



Universidade de Aveiro

*Departamento de
Engenharia Mecânica*

Informática Industrial 2022/2023

Aula prática Nº 1

Introdução ao Visual Basic

Professor Jorge Almeida

1. Introdução ao Visual Basic

Longe vão os dias em que era necessário usar um editor de texto genérico para escrever o código e para o gravar num ficheiro em disco. Depois era necessário chamar um programa (compilador) que abria esse ficheiro de texto e gerava um segundo ficheiro com o código máquina (obj), depois o programador necessitava de chamar um terceiro programa “Linker” para juntar os vários ficheiros de código máquina gerados pelo compilador, com outros ficheiros com código máquina (lib) fornecidos pela empresa do “compilador”, para que finalmente fosse possível gerar em disco um ficheiro executável que fazia o que o programador pretendia. (quando tudo corria bem...)

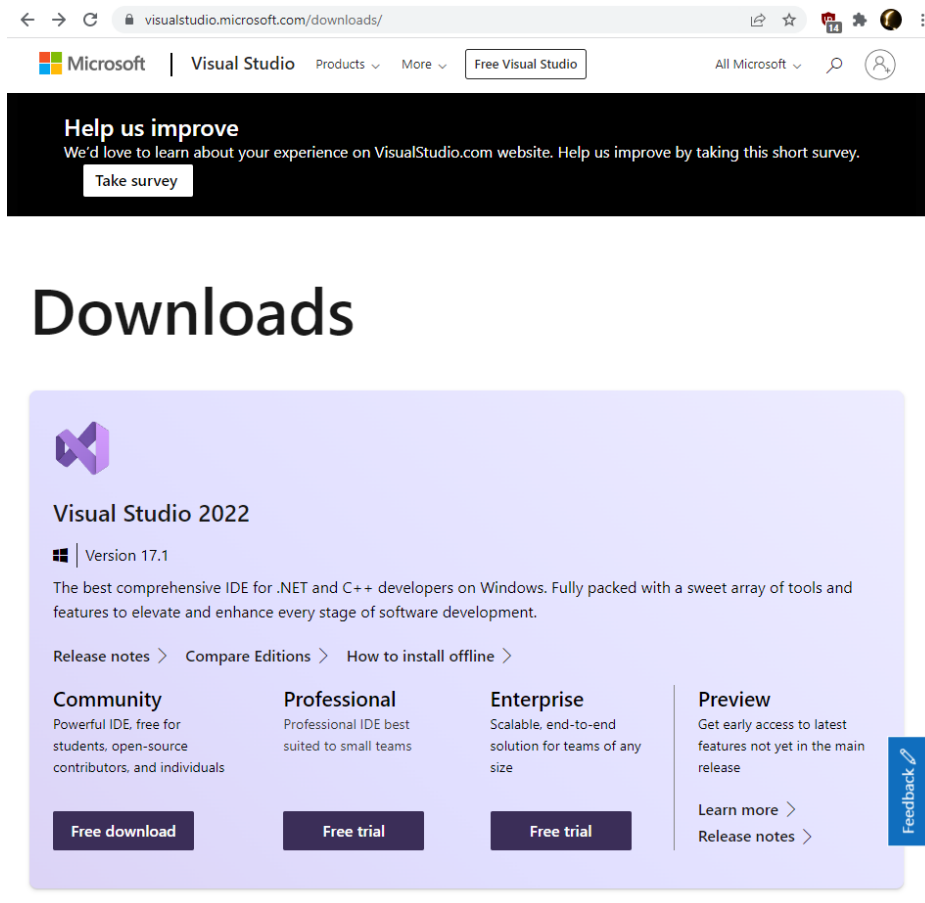
Hoje em dia existem programas comerciais, como o Visual Basic 2017 que integram todas estas funcionalidades, permitindo que o programador a partir de um ambiente integrado de desenvolvimento (IDE), possa editar vários ficheiros com o código fonte, associá-los a um mesmo projeto, compilar todos estes ficheiros, linkar e executar o programa sempre no mesmo ambiente.

Por outro lado, todos os programas desenvolvidos por um programador necessitam de ter uma interface para que o utilizador possa interagir com o programa e visualizar os resultados. Uma das maiores vantagens do Visual Basic é a enorme facilidade e rapidez com que um programador pode criar novos programas com interfaces gráficas amigáveis (secção 2.3 Construção da interface gráfica). Naturalmente que a interface gráfica só por si não chega, é necessário saber desenvolver o código necessário para processar os dados recebidos e gerar os resultados (secção 2.4 Linguagem de programação).

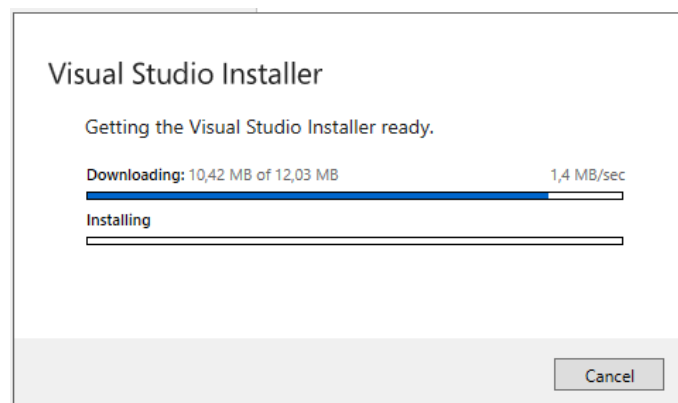
2. Instalar o Visual Basic

O **Visual Basic Community** é gratuito e pode ser transferido a partir do sítio da Microsoft. Instale a versão mais recente disponibilizada pela Microsoft.

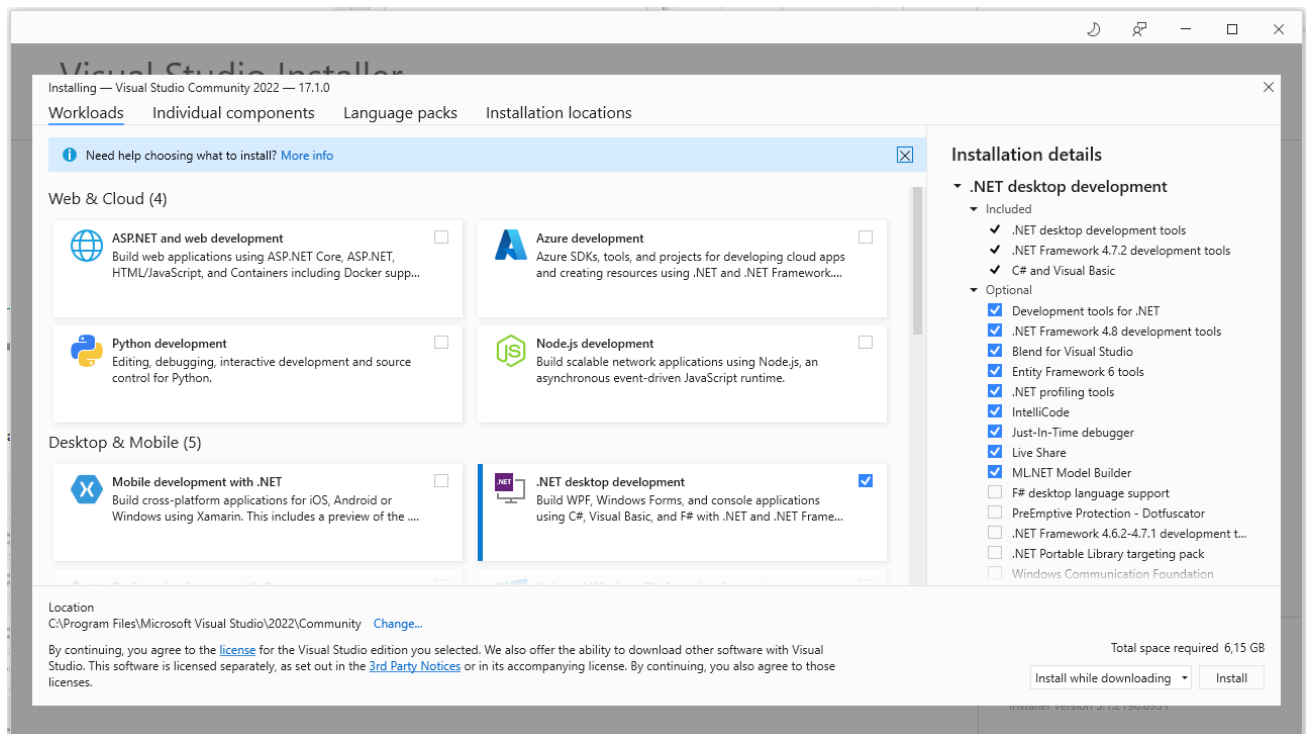
- Para instalar o **Visual Studio** utilizar o link: <https://visualstudio.microsoft.com/downloads/>
- Selecione a opção **Community Free Download**



- O *installer* do **Visual Studio** vai ser descarregado e pode usá-lo para instalar.



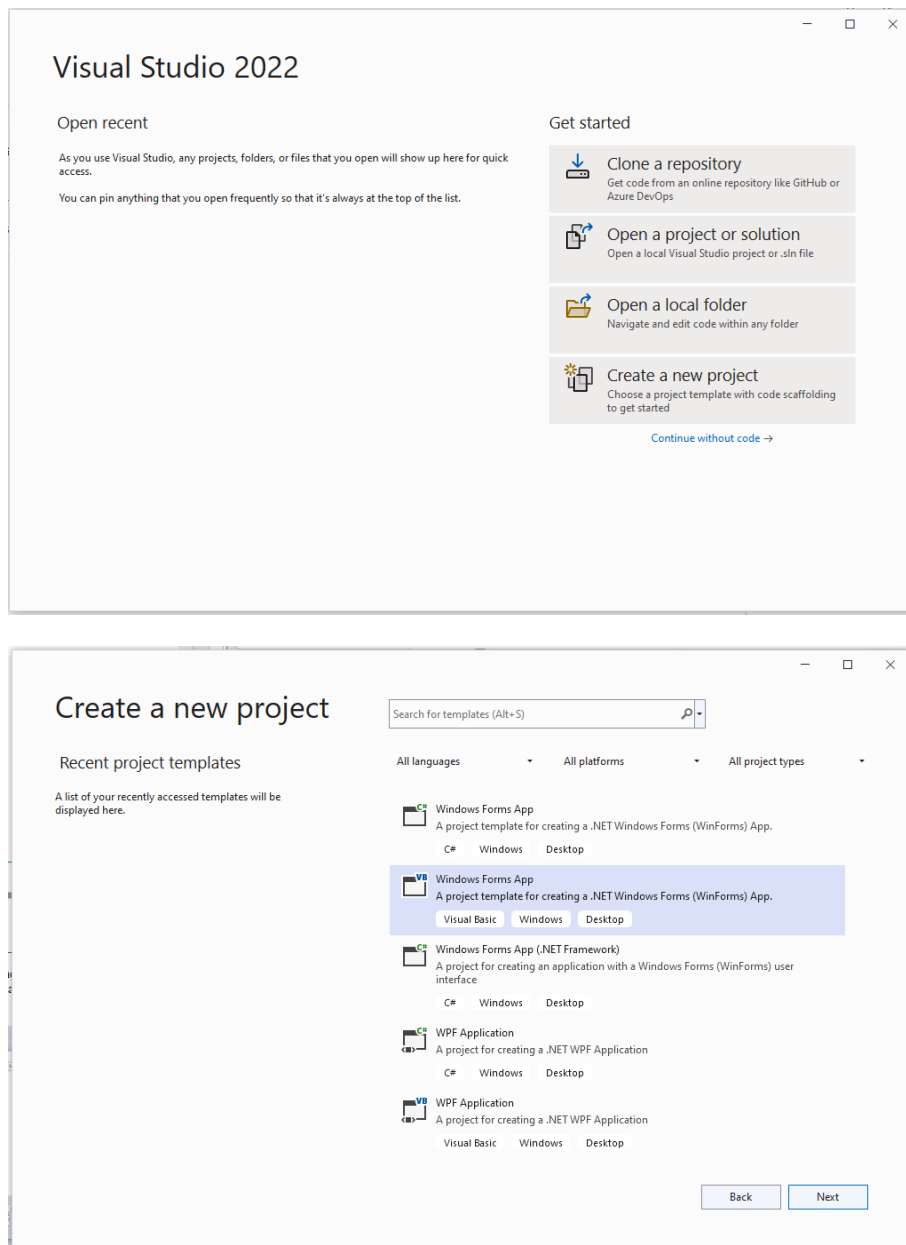
- Selecionar a opção indicada, instalando suporte para diferentes linguagens de programação: C#, Visual Basic, F# e C++;



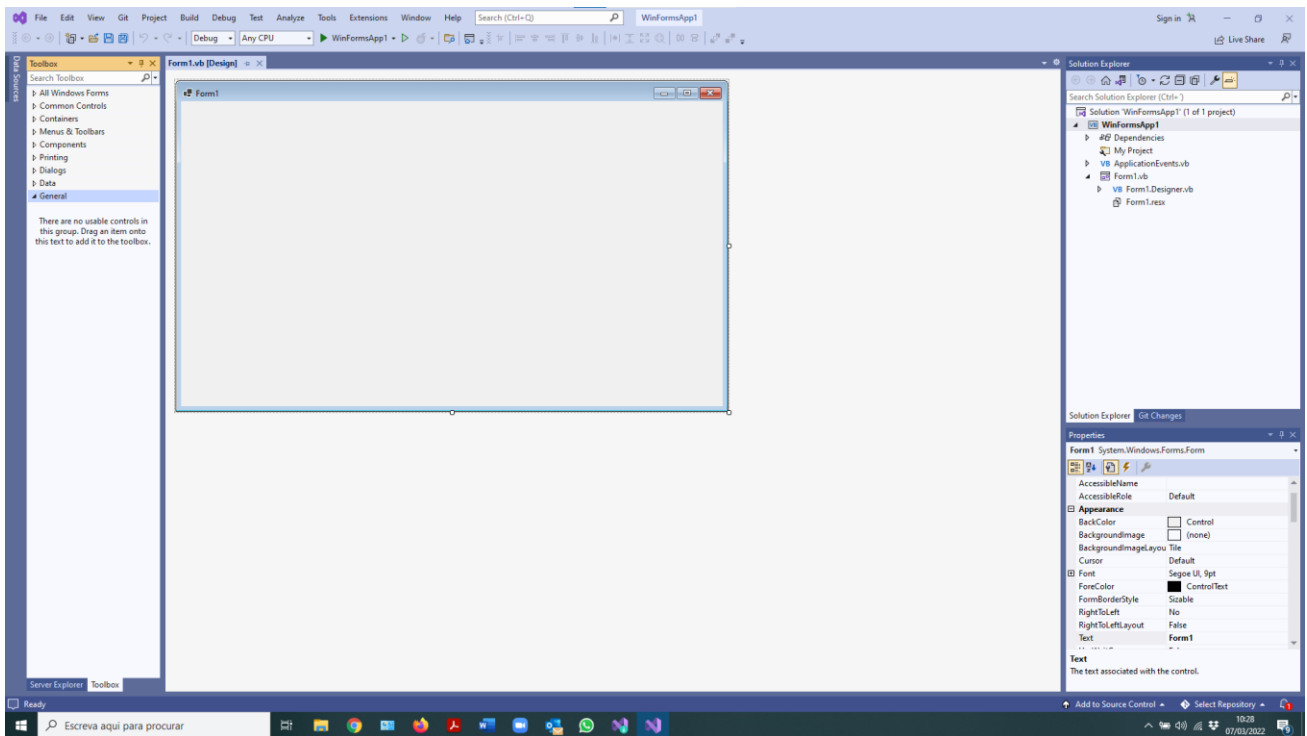
Na disciplina de Informática Industrial vamos utilizar apenas a linguagem de programação: **Visual Basic**;

3. Ambiente integrado de desenvolvimento

Para criar um novo projeto que permita desenvolver uma aplicação Windows, com interface, deve seleccionar a opção “*Create a new project*”, seleccionar a opção **Windows Forms App** com a linguagem **Visual Basic** conforme apresentado na figura.



A figura seguinte apresenta a janela inicial do ambiente integrado de desenvolvimento IDE. Ao centro aparece a janela *Form1*, à esquerda aparece a janela *Toolbox* onde é possível ir buscar os objetos gráficos para colocar na janela *Form1* (botões, caixas de texto,...), à direita aparece a janela das propriedades.



Todos os objetos gráficos (controlos) que colocarmos no *Form*, e o próprio *Form*, têm propriedades (cor, localização no ecrã, nome,...). Sempre que seleccionarmos um objeto do *Form* “clikando” sobre ele, a janela *Properties* apresenta as propriedades do objeto (controlo) seleccionado.

Se alguma destas janelas não aparecer no écran, pode seleccionar o menu *View*, a opção *OtherWindows* e depois seleccionar a(s) janela(s) que pretender ver: *Toolbox*, *Solution Explorer*, *Properties Window*, ...

Para mais informações sobre o IDE deve consultar o livro recomendado.

4. Construção da interface gráfica dos programas

Uma das maiores vantagens do **Visual Basic** reside precisamente na facilidade e rapidez com que é possível desenvolver uma interface gráfica para o nosso programa.

Na janela “*Toolbox*” temos à disposição muitos objetos “Controlos” que podemos “arrastar” para a janela “*Form*”. É esta janela que o utilizador irá ver, quando executar o nosso programa.

Nesta secção iremos estudar vários controlos, este estudo baseia-se no princípio de que “uma imagem vale mil palavras” por isso iremos analisar vários exemplos práticos. Será apresentado o código correspondente a cada exemplo, e estudado cada um dos controlos usados nesses exemplos.

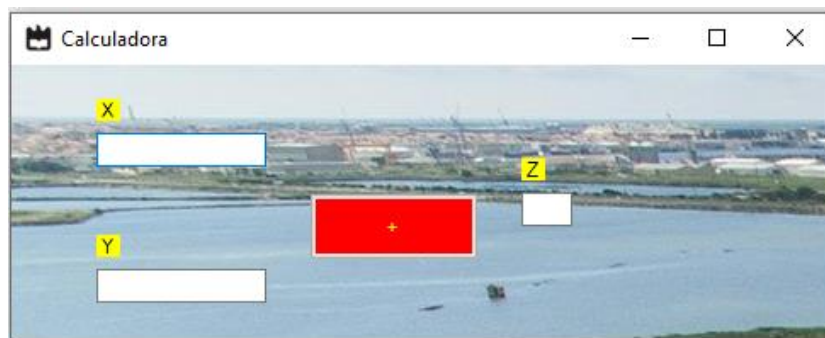
No final desta secção, depois de realizar todos os exemplos, o leitor terá implementado um editor de texto com várias janelas/*Forms* (interfaces) e terá usados todos os controlos que a seguir de enumeram:

Form, Label, TextBox, Button, RadioButton, PictureBox, Combobox, OpenFileDialog, SaveFileDialog, StreamWriter, StreamReader.

Exemplo “Calculadora_soma”

Vamos começar por um exemplo simples, intitulado Calculadora_Soma. Este programa permite calcular a soma de dois valores. O utilizador introduz dois valores, um em X, o outro em Y e quando premir o botão vermelho (+) aparece o resultado da soma em Z (figura seguinte).

Nesta interface Form1 o programador utiliza três controlos do tipo Label, três controlos do tipo *Textbox* e um controlo do tipo *Button*.



Código Visual Basic associado ao *form*

'Nesta classe é definido todo o código necessário para a execução da interface (Janela) Form1

Public Class Form1

'A subrotina Form1_Load é executada automaticamente, uma vez, quando a janela Form1 aparece no ecrã

Private Sub Form1_Load(sender **As** Object, e **As** EventArgs) **Handles** MyBase.Load

'Esta função executada no arranque da aplicação

'Propriedades da Form1

Text = "Calculadora" 'A palavra Calculadora aparece na parte de cima da janela Form1

BackgroundImage = Image.FromFile("C:\Users\jalme\Google Drive\Docência\Informática Industrial 2021-2022\Práticas\Recursos gráficos\ua-top-1.jpg") 'Esta passa a ser a imagem de fundo da janela Form1

```
'BackgroundImageLayout = ImageLayout.Stretch
Height = 200 'A janela Form1 ocupa 200 por 500 pixels no écran
Width = 500
Icon = New Icon("C:\Users\jalme\Google Drive\Docência\Informática Industrial 2021-2022\Práticas\Recursos
gráficos\ua-favicon.ico") ' Esta imagem "icon" aparece no canto superior da janela Form1
```

'Propriedade da TextBox1

'A caixa de texto TextBox1, o seu canto superior esquerdo, tem as coordenadas X=50,Y=40

```
TextBox1.Location = New Point(50, 40)
```

```
TextBox1.Width = 100
```

```
TextBox1.Height = 30
```

```
TextBox2.Location = New Point(50, 120)
```

```
TextBox2.Width = 100
```

```
TextBox2.Height = 30
```

```
TextBox3.Location = New Point(300, 75)
```

```
TextBox3.Height = 100
```

```
TextBox3.Width = 30
```

```
Button1.Text = "+"
```

Button1.BackColor = Color.Red 'O botao passa a ter a cor de fundo vermelha

Button1.ForeColor = Color.Yellow 'e o texto a amarelo

```
Button1.Location = New Point(175, 75)
```

```
Button1.Visible = True
```

```
Button1.Height = 40
```

```
Button1.Width = 100
```

```
Label1.Text = "X"
```

Label1.BackColor = Color.Yellow 'A cor de fundo do Label1 passa a amarelo

Label1.ForeColor = Color.Black 'e o texto que aparece no Label1, neste caso a letra X, passa preto

```
Label1.Width = 100
```

```
Label1.Location = New Point(50, 20)
```

```
Label2.Text = "Y"
```

```
Label2.BackColor = Color.Yellow
```

```
Label2.ForeColor = Color.Black
```

```
Label2.Width = 100
```

```
Label2.Location = New Point(50, 100)
```

```
Label3.Text = "Z"
```

```
Label3.BackColor = Color.Yellow
```

```
Label3.ForeColor = Color.Black
```

```
Label3.Width = 100
```

```
Label3.Location = New Point(300, 55)
```

```
End Sub
```

' Quando o utilizador do programa permir o botao (+) da interface Form1 esta subrotina será executada

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
```

' Os conteúdos das caixas de texto da esquerda são convertidos em números inteiros

' e o resutado da sua soma é visulizado na caixa de texto da direita (textbox3)

```
If IsNumeric(TextBox1.Text) And IsNumeric(TextBox2.Text) Then
```

```

        TextBox3.Text = CInt(TextBox1.Text) + CInt(TextBox2.Text)
    End If
End Sub
End Class

```

Depois de apresentarmos a interface e o código, vamos analisar com maior detalhe as propriedades de cada um dos controlos/objetos utilizados na interface.

Objeto “Form1”	
Altere o título da janela “Form1” para “Calculadora”	Form1.Text = "Calculadora"
Altere a altura da janela “Form1” para 200 pixels	Form1.Height = 200
Altere a largura da janela “Form1” para 500 pixels	Form1.Width = 500
Altere a imagem do canto superior esquerdo da janela Form1	Form1.Icon = New Icon("...")
Altere a imagem de fundo da janela Form1	Form1.BackgroundImage = Image.FromFile("...")

Arraste um controlo do tipo “Label” para a janela “Form1”. A partir desse momento passa a ter um objeto “Label1” visível na interface. Este controlo tem o aspecto de uma linha de texto, permite ao utilizador ler as mensagens de texto que o programador previamente definiu.

Objeto “Label1”	
Escreva a mensagem “Caixa de texto” na caixa de texto	Label1.text = "X"
Altere a cor de fundo da janela para amarelo	Label1.BackColor = Color.Yellow
Altere a cor de fundo da janela para amarelo	Label1.ForeColor = Color.Black
Altere a largura da caixa de texto para 100 pixels	Label1.Width = 100
Altere a imagem de fundo da janela Form1	Form1.BackgroundImage = Image.FromFile("...")

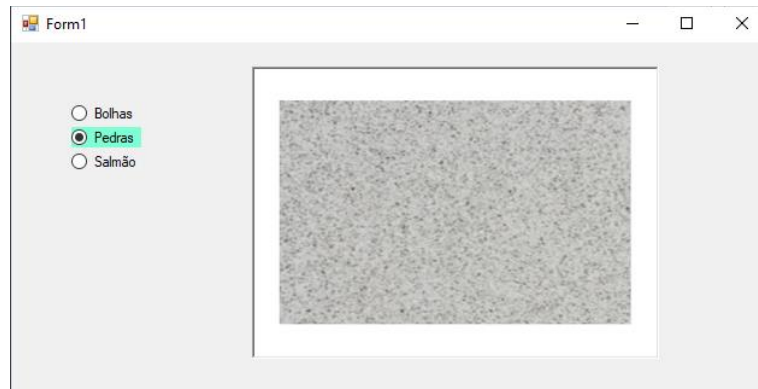
Arraste um controlo do tipo “TextBox” para a janela “Form1”. A partir desse momento passa a ter uma caixa de texto com o nome “TextBox1”. Este controlo tem o aspecto de uma caixa de texto, permite ao utilizador e ao programador escrever ou ler mensagens de texto nela.

Arraste um controlo do tipo "Button" para a janela "Form1". A partir desse momento passa a ter um botão com o nome "Button1". Este controlo permite ao utilizador "clicar" sobre o botão fazendo com que o código associado a este botão seja executado.

Objeto "Button1"	
Escreva a mensagem "+" na propriedade	Button1.text = "+ "
Altere a cor de fundo do botão para vermelho	Button1.BackColor = Color.Red
Altere a cor da letra para amarelo	Button1.ForeColor = Color.Yellow
Altere a altura do botão para 40 pixeis	Button 1.height = 40
Altere a largura do botão para 100 pixeis	Button1.Width = 100

Exemplo “Seleção de imagens”

Neste exemplo, o utilizador pode seleccionar uma de várias imagens na interface *Form1* do programa. O utilizador selecciona um de três *Radiobutton*, fazendo aparecer a imagem respectiva na *PictureBox* existente na janela.



Public Class **Form1**

Private Sub **Form1_Load**(sender As Object, e As EventArgs) Handles MyBase.Load

With RadioButton1

.Text = "Bolhas"

.Location = New Point(50, 50)

.Width = 50

.Height = 25

End With

With RadioButton2

.Text = "Pedras"

.Location = New Point(50, 70)

.Width = 50

.Height = 25

.Select()

.BackColor = Color.Aquamarine

End With

With RadioButton3

.Text = "Salmão"

.Location = New Point(50, 90)

.Width = 50

.Height = 25

End With

With PictureBox1

.Location = New Point(200, 20)

.BackColor = Color.AliceBlue

.BorderStyle = BorderStyle.Fixed3D

End With

End Sub

Private Sub **RadioButton1_CheckedChanged**(sender As Object, e As EventArgs) Handles

RadioButton1.CheckedChanged

RadioButton1.BackColor = Color.Aquamarine

RadioButton2.BackColor = Color.Empty

```
RadioButton3.BackColor = Color.Empty
PictureBox1.BackgroundImage = Image.FromFile("C:\Users\jalme\Google Drive\Docência\Informática Industrial
2021-2022\Práticas\Recursos gráficos\bolhas.jpg")
PictureBox1.BackgroundImageLayout = ImageLayout.Stretch
End Sub
```

```
Private Sub RadioButton2_CheckedChanged(sender As Object, e As EventArgs) Handles
RadioButton2.CheckedChanged
    RadioButton2.BackColor = Color.Aquamarine
    RadioButton1.BackColor = Color.Empty
    RadioButton3.BackColor = Color.Empty
    PictureBox1.BackgroundImage = Image.FromFile("C:\Users\jalme\Google Drive\Docência\Informática Industrial
2021-2022\Práticas\Recursos gráficos\pedra.jpg")
    PictureBox1.BackgroundImageLayout = ImageLayout.Stretch
End Sub
```

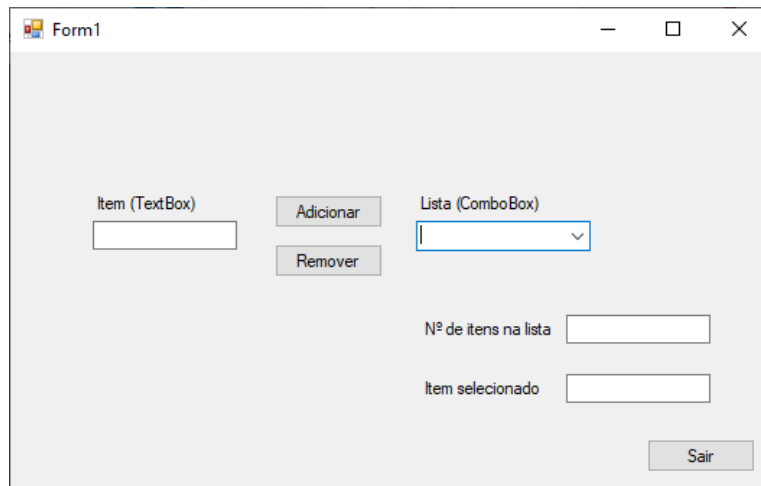
```
Private Sub RadioButton3_CheckedChanged(sender As Object, e As EventArgs) Handles
RadioButton3.CheckedChanged
    RadioButton3.BackColor = Color.Aquamarine
    RadioButton1.BackColor = Color.Empty
    RadioButton2.BackColor = Color.Empty
    PictureBox1.BackgroundImage = Image.FromFile("C:\Users\jalme\Google Drive\Docência\Informática Industrial
2021-2022\Práticas\Recursos gráficos\salmao.jpg")
    PictureBox1.BackgroundImageLayout = ImageLayout.Stretch
End Sub
End Class
```

Exemplo “Lista de Strings”

Neste exemplo pretende-se apresentar o controlo *ComboBox*.

O texto que o utilizador introduzir na *TextBox* da esquerda será adicionado ou removido da *ComboBox* da direita quando o utilizador premir o botão adicionar ou remover, respectivamente.

Nesta interface Form1 o programador utiliza cinco controlos do tipo Label, três controlos do tipo *Textbox*, três controlos do tipo *Button* e um controlo do tipo *ComboBox*.



Public Class Form1

' Quando o utilizador clicar no botao Adicionar esta subrotina é chamada

Private Sub btnAdicionar_Click(sender As Object, e As EventArgs) Handles btnAdd.Click

cbxList.Items.Add(txtItem.Text) ' adiciona à combobox o texto presente na textbox

txtNItens.Text = cbxList.Items.Count ' visualiza na textbox o n°mero de linhas/itens presentes na ComboBox

End Sub

' Quando o utilizador clicar no botao Remover esta subrotina é chamada

Private Sub btnRemove_Click(sender As Object, e As EventArgs) Handles btnRemove.Click

cbxList.Items.Remove(txtItem.Text) 'remove da combobox o item/linha/texto presente na textbox

txtNItens.Text = cbxList.Items.Count

End Sub

' Quando o utilizador selecionar na combobox uma nova linha esta subrotina é chamada

Private Sub cbxList_SelectedIndexChanged(sender As Object, e As EventArgs) Handles

cbxList.SelectedIndexChanged

txtSelected.Text = cbxList.SelectedItem 'Visualiza a linha/item selecionado pelo utilizador

End Sub

'Esta subrotina é chamada quando o utilizador selecionar o botao Sair

Private Sub btnClose_Click(sender As Object, e As EventArgs) Handles btnClose.Click

End 'Esta instrução termina a execução do programa

End Sub

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

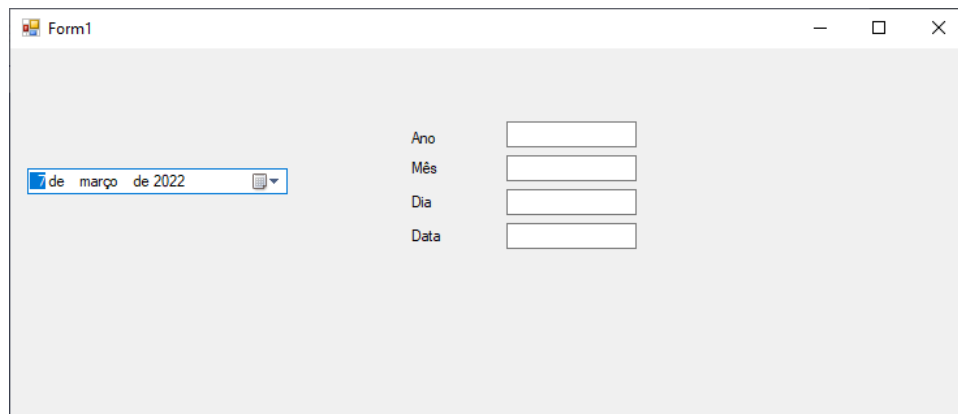
End Class

Exemplo “Calendário”

Neste exemplo pretende-se apresentar o controlo *DateTimePicker*.

Quando o utilizador seleccionar uma nova data, usando o controlo *DateTimePicker* (à esquerda na figura), o sub procedimento *DateTimePicker1_ValueChanged* será executado e nas textbox da direita aparecerá o ano, o mês, o dia e a data que o utilizador seleccionou.

Nesta interface o programador utiliza um controlo do tipo *DateTimePicker*, quatro controlos tipo *Label*, e quatro controlos do tipo *Textbox*.



```
Public Class Form1
```

```
    ' Quando o utilizador seleccionar na DateTimePicker uma nova data esta subrotina é chamada
```

```
    Private Sub DateTimePicker1_ValueChanged(sender As Object, e As EventArgs) Handles
```

```
        DateTimePicker1.ValueChanged
```

```
            txtYear.Text = DateTimePicker1.Value.Year
```

```
            txtMonth.Text = DateTimePicker1.Value.Month
```

```
            txtDay.Text = DateTimePicker1.Value.Day
```

```
            txtDate.Text = DateTimePicker1.Value.Date
```

```
        End Sub
```

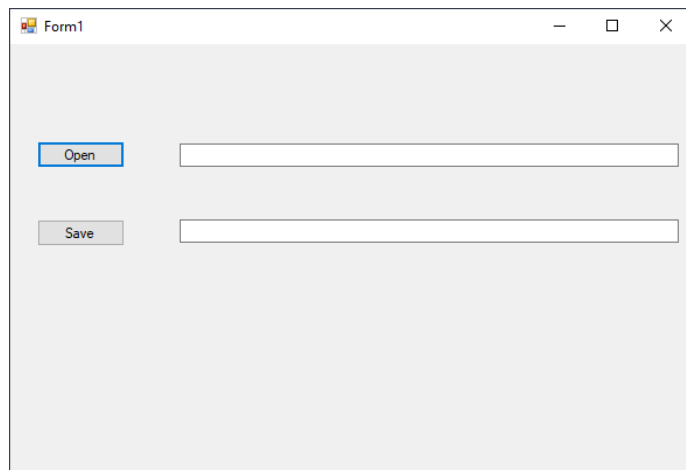
```
    End Class
```

Exemplo “Selecionar um ficheiro no disco”

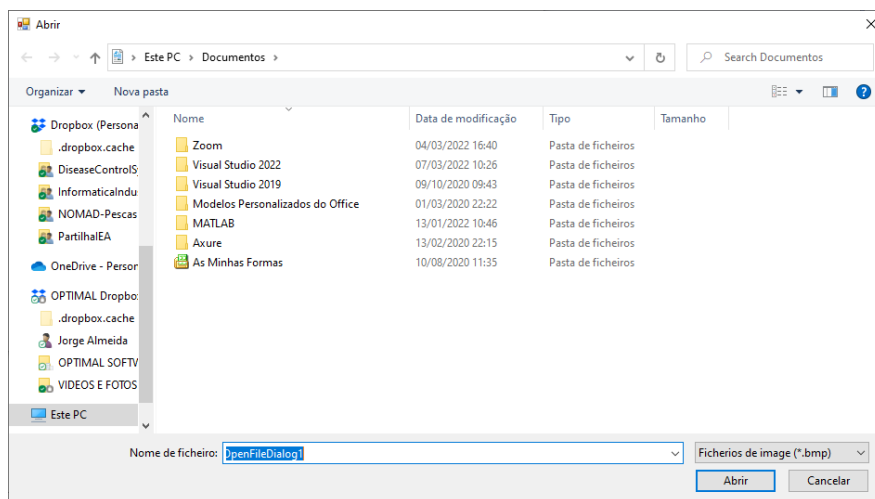
Neste exemplo pretende-se apresentar os controlos *SaveFileDialog* e *OpenFileDialog*.

Ao contrário do que o nome deste tipo de objetos daria a entender, eles não permitem gravar ou abrir ficheiros em disco, permitem apenas que o utilizador veja o conteúdo do disco e selecione o diretório e o nome do ficheiro que pretende posteriormente abrir ou onde pretende gravar dados.

Neste exemplo, o programador utiliza dois controlos do tipo *Label*, dois controlos do tipo *Textbox*, dois controlos do tipo *Button*, um controlo do tipo *SaveFileDialog* e outro do tipo *OpenFileDialog*.



Quando o utilizador pressionar o botão “*Open*”, o método *Showdialog()* do objeto *OpenFileDialog* é executado e surge no ecrã uma janela de diálogo que permite ao utilizador navegar no disco, visualizar os diretórios e ficheiros existentes.



Quando o utilizador selecionar um dos ficheiros existentes, a sua localização e nome ficam guardados na propriedade *FileName* do objeto *OpenFileDialog1* e essa localização e nome podem ser por exemplo visualizadas numa *textbox* (`TextBox1.Text = OpenFileDialog1.FileName`).

Se o utilizador sair da janela de diálogo sem selecionar um ficheiro “Cancel”, a função `OpenFileDialog1.ShowDialog()` retorna o número dois, o que poderá ser visualizado numa *textbox* (`TextBox1.Text = OpenFileDialog1.ShowDialog()`).

Arraste um controlo do tipo *"OpenFileDialog"* para a janela *"Form1"*. A partir desse momento passa a ter um objeto com o nome *"OpenFileDialog1"*. Este controlo permite ao utilizador navegar no disco e visualizar os diretórios e ficheiros existentes. Depois da janela de diálogo estar aberta OpenFileDialog.ShowDialog() o utilizador pode selecionar um ficheiro ou cancelar e sair da janela.

Se sair da janela depois de ter selecionado um ficheiro esta função retorna o número 1 e o nome e localização do ficheiro é guardada na propriedade FileName, se o utilizador sair da janela de diálogo fazendo "cancel" esta função retorna o número 2.

Public Class Form1

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

' Filter: Quando a Janela de di-logo aparecer ao utilizador

' serao mostrados apenas os ficheiros com a extensao *.bmp

OpenFileDialog1.Filter = "Ficherios de image|*.bmp|jpg|*.jpg"

End Sub

Private Sub btnOpen_Click(sender As Object, e As EventArgs) Handles btnOpen.Click

Dim result As Integer

' Mostra ao utilizador a janela de diálogo

' Na janela de diálogo o utilizador pode retornar "Cancel"

' sem selecionar nenhum ficheiro.

' Se o ficheiro foi seleccionado a função retorna 1 caso contrário retorna 2

result = OpenFileDialog1.ShowDialog()

' Se um ficheiro imagem foi seleccionado

If result = 1 Then

' Quando o utilizador sai da janela de diálogo,

' o nome e a localização do ficheiro seleccionado

' está memorizado na propriedade "FileName"

txtOpen.Text = OpenFileDialog1.FileName

Else

txtOpen.Text = "Nenhum ficheiro selecionado"

End If

End Sub

Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles btnSave.Click

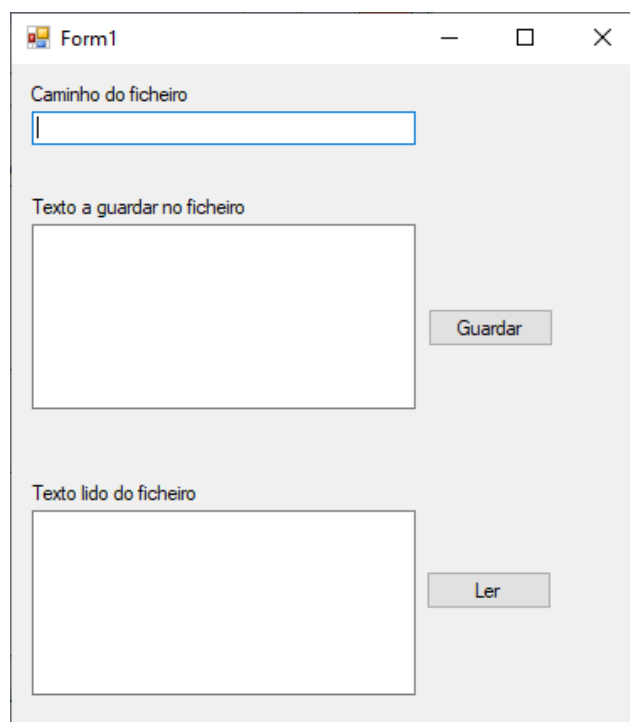
SaveFileDialog1.ShowDialog()

txtSave.Text = SaveFileDialog1.FileName

End Sub

End Class

Exemplo “ler e graver ficheiros de texto”



Neste exemplo pretende-se abrir ficheiros de texto e visualizar o seu conteúdo numa caixa de texto. Pretende-se também gravar uma string num ficheiro texto.

StreamWriter e StreamReader

Os objetos do tipo *StreamReader* e *StreamWriter* permitem, respectivamente, ler ou escrever em ficheiros de texto.

Para poder usar objetos do tipo *StreamReader* e *StreamWriter* deve importar o “System.IO”, acrescentando no seu código:

```
Imports System.IO
```

Para escrever o texto da Caixa de texto no ficheiro, deve declarar um objeto do tipo *StreamWriter*, por exemplo com o nome “fileToSave”, fazendo:

```
Dim fileToSave As StreamWriter
```

Para guardar o texto no ficheiro, apagando o que lá estava antes, deve fazer:

```
fileToSave = New StreamWriter(txtPath.Text, false)  
fileToSave.Write(txtWrite.Text)
```

Para acrescentar o texto “Ola”, mantendo o que estava antes, deve fazer:

```
fileToSave = New StreamWriter(txtPath.Text, true)  
fileToSave.Write("Olá")
```

Finalmente, para que o texto seja de facto escrito em disco deve fazer:

```
fileToSave.Close()
```

Para ler o conteúdo de um ficheiro de texto, por exemplo, o conteúdo do ficheiro a seleccionar, deve declarar um objeto do tipo StreamReader, por exemplo com o nome "file", fazendo:

```
Dim result As Integer
result = OpenFileDialog1.ShowDialog()
If result = 1 Then
    Dim file As StreamReader
    file = New StreamReader(OpenFileDialog1.FileName)
    textbox.Text = file.ReadToEnd()
    file.Close()
End If
```

Após esta apresentação dos objetos do tipo StreamReader e StreamWriter será fácil perceber o exemplo seguinte:

```
Imports System.IO
```

```
Public Class Form1
```

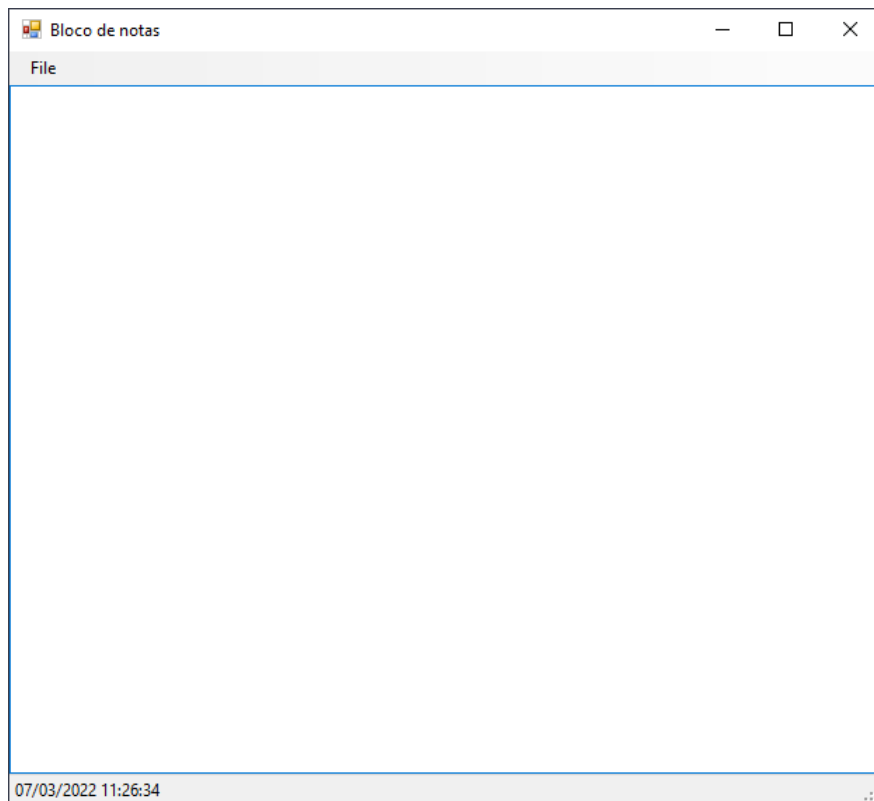
```
    Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles btnSave.Click
        Dim fileToSave As StreamWriter
        fileToSave = New StreamWriter(txtPath.Text, True)
        fileToSave.Write("Olá")
        fileToSave.Close()
    End Sub
```

```
    Private Sub button_Click(sender As Object, e As EventArgs) Handles btnRead.Click
        Dim result As Integer
        result = OpenFileDialog1.ShowDialog()
        If result = 1 Then
            Dim file As StreamReader
            file = New StreamReader(OpenFileDialog1.FileName)
            textbox.Text = file.ReadToEnd()
            file.Close()
        End If
    End Sub
```

```
End Class
```

5. Trabalho número 1 – Editor de texto

Pretende-se desenvolver um pequeno editor de texto que permita abrir e gravar ficheiros utilizando os objetos do tipo `MenuStrip`, `StatusStrip`, `OpenFileDialog`, `SaveFileDialog`, `StreamReader` e `StreamWriter`.



Imports System.IO

Public Class Form1

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Me.Text = "Bloco de notas"

LastOperationStatusLabel.Text = ""

OpenFileDialog1.Filter = "Ficheiros de texto|*.txt"

OpenFileDialog1.FileName = ""

SaveFileDialog1.Filter = "Ficheiros de texto|*.txt"

UpdateLastChange("")

End Sub

Private Sub OpenToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles

OpenToolStripMenuItem.Click

Dim result = OpenFileDialog1.ShowDialog()

If result = 1 Then

Dim fileToOpen = New StreamReader(OpenFileDialog1.FileName)

textbox.Text = fileToOpen.ReadToEnd()

Me.Text = OpenFileDialog1.FileName

UpdateLastChange("Ficheiro aberto com sucesso.")

End If

End Sub

```

Private Sub SaveToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
SaveToolStripMenuItem.Click
    If textbox.Text.Length = 0 Then
        UpdateLastChange("Não existem dados a guardar no ficheiro.")
        Return
    End If

    SaveFileDialog1.FileName = OpenFileDialog1.FileName
    Dim result = SaveFileDialog1.ShowDialog()

    If result = 1 Then
        Dim fileToSave As StreamWriter
        fileToSave = New StreamWriter(SaveFileDialog1.FileName, False)
        fileToSave.Write(textbox.Text)
        fileToSave.Close()
        Me.Text = SaveFileDialog1.FileName
        UpdateLastChange("Ficheiro guardado com sucesso")

    End If
End Sub

Private Sub UpdateLastChange(status)
    LastChangeStatusLabel.Text = DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss")
    LastOperationStatusLabel.Text = status
End Sub

Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ExitToolStripMenuItem.Click
    End
End Sub
End Class

```