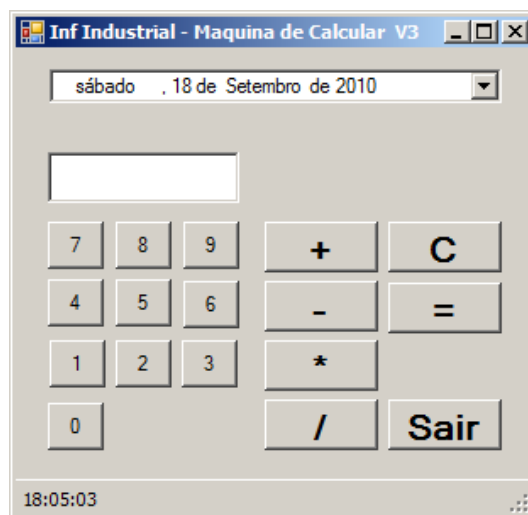




### 1. Introdução

Desenvolvimento de uma máquina de calcular elementar. Visualização da data e hora do computador, escrita e leitura de dados em ficheiros de texto para familiarização e aquisição de conhecimentos de algumas instruções e objetos do Visual Basic.



### 2. Objetivos

O trabalho prático pretende familiarizar o aluno com o ambiente de programação do visual studio.

Familiarização com os objetos de programação:

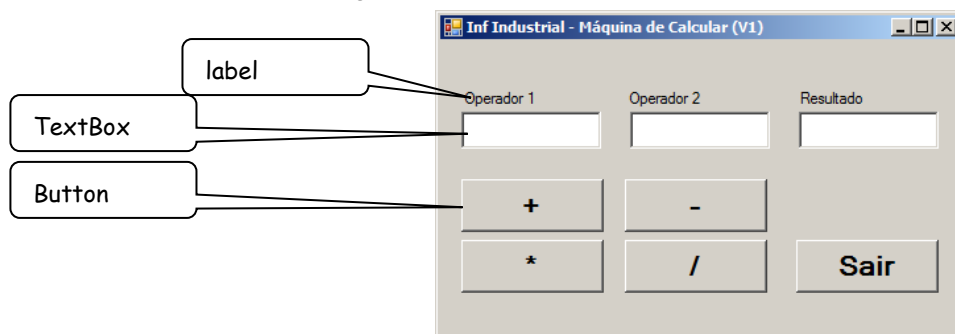
- Textbox, Button, Label, Checkbox, Picture, Panel, RadioButton, Form, Modules, etc.

Familiarização com as instruções de programação:

- End - termina a execução do programa,
- Operadores matemáticos: +, -, \*, /
- Cint – Converte texto num número inteiro,
- Escrita e leitura em ficheiros de texto,
- While ... End While : execução em ciclo,
- If Then Else – Instrução de condição,
- Escrita e Leitura em ficheiro de texto;

O trabalho prático é composto por diferentes versões da máquina de calcular, inicialmente versões mais simples, até à realização de uma máquina de calcular como a indicada na figura anterior, com funções matemáticas especiais e gravação em ficheiro.

### Exercício 1: Máquina de calcular com 4 operações matemáticas fundamentais



Máquina de calcular com as 4 operações matemáticas principais: adição, subtração, divisão e multiplicação.

Inserir os objetos indicados na figura:

- Label, Textbox e Button, como indicado na figura.

Para cada um dos objetos, pode-se alterar as propriedades: Nome, texto, tipo de letra, etc. e associar a cada objeto um conjunto de instruções a executar, quando o respetivo objeto é selecionado.

Para cada **button** associar uma das operações matemáticas e, escrever o código indicado na listagem, clicando 2 vezes em cada um dos objetos. A cada objeto está associado uma rotina:

**Private Sub**, onde o cabeçalho é inserido automaticamente:

```
Private Sub Btn_Terminar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_Terminar.Click
    'Terminar a aplicação
End
End Sub
```

### Instruções:

- End - termina a execução do programa;
- Cint - converte um texto numa variável numérica inteira;
- CDbI - converte uma expressão em double;

### Notas:

- É importante a organização dos programas;
- Os programas devem estar devidamente identificados:
  - Data, Identificação do aluno, Número da aula prática;
- Os programas devem ser comentados;
- Os objetos e as rotinas devem ser identificados utilizando critérios bem definidos. Os objetos podem ser identificados completando o seu tipo com um prefixo adequado, por exemplo:
  - TextBox - TextBoxLeitura
  - Button - ButtonSair
  - Label - LabelFicheiro

### Listagem do programa:

```
Public Class Form1
```

```
Private Sub Btn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Bt.Click
    End ' Terminar programa
End Sub
```

```
Private Sub Btn_Soma_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Bt_Soma.Click
    ' Adição
    Txt_Res.Text = Cint(Txt_Op1.Text) + Cint(Txt_Op2.Text)
End Sub
```

```
Private Sub Btn_Sub_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Bt_Sub.Click
    ' Subtração
    Txt_Res.Text = Cint(Txt_Op1.Text) - Cint(Txt_Op2.Text)
End Sub
```

```
Private Sub Btn_Mul_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Bt_Mul.Click
    ' Multiplicação
    Txt_Res.Text = Cint(Txt_Op1.Text) * Cint(Txt_Op2.Text)
End Sub
```

```
Private Sub Btn_Div_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Bt_Div.Click
    ' Divisão
    Txt_Res.Text = Cint(Txt_Op1.Text) / Cint(Txt_Op2.Text)
End Sub
```

```
End Class
```

## Exercício 2: Máquina de calcular com registo das operações realizadas em ficheiro texto

Realizar a leitura e escrita em ficheiro de texto, dos resultados das operações matemáticas realizadas na máquina de calcular. Na janela lateral de leitura poderá visualizar o conteúdo desse ficheiro.

**Escrita em ficheiro** de texto, é necessário definir o nome e local de armazenamento do ficheiro de texto e **abrir o ficheiro** através da instrução:

```
'Declaração e abertura do ficheiro de escrita em disco  
FileOpen(1, "E:\resultados.Txt", OpenMode.Output)
```

Esta instrução deve ser executada sempre que se quer escrever no ficheiro.

**Escrita de dados no ficheiro:**

```
Print(1, " + " + Txt_Op1.Text)
```

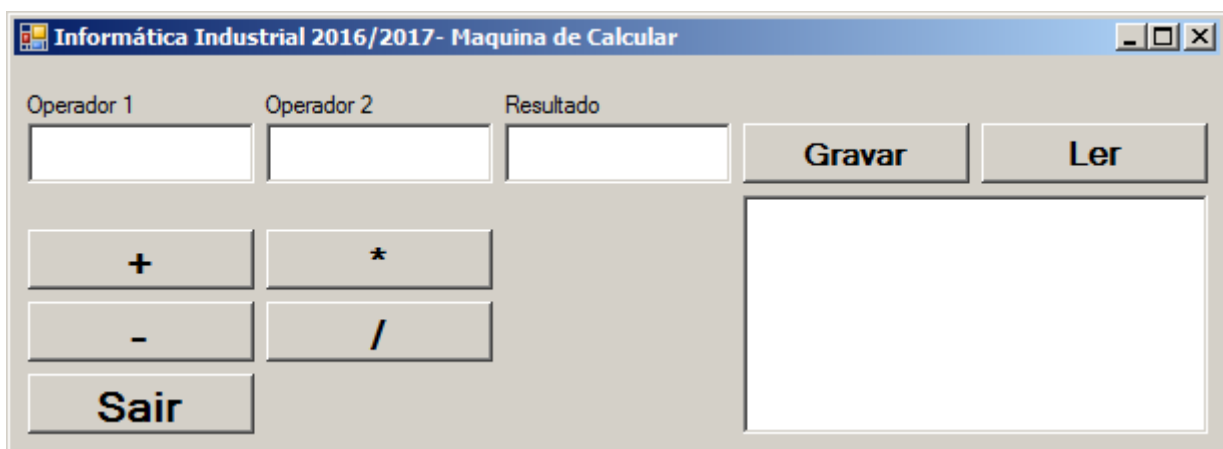
Para **fechar e “gravar” em disco**, executar a instrução:

```
' Fecha ficheiro de escrita em disco  
FileClose(1)
```

**Leitura do ficheiro:**

```
' Cada vez que esta função é executada devolve uma linha de texto existente no ficheiro  
Txt_Ficheiro.Text = Txt_Ficheiro.Text + LineInput(1) + vbCrLf
```

Ao exemplo anterior acrescentar dois botões com os nomes “Btn\_Gravar” e “Btn\_Ler”, acrescentar uma janela de texto com o nome “Txt\_Ficheiro”.



```
Private Sub Btn_Gravar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Bt_Gravar.Click  
    FileOpen(1, "E:\resultados.txt", OpenMode.Append)  
    PrintLine(1, "Resultado= " + Txt_Res.Text)  
    FileClose()  
End Sub
```

```
Private Sub Btn_Ler_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Bt_Ler.Click  
    FileOpen(1, "E:\resultados.txt", OpenMode.Input)  
  
    While (EOF(1) = False)  
        Txt_Ficheiro.Text = Txt_Ficheiro.Text + LineInput(1) + vbCrLf  
    End While  
  
    FileClose()  
End Sub
```

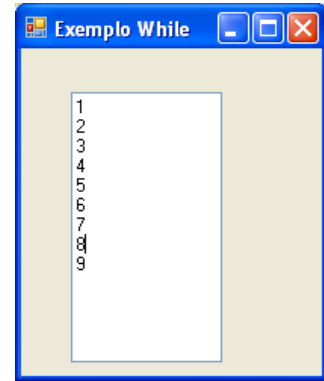
A função **EOF(1) (End Of File)** indica o fim do ficheiro, ou seja, já foram lidas todas as linhas do ficheiro (1), devolvendo o valor "True"

O ciclo **While ... End While** repete automaticamente as instruções existentes no seu interior, ou seja enquanto a função **EOF(1)** devolver o valor falso o ciclo é repetido, realizando a leitura da próxima linha do ficheiro.

Outro exemplo para o **ciclo While**, pode ser realizado para incrementar uma variável inteira, com o nome "i". Durante a execução do programa, a variável "i" assume inicialmente o valor 1, sendo incrementada sequencialmente até 10. O valor 10 já não é visualizado na caixa de texto "Txt\_Res".

Dim i As Integer

```
i = 1
While (i < 10)
    Txt_Res.Text = Txt_Res.Text + CStr(i) + vbCrLf
    i = i + 1
End While
```

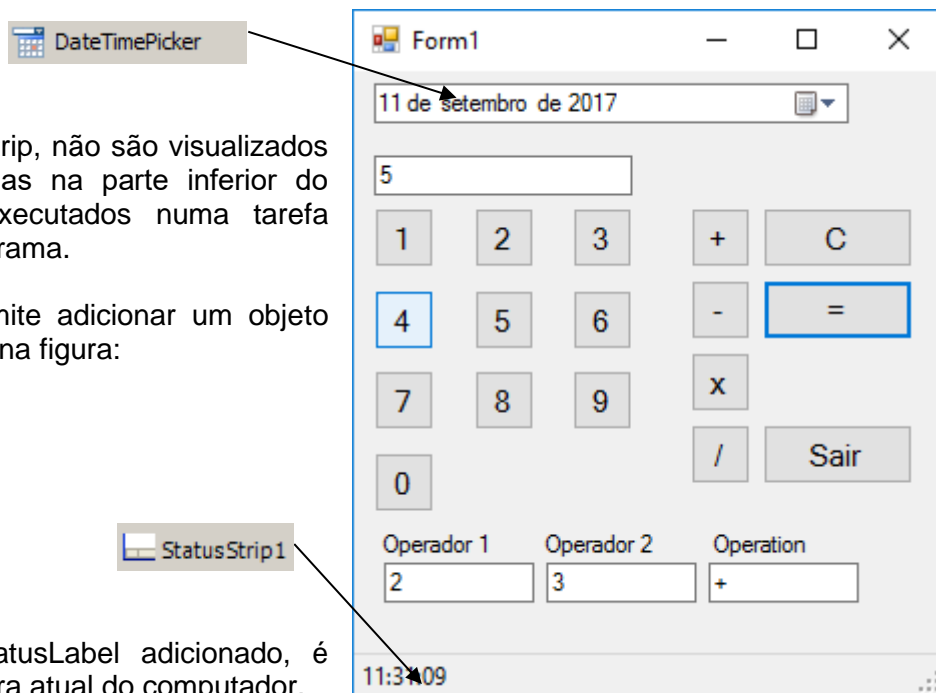
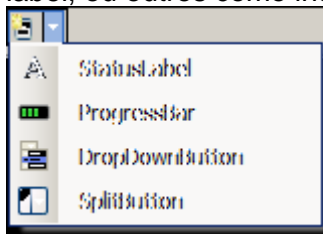




Pretende-se desenvolver uma máquina de calcular com 4 operações e indicação da data e hora do computador.

Os objetos Timer e StatusStrip, não são visualizados na “form” do programa, mas na parte inferior do form1.vb(design) e são executados numa tarefa paralela à execução do programa.

O objeto StatusStrip1, permite adicionar um objeto label, ou outros como indica na figura:



O objeto StatusStrip - StatusLabel adicionado, é utilizado para visualizar a hora atual do computador.

O Objeto timer (com o nome TimerDate) é executado ciclicamente, sendo necessário ativá-lo, nas propriedades gerais do objeto ou através de instruções de código, como é feito mais adiante.

As instruções escritas no procedimento associado ao Form, são executadas no início do programa e utilizadas para atribuir valores iniciais às variáveis do programa (predefinições).

As instruções associadas aos temporizadores são executadas ciclicamente, quando ativo.

O botão “C”, limpa o conteúdo do campo de entrada de dados.

```

1 Public Class Form1
2
3     'Global program variables
4     Dim operator1 As String
5     Dim operator2 As String
6     Dim operation As String
7
8     '-----
9     'Initialization routine2
10    '-----
11    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles ↗
12        MyBase.Load
13        'Initialize the Timer
14        TimerDate.Interval = 500
15        TimerDate.Enabled = True
16    End Sub
17
18    '-----
19    'Mathematical operation Button callbacks
20    '-----
21    Private Sub ButtonAdd_Click(sender As Object, e As EventArgs) Handles ↗
22        ButtonAdd.Click
23        SetOperationAndOperator1("+", TextBoxOpResult.Text)
24    End Sub
25
26    Private Sub ButtonSubtract_Click(sender As Object, e As EventArgs) Handles ↗
27        ButtonSubtract.Click
28        SetOperationAndOperator1("-", TextBoxOpResult.Text)
29    End Sub
30
31    Private Sub ButtonMultiply_Click(sender As Object, e As EventArgs) Handles ↗
32        ButtonMultiply.Click
33        SetOperationAndOperator1("*", TextBoxOpResult.Text)
34    End Sub
35
36    Private Sub ButtonDivide_Click(sender As Object, e As EventArgs) Handles ↗
37        ButtonDivide.Click
38        SetOperationAndOperator1("/", TextBoxOpResult.Text)
39    End Sub
40
41    'Called when +, -, * or / are pressed
42    Private Sub SetOperationAndOperator1(operation_arg As String, ↗
43        operator1_arg As String)
44        'Set operation
45        operation = operation_arg
46
47        'Set operator1
48        operator1 = operator1_arg
49
50        'Delete from TextBoxOpResult
51        TextBoxOpResult.Text = ""
52
53        'for debug, show operator1 and operation
54        TextBoxOp1.Text = operator1
55        TextBoxOperation.Text = operation
56    End Sub
57
58 End Class

```

```

51     End Sub
52
53     Private Sub ButtonClearResult_Click(sender As Object, e As EventArgs) Handles ButtonClearResult.Click
54         TextBoxOpResult.Text = ""
55         TextBoxOp1.Text = ""
56     End Sub
57
58     Private Sub ButtonEqual_Click(sender As Object, e As EventArgs) Handles ButtonEqual.Click
59         operator2 = TextBoxOpResult.Text
60
61         'debug: show operator2 in textboxOp2
62         TextBoxOp2.Text = operator2
63
64         If IsNumeric(operator1) And IsNumeric(operator2) Then
65             If operation = "+" Then
66                 TextBoxOpResult.Text = CDb1(operator1) + CDb1(operator2)
67             ElseIf operation = "-" Then
68                 TextBoxOpResult.Text = CDb1(operator1) - CDb1(operator2)
69             ElseIf operation = "*" Then
70                 TextBoxOpResult.Text = CDb1(operator1) * CDb1(operator2)
71             ElseIf operation = "/" Then
72                 TextBoxOpResult.Text = CDb1(operator1) / CDb1(operator2)
73             End If
74
75         Else
76             TextBoxOpResult.Text = "Error"
77         End If
78     End Sub
79
80     '-----
81     'Numeric Button callbacks
82     '-----
83     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
84         TextBoxOpResult.Text = TextBoxOpResult.Text + "1"
85     End Sub
86
87     Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
88         TextBoxOpResult.Text = TextBoxOpResult.Text + "2"
89     End Sub
90
91     Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
92         TextBoxOpResult.Text = TextBoxOpResult.Text + "3"
93     End Sub
94
95     Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
96         TextBoxOpResult.Text = TextBoxOpResult.Text + "4"
97     End Sub
98
99     Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click

```

```

...box\AulasUA\II_2017-2018\Pratica\Aula2\Ex3\Ex3\Form1.vb 3
100     TextBoxOpResult.Text = TextBoxOpResult.Text + "5"
101     End Sub
102
103     Private Sub Button6_Click(sender As Object, e As EventArgs) Handles  ➤
        Button6.Click
104         TextBoxOpResult.Text = TextBoxOpResult.Text + "6"
105     End Sub
106
107     Private Sub Button7_Click(sender As Object, e As EventArgs) Handles  ➤
        Button7.Click
108         TextBoxOpResult.Text = TextBoxOpResult.Text + "7"
109     End Sub
110
111     Private Sub Button8_Click(sender As Object, e As EventArgs) Handles  ➤
        Button8.Click
112         TextBoxOpResult.Text = TextBoxOpResult.Text + "8"
113     End Sub
114
115     Private Sub Button9_Click(sender As Object, e As EventArgs) Handles  ➤
        Button9.Click
116         TextBoxOpResult.Text = TextBoxOpResult.Text + "9"
117     End Sub
118
119     Private Sub Button0_Click(sender As Object, e As EventArgs) Handles  ➤
        Button0.Click
120         TextBoxOpResult.Text = TextBoxOpResult.Text + "0"
121     End Sub
122
123     '-----
124     'Other methods
125     '-----
126     Private Sub ButtonExit_Click(sender As Object, e As EventArgs) Handles  ➤
        ButtonExit.Click
127     End
128     End Sub
129
130     'Periodic timer callback to update time on ToolStripStatusLabel
131     Private Sub TimerDate_Tick(sender As Object, e As EventArgs) Handles  ➤
        TimerDate.Tick
132         ToolStripStatusLabel1.Text = Format(Now, "HH:mm:ss")
133     End Sub
134 End Class
135

```