



## APOSTILA VISUAL BASIC 6

### Sumário

- I. INTRODUÇÃO
- II. INICIANDO O VISUAL BASIC 6
- III. AMBIENTE DE TRABALHO
- IV. BARRA DE MENUS
- V. BARRA DE FERRAMENTAS
- VI. CAIXA DE CONTROLE (TOOLBOX)
- VII. A JANELA FORM
- VIII. A JANELA FORM LAYOUT
- IX. A JANELA DE PROPRIEDADES (PROPERTIES)
- X. A JANELA PROJECT EXPLORER
- XI. A JANELA CÓDIGO
- XII. EVENTOS
- XIII. CONSTRUINDO SEU PRIMEIRO PROGRAMA
- XIV. SALVE SEU PROJETO
- XV. COMPILANDO SEU PROGRAMA
- XVI. PREFIXOS
- XVII. TIPOS DE DADOS
- XVIII. SUFIXOS DE TIPO DE DADOS
- XIX. VARIÁVEIS
- XX. A INSTRUÇÃO IF
- XXI. EXPRESSÕES E OPERADORES MATEMÁTICOS
- XXII. MSGBOX
- XXIII. INPUTBOX
- XXIV. COMENTÁRIOS
- XXV. LOOPS
- XXVI. MENU EDITOR
- XXVII. BANCO DE DADOS
- XXVIII. ACTIVEX
- XXIX. DISTRIBUINDO SUA APLICAÇÃO
- XXX. ENDEREÇOS INTERNET
- XXXI. TERMO DE USO

## INTRODUÇÃO

O Visual Basic é uma linguagem de programação para Windows, baseada em eventos e em objetos. Gerando aplicativos cuja operação é determinada pela escolha de algum objeto gráfico, que representa uma ação, por parte do usuário.

O Visual Basic permite criar aplicativos tanto para máquina local quanto para ambiente de rede. Estes aplicativos podem tanto compartilhar banco de dados quanto acessar um banco de dados cliente-servidor.

O Microsoft Visual Basic 6 é a última e mais poderosa versão da linguagem desenvolvida pela Microsoft Corporation, baseada na linguagem Basic.

O Visual Basic 6 é vendido em três versões: Standard, Professional e Enterprise.

## INICIANDO O VISUAL BASIC 6

Para iniciar o Visual Basic 6 clique em iniciar.

Posicione o cursor do mouse em Programas/ Microsoft Visual Basic 6.

Clique em Visual Basic 6.

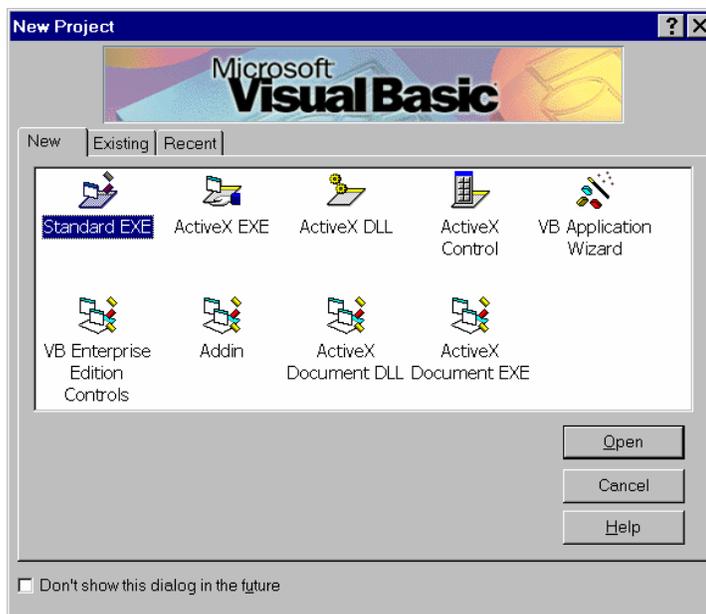
## AMBIENTE DE TRABALHO

Sempre que iniciar o Visual Basic, aparece a caixa de diálogo New Project.

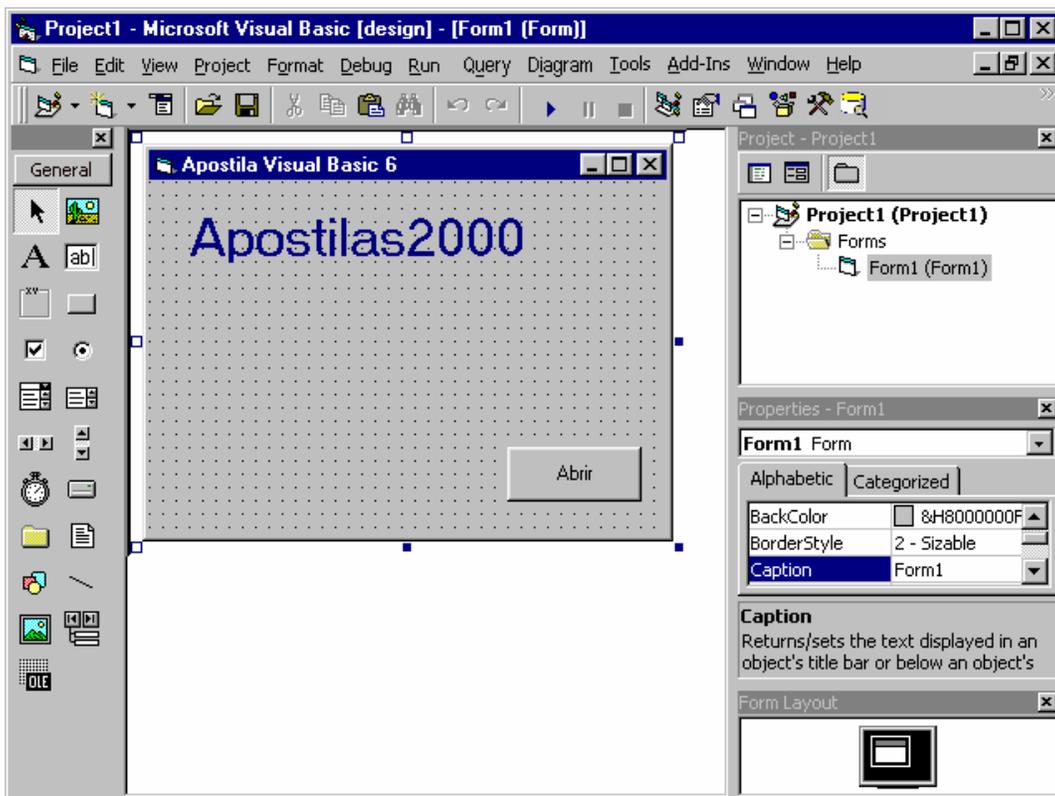
Solicitando que você escolha um projeto.

Dê um clique no botão Abrir (Open) para aceitar o novo projeto padrão (Standard.EXE), um aplicativo padrão do Visual Basic.

Dica: Se você der um clique na opção Don't show this dialog Box in the future, o Visual Basic não exibirá a caixa de diálogo de abertura quando o Visual Basic for iniciado.



O ambiente de programação do Visual Basic contém todas as ferramentas necessárias para construir seus programas para Windows com rapidez e eficiência.



### BARRA DE MENUS

A barra de menus proporciona acesso à maioria dos comandos que controlam o ambiente de programação. Os menus e comandos são convencionais, padrão Windows 9x.



Barra de menus

### BARRA DE FERRAMENTAS

A barra de ferramentas está localizada abaixo da barra de menus. Com botões que funcionam como atalhos para executar comandos e controlar o ambiente de programação.



Barra de ferramentas

	Adiciona um projeto padrão. Clique na seta e você terá outras opções de projeto
	Adiciona um novo formulário. Clique na seta e você terá acesso a diversos tipos de formulários.
	Mostra o menu Editor (editor de menus)
	Abre um projeto existente
	Salva o projeto atual
	Corta uma área selecionada
	Copia um objeto selecionado
	Cola um objeto da área de transferência
	Localiza uma palavra no código
	Desfaz operações
	Refaz operações
	Executa aplicação
	Pausa na execução da aplicação
	Finaliza a execução da aplicação
	Mostra a Janela de Projetos
	Mostra a Janela de Propriedades
	Mostra a Caixa de Layout do formulário
	Mostra o Object Browser
	Mostra a Caixa de Controles

Dica: No Visual Basic, você pode alinhar e acoplar, ou ancorar, as janelas para tornar visíveis e acessíveis todos os elementos do sistema de programação.

## CAIXA DE CONTROLE (TOOLBOX)



Contém ferramentas para inserir controle no seu formulário.

A Caixa de Controle mostra inicialmente os controles padrões do Visual Basic. Mas você pode inserir em sua Caixa de Controle diversos controles, chamados ActiveX.

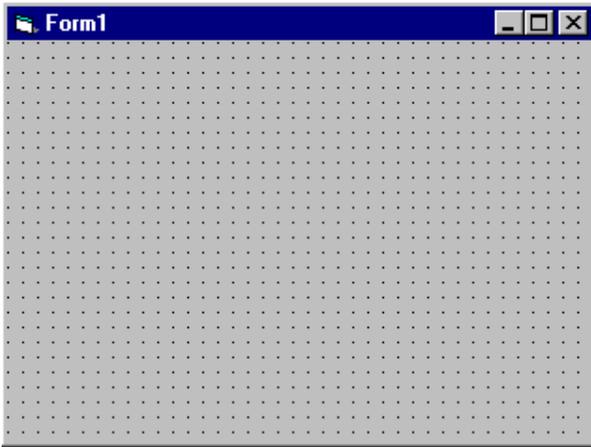
Os controles padrão são:

	Ferramenta de seleção. Usada para dimensionar e mover controle.
	PictureBox. Usada para mostrar imagens gráficas.
	Label. Usada para inserir texto. Que serve como rótulo.
	TextBox. Usado para criar caixas de edição, onde usuário inseri texto.
	Frame. Usado para criar uma moldura que agrupe controles.
	CommandButton. Usado para desenha um botão que poderá ser acionado para iniciar um comando.
	CheckBox. Usado para criar um botão de checagem.
	OptionButton. Usado para criar um botão de opção. Em conjunto com outros OptionButtons, o usuário pode escolher uma opção dentre várias.
	ComboBox. Usado para desenhar uma lista de itens em uma caixa combo. O usuário poderá interagir com a lista inserida na caixa combo.
	ListBox. Usada para mostrar uma lista de itens em uma caixa de lista, para que o usuário escolha um item.
	Hscrollbar. Usada para inserir uma barra de rolagem horizontal.
	Vscrollbar. Usada para inserir uma barra de rolagem vertical.
	Timer. Temporizador para controlar eventos. Invisível durante a execução da aplicação.
	Drivelistbox. Usada para mostrar uma lista de drives.
	DirListBox. Usada para mostrar uma lista de diretórios.
	FileListBox. Usada para mostrar uma lista de arquivos.
	Shape. Usada para desenhar algumas figuras geométricas.
	Line. Desenha linhas.
	Image. Usada para mostrar imagens bitmaps, ícones ou metafile.
	Data. Usada para acessar dados em banco de dados.
	OLE. Usada para ligar ou embutir objetos de outras aplicações.

Dica: Para inserir controles ActiveX na Caixa de Controles, clique com o botão direito em uma área vazia da Caixa de Controles. Clique em Componets. Defina o controle que você deseja inserir, e clique em OK.

### A JANELA FORM

A janela Form é sua principal área de trabalho. A janela Form sustenta os objetos do programa, tais como botões de comando, rótulos, caixas de texto, barra de rolagem entre outros.



A Janela Form

Ao dar um duplo clique em um dos controles da Caixa de controles ele será inserido na janela Form. A janela Form nada mais é do que seu programa, embora pareça pequena em relação ao resto da tela, ela representa todo o background do aplicativo.

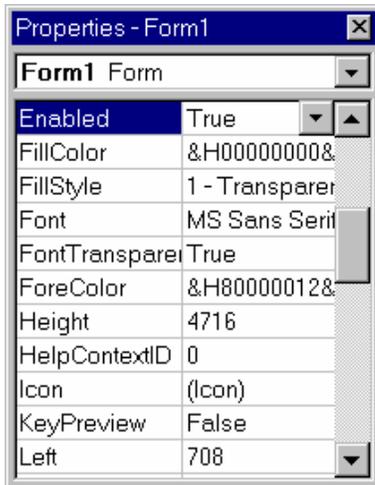
### A JANELA FORM LAYOUT

Esta janela permite que você posicione seu formulário na tela, durante o projeto. Quando você clica sobre o formulário mostrado na tela, o cursor muda de formato, e você pode move-lo na tela.



### A JANELA DE PROPRIEDADES (PROPERTIES)

Properties (Propriedades) são informações descritivas detalhadas sobre um controle.



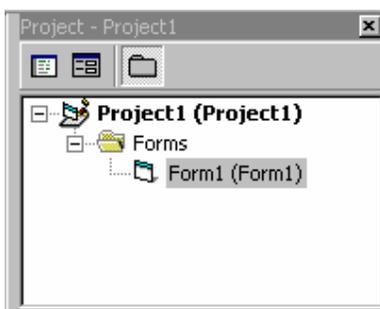
Quando um botão (Commandbutton) é inserido no formulário (Form) seu nome será Command1. O usuário verá este botão como Command1.

Você pode mudar este nome (Name) para cmd1 e o nome representativo (Caption) para Abrir.

É na Janela Propriedades (Properties), onde você altera as propriedades dos controles e do próprio formulário (Form).

### A JANELA PROJECT EXPLORER

A janela Project Explorer, freqüentemente chamada janela Project, oferece uma visão estruturada em árvore de todos os arquivos no aplicativo.

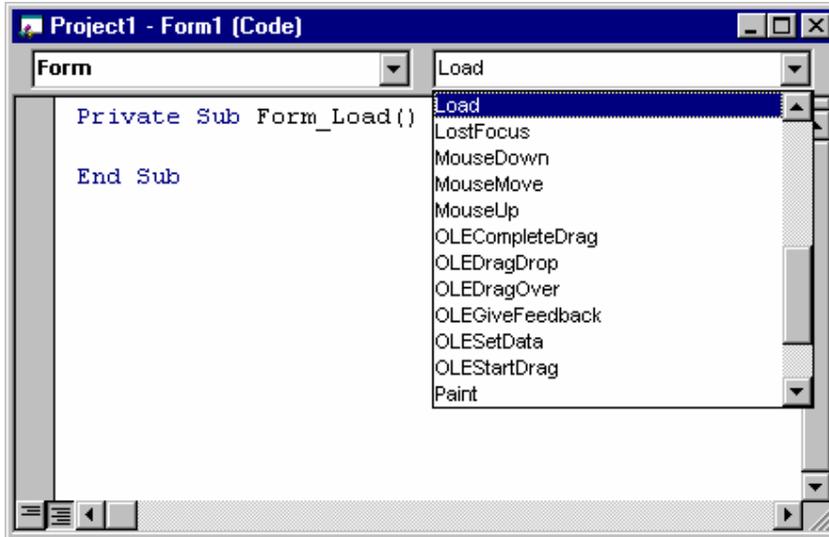


Bem semelhante à estrutura de árvore do Explorer do windows 9x e do NT.

A janela Project Explorer exhibe formulários, módulos (arquivos que contêm o código de suporte para o aplicativo), classes (módulos avançados) e aplicativo carregado.

## A JANELA CÓDIGO

A cada formulário está associada uma Janela de código, que contém declarações de variáveis, rotinas e funções referentes ao formulário e controle inseridos dentro do formulário.



Janela de Código

Na construção do código associado ao formulário, o Visual Basic utiliza a linguagem Basic.

Todas as procedures do formulário são mostradas na mesma Janela de Código.

Para abrir a Janela de Código use uma dessas formas:

- Após selecionar o formulário, clique no ícone (  ) da Janela de Projetos.
- Clique duas vezes em cima do formulário ou controle.
- Tecele F7
- Escolha a opção Code no menu View.
- Clique com o botão direito do mouse sobre o objeto e escolha View Code.

## EVENTOS

Os programas desenvolvidos em Visual Basic são orientados a eventos. Eventos são ações geradas pelo usuário clicar o mouse, apertar uma tecla, etc. Os eventos podem também ser gerados pelo próprio Windows. Existem eventos associados ao formulário, e cada controle inserido neste.

Alguns eventos comuns são:

Click	Ocorre quando o botão do mouse é clicado sobre uma área vazia ou sobre um controle que não está habilitado.
Dblclick	Ocorre quando o botão do mouse é clicado duas vezes seguido.
Gotfocus	Ocorre quando o objeto recebe o foco.
KeyDown	Ocorre ao se apertar uma tecla.
KeyPress	Ocorre ao ser apertar e soltar uma tecla.
KeyUp	Ocorre ao se liberar uma tecla.
Load	Ocorre quando um formulário é carregado.
Lostfocus	Ocorre quando um objeto perde o foco.
Mousedown	Ocorre quando o usuário pressiona o botão do mouse. Neste evento há identificação do botão pressionado (esquerdo ou direito) e ainda identifica as teclas Shift, Ctrl e Alt.
Mouseup	Ocorre quando o usuário libera o botão do mouse.

## CONSTRUINDO SEU PRIMEIRO PROGRAMA

Como vimos, o Visual Basic cria programas orientados a eventos. Chegou a hora de colocar tudo que foi aprendido até agora em prática. Tenha em mente o seguinte:

- 1º Qualquer objeto (botão, caixa de texto, rótulo, etc) inserido no formulário (Form) deve ter uma utilidade.
- 2º Modifique nome, tamanho, cor, fonte dos objetos caso necessário. Faça isso na Janela Propriedades.
- 3º Somente inicie a construção do código de seu programa após ter todos os objetos necessários inseridos no formulário (Form) e suas propriedades alteradas.

Inicie o Visual Basic 6

Clique em Iniciar

Posicione o cursor do mouse em Programas/ Microsoft Visual Basic 6.

Clique em Visual Basic 6.

Dê um clique no botão Abrir (Open) para aceitar o novo projeto padrão (Standard.EXE).

Dê um clique em Name na Janela Properties (Propriedades)

Altere Name "Form 1" para "Meuprograma".  
Altere Caption "Form1" para "Meu Primeiro Programa".

Dê um clique duplo no controle () Label.  
Na Janela Propriedades altere Name "Label1" para "lbl1"  
Altere Caption "Label1" deixe em branco "  
Altere Height para "200".  
Altere Width para "3000"

Dê um clique duplo no controle () Textbox  
Na janela Propriedades altere Name "Text1" para "txt1"  
Em Text "Text1" deixe em branco.  
Altere Height para "200".  
Altere Width para "3000"  
Deixe esse controle abaixo do rótulo.

Dê um clique duplo no controle () Commandbutton  
Na janela Propriedades altere Name "Command1" para "cmd1"  
Altere Caption "Command1" para "Pesquisar"

Dê um duplo clique no botão pesquisar  
Será aberta a Janela de Código.  
Com o seguinte código:

```
Private Sub cmd1_Click()  
End Sub
```

Digite entre estas linhas o seguinte código:

```
If txt1.Text = "a" Then  
lbl1.Caption = "Primeira letra do alfabeto"  
End If
```

Código completo:

```
Private Sub cmd1_Click()  
If txt1.Text = "a" Then  
lbl1.Caption = "Primeira letra do alfabeto"  
End If  
End Sub
```

Descrição do código:

Se (If) o texto da caixa de texto(txt1.text) é igual (=) à "a" então (Then) o rótulo (lbl1) Capta (Caption) Primeira letra do alfabeto

Para rodar seu programa clique no botão (  ) Start.  
 Digite a letra "a" na caixa de texto e clique no botão "Pesquisar".  
 Para finalizar o programa clique no botão (  ) End.

### SALVE SEU PROJETO

Clique no botão (  ) Save Project  
 Escolha a pasta onde serão armazenados seus Projetos.  
 Salve o Form1 com o nome de MeuPrograma. Dê um clique no botão Salvar.  
 Project1 como MeuPrograma. Dê um clique em no botão Salvar

### COMPILANDO SEU PROGRAMA

Para compilar seu programa e criar um executável  
 Clique em File/ Make MeuPrograma.exe...  
 Escolha a pasta onde serão armazenados seus arquivos com extensão.EXE.  
 Dê um nome para o executável (Ex: Dicionário) e clique no botão "OK".

Você criou um pequeno dicionário da língua portuguesa. Bem pequeno para falar a verdade, talvez você queira criar um dicionário completo. O primeiro passo já foi dado, já mostrei como criar a interface, propriedades e o código para primeira letra do alfabeto, só falta mais 30.000 palavras. Mãos a obra.

### PREFIXOS

Use estes prefixos para os nomes de controle

Prefixo	Controle	Nome em Inglês
cbo	Caixa combo	Combo Box
chk	Caixa de verificação	Check Box
cmd	Botão de comando	Command button
dir	Caixa de listagem de diretórios	Directory list Box
drv	Cx. de listagem de unidades de disco	Drive list Box
fil	Caixa de listagem de arquivos	File list Box
frm	Moldura	Frame
grd	Grade	Grid
hsb	Barra de rolagem horizontal	Horizontal scrollbar
img	Imagem	Image
lbl	Rótulo	Label
lin	Linha	Line
lst	Caixa de listagem	List Box
mnu	Menu	Menu
ole	Cliente OLE	OLE client
opt	Botão de opção	Option Button
pic	Caixa de Figura	Picture Box
shp	Forma	Shape
tmr	Temporizador	Timer
txt	Caixa de texto	Text Box
vsb	Barra de rolagem vertical	Vertical scrollbar

## TIPOS DE DADOS

Os dados dividem-se em três categorias: numérico, string e especial. Se você quiser trabalhar com um número, você precisará usar um que se encaixe em uma das categorias de tipo de dados do Visual Basic. Se quiser trabalhar com dados de texto, precisará usar uma string. Outros dados podem se encaixar em uma das várias categorias de tipos de dados especiais, como um item que representa um valor do tipo caixa de verificação True ou False.

O Visual Basic oferece os seguintes tipos de dados:

**Byte** (byte): é utilizado para criação de DLL's e para OLE. Ocupa de memória 1 byte.

**Boolean** (booleano): os valores que este tipo de dado poderá ter são True ou False / Yes ou No. Quando variáveis do tipo numérico se convertem para booleanas, tornam-se 0 se forem falsas e assumem qualquer outro valor para um resultado verdadeiro. Quando a variável booleana se converte para outro tipo, o False se torna 0, enquanto o True assume o valor de -1. Ocupa de memória 2 bytes.

**Integer** (inteiro): números sem decimais, cujo intervalo de ocorrência deverá ser do -32.768 a 32.767. Ocupa na memória 2 bytes.

**Long** (inteiro longo): qualquer tamanho de números sem decimais. Ocupa 4 bytes de memória.

**Single** (simples): números com até sete dígitos significativos. Ocupa 4 bytes de memória.

**Double** (duplos): números com até quinze dígitos significativos. Ocupa 8 bytes de memória.

**Currency** (monetário): armazena números com precisão de 15 casas à esquerda do ponto decimal, e quatro casas à direita. Esse tipo de dado é útil para cálculos que envolvam dinheiro. Ocupa 8 bytes de memória.

**Date** (data): representa uma data que poderá estar entre os dias 01/01/100 a 31/12/9999, e tempo que abrange o intervalo de 0:00:00 a 23:59:59. Ocupa 8 bytes de memória.

**Object** (objeto): uma variável declarada como object poderá referir-se a qualquer objeto criado pela aplicação. Ocupa na memória 4 bytes.

**String** (simples): Valores numéricos que variam de -3,408823E+38 para 3,402823E+38. Variável ou fixo. O tamanho que ocupa na memória: para string variável é de 10 bytes mais o tamanho da string e para string fixo apenas o tamanho da string.

**Variant** (variante): Dados de qualquer tipo usados para controle e outros valores para os quais o tipo de dados é desconhecido.

## SUFIXOS DE TIPO DE DADOS

Integer	%
Long	&
Simple	!
Double	#
String	\$
Currency	@
Variant	sem sufixo

## VARIÁVEIS

Para declarar variáveis, você precisa dar a uma variável tanto um nome quanto um tipo. Os nomes de variáveis podem ter até 40 caracteres, e precisam começar com uma letra.

Determinado o nome da variável, poderá declarar seu tipo usando um sufixo. Para que sua aplicação seja eficiente e clara, o mais indicado é especificar o tipo da variável, pois do contrário, a variável será vista pelo Visual Basic como do tipo Variant, e este tipo poderá ocupar mais memória do que o necessário.

**Declarando com o Comando Dim**

Veja a sintaxe:

Dim <nome da variável> As <tipo da variável>

Exemplo:

```
Dim Apostilas As String
```

Você pode declarar sua variável dentro de uma procedure, na seção General de um formulário, ou dentro de um arquivo módulo.

Exemplo de uma variável do tipo Integer dentro de uma procedure.

```
Private Sub Command1_Click()
    Dim Apostilas As Integer
    Apostilas = Val(txt1.Text)
End Sub
```

**Atribuindo Valores**

A atribuição de valores no Visual Basic é feita com o sinal de igualdade “=”

O formato da atribuição de valores é:

resultado = expressão

Exemplo:

```
Total = txt1 * txt2
```

### **Constantes**

Utilizamos constantes para substituir no programa a aparição freqüente de valores, que durante a execução do aplicativo não sofrem alterações. Uma constante poderá ser uma string, um valor numérico, outra constante, ou qualquer combinação que inclua os operadores lógicos ou aritméticos.

### **Declarando Constante**

Constantes podem ser declaradas dentro de uma procedure, no arquivo de módulo ou na seção General do formulário.

A sintaxe para a declaração de constante é:

```
Const nome_da_constante = valor
```

Exemplo:

```
Private Sub Command1_Click()  
    Const Apostilas = "Pesquisar"  
    Command1.Caption = Apostilas  
End Sub
```

### **A INSTRUÇÃO IF**

Talvez a mais importante instrução em um programa seja a instrução If. Seu aplicativo pode analisar dados e tomar decisões com base nessa análise.

If utiliza os operadores de comparação. If realiza uma entre duas possíveis ações do código, dependendo do resultado da comparação.

Se um teste comparativo for verdadeiro, o corpo de uma instrução If é executado.

Eis um formato de If:

```
If testeComparativo Then  
    Uma ou mais instruções Visual Basic  
End If
```

End If permite que o Visual Basic saiba onde o corpo de uma instrução If termina.

## EXPRESSÕES E OPERADORES MATEMÁTICOS

Para poder calcular e atribuir resultados de uma expressão a variáveis e quando codificar declarações que contêm expressões você deve aprender operadores matemáticos do Visual Basic.

Os principais operadores matemáticos são:

Operador	Descrição
+	Adiciona dois valores
-	Subtrai um valor de outro
*	Multiplica dois valores
/	Divide um valor por outro
^	Eleva o valor a uma potência
& (ou +)	Concatena dois strings

Exemplo:

Resultado = 6 / 3 + 5 + 4 \* 2

O Visual Basic computa a divisão primeiro que aparece a esquerda da multiplicação. Se a multiplicação aparecesse à esquerda da divisão, o Visual Basic faria primeiro a multiplicação. Depois o Visual Basic calcula a adição mostrando a resposta final, em Resultado.

## FUNÇÃO

Função é uma instrução que desempenha um trabalho importante (tal como solicitar uma informação do usuário ou calcular uma equação) e então retorna um resultado ao programa.

## MSGBOX

A função MsgBox() só permite um diálogo restrito; você coloca uma mensagem na tela em uma janela, e o usuário fica restrito a comunicar-se de volta por meio de botões.

Sintaxe:

Msgbox (mensagem, opções, título, ar\_ajuda, contexto).

Onde:

Mensagem: é a mensagem que você quer mostrar.

Opções: indica a soma dos valores escolhidos entre as tabelas abaixo:

### Botões

0	Somente botão OK
1	Botão OK e Cancel
2	Botões Abort, Retry, Ignore
3	Botões yes, no, cancel
4	Botões yes, no
5	Botões Retry, Cancel

Ícones

16	Erro grave
32	Pergunta
48	Sinal de aviso
64	Informação

Botão Padrão

0	Primeiro botão
1	Segundo botão
2	Terceiro botão

Por exemplo, se você criar um quadro de diálogo contendo os botões Yes e No, o ícone de interrogação e com, o primeiro botão padrão, o valor do argumento opções seria 36:

$$4 + 32 + 0 = 36$$

Título: é a string que você deseja posicionar na legenda da janela do quadro de mensagem.

Arq\_ajuda e contexto: só são usados se você quiser associar o quadro de diálogo de um arquivo de help.

Como MsgBox é uma função, você deve associa-la a uma variável de retorno. Os valores de retorno de MsgBox() são:

1	Botão OK foi pressionado
2	Botão Cancel foi pressionado
3	Botão Abort foi pressionado
4	Botão Retry foi pressionado
5	Botão Ignore foi pressionado
6	Botão Yes foi pressionado
7	Botão No foi pressionado

Exemplo:

Mensagem = MsgBox ("Salvar o documento?", 36, "Salvar")

Caso não haja valor de retorno para MsgBox, não é necessário colocar os parênteses que contem os parâmetros.

MsgBox passa a ser uma procedure, ao invés de uma função.

Exemplo:

MsgBox "Salvar o documento?", 36, "Salvar"

## INPUTBOX

A função InputBox fornece um quadro de diálogo padrão, que permite entrada de dados.

Sintaxe:

InputBox (mensagem, título, default)

Onde:

mensagem: é o texto que mostramos para indicar que tipo de entrada é desejada;

título: é a legenda que queremos dar ao quadro de entrada;

default: é o string que aparece no quadro de edição do quadro diálogo (caso o usuário não faça nenhuma entrada);

Exemplo:

```
Private Sub Form_Load()  
    apostilas = InputBox("Digite seu nome", "Login")  
    Label1.Caption = apostilas  
End Sub
```

## COMENTÁRIOS

Os comentários de instrução ajudam você e outros programadores a modificar e atualizar seus projetos ou programas Visual Basic. As vezes será necessário retornar a seu projeto para modificar certas instruções no código do programa.

Se você programa para uma empresa, é bem provável que outras pessoas modifiquem os programas que você escreveu.

Então nada melhor do que deixar uma pista de sobre determinadas instruções.

No Visual Basic utilizamos dois tipos de comentários

Que começam com a instrução Rem

Exemplo:

Rem este comando abre o formulário vendas

Que começam com o apóstrofo (')

Exemplo:

' este comando abre o formulário vendas

Exemplo completo:

```
Private Sub Command1_Click()  
    ' este comando abre o formulário vendas  
    Form2.Show  
End Sub
```

## LOOPS

Um loop é um conjunto de instruções de programa que executam repetidamente.

Loops desempenham um importante papel nos programas porque você precisará às vezes repetir seções de um programa para processar múltiplos valores de dados.

### Loop Do While

Do While funciona com expressões de comparações exatamente como uma instrução If.

Exemplo:

```
Do  
Apostilas = InputBox("Digite o seu nome ou Fim para sair.")  
If Apostilas <> "Fim" Then Print Apostilas  
Loop While Apostilas <> "Fim"
```

### Loop Do Until

O loop Do Until executa o corpo do loop enquanto o teste comparativo for falso.

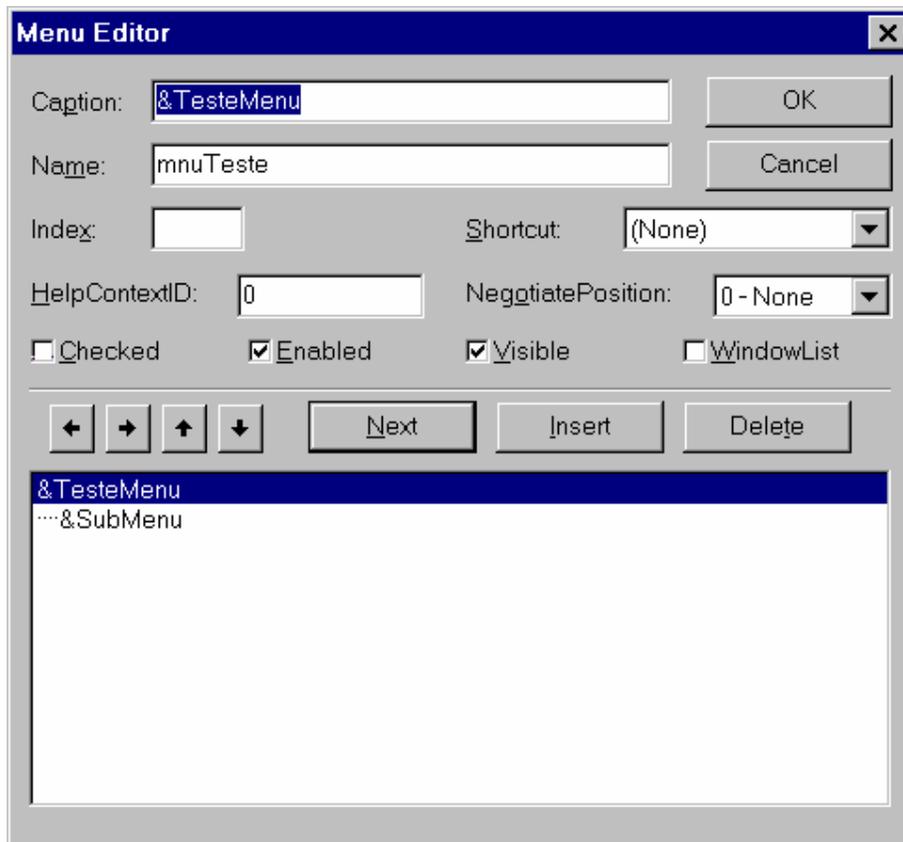
Exemplo:

```
Do  
Apostilas = InputBox("Digite o seu nome ou Fim para sair.")  
If Apostilas <> "Fim" Then Print Apostilas  
Loop Until Apostilas <> "Fim"
```

Caso tenha notado no loop (Do While) basta digitar Fim na janela prompt e clicar no botão "OK" para entrar no programa enquanto no loop (Do Until) caso você digite Fim na janela prompt e clique no botão "OK" o laço continuará.

## MENU EDITOR

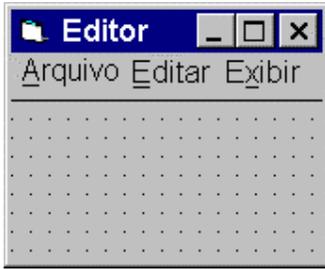
Para colocarmos um menu em nosso formulário, temos que projetá-lo primeiro usando o (  ) Menu Editor.



Para criar um meu em seu formulário:

- Entre com o nome do menu (aquele que aparecerá na Barra de Menu), no quadro Caption. A medida que você digita o nome do menu em Caption, a mesma palavra aparece no quadro abaixo, que é local onde o menu projetado será mostrado.
- Entre com uma identificação para o controle, no quadro Name;
- Pressione Enter ou clique no botão Next para terminar o primeiro item e passar para o próximo. Caso seja um subitem, clique sobre a seta para direita, para dar uma endentação;
- Entre com os demais itens. Se quiser incluir um separador entre os itens de menu, digite um hífen(-) na caixa Caption;
- Feche a janela de projeto de menu clicando no botão "OK".

O menu é inserido na parte superior de seu formulário.



Para inserir o código clique sobre o item de menu desejado e digite as instruções necessárias.

Exemplo:

```
Private Sub mnufechar_Click()  
End  
End Sub
```

## BANCO DE DADOS

A maioria dos objetos que você cria usando os controles da caixa de ferramenta do Visual Basic possuem aptidão interna para exibir informações de banco de dados. Um objeto está limitado a um banco de dados quando sua propriedade DataSource estiver definida com um nome válido de banco de dados e sua DataField como uma tabela válida do banco de dados. É possível vincular o seu programa em Visual Basic a um banco de dados por meio de um objeto dados. Depois que a conexão for estabelecida, você poderá exibir informações do banco de dados usando objetos criados no formulário.

### Conexão

Para conectar seu programa Visual Basic um banco de dados inclua os seguintes objetos:

- Data
- Text
- Label

Modifique:

- Label1: Caption para Autor
- Text1: Text em branco
- Data: Caption para Autores

Para o Objeto Data:

Clique em Recordset Type escolha 0- Table

Clique em Databasename escolha o banco de dados exemplo do Visual Basic, escolha o caminho (que pode ser outro) C:\Arquivos de programas\Microsoft Visual Studio\vb98\Biblio.mdb.

Em:

ReadOnly marque True

Recordset Type marque 0 – Table

RecordSource escolha Authors

O programa criado exibe somente os campos desejados. Usando um objeto dados e diversas caixas de texto de dados limitado, você pode criar um aplicativo de banco de dados eficiente que acesse seu próprio banco de dados, seja ele Sql, Access, dBase entre outros.

## ACTIVEX

Procure em uma livraria e você encontrará muitos livros grossos que discutem ActiveX. ActiveX é a nova tecnologia aberta da Microsoft que foi projetada para mesclar a área de trabalho com a Internet de maneira transparente.

Um controle ActiveX é um controle que você pode adicionar à janela Toolbox do Visual Basic. Se a janela Toolbox não contiver o controle de que você precisa, você poderá localizar um controle ActiveX que sirva aos seus propósitos e adicionar esse controle à janela Toolbox para utilizar em seu aplicativo.

Muitos fabricantes comercializam controles ActiveX e você pode pesquisar na Internet utilizando alguns sistemas de busca para encontrar um lista de sites relacionados com Activex.

Acredita-se que os futuros sistemas operacionais serão baseados em ActiveX. Na verdade, todos os programas serão controles ActiveX.

## Documentos ActiveX

Os documentos ActiveX são objetos difíceis de criar do zero. Um documento ActiveX deve estar contido dentro de um aplicativo ActiveX recipiente, como o Internet Explorer.

Inicie o Internet Explorer. Abra um documento Word. O internet Explorer pode exibir o documento Word, completamente formatado, e você pode editar o documento como se estivesse dentro do Word.

Palavras não reconhecidas, palavras estrangeiras e algumas abreviaturas são sublinhadas como possíveis erros de digitação; e você pode selecionar e formatar o texto.

Quando um recipiente ActiveX ativa um documento ActiveX, todos os controles usuais do documento e suas características tornam-se disponíveis.

Um documento Word é um documento ActiveX.

O Internet Explorer é um recipiente ActiveX.

### Visual Basic ActiveX Document Migration Wizard

Inicie um no projeto (StandardEXE).

Iniciar/ Programas/Microsoft Visual Basic 6/ Visual Basic 6.

Insira em seu formulário (Form) um Label e um Commandbutton.

Para Label modifique:

Name para lbl1

Caption deixe em branco

Width para 3000

Para Commandbutton

Name para cmd1

Caption para Clique aqui

Dê um clique duplo no botão cmd1 e digite o seguinte código.

```
lbl1.Caption = "Visual Basic é legal"
```

Código completo

```
Private Sub cmd1_Click()  
    lbl1.Caption = "Visual Basic é legal"  
End Sub
```

Faça os testes necessários clicando em Run/Start

Feche seu pequeno aplicativo.

Salve seu projeto clicando no botão  Save project.

Dê um nome para Form1, exemplo "Teste".

Dê um nome para Project1, exemplo "Teste".

Clique em Add-Ins/ Add-In manager...

Dê um clique duplo em VB 6 ActiveX Doc Migration Wizard.

Clique em "OK"

Clique em Add-Ins/ ActiveX Document Migration Wizard...

Selecione todas opções padrão menos uma, escolha "dll" ao invés de "exe" e clique em finalizar.

Feche a caixa de diálogo final.

Clique em File/ Save docForm1.dob.

Clique em File/Make Teste.dll.

Escolha o local e clique em "OK".

Para testar, abra o Internet Explorer 4.0 ou superior (não funciona no Netscape).

Clique em Arquivo/ Abrir localize seu arquivo com extensão .vbd (docForm1.vbd) clique em abrir.

Faça os testes necessários.

Se o documento não abrir ou mostrar a caixa de diálogo “Abrir como” escolha o programa Internet Explorer.

Este é apenas um exemplo simples de documento ActiveX. O limite é sua criatividade e imaginação.

#### DISTRUINDO SUA APLICAÇÃO

A aplicação desenvolvida no Visual Basic poderá ser distribuída em discos, CD's, rede ou Internet, que poderão ser instalados em seu cliente. Não basta, porém, dar o arquivo executável de sua aplicação para o cliente.

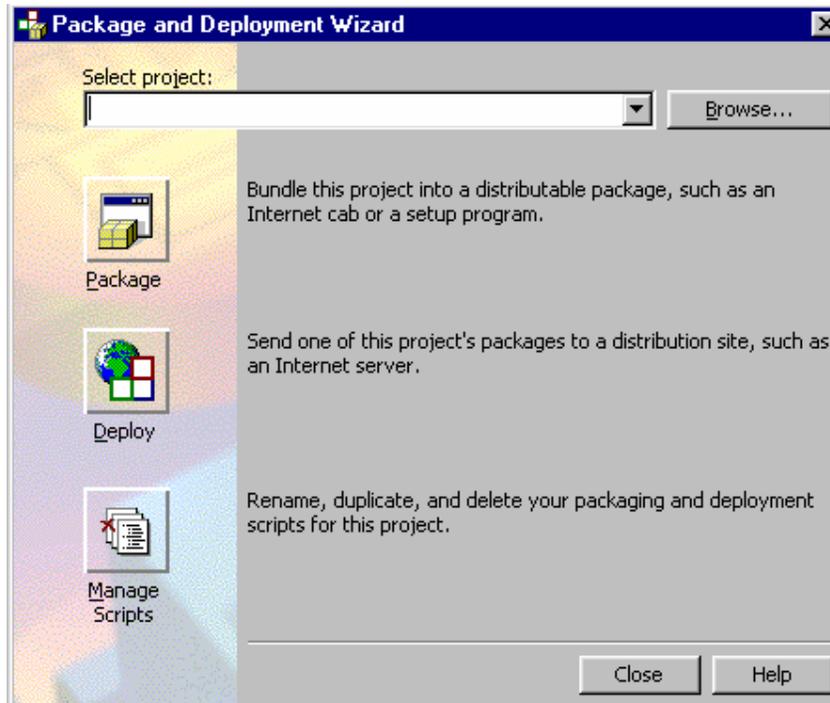
Inúmeros outros arquivos são necessários.

Para facilitar a distribuição do aplicativo, o Visual Basic oferece utilitário Package and Deployment Wizard, que cria os disco que conterão todos os arquivos necessários para a instalação de seu aplicativo no cliente.

Com este utilitário você poderá:

- Recompilar automaticamente o arquivo .EXE;
- Comprimir arquivos e distribuí-los em disquetes;
- Ser avisado da necessidade de certos arquivos para que sua aplicação funcione;
- Ser notificado sobre o número de disquetes necessários para distribuir sua aplicação nos discos de instalação.

Para usar o Package and Deployment Wizard no grupo de programas do Visual Basic, selecione a opção Package and Deployment Wizard.



Selecione o projeto que será distribuído, e a forma de distribuição.  
Selecione o tipo de pacote, e clique em Next.  
Selecione a pasta onde o pacote será criado, e clique em Next.  
Selecione os arquivos que farão parte do pacote, e clique Next.  
Selecione o tipo de arquivo CAB desejado, e clique em Next.  
Digite o nome do projeto que será mostrado durante a execução do programa de instalação, e clique em Next.  
Determine os grupos e itens de menu para o projeto, e clique em Next.  
Determine o local onde será instalado o arquivo em seu cliente, e clique em Next.  
Defina se os arquivos serão compartilhados e clique em Next.  
Digite um nome para o script, e clique em Finish.  
Os arquivos CAB são criados.