Layout

Salve a solução com um nome apropriado dentro da pasta C# criada anteriormente no Desktop.

1. Formulário – Propriedades
    a. Em Gerenciador de Projetos, alterar o nome de Form1.cs para frmCalculadora.cs;
    b. Nas propriedades, alterar os seguintes itens:
        i. FormBorderStyle: FixedSingle
        ii. MaximizeBox e MinimizeBox : False
        iii. StartPosition: CenterScreen
        iv. Text: EtecCALC

Insira uma caixa TextBox, posicionando o objeto como mostrado em ***Figura 1 – Layout da Calculadora***.

2. TextBox – Propriedades
    a. (Name): txtVisor
    b. BackColor: DarkOrange
    c. Font(Name): Impact
    d. Font(Size): 30
    e. ReadOnly: True
    f. Text: (Em branco)

Insira um Button, posicionando o objeto como mostrado em ***Figura 1 – Layout da Calculadora***.

3. Button1 – Propriedades
    a. (Name): btn1
    b. AutoSize: True
    c. FlatSytle: Flat
    d. FlatAppearance(BorderSize): 0
    e. Image: Clique no botão com [...] no final da caixa e escolha a imagem correspondente ao número colocado no (Name)[1]. Observe a seleção em ***Figura 2 - Inserindo imagem no botão***.
    f. Text: (Em branco)

[1] As imagens podem ser inseridas e gerenciadas através do painel acessível em: Gerenciador de Soluções/(Nome da Solução)/Properties/Resources.resx.

Repita os passos do item “3”, inserindo a quantidade de botões suficientes para montar o layout da calculadora como visto em ***Figura 1 – Layout da Calculadora***. Utilize as alterações realizadas no *Button1* para cada um dos botões inseridos no formulário. *Dica: Copie e cole o Buton1, depois de alterar as propriedades do mesmo. Provavelmente, a única alteração a ser feita será o nome do objeto e a imagem.*

Figura 1 - Layout da Calculadora

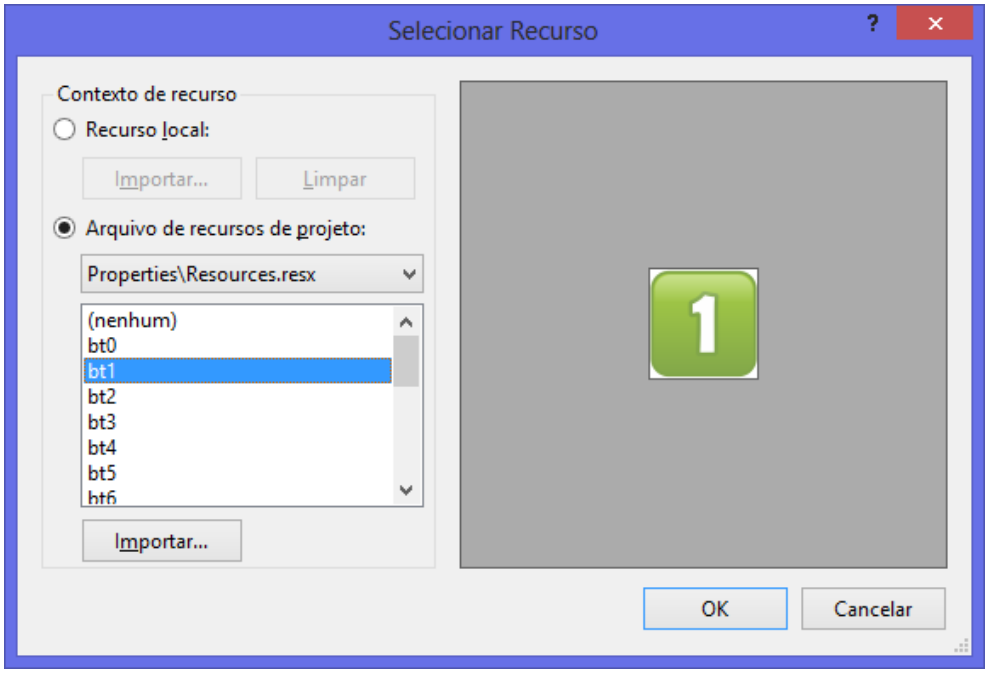

Figura 2 - Inserindo Imagem no botão

Programação

1. Criaremos a calculadora limitando o visor a nove caracteres. Para tanto, declare uma variável global do tipo inteiro, chamada contVisor. Observe declaração de variáveis globais visto em ***Figura 3 – Declaração de variáveis globais***.

```
namespace CalcSimples
{
    public partial class frmCalculadora : Form
    {

        //Declare suas variáveis globais aqui!

        public frmCalculadora()
        {
            InitializeComponent();
        }
```

Figura 3 - Declaração de variáveis globais

2. Clique 2X no formulário para criarmos o evento ***frmCalculadora_Load***. Ou, clique no formulário e depois no botão de Eventos (Raio). Em seguida clique duas vezes na propriedade Load. Atribua o valor 0 (zero) à variável.

```
private void frmCalculadora_Load(object sender, EventArgs e)
    {
        contVisor = 0; //Código a ser inserido!
    }
```

3. Iremos agora associar os valores dos botões ao visor. Clique 2X no ***btn1***, para criarmos o evento ***btn1_Click***. Ou, clique no botão e depois no botão de Eventos (Raio). Em seguida clique duas vezes na propriedade Click. Insira o código abaixo.

```
    private void btn1_Click(object sender, EventArgs e)
        {
            if (contVisor != 9) //Verificando se o limite do visor foi
ultrapassado!
            {
                txtVisor.Text += '1'; //Concatenação do valor 1 ao texto do
visor
                contVisor++; //Incremento da variável de controle de texto do
visor
            }
        }
```

Repita os passos do item “3” para todos os botões numéricos, alterando os valores a serem concatenados de acordo com o número do botão.

Este exemplo de calculadora realiza cálculos simples com as 4 operações. Todos os cálculos partem do princípio que a operação sempre realiza um cálculo de cada vez, independente da expressão que tenha que ser resolvida. Exemplo: 8 + 2 – 5. Primeiro, realizamos a soma dos valores 8 e 2. Do seu resultado, subtrai-se 5. Assim, podemos concluir que, pelo menos por enquanto, devemos criar 2 variáveis. Quando o usuário clicar no número 8 e logo em seguida no botão de soma, o valor que estava no visor irá para uma variável e o tipo de operação também será armazenado. O visor será limpo esperando que o usuário digite o segundo valor. O calculo será quando o usuário clicar no botão igual.

4. Declare as variáveis globais do tipo ***double*** valor1 e memAux. Veja como em ***Figura 3 – Declaração de variáveis globais***. Embora nossa calculadora realize apenas digitação de valores inteiros, o resultado de uma divisão pode gerar valores decimais, por isso criamos as variáveis do tipo decimal (double).

5. Assim como no passo “2”, associe o valor 0 (zero) à variável valor1.

6. Crie uma variável global do tipo string, chamada ***operador***. Esta variável irá armazenar o tipo de operação que será realizada. Todos os cálculos serão executados de fato quando clicarmos no botão “igual”.

7. Clique 2X no botão ***btnSoma***. Ou, clique no botão e depois no botão de Eventos (Raio). Em seguida clique duas vezes na propriedade Click. Insira o código abaixo.

```
        private void btnSoma_Click(object sender, EventArgs e)
         {
             if (txtVisor.Text != "") //Compara se a caixa de texto é diferente de
vazio
             {
                         valor1 = Convert.ToDouble(txtVisor.Text); // Armazena o
        valor do visor numa memória auxiliar

                 operador = "+"; //Atribui o tipo de operação
                 txtVisor.Clear(); //Limpa a caixa de texto
                 contVisor = 0; //Zera o contador do visor

                 btnSoma.Enabled = false;
                 btnSub.Enabled = false;
                 btnMult.Enabled = false;
                 btnDiv.Enabled = false;
             }
           }
```

Iremos desabilitar os botões para Somar, Subtrair, Multiplicar e Dividir para forçar o usuário a clicar no botão **“Igual”.**

8. Repita os passos do item “7” para todos os botões de subtração, multiplicação e divisão, alterando os sinais para cada tipo de operação.

9. Clique 2X no botão ***btnIgual***. Ou, clique no botão e depois no botão de Eventos (Raio). Em seguida clique duas vezes na propriedade Click. Insira o código abaixo.

```
private void btnIgual_Click(object sender, EventArgs e)
        {
            if (txtVisor.Text != "") //Compara se a caixa de texto é
diferente de vazio
            {
                memAux = Convert.ToDouble(txtVisor.Text);

                switch (operador)
                {
                    case "+":
                        valor1 += memAux;
                        txtVisor.Text = Convert.ToString(valor1);
                        valor1 = 0;
                        break;
                    case "-":
                        valor1 -= memAux;
                        txtVisor.Text = Convert.ToString(valor1);
                        valor1 = 0;
                        break;
                    case "*":
                        valor1 *= memAux;
                        txtVisor.Text = Convert.ToString(valor1);
                        valor1 = 0;
                        break;
                    case "/":
                        if (valor1 != 0 && memAux != 0)
                        {
                            valor1 = valor1 / memAux;
                            txtVisor.Text = Convert.ToString(valor1);
                            valor1 = 0;
                        }
                        else
                        {
                            MessageBox.Show("Impossível realizar divisão por
0!", "Erro de execução");
                            valor1 = 0;
                            memAux = 0;
                            txtVisor.Clear();
                        }
                        break;

                    default:
                        break;
                }

                contVisor = 0;

                btnSoma.Enabled = true;
                btnSub.Enabled = true;
                btnMult.Enabled = true;
                btnDiv.Enabled = true;
            }
                                        }
```

10. Finalmente, para encerrarmos, incluiremos o código que faz a limpeza de todos os cálculos, zerando a calculadora como se esta estivesse sendo inicializada neste momento.

    Execução

```
private void btnLimpar_Click(object sender, EventArgs e)
    {
        txtVisor.Clear();
        valor1 = memAux = contVisor = 0;

        btnSoma.Enabled = true;
        btnSub.Enabled = true;
        btnMult.Enabled = true;
        btnDiv.Enabled = true;
    }
```