Adicionar" and "Inserir". Outside the dashed border, at the bottom of the window, there are two more buttons: "Limpar" and "Sair". The "Sair" button is highlighted with a blue border.

Exemplo – DataGridView e TabControl

The screenshot shows a Windows application window titled "DataGridView e TabControl". Inside the window, there is a TabControl with two tabs: "Entrada de Dados" and "Consultar Dados". The "Consultar Dados" tab is currently selected and active. Within this tab, there is a DataGridView control. The DataGridView has two columns: "Código" and "Nome". The body of the DataGridView is currently empty. Below the DataGridView, there is a button labeled "Eliminar". At the bottom of the application window, there are two more buttons: "Limpar" and "Sair".

Exemplo – Código Fonte

```
private void btnSair_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

```
private void btnAdicionar_Click(object sender, EventArgs e)
{
    dgvDados.Rows.Add(txtCodigo.Text, txtNome.Text);
}
```

```
private void btnEliminar_Click(object sender, EventArgs e)
{
    dgvDados.Rows.RemoveAt(dgvDados.CurrentRow.Cells[0].RowIndex);
}
```

Exemplo – Código Fonte

```
private void btnInserir_Click(object sender, EventArgs e)
{
    dgvDados.Rows.Insert(1, txtCodigo.Text, txtNome.Text);
}
```

```
private void btnLimpar_Click(object sender, EventArgs e)
{
    txtCodigo.Text = "";
    txtNome.Text = "";
    dgvDados.Rows.Clear();
}
```

Método para limpar
(Eliminar) todas as linhas
do DataGridView.