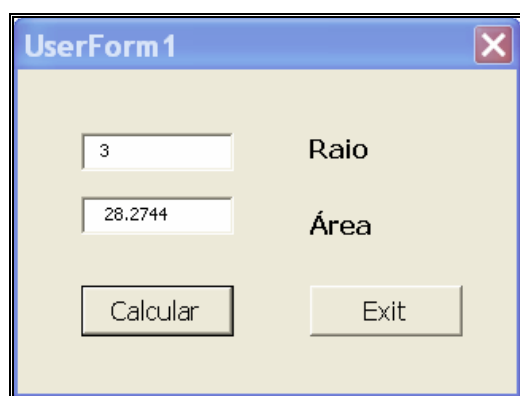


Capítulo 18

Caixas de diálogo personalizadas (formulários, userforms)

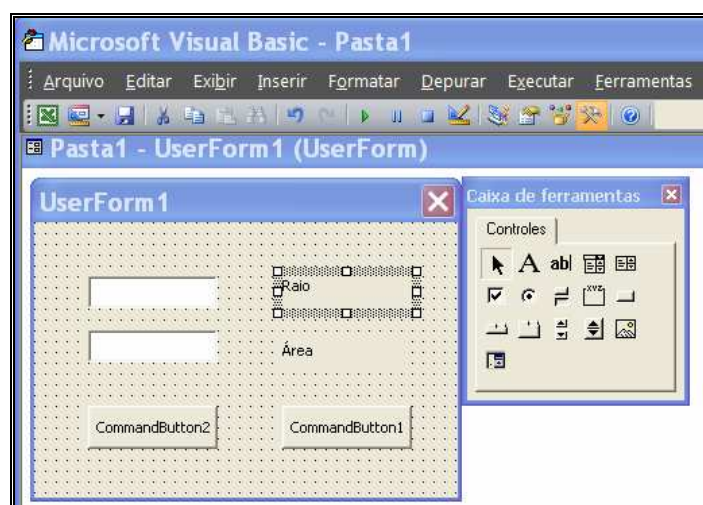
Na seguinte figura, vemos uma calculadora que determina a área de um círculo.



A pergunta óbvia é: Puxa, como se faz isto?

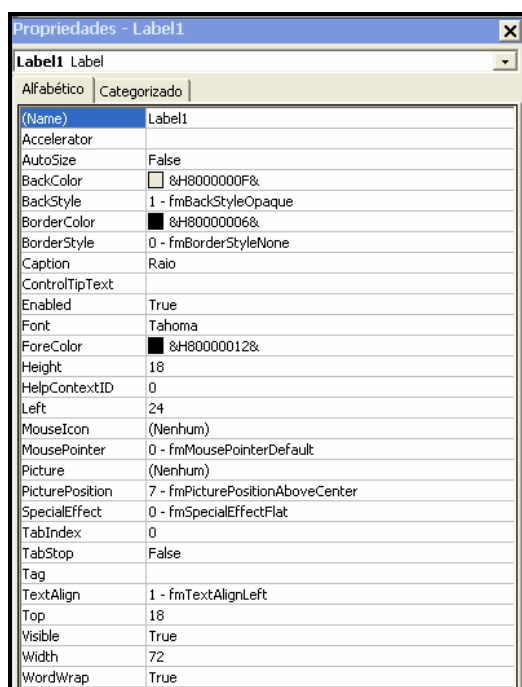
Resposta: Abra o editor do VBA (Alt+F11) > *Inserir* > *UserForm*.

Você vai ver uma grade de pontos ("grid") e uma Caixa de ferramentas. Para criar a calculadora, você precisará de três destas ferramentas: dois rótulos (Label, "A"), dois caixas de texto (TextBox, "ab") -para inserir o raio e para mostrar o resultado- e dois botões de comando (CommandButton). Os números nas caixas de texto serão interpretados como strings, ou seja textos. Dê um clique com o botão esquerdo sobre uma ferramenta e arraste a sobre a grade de pontos. Este conjunto de grade e objetos é o **formulário**. Ajuste a posição e o tamanho destes elementos no formulário conforme indicado na figura.



Agora podemos substituir os "Labels" e as inscrições nos botões pelas designações desejadas, por exemplo: "Raio", "Área", "Calcular" e "Exit".

(As propriedades de quaisquer controles no formulário podem ser mudadas. Mas pode-se também usar as designações propostas pelo editor do formulário.) Dê um clique no elemento a editar e coloque, com outro clique, o cursor no texto. Escreva "Raio", etc. Selecione o novo texto e dê um clique do botão direito nele. Abra a janela *Propriedades* (F4). Com *Font* podemos ajustar o fonte, o estilo e o tamanho das letras. Com *Width* = 35 reduzimos a largura da caixa a 35 pontos. Este processo repetimos com o outro rótulo e com os botões. (Podemos, também, mudar o tamanho do elemento e sua posição diretamente com o mouse. Podemos fazer todo o processo da nomeação com a janela de Propriedades aberta e editando um controle após outro. Note que com "Caption" damos um nome a um botão.)



Também podemos nomear as duas caixas de texto -e o formulário mesmo- pressionando F4. No código, vamos referirmos a estes controles por meio dos seus nomes, "raio" e "area". (Poderíamos usar os nomes propostos pelo editor, mas eles fazem com que o código seja um pouco críptico. Mais à frente, utilizamos esta técnica no exemplo da calculadora para números complexos.)

Finalmente, é preciso escrever, para cada botão de comando, um pequeno procedimento que dirá ao botão o que deve fazer quando é pressionado. (Precisamos escrever o código que será executado quando o evento Click dispara; caso contrário, clicar no botão não terá conseqüências, nada acontecerá.)

```

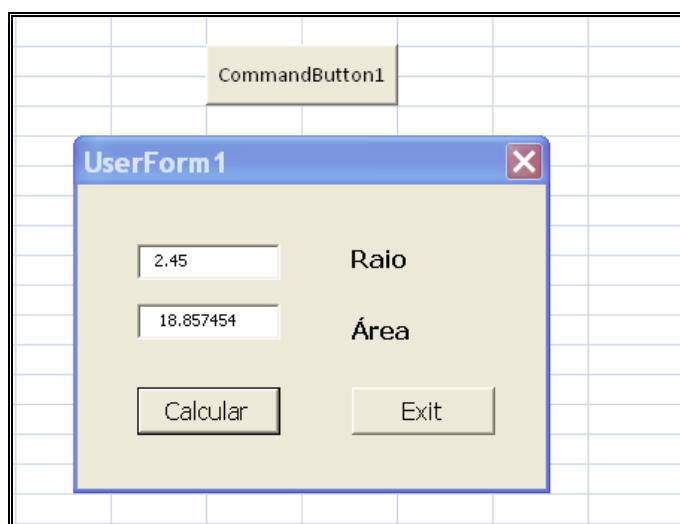
Private Sub CommandButton1_Click()
    Unload Circulo 'ou: Unload Me
End Sub

Private Sub CommandButton2_Click()
    Dim r, A As Double
    r = Val(raio)
    A = r * r * 3.1416
    area = Str(A)
End Sub

```

Com F5 podemos executar o formulário. (Deve-se usar um ponto decimal no valor do raio, pois, no capítulo 7, vimos que Val não reconhece vírgulas.)

Seria muito agradável, poder executar o formulário diretamente da planilha por meio de um botão de comando.



Isso pode ser realizado facilmente:

Na planilha, selecionamos por meio de *Desenvolver>Inserir>Controles ActiveX* um botão de comando. Dê um duplo-clique com o botão esquerdo do mouse sobre o botão de comando e escreva o seguinte código:

```

Private Sub CommandButton1_Click()
    Load Circulo
    Circulo.Show
End Sub

```


Para ativar o botão pode ser necessário executar, primeiro, o formulário com F5.

Exemplos:

Triângulo

No formulário, vemos uma calculadora que determina a área de um triângulo por meio de $A = \sqrt{s(s-a)(s-b)(s-c)}$ onde $s = (a+b+c)/2$.

No código do botão "Calcular", incluímos também as condições sob as quais os três números introduzidos formam um triângulo, ou seja " $a < b+c$ e $b < a+c$ e

$c < a+b$ ". Também foi incluído o controle "Quadro" com o ícone . O desenho mesmo foi feito com o "Paint" e introduzido no formulário com Ctrl+C, Ctrl+V. (Deve-se selecionar na janela *Propriedades* a propriedade *Picture*, colocando o cursor na segunda coluna, ou seja sobre "(Nenhum)".) Neste exemplo, o controle Quadro foi introduzido para mostrar que ele existe e como pode ser usado. Muita utilidade não tem, neste exemplo. Mas, em outros casos, pode ser bastante útil, por exemplo num formulário para determinar as correntes, tensões, etc. num circuito elétrico ilustrado num "Quadro".

```
Private Sub CommandButton1_Click()
    Dim a, b, c, s, F As Double
    a = Val(lado_a)
    b = Val(lado_b)
    c = Val(lado_c)
    If a < b + c And b < a + c And c < a + b Then
        s = (a + b + c) / 2
        F = (s * (s - a) * (s - b) * (s - c)) ^ (1 / 2)
        area = Str(F)
    Else: MsgBox "Não é nenhum triângulo"
    End If
End Sub

Private Sub CommandButton2_Click()
    Unload Triângulo
End Sub
```

O código do botão na planilha reza

```
Private Sub CommandButton1_Click()
    Load Triângulo
    Triângulo.Show
End Sub
```

Uma calculadora para números complexos

No capítulo 7, falamos muito sobre os números complexos e sua implementação no Excel. No capítulo 7, encontramos também as fórmulas necessárias para criar a calculadora. O exemplo calculado embaixo, foi tomado do mesmo capítulo, trata-se da divisão dos números $z_1 = -0,5 - 0,866i$ e $z_2 = -1 + 1i$.

Os botões **+**, **-**, **x** e **/** foram feitas reduzindo apropriadamente o tamanho de quatro botões de comando. Neste exemplo, usamos os "Labels" e os "TextBoxes" que o editor fornece, ou seja, não foram introduzidos novos nomes ou títulos. Este método não é muito aconselhável, mas, ele resulta ser muito rápido e efetivo quando o número dos controles for muito grande.

Como sempre, criamos, primeiro, o formulário e só depois escrevemos o código. (Clique no menu *Janela* do Visual Basic Editor, para poder eleger entre código ou formulário.)

```

Private Sub CommandButton1_Click()
    a1 = Val(TextBox1)
    b1 = Val(TextBox2)
    a2 = Val(TextBox3)
    b2 = Val(TextBox4)
    a = a1 + a2
    b = b1 + b2
    TextBox5 = Str(a)
    TextBox6 = Str(b)
End Sub

Private Sub CommandButton2_Click()
    a1 = Val(TextBox1)
    b1 = Val(TextBox2)
    a2 = Val(TextBox3)
    b2 = Val(TextBox4)
    a = a1 - a2
    b = b1 - b2
    TextBox5 = Str(a)
    TextBox6 = Str(b)
End Sub

Private Sub CommandButton3_Click()
    a1 = Val(TextBox1)
    b1 = Val(TextBox2)
    a2 = Val(TextBox3)
    b2 = Val(TextBox4)
    a = a1 * a2 - b1 * b2
    b = a1 * b2 + a2 * b1
    TextBox5 = Str(a)
    TextBox6 = Str(b)
End Sub

Private Sub CommandButton4_Click()
    a1 = Val(TextBox1)
    b1 = Val(TextBox2)
    a2 = Val(TextBox3)
    b2 = Val(TextBox4)
    d = a2 ^ 2 + b2 ^ 2
    a = (a1 * a2 + b1 * b2) / d: b = (a2 * b1 - a1 * b2) / d
    TextBox5 = Str(a)
    TextBox6 = Str(b)
End Sub

Private Sub CommandButton5_Click()
    Unload Me
End Sub

Private Sub CommandButton6_Click()
    TextBox1 = ""
    TextBox2 = ""
    TextBox3 = ""
    TextBox4 = ""
    TextBox5 = ""
    TextBox6 = ""
End Sub

```

Os procedimentos são rapidamente escritos, pois trata-se, essencialmente, de um original e três cópias do mesmo código.

O código para o botão de comando na planilha contém a designação padrão "UserForm1".

```
Private Sub CommandButton1_Click()  
    Load UserForm1  
    UserForm1.Show  
End Sub
```

Antes de usar o botão, deve-se executar uma vez o formulário com F5.