

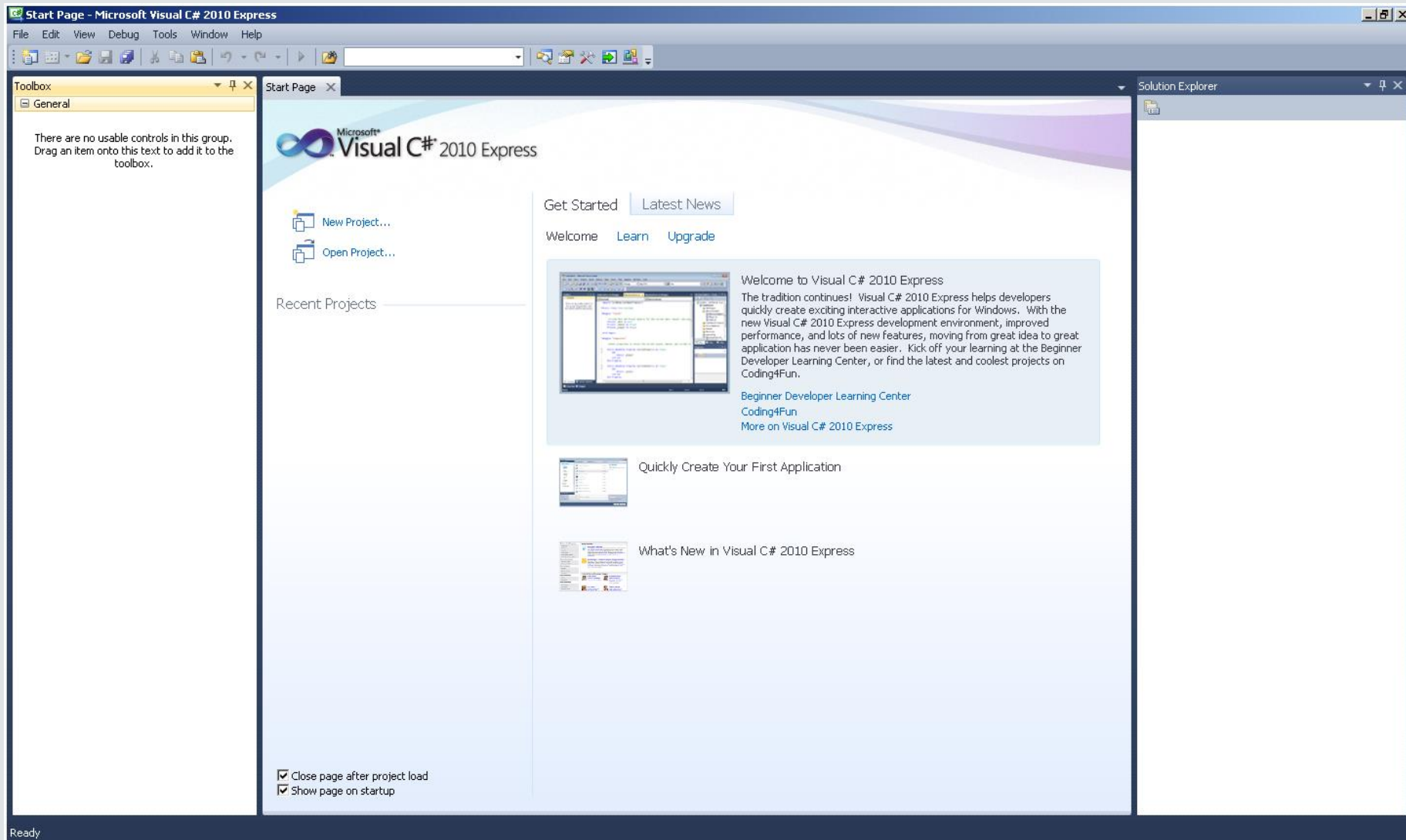
FORMULÁRIOS C#

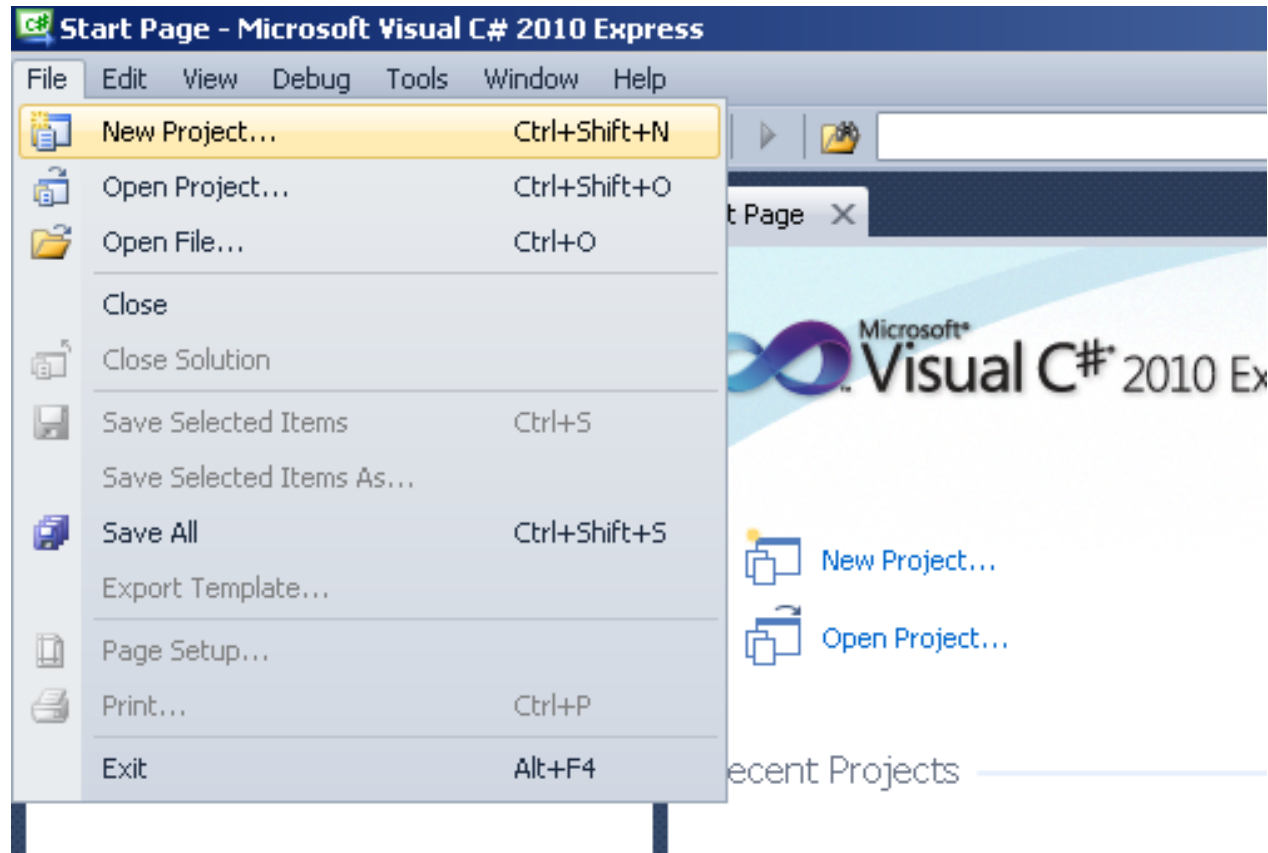
PROF. ME. HÉLIO ESPERIDIÃO



Universidade do Vale do Paraíba

Eng. Computação

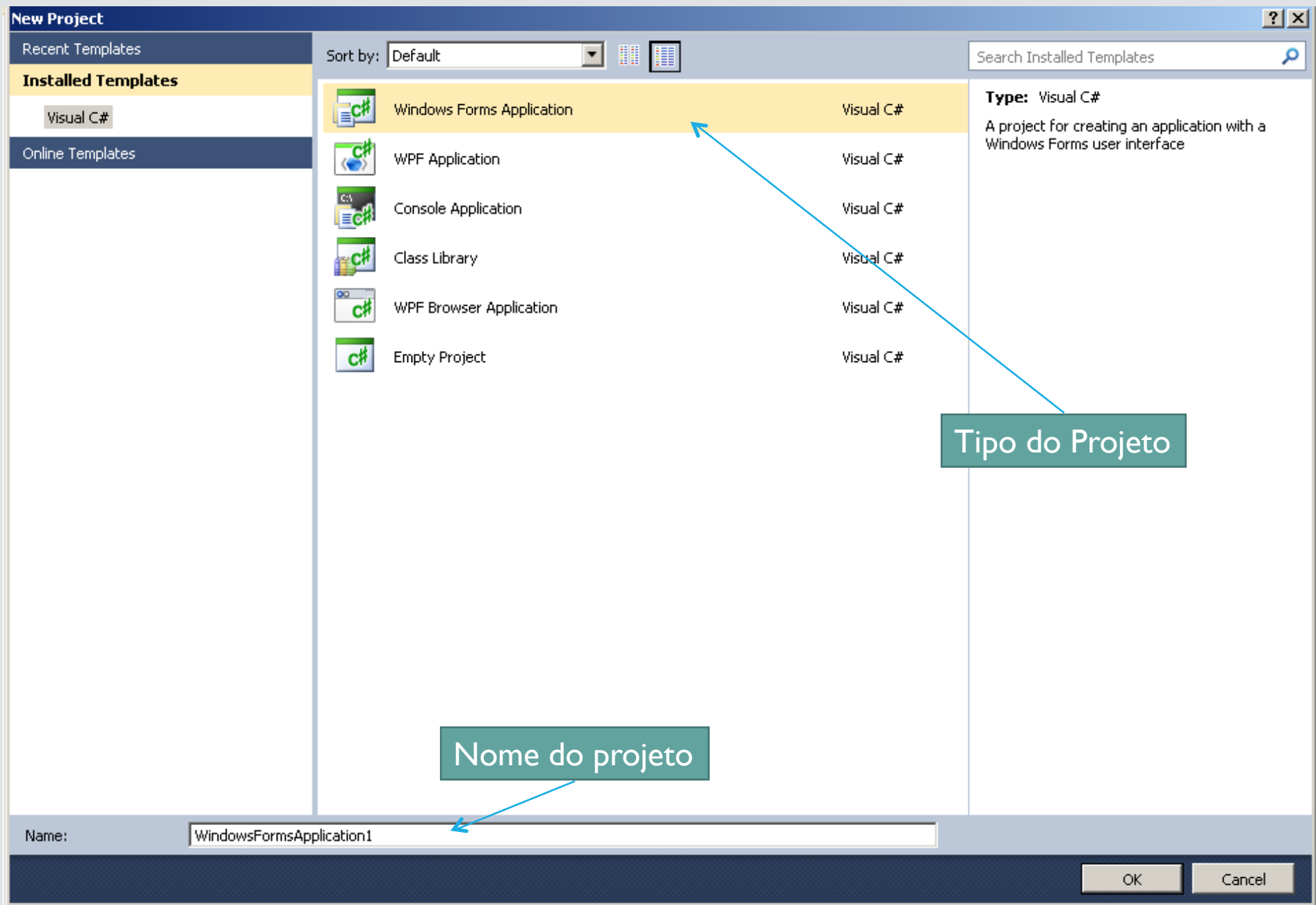




CRIANDO PROJETOS

Menu File

New Project



O VISUAL STUDIO FAZ POR VOCÊ...

1

Quando você inicia um novo projeto, assim que você o salva, o IDE cria os arquivos Form1.cs, Form1.Designer.cs, e Program.cs

2

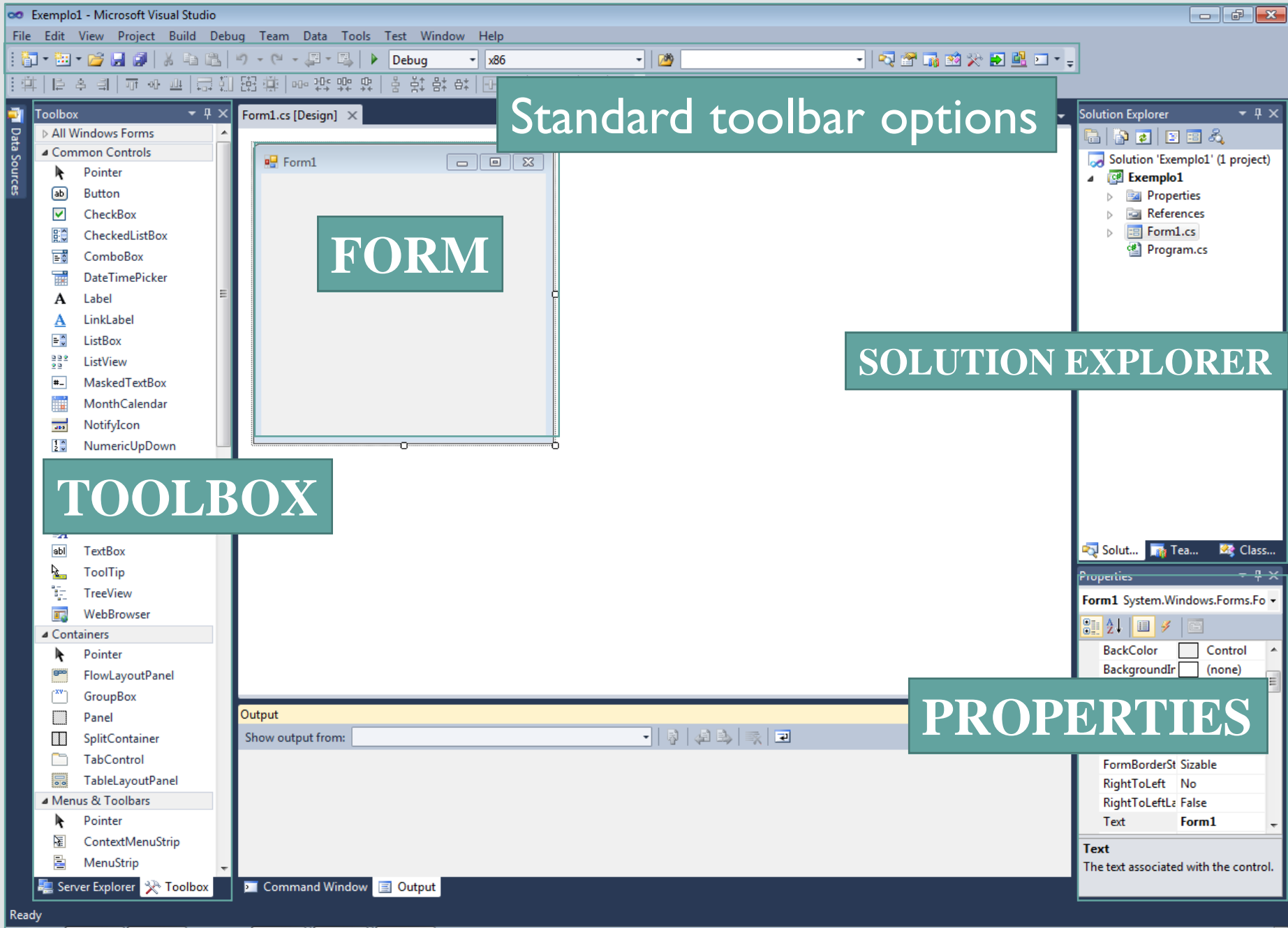
Form1.cs: contém o código C# que define o comportamento do formulário.

3

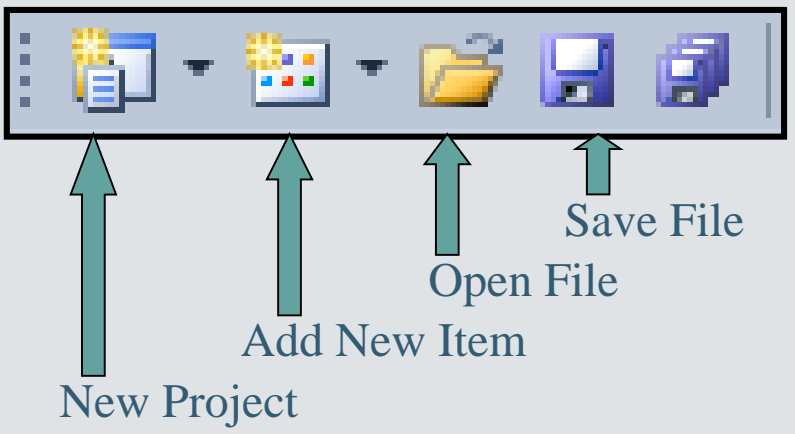
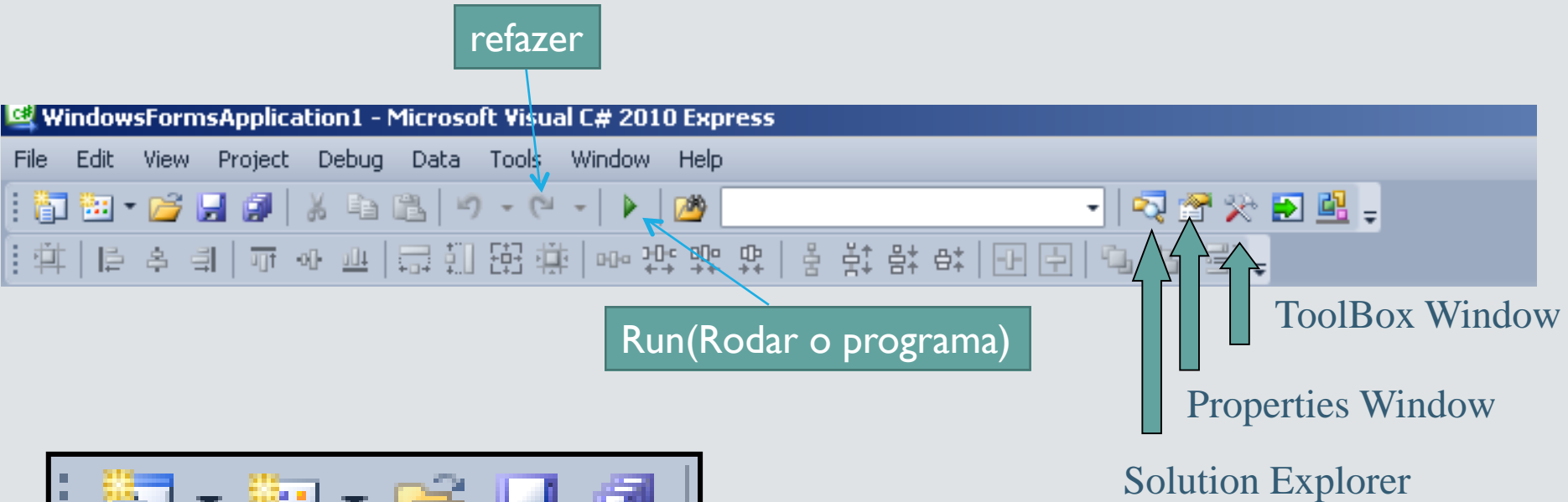
Program.cs : Este possui o código que inicia o programa e exibe o formulário.

4

Form1.Designer.cs : O código que define o formulário e seus objetos está aqui.

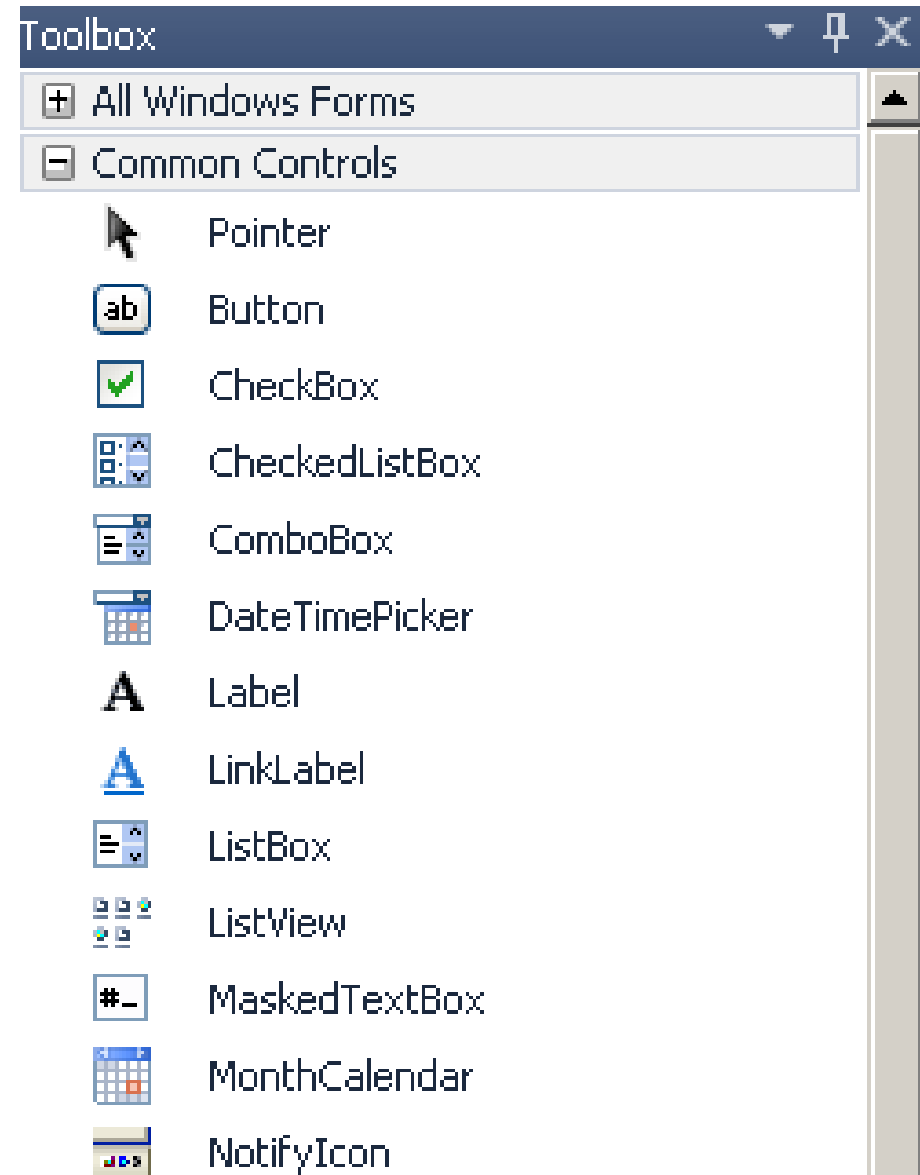


STANDARD TOOLBAR OPTIONS



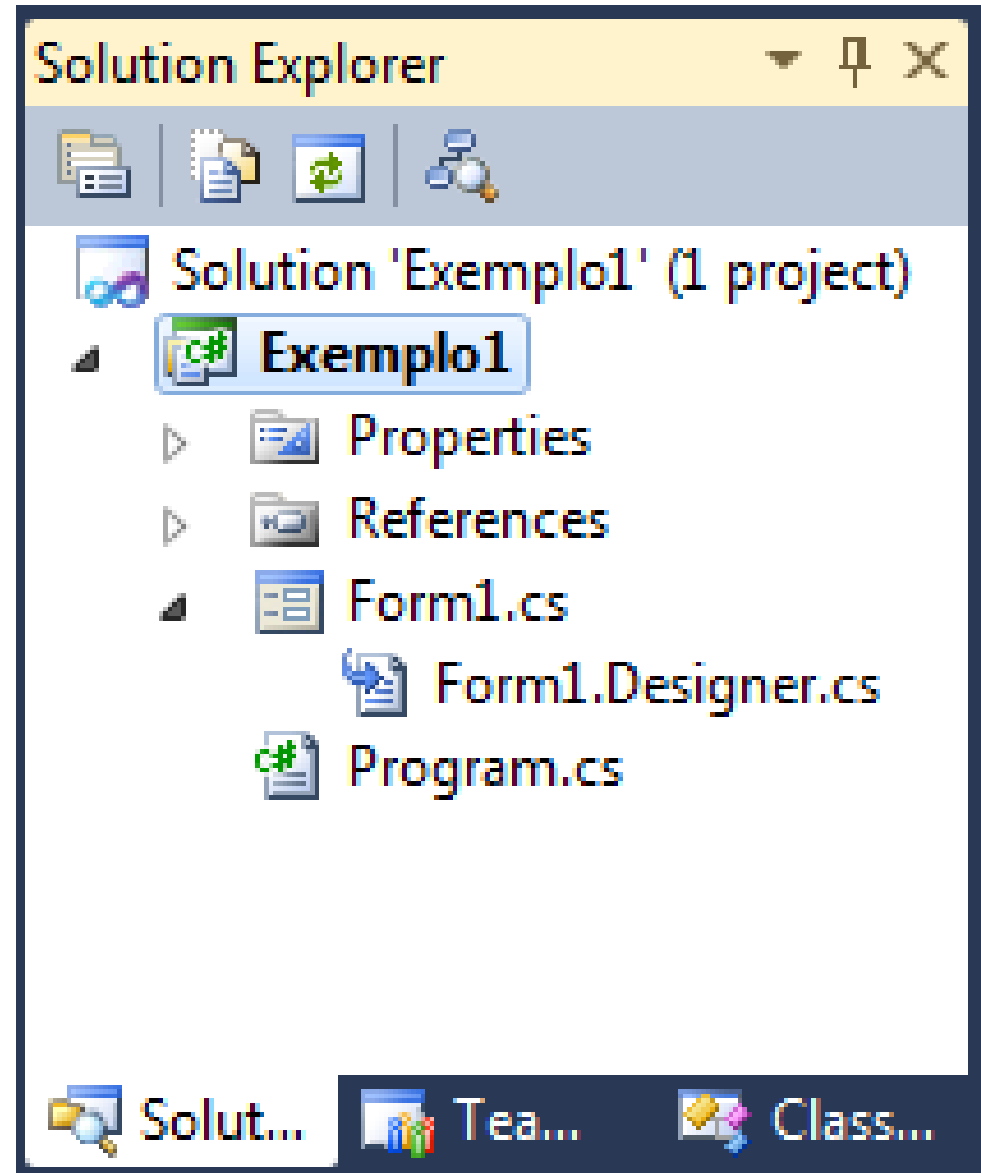
TOOLBOX

- é onde estão localizados todos a maioria dos componentes visuais que podem ser utilizados para criar aplicações.
- É normalmente localizado no lado esquerdo da janela, mas sua posição pode ser personalizada de acordo com as preferências do programador.



SOLUTION EXPLORER

- Esta caixa de diálogo é usada para gerenciar os arquivos que foram criados no nosso projeto.

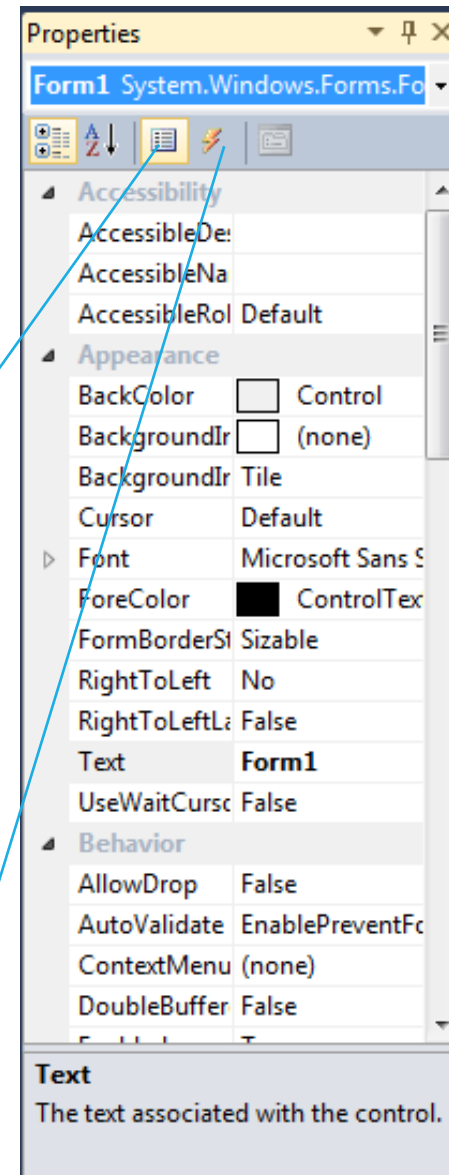


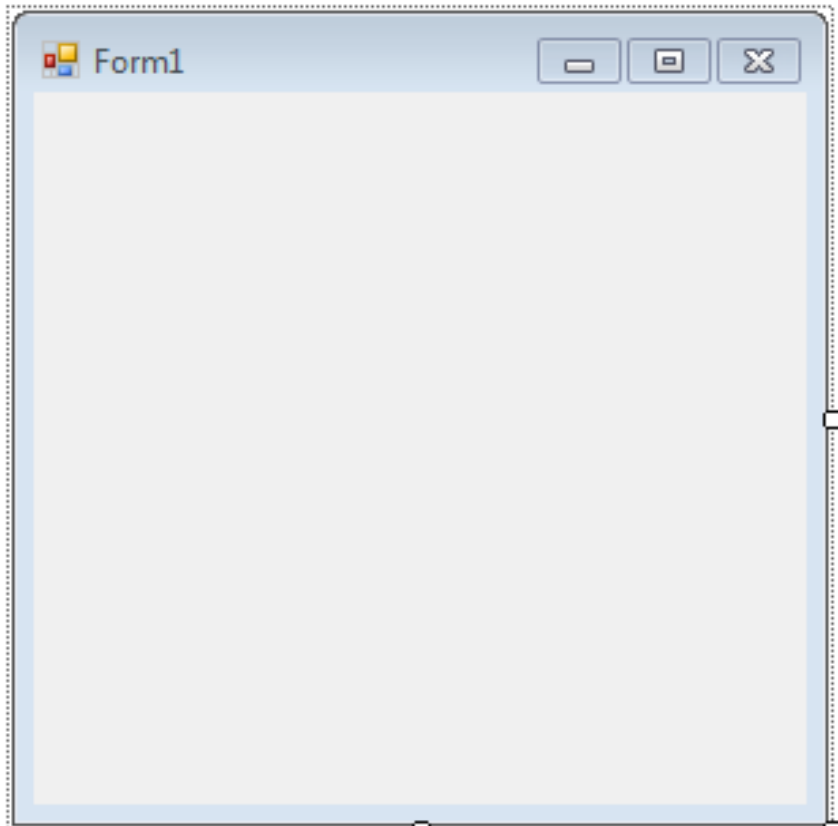
PROPERTIES

- Esta caixa de diálogo é muito importante pois é nela que configuraremos os objetos gráficos adicionados no nosso programa.
- Ela é dividida em 2 partes: Properties e Events

Para exibir as propriedades do objeto selecionado

Para exibir os eventos do objeto selecionado

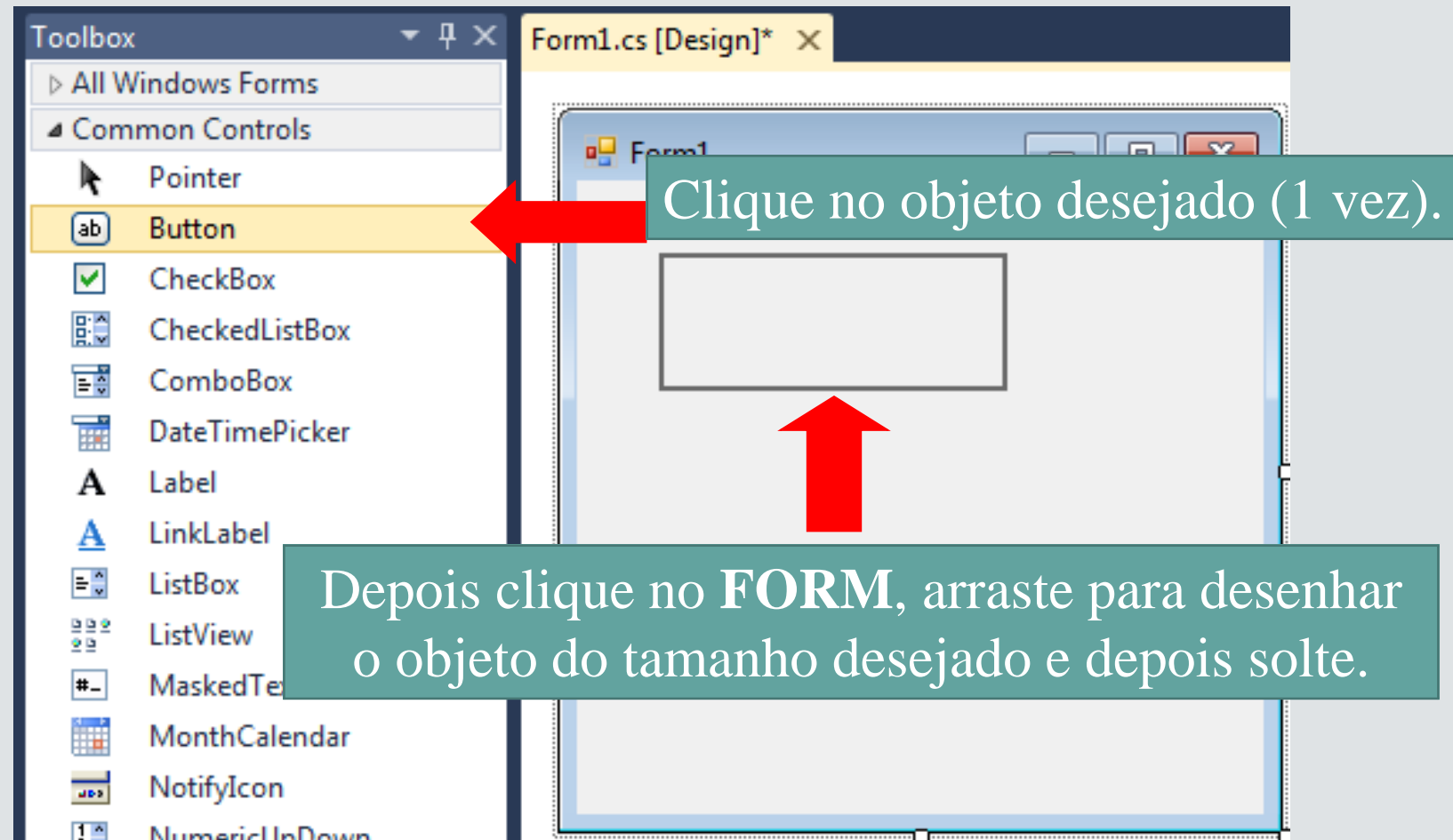




JANELAS

JANELAS TAMBÉM SÃO CONHECIDAS
COMO FORMS OU FORMULÁRIOS

ADICIONANDO CONTROLES AO FORM

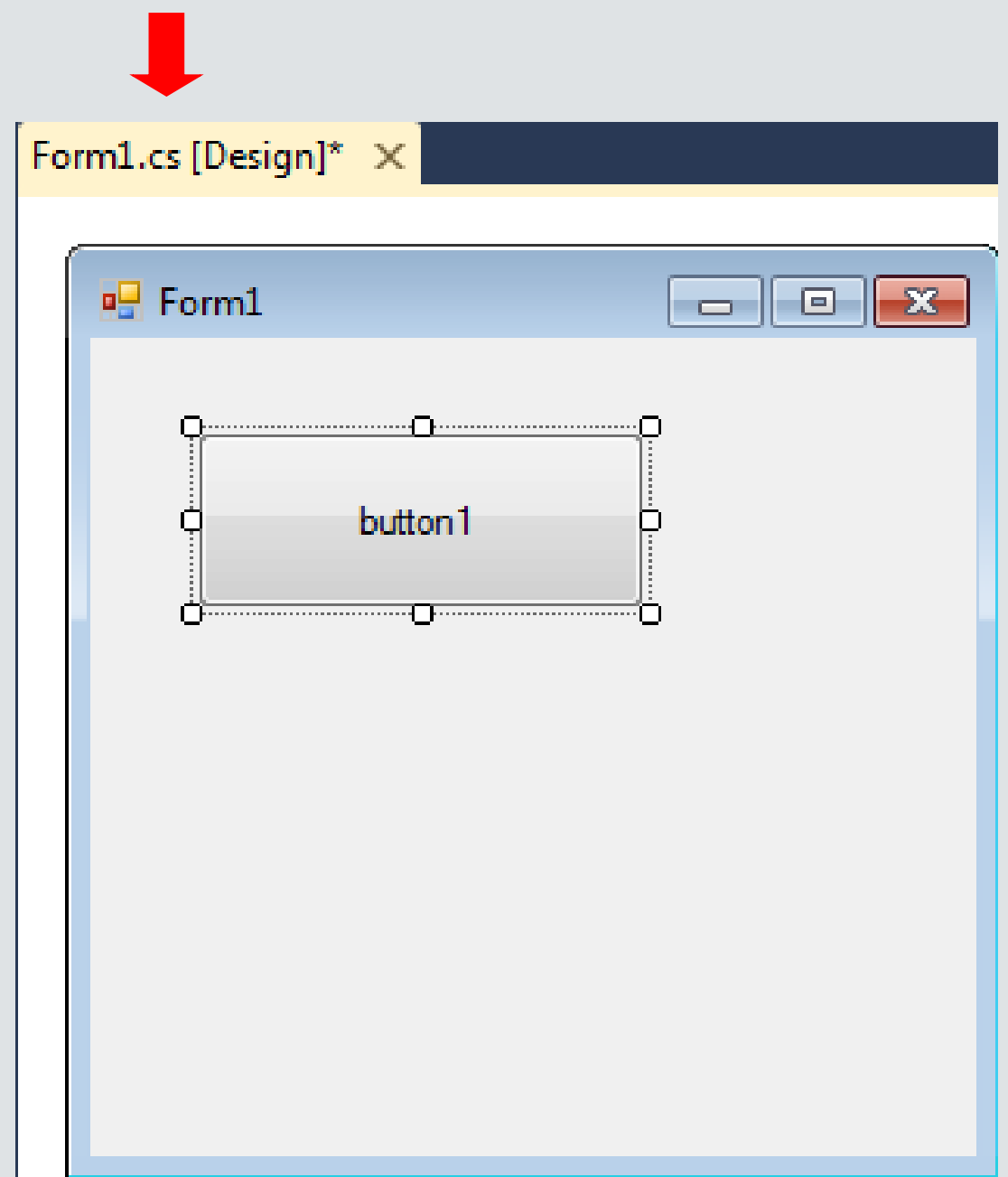


O * SIGNIFICA QUE O SEU PROGRAMA AINDA NÃO FOI SALVO

Todos objetos adicionados ao FORM podem ter alteradas as propriedades NAME.

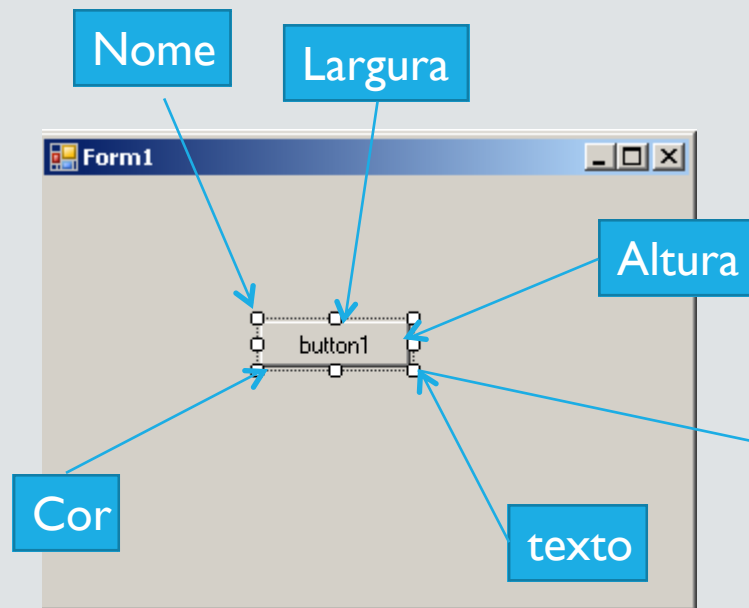
A propriedade NAME serve para identificar o objeto a nível do programador (“VARIÁVEL”).

A propriedade TEXT serve para identificar o objeto a nível do usuário (RÓTULO).

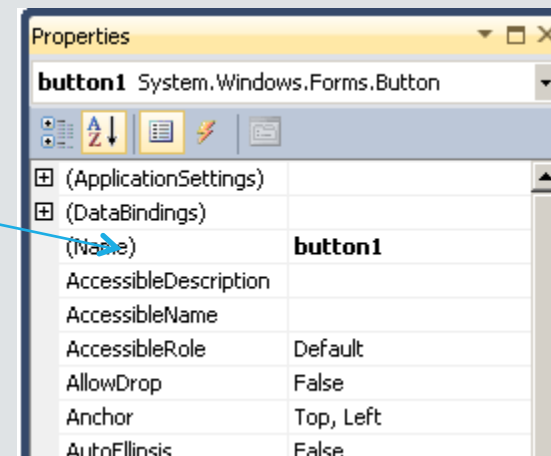


PROPRIEDADES

- Todos os componentes visuais possuem uma série de propriedades associadas a ele.

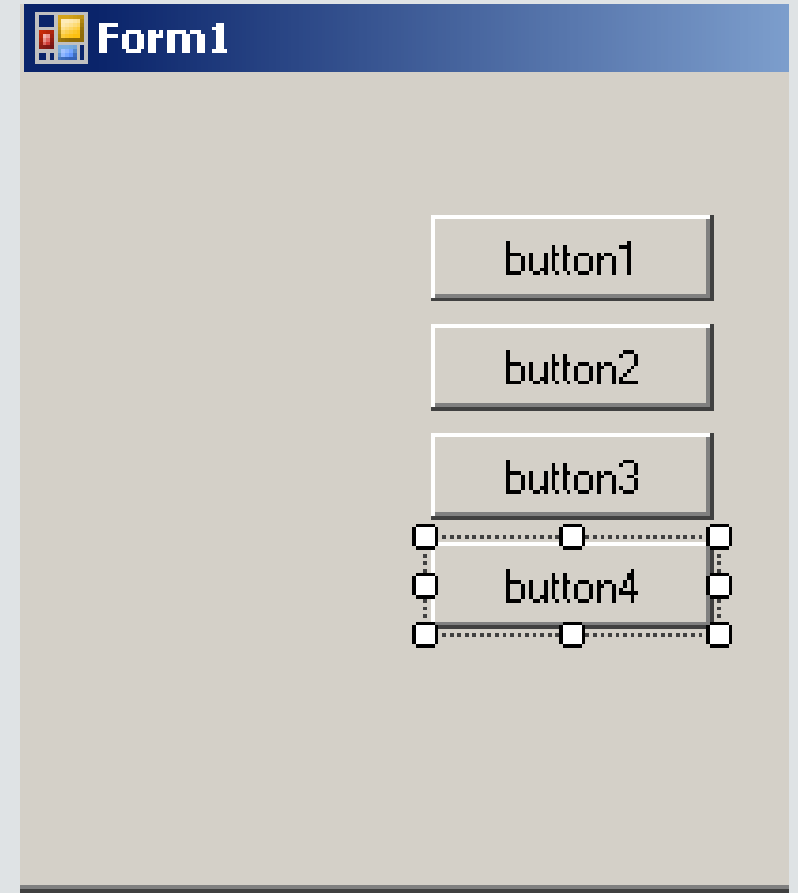


- Todas essas propriedades podem ser alteradas por meio da barra de propriedades ou via programação.



A PROPRIEDADE NAME.

- Quando temos muitos componentes em uma janela é difícil lembrar porque eles foram adicionados ao formulários.
- Mudar o nome dos componentes visuais é uma tarefa que deve ser realizada logo após adicionar o objeto em tela.



A PROPRIEDADE NAME.

- Existem alguns padrões que são adotados para nomeação de componentes.
- O mais utilizado sugere que identifiquemos no nome do componente o seu tipo.
- Essa identificação é realizada por meio de prefixos.
 - Exemplo: **button** : **btnCadastro**.
 - Exemplo: **textBox** : **txtNome**

Prefixo que identifica o nome do componente

1º Letra deve ser maiúscula

O resto do nome diz respeito a funcionalidade do componente na tela.

REGRAS PARA NOMEAR COMPONENTES

Componente	Prefixo	Exemplo
Form	frm	frmEntry
Label	lbl	lblHelpMessage
LinkLabel	Ink	InkEmail
Button	btn	btnExit
TextBox	txt	txtLastName
Menu	mnu	mnuFileOpen
CheckBox	chk	chkReadOnly
RadioButton	rad	radType
GroupBox	grp	grpActions
PictureBox	pic	picIcon

REGRAS PARA NOMEAR

Componente	Prefixo	Exemplo
Panel	pnl	pnlGroup
DataGrid	grd	grdQueryResult
ListBox	lst	lstPolicyCodes
CheckedListBox	clb	clbOptions
ComboBox	cbo	cboEnglish
ListView	lvw	lvwHeadings
TreeView	tre	treOrganization
TabControl	tbc	tbcOptions
DateTimePicker	dtp	dtpPublished
MonthCalendar	mcl	mclPeriod

REGRAS PARA NOMEAR COMPONENTES

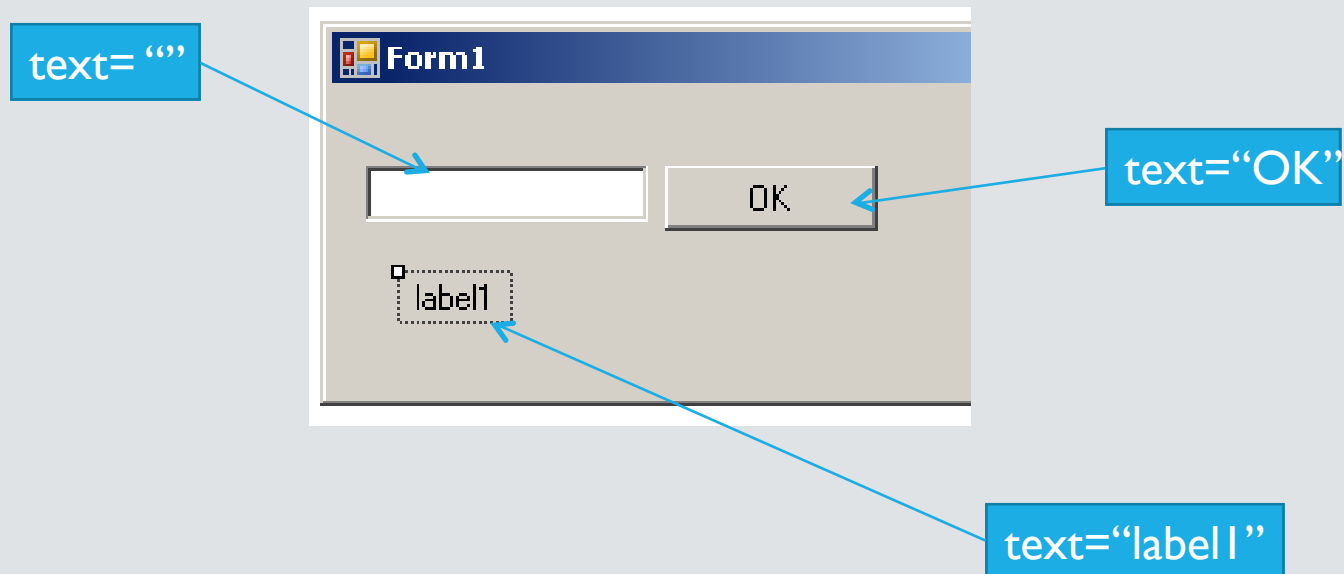
Componente	Prefixo	Exemplo
HScrollBar	hsb	hsbMove
VScrollBar	vsb	vsbMove
Timer	tmr	tmrAlarm
Splitter	spt	sptDivision
DomainUpDown	upd	updPages
NumericUpDown	nud	nudPieces
TrackBar	trb	trblIndex
ProgressBar	prg	prgLoadFile
RichTextBox	rtf	rtfReport

REGRAS PARA NOMEAR COMPONENTES

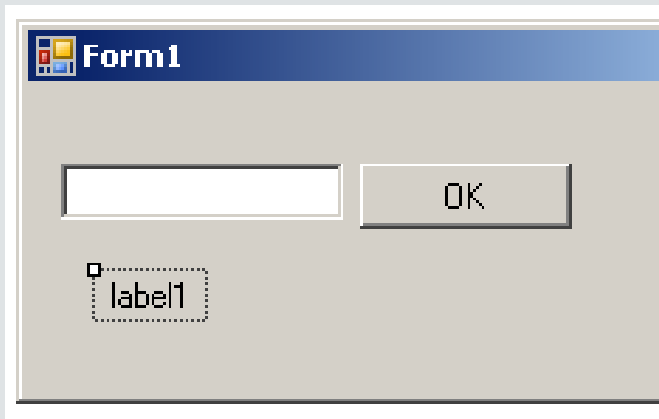
Componente	Prefixo	Exemplo
ImageList	ils	ilsAllIcons
HelpProvider	hlp	hlpOptions
ToolTip	tip	tipIcons
ContextMenu	cmn	cmnOpen
ToolBar	tlb	tlbActions
StatusBar	sta	staDateTime
NotifyIcon	nti	ntiOpen
OpenFileDialog	ofd	ofdImage
SaveFileDialog	sfd	sfdImage

A PROPRIEDADE TEXT

- A propriedade `text` determina o texto que será exibido por componentes visuais.
- Veja abaixo a propriedade `text` dos componentes:



O EVENTO CLICK



Ao efetuar o click duplo no botão “OK”
você será direcionado para a área de
edição de código.

Nome do componente

Evento que será programado

```
private void BtnTransferir_Click(object sender, EventArgs e)
{
}
}
```

MUDANDO A PROPRIEDADE TEXT VIA PROGRAMAÇÃO

Nome do componente

propriedade

Novo valor

```
private void BtnTransferir_Click(object sender, EventArgs e)
{
    txtNome.Text = "OLA Mundo";
    lblNomeDigitado.Text = "Oi Mundo 2";
}
```

O QUE ACONTECE NO CÓDIGO ABAIXO?

```
private void BtnTransferir_Click(object sender, EventArgs e)
{
    lblNomeDigitado.Text = txtNome.Text;
}
```


VARIÁVEIS LOCAIS.

Cria uma variável local do tipo string

Variáveis locais só podem ser utilizadas onde foram criadas

```
private void BtnTransferir_Click(object sender, EventArgs e)
{
    string nome = "";

    int nota1;

    int nota2 = 7;
}
```

Tipo do dado

Valor de inicialização é opcional.

Nome da variável

O QUE ACONTECE NO CÓDIGO ABAIXO?

```
private void BtnTransferir_Click(object sender, EventArgs e)
{
    string nome = "";
    nome = txtNome.Text;
    lblNomeDigitado.Text = nome;
}
```

QUAL O PROBLEMA DO CÓDIGO ABAIXO?

```
private void BtnTransferir_Click(object sender, EventArgs e)
{
    int nota1;
    int nota2;
    float media;
    nota1 = txtNota1.Text;
    nota2 = txtNota2.Text;
    media = (nota1 + nota2) / 2;
    lblMedia.Text = media;
}
```

O PROBLEMA É O TIPO DE DADOS

- A variável nota é do tipo **int**.
- E o texto que está presente na caixa de textos é do tipo **string**.
- Os tipos de dados são **incompatíveis**.
 - É necessário fazer uma conversão entre tipos.
 - Converter o texto do tipo **string** para um numero do tipo **int**.

PASSO A PASSO.

Declara as variáveis para calculo da média.

```
private void BtnTransferir_Click(object sender, EventArgs e)
{
    int nota1;
    int nota2;
    float media;
    nota1 = Convert.ToInt32(txtNota1.Text);
    nota2 = Convert.ToInt32(txtNota2.Text);
    media = (nota1 + nota2) / 2;
    lblMedia.Text = Convert.ToString(media);
}
```

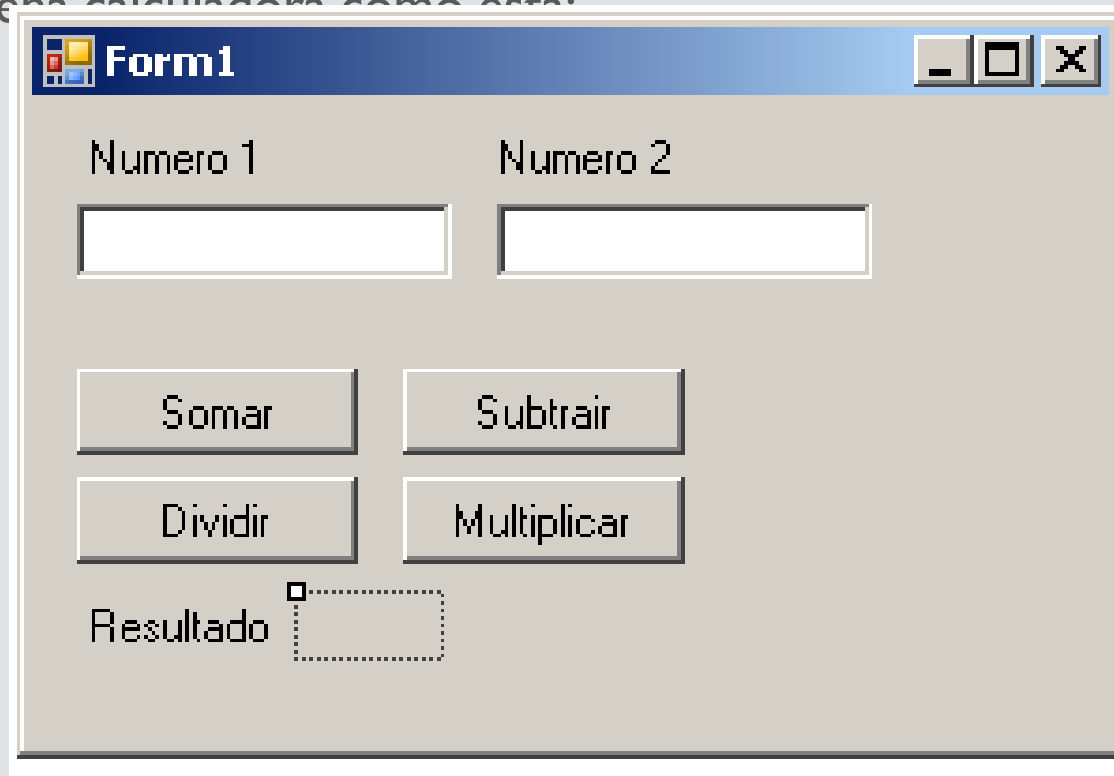
Converte os textos digitados nas caixas de texto para números

Calcula a média

Converte a média que é do tipo float para o tipo string, compatível com o texto.

FAÇA VOCÊ MESMO!

- Faça uma pequena calculadora como esta:



The image shows a screenshot of a Windows-style application window titled "Form1". The window has a standard title bar with minimize, maximize, and close buttons. The main content area is light gray and contains the following elements:

- Two input fields labeled "Numero 1" and "Numero 2" at the top, each with a white rectangular box below it.
- Four buttons arranged in a 2x2 grid: "Somar" (Add), "Subtrair" (Subtract), "Dividir" (Divide), and "Multiplicar" (Multiply).
- A label "Resultado" (Result) at the bottom left, followed by a dashed rectangular box for the output.

ESTRUTURAS CONDICIONAIS

- Estas estruturas modificam a seqüência natural de um programa e podem ou não permitir que certos trechos de código possam ser executados.

CONDICIONAL IF(SE)

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    double nota1;
    double nota2;
    double media;

    nota1 = 7.25;
    nota2 = 6.15;

    media = (nota1 + nota2) / 2;

    if (media >= 6) {
        MessageBox.Show("Aprovado");
    }
}
```

Testa se a condição é verdadeira

Só executa caso a condição seja verdadeira

Envia uma mensagem para o usuário

CONDICIONAL IF(SE)

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    double nota1;
    double nota2;
    double media;

    nota1 = 7.25;
    nota2 = 6.15;

    media = (nota1 + nota2) / 2;

    if (media >= 6)
    {
        MessageBox.Show("Aprovado");
    }
    else {
        MessageBox.Show("Reprovado");
    }
}
```

Testa se a condição é verdadeira

Só executa caso a condição seja falsa

Envia uma mensagem para o usuário

OPERADORES LÓGICOS

Operador	Função	Exemplo
&&	.e. Lógico.	$(A > B) \ \&\& \ (A < B)$
	.ou. Lógico.	$(A > B) \ \ (A < B)$
!	Negação	!z

VERIFICAR SE NOTA DIGITADA É VALIDA

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    double nota1;
    nota1 = Convert.ToDouble(txtNota1.Text);

    if (nota1 < 10 || nota1 > 10) {
        MessageBox.Show("Nota invalida!");
    }
}
```

Operador Lógico ou

Operador Lógico e

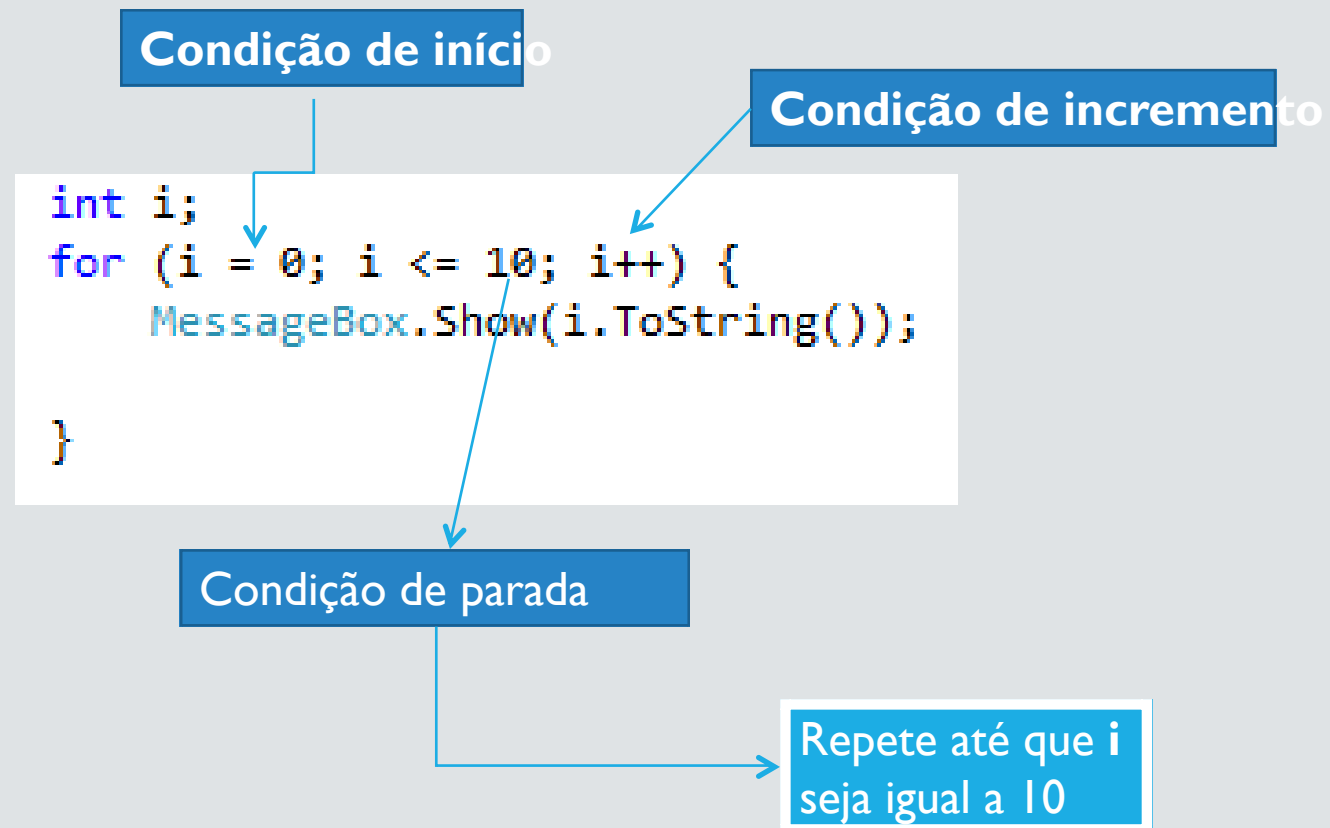
```
if (nota1 >= 0 && nota1 <= 10) {
    MessageBox.Show("Nota valida");
}
}
```

Testa se as condições são verdadeiras

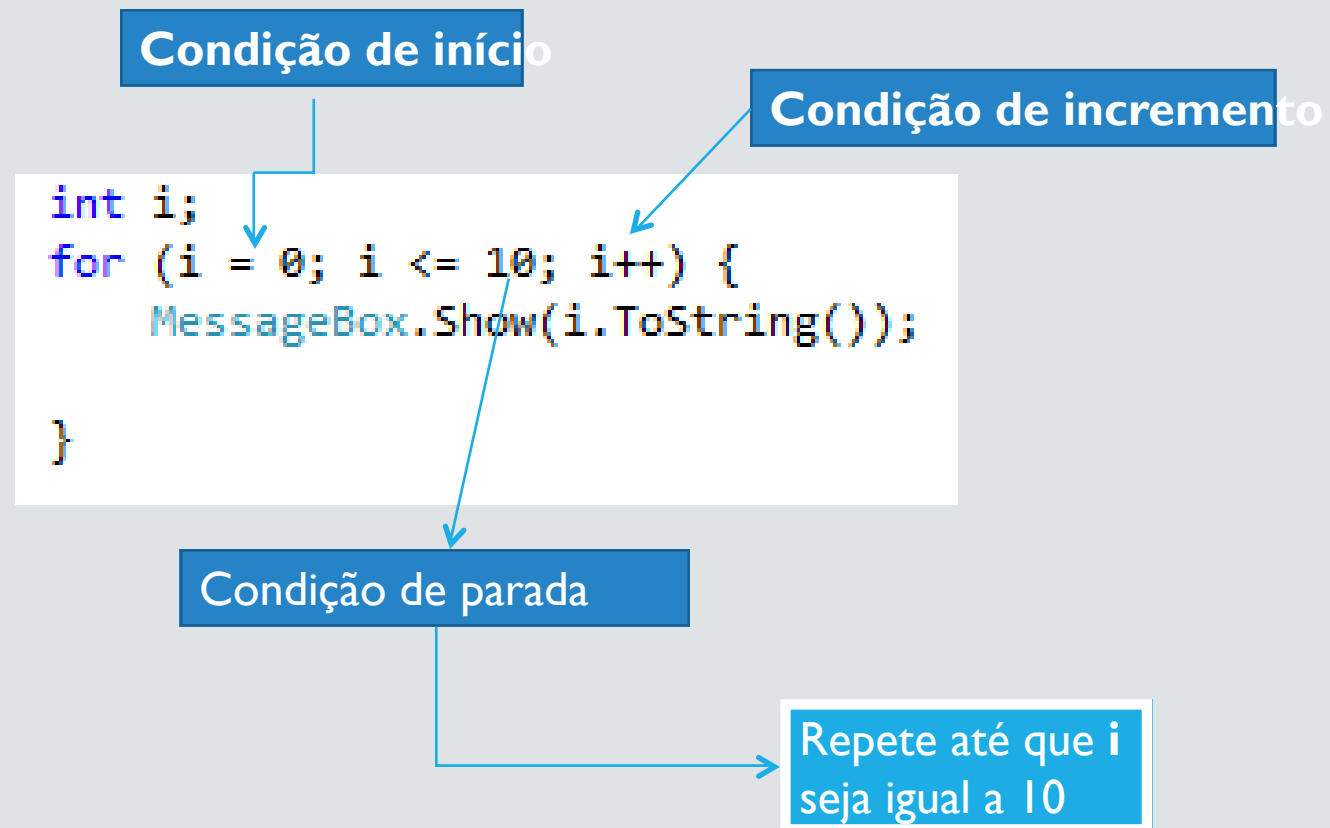
Só executa caso as condições sejam verdadeira

REPETIÇÃO

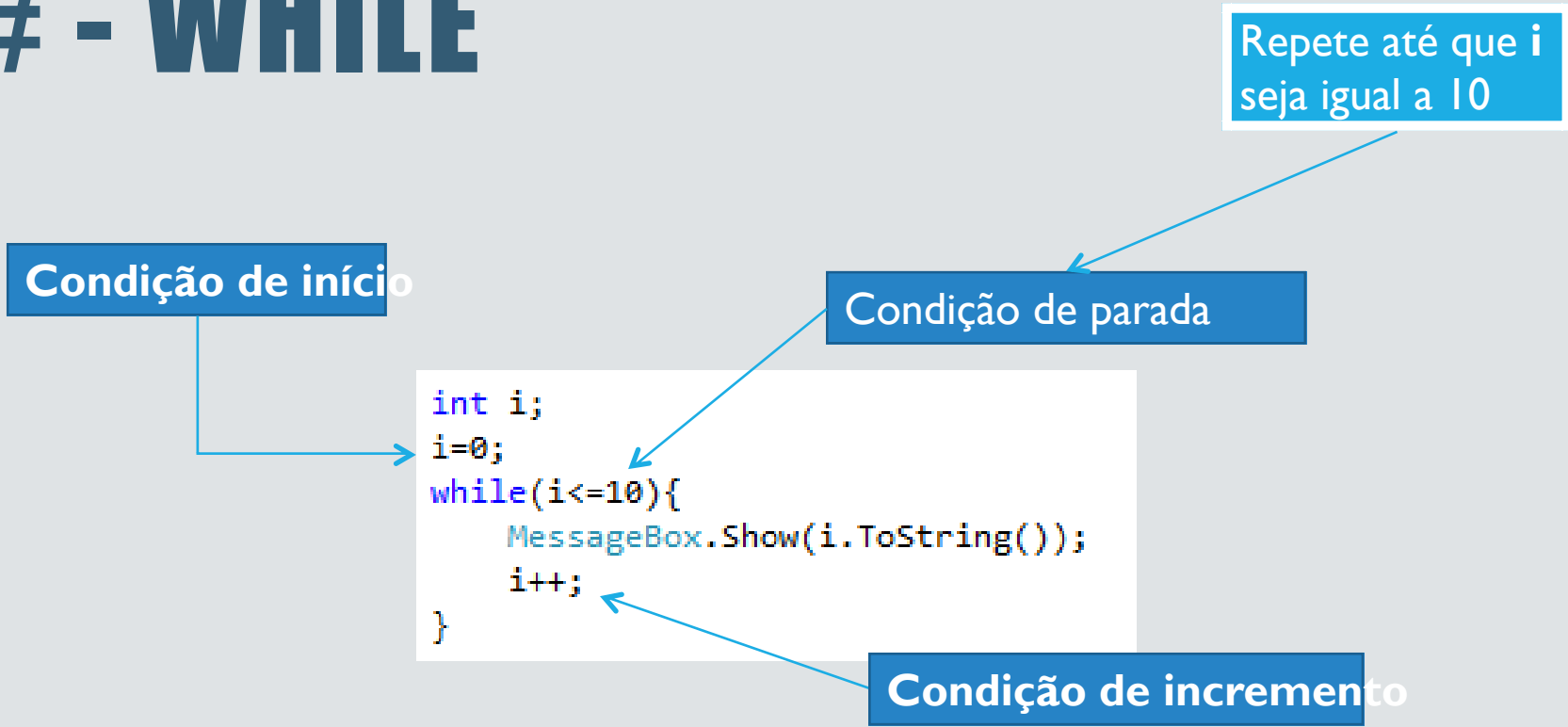
C# - FOR



C# - FOR



C# - WHILE



C# - DO...WHILE

- Do...while executa pelo menos uma vez, pois o teste é feito no final.

Condição de início

```
int i;  
i=0;  
do  
{  
    MessageBox.Show(i.ToString());  
    i++;  
} while (i <= 10);
```

Condição de incremento

Condição de parada

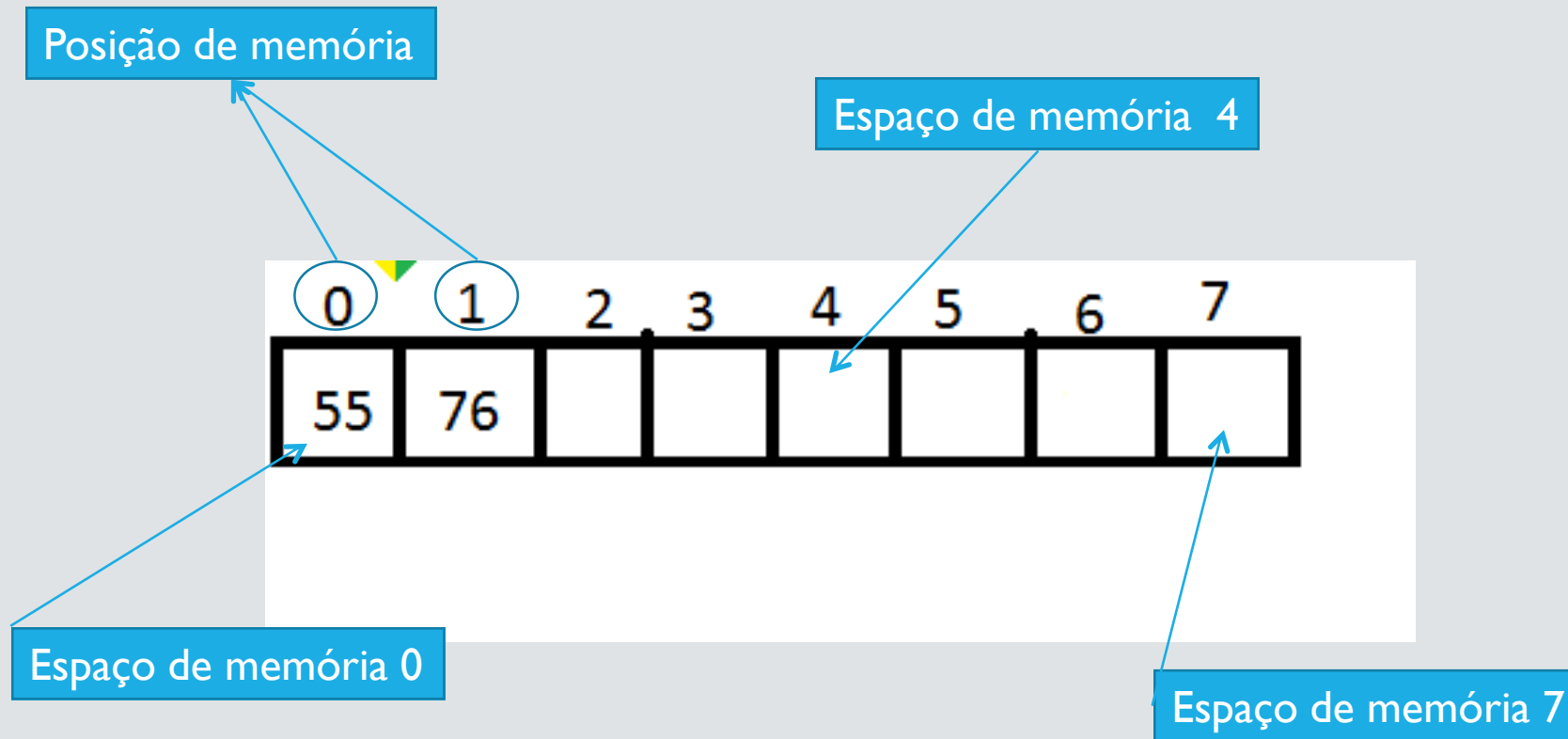


MATRIZES

MATRIZ UNIDIMENSIONAL

- So **Estruturas de Dados** capazes de armazenar uma quantidade *pré-definida* de valores **do mesmo tipo**.
- Permitem que sejam armazenados diversos valores em uma mesma “variável”.

MATRIZ UNIDIMENSIONAL



MATRIZ UNIDIMENSIONAL

Posição de memória

Memória			
posição 0	posição 1	posição 2	posição 3
	540		8456
Espaço de memória	Espaço de memória	Espaço de memória	Espaço de memória

Espaço de memória 0

DECLARANDO MATRIZ UNIDIMENSIONAL

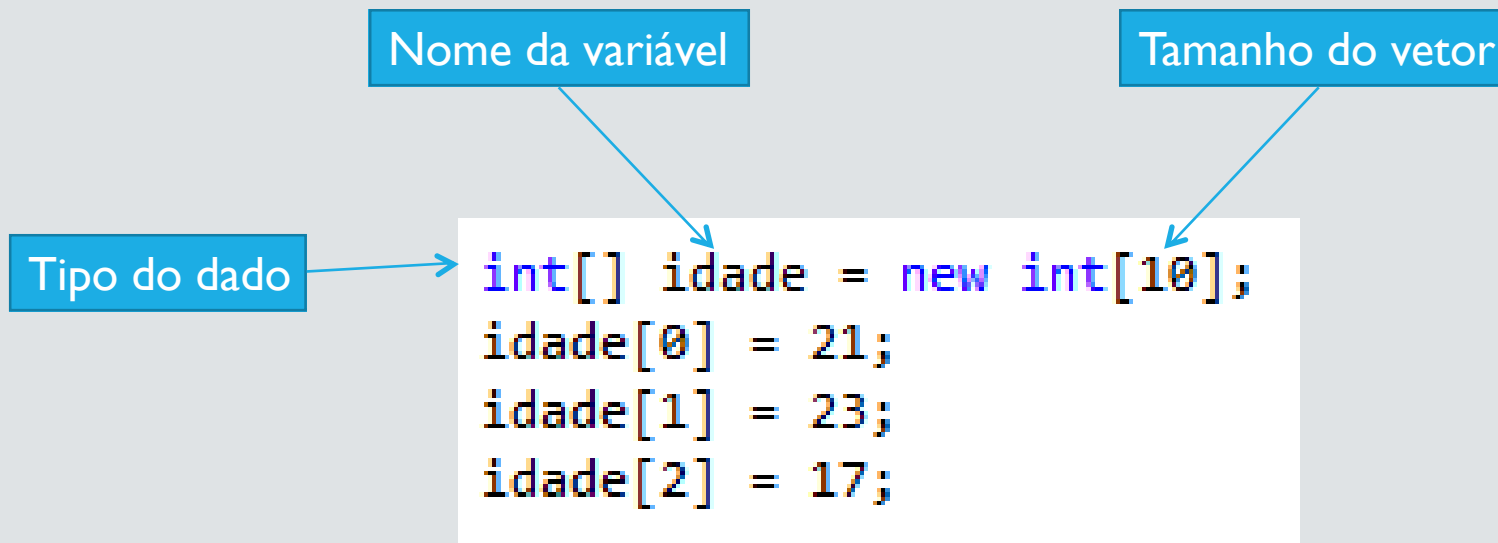
Valores que serão atribuídos

Tipo do dado

Nome da variável

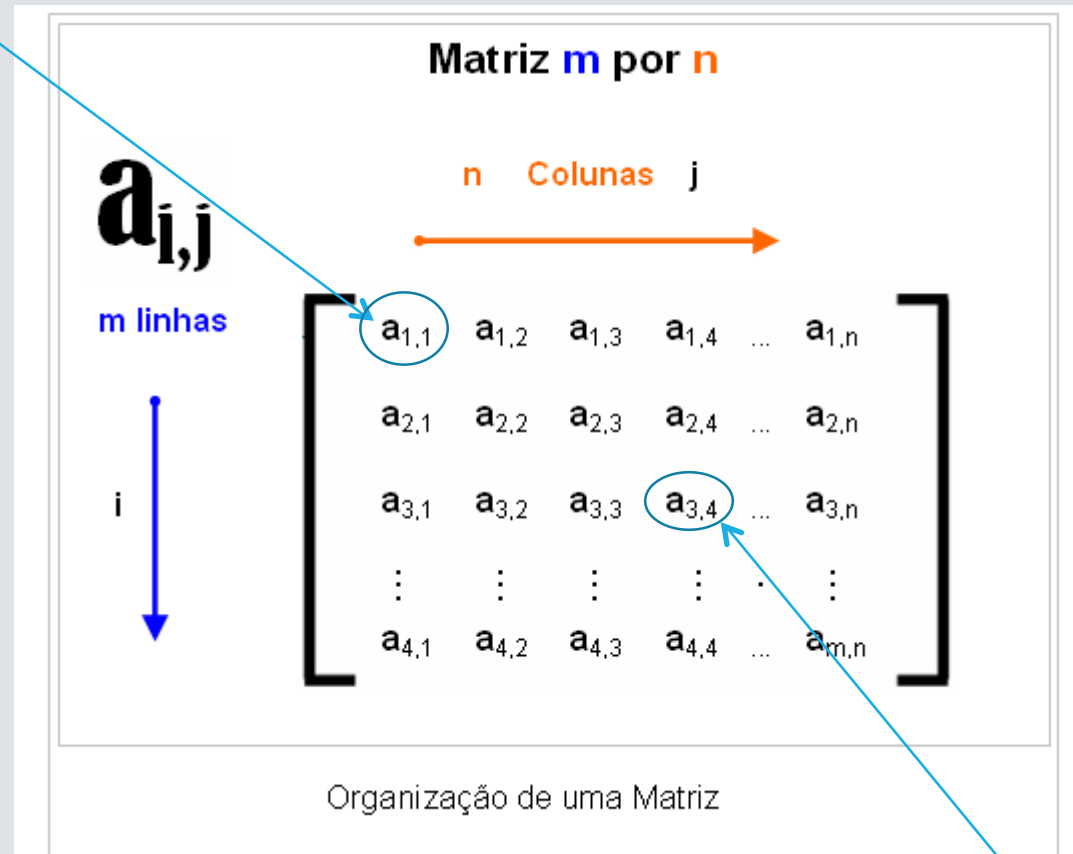
```
string[] sSemana = { "domingo", "segunda", "terça", "quarta", "quinta", "sexta", "sábado" };  
int[] idade = { 17, 16, 32, 22, 21, 15 };
```

DECLARANDO MATRIZ UNIDIMENSIONAL



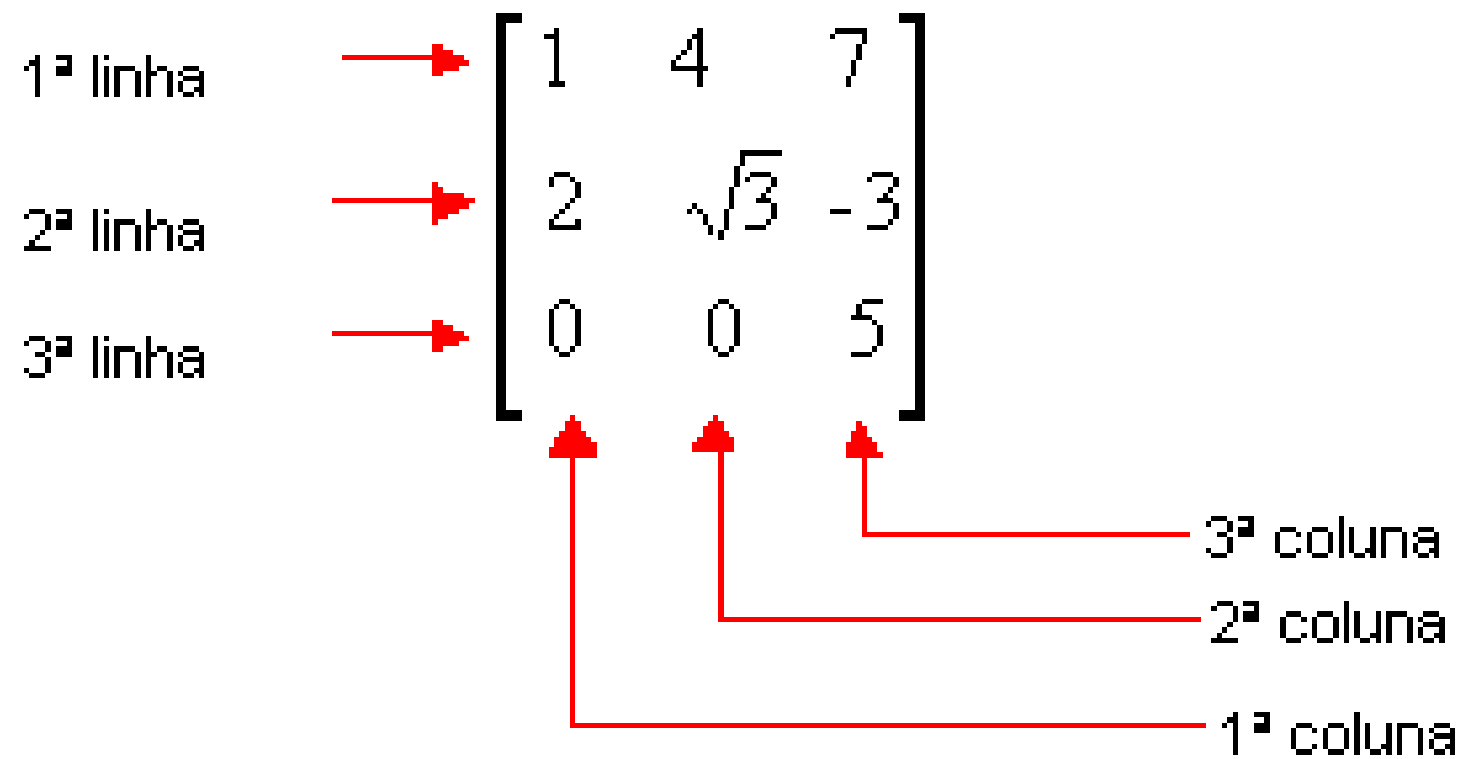
MATRIZ

Posição de memória (1,1) –
Linha 1 e Coluna 1



Posição de memória (3,4) –
Linha 3 e Coluna 4

VETORES



MATRIZES

- As matrizes são, comumente referenciadas através de suas dimensões (quantidade de linhas e colunas).
- A notação comum é: $M \times N$, onde:
 - M é a dimensão vertical (quantidade de linhas)
 - N é dimensão horizontal (quantidade de colunas)

MATRIZ

3x3

3x2

2x3

Linhas x Colunas

DECLARANDO MATRIZES

