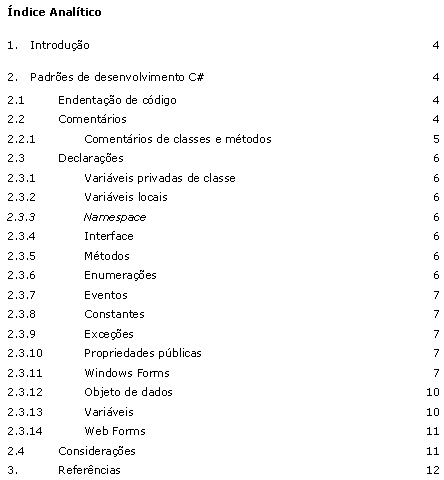
**C#.NET**

Olá pessoal, meu nome é Mauricio Júnior e estou disposto a colocar para vocês um documento de padrão para codificação de sistemas e códigos. Dessa forma, se todos programarem de uma mesma forma dentro de uma empresa ou sistema, o código será entendido por qualquer pessoa no futuro que for executar alguma modificação no sistema. Esse tipo de documento é ótimo para começar um sistema por uma equipe, tudo se torna mais fácil e ágil programar.

 Espero que possa ajudar a todos como me ajudou depois que fiz.



**1. Introdução**

Este documento tem como objetivo elaborar uma padronização para o desenvolvimento em C#, considerando os seguintes tópicos:

• Padronização de nomenclatura usada na programação de métodos, objetos, variáveis e constantes. Além disso, trata como deverá ser feita a documentação dentro do código.

**2. Padrões de Desenvolvimento C#**

**2.1 Endentação de código**

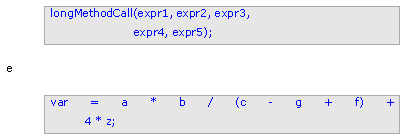
Alguns trechos de código podem ficar muito extensos e você deverá encarar o dilema de quebrar a linha.

• Quebrar a linha após uma vírgula;

• Quebrar a linha após um operador;

• Alinhar a nova linha no inicio da expressão no mesmo nível da linha anterior.

Exemplo:

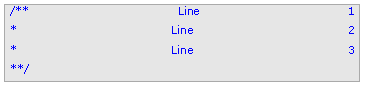


Outro fator importante é a utilização de espaços em branco para endentação. Não use espaços em branco para endentação, use tabulação. Motivos: Facilidade de incrementar e decrementar blocos de código através de atalhos de teclas do editor de código.

**2.2 Comentários**

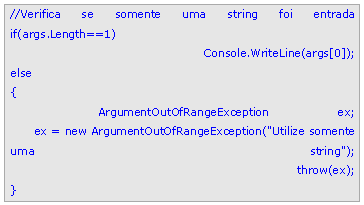
Utilizar as três barras “///” para comentários de classes e métodos. O motivo é utilizar a funcionalidade do *Visual Studio .NET* de transformar comentários em documentação de código. Entretanto, para comentários que não necessitam ser publicados, seguem algumas sugestões:

O comentário é importante para você ou outra pessoa ser orientada sobre a manutenção de um código fonte, tenha atenção à forma de destacar o comentário. Por exemplo, comentários com mais de uma linha poderiam ser assim:



Para comentários de uma linha somente, o comentário deve ser uma espécie de marcador de *loops* ou não deve ser aplicado. A questão é que como exposto em linhas anteriores, os comentários devem chamar a atenção visando facilitar e direcionar a manutenção. Somente justifica-se um comentário de uma linha quando você necessita marcar dentro de um bloco de código o início de um nível de endentação ou *loop*.

Exemplo:

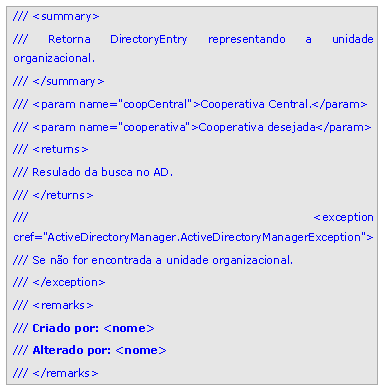


Outra boa aplicação para comentários de uma linha é a explicação de uma declaração. Por exemplo:

http://www.juliobattisti.org/tutoriais/mauricioborges/padraocodificacao001_clip_image007.gif

**2.2.1 Comentários de classes e métodos**

Toda classe e método devem ser documentados. O padrão utilizado segue abaixo:



**2.3 Declarações**

**2.3.1 Variáveis privadas de classe**

Utilizar a definição *CamelCase* (a primeira letra do identificador é minúscula e a primeira letra de cada identificador subseqüente concatenado é maiúscula).

Exemplo:

http://www.juliobattisti.org/tutoriais/mauricioborges/padraocodificacao001_clip_image009.gif

**2.3.2 Variáveis locais**

Utilizar a definição *CamelCase*.

**2.3.3 Namespace**

Utiliza-se o nome da empresa seguido pelo nome do projeto, camada de negocio e o modulo que está sendo desenvolvido.

Exemplo:

http://www.juliobattisti.org/tutoriais/mauricioborges/padraocodificacao001_clip_image010.gif

**2.3.4 Interface**

O nome de interface deve ser *PascalCase* (a primeira letra do identificador é maiúscula e a primeira letra de cada identificador subseqüente concatenado é maiúscula), e começar com o prefixo “I”, para indicar que o tipo é uma interface.

Exemplo:

http://www.juliobattisti.org/tutoriais/mauricioborges/padraocodificacao001_clip_image011.gif

**2.3.5 Métodos**

Utilizar a definição *PascalCase*.

**2.3.6 Enumerações**

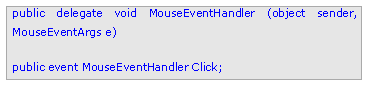
Utilizar a definição *PascalCase*.

**2.3.7 Eventos**

Utilizar a definição *PascalCase* para o nome do evento.

Utilizar o sufixo *EventHandler* para o nome do *delegate* associado ao evento. Especificar dois parâmetros “*sender*” e “e”. O parâmetro “*sender*” representa o objeto que disparou o evento e deve ser sempre do tipo *Object*, mesmo sendo possível utilizar um tipo especifico. O estado associado com o evento é encapsulado em uma instancia de um evento de classe chamado “e”.

Exemplos:



**2.3.8 Constantes**

Nomes de constantes deverão ser todos maiúsculos com as palavras separadas por um *underscore*.

Exemplo:

http://www.juliobattisti.org/tutoriais/mauricioborges/padraocodificacao001_clip_image013.gif

**2.3.9 Exceções**

O nome de uma classe de exceção deve utilizar a definição *PascalCase,  e* finalizar com a expressão *Exception*, para indicar que o tipo é uma exceção.

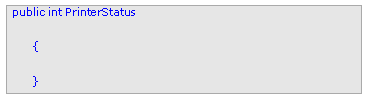
Exemplo:

http://www.juliobattisti.org/tutoriais/mauricioborges/padraocodificacao001_clip_image014.gif

**2.3.10 Propriedades públicas**

Utilizar a definição *PascalCase* para o nome de propriedades.

Exemplo:



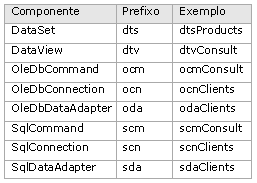
**2.3.11 Windows Forms**

Abaixo temos a lista de componentes e os prefixos:

|  |  |  |
| --- | --- | --- |
| Componente | Prefixo | Exemplo |
| Assembly | asm |  |
| Button | btn | btnExit |
| Char | ch | chxxx |
| CheckBox | chk | chkReadOnly |
| CheckedListBox | clb | clbOptions |
| ColorDialog | cld | cldText |
| ComboBox | cbo | cboEnglish |
| Container | ctr |  |
| ContextMenu | cmn | cmnOpen |
| CrystalReportViewer | rpt | rptSales |
| DataColumn | dcol |  |
| DataGrid | grd | grdQueryResult |
| DataGridDateTimePickerColumn | dgdtpc |  |
| DataGridTableStyle | dgts |  |
| DataGridTextBoxColumn | dgtbc |  |
| DataReader | dreader |  |
| DataRow | drow |  |
| DataTable | dtable |  |
| Dialog | dialog |  |
| DialogResult | dr |  |
| DomainUpDown | upd | updPages |
| ErrorProvider | err | errOpen |
| Exception | ex |  |
| FontDialog | ftd | ftdText |
| Form | frm | frmEntry |
| GroupBox | grp | grpActions |
| HashTable | htbl |  |
| HelpProvider | hlp | hlpOptions |
| HScrollBar | hsb | hsbMove |
| ImageList | ils | ilsAllIcons |
| Label | lbl | lblHelpMessage |
| LinkLabel | lnk | lnkEmail |
| ListBox | lst | lstPolicyCodes |
| ListView | lvw | lvwHeadings |
| MarshallByRefObject | rmt |  |
| Mainmenu | mm |  |
| MDI-Frame | frame |  |
| MDI-Sheet | sheet |  |
| Menu | mnu | mnuFileOpen |
| MenuItem | mi |  |
| MonthCalendar | mcl | mclPeriod |
| NotifyIcon | nti | ntiOpen |
| NumericUpDown | nud | nudPieces |
| OpenFileDialog | ofd | ofdImage |
| PageSetup Dialog | psd | psdReport |
| Panel | pnl | pnlGroup |
| PictureBox | pic | picIcon |
| PrintDialog | ptd | ptdText |
| PrintDocument | prn | prnText |
| PrintPreviewControl | ppc | ppcText |
| PrintPreviewDialog | ppd | ppdText |
| ProgressBar | prg | prgLoadFile |
| RadioButton | rad | radType |
| RichTextBox | rtf | rtfReport |
| SaveFileDialog | sfd | sfdImage |
| SDI-Form | form |  |
| Splitter | Spt | sptDivision |
| StatusBar | sta | staDateTime |
| TabControl | tbc | tbcOptions |
| TextBox | txt | txtLastName |
| Timer | tmr | tmrAlarm |
| ToolBar | tlb | tlbActions |
| ToolTip | tip | tipIcons |
| TrackBar | trb | trbIndex |
| TreeView | tre | treOrganization |
| UserControl | usr |  |
| VScrollBar | vsb | vsbMove |
| WindowsPrincipal | wpl |  |

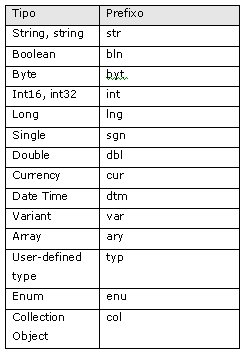
**2.3.12 Objeto de dados**

Abaixo temos alguns componentes de comunicação de dados:



**2.3.13 Variáveis**

Veja a lista abaixo de tipos e prefixos:



**2.3.14 Web Forms**

Veja a lista abaixo de tipos e prefixos:

|  |  |
| --- | --- |
| Tipo | Prefixo |
| AdRotator | adrtr |
| Button | btn |
| Calendar | cldr |
| CheckBox | cbx |
| CheckBoxList | cbxl |
| CompareValidator | cvdr |
| CrystalReportViewer | crvwr |
| DataGrid | dgrd |
| DataList | dlst |
| DropDownList | ddl |
| HyperLink | hlnk |
| Image | img |
| ImageButton | ibtn |
| Label | lbl |
| LinkButton | lbtn |
| ListBox | lbx |
| Literal | ltrl |
| Panel | pnl |
| PlaceHolder | phdr |
| RadioButton | rbtn |
| RadioButtonList | rbtnl |
| RangeValidator | rvdr |
| RegularExpressionValidator | rev |
| Repeater | rptr |
| RequiredFieldValidator | rfv |
| Table | tbl |
| TextBox | tbx |
| ValidationSummary | vsmy |
| Xml | xml |

**2.4 Considerações**

Ao declarar variáveis, procure seguir as considerações abaixo.

Ao invés de:

http://www.juliobattisti.org/tutoriais/mauricioborges/padraocodificacao001_clip_image018.gif

usar:

http://www.juliobattisti.org/tutoriais/mauricioborges/padraocodificacao001_clip_image019.gif

Sempre inicializar suas variáveis no local aonde são declaradas.

http://www.juliobattisti.org/tutoriais/mauricioborges/padraocodificacao001_clip_image020.gif

**3. Referências**

[http://www.microsoft.com/brasil/msdn/Tecnologias/visualc/  
visual\_c\_boaspraticas.aspx](http://www.juliobattisti.com.br/tutoriais/mauricioborges/1.http:/www.microsoft.com/brasil/msdn/Tecnologias/visualc/visual_c_boaspraticas.aspx)

<http://www.akadia.com/services/naming_conventions.html>

 Mauricio Junior

[mauricio@ascompras.com](mailto:mauricio@ascompras.com)

[www.ascompras.com.br](http://www.ascompras.com.br/)

w[ww.aspneti.com.br](http://www.aspneti.com.br/)