

# Visual Basic .NET

Scroll Bars, Splitters, Track Bars, Pickers, Notify Icons,  
Tool Tips e Timers

Professor: Danilo Giacobbo

Página pessoal: [www.danilogiacobo.eti.br](http://www.danilogiacobo.eti.br)

E-mail: [danilogiacobo@gmail.com](mailto:danilogiacobo@gmail.com)

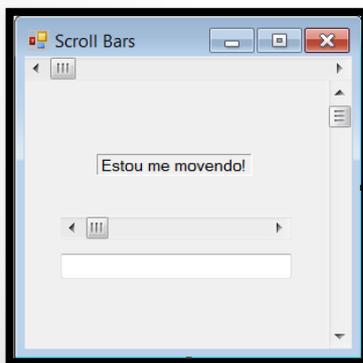
# Objetivos da aula

- ✓ Trabalhar com o controle **Scroll Bar**
- ✓ Trabalhar com o controle **Splitter**
- ✓ Trabalhar com o controle **Track Bar**
- ✓ Trabalhar com o controle **Data-Time Picker**
- ✓ Trabalhar com o controle **Month Calendar**
- ✓ Trabalhar com o controle **Notify Icon**
- ✓ Trabalhar com o controle **Tool Tip**
- ✓ Trabalhar com o controle **Timer**



# Scroll Bars

- Há dois tipos: **horizontal** e **vertical**.
- Os principais eventos são: **Scroll** e **ValueChanged**.
- Você pode usar as propriedades **Minimum** e **Maximum** para configurar o intervalo de valores que a pessoa pode selecionar.
- Você pode trabalhar também com as propriedades **SmallChange** e **LargeChange**.
- Os valores padrão para as propriedades de uma Scroll Bar são:
  - Minimum: 0
  - Maximum: 100
  - SmallChange: 1
  - LargeChange: 10
- Usando a propriedade **Value** você obtém a configuração atual do controle.



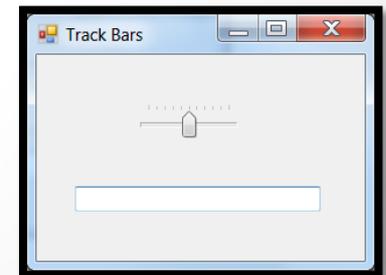
# Splitters

- Você pode usar esse controle para permitir que a pessoa redimensione outros controles.
- O exemplo abaixo mostra um controle do tipo **TextBox** que é redimensionado quando eu movo o controle **Splitter**.
- Ele é apenas visível quando eu passo o mouse perto dele.



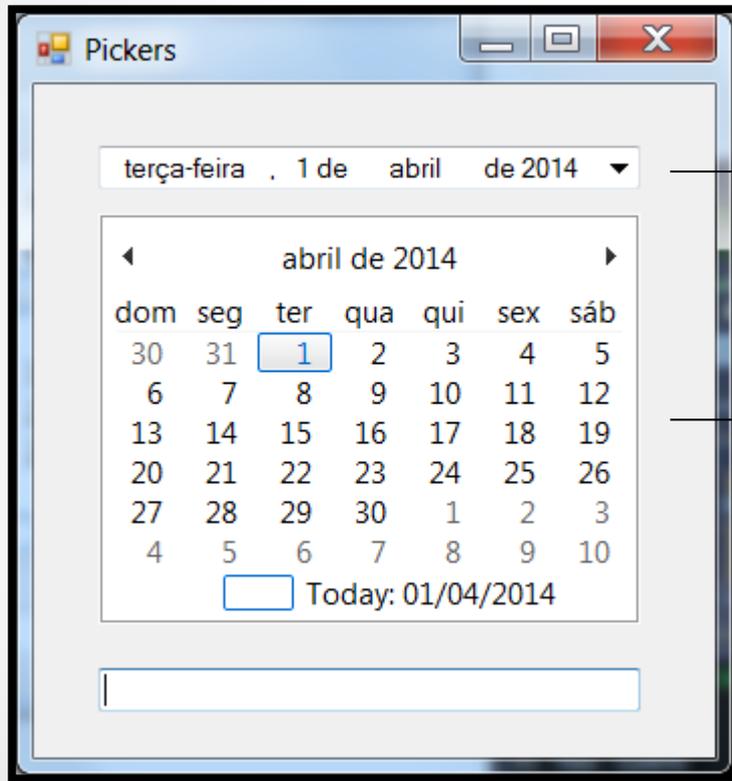
# Track Bars

- São parecidos com as Scroll Bars.
- Permitem que o usuário especifique o intervalo de valores a ser selecionado.
- Você pode configurar o intervalo com as propriedades **Minimum** (padrão = 0) e **Maximum** (padrão 10).
- Você pode trabalhar também com as propriedades **SmallChange** (padrão = 1) e **LargeChange** (padrão = 5).
- A propriedade **Value** contém o valor a ser incrementado em cada clique do usuário.
- Para exibir o controle em modo horizontal ou vertical é necessário alterar a propriedade **Orientation**.
- Por fim você ainda pode modificar as marcações da Track Bar alterando a propriedade **TickStyle**.
- A distância entre as marcações é controlada pela propriedade **TickFrequency** (o padrão é 1).



# Pickers

- Há dois tipos: **Date-Time** e **Month Calendar**.



→ **Data-Time Picker**

→ **Month Calendar**

# Date-Time Pickers

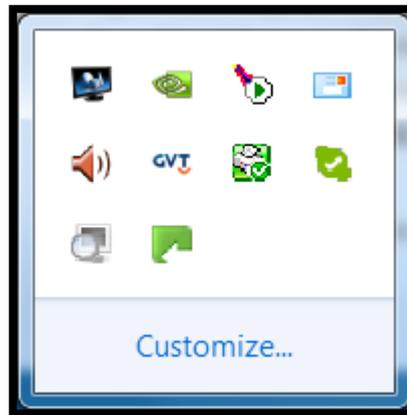
- Você pode digitar uma data neste controle ou selecionar uma a partir de um calendário exibido pelo mesmo.
- Você pode limitar as datas exibidas por meio das propriedades **MinDate** e **MaxDate**.
- Para mudar a aparência do calendário você usa as propriedades:
  - ✓ CalendarForeColor
  - ✓ CalendarFont
  - ✓ CalendarTitleBackColor
  - ✓ CalendarTitleForeColor
  - ✓ CalendarTrailingForeColor
  - ✓ CalendarMonthBackground
- Quando a pessoa seleciona uma data o evento **ValueChanged** ocorre.
- Você pode usar as propriedades **Text** ou **Value** para obter o valor exibido no controle.
- O tipo de objeto **DateTime** é bastante usado neste controle.
- A propriedade **Format** configura a exibição da data e hora do controle nos formatos pré-definidos ou customizados.
- Tente usar a propriedade **ShowUpDown** para ver o novo funcionamento do controle.

# Month Calendar

- Ele permite escolher a data e a hora de forma visual.
- Você pode limitar as datas exibidas por meio das propriedades **MinDate** e **MaxDate**.
- Quando uma nova data é selecionada o evento **DateSelected** ocorre.
- Quando uma data é selecionada o evento **DateChanged** ocorre.
- Este controle provê uma forma da pessoa selecionar um intervalo de datas por meio da propriedade **SelectionRange** que possui outras duas propriedades úteis: **Start** e **End**.
- Você pode usar as propriedades **SelectionStart** e **SelectionEnd** para diretamente acessar as datas selecionadas. Ambas retornam um objeto do tipo **DateTime**.
- Para mudar a aparência deste controle use as seguintes propriedades:
  - ForeColor
  - Font
  - TitleBackColor
  - TitleForeColor
  - TrailingForeColor
  - BackColor

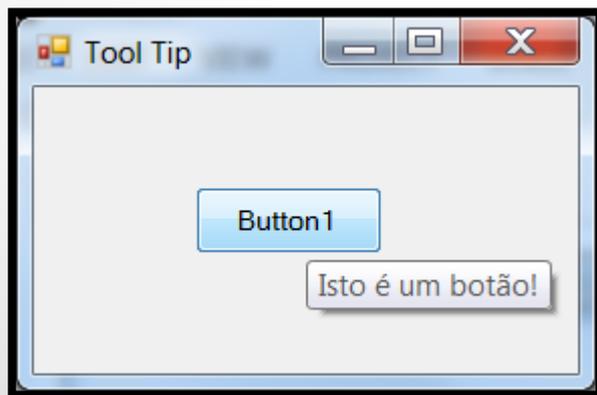
# Notify Icons

- Permite mostrar um ícone na barra de tarefas do Windows.
- Para configurar o ícone a ser mostrado você usa a propriedade **Icon**.
- Você pode escrever algum código no evento **DoubleClick** para que algo aconteça quando a pessoa clicar no ícone.
- Para fazer o ícone aparecer ou desaparecer você usa a propriedade **Visible**.
- A nova versão do VB .NET permite a você criar serviços **Windows** o que pode ser útil para este controle mostrar.



# Tool Tips

- São pequenas janelas que aparecem com um texto explanatório quando você deixa o mouse pousar sobre um controle ou janela.
- Ele é geralmente usado apenas para dar uma ajuda rápida sobre um determinado item.
- Para conectar um **Tool Tip** a um controle você usa o método **SetToolTip**.
- Exemplo: `ToolTip1.SetToolTip(Button1, "Isto é um botão!")`
- Para recuperar informações sobre um objeto use a propriedade **GetToolTip**.
- Você deve configurar a propriedade **Active** para **True** para o texto do **ToolTip** aparecer.
- A propriedade **AutomaticDelay** controla o tempo de exibição do mesmo.



# Timers

- São controles úteis porque permitem criar eventos periódicos.
- Eles não aparecem em uma janela em modo de execução.
- Foram criados para um ambiente single-threaded.
- O principal evento deste componente é chamado de **Tick**.
- Você controla o intervalo da geração deste evento por meio da propriedade **Interval** (em milissegundos).
- A cada ocorrência do evento **Tick** você pode adicionar um código para ele executar.
- Para habilitar/desabilitar o timer você usa a propriedade **Enabled**.
- Há também os métodos **Start** e **Stop** para fazer a mesma tarefa.

# Usando as classes HScrollBar e VScrollBar

- Conforme mencionado anteriormente há dois tipos de Scroll Bars:
  - **Horizontais (HScrollBar)**
  - **Verticais (VScrollBar)**
- A hierarquia de classes das mesmas é:

System  
  Windows  
    Forms  
      **HScrollBar**

System  
  Windows  
    Forms  
      **VScrollBar**

# Configurando os valores mínimo e máximo

- A primeira coisa que você deve fazer ao inserir uma scroll bar em seu form é alterar o possível intervalo de valores da mesma, que por padrão é 0-100.
- O valor mínimo pode ser configurando com a propriedade **Minimum**.
- O valor máximo pode ser configurando com a propriedade **Maximum**.
- Elas podem ser usadas em modo design ou em tempo de execução.

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         HScrollBar1.Minimum = 0
4         HScrollBar1.Maximum = 10
5     End Sub
6 End Class
```

# Configurando os cliques da barra

- Quando a pessoa clica na parte não preenchida da scroll bar a barra da mesma é movida.
- Para configurar o quanto essa barra deve ser movida você usa a propriedade **LargeChange**.
- Essa propriedade está intimamente ligada com o intervalo de valores.
- Ela pode ser usada em modo design ou em tempo de execução.

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         HScrollBar1.Minimum = 0
4         HScrollBar1.Maximum = 100
5         HScrollBar1.LargeChange = 10
6     End Sub
7 End Class
```

# Configurando os cliques da flecha

- Quando a pessoa clica na flecha da scroll bar a barra da mesma é movida.
- Para configurar o quanto essa barra deve ser movida você usa a propriedade **SmallChange**.
- Essa propriedade está intimamente ligada com o intervalo de valores.
- Ela pode ser usada em modo design ou em tempo de execução.

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         HScrollBar1.Minimum = 0
4         HScrollBar1.Maximum = 100
5         HScrollBar1.LargeChange = 10
6         HScrollBar1.SmallChange = 1
7     End Sub
8 End Class
```

# Configurando o valor da Scroll Bar

- Para alterar o valor padrão a ser exibido pela scroll bar é necessário usar a propriedade **Value**.
- Ela pode ser usada em modo design ou em tempo de execução.
- Ela deve possuir um valor que esteja contemplado pelas propriedades **Minimum** e **Maximum**.
- Quando a faz uma mudança na scroll bar o evento Scroll é disparado e o novo valor é colocado na propriedade **Value**.

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         HScrollBar1.Minimum = 0
4         HScrollBar1.Maximum = 100
5         HScrollBar1.LargeChange = 10
6         HScrollBar1.SmallChange = 1
7         HScrollBar1.Value = 20
8     End Sub
9 End Class
```

# Scroll Bars - Eventos

- Há dois tipos de eventos: **Scroll** (mais usado) e **ValueChanged**.
- O evento Scroll acontece quando a pessoa move a barra de rolagem por meio do mouse ou teclado.
- O novo valor que foi configurado pode ser obtido na propriedade **Value**.
- O objeto passado para o evento possui dois membros: **NewValue** e **Type**.
- O membro **Type** é uma enumeração e possui os seguintes valores:
  - **EndScroll**
  - **First**
  - **LargeDecrement**
  - **LargeIncrement**
  - **Last**
  - **SmallDecrement**
  - **SmallIncrement**
  - **ThumbPosition**
  - **ThumbTrack**

# Scroll Bars - Eventos

Exemplo:

```
1 Public Class Form1
2     Private Sub HScrollBar1_Scroll(sender As Object, e As ScrollEventArgs) Handles HScrollBar1.Scroll
3         Label1.Location = New Point(e.NewValue * CInt(Me.Size.Width / 100), Label1.Location.Y)
4         Label1.Text = "Estou me movendo!"
5     End Sub
6
7     Private Sub VScrollBar1_Scroll(sender As Object, e As ScrollEventArgs) Handles VScrollBar1.Scroll
8         Label1.Location = New Point(Label1.Location.X, e.NewValue * CInt(Me.Size.Height / 100))
9         Label1.Text = "Estou me movendo!"
10    End Sub
11
12    Private Sub HScrollBar2_Scroll(sender As Object, e As ScrollEventArgs) Handles HScrollBar2.Scroll
13        TextBox1.Text = "Posição de Scroll: " & e.NewValue
14    End Sub
15 End Class
```

# Mostrando e Escondendo Scroll Bars

- De um modo geral você usa a propriedade **Visible** para mostrar ou esconder uma barra de rolagem.
- Você usar como alternativa os métodos **Show** e **Hide**.
- Para desabilitar uma scroll bar use a propriedade **Enabled**.

Exemplo:

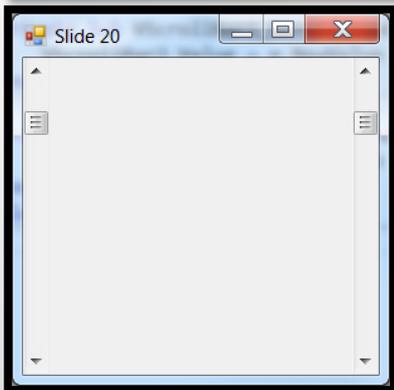
```
1 Public Class Form1
2     Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
3         HScrollBar1.Hide()
4     End Sub
5
6     Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
7         HScrollBar1.Show()
8     End Sub
9
10    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
11        HScrollBar1.Enabled = True
12    End Sub
13
14    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
15        HScrollBar1.Enabled = False
16    End Sub
17 End Class
```

# Coordenando pares de Scroll Bars

- Para trabalhar de forma sincronizada com um par de barras de rolagem é necessário usar o evento **Scroll** juntamente com a propriedade **Value**.
- Outra forma de fazer isso é usar o mesmo manipulador de eventos.

Exemplo:

```
1 Public Class Form1
2     Private Sub VScrollBar1_Scroll(sender As Object, e As ScrollEventArgs) Handles VScrollBar1.Scroll
3         VScrollBar2.Value = e.NewValue
4     End Sub
5
6     Private Sub VScrollBar2_Scroll(sender As Object, e As ScrollEventArgs) Handles VScrollBar2.Scroll
7         VScrollBar1.Value = e.NewValue
8     End Sub
9 End Class
```



# Colocando Barras de Rolagem em Imagens

Atenção!

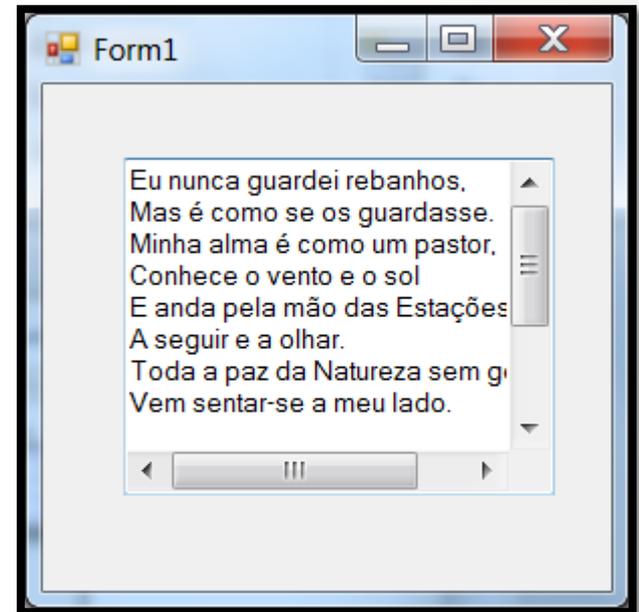
Este exemplo é longo e deve ser visto no próprio Visual Studio .NET.



# Adicionando Scroll Bars em Text Boxes

- Para colocar barras de rolagem em um campo do tipo Text Box basta usar a propriedade **ScrollBars**.
- Primeiramente assegure-se que a propriedade **MultiLine** esteja igual a **True**.
- Por fim decida qual tipo de barra de rolagem você colocará no seu campo texto. Os possíveis valores da enumeração são:
  - ✓ None (padrão)
  - ✓ Horizontal
  - ✓ Vertical
  - ✓ Both

Exemplo:



# Usando a classe Splitter

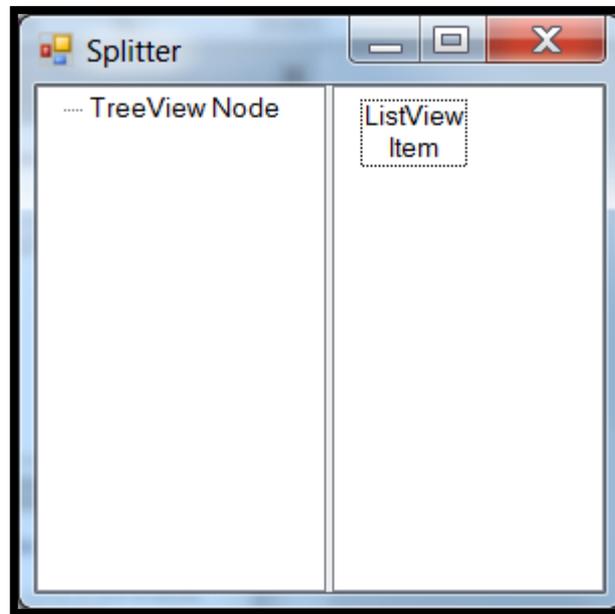
- ❑ São relativamente simples de usar.
- ❑ Controle usado para redimensionar um controle em tempo de execução.
- ❑ A hierarquia de classes do controle **Splitter** é a seguinte:

System

Windows

Forms

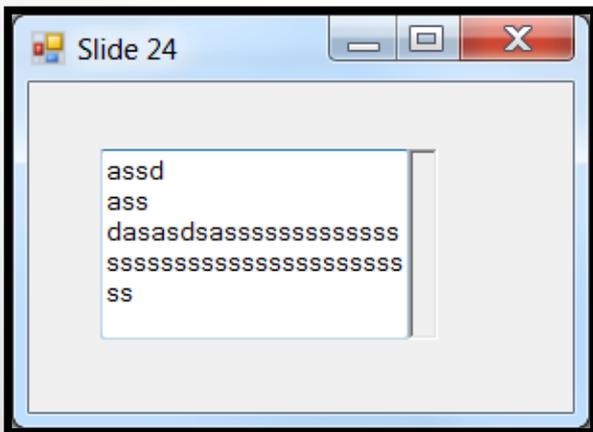
**Splitter**



# Criando um controle Splitter

- Você pode habilitar ou desabilitar este controle usando a propriedade **Enabled**.
- Para mudar a aparência do cursor use a propriedade **Cursor** (**VSplit** e **HSplit**).
- O estilo da borda também pode ser alterado usando a propriedade **BorderStyle** (None, Fixed e Fixed3D).
- Este controle é criado em modo de design.
- No exemplo deste slide foi usado um **Panel** para agrupar um text box e um controle **Splitter** para ele não usar todo o espaço do form.

Exemplo:



# Usando a classe **TrackBar**

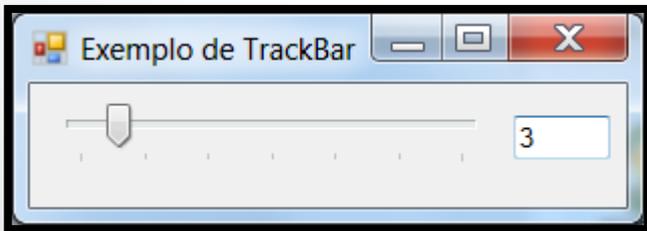
- ❑ Semelhante a uma barra de rolagem mas com jeito de controle de rádio.
- ❑ A propriedade **Value** contém o valor da configuração atual.
- ❑ Os eventos mais importantes são **Scroll** e **ValueChanged**.
- ❑ A hierarquia de classes do controle **TrackBar** é a seguinte:

System

Windows

Forms

**TrackBar**

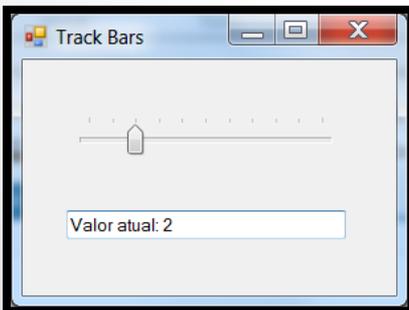


# Track Bar - Eventos

- ❑ O evento **Scroll** ocorre quando a pessoa manipula o controle na tela usando o mouse ou teclado.
- ❑ O evento **ValueChanged** ocorre quando a o valor da propriedade **Value** muda.
- ❑ No exemplo abaixo foi usado o evento **Scroll** para exibir em um text box o valor atual selecionado.

Exemplo:

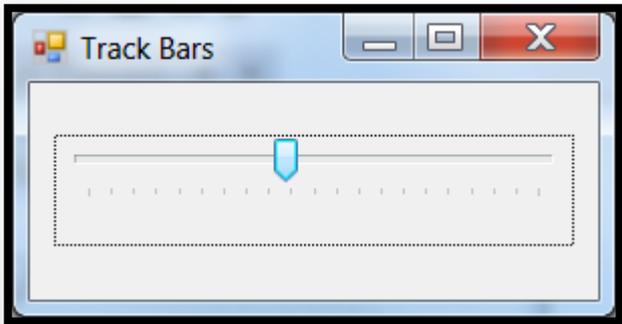
```
1 Public Class Form1
2     Private Sub TrackBar1_Scroll(sender As Object, e As EventArgs) Handles TrackBar1.Scroll
3         TextBox1.Text = "Valor atual: " & TrackBar1.Value
4     End Sub
5 End Class
```



# Track Bar - Configurando os Ticks

- ❑ **Ticks** são os pequenos traços verticais que marcam os valores de uma track bar.
- ❑ Para alterar o estilo use a propriedade **TickStyle**. Ela é uma enumeração e aceita os valores **Both**, **BottomRight**, **None** e **TopLeft**.
- ❑ A frequência é determinada pela propriedade **TickFrequency** (padrão é 1).
- ❑ No exemplo abaixo foi usado o valor **BottomRight**, **TickFrequency** igual a 5 e **Maximum** igual a 100.

Exemplo:



# Usando a classe `DateTimePicker`

- ❑ Permite que o usuário escolha uma data e hora.
- ❑ A hierarquia de classes do controle **`DateTimePicker`** é a seguinte:

System

  Windows

    Forms

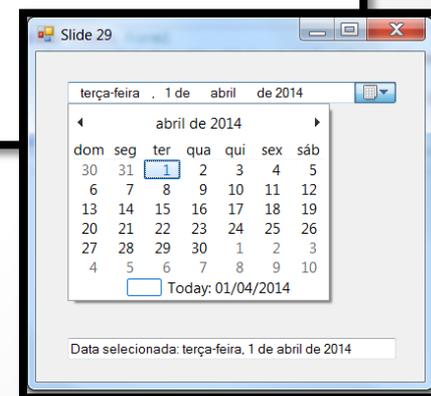
**`DateTimePicker`**

# Date-Time Picker - Eventos

- ❑ A pessoa pode editar a data/hora diretamente nos campos apresentados ou clicar na flechinha que aparece ao lado do mesmo para visualizar um calendário.
- ❑ Quando a data e/ou hora muda o evento **ValueChanged** ocorre.
- ❑ Usando a propriedade **Text** você obtém a data em um objeto **String**.
- ❑ Usando a propriedade **Value** você obtém a data em um objeto **DateTime**.

Exemplo:

```
1 Public Class Form1
2     Private Sub DateTimePicker1_ValueChanged(sender As Object, e As EventArgs) Handles
3         DateTimePicker1.ValueChanged
4         TextBox1.Text = "Data selecionada: " & DateTimePicker1.Text
5     End Sub
6 End Class
```

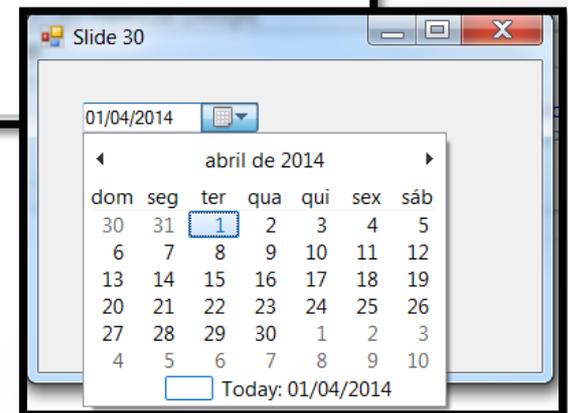


# Date-Time Picker - Formatos

- ❑ Por padrão este controle não deixa a pessoa editar a hora.
- ❑ Você pode mudar o formato de exibição data/hora com os membros da enumeração **DateTimePickerFormat**:
  - Custom
  - Long
  - Short
  - Time

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         DateTimePicker1.Format = DateTimePickerFormat.Short
4     End Sub
5 End Class
```

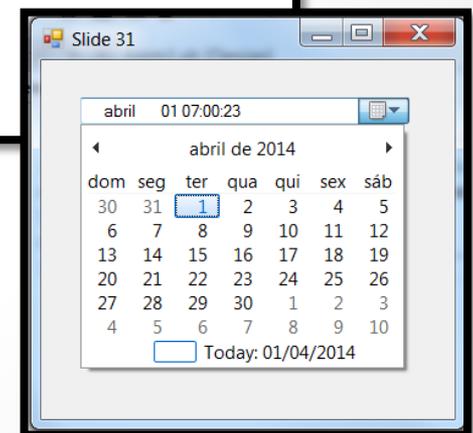


# Date-Time Picker - Customizar Formato

- ❑ Para configurar um formato não padrão (customizável) é necessário alterar o valor da propriedade **Format** para **Custom**.
- ❑ Depois é só escolher o formato desejado usando a propriedade **CustomFormat**.
- ❑ Para exibir literais na data ("/", ":") é necessário colocar eles entre ".  
Exemplo: MM'/'dd'/'yyyy hh':'mm tt.

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         DateTimePicker1.Format = DateTimePickerFormat.Custom
4         DateTimePicker1.CustomFormat = "MMM dd hh:mm:ss tt"
5     End Sub
6 End Class
```



# Usando a classe **MonthCalendar**

- ❑ Permite que o usuário selecione dias de um mês.
- ❑ A hierarquia de classes do controle **MonthCalendar** é a seguinte:

System

Windows

Forms

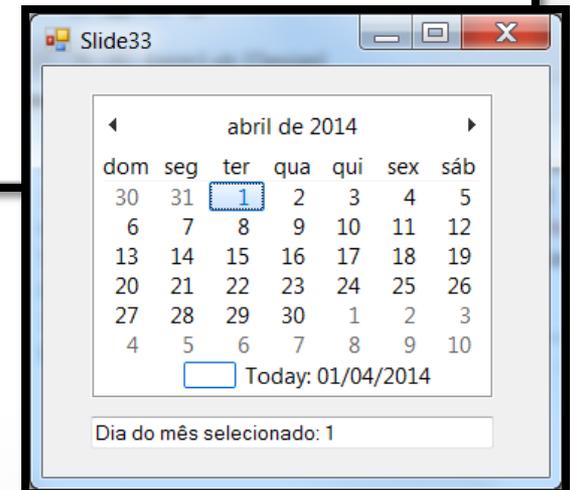
**MonthCalendar**

# MonthCalendar - Eventos

- ❑ O evento **DateChanged** ocorre quando a data no controle muda.
- ❑ O evento **DateSelected** ocorre quando uma nova data é escolhida.
- ❑ Você pode selecionar várias datas por meio das propriedades **SelectionStart**, **SelectionEnd**, **SelectionRange.Start** e **SelectionRange.End**. Elas retornam um objeto do tipo **DateTime**.

Exemplo:

```
1 Public Class Form1
2     Private Sub MonthCalendar1_DateChanged(sender As Object, e As DateRangeEventArgs) Handles _
3         MonthCalendar1.DateChanged
4         TextBox1.Text = "Dia do mês selecionado: " & _
5             MonthCalendar1.SelectionRange.Start.Day
6     End Sub
7 End Class
```



# Usando a classe NotifyIcon

- ❑ Mostram ícones na bandeja do sistema Windows.
- ❑ As propriedades mais interessantes são: **ContextMenu**, **Icon**, **Text** e **Visible**.
- ❑ Os eventos dignos de nota dessa classe são: **Click**, **DoubleClick**, **MouseDown**, **MouseMove** e **MouseUp**.
- ❑ A hierarquia de classes do controle **NotifyIcon** é a seguinte:

System

Windows

Forms

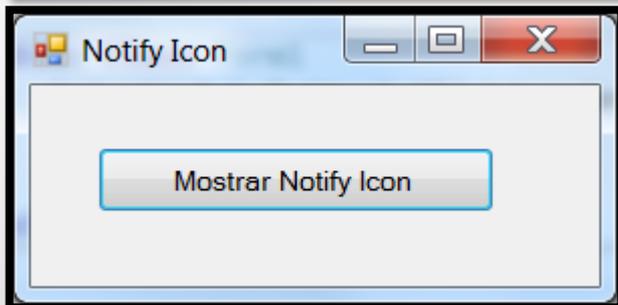
**NotifyIcon**

# Criando um Notify Icon

- ❑ Adicione um componente do tipo **NotifyIcon** ao seu projeto.
- ❑ Ele aparecerá na bandeja de controles do Visual Studio .NET.
- ❑ Altere a propriedade **Icon** com uma imagem com a extensão **.ico**.
- ❑ Altere também propriedade **Text** com o texto a ser mostrado no estilo tool tip.
- ❑ Neste exemplo eu coloquei a propriedade **Visible** igual a **False** para que o ícone seja mostrado por um botão.

Exemplo:

```
1 Public Class Form1
2     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3         NotifyIcon1.Visible = True
4     End Sub
5 End Class
```

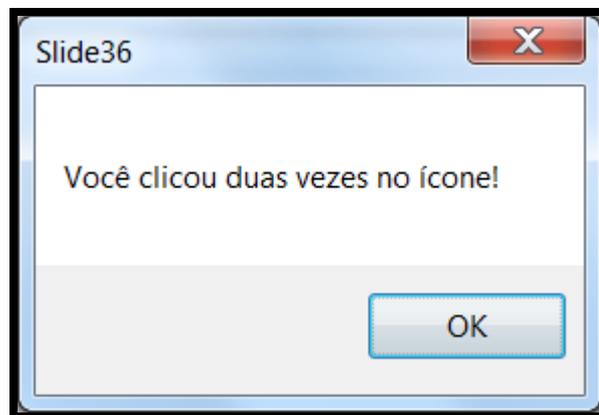
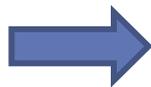
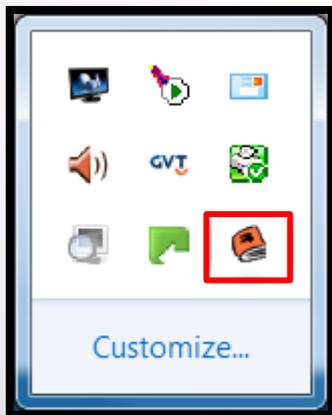


# Notify Icon - Eventos

- ❑ Os eventos mais usados para esta classe são: **Click** e **DoubleClick**.
- ❑ No exemplo abaixo foi usado o evento **DoubleClick**.

Exemplo:

```
1 Public Class Form1
2     Private Sub NotifyIcon1_DoubleClick(sender As Object, e As EventArgs) Handles _
3         NotifyIcon1.DoubleClick
4         MsgBox("Você clicou duas vezes no ícone!")
5     End Sub
6 End Class
```



# Usando a classe **ToolTip**

- ❑ Janelas que explicam com um texto algum item de um form.
- ❑ As propriedades mais interessantes são: **Active**, **AutomaticDelay**, **InitialDelay** e **ShowAlways**.
- ❑ Os eventos dignos de nota dessa classe são: **GetToolTip** e **SetToolTip**.
- ❑ A hierarquia de classes do controle **ToolTip** é a seguinte:

System

Windows

Forms

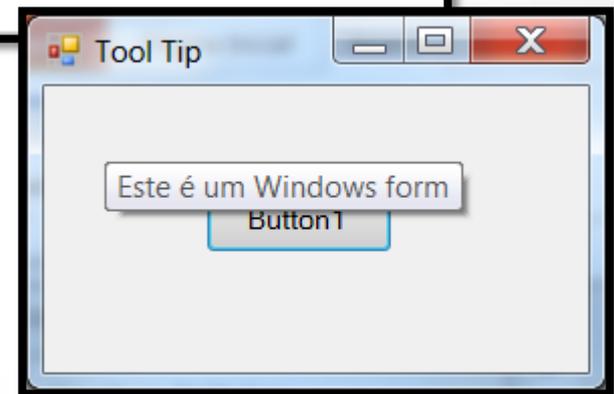
**ToolTip**

# Criando um Tool Tip

- ❑ São componentes não controles.
- ❑ Ele aparecerá na bandeja de componentes do Visual Studio .NET do seu projeto.
- ❑ Você pode associar ele a um controle usando o método **SetToolTip**.

Exemplo:

```
1 Public Class Form1
2     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         ToolTip1.SetToolTip(Me, "Este é um Windows form")
4         ToolTip1.SetToolTip(Button1, "Este é um botão")
5     End Sub
6 End Class
```



# Usando a classe **Timer**

- ❑ Causam eventos **Ticks** periódicos que você pode executar algum código em intervalos específicos.
- ❑ As propriedades mais interessantes são: **Enabled** e **Interval**.
- ❑ Os eventos mais usados dessa classe são: **Start** e **Stop**.
- ❑ O evento digno de nota dessa classe é chamado de **Tick**.
- ❑ A hierarquia de classes do controle **Timer** é a seguinte:

System

  Windows

    Forms

**Timer**

# Usando a classe Timer

## Configurando o intervalo de tempo

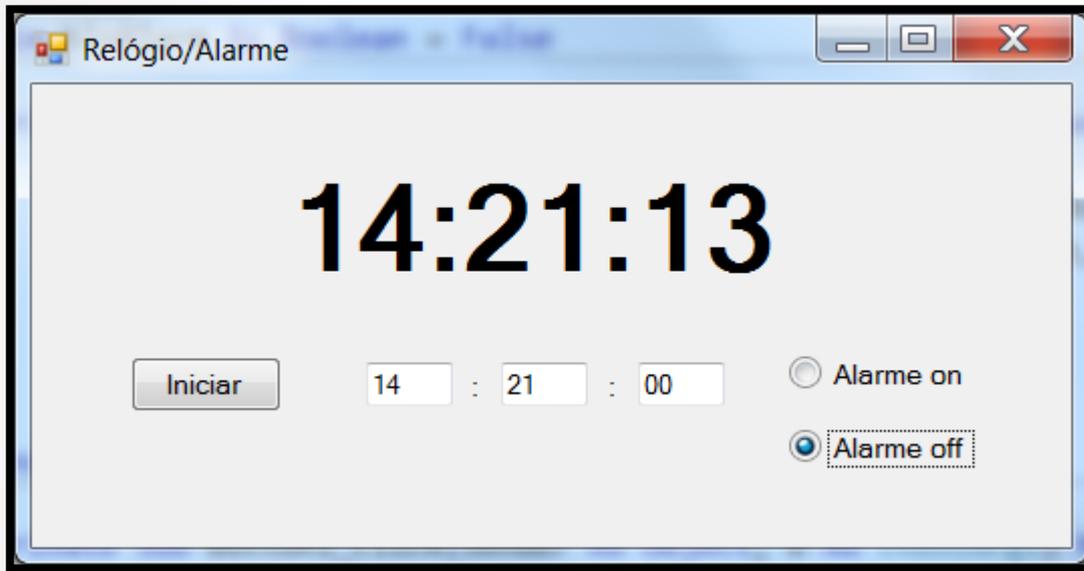
- Configurar o intervalo de um Timer é simples.
- Ele é o tempo entre os eventos **Tick**.
- Basta alterar a propriedade **Interval**. Ela é mensurada em **milissegundos**.

## Ligando e Desligando um Timer

- Use a propriedade **Enabled** para ligar (o evento **Tick** irá ocorrer então) e desligar um controle **Timer**.
- Você pode usar também os métodos **Start** e **Stop** para realizar a mesma tarefa.

# Criando um Relógio com Alarme

A interface do nosso relógio com alarme é a seguinte:



Ela possui:

- 3 Labels
- 3 Text Boxes
- 1 Button
- 2 Radio Buttons
- 1 Timer

# Criando um Relógio com Alarme

O código completo é o que segue:

```
1 Public Class Form1
2     Dim blnAlarm As Boolean = False
3
4     Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
5         Label1.Text = CStr(TimeOfDay)
6         If TextBox1.Text <> "" And TextBox2.Text <> "" And TextBox3.Text <> "" Then
7             Dim AlarmTime = New DateTime(Today.Year, Today.Month, Today.Day, CInt(TextBox1.Text), CInt(TextBox2.Text), CInt(TextBox3.Text))
8             If Now > AlarmTime And blnAlarm Then
9                 Beep()
10            End If
11        End If
12    End Sub
13
14    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
15        Timer1.Enabled = True 'Começa o evento Tick
16    End Sub
17
18    Private Sub RadioButton1_CheckedChanged(sender As Object, e As EventArgs) Handles RadioButton1.CheckedChanged
19        If RadioButton1.Checked Then
20            blnAlarm = True
21        End If
22    End Sub
23
24    Private Sub RadioButton2_CheckedChanged(sender As Object, e As EventArgs) Handles RadioButton2.CheckedChanged
25        If RadioButton2.Checked Then
26            blnAlarm = False
27        End If
28    End Sub
29 End Class
```

# Referências Bibliográficas

- HOLZNER, Steven. **Visual basic.NET: black book**. Arizona: Coriolis Group Books, 2002. xxxviii, 1144 p ISBN 1-57610-835-X.