

Padrão de Codificação

**Microsoft Visual Studio
VB.NET**

Microsoft®



Índice

Controle de Versão	3
Introdução.....	4
Comentários	5
Formulários, Módulos e Classes	5
Funções e Métodos	5
Variáveis	Error! Bookmark not defined.
Stored Procedures	6
Regions	6
Nomenclaturas	7
Nome de Soluções e Projetos	7
Nome dos Controles	7
Nome de Formulário, Módulos e Classes.	9
Nome de Variáveis.	10
Nome de Constantes.	11
Nome de Classes.	11
Nome de Métodos e Funções.	11
Considerações Importantes.....	12
Dicas	13

Controle de Versão

Versão	Descrição	Responsável	Data
1.0	Criação do Padrão De codificação	Michel César Leme Banagouro Bruno Gustavo Carvalho Capel	

Introdução

Esta proposta visa à adoção de convenções para a codificação de nomes de variáveis e de objetos e utilização de ferramentas de autodocumentação do Visual Studio .NET com o objetivo de auxiliar a manutenção de programas que muitas vezes é feita por pessoas diferentes daquelas que começaram o desenvolvimento. A convenção para nomes de variáveis e objetos ajuda na utilização das ferramentas de autodocumentação do código.

Vale também lembrar, que as convenções abaixo citadas são orientadas por padrões de mercado bem como por fornecedores da plataforma de desenvolvimento e foram customizadas para as necessidades internas da equipe de desenvolvimento, gerando o presente documento.

Comentários

O comentário do código durante seu desenvolvimento é essencial para manutenções futura e até para a documentação do projeto, pois muitas vezes a própria pessoa que desenvolveu determinada rotina tem dificuldades para entendê-la após um certo período de tempo. Para resolver este problema, deve-se adotar um padrão para os comentários. Abaixo veremos o padrão de comentários que devem ser utilizados em VB.NET.

Formulários, Módulos e Classes

Nome: Contém o nome da classe, formulário, módulo, funções ou métodos.

Objetivo: Descreve a sua finalidade.

Exemplo:

```
''' <summary>
''' ***** <BR/>
''' Nome.....: Usuario. <BR/>
''' Objetivo.....: Classe responsável por manipular e acessar aos dados <BR/>
''' dos usuários. <BR/>
''' ***** <BR/>
''' </summary>
Public Class Usuario
```

Funções e Métodos

```
''' <summary>
''' Função que retorna as informações de um determinado usuário por meio
''' do número da matrícula.
''' </summary>
''' <param name="matricula">Número da matrícula</param>
''' <returns>Retorna um Objeto usuário com seus dados carregados</returns>
''' <remarks></remarks>
Public Shared Function RecuperaUsuario(ByVal matricula As Integer) As Usuario
```

```
''' <summary>
''' Método para redefinir a senha do usuário.
''' </summary>
''' <param name="matricula">Número da matrícula</param>
''' <param name="senhaAtual">Senha atual do usuário</param>
''' <param name="senhaNova">Nova senha informada pelo usuário</param>
''' <remarks></remarks>
Public Sub AtualizarSenha(ByVal matricula As Integer, _
                        ByVal senhaAtual As String, _
                        ByVal senhaNova As String)
```

Stored Procedures

```
|-- =====
-- Nome.....: proc_UsuarioInserir
-- Objetivo.....: Cadastra um Usuario na tabela Usuarios
-- =====
DROP PROCEDURE [dbo].[proc UsuarioInserir]
```

Regions

A diretiva Region será utilizada para separar métodos, eventos, propriedades e classes.

Padrão de Region para Atributos, Propriedades, Método e Eventos.

```
#Region "Atributos"
' Código para o atributos
#End Region

#Region "Propriedades"
' Código para as propriedades
#End Region

#Region "Construtor"
' Código para os construtores
#End Region

#Region "Métodos"
' Código para os métodos e funções
#End Region

#Region "Eventos"
' Código para os eventos
#End Region
```

Nomenclaturas

Para a padronização de nomes de variáveis é indicado utilizar o estilo Camel case, onde a primeira letra do nome deve ser minúscula e a primeira letra de cada palavra subsequente deve ser maiúscula.

Nome de Soluções e Projetos

<nome do cliente(empresa)>.<nome do projeto>.<modulo>

Ou

System.<nome do projeto>.<modulo>

Ex:

UTFPR.SGTCC.Interface

UTFPR.SGTCC.MóduloAP

Nome dos Controles

O nome da variável deve ser claro o suficiente para indicar o papel da mesma dentro do programa e possuirá prefixos que determinarão o controle a que estão associadas conforme lista abaixo, separado por WebForm Prefix e WinForm Prefix. Quando for necessário usar abreviaturas sugere-se eliminar primeiro as vogais.

Padronização para nomes de controle de WinForms:

Controle	Prefixo
Form	frm
Label	lbl
LinkLabel	lnk
Button	btn
TextBox	txt
Menu	mnu
CheckBox	chk
RadioButton	rad
GroupBox	grp
PictureBox	pic
Panel	pnl
DataGrid	grd
ListBox	lst
CheckedListBox	clb
ComboBox	cbo
ListView	lvw
TreeView	tre
TabControl	tbc
DateTimePicker	dtp
MonthCalendar	mcl
HScrollBar	hsb
VScrollBar	vsb
Timer	tmr
Splitter	spt
DomainUpDown	upd

Controle	Prefixo
NumericUpDown	nud
TrackBar	trb
ProgressBar	prb
RichTextBox	rtb
ImageList	ils
HelpProvider	hlp
ToolTip	tip
ContextMenu	cmn
ToolBar	tlb
StatusBar	sta
NotifyIcon	nti
OpenFileDialog	ofd
SaveFileDialog	sfd
FontDialog	ftd
ColorDialog	cld
PrintDialog	ptd
PrintPreviewDialog	ppd
PrintPreviewControl	ppc
ErrorProvider	err
PrintDocument	prn
PageSetupDialog	psd
CrystalReportViewer	rpt
DataSet	ds
DataView	dtv

Padronização para nomes de controle de WebForms:

- HTML Controls

Controle	Prefixo
Label	lbl
Button	btn
ResetButton	rst
SubmitButton	sbt
TextField	txt
TextArea	txa
FileField	fle
PasswordField	pws
CheckBox	chk
RadioButton	rad
Hidden	hdn
Table	tbl
FlowLayoutPanel	flp

GridLayoutPanel	glp
Image	img
ListBox	lst
DropDown	ddl
HorizontalRule	htr

▪ Server Control

Controle	Prefixo
Label	lbl
TextBox	txt
Button	btn
LinkButton	lkb
ImageButton	imb
HyperLink	hlk
DropDownList	ddl
ListBox	lst
DataGrid	dgr
DataList	dls
Repeater	rep
CheckBox	chk
CheckBoxList	chl
RadioButtonList	rdl
RadioButton	rad
Image	img
Panel	pnl
Placeholder	plh
Calendar	cld
AdRotator	art
Table	tbl
RequiredFieldValidator	rfv
CompareValidator	cpv
RangeValidator	rgv
RegularExpressionValidator	ver
CustomValidator	cmv
ValidationSummary	vts
Xml	xml
Literal	ltl
CrystalReportViewer	crv

Nome de Formulário, Módulos.

Para a padronização do nome dos formulários, módulos e classes foram considerados a sigla do sistema, nome da aplicação e função do objeto.

Objeto	Nomeação	Exemplo
MDI Form	mdi<nome_do_mid>	mdiControle.frm
Form (não modal,	frm<nome_do_formulário>	frmMoedas.frm

abertura maximizada, child ou não)		
Form (modal, abertura com tamanho pré-definido)	Dlg<nome_do_formulário>	dlgLogon.frm

Nome de Variáveis.

Para a padronização de nomes de variáveis foram considerados o tipo de dado que será armazenado, o escopo e o propósito geral da variável.

Os nomes de variáveis seguem o seguinte modelo:

<Tipo de Dado> <Nome Base>

Onde:

<Tipo de Dado> é indicado em letras minúsculas.

<Nome Base> com letras maiúsculas e contém um nome significativo para a variável.

Exemplos:

strMensagem	Variável Mensagem do tipo String
intLinha	Variável Linha do tipo Inteiro
blnPrimeiraLinha	Variável PrimeiraLinha do tipo Booleano

O *<Nome Base>* deve ser claro o suficiente para indicar o papel da variável dentro do programa. Para melhor exprimir esse papel, se desejado, poderão ser usadas duas ou mais palavras, concatenadas, sempre com as iniciais maiúsculas. Quando for necessário usar abreviaturas sugere-se eliminar primeiro as vogais.

Exemplos:

intContadorDias	`Contador de Dias
strMensagemErro	`Mensagem de Erro

É interessante também tentar relacionar variáveis que se referem a uma mesma entidade, fazendo com que o nome da variável tenha sempre como palavra inicial o nome da entidade a que se refere.

Exemplos:

```
Dim intClienteCodigo
Dim strClienteNome
Dim strClienteEndereco
```

Obs.: Utilize sempre o nome da entidade como palavra inicial, como nos exemplos acima. Nomes do tipo intCodigoCliente, strNomeCliente, strEnderecoCliente dificultam o agrupamento e podem não deixar evidente a entidade em questão.

Nome de Constantes.

Para declaração de constantes sugere-se que o nome seja declarado sempre em letras maiúsculas e com underscore ("_") como caracter separador de palavras, se necessário.

Exemplo: Const MAX_LIN = 100

Nome de Classes.

Identificador	Exemplo
Classe	Public Class Usuario
Exception	Public Class AppException Nota: Sempre termina com o sufixo Exception
Interface	Public Interface IRepresentante Nota: Sempre inicia com o prefixo I
Abstratas	Public MustInherit Pessoa
Structures	Public Structure Usuario

Considerações Importantes

Algumas linguagens possuem comandos que quando declarados, gerenciam a qualidade do código produzido. Adotamos como padrão a utilização dos seguintes comandos.

Option Explicit – Sua utilização é obrigatória, e sua função é garantir que todas as variáveis referenciadas no código foram declaradas.

Dicas

Esta seção tem o objetivo de apresentar boas práticas de programação e dicas importantes para facilitar o desenvolvimento das aplicações assim como suas futuras manutenções.

Agrupamento de Variáveis

O Visual Basic, implicitamente converte variáveis para os tipos aos quais seus valores estão sendo atribuídos. Por este motivo, recomenda-se a utilização do agrupador & ao invés do sinal positivo (+). Na ocorrência do agrupamento de variáveis de tipos diferentes, problemas podem ocorrer quando da utilização do sinal positivo (+), como podemos ver no exemplo abaixo:

```
vntVar1 = "10.01"
vntVar2 = 11
vntResult = vntVar1 + vntVar2 'vntResult = 21.01
vntResult = vntVar1 & vntVar2 'vntResult = 10.0111
```

Criando Strings para MessageBox, InputBox e SQL Queries.

Quando criamos uma string muito longa, usamos o underscore ("_") para a criar múltiplas linhas de código proporcionando uma fácil leitura e manutenção. Esta técnica é particularmente usada quando necessitamos apresentar uma mensagem muito longa em controles do tipo MsgBox, InputBox e quando criamos queries SQL. Abaixo temos um exemplo desta utilização:

```
Dim Msg As String
Msg = "This is a paragraph that will be " _
& "in a message box. The text is" _
& " broken into several lines of code" _
& " in the source code, making it easier" _
& " for the programmer to read and debug."
MsgBox Msg

Dim QRY As String
QRY = "SELECT *" _
& " FROM Titles" _
& " WHERE [Year Published] > 1988"
TitlesQry.SQL = QRY
```