

## Tarefa Orientada 5

### **Código do formulário *frmCalcularInvestimento***

#### Objectivos:

- Desenvolver código para responder a eventos
- Efectuar validação de dados

Nesta tarefa iremos analisar o código desenvolvido para o formulário *frmCalcularInvestimento*.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *Activated* associado ao formulário *frmCalcularInvestimento*. Este evento ocorre quando o formulário *frmCalcularInvestimento* se torna no formulário activo. O código permite mostrar uma nova mensagem no painel central *sblblDescrição* da barra de estados do formulário.

```
Public Class frmCalcularInvestimento
    Inherits System.Windows.Forms.Form

    Private Sub frmCalcularInvestimento_Activated(ByVal sender As _
Object, ByVal e As System.EventArgs) Handles Me.Activated

        frmPrincipal.sbPrincipal.Items("sblblDescrição").Text _
            = "Calcular o valor acumulado de um investimento com _
pagamentos mensais"

    End Sub
```

Note-se que por a barra de estados *sbPrincipal* pertencer ao formulário (pai) *frmPrincipal*, e por a propriedade *Text* estar a ser definida a partir do formulário (filho) *frmCalcularInvestimento*, a instrução de atribuição inicia-se com a identificação do formulário *frmPrincipal* seguido da barra de estados *sbPrincipal*. Depois é utilizada a propriedade *Items* da barra de estados com o valor *sblblDescrição*, de modo a identificar o painel central da barra de estados que irá mostrar o texto indicado na parte direita da instrução de atribuição.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento

*Click* associado ao botão *btnCalcular*.

```

Private Sub btnCalcular_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCalcular.Click

    If rdoValorAcumulado.Checked Then
        If ValidarEntradas(txtInvetimentoMensal.Text, 100, 1000) Then
            If ValidarEntradas(txtTaxaAnual.Text, 0, 24) Then
                If ValidarEntradas(txtNúmeroAnos.Text, 1, 65) Then
                    txtValorAcumulado.Text =
FormatCurrency(Me.CalcularValorAcumulado(txtInvetimentoMensal.Text, _
txtTaxaAnual.Text / 12 / 100, txtNúmeroAnos.Text * 12))
                    txtInvetimentoMensal.Focus()
                Else
                    MessageBox.Show("Deve Introduzir um número entre 1 e
65 para o número de anos.", "Erro")
                    txtNúmeroAnos.Focus()
                End If
            Else
                MessageBox.Show("Deve Introduzir um número entre 0 e 24
para a taxa anual.", "Erro")
                txtTaxaAnual.Focus()
            End If
        Else
            MessageBox.Show("Deve Introduzir um número entre 100 e 1000
para o investimento mensal.", "Erro")
            txtInvetimentoMensal.Focus()
        End If
    Else
        If ValidarEntradas(txtTaxaAnual.Text, 0, 24) Then
            If ValidarEntradas(txtNúmeroAnos.Text, 1, 65) Then
                If ValidarEntradas(txtValorAcumulado.Text, 1000,
1000000) Then
                    Me.CalcularInvestimentoMensal()
                    txtTaxaAnual.Focus()
                Else
                    MessageBox.Show("Deve Introduzir um número entre
1000 e 1000000.", "Erro")
                    txtValorAcumulado.Focus()
                End If
            Else
                MessageBox.Show("Deve Introduzir um número entre 1 e
65.", "Erro")
                txtNúmeroAnos.Focus()
            End If
        Else
            MessageBox.Show("Deve Introduzir um número entre 0 e 24.", _
"Erro")
            txtTaxaAnual.Focus()
        End If
    End If

End Sub

```

Neste procedimento de evento começa-se por verificar se o *radio buton* *rdoValorAcumulado* está, ou não, seleccionado.

Em caso afirmativo, utiliza-se a função *ValidarEntradas* para verificar se todas as entradas contêm valores válidos. Se as entradas forem todas válidas, chama-se a função *CalcularValorAcumulado* para calcular o valor

acumulado correspondente ao montante do investimento mensal, taxa de juro e duração do investimento, introduzidos nas caixas de texto adequadas. O resultado é apresentado na caixa de texto *txtValorAcumulado*. Para tal, usa-se a sua propriedade *Text*. Note-se que se utilizou a função *FormatCurrency* para apresentar o resultado no formato monetário. Sempre que exista uma entrada não válida, apresenta-se a mensagem de erro correspondente, através da utilização do método *Show* da classe *MessageBox*, e define-se o focus para a caixa de texto apropriada, através da utilização do método *focus*.

Caso o *radio boton rdoValorAcumulado* não esteja seleccionado, significa que está seleccionado o *radio boton rdoInvestimentoMensal*. Nessa situação, utiliza-se a função *ValidarEntradas* para verificar se todas as entradas contêm valores válidos e a função *CalcularInvestimentoMensal* para calcular o valor do investimento mensal que tem que se efectuar para obter um determinado valor acumulado, de acordo com a taxa anual de juros e o número de anos de duração do investimento introduzidos nas caixas de texto apropriadas.

A seguir, apresenta-se a função *CalcularValorAcumulado*.

```
Private Function CalcularValorAcumulado(ByVal InvestimetnoMensal As _
Decimal, ByVal TaxaJuroMensal As Decimal, ByVal Meses As Integer) _
As Decimal

    Dim ValorAcumulado As Decimal
    For indice As Integer = 1 To Meses
        ValorAcumulado = (ValorAcumulado + InvestimetnoMensal) _
            * (1 + TaxaJuroMensal)
    Next indice
    Return ValorAcumulado

End Function
```

A seguir, apresenta-se a função *CalcularInvestimentoMensal*.

```
Private Sub CalcularInvestimentoMensal()

    txtInvetimentoMensal.Text = _
        FormatCurrency(Pmt(txtTaxaAnual.Text / 12 / 100, _
            txtNúmeroAnos.Text * 12, 0, -txtValorAcumuladõ.Text, _
            DueDate.BegOfPeriod))

End Sub
```

Neste procedimento utiliza-se a função predefinida *Pmt* para calcular o investimento mensal que deve ser feito para obter um determinado valor acumulado, mediante uma determinada taxa de juros e uma determinada duração do período de investimento. Pode consultar a ajuda on-line para obter mais informações sobre esta função. O resultado é apresentado, em formato monetário, na caixa de texto *txtInvestimentoMensal*.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *CheckedChanged* associado ao *radio buton rdoValorAcumulado*. Este evento ocorre quando o controlo for seleccionado.

```
Private Sub rdoValorAcumulado_CheckedChanged(ByVal sender As _  
System.Object, ByVal e As System.EventArgs) Handles _  
rdoValorAcumulado.CheckedChanged  
  
    txtValorAcumulado.ReadOnly = True  
    txtValorAcumulado.TabStop = False  
    txtValorAcumulado.BackColor = SystemColors.Control  
    txtInvetimentoMensal.ReadOnly = False  
    txtInvetimentoMensal.TabStop = True  
    txtInvetimentoMensal.BackColor = SystemColors.Window  
  
    Me.LimparContraLos()  
    txtInvetimentoMensal.Focus()  
  
End Sub
```

Neste procedimento utiliza-se a propriedade *ReadOnly* do controlo *Text Box* para o utilizador poder, ou não, introduzir dados na caixa de texto correspondente. Usa-se a propriedade *TabStop* do controlo *Text Box* para o utilizador poder, ou não, mover o focus para a caixa de texto correspondente. Define-se o valor *SystemColors.Control* para a propriedade *BackColor* do controlo *Text Box*, de modo a que o controlo pareça desactivado. Define-se o valor *SystemColors.Window* para a propriedade *BackColor* do controlo *Text Box*, para que o controlo pareça activado. Finalmente, Chama-se o procedimento *LimparContraLos* para “limpar” as caixas de texto e coloca-se o focus na caixa de texto *txtInvestimentoMensal*.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *CheckedChanged* associado ao *radio buton rdoInvestimentoMensal*.

```
Private Sub rdoInvestimentoMensal_CheckedChanged(ByVal sender As _  
System.Object, ByVal e As System.EventArgs) Handles _  
rdoInvestimentoMensal.CheckedChanged  
  
    txtValorAcumulado.ReadOnly = False  
    txtValorAcumulado.TabStop = True  
    txtValorAcumulado.BackColor = SystemColors.Window  
    txtInvetimentoMensal.ReadOnly = True  
    txtInvetimentoMensal.TabStop = False  
    txtInvetimentoMensal.BackColor = SystemColors.Control  
  
    Me.LimparContraLos()  
    txtTaxaAnual.Focus()  
  
End Sub
```

O código deste procedimento é análogo ao código do procedimento anterior.

A seguir, apresenta-se o procedimento *LimparControlos*.

```
Private Sub LimparControlos()  
  
    txtInvetimentoMensal.Text = ""  
    txtTaxaAnual.Text = ""  
    txtNúmeroAnos.Text = ""  
    txtValorAcumulado.Text = ""  
  
End Sub
```

Este procedimento permite “limpar” as caixas de texto do formulário.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *Click* associado ao botão *btnFechar*.

```
Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnFechar.Click

    Me.LimparControlos()
    Me.Hide()
End Sub

End Class
```

Neste procedimento começa-se por chamar o procedimento *LimparControlos* para “limpar” as caixas de texto do formulário. Depois, utiliza-se o método *Hide* do formulário para esconder o formulário, mas não o retirar da memória. Assim, o formulário pode voltar a ser mostrado quando for executado o método *Show*. Se fosse utilizado o método *Close*, então teria que ser criada uma nova instância do formulário.