

# Visual Cobol

O futuro no presente

Copyright © 2015 do Autor



Todos os direitos reservados e protegidos pela Lei 5988 de 14/12/1973. Nenhuma parte deste livro, sem autorização prévia do autor, poderá ser reproduzida ou transmitida, sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

Altair Borges

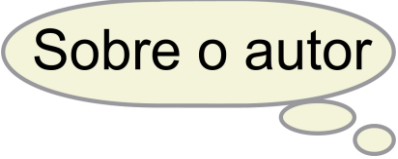
## Visual Cobol for Visual Studio

Capa: Layout: Diagramação: Editoração: Revisão: Altair Borges

1ª Edição

Belo Horizonte - Setembro/ 2015

Copyright © 2015 – Todos os direitos reservados



## Sobre o autor

Altair Borges é natural de Vazamor, distrito de Vazante, MG. Passou sua infância na fazenda, lidando com gado, andando a cavalo, nadando e pescando no córrego Renegado. Graduado em Gestão de Software pela Universidade FUMEC, iniciou sua carreira em informática em 1989 escrevendo seus primeiros programas para o compilador Microsoft Cobol 2.1. Depois para as versões MS 3.0, 4.x, 5.0 e Micro Focus COBOL 3.x, Net Express 4.0 e 5.0. Atualmente desenvolve para desktop utilizando o Visual Cobol for Visual Studio. É também autor dos livros [Saia do Vermelho](#) e [CobolWeb](#). Atualmente é Encarregado de TI no 1º Tabelionato de Protesto de Títulos da Comarca de Belo Horizonte – MG.

Dedico este livro a Edna Aurita, minha esposa, uma pessoa muito especial na minha vida e aos meus filhos, Rafael, Kaio e Gustavo.

Dedico também aos meus pais, Romão e Iva que se encontram no plano espiritual.



## *A fome*

O estômago reclama  
A criança implora  
O pai se desespera  
A lágrima não demora.

A alegria está ausente  
No lixo, jogada fora  
E o choro continua  
No dia a dia, mundo afora.

Altair Borges

## O COBOL faz parte da história

Como entusiasta e defensor da linguagem COBOL que para muitos é uma linguagem “morta”, mas que continua viva e movimentando a economia mundial. Escrever este livro seria uma forma de contribuir com aqueles que acreditam na **C**ommon **B**usiness **O**riented **L**anguage como ferramenta de desenvolvimento utilizando o ambiente .NET e ao mesmo tempo, comprovar que não é uma tecnologia ultrapassada e muito menos em extinção. Talvez o conceito em relação a ela é que precisa ser mudado. Meu primeiro contato com o COBOL foi na década de oitenta, mais precisamente em 1989 com o compilador da Microsoft COBOL 2.1. Durante este período vi o surgimento de várias linguagens até então tidas como a linguagem do futuro e que simplesmente desapareceram. O COBOL continua vivo e totalmente compatível com os novos recursos tecnológicos existentes no momento. Aqueles que pensam o contrário, não conhecem a história desta quinquagenária linguagem de programação e sua importância para o mundo dos negócios, no passado, presente e futuro.

## Sumário

<i>Sobre o livro</i> .....	12
<i>Sobre o Visual Cobol Micro Focus</i> .....	13
<b>Capítulo 1</b> .....	<b>14</b>
<i>Introdução</i> .....	14
Já imaginou sua vida sem o Cobol?.....	15
O futuro com o Visual Cobol.....	16
Especificações técnicas .....	17
Ambiente utilizado na criação das soluções.....	18
Convenções utilizadas .....	18
Obtendo o Visual Cobol .....	18
Familiarizando com o IDE do Visual Studio.....	19
Código gerenciado.....	20
Conceito de solução .....	21
Exemplos que acompanham o Visual Cobol.....	21
Estrutura de pastas .....	22
Primeiro projeto.....	22
Conhecendo a área de trabalho .....	23
Configurando o projeto .....	29
Inserindo controles no formulário.....	30
Propriedades .....	31
Text.....	32
Visible .....	32
BackColor .....	32
Font.....	33
Eventos .....	34
<b>Capítulo 2</b> .....	<b>37</b>
<i>Edição</i> .....	37
Recursos de edição e navegação .....	38
Quick Find .....	39
Quick Browser.....	40
Code Definition Window .....	40
Navigate To.....	41
Find All References .....	42
Separador .....	42
Bookmarks.....	43
Copiar e colar bloco.....	44
Personalizando as cores do IDE.....	45
Snippets Code.....	46
Criando Code Snippet .....	47
IntelliSense .....	49
Lista de membros .....	49
Informações de parâmetro .....	50
Informações rápidas .....	50
Corrigindo erros durante a codificação.....	50
Debug.....	52
Breakpoint.....	53

Breakpoint conditional.....	53
Add Watch.....	54
TaskList.....	54
<b>Capítulo 3.....</b>	<b>56</b>
<i>Data Type</i> .....	56
Data Type Reference .....	57
Declarando Data Type .....	58
Operadores.....	58
Declarando e utilizando variáveis .....	59
Strings.....	59
Concatenação .....	60
SubString.....	60
ToUpper .....	60
ToLower .....	60
IndexOf.....	60
LastIndexOf.....	60
Níveis .....	61
De classe.....	61
Locais.....	61
De contexto .....	61
Arrays .....	62
Fixo.....	63
Variável.....	63
Simples.....	63
Bidimensional.....	63
Com argumento.....	64
Como coleção de objetos.....	64
<b>Capítulo 4.....</b>	<b>66</b>
<i>OOP</i> .....	66
Conceito.....	67
Classe .....	67
Objeto .....	67
Atributos .....	68
Métodos .....	68
Classe String.....	73
Instanciação .....	73
Construtores .....	74
Sobrecarga.....	75
Declaração Property.....	76
Método private .....	78
Herança.....	78
Representação de herança - UML simplificado .....	79
Projeto oopHeranca.....	79

<b>Capítulo 5.....</b>	<b>85</b>
<i>Namespaces</i> .....	85
Entendendo os namespaces.....	86
Classe MessageBox.....	87
Classe DateTime.....	91
Os namespaces mais utilizados.....	94
 <b>Capítulo 6.....</b>	 <b>95</b>
<i>Formulários</i> .....	96
Eventos .....	96
Button.....	99
Resource.....	100
Efeito Mouse-over .....	102
Label.....	103
CheckBox.....	104
ComboBox.....	105
RadioButton.....	107
TextBox.....	108
MaskedTextBox.....	112
ErrorProvider.....	115
ImageList.....	117
TreeView .....	125
NotifyIcon .....	132
DateTimePicker.....	134
ToolStrip.....	137
DataGridView.....	146
ToolTip.....	151
HelpProvider .....	153
Formulário MDI.....	156
Login .....	160
 <b>Capítulo 7.....</b>	 <b>172</b>
<i>SisContratos</i> .....	173
Criando o projeto.....	174
Tabelas descritivas dos controles.....	175
Overview sobre arquivo indexado (ISAM).....	181
Copybooks.....	182
Limitação do arquivo indexado.....	183
Diretivas.....	184
Rebuild .....	183
Variáveis de classe .....	186
Limpando os controles.....	187
Povoando o DataGridView.....	189
Tratando o file status .....	190
Substituindo o TAB pelo ENTER .....	191
Formatação dinâmica do MaskedTextBox.....	193
Validando TextBox.....	193
Vinculando imagem ao cadastro .....	194

Evento RowEnter do DataGridView.....	196
Método Compare da classe DateTime.....	198
<b>Capítulo 8.....</b>	<b>200</b>
<i>VisRel.....</i>	<i>200</i>
Introdução.....	201
Comentário e programação.....	209
Gerando copybook.....	211
Chamada via programa COBOL.....	212
Codepage.....	216
Macros.....	216
Dicas.....	217
<b>Exemplos avançados.....</b>	<b>220</b>
<b>Capítulo 9.....</b>	<b>221</b>
<i>Exemplos avançados parte 1.....</i>	<i>221</i>
SendMail.....	222
Namespaces.....	225
System.Net.....	225
System.Net.Mail.....	225
WebRequest.....	225
WebResponse.....	225
MailMessage.....	225
SmtpClient.....	225
MailAddress.....	226
Enviando e-mail.....	230
Código completo do projeto.....	232
ReadPDF.....	236
Namespace System.IO.....	236
Classe DirectoryInfo.....	236
Classe FileInfo.....	236
Objeto AcroPDF.....	238
Código completo do projeto.....	244
GetTimeServer.....	247
Gerando DLL.....	250
Configurando as dependências.....	252
Configurando a arquitetura.....	252
<b>Capítulo 10.....</b>	<b>254</b>
<i>Exemplos avançados parte 2.....</i>	<i>254</i>
Criar XML.....	255
Criptografar elementos XML.....	259
Descriptografar elementos XML.....	264
Assinar documentos XML com x509Certificate.....	267
Password usando SHA1/SHA512.....	271
Criptografar e Descriptografar arquivo.....	274

<b>Capítulo 11</b> .....	<b>277</b>
<i>Exemplos avançados parte 3</i> .....	277
Web Service.....	278
IPAddress.....	282
 <b>Capítulo 12</b> .....	 <b>284</b>
<i>Extras</i> .....	284
DemoIsamNativo.Cbl.....	285
Get e Set (TestObject) .....	287
Teclas de função.....	290
Formata TextBox.....	291
Deploy .....	294
 <b>Capítulo 13</b> .....	 <b>298</b>
<i>C# e Cobol .NET</i> .....	299
Program Structure .....	299
Comments.....	299
Data Types.....	300
Constants.....	301
Enumeration.....	301
Operators .....	301
Choices .....	302
Loops.....	303
Arrays .....	304
Functions.....	304
Strings .....	305
Exception Handling.....	306
Namespaces .....	307
Classes / Interfaces.....	307
Constructors / Destructors.....	308
Using Objects.....	308
Structs .....	309
Properties .....	310
Delegates / Events.....	310
Enumb.cbl .....	311
Choices.cbl.....	311
Loops.cbl .....	312
Arrays.cbl.....	313
Functions-a.cbl.....	313
Functions-b.cbl .....	314
Strings.cbl .....	314
Exceptions.cbl .....	315
ClasseInterfaces.cbl .....	315
ConstructorsDestructors.cbl.....	316
UsingObjects.cbl.....	316
Structs.cbl .....	317
Properties.cbl .....	318
DelegatesEvents.cbl.....	318

## Sobre o livro



Este livro é uma introdução ao Visual Cobol For Visual Studio e parte-se do pressuposto que você, caro leitor, já tenha um pré-conhecimento da linguagem de programação COBOL e deseja desenvolver aplicativos para ambiente desktop usando a plataforma .NET Framework. O COBOL evoluiu e a prova disso é o plug-in COBOL .NET da Micro Focus. Com ele, é possível usufruir de todos os recursos desta poderosa plataforma de desenvolvimento da Microsoft.

Ao ler este livro você terá conhecimento necessário para o desenvolvimento de aplicativos usando o template Windows Forms Application, através de uma gama de exemplos práticos. Seu conteúdo começa com o simples aplicativo Alô Mundo passando pela Introdução a OOP até soluções mais avançadas com a utilização de classes como System.Security.Cryptography.Xml para criptografar um elemento XML com certificado X.509, System.Security.Web para consumir um Webservice e várias outras como a System.Text para trabalhar com arquivo texto. Nele, você encontrará também soluções completas de manutenção em arquivos ISAM usando controles e recursos disponíveis no VS. Como a tecnologia caminha a passos largos, podem surgir novas implementações tanto no Visual Studio como no Visual Cobol que não serão encontradas aqui.

O capítulo 8 é dedicado ao VisRel, uma das melhores ferramentas para criação, visualização e impressão de relatórios utilizando diversas linguagens de programação como C#, DataFlex, Delphi, Visual Basic e Cobol nos dialetos: AcuCOBOL, Micro Focus 4.5, Net Express, PowerCobol e Visual Cobol. Sua IDE, o VisRelEdit, permite a utilização de recursos como marca d'água, logos, grades, fontes, cores, macros e código de barras de vários formatos como EAN 13, EAN 128, 2 de 5 intercalado.

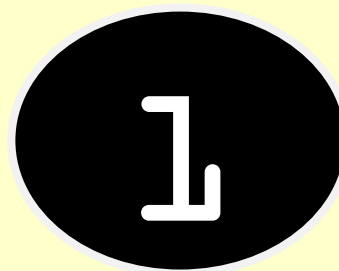
A inclusão das telas do ambiente de desenvolvimento e de cada etapa da construção das soluções apresentadas neste livro é um facilitador para o entendimento de quem busca ampliar seus conhecimentos para desenvolver aplicativos em Visual Cobol for Visual Studio.

## Sobre o Visual Cobol

O Visual Cobol possibilita a portabilidade, migração e integração do COBOL para plataformas Windows, .NET, JVM, UNIX, Linux e Cloud. Ele permite aos desenvolvedores ampliarem o COBOL a plataformas mais modernas, sem que o código tenha de ser reescrito, o que é de extrema importância na modernização das aplicações.

Sua documentação é repleta de exemplos que podem ser consultados a qualquer instante além da comunidade oficial da Micro Focus [community.microfocus.com](https://community.microfocus.com) que conta com profissionais da própria companhia para responder as dúvidas postadas pelos membros. Para fazer parte da comunidade basta se cadastrar no site. Outro canal que também pode ser consultado é o NutShell Visual Cobol no YouTube além dos webinars gravados por especialistas da Micro Focus incluindo versões em português.

# Introdução



## Tópicos

- 1.1 - Já imaginou sua vida sem o COBOL?;
- 1.2 - O futuro com o Visual Cobol;
- 1.3 - Especificações técnicas;
- 1.4 - Ambiente utilizando na criação das soluções;
- 1.5 - Convenções utilizadas;
- 1.6 - Obtendo o Visual Cobol;
- 1.7 - Familiarizando com o IDE do Visual Studio;
- 1.8 - Código gerenciado;
- 1.9 - Conceito de solução;
- 1.10 - Exemplos que acompanham o Visual Cobol;
- 1.11 - Estrutura de pastas;
- 1.12 - Primeiro projeto;
- 1.13 - Conhecendo a área de trabalho;
- 1.14 - Configurando o projeto;
- 1.15 - Inserindo controles no formulário;
- 1.16 – Propriedades;
  - ▶ 1.16.1 – Text;
  - ▶ 1.16.2 – Visible;
  - ▶ 1.16.3 – BackColor;
  - ▶ 1.16.4 – Font;
- 1.17 – Eventos;

## Já imaginou sua vida sem o COBOL?

1 • 1 - Os dados a seguir tem como base o vídeo disponibilizado pela MF no evento Developer Day realizado no dia 25 de Março de 2015 em São Paulo. Estas informações ratificam o que a grande maioria dos desenvolvedores COBOL já sabe, mas para quem esta iniciando é algo novo e que precisa ser divulgado para que se tenha a noção da dimensão e da importância desta linguagem de programação.

O que vem à sua mente quando você ouve a palavra ‘COBOL’? Tecnologia desatualizada da ‘velha guarda’? Uma linguagem em extinção? Você sabia que o COBOL impulsiona 70% de todas as transações de negócios? Mesmo que exista há décadas o COBOL tem sempre evoluído, acompanhado o ritmo do nosso mundo em evolução. Hoje, a única coisa que está desatualizada é a percepção das pessoas. Há mais de 250 bilhões de linhas de código COBOL dando suporte a aplicativos de negócios centrais. Em um único ano, existem 200 vezes mais transações no COBOL do que buscas no Google e Youtube juntas.

E então? Já pensou nessa hipótese? Como seria ter os seguintes serviços comprometidos?



Semáforos...



Programação de trens...



Controle de tráfego aéreo...



Ligações de telefones celulares, envio de mensagens ou twitter



Quase todas as comunicações seriam eliminadas.



Você não poderia comprar alimentos no supermercado, pois a maioria dos sistemas nos pontos de venda cairia.



Uma falha no COBOL pode derrubar a maioria dos sistemas bancários, dos planos de saúde e do comércio em geral.

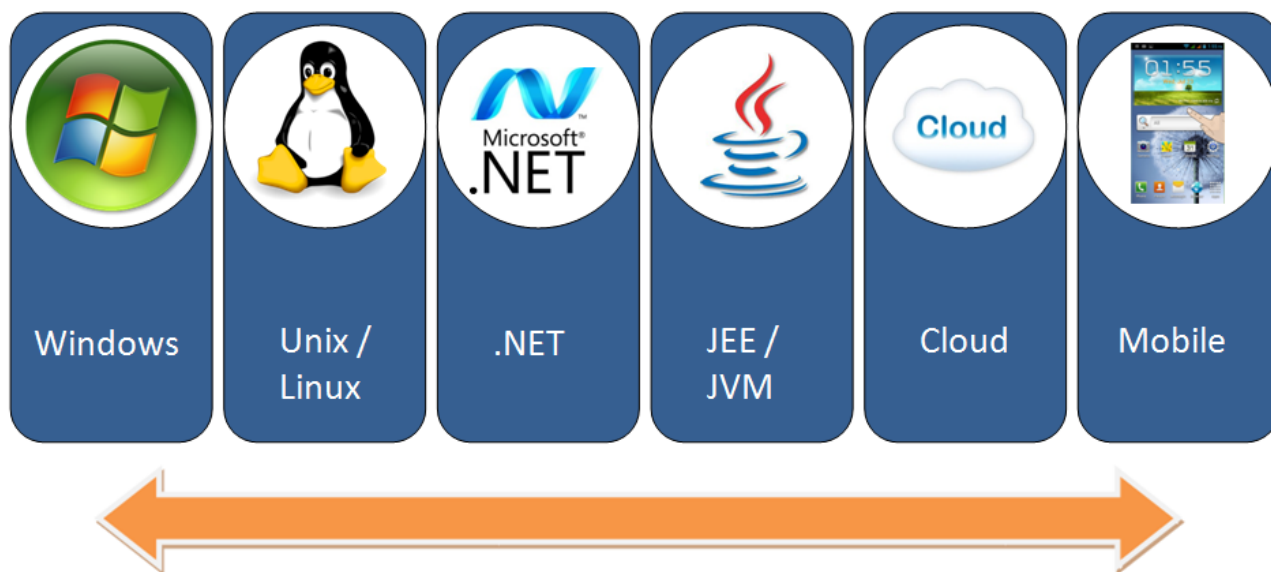


Coisas que você tinha como garantidas poderiam simplesmente desaparecer.



O COBOL está em todos os lugares e há uma redução massiva de competência nessa linguagem. A maioria dos programadores esta chegando à idade da aposentadoria e deixando o mercado de trabalho. Isto abre uma grande perspectiva, pois quem tiver o conhecimento em COBOL no seu currículo, certamente levará vantagem sobre os demais. O Visual Cobol conecta o antigo ao novo e é executado dentro de IDEs modernas como Eclipse e Visual Studio sendo esse último, o foco deste livro.

## *O Futuro com o Visual COBOL*



1.2 – A Micro Focus está comprometida em evoluir o COBOL e a aprova disso é o investimento anual de milhares de dólares para garantir o Visual Cobol como líder da indústria para o desenvolvimento de aplicações COBOL e implantação em sistemas Windows, Unix e Linux. Ele combina as melhores ferramentas de desenvolvimento de sua classe dentro do Eclipse e Visual Studio, com a flexibilidade de implantação nas diversas plataformas corporativas, incluindo .NET e Java Virtual Machine.

## 1.3 - Especificações técnicas

<p><b>SUPORTE À PLATAFORMA</b></p> <p>A Micro Focus executa milhares de casos de teste em todas as combinações de sistema operacional e processadores suportados, incluindo 32 e 64 bits, .NET e JVM, garantindo que as aplicações COBOL rodem de forma confiável, não importando onde a aplicação seja implantada.</p> <p><b>SISTEMAS OPERACIONAIS SUPORTADOS</b></p> <p><b>Windows no Intel x86/x64</b></p> <ul style="list-style-type: none"> <li>• Windows Vista</li> <li>• Windows XP</li> <li>• Windows 7</li> <li>• Windows 8</li> <li>• Windows Server 2003</li> <li>• Windows Server 2008</li> <li>• Windows Server 2012</li> </ul> <p><b>AIX</b> 6.1 / 7.1</p> <p><b>Linux em Intel x86/x64</b> SUSE 11 SP 2, SP3 Red Hat 5.9, 6.4 Oracle Linux 6.4</p> <p><b>z/Linux em 390</b> SUSE 11 SP 2, SP3 Red Hat 6.4</p> <p><b>Solaris</b> Solaris 10, 11.1 on SPARC Solaris 11.1 on Intel</p>	<p><b>HPUX</b> HPUX 11i v3 (11.31) em Itanium</p> <p><b>SERVIDORES DE APLICAÇÃO SUPORTADOS PARA COBOL JVM</b> A Micro Focus testou a implantação COBOL JVM com estas configurações conhecidas:</p> <table border="1"> <tr> <td>Tomcat 7.0.39</td> <td>JRE 1.6.027</td> <td></td> </tr> <tr> <td>JBoss 6.1</td> <td>JRE 1.6.027</td> <td></td> </tr> <tr> <td>Oracle WebLogic 12.1.1</td> <td>JRE 1.6.027</td> <td>IBM JRE 1.6.1 SR10 FP1</td> </tr> <tr> <td>IBM WebSphere 8.5</td> <td>JRE 1.6.027</td> <td>IBM JRE 1.6.1 SR10 FP1</td> </tr> </table> <p><b>SERVIDORES DE APLICAÇÃO SUPORTADOS PARA ENTERPRISE SERVER</b> A Micro Focus testou a implantação J2EE com Enterprise Server com estas configurações: WebSphere 8.0, 8.5 Oracle WebLogic Server 10.3.5, 12.1.1 JBoss Application Server 5.1, 6.1</p> <p><b>BANCOS DE DADOS SUPORTADOS</b> A Micro Focus suporta acesso a banco de dados ODBC, JDBC e ADO.NET usando OpenESQL. Configurações de banco de dados suportados incluem:</p> <ul style="list-style-type: none"> <li>• Microsoft SQL Server 2008 R2, 2012</li> <li>• IBM DB2 9.5, 9.7, 10.1</li> <li>• Oracle 11g R1, 11g R2</li> </ul>	Tomcat 7.0.39	JRE 1.6.027		JBoss 6.1	JRE 1.6.027		Oracle WebLogic 12.1.1	JRE 1.6.027	IBM JRE 1.6.1 SR10 FP1	IBM WebSphere 8.5	JRE 1.6.027	IBM JRE 1.6.1 SR10 FP1
Tomcat 7.0.39	JRE 1.6.027												
JBoss 6.1	JRE 1.6.027												
Oracle WebLogic 12.1.1	JRE 1.6.027	IBM JRE 1.6.1 SR10 FP1											
IBM WebSphere 8.5	JRE 1.6.027	IBM JRE 1.6.1 SR10 FP1											




Portal Micro Focus. Disponível em: <[http://www.microfocus.com.br/assets/visual-cobol-22-whats-new\\_tcm22-212828.pdf](http://www.microfocus.com.br/assets/visual-cobol-22-whats-new_tcm22-212828.pdf)>  
Acesso em: 05 de Fevereiro de 2015.

## 1.4 - Ambiente utilizado na criação das soluções

Os códigos apresentados neste livro foram executados no seguinte ambiente:

Sistema Operacional	Windows 7 Home Basic com Service Pack 1
Tipo de sistema	64 Bits
Memória (RAM)	8,00 GB
Processador	Intel(R) Core™ i3-4160 CPU @ 3.60 GHz 3.60 GHz
IDE	Visual Studio 2012 Shell (Integrated)
Framework	4.0 e 4.5
Compilador	Visual Cobol 2.2.00151
Gerador de relatório	VisRel 4.0 Lite

## 1.5 - Convenções utilizadas

VS	Visual Studio
VC	Visual Cobol
IDE	Integrated Development Environment (Ambiente Integrado de Desenvolvimento)
MF	Micro Focus
MS	Microsoft
Click-right	Clicar com o botão direito do mouse
Double-Click	Duplo clique com o botão esquerdo do mouse
	Dica importante
	Mensagem importante sobre determinado erro
	Erro crítico

## 1.6 - Obtendo o Visual Cobol



Para baixar o Visual Cobol é necessário cadastrar-se no site da Micro Focus para em seguida receber um e-mail disponibilizando um link temporário para o download.

Visual COBOL® 2013 Trial	<a href="https://www.microfocus.com/product-trials/visual-cobol/vcvs2013/index.aspx">https://www.microfocus.com/product-trials/visual-cobol/vcvs2013/index.aspx</a>
Visual COBOL® Personal Edition	<a href="https://www.microfocus.com/product-downloads/vcpe/vcpe22/index.aspx">https://www.microfocus.com/product-downloads/vcpe/vcpe22/index.aspx</a>

A instalação pode durar alguns minutos e após o término faça o registro on-line e comece a desfrutar desta poderosa ferramenta de desenvolvimento da Micro Focus.



## 1.7 - Familiarizando com o IDE do Visual Studio

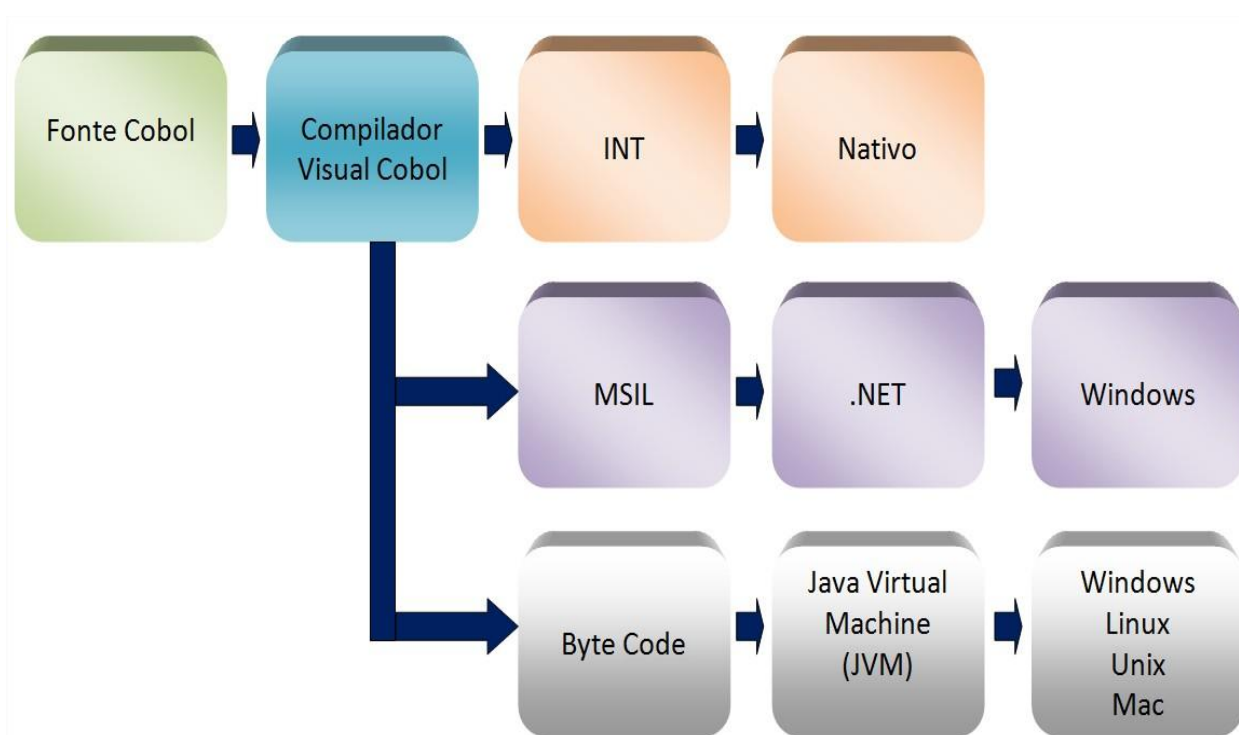
O ambiente integrado de desenvolvimento (IDE - Integrated Development Environment) do Visual Studio oferece um conjunto de ferramentas que o ajudam a escrever e modificar códigos, como ainda, detectar e corrigir erros. Com ele é possível construir aplicativos móveis, para desktop, aplicativos Web ASP.NET e serviços Web XML. Você pode usar várias linguagens de programação como VB, C, C++, C#, J# e também COBOL.

A .Net Framework é uma infraestrutura que contém os principais produtos da Microsoft e de outras empresas que desenvolvem aplicativos, não só para Windows. É uma estrutura de desenvolvimento independente de linguagem e plataforma, onde se podem desenvolver soluções corporativas de forma rápida e dinâmica. Os programas que são desenvolvidos usando essa tecnologia precisam do Framework para sua execução. Como o objetivo deste livro é auxiliar os leitores a conhecerem o Visual Cobol e como ele se integra ao Visual Studio, faz-se necessário o conhecimento de alguns de seus recursos, principalmente os de edição e depuração.

## 1.8 - Código gerenciado

CLR

Código gerenciado é um termo criado pela Microsoft. Ele se refere ao código de programa que requer o uso do Common Language Runtime (CLR) da máquina virtual para ser usado. Código não gerenciado é usado para se referir ao código de programa que não exige uma máquina virtual. O CLR, é responsável por gerenciar a execução das aplicações desenvolvidas em .NET como C#, VB, Visual Cobol etc. É ele que garante suporte unificado a todas as linguagens de programação que rodam neste Framework e é justamente por essa forma de utilização genérica que o ambiente .NET também é chamado de ambiente gerenciado. O mesmo se aplica ao JVM (Java Virtual Machine). No exemplo abaixo, temos três opções de compilação no Visual Cobol: Int, MSIL e Byte Code. O Int é nativo COBOL, o MSIL é interpretado pelo Framework e o Byte Code pelo JVM. Uma das grandes vantagens da utilização do CLR é poder tratar as exceções através da especificação de blocos Try, Catch e Finally.



## 1.9 - Conceito de solução

O Visual Studio trabalha com vários projetos inter-relacionados que juntos formam uma aplicação. O conjunto de projetos é denominado solução. Para criar uma aplicação no Visual Studio é preciso compreender três conceitos: solução, projeto e itens do projeto.

**Solução:** Todo projeto criado no Visual Studio fica dentro de uma solução e uma solução pode ter (N) projetos e arquivos relacionados integrando a aplicação.

**Projeto:** Um projeto consiste numa série de arquivos, que podem ser formulários, classes, arquivo de configuração, imagens etc. representando cada, um elemento da aplicação.

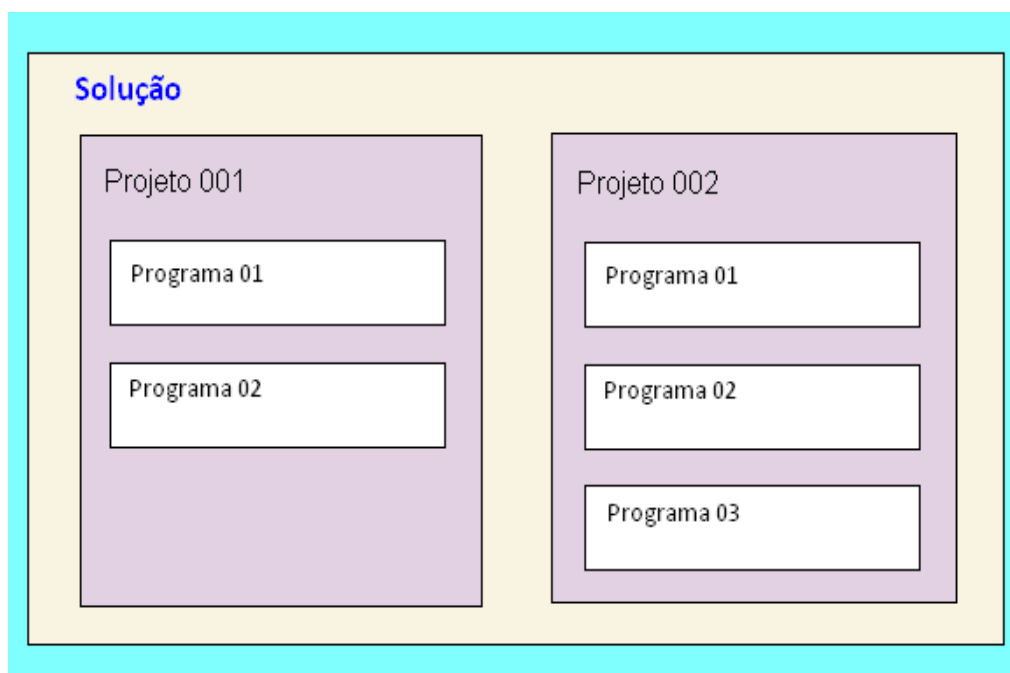


Figura 1.0 – Hierarquia dos projetos dentro da solução e dos arquivos dentro de cada projeto.

## 1.10 - Exemplos que acompanham o Visual Cobol

O Visual Cobol instala na pasta `C:\Documents and Settings>All Users>Documentos>Micro Focus> Visual COBOL>Samples` os projetos exemplos, figura 1.1. Clique em `Iniciar>Arquivos de Programas>Micro Focus Visual Cobol>Samples` para acessar a janela com todos os projetos disponíveis incluindo Win Book que mostra um Winform comunicando com um programa Cobol já existente e games como CobolDoKu e Cobol Blitz.

Na edição de dezembro/2013 do Cobol Code Contest promovido pela Micro Focus, dois brasileiros dividiram a terceira colocação. Antônio João com o game Campo Minado e Bráulio Fidellis com a Torre de Hanoi. Fonte: Portal Micro Focus. Disponível em: <http://academic.microfocus.com/code/> Acesso em 08 de Janeiro de 2015. Embora o COBOL não seja a linguagem ideal para a criação de jogos, os participantes mostraram o quanto esta linguagem é versátil.

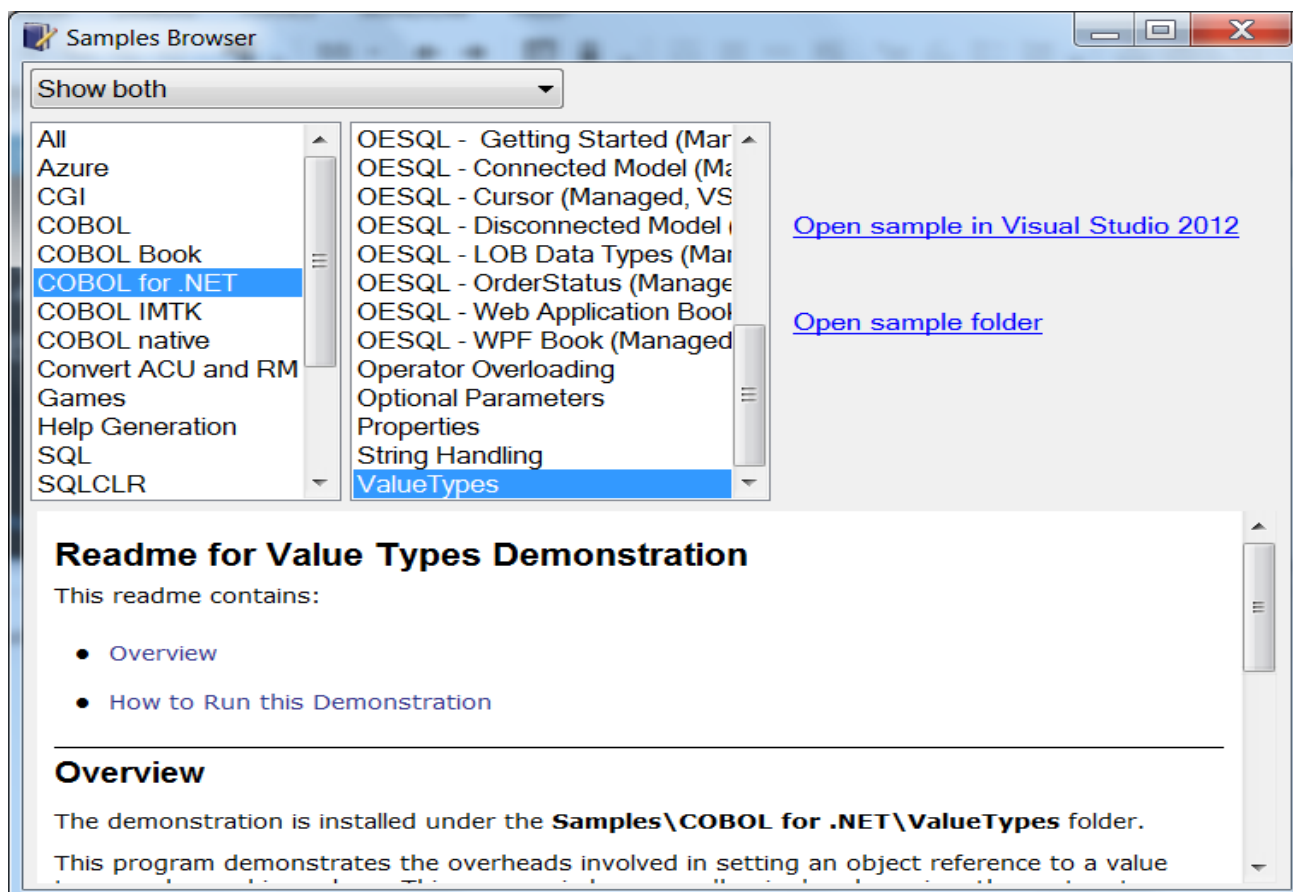


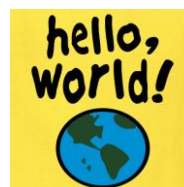
Figura 1.1 – Browser contendo os exemplos que acompanham o Visual Cobol.

## 1.11 - Estrutura de pastas



Crie a seguinte estrutura de pastas	
c:\vc2012	Soluções / Projetos
c:\vc2012\imagens	Imagens utilizadas nos formulários
c:\vc2012\dados	Bases de dados
c:\vc2012\copybooks	Bibliotecas utilizadas nos projetos
c:\vc2012\ajuda	Arquivos de ajuda .html etc.

## 1.12 - Primeiro projeto



Nosso primeiro projeto será o familiar *Alô mundo*. Ele será utilizado, apenas, para mostrar alguns dos recursos disponíveis no Visual Studio para edição e compilação de um programa COBOL. No decorrer dos próximos capítulos, serão apresentados novos recursos que lhe ajudarão a familiarizar-se com este ambiente de desenvolvimento.

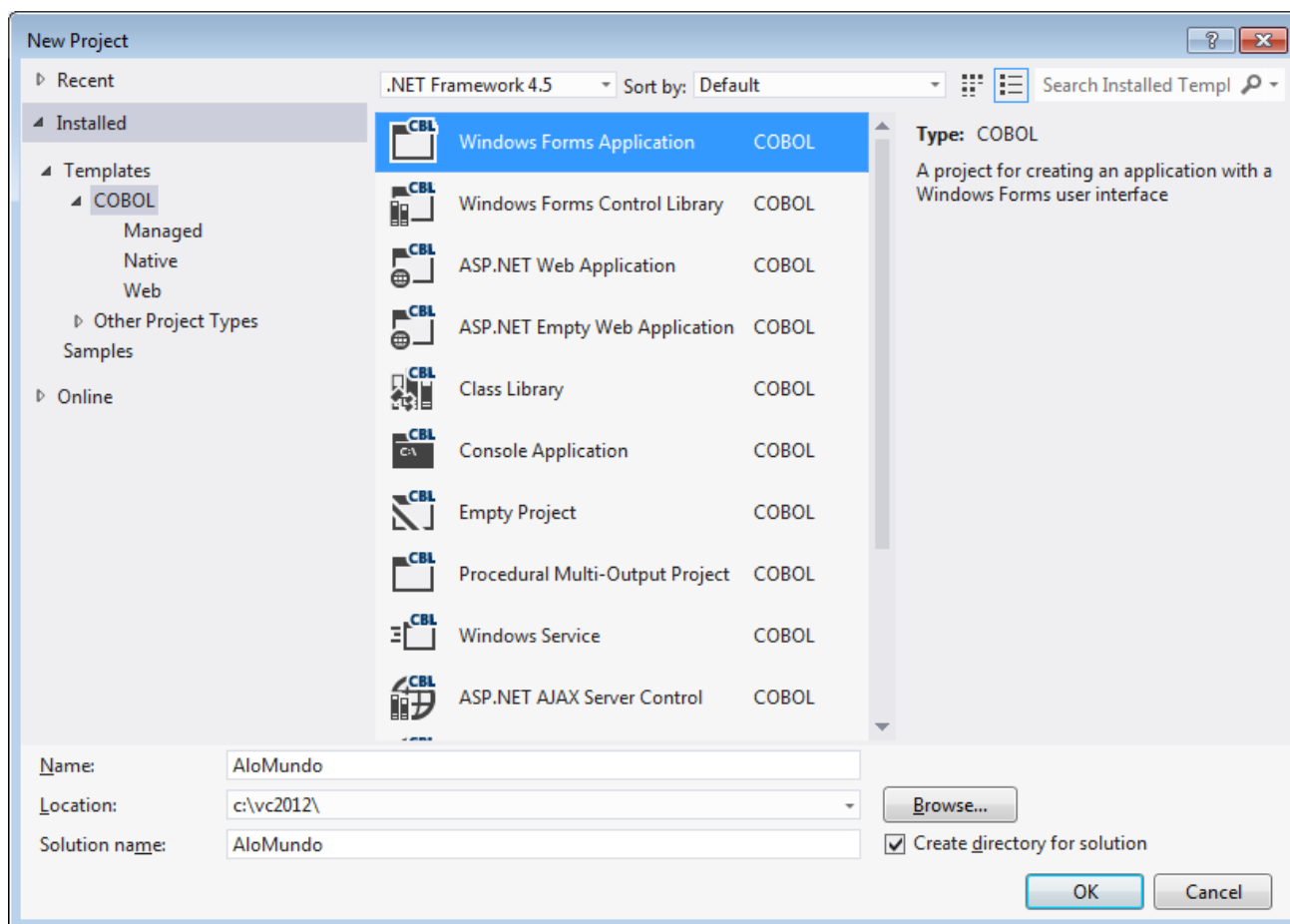




Figura 1.2 – Janela de criação de um novo projeto

Abra o Visual Studio e selecione **File>New>Project** (Ctrl + Shift + N) para visualizar a janela New Project. A caixa de diálogo disponibiliza vários tipos de projetos como ASP.NET Web Application, Windows Service, Console Application, Empty Project etc. Como o foco deste livro são as aplicações tipo Windows Forms selecione o template Windows Forms application. Altere as caixas de texto conforme figura 1.2. *Name* (nome do projeto), *Location* (Pasta onde será gravada a solução) e *Solution name* (nome da solução). Deixe marcada a opção *Create directory for solution* com isso, será criada uma subpasta dentro da pasta c:\vc2012 onde serão gravados todos os arquivos do projeto. Clique no botão OK para criar a solução e ir para o Form Designer.

### 1.13 - Conhecendo a área de trabalho



Após a criação do projeto Windows Forms, o VS disponibiliza o Form Designer, figura 1.3, clique em *VIEW* e nas opções *Solution Explorer*  (Ctrl + Alt + L) e *Toolbox*  (Ctrl + Alt + X) para que as janelas sejam visualizadas, caso ainda não estejam disponíveis.

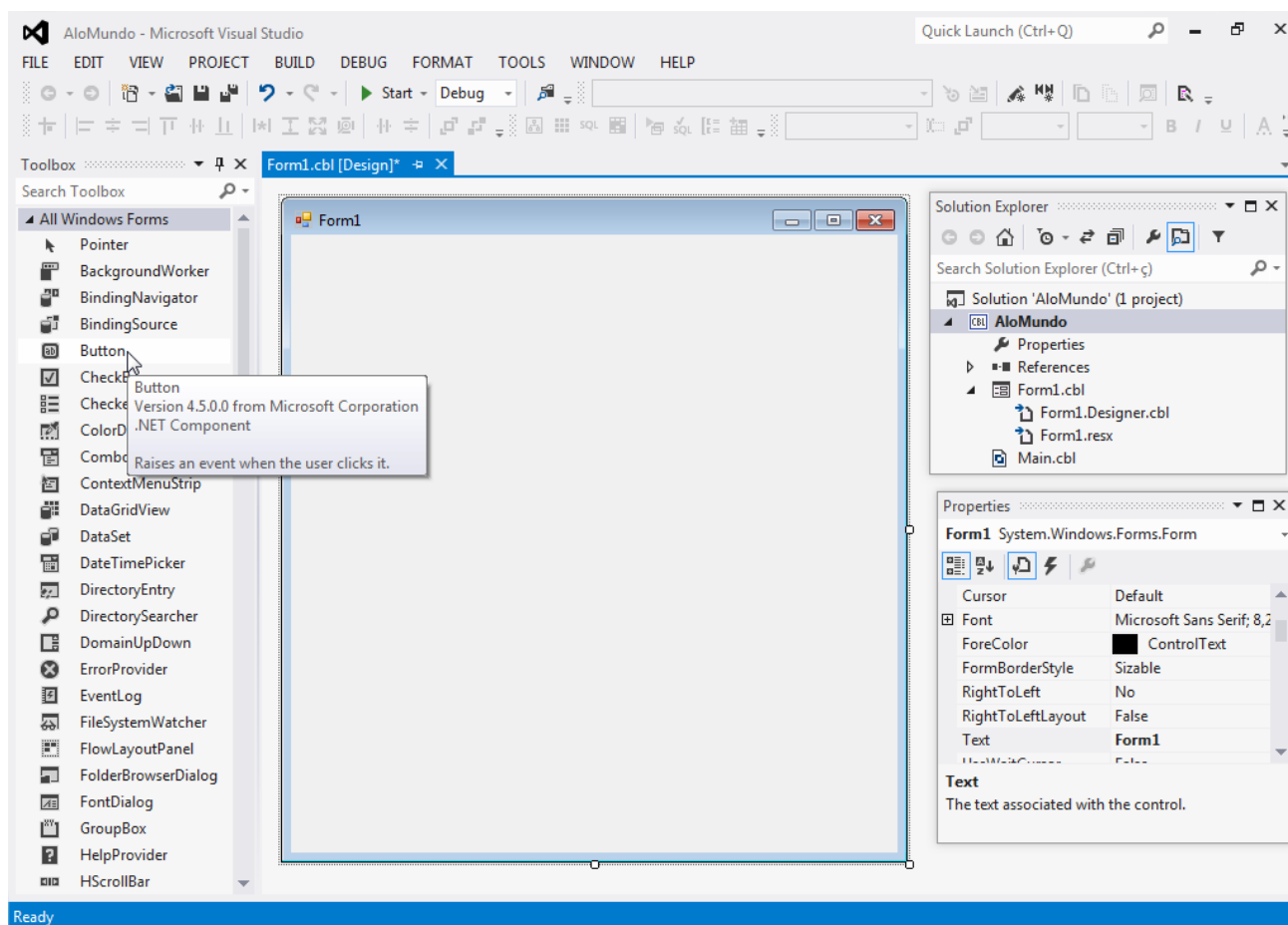


Figura 1.3 – Área de trabalho.

À esquerda, a *toolbox* com seus controles previamente disponibilizados. Ao centro, o formulário *Form1* criado automaticamente junto com o projeto e a direita, na parte superior, a janela *Solution Explorer* e na inferior, a de *propriedades*.

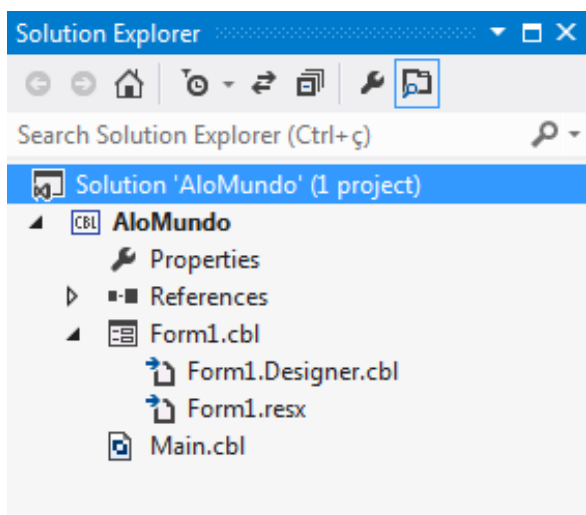


Figura 1.4 – Exemplo de um *Solution Explorer*.

O IDE disponibiliza ao programador, ter num único local, os recursos necessários para o gerenciamento dos itens do projeto através da janela *Solution Explorer*. Na figura 1.4 temos a solução *AloMundo* e apenas um projeto com o mesmo nome e que possui vários itens. Por padrão, quando um projeto do tipo *Windows Forms Application* em Cobol é criado, são adicionadas três classes **Form1.cbl**, **form1.Designer.cbl** e **Main.cbl**. A classe *Main.cbl* tem a função de instanciar o formulário principal da aplicação, neste exemplo, o *Form1*.

A classe *Form1.Designer.cbl* possui todo código necessário para a criação e customização do formulário e seus controles. Qualquer alteração nas propriedades desses controles o Visual Cobol se encarregará de atualizá-la.

As janelas Form designer e Code Editor são organizadas nos painéis com guias no quadro do IDE. O losango de guia irá auxiliá-lo no posicionamento da janela selecionada.

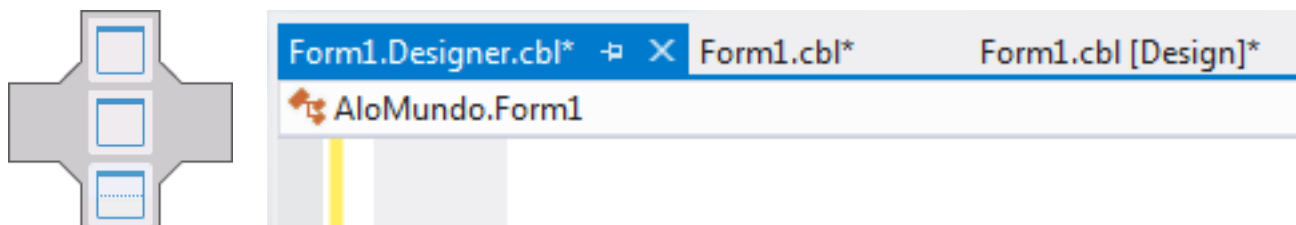


Figura 1.5 – Losango de guia para o editor e o designer do Windows.

As janelas de ferramentas como Toolbox, Solution Explorer e Properties podem ser fixadas nas extremidades do quadro na IDE ou na estrutura de edição, figura 1.5. Um diamante de guia aparece quando você arrasta uma janela de ferramentas para outro local. Clique no título da janela e arrasta-a colocando o título na guia onde indica o local desejado para fixá-la. Na figura 1.6, a janela Toolbox será fixada na parte superior do quadro. Uma faixa azul indica o local.

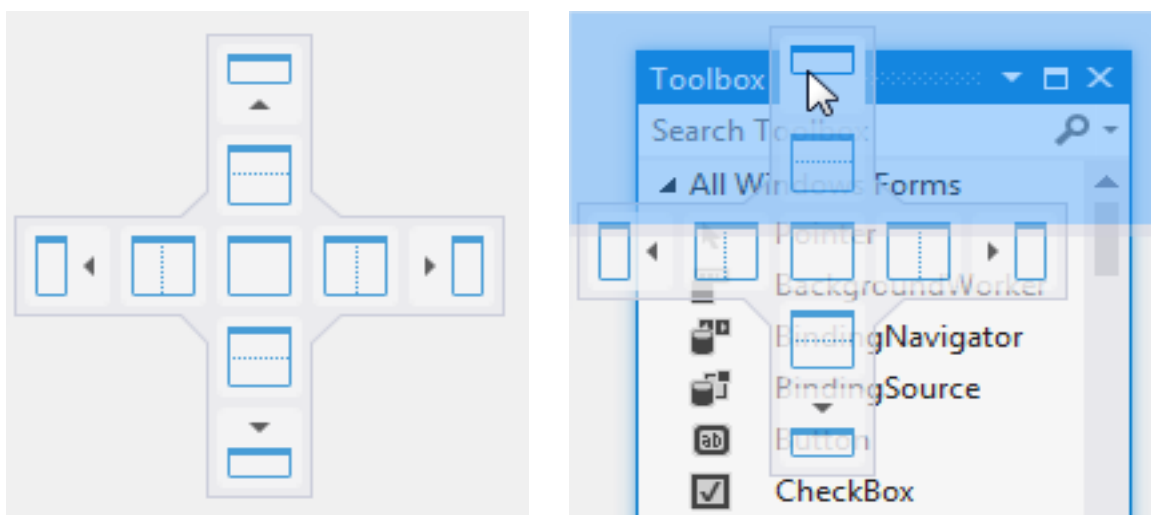



Figura 1.6 – Losango de guia para as janelas flutuantes do Windows.

Na janela de edição a opção *Split* (dividir a janela), figura 1.7, pode ser realizada de duas formas: acessando **Windows>Split** ou clicando no ícone  e arrastando-o até a posição desejada. Para desfazer clique em **Windows>Remove Split**.

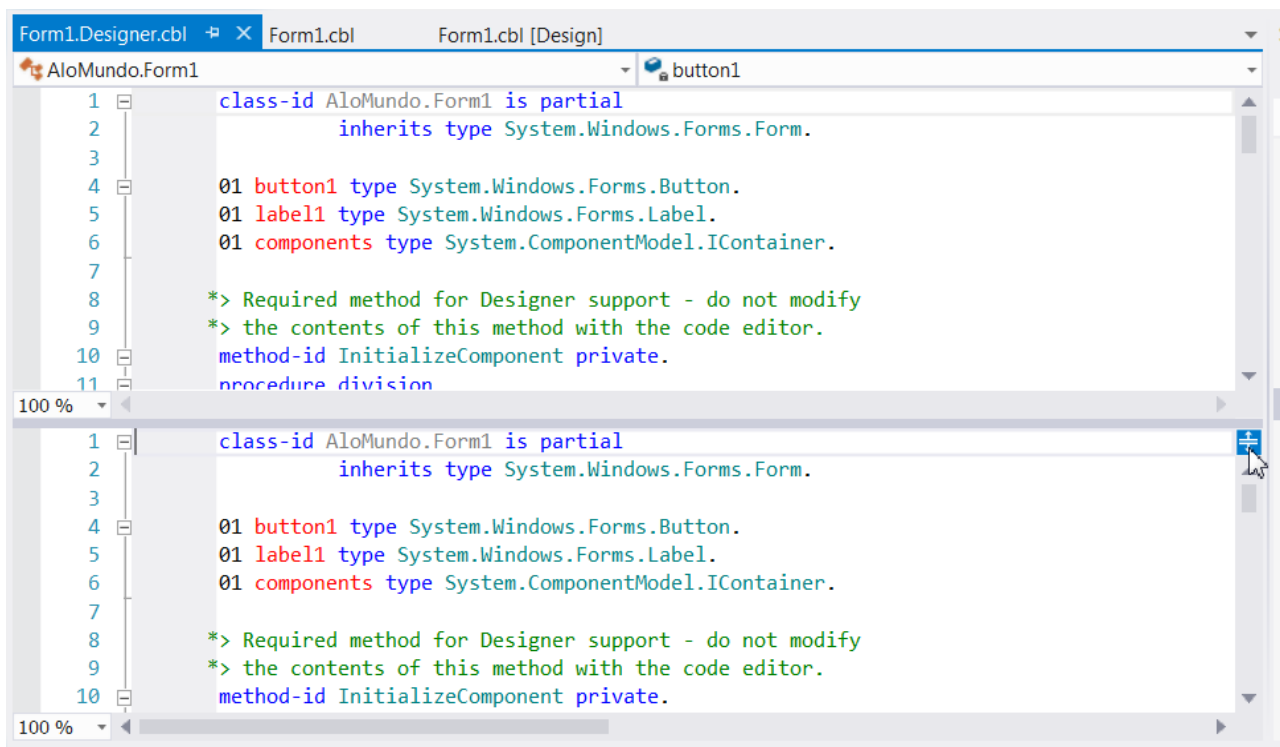


Figura 1.7 – Split.

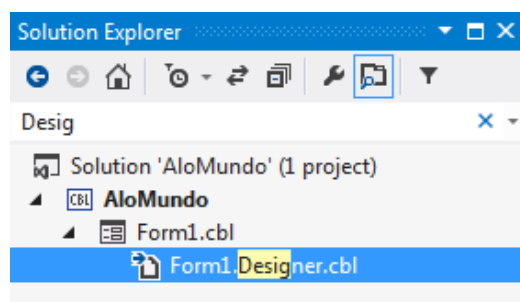
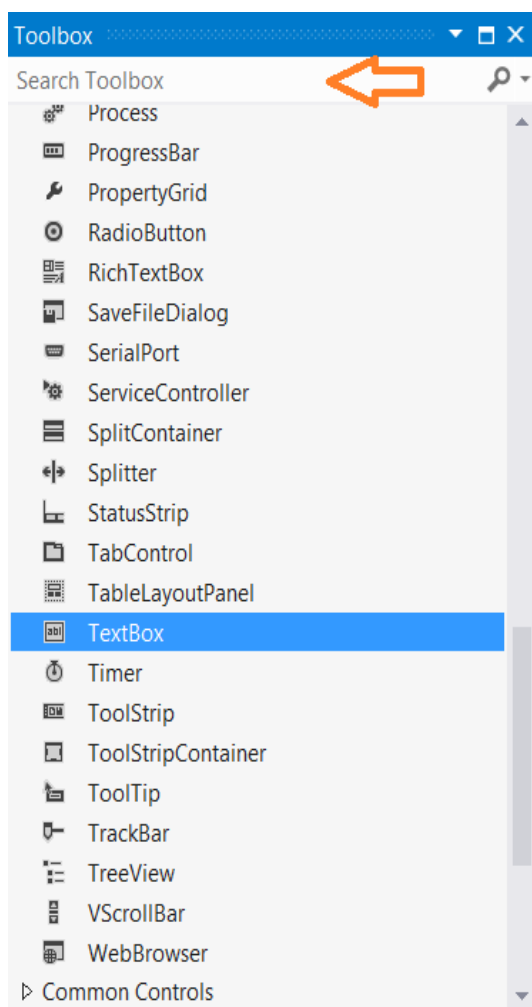


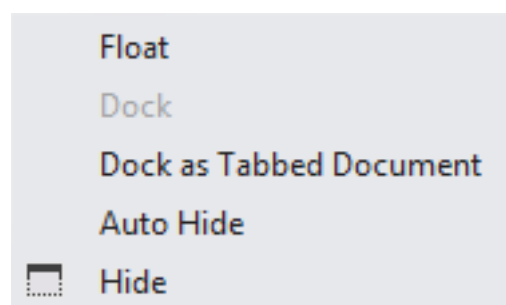
Figura 1.8 – Search.

No Visual Studio 2012 a localização de um determinado arquivo ficou mais fácil. É necessário apenas informar o nome ou parte dele na caixa de pesquisa, *Search Solution Explorer*, esperar alguns segundos e o resultado será exibido na estrutura do projeto com a parte do texto digitada em amarelo conforme figura 1.8.




A janela *Toolbox* disponibiliza acesso aos controles como *Label*, *TextBox*, *Button* etc. Quando criamos um formulário, o primeiro passo é desenhar a interface adicionando controles e definindo suas propriedades para depois inserir o código que executará as ações determinadas. Para inserir um controle é necessário apenas dar um double-click nele ou usar a técnica arrastar e soltar “drag and drop”.

Click-right no título da janela *Toolbox* para ver as opções de posicionamento dela dentro do IDE.



**Float** Torna a janela flutuante.

**Dock** Fixa a janela à esquerda e o alfinete  fixa a janela à esquerda da guia.

**Hide** Oculta a janela.

Figura 1.9 – Janela Toolbox



Use a caixa de pesquisa **Search Toolbox** no top da janela para localizar o controle desejado, figura 1.9. Caso deseje customizá-la, click-right sobre ela para visualizar a janela suspensa contendo várias opções como inserir, alterar ou excluir uma tabela.

Para adicionar outros controles na Toolbox, click-right sobre ela e selecione *Choose Items...* e você terá acesso à janela *Choose Toolbox Items* onde poderá escolher os componentes desejados. Exemplo: Adobe PDF Reader - Figura 1.10.

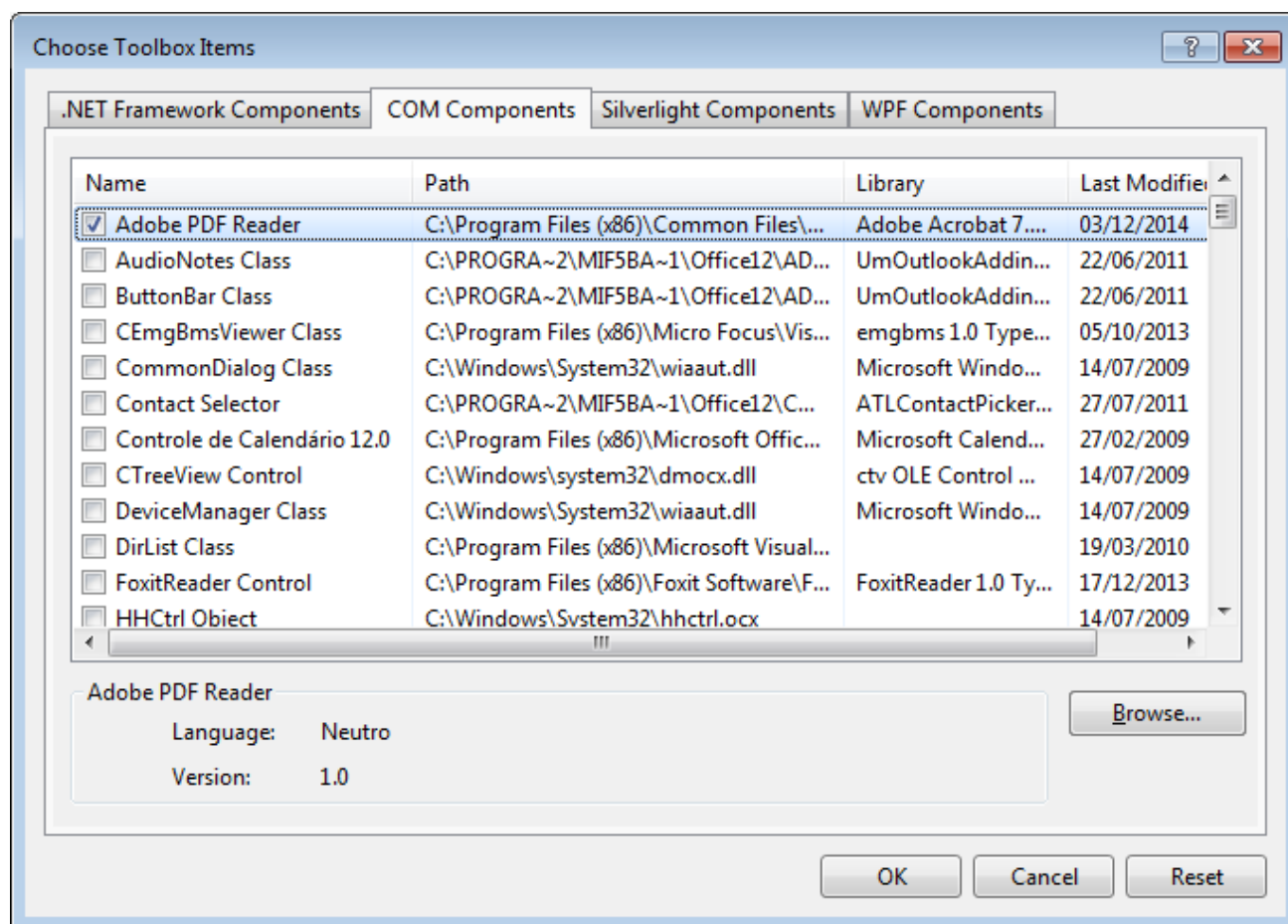


Figura 1.10 – Janela Choose Toolbox Items

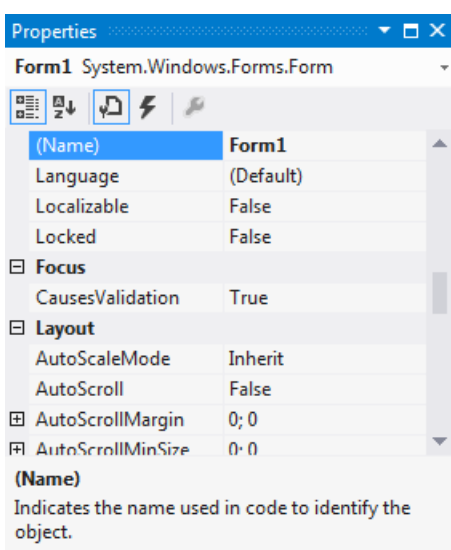



Figura 1.11 – Janela de propriedades

Para customizar o formulário e seus controles é necessário atribuir o valor desejado as suas propriedades e isto pode ser feito através da janela Windows Designer ou via código. Clique no ícone  ou selecione o controle e pressione F4. Quanto às propriedades disponíveis, elas dependem do controle selecionado como um Form, TextBox, ListBox etc. O TextBox é um dos mais utilizados e tem algumas propriedades que precisam ser destacadas.

CharacterCasing	Normal, Upper ou Lower
MaxLength	Quantidade de caracteres permitidos
MultiLine	Pode conter várias linhas de texto
PasswordChar	Caracter mostrado na edição
TextAlign	Alinhamento do texto

## 1.14 - Configurando o projeto

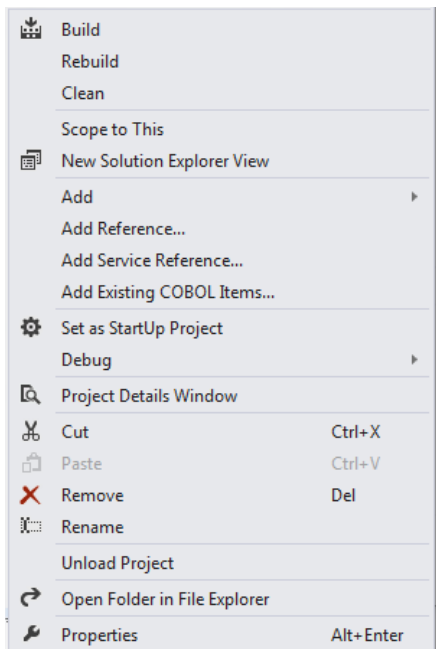
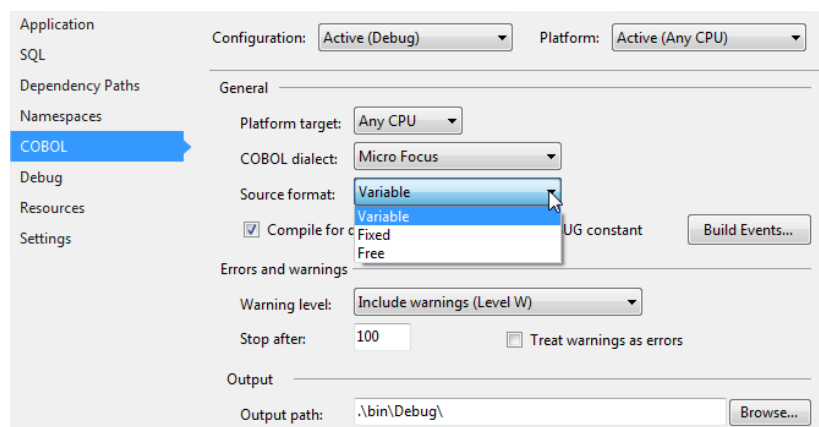



Figura 1.12 – Janela de propriedades.

Click-right sobre o nome do projeto para que a janela, figura 1.12, seja visualizada. Selecione *Properties* e o VS disponibilizará as opções de configuração do projeto como: Versão do Framework, Plataforma x86 ou x64, ícone da aplicação, dialeto do COBOL, Diretivas de compilação, formatação do programa fonte, se fixo, variável ou livre etc.



O programa COBOL na formatação fixa é dividido nas seguintes áreas:

Formatação Fixa	
Colunas 1 - 6	Branco, podendo ser preenchido com o número da linha.
Coluna 7	Se contiver um (* ou /) trata-se de um comentário. Se contiver um (-) é a continuação da linha imediatamente anterior.
Colunas 8 - 11	Margem A: Divisões, seções, parágrafos, descrição de arquivo e de dados bem como da numeração dos níveis.
Colunas 12 - 72	Margem B: Codificação dos comandos e instruções
Colunas 73 - 80	Área ignorada pelo Cobol
<p> Um programa com formatação livre, a 1ª linha deverá ter a diretiva <code>\$set sourceformat "free"</code> iniciando na coluna 7 e no máximo 256 colunas. Na formatação variável, <code>\$set sourceformat "variable"</code> o limite de colunas baixa para 250. Como é sabido, o Cobol não é case sensitive e portando não há distinção entre os nomes <i>CalculaPreco</i>, <i>calculapreco</i> e <i>CALCULAPRECO</i>. Já o ambiente .NET é, e interpretará estes três nomes de forma distinta.</p>	

## 1.15 - Inserindo controles no formulário

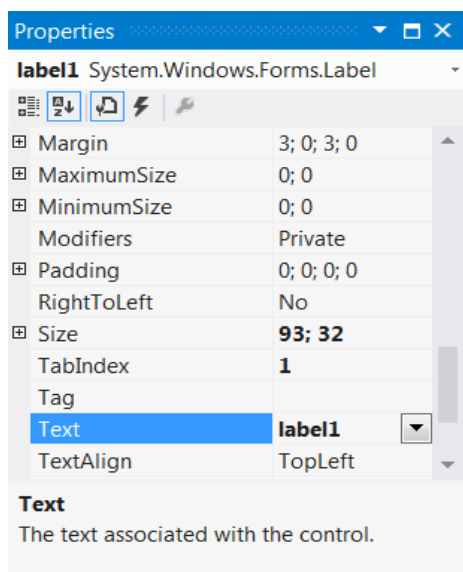
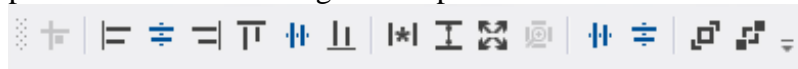




Figura 1.13 – Janela de propriedades

O IDE disponibiliza uma gama de controles que servem para diferentes fins. Mas, veremos aqui apenas os mais utilizados e suas principais funções. Na janela Toolbox dê um double-click nos controles Button e Label. Se preferir, pode usar a técnica “drag and drop” arrastar e soltar.



Utilize a barra de ferramenta layout, **View>Toolbars>Layout**, para alinhar os controles dentro do formulário ou de um container como Panel. O primeiro controle selecionado será o dominante e o alinhamento ou espaçamento o terá como referência. Pressione (Ctrl + A) para selecionar todos e depois F4 para acessar a janela *Properties*, figura 1.13. Altere as propriedades dos controles conforme tabela abaixo e depois pressione F5 ou clique no botão  para executar a aplicação.

Controle	Propriedade	Valor
Label1	Font	16
Button1	Font	16
	Text	Clique me

 **Pratique.** Customize o formulário a seu gosto e descubra as propriedades de cada controle. Clique numa área livre do Form e pressione F4 para acessar suas propriedades. Altere o título para Alô Mundo e a cor de fundo BackColor para Moccasin. Na opção BackColor é disponibilizada uma janela com três abas Custom, Web e System com as cores disponíveis. Se preferir, pode inserir uma imagem como pano de fundo. Clique em BackgroundImage para selecioná-la. Caso a imagem seja maior que o form, clique em BackgroundImageLayout e selecione Stretch. Esta propriedade ajusta a imagem ao tamanho do objeto e isto vale também para os controles PictureBox, Button etc.

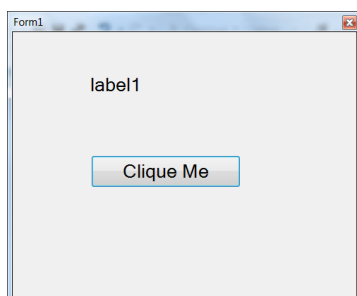



Figura 1.14 – Projeto Alô Mundo

Feche o aplicativo em execução ou na janela de edição clique no botão , pois o próximo passo será escrever o código que executará as ações determinadas. Dê um double-click no button1 (Clique Me) para ver a classe AloMundo.Form1 em modo de edição - figura 1.15. Insira o código da linha 15 e pressione F5 para executá-lo.



Para alterar as propriedades de vários controles de uma só vez, basta selecioná-los e inserir os valores desejados como cor, fonte etc.

```

1  class-id AloMundo.Form1 is partial
2      inherits type System.Windows.Forms.Form.
3
4  working-storage section.
5
6  method-id NEW.
7  procedure division.
8      invoke self::InitializeComponent
9      goback.
10 end method.
11
12 method-id button1_Click final private.
13 procedure division using by value sender as object
14                                     e as type System.EventArgs.
15     set label1::Text to "Visual Cobol .NET"
16 end method.
17 end class.

```

Figura 1.15 – Classe AloMundo.Form1.cbl em modo de edição.

## 1.16 - Propriedades

Os controles têm propriedades e elas podem ser modificadas através do Form Designer ou em tempo de execução. Quando trabalhamos com propriedades geralmente a sintaxe usada para alterá-la é:

**SET nomeDoControle::propriedade TO {valor}**

A tabela abaixo contém algumas propriedades existentes no controle Label. Para visualizar as propriedades de um determinado controle basta selecioná-lo e pressionar F4.

Propriedade	Descrição
Text	Contem o texto a ser mostrado para o usuário.
Visible	Aceita valor booleano Verdadeiro ou Falso (True or False).
BackColor	Cor de fundo do controle.
ForeColor	Cor do texto do controle.
Font	Tipo de fonte, Tamanho.

### 1.16.1 - Text

Set `label1::Text` to “nome do cliente”

### 1.16.2 - Visible

Esta propriedade serve para mostrar e/ou ocultar o controle dentro do formulário.

```
method-id Form1_Load final private.
procedure division using by value sender as object e as type System.EventArgs.
    set label1::Visible to false *> Oculta label1
    set label2::Visible to false *> Oculta label2
    set label3::Visible to True *> Mostra label3
end method.
```

Para inserir um comentário, utilize a barra pra direita ( / ) na coluna sete ou ( \*> ) em qualquer posição da linha.

### 1.16.3 - BackColor

Ela é usada para atribuir a cor de fundo ao controle. O Visual Cobol acessa diretamente as classes nativas do .NET. Ao inserir a palavra chave **type** você terá acesso as classes e propriedades através do *IntelliSense* que veremos mais adiante.



Quando definimos uma cor ou qualquer outra propriedade para um determinado controle através da janela de propriedades, o código COBOL é gerado automaticamente na classe {NomeDoForm}.Designer.cbl na janela Solution Explorer. Abra-a e utilize seu código como fonte de pesquisa. Ela inicializa e configura os Formulários e seus controles. Ela é invocada pelo método construtor **NEW** da classe Form1.cbl.

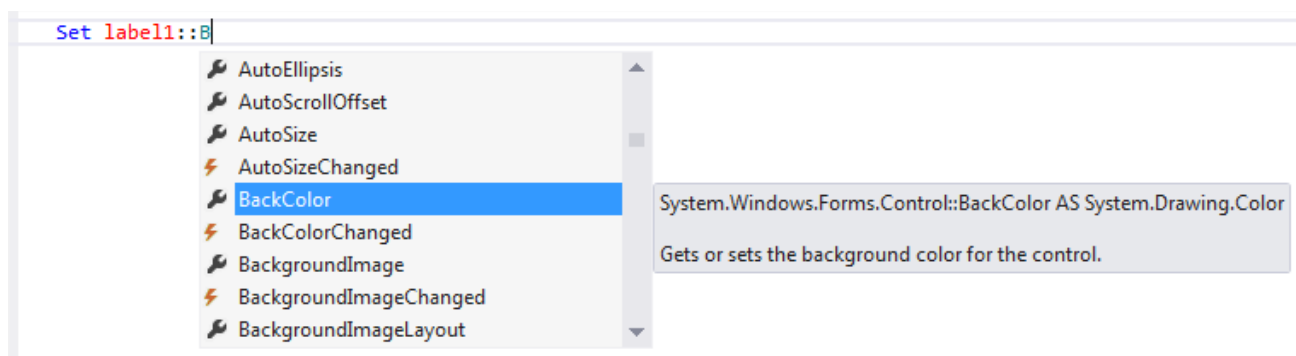


Figura 1.16 – IntelliSense mostrando as propriedades disponíveis para o controle



Para aplicar Zoom no tamanho da fonte , pressione a tecla Ctrl e role o scroll do mouse para frente. Para diminuir, role para traz.