

Módulo
5

Delphi - Ambiente (IDE) e Criação de Programas I

INTRODUÇÃO

Esta lição visa familiarizá-lo com o Delphi 7.0 Enterprise. Durante a lição abordaremos os principais componentes do Delphi e suas principais propriedades.

CARACTERÍSTICAS DO DELPHI

- * Gera um executável verdadeiro, independente de run-time.
- * Utiliza a linguagem Object Pascal para escrever os procedimentos do programa.
- * Utiliza o processo de desenvolvimento Two-Way, que permite tanto escrever o código em Object Pascal gerando os objetos visuais, como utilizar os métodos visuais gerando código em Object Pascal.
- * Os componentes são definidos como objetos, o que permite a herança.
- * Permite a criação de novos componentes na própria linguagem.
- * Possui acesso facilitado a banco de dados.
- * Possui ambiente de depuração integrado.
- * Possui componentes para a internet.

CARACTERÍSTICAS DA PROGRAMAÇÃO DELPHI

- * Um programa Delphi é uma estrutura de aplicativo orientada ao desenho de formulários ou janelas.
- * Interface com usuário feita através de componentes.
- * Contém um conjunto de controles pré-desenvolvidos que dão acesso às características do sistema.
- * Os componentes são objetos.
- * Cada controle ou componente possui propriedades, métodos e pode responder a eventos.
- * As propriedades podem ter seus valores definidos em tempo de desenvolvimento e alterados em tempo de execução.
- * Os eventos são as mensagens que cada componente pode responder, tendo associado a eles um procedimento de evento.

ELEMENTOS DA PROGRAMAÇÃO DELPHI

Elemento	Descrição
Formulário (Form)	É uma janela, elemento básico onde agrupamos os componentes para formar a interface com o usuário.
Unidade (Unit)	Arquivo que contém código em object pascal. Para cada formulário existe uma unidade associada.
Componente	Objetos utilizados para a construção das nossas aplicações (projeto).
Propriedade	Representam os atributos dos componentes.
Método	Procedimento ou função própria do objeto.
Evento	Representam a capacidade de resposta dos componentes aos estímulos.
Processador de Evento	Procedimento responsável por responder a determinado evento.
Projeto (Project)	Conjunto de formulários, componentes e unidades que compõem uma aplicação.

ARQUIVOS PRODUZIDOS PELO SISTEMA

Ext.	Tipo	Descrição
BMP ICO	Arquivos gráficos	Arquivos de imagens nos formatos BitMaP e ICOne.
~DF	Backup de DFM	Backup de um arquivo DFM.
~DP	Backup de Projeto	Backup de um arquivo DPR.
DSK	Configurações de Desktop	Arquivo texto contendo as informações sobre a posição das janelas, os arquivos abertos no editor e outras configurações de Desktop.
DSM	Dados do Object Browser	Armazena as informações do Object Browser.
EXE	Arquivo executável linkeditado	Arquivo executável contendo as unidades, recursos e formulários compilados de um projeto.
OPT	Opções do Projeto	Arquivo de teste com as configurações atuais para as opções do projeto.
PAS	Código-fonte de uma unidade	Arquivo contendo o código fonte de uma unit em object pascal, o qual pode ser de um formulário ou arquivo fonte independente. Sendo de um formulário contém a sua definição de classe e código dos seus manipuladores de eventos.
~PA	Backup de um PAS	Backup de um arquivo PAS.
RES	Arquivo de recursos compilado	Arquivo binário associado ao projeto contendo recursos compilados, por padrão contem o ícone do projeto.
DCU	Unit Compilada	Arquivo PAS compilado.
DFM	Arquivo de formulário gráfico	Arquivo binário contendo as propriedades e componentes de um formulário.
DPR	Arquivo de Projeto	Escrito em Object Pascal contendo os componentes de um projeto e permite uso de código de inicialização do projeto.

Estrutura de um Projeto: Projeto (*.DPR), Units (*.PAS) e Forms (*.DFM)

Segue uma descrição das mais importantes opções de menu para o gerenciamento de projetos, algumas dessas opções têm um botão correspondente na barra de ferramentas.

File	
New	Abre um submenu com novos itens que podem ser adicionados ao projeto.
Open	Abriu projetos, pode abrir também Units, Forms e texto no editor de código.
Reopen	Abre um submenu com os 5 últimos projetos usados e as 10 últimas units utilizadas.
Save (Ctrl+S)	Salva o arquivo aberto no editor de código.
Save Project As	Salva o projeto com outro nome ou em outro local.
Save All (Shift+Ctrl+S)	Salva as alterações realizadas em todas as units do projeto. (Uso recomendado).
Use Unit (Alt+F11)	Faz com que a Unit atual possa usar outra Unit do projeto.
View	
Project Manager (Ctrl+Alt+F11)	Mostra o gerenciador de projeto.
Object Inspector (F11)	Mostra o Object Inspector.
Toggle Form/Unit (F12)	Alterna entre o Form e a Unit.
Units (Ctrl+F12)	Mostra o código fonte de uma Unit ou do Projeto a partir de uma lista.
Forms (Shift+F12)	Seleciona um Form a partir de uma lista.
Project	
Add to Project (Shift+F12)	Adiciona uma Unit em disco ao projeto.
Remove from Project	Remove uma Unit do projeto.
View Source	Mostra o código do projeto.
Compile (Ctrl+F9)	Compila o projeto.
Options (Shift+Ctrl+F11)	Opções do projeto, como ícone do executável, nome da aplicação e opções de compilação.
Run	
Run (F9)	Compila e executa o projeto.
Program Reset (Ctrl+F2)	Pára a execução do programa.

PROGRAMAÇÃO EM WINDOWS: JANELAS E EVENTOS

Existem muitas diferenças entre a programação em DOS e a programação em Windows. Vejamos a principal: Quando programamos em DOS, nosso programa é responsável pelo fluxo de processamento. Temos que definir claramente não só que instruções, mas também em que ordem devem ser executadas. Em Windows não é bem assim. Nosso programa não controla o fluxo de processamento, ele responde e trata eventos que ocorrem no sistema. Existem muitos eventos que podem ocorrer, sendo que os principais são aqueles gerados pelo usuário através do mouse e do teclado. A coisa acontece mais ou menos assim: O usuário clica o mouse e o Windows verifica que aplicação estava debaixo do mouse no momento em que foi clicado. Em seguida ele manda uma mensagem para a aplicação informando que ocorreu um clique e as coordenadas do cursor do mouse na tela no momento do clique. A aplicação então responde à mensagem executando uma função de acordo com a posição do mouse na tela. É claro que o Delphi toma conta do serviço mais pesado e facilita muito as coisas para o programador. Detalhes como as coordenadas da tela em que ocorreu o clique, embora estejam disponíveis, dificilmente são necessários nos programas.

PROGRAMAÇÃO ORIENTADA A OBJETO (POO)

Embora não seja objetivo desta lição ensinar POO, uma breve introdução é necessária, já que o Delphi é essencialmente orientado a objeto.

De maneira prática, podemos pensar no objeto sendo uma estrutura que agrupa dados e funções para manipular estes dados. Como as funções são sempre "íntimas" dos dados, o sistema todo funciona de maneira mais segura e confiável. Além disso, a POO utiliza conceitos como encapsulamento e herança que facilitam muito a programação e a manutenção dos programas.

Neste ponto é oportuno citar que os dados de um objeto costumam ser chamados de variáveis de instância e as funções de métodos. As variáveis de instância definem as propriedades (às vezes chamadas de atributos) do objeto e os métodos definem seu comportamento.

Encapsulamento

Como as variáveis e métodos estão na mesma estrutura, pode-se pensar em variáveis e métodos privados, ou seja, dados e funções que só podem ser manipulados pelas funções que estão dentro da estrutura. Desta maneira é possível formar uma camada protetora nos dados e evitar atribuições desastradas que comprometeriam o

funcionamento do programa. Os defensores mais ortodoxos da POO dizem que todos os dados de um objeto deveriam ser privados e o número de funções públicas deve ser o menor possível, mas isso nem sempre é viável ou prático. O Delphi implementa este conceito e oferece dados/funções públicas (public) e privadas (private). Outra consequência do encapsulamento é que os objetos podem ser “caixas pretas”. Não é necessário (teoricamente) conhecer detalhes de funcionamento de um objeto para usá-lo, basta enviar as mensagens apropriadas que ele responde com a ação desejada.

Classes

A classe representa um tipo ou categoria de objetos, o modelo a partir do qual um objeto pode ser construído. É a estrutura propriamente dita, que define os dados e métodos daquela classe de objetos. O objeto em si é uma instância da classe. Na programação estruturada podemos fazer uma analogia com os tipos e variáveis, onde a classe equivale ao tipo e o objeto à variável desse tipo.

Herança

É a capacidade que uma classe de objetos tem de herdar variáveis e métodos de outra classe. Esta capacidade permite que o código já escrito seja reutilizado de maneira muito mais eficiente e simples do que na programação estruturada. Um programa orientado a objeto costuma implementar verdadeiras árvores genealógicas de classes, com vários níveis de herança.

Para programar no nível do designer (veja adiante o que significa) não é necessário um conhecimento profundo de POO. Mas é preciso conhecer pelo menos a sintaxe. Todas as aplicações para Windows precisam de pelo menos uma janela, que no Delphi é chamada de Form. Cada form (assim como todos os objetos visuais) tem um objeto associado a ele e sua representação visual, que vemos na tela. Todos os componentes que incluímos no form passam a fazer parte do objeto que define o form. Exemplo: Se colocarmos um botão no form, a classe deste form será modificada para incluir este botão. Os eventos e métodos deste form, também estão na classe. Assim, supondo que o form se chame Form1 (nome default), para, por exemplo, desativar o botão que incluímos (de nome Button1) faríamos:

```
Form1.Button1.Enabled := false;
```

Note como o Button1 faz parte da estrutura que define o form. Mas se quisermos ativar o método RePaint (Repintar) do form faríamos:

```
Form1.Repaint;
```

Veja que Repaint, não é uma variável, tecnicamente é uma procedure, mas fazemos referência a ela como parte da estrutura Form1. Pode parecer confuso no início, mas facilita muito a programação.

O Delphi oferece dois níveis de programação distintos. Existe o nível que é chamado de *designer*, que se utiliza dos recursos de programação visual e aproveita componentes prontos, e o nível do *component writer*, que escreve os componentes para o designer utilizar nas aplicações. Podemos dizer que o *component writer* programa em um nível mais baixo (mais difícil) e o *designer* em um nível mais alto (mais fácil e compreensível). Neste curso, estamos estudando a programação no nível do *designer*.

Quando ativamos o Delphi, a tela inicial é parecida com a figura 5.1. Na janela superior, temos a Barra de Menu Principal do Delphi, à esquerda a SpeedBar (Barra de Ferramentas Rápida), com as opções mais comuns e à direita a Paleta de Componentes. Estes componentes são a base da programação visual e é onde o designer vai buscar recursos para sua aplicação. Veja com mais detalhes na figura 5.2.

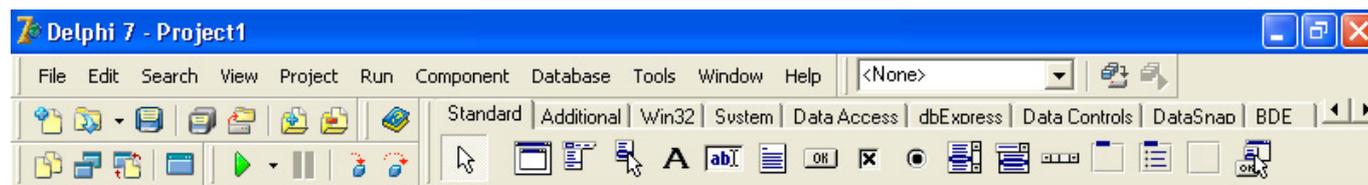


Figura 5.2 – Barra de Menu, SpeedBar e Paleta de Componentes.

O AMBIENTE DO DELPHI

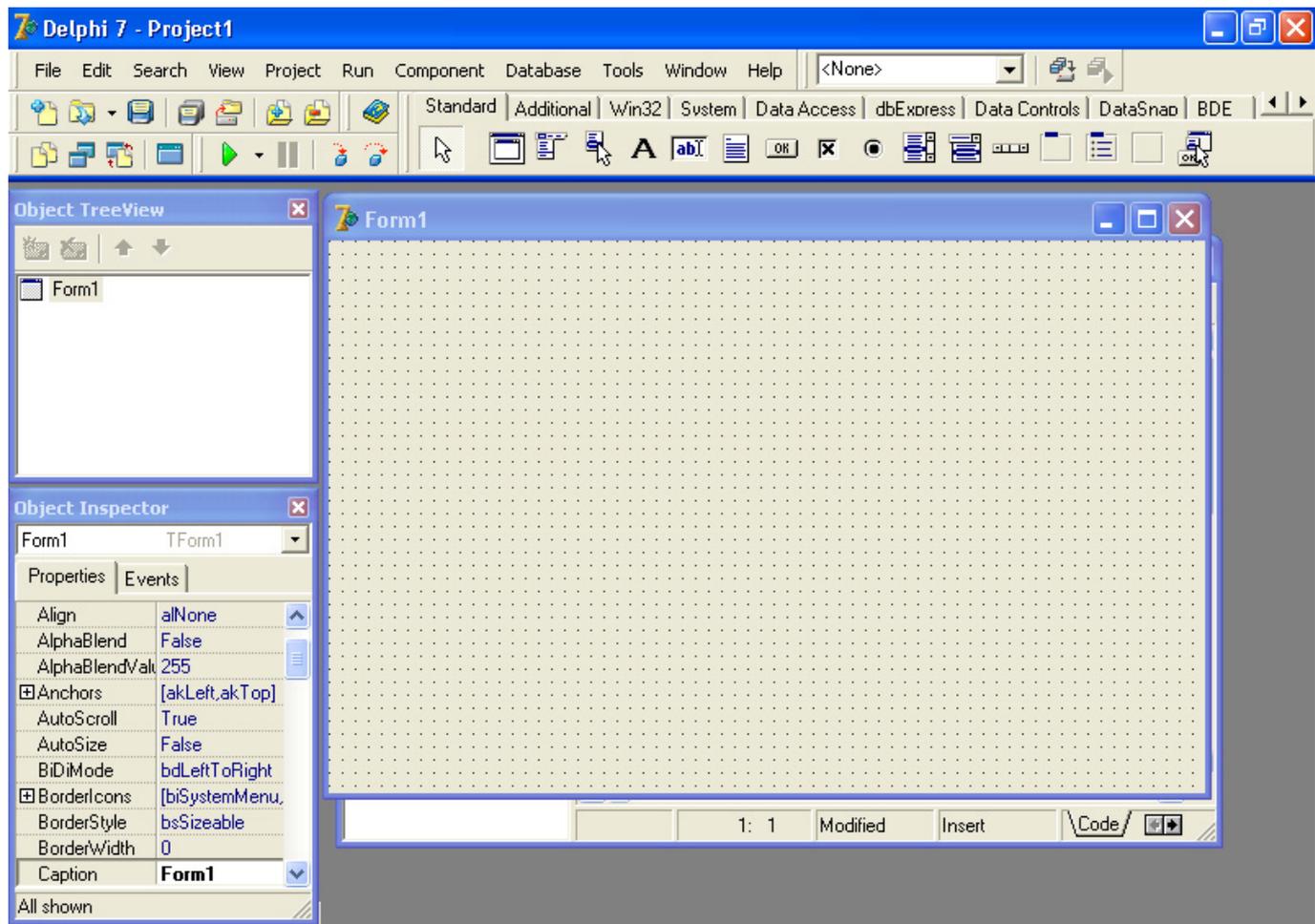


Figura 5.1 - Quando iniciar o Delphi pela primeira vez você verá a tela acima, que representa a IDE do Delphi.

Abaixo da SpeedBar, está a janela do Object TreeView, que permite visualizar e acessar todos os componentes dentro do formulário que você está utilizando no momento. Observe na figura 5.3.

Abaixo do Object TreeView encontra-se o Object Inspector, que permite visualizar e modificar as propriedades e eventos de todos os componentes. É também largamente utilizado pelo designer. Veja na figura 5.4.

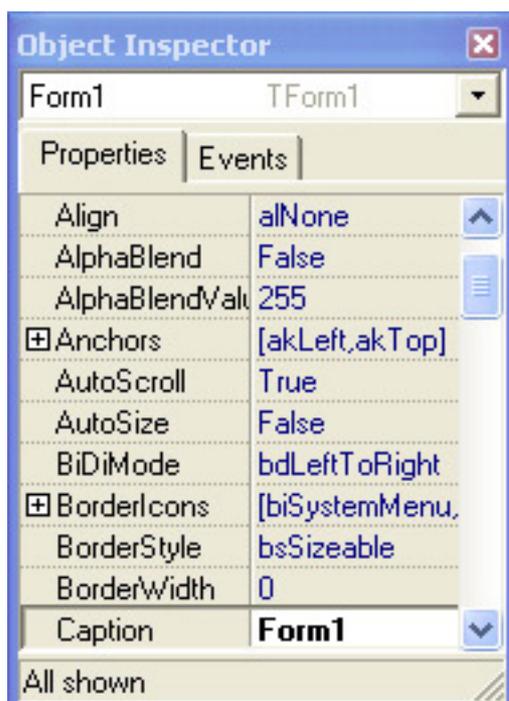


Figura 5.4 – Object Inspector

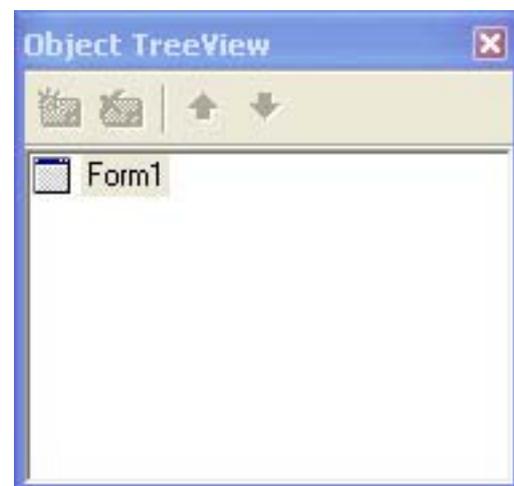


Figura 5.3 – Object TreeView

Abaixo da Paleta de Componentes ficam a janela de código-fonte e as janelas que estão sendo construídas.

As janelas que estão sendo construídas são chamadas de Forms (formulários) e cada form tem uma janela de código-fonte pertencente a ela, que são chamadas de Units.

Veja as figuras 5.5 e 5.6.

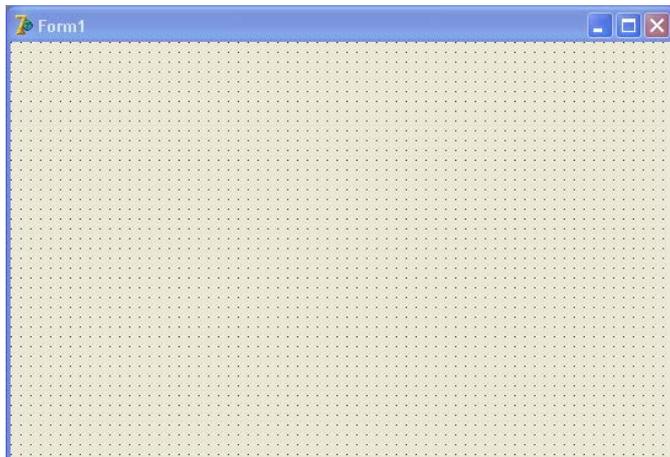


Figura 5.5 – Form (Formulário)



Figura 5.6 – Unit (Código-Fonte)

PALETAS DE COMPONENTES

A seguir visualizaremos as paletas de componentes que estudaremos durante este curso. O Delphi 7 possui outras paletas padrões de componentes, mas não é nosso objetivo estudá-las agora. Além disso você pode encontrar milhares de componentes na Internet.



Figura 5.7 – Paleta Standard - Uma das paletas mais utilizadas pelo programador

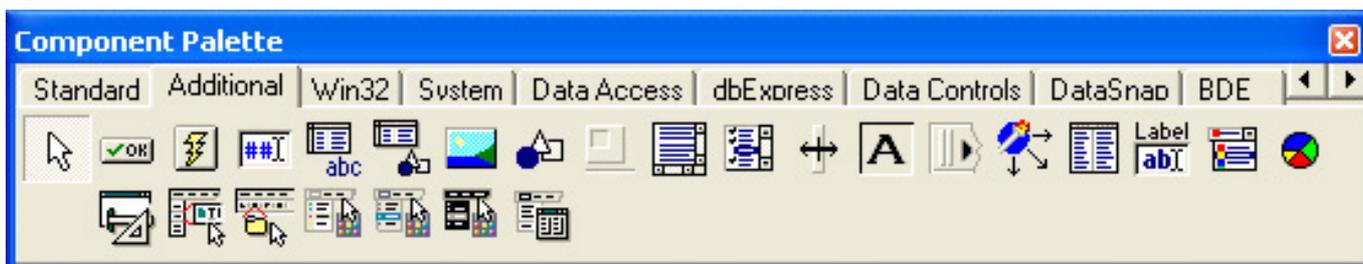


Figura 5.8 – Paleta Additional - Componentes adicionais, também muito utilizados



Figura 5.9 – Paleta Win32 - Componentes que utilizam recursos visuais do Windows

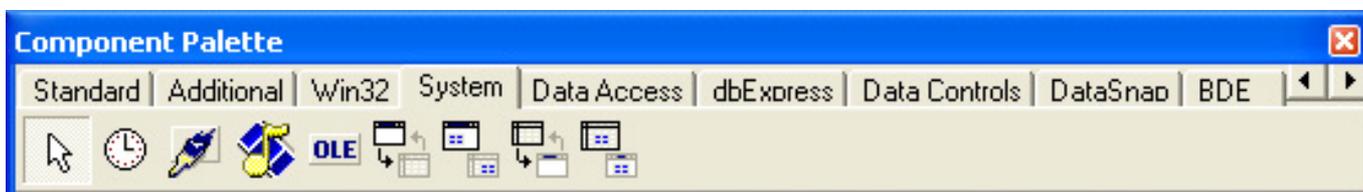


Figura 5.10 – Paleta System - Componentes que utilizam outros recursos do Windows

PRIMEIRO CONTATO

Para iniciarmos poderíamos criar a versão Delphi do famoso “Alô Mundo”, mas vamos partir para alguma coisa um pouco mais interessante e aproveitar para apresentar uma propriedade que pode ser útil no futuro. Como os programas em Windows são orientados a eventos é comum desativar opções de menus e botões, até que o usuário ative as opções que o sistema precisa para inicializar. Neste primeiro programa, vamos criar uma espécie de gangorra eletrônica. Mas antes de começar, vejamos o que está acontecendo no Delphi.

Se a tela está parecida com a figura 5.1, então não há nenhum projeto selecionado e o Delphi tomou a liberdade de criar um novo projeto para você, chamando-o de Project1. Um projeto é a coleção de arquivos necessários para compilar o sistema. Como toda aplicação precisa de pelo menos um form (Form1), ele também foi criado. Finalmente, todo form tem uma Unit correspondente que é mostrada no Editor de Código (Unit1).

Dica: Para intercalar entre o Form e a Unit use a tecla F12.

Não há problema em utilizar esse projeto inicial que é oferecido pelo Delphi, mas é uma boa idéia renomear e salvar o projeto o quanto antes. Se você não escolher outro diretório, o projeto será salvo no diretório do Delphi e não é aconselhável salvar todos os seus projetos no mesmo diretório. Portanto vamos começar este programa criando um novo diretório para ele. Após isso, vamos seguir os seguintes passos:

1. Selecione a página Standard na paleta de componentes e clique no componente Button. A seguir clique no form. Um botão deve aparecer no form. Coloque mais dois botões.
2. Alinhe na horizontal os dois primeiros e coloque o terceiro logo abaixo dos dois. Se quiser um alinhamento exato, use a opção Edit/Align... do menu. Para marcar mais de um componente, deixe o Shift pressionado enquanto clica os componentes. Brinque um pouco com o tamanho dos botões e do form.
3. Clique no primeiro botão para selecioná-lo, a seguir procure no Object Inspector a propriedade Caption. Mude a string para “ON”. Para o segundo botão a string é “OFF” e para o terceiro “Close”. Para o segundo botão, mude a propriedade Enabled para False.
4. Neste ponto é uma boa idéia renomear e salvar o projeto. Escolha a opção File/Save Project As...Selecione o diretório criado para o projeto e use o nome Main.pas para a unit1 e Gangorra para o projeto.
5. Clique na área pontilhada do form e em seguida ache a propriedade Caption no Object Inspector. Mude a string para “Gangorra”. Aproveite e mude a propriedade Name para “FMain”.
6. Dê um clique duplo sobre o primeiro botão. A janela de código será ativada já com a função correspondente ao evento OnClick criada e posicionada sob o cursor. Digite o seguinte código:

```
Button1.Caption := 'OFF';
Button1.Enabled := False;
Button2.Caption := 'ON';
Button2.Enabled := True;
```

7. Selecione a janela do form e faça o mesmo com o segundo botão, mas o código fica invertido:

```
Button2.Caption := 'OFF';
Button2.Enabled := False;
Button1.Caption := 'ON';
Button1.Enabled := True;
```

8. Finalmente para o terceiro botão o código é simplesmente:

```
Close;
```

9. Já podemos executar o programa pressionando F9.

O que está acontecendo: O evento OnClick ocorre sempre que clicamos o mouse sobre um componente. O Delphi se encarrega de determinar qual é o componente que deve responder ao evento e direciona o evento para ele. Neste caso, usamos este evento para ativar/desativar botões manipulando a propriedade Enabled, que determina se o componente está ativo ou não. Quando o componente está inativo, ele não responde aos eventos. Também usamos a propriedade Caption para mudar a mensagem que aparece nos botões. Finalmente usamos o método Close do form para fechar a aplicação.

PROPRIEDADES COMUNS

Propriedade	Descrição
Align	Determina o alinhamento do componente.
Canvas	Superfície de desenho, do tipo TCanvas, onde pode se desenhar a imagem do componente.
Caption	Legenda do componente (& indica tecla de atalho para alguns componentes).
Color	Cor do componente.
ComponentCount	O número de componentes possuídos.
Components	Matriz de componentes possuídos.
Ctl3D	Define a aparência 3D do componente.
Enabled	Define se o componente está ativo, se pode ser usado.
Font	Fonte utilizada no componente.
Height	Altura.
HelpContext	Número utilizado para chamar o Help on-line.
Hint	String utilizada em dicas instantâneas.
Left	Posição esquerda.
Name	Nome do componente. O Delphi nomeia automaticamente todos os componentes que são incluídos no form (inclusive o próprio form). Usa o nome da classe do componente mais um número seqüencial. O nome atribuído pelo Delphi pode ser mantido, mas é aconselhável renomear os componentes que serão referidos no programa. Por exemplo, no programa da Gangorra, Button1 e Button2 deveriam ser renomeados, já que é feita referência a eles no código fonte, já para o Button3 não há necessidade por que não há referência a ele. Quando você renomeia um componente, o Delphi atualiza automaticamente todo o código gerado pelo Delphi, o que inclui o cabeçalho da Unit, os eventos do componente e as propriedades de outros componentes que fazem referência ao componente renomeado, mas não atualiza o código gerado por você. Exemplo: se renomearmos agora o Button1, o Delphi atualizará o cabeçalho da unit, o nome dos eventos de Button1, mas você terá que atualizar as referências que você fez ao Button1 com o novo nome. Aliás, esta é uma regra geral no Delphi: ele nunca modifica automaticamente o código gerado pelo programador, mesmo que esteja em comentário.
PopupMenu	Menu de contexto do componente.
ShowHint	Define se o Hint será mostrado.
TabOrder	A ordem de tabulação do componente, usada quando o usuário tecla TAB.
TabStop	Indica se o componente será selecionado quando o usuário teclar TAB.
Tag	Propriedade não utilizada pelo Delphi, que pode ser usada como propriedade personalizada.
Top	Posição superior.
Visible	Define se o componente está visível.
Width	Largura.

EVENTOS

Os Eventos acontecem em resposta a uma ação do usuário ou do próprio sistema, ao programar um método de evento, devemos levar em consideração que este só será executado quando o evento acontecer. Uma das tarefas mais importantes na programação baseada em eventos é determinar quais eventos serão usados e qual a ordem desses eventos, por exemplo, quando o usuário clicar em um botão, qual evento acontecerá primeiro, OnEnter, OnMouseDown ou OnClick?

Os eventos podem ser compartilhados entre componentes, dessa forma, você pode ter um botão na barra de ferramentas que faz a mesma coisa que uma opção de menu. Para isso, basta escolher o evento na lista em vez de clicar duas vezes no Object Inspector.

Podemos também mudar os métodos de evento em código, pois os eventos também são propriedades e podem ser usados como tal. Você pode atribuir um evento de outro componente ou diretamente o nome do método,

como mostrado abaixo.

```
Button1.OnClick := Edit1.OnExit;
Button2.OnClick := Edit2.Click;
```

EVENTOS COMUNS

Evento	Descrição
OnChange	O conteúdo do componente é alterado.
OnClick	O componente é acionado.
OnDblClick	Duplo-clique no componente.
OnEnter	O componente recebe o foco.
OnExit	O componente perde o foco.
OnKeyDown	Tecla pressionada.
OnKeyPress	Uma tecla é pressionada e solta.
OnKeyUp	Tecla é solta.

MÉTODOS

Os métodos realizam ações definidas pelo componente, veja os exemplos abaixo e atente para os parâmetros passados. Note que podemos chamar os métodos de evento como qualquer outro método e que os métodos de evento pertencem ao Form, não aos componentes.

```
Edit1.Clear;
Form2.Show;
Close;
ScaleBy(110, 100);
Button1.OnClick(Sender);
Button1Click(Self);
Form2.Button1Click(Sender);
```

MÉTODOS COMUNS

Método	Descrição
Create	Cria um novo Objeto de uma Classe.
Free	Destrói um Objeto e libera a memória ocupada por ele.
Show	Torna o componente visível.
Hide	Torna o componente invisível.
SetFocus	Coloca o foco no componente.
Focused	Determina se o componente tem o foco.
BringToFront	Coloca o componente na frente dos outros.
SendToBack	Coloca o componente atrás dos outros.
ScrollBy	Move o componente.
ScaleBy	Gradua o componente em determina escala.
SetBounds	Muda a posição e o tamanho do componente.

JANELAS

Todo aplicativo Windows é composto por janelas, que são o elemento básico no desenvolvimento Delphi, sobre o qual um aplicativo é construído. O tipo TForm é usado no Delphi como classe base para todas as janelas, veja abaixo algumas propriedades, eventos e métodos dessa classe.

Propriedade	Descrição
Active	Indica se o Form está ativo.
ActiveControl	Determina o controle que receberá o foco por default.
AutoScroll	Adiciona barras de rolagem automaticamente, quando necessário.
BorderIcons	Define quais ícones de controle serão visíveis, quais botões vão aparecer na barra de título.
BorderStyle	Estilo da borda do Form.
FormStyle	Tipo de Form, normal, MDI pai, MDI filho ou sempre visível.
Icon	Ícone do Form.
Menu	Indica qual o menu do Form.
Position	Permite controlar a posição e tamanho do Form na exibição.
WindowMenu	Automatiza o item de menu Window (MDI).
WindowState	Estado do Form, maximizada, minimizada ou normal.
Evento	Descrição
OnCreate	Quando o Form é instanciado.
OnDestroy	Quando o Form é liberado da memória.
OnShow	Exatamente antes de mostrar o Form.
OnCloseQuery	É chamada para validar se o Form pode ser fechado.
OnClose	Quando o Form é fechado.
OnActivate	Quando o Form recebe o foco.
OnDeactivate	Quando o Form perde o foco.
OnResize	Quando o Form muda de tamanho.
Método	Descrição
Cascade	Organiza as Forms filhos em cascata (MDI).
Tile	Organiza as Forms filhos lado a lado (MDI).
ArrangeIcons	Organiza os ícones dos Forms Filhos minimizados (MDI).
ShowModal	Ativa o Form modal, que o usuário tem que fechar para poder continuar a usar a aplicação.
Show	Mostra o Form.
Close	Fecha o Form.
Previous	Ativa o Form anterior (MDI).
Next	Ativa a próximo Form (MDI).

COMPONENTES PADRÕES

TButton

Componente botão padrão do Windows, utilizado para executar ações.

Propriedade	Descrição
Cancel	Dispara o evento OnClick do botão quando a tecla ESC é pressionada em qualquer controle.
Default	Dispara o evento OnClick do botão quando a tecla ENTER é pressionada em qualquer controle.
ModalResult	Associa o botão a opção de fechamento de um Form modal.
Método	Descrição.
Click	Ativa o evento OnClick do botão.

TBitBtn

Botão especializado, com Bitmap.

Propriedade	Descrição
Glyph	Bitmap exibido pelo botão.
LayOut	Posição do Bitmap no Botão.
Margin	Indica o espaço entre a borda do botão e o Bitmap.
Spacing	Indica o espaço entre o Bitmap e o texto do botão.
Kind	Seleciona um tipo padrão para o botão, mudando várias propriedades, como Glyph e ModalResult.

TSpeedButton

Botão com Bitmap, normalmente utilizado em barras de ferramentas.

Propriedade	Descrição
Down	Estado do botão (Pressionado ou não).
GroupIndex	Indica quais botões pertencerão ao mesmo grupo.
AllowAllUp	Permite que todos os botões de um grupo possam ficar não pressionados.
Flat	Define se a borda do botão deve aparecer apenas quando ele for apontado.

TLabel

Utilizado para exibir rótulos

Propriedade	Descrição
Alignment	Alinhamento do texto no componente.
AutoSize	Define se o tamanho do componente será automaticamente ajustado ao tamanho do Caption.
WordWrap	Retorno automático de linha.
Transparent	Define se o componente será transparente.
FocusControl	Componente que receberá o foco quando a tecla de atalho do Caption (&) for pressionada.
ShowAccelChar	Indica se o caractere & será usado para definir tecla de atalho.

TEdit

Utilizado para entrada de texto em uma única linha.

Propriedade	Descrição
Text	Texto do componente.
AutoSelect	Indica se o texto será ou não selecionado quando o componente receber o foco.
MaxLength	Número máximo de caracteres permitidos.
CharCase	Define se as letras aparecerão em maiúsculo, minúsculo ou normal.
PasswordChar	Caractere utilizado para esconder o texto digitado (Senhas).
ReadOnly	Define se será permitido alterar o texto.
Método	Descrição
Clear	Limpa o conteúdo do componente.
ClearSelection	Limpa o texto selecionado no componente.

TMaskEdit

Permite entrada de dados texto em uma linha, utilizando uma máscara de edição. Possui todas as propriedades do componente TEdit.

Propriedade	Descrição
EditMask	Máscara de edição.

Máscaras

Uma máscara é composta por três partes, a primeira parte é a máscara propriamente dita, a segunda parte indica se os caracteres literais serão salvos e a terceira parte indica qual o caractere utilizado para representar os espaços a serem digitados no texto.

Estes são os caracteres especiais que podem compor a máscara de edição:

Caractere	Descrição
!	Espaços em branco não serão considerados no texto
>	Todos os caracteres seguintes serão maiúsculos até que apareça o caractere <
<	Todos os caracteres seguintes serão minúsculos até que apareça o caractere >
\	Indica um caractere literal
	Somente caractere alfabético
L	Obrigatoriamente um caractere alfabético
a	Somente caractere alfanumérico
A	Obrigatoriamente caractere alfanumérico
9	Somente caractere numérico
0	Obrigatoriamente caractere numérico
c	Permite um caractere
C	Obrigatoriamente um caractere
#	Permite um caractere numérico ou sinal de mais ou de menos, mas não os requer.
:	Separador de horas, minutos e segundos
/	Separador de dias, meses e anos

TMemo

Permite entrada de dados texto em múltiplas linhas. Contém propriedades e métodos do TEdit.

Propriedade	Descrição
Lines	Propriedade do tipo TStrings que armazena as linhas de texto do componente.
WantReturns	Define se a tecla ENTER será tratada como quebra de linha.
WantTabs	Define se a tecla TAB será tratada como espaço de tabulação.
ScrollBar	Define as barras de rolagem.

TStrings

Muitos componentes, como o TMemo, possuem propriedades do Tipo TStrings, essa classe permite armazenar e manipular uma lista de Strings. Toda propriedade do tipo TStrings permite acesso indexado aos itens da lista.

Propriedade	Descrição
Count	Número de strings.
Text	Conteúdo do memo na Forma de uma única string.
Método	Descrição
Add	Adiciona uma nova string no final da lista.
Insert	Inserir uma nova string numa posição especificada.
Move	Mover uma string de um lugar para outro.
Delete	Apaga uma string.
Clear	Apaga toda a lista.
IndexOf	Retorna o índice do item e - 1 caso não encontre.
LoadFromFile	Carrega texto de um arquivo.
SaveToFile	Salva texto para um arquivo.

TCheckBox

Propriedade	Descrição
AllowGrayed	Determina se o checkbox terá três possibilidades de estado.
Checked	Determina se o checkbox está marcado.
State	Estado atual do checkbox .

TRadioButton

Usado em grupo, pode ser utilizado para obter informações lógicas mutuamente exclusivas, mas é recomendado usar o RadioGroup em vez de RadioButtons.

TRadioGroup

Componente que agrupa e controla RadioButtons automaticamente.

Propriedade	Descrição
Columns	Número de colunas de RadioButtons.
Items	Lista de strings com os itens do RadioGroup, cada item da lista representa um RadioButton.
ItemIndex	Item selecionado, iniciando em 0.

TPanel

Componente Container utilizado para agrupar componentes em um panel.

Propriedade	Descrição
BevelInner	Estilo da moldura interna do panel.
BevelOuter	Estilo da moldura externa do panel.
BevelWidth	Largura das molduras.
BorderStyle	Estilo da Borda.
BorderWidth	Largura da borda, distância entre as molduras interna e externa.

TScrollBar

Container com barras de rolagem automáticas.

TGroupBox

Componente container com um título e borda 3D.

TBevel

Moldura ou linha com aparência 3D.

Propriedade	Descrição
Shape	Tipo de moldura a ser desenhada.
Style	Define alto ou baixo relevo para a linha.

TListBox

Utilizado para exibir opções em uma lista.

Propriedade	Descrição
Columns	Número de colunas de texto da lista.
MultiSelect	Define se será permitida a seleção de múltiplos itens.
ExtendedSelect	Define se a seleção poderá ser estendida pelo uso das teclas Shift e Ctrl.
IntegralHeight	Define se os itens poderão aparecer parcialmente ou somente por completo.
Items	Lista de strings com os itens da lista.
ItemIndex	Índice do item selecionado, começando em 0.
Selected	De acordo com o índice indica se um item em particular está selecionado.
SelCount	Indica quantos itens estão selecionado.
Sorted	Define se os itens aparecerão ordenados.

TComboBox

Caixa combinada com lista suspensa.

Propriedade	Descrição
Items	Lista de strings com os itens da lista.
DropDownCount	Número de itens visíveis da lista suspensa.
Style	Estilo do ComboBox, os principais estilos são csDropDown, csDropDownList, csSimple.

TImage

Componente usado para exibir figuras.

Propriedade	Descrição
Center	Determina de a figura será centralizada no componente.
Picture	Figura a exibida, pode ser BMP, ICO, WMF ou EMF.
Stretch	Define se o tamanho da figura deve ser ajustado ao do componente

TPicture

Classe usada para guardar ícones, Bitmaps, meta arquivos do Windows ou gráficos definidos pelo usuário.

Método	Descrição
LoadFromFile	Carrega figura de um arquivo.
SaveToFile	Salva figura para um arquivo.

TPageControl

Usado para criar controles com múltiplas páginas, que podem ser manipuladas, em tempo de projeto, através do menu de contexto. Cada página criada é um objeto do tipo TTabSheet.

Propriedade	Descrição
ActivePage	Página ativa.
MultiLine	Define múltiplas linhas de guias de páginas.
TabHeight	Altura das guias.
TabWidth	Largura das guias
Evento	Descrição
OnChange	Após uma mudança de página.
OnChanging	Permite a validação de uma mudança de página.
Método	Descrição
FindNextPage	Retorna a próxima página.
SelectNextPage	Seleciona a próxima página.

TTabSheet

Página de um PageControl.

Propriedade	Descrição
PageIndex	Ordem da página.
TabVisible	Define se a aba da página é visível.

TShape

Gráfico de uma Forma geométrica.

Propriedade	Descrição
Brush	Preenchimento da figura, objeto do tipo TBrush.
Pen	Tipo da linha, objeto do tipo TPen.
Shape	Forma geométrica.

TTimer

Permite a execução de um evento a cada intervalo de tempo.

Propriedade	Descrição
Interval	Tempo em milisegundos quando o componente irá disparar o evento OnTimer.
Evento	Descrição
OnTimer	Chamado a cada ciclo de tempo determinado em Interval.

TStatusBar

Utilizado para criar barras de status para exibir informações.

Propriedade	Descrição
SimplePanel	Indica se haverá apenas um panel.
SimpleText	Texto exibido caso SimplePanel seja True.
SizeGrip	Define se a alça de redimensionamento padrão deve ser mostrada.
Panels	Propriedade do tipo TStatusPanels, com os painéis do StatusBar.

TStatusPanels

Lista de panels de um StatusBar.

Propriedade	Descrição
Count	Número de panels.
Items	Lista de panels, cada panel é um objeto do tipo TStatusPanel.
Método	Descrição
Add	Adiciona um novo panel à lista.

Caixas de Diálogo

Grupo de caixas de diálogo comuns do Windows.

Método	Descrição
Execute	Mostra a caixa de diálogo e retorna True caso o usuário clique em Ok.

TOpenDialog / TSaveDialog

Caixas de diálogo para abrir e salvar arquivos.

Propriedade	Descrição
FileName	Nome do arquivo.
DefaultExt	Extensão padrão para os arquivos.
Filter	Filtro, com os tipos de arquivos que serão abertos ou salvos.
FilterIndex	Índice do filtro default.
InitialDir	Pasta inicial.
Title	Título da janela.
Options	Define características gerais do diálogo.

TFontDialog

Caixa de diálogo de escolha de fonte.

Propriedade	Descrição
Device	Define se deve utilizar fontes para tela, impressora ou ambos.
MinFontSize	Tamanho mínimo da fonte.
MaxFontSize	Tamanho máximo da fonte.
Options	Define características das fontes
Evento	Descrição
OnApply	Ocorre após o usuário pressionar o botão Aplicar, antes da janela fechar.

MENUS

No Delphi os menus serão desenhados no Menu Designer, que pode ser acessado no menu de contexto de qualquer componente de menu.

TMainMenu

Menu principal de um Form.

Propriedade	Descrição
Items	Itens de menu, essa propriedade guarda todas as alterações feitas no Menu Designer.

TPopupMenu

Menu de contexto de um componente. Cada componente tem uma propriedade PopUpMenu, que indica seu menu de contexto.

TMenuItem

Item de menu.

Propriedade	Descrição
Checked	Indica se o item está marcado ou não.
GroupIndex	Índice do grupo do item, semelhante ao SpeedButton.
RadioGroup	Indica se o item pode ser mutuamente exclusivo com outros itens do mesmo grupo.
ShortCut	Tecla de atalho do item.

Para concluir essa lição vamos criar mais um pequeno programa. Vamos criar uma pequena calculadora, como mostra a figura 5.18.

Para isso selecione a opção File/New/ Application. Clique em cima do novo formulário e altere as seguintes propriedades:

Propriedade	Valor
Caption	Calculadora
Name	FmCalculadora
Height	200
Width	150
BorderStyle	bsDialog

Agora insira 3 Edits. Altere as seguintes propriedades:

Componente	Propriedade	Valor
Edit1	Left	11
	Top	8
Edit2	Left	11
	Top	40
Edit3	Left	11
	Top	104

Agora insira 5 Buttons e altere as seguintes propriedades:

Componente	Propriedade	Valor
Button1	Left	11
	Top	72
	Width	25
	Font	Arial – 14 – Negrito
	Name	btnSoma
	Caption	+
Button2	Left	43
	Top	72
	Width	25
	Font	Arial – 14 – Negrito
	Name	btnSubtracao
	Caption	-
Button3	Left	75
	Top	72
	Width	25
	Font	Arial – 14 – Negrito
	Name	btnMultiplica
	Caption	x (xis minúsculo)
Button4	Left	104
	Top	72
	Width	25
	Font	Arial – 14 – Negrito
	Name	btnDivide
	Caption	/
Button5	Left	11
	Top	136
	Width	121
	Font	Arial – 14 – Negrito
	Name	btnLimpa
	Caption	Limpar
	Height	33

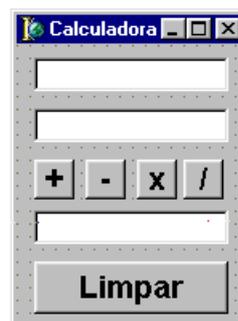


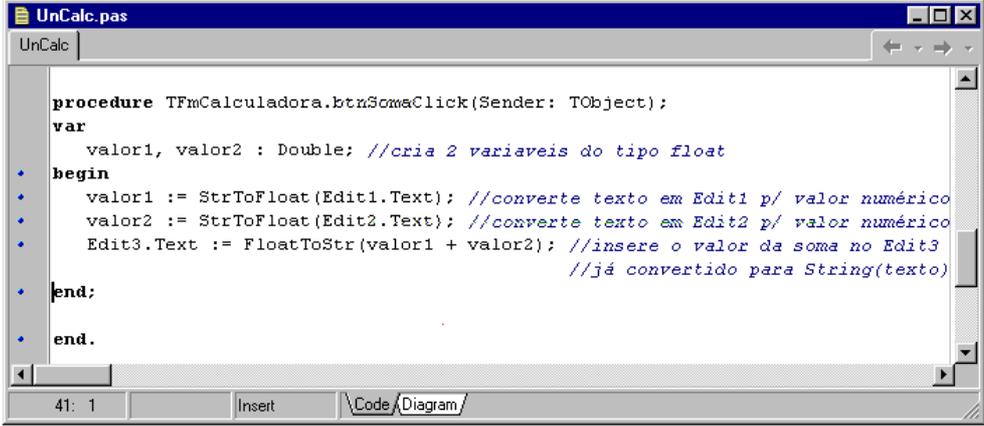
Fig. 5.18 – Proj. Calculadora

Além disso, selecione os 3 Edits e apague o conteúdo da propriedade Text.

Clique em File/Save All. Salve a Unit como UnCalc.pas e o projeto como Calculadora.dpr.

Passemos para a parte mais interessante: O código. Para que essa simples calculadora funcione precisamos utilizar algum código relacionado com o evento OnClick nos botões que inserimos. Vamos pôr em prática o que aprendemos na lição sobre Object Pascal.

Dê um duplo clique no btnSoma para ativar o evento OnClick do mesmo. Insira o código conforme mostra a Figura 5.19.



```

UnCalc.pas
UnCalc

procedure TFMCalculadora.btnSomaClick(Sender: TObject);
var
  valor1, valor2 : Double; //cria 2 variaveis do tipo float
begin
  valor1 := StrToFloat(Edit1.Text); //converte texto em Edit1 p/ valor numérico
  valor2 := StrToFloat(Edit2.Text); //converte texto em Edit2 p/ valor numérico
  Edit3.Text := FloatToStr(valor1 + valor2); //insere o valor da soma no Edit3
  //já convertido para String(texto)
end;
end.
41: 1
Insert
\Code\Diagram\

```

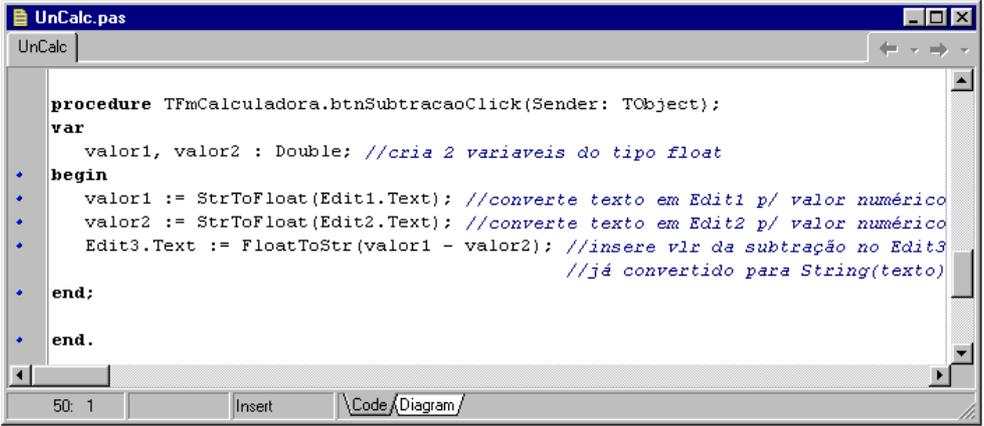
```

Var
  Valor1, Valor2:Double;
Begin
  Valor1 :=
  StrToFloat(Edit1.Text);
  Valor1 :=
  StrToFloat(Edit2.Text);
  Edit3.Text := FloatToStr(Valor1
+ Valor2);
End;

```

Fig. 5.19 – Código do botão btnSoma

Dê um duplo clique no btnSubtracao para ativar o evento OnClick do mesmo. Insira o código conforme mostra a Figura 5.20.



```

UnCalc.pas
UnCalc

procedure TFMCalculadora.btnSubtracaoClick(Sender: TObject);
var
  valor1, valor2 : Double; //cria 2 variaveis do tipo float
begin
  valor1 := StrToFloat(Edit1.Text); //converte texto em Edit1 p/ valor numérico
  valor2 := StrToFloat(Edit2.Text); //converte texto em Edit2 p/ valor numérico
  Edit3.Text := FloatToStr(valor1 - valor2); //insere vlr da subtração no Edit3
  //já convertido para String(texto)
end;
end.
50: 1
Insert
\Code\Diagram\

```

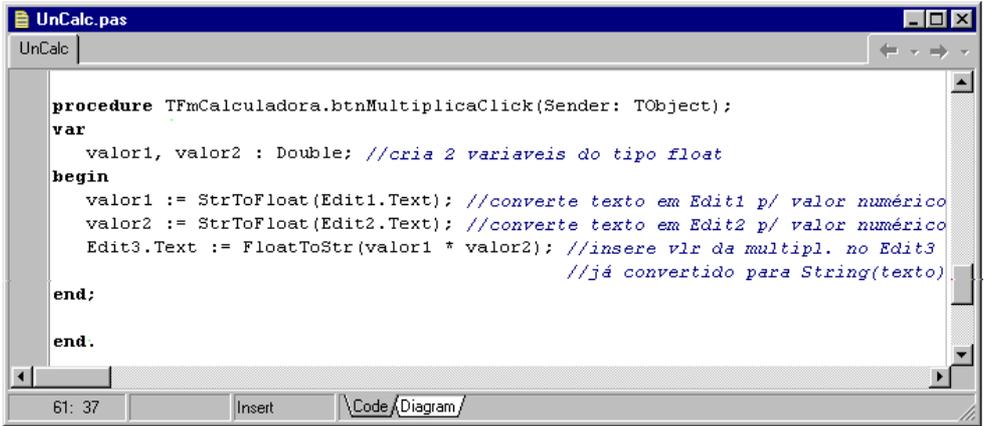
```

Var
  Valor1, Valor2:Double;
Begin
  Valor1 :=
  StrToFloat(Edit1.Text);
  Valor1 :=
  StrToFloat(Edit2.Text);
  Edit3.Text := FloatToStr(Valor1
- Valor2);
End;

```

Fig. 5.20 – Código do botão btnSubtração

Dê um duplo clique no btnMultiplica para ativar o evento OnClick do mesmo. Insira o código conforme mostra a Figura 5.21.



```

UnCalc.pas
UnCalc

procedure TFMCalculadora.btnMultiplicaClick(Sender: TObject);
var
  valor1, valor2 : Double; //cria 2 variaveis do tipo float
begin
  valor1 := StrToFloat(Edit1.Text); //converte texto em Edit1 p/ valor numérico
  valor2 := StrToFloat(Edit2.Text); //converte texto em Edit2 p/ valor numérico
  Edit3.Text := FloatToStr(valor1 * valor2); //insere vlr da multipl. no Edit3
  //já convertido para String(texto)
end;
end.
61: 37
Insert
\Code\Diagram\

```

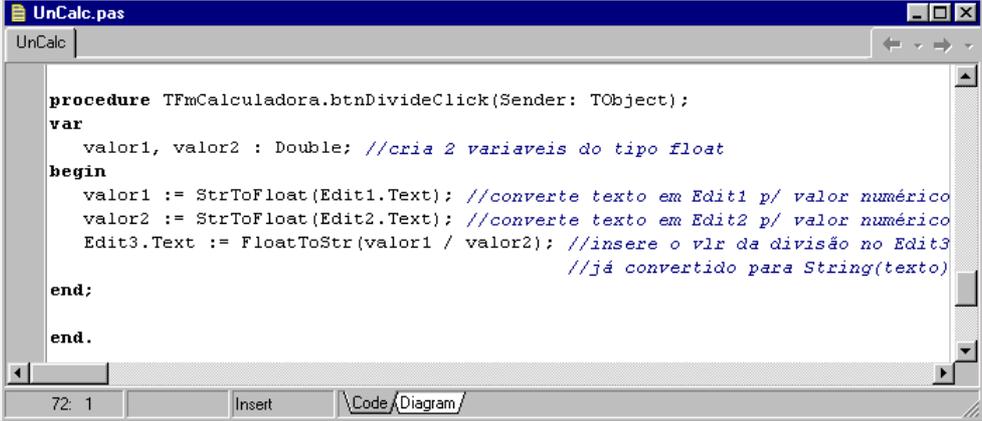
```

Var
  Valor1, Valor2:Double;
Begin
  Valor1 :=
  StrToFloat(Edit1.Text);
  Valor1 :=
  StrToFloat(Edit2.Text);
  Edit3.Text := FloatToStr(Valor1
* Valor2);
End;

```

Fig. 5.21 – Código do botão btnMultiplica

Dê um duplo clique no btnDivide para ativar o evento OnClick do mesmo. Insira o código conforme mostra a Figura 5.22.



```

UnCalc.pas
UnCalc

procedure TfmCalculadora.btnDivideClick(Sender: TObject);
var
  valor1, valor2 : Double; //cria 2 variaveis do tipo float
begin
  valor1 := StrToFloat(Edit1.Text); //converte texto em Edit1 p/ valor numérico
  valor2 := StrToFloat(Edit2.Text); //converte texto em Edit2 p/ valor numérico
  Edit3.Text := FloatToStr(valor1 / valor2); //insere o vlr da divisão no Edit3
  //já convertido para String(texto)
end;
end.
72: 1
Insert
Code/Diagram

```

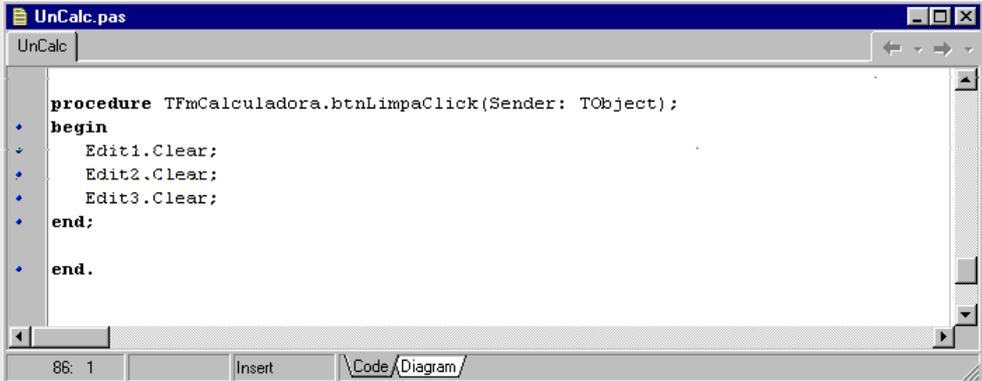
```

Var
  Valor1, Valor2:Double;
Begin
  Valor1 :=
  StrToFloat(Edit1.Text);
  Valor1 :=
  StrToFloat(Edit2.Text);
  Edit3.Text := FloatToStr(Valor1
/ Valor2);
End;

```

Fig. 5.22 – Código do botão btnDivide

Dê um duplo clique no btnLimpa para ativar o evento OnClick do mesmo. Insira o código conforme mostra a Figura 5.23.



```

UnCalc.pas
UnCalc

procedure TfmCalculadora.btnLimpaClick(Sender: TObject);
begin
  Edit1.Clear;
  Edit2.Clear;
  Edit3.Clear;
end;
end.
86: 1
Insert
Code/Diagram

```

```

Edit1.Clear;
Edit2.Clear;
Edit3.Clear;

```

Fig. 5.23 – Código do botão btnLimpa

É evidente que esse programa é bem simples e contém diversos “furos”. Por exemplo, insira uma letra no Edit1 e um número no Edit2, após isso clique no botão de soma. Aparecerá um erro, afinal você não pode transformar uma letra em um número. Tente também inserir 10 no Edit1 e 0 no Edit2. Clique no botão de divisão. Outro erro, pois não se pode dividir por zero. Você teria que prever todas essas possibilidades e criar um código compatível para a solução de cada problema. No entanto, o exemplo proposto alcançou o objetivo.

EXERCÍCIOS

- 01) Crie um Form que mostre a hora atualizada a cada segundo.
- 02) Crie um programa igual à calculadora padrão do Windows.

Veja o Conteúdo Exclusivo do Curso Delphi Avançado



1 CD - 130 Minutos

- * Compreender o modelo Entidade-Relacionamento e Lógico-Relacional;
- * Saber definir e identificar: Entidades, Relacionamentos, Atributos e tudo mais que tenha a ver com o Modelo Entidade-Relacionamento;
- * Conhecer os tipos de relacionamentos: Condicionais e Incondicionais;
- * Saber os graus de relacionamento: Um-Para-Um, Um-Para-Muitos, Muitos-Para-Muitos;
- * Compreender uma realidade e a partir dessa compreensão criar um modelo de dados e logo depois o banco de dados em si;
- * Dominar o conceito de chaves, índices e integridade referencial;
- * Saber normalizar um modelo através das Formas Normais (1FN a 5FN);
- * Usar programas para a criação do modelo de dados;
- * Utilização do DBDesigner.



1 CD - 159 Minutos

- * Definição e História da Linguagem SQL;
- * Entendendo as partes da Linguagem: DDL, DML, DCL e Transactions Control;
- * Instrução Select (alias, order by, like, asc, desc, group by, having, where, not, in, exists, sum, max, min, avg, count, between, distinct, etc);
- * Instrução Insert Into;
- * Instrução Update;
- * Instrução Delete;
- * Consultas Encadeadas (SubQueries);
- * Junções (inner join, left join, right join e full outer join);
- * Criação, alteração e deleção de tabelas e índices;
- * Controle de Transações;
- * Diversos exemplos feitos no programa SQL Explorer;
- * Criação de um sistema no Delphi implementando os comandos SQL juntamente com o componente Query. Cadastro de Clientes, Cadastro de

Usuarios e Formulário Mestre Detalhe com duas tabelas (CVenda e DVenda).



2 CDs - 340 Minutos

- * Introdução: Definição, História, Características, Licenças e Versões;
- * Instalação da versão 3.23 no Windows 98, tipos de arquivos, estrutura de diretórios do MySQL, uso do Console (prompt), etc;
- * Utilização da ferramenta gráfica MySQLFront 2.3, criação de bancos de dados, tabelas, explicação dos tipos de campos, importação de dados do paradox, atributos especiais de campos do MySQL, uso do Manual de Referência, uso das funções do MySQL;
- * Instalação da Versão 4 do MySQL no Windows XP, uso do MySQLFront 3.2, importação de arquivos do Excel, Exportação de dados para o formato HTML, etc;
- * Instalação do MySQL no Linux Slacware, uso do console no Linux, uso da ferramenta gráfica PHPMyAdmin;
- * Instalação e configuração da versão 5 no Windows XP;
- * Uso da ferramenta gráfica DBManager, tipos de tabelas, foreign keys, constraints, views, backup e restauração de dados, controle de transações;

- * Uso da ferramenta gráfica MySQL Administrator, controle de usuários, backup e restauração de dados, replicação;
- * Uso da ferramenta gráfica MySQL Query Browser;
- * Uso da ferramenta gráfica MySQL Migration Toolkit, migração de um banco de dados do MS-Access;
- * DBDesigner - Sincronização de dados e engenharia reversa;
- * Apresentação dos sites oficiais do MySQL - Internacional e Nacional;
- * Stored Procedures e Triggers.

Curso Delphi Avançado - 15 Módulos (19 CDs)



2 CDs - 254 Minutos

- * Introdução: Definição, História, Características, Licenças e Especificações Firebird;
- * Comparativo das arquiteturas Classic x SuperServer;
- * Descrição dos tipos de dados suportados pelo Firebird;
- * Dialetos;
- * Tipos de Transações;
- * Instalação passo a passo da versão 1.5 do Firebird no Windows;
- * Explicação da estrutura de diretórios;
- * Serviços executados no Windows e opções no Painel de Controle;
- * Utilização das seguintes ferramentas gráficas: IBExpert, IBEasy+, FlameRobin, IBAccess, SQLHammer, DBak Win, IB First AID, Interbase Grant Manager;
- * Criação de tabelas, triggers, generatos, domains, views, queries;
- * Controle de acesso para usuários;
- * Backup do BD;

- * Constraints: Check, PK, FK, Unique;
- * Corrigindo BDs corrompidos;
- * Criação de um projeto de BD;
- * Utilização do ISQL (console);
- * Criação de triggers para controle de estoque;
- * Exceptions;
- * Criação de Stored Procedures;
- * UDFs - Conceito; Criação de uma UDF no Delphi e uso no Firebird;
- * Corrupção de Dados: Conhecendo, evitando, corrigindo;
- * Utilização do Gfix (console);
- * Comparativo: MySQL x Firebird;
- * Apresentação dos sites importantes internacionais e nacionais;
- * Instalação do Firebird Cliente numa máquina virtual Windows 98;
- * Conexão com o banco de dados através da rede;
- * Apresentação da documentação.



1 CD - 116 Minutos

- * Definição e estrutura das empresas;
- * Níveis de decisão das empresas;
- * O Analista de Sistemas;
- * Engenharia de Software;
- * Ciclo de vida de um sistema;
- * Coleta de Informações: Entrevistas e pesquisas;
- * Entrevistas: planejamento, preparação, comportamento, linguagem, fatos x opiniões, desejos dos usuários, observações gerais;
- * Contratos e lei: análise das principais cláusulas de um contrato (identificação das partes, documentação, prazo de entrega, códigos fontes, confiabilidade);
- * Lei do Software - comentários sobre os principais artigos;
- * O que é melhor: autônomo ou empresa;
- * Estudo de caso: Sistema para Hotel - apresentação de um estudo de caso mostrando desde o início um problema enfrentado por um hotel e a

solução adotada por um analista de sistemas. Neste estudo são mostrados e comentados a proposta e o contrato utilizado pelo analista;

- * Apresentação de recibos: simples e RPA;
- * Vantagens do aluguel de software;
- * A carreira do consultor independente;
- * Comentários acerca do Guia do Empreendedor e do manual A Pequena Empresa e o Novo Código Civil;
- * Prosperidade - Mapas Mentais - explanação a apresentação do programa freemind.

Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 208 Minutos

- * Compreensão da realidade do SisCom;
- * Criação das tabelas (Produto - Cliente - Fornecedor - Funcionario - Banco - Departamento - Cartao - Unidade - Plano_Conta - Tipo_Pgto - CFOP - Pagamento - Recebimento - C_NFE - D_NFE - C_Venda - D_Venda - C_Requisicao - D_Requisicao - C_Cotacao - D_Cotacao - C_Pedido - D_Pedido - Configuracao - Nivel_Acesso);
- * Comentário detalhado sobre cada tabela e seus campos;
- * Apresentação do DER do SisCom e comentários sobre cada relacionamento;
- * Criação das tabelas e dos relacionamentos no DBDesigner;
- * Sincronização do DBDesigner com o MySQL;
- * Migrando o BD do MySQL para o Firebird;
- * Conectando o DBDesigner com o Firebird através de ODBC;
- * Conectando o DBDesigner com o Firebird diretamente;
- * Fazendo Engenharia Reversa de tabelas do Firebird no DBDesigner;

- * Visualizando a estrutura de arquivos do DBDesigner para fazê-lo conectar-se ao Firebird;
- * Visualizando o Código Fonte do DBDesigner dentro do Delphi;
- * Criação passo a passo do BD no Firebird utilizando o IBExpert;
- * Padronização de campos e tabelas;
- * Erros encontrados por falta de padronização.



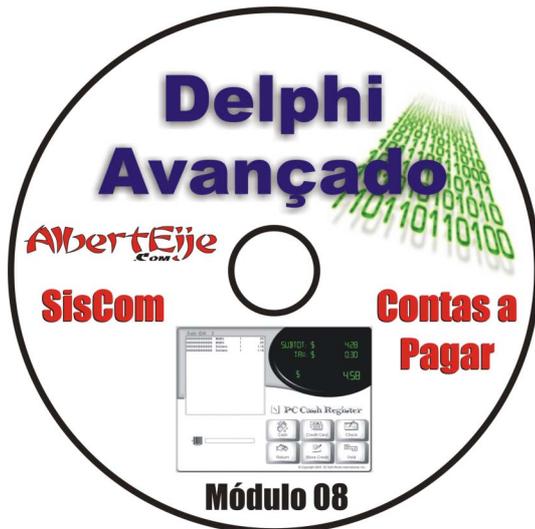
2 CDs - 416 Minutos

- * Apresentação da Suite de Componentes ZeosDBO;
- * Vantagens de usar o ZeosDBO;
- * Instalação do ZeosDBO;
- * Explicação sobre cada componente do ZeosDBO;
- * Comparação dos componentes do ZeosDBO com os seus equivalentes no BDE;
- * Diferença entre sistemas MDI e SDI e explicação sobre qual o melhor;
- * Início do desenvolvimento do sistema SisCom;
- * Criação do Formulário Principal: Menu, Barra de Ferramentas, Barra de Status. Todos os hints do sistema serão enviados para a barra de status, possibilitando uma ajuda online através de dicas rápidas para os usuários;
- * Carregando um logotipo para a tela principal do sistema;
- * Uso de Application.MessageBox;
- * Componentes de acesso ao banco em DataModule;
- * Criação dos cadastros utilizando ZTable (equivalente a TTable). Para

quem vem do paradox ficará muito fácil desenvolver dessa forma;

- * Criação dos cadastros utilizando ZQuery (Equivalente a TQuery). Utilização de Edits Simples no lugar dos DBEdits, com procedimentos específicos para o controle de edição e gravação dos campos;
- * Utilização de comandos dos tipos TTable (Append, Edit, Post, Delete) e cadastros usando apenas SQL puro (Insert Into, Update Set, Delete From);
- * O sistema roda AUTOMATICAMENTE em Firebird e MySQL. O usuário escolhe qual banco usar. Não existe alteração nenhuma no código;
- * Utilização dos Generators (firebird) para autonumerar campos no sistema;
- * Implementação de uma rotina onde o usuário procura por qualquer informação em qualquer campo do sistema, bastando para isso clicar no título do DBGrid;
- * Criação do nível de acesso para controle de acessos dos usuários aos módulos do sistema;
- * Verificação e análise de várias mensagens de erro;
- * Implementação de um controle de auditoria, onde o sistema armazena qual o usuário que inseriu ou alterou registros, juntamente com a data e hora.

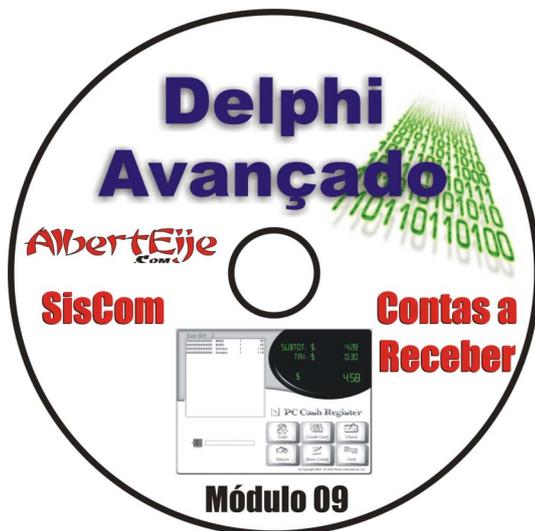
Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 236 Minutos

- * Introdução a tecnologia dbExpress;
- * Vantagens e desvantagens do dbExpress;
- * Comparativo dbExpress x BDE;
- * Apresentação de todos os componentes da paleta dbExpress;
- * Utilização do SQLQuery + DataSetProvider + ClientDataSet para criar o Contas a Pagar;
- * Desenvolvimento do Contas a Pagar passo a passo;
- * Acessando Firebird e MySQL ao mesmo tempo através do SQLConnection;
- * Utilizando um método de procura ao clicar no título do DBGrid, tendo por base um período passado por 2 MaskEdits;
- * Pegando o código AutoIncrement no MySQL para o ClientDataSet;
- * Pegando o código do Generator no Firebird para o ClientDataSet (Stored Procedures);
- * Explicação do uso de ActionLists para otimizar o trabalho;

* O SisCom funciona automaticamente com MySQL e Firebird. Os módulos de cadastro estão utilizando a tecnologia Zeos. O módulo de Contas a Pagar está usando a tecnologia dbExpress + ClientDataSet.



1 CD - 140 Minutos

- * Desenvolvimento do Contas a Receber do SisCom utilizando a tecnologia dbExpress + ClientDataSet;
- * Implementação do conceito de reconciliação de erros no SisCom;
- * Tradução do Formulário de Reconciliação de Erros para o Português;
- * Utilização apenas do SQLConnection para realizar consultas SQL;
- * Explicação detalhada do uso do SQLDataSet (Table, Query, StoredProc);
- * Utilização de parâmetros para realizar consultas SQL;
- * Manipulação dos dados do ClientDataSet antes de serem enviados para o Banco de Dados - ainda na memória do micro cliente (utilização da propriedade Delta do ClientDataSet);
- * Criação de consultas Mestre/Detalhe. (Dados mostrados numa mesma DBGrid e posteriormente em duas DBGrids);
- * Criação de bancos de dados locais com o ClientDataSet (utilização de arquivos XML);

- * Criação de Campos Agregados;
- * Utilização do componente SimpleDataSet;
- * Utilização do componente SQLMonitor para mapear o que está ocorrendo entre a aplicação e o banco de dados (tudo que está ocorrendo em baixo nível é mostrado num Memo);
- * Implementando o Controle de Transações com o dbExpress;
- * Explicação sobre o nível de isolamento (TransIsolationLevel) das transações do dbExpress.

Após assistir às vídeo aulas do curso você estará apto para desenvolver qualquer sistema comercial. Este conteúdo você não encontra em nenhum curso presencial ou online. Conteúdo único e exclusivo.

Curso Delphi Avançado - 15 Módulos (19 CDs)



2 CDs - 447 Minutos

- * Apresentação de cada componente da paleta ADO - principais propriedades;
- * Criando uma conexão no SisCom para Firebird e MySQL (alternando entre um e outro de acordo com a escolha do usuário);
- * Instalação dos Drivers ODBC para MySQL e Firebird e criação de DSN para acesso ao banco de dados do SisCom;
- * Criação do Módulo Requisição de Compras;
- * Utilização do ADOConnection, ADOQuery, ADOTable e ADOStoredProc;
- * Análise de erros ao construir o módulo;
- * Construção do formulário para importar itens para a requisição;
- * Trabalhando com passagem de parâmetros;
- * Localizando e eliminando erros do projeto do BD;
- * Criação do Módulo Cotação de Compras;
- * Utilização do repositório de formulários do Delphi;
- * Utilização de um componente permite abrir um menu em um botão;

- * Construção do formulário para importar os itens das requisições que vão compor a cotação;
- * Construção do Módulo Confirmação da Cotação (neste formulário o usuário insere os preços e condições enviadas pelos fornecedores);
- * Construção do Módulo Pedido de Compras (neste formulário o usuário visualiza as cotações e pode fazer pedidos aos fornecedores baseado nas informações prestadas pelos mesmos. É permitido realizar vários pedidos através de uma cotação);
- * Construção do formulário para importar os itens das cotações.



1 CD - 165 Minutos

- * Construção do Sistema de Vendas utilizando os componentes da paleta ADO (ADOConnection, ADOQuery, ADOTable, ADOStoredProc);
- * Chamadas a Stored Procedures armazenadas no BD com passagem e retorno de parâmetros;
- * Importação dos dados da tabela oficial CFOP de um BD diferente;
- * Análise de alguns erros que ocorrem com ADO;
- * Usando ADO com ClientDataSet;
- * Acessando o BD sem o uso do componente ADOConnection;
- * Construindo consultas Mestre-Detalhe com ADO (3 tabelas no exemplo);
- * ADO e Controle de Transações;
- * Enviando informações para o BD sem Result Set com o componente ADOCommand.



1 CD - 188 Minutos

- * Descrição detalhada dos 13 Componentes da paleta IBX (Interbase Express);
- * Comentários sobre suas propriedades;
- * Construção do Formulário de Entrada de Notas Fiscais (IBDataBase, IBTransaction, IBDataset, IBQuery, IBUpdateSQL);
- * Passagem de parâmetros;
- * Atualização do Estoque via código feito no Delphi;
- * Atualização do Estoque via Triggers criadas no Firebird (AfterInsert, AfterUpdate, AfterDelete);
- * Construção do Formulário para Ajuste de Preços;
- * Código para ajustar preços na tabela de produtos utilizando uma taxa (aumentando e diminuindo valores).

Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 198 Minutos

- * Instalação do Quick Report no Delphi 7;
- * Descrição detalhada dos componentes da Suite Quick Report;
- * Desenvolvimento dos relatórios de cadastro do SisCom utilizando o Quick Report;
- * Criação de relatórios Mestre-Detalhe com o Quick Report;
- * Exportando relatórios para TXT e HTML através do Quick Report;
- * Descrição detalhada dos componentes da Suite Rave Reports;
- * Desenvolvimento dos relatórios de Contas a Pagar e a Receber utilizando o Rave Reports;
- * Criação de relatórios Mestre-Detalhe com o Rave Reports;
- * Utilização de imagens e códigos de barras usando o Rave Reports;
- * Exportando relatórios para PDF através do Rave Reports;
- * Apresentação do Componente PrintFast (componente que envia o relatório diretamente para a impressora - excelente para impressoras matriciais e Clippeiros de plantão);
- * Instalação do PrintFast no Delphi 7;

* Desenvolvimento dos seguintes relatórios usando o PrintFast: Requisição de Compras, Mapa Comparativo de Preços e Pedido de Compras;

* Apresentação dos Geradores de Relatórios: FreeReport e Fortes Reports.



1 CD - 168 Minutos

- * Análise de 10 Manuais em formato PDF;
- * Criação do Manual do SisCom em formato PDF;
- * Criação de um arquivo HLP (ajuda) utilizando ferramentas gratuitas;
- * Chamando a ajuda de dentro do SisCom ao pressionar a tecla F1 (ajuda sensível ao contexto);
- * Desenvolvimento de um manual HTML para o SisCom;
- * Convertendo arquivos HTML para CHM (HTML Help);
- * Convertendo arquivos CHM para 42 outros formatos (PDF, HLP, RTF, TXT, XLS, DOC e muitos outros);
- * Utilização de uma ferramenta que cria o Help e já salva em diversos formatos diferentes (HLP, CHM, PDF e outros). Ferramenta muito fácil de utilizar;
- * Chamando a ajuda CHM de dentro de uma aplicação Delphi ao pressionar F1 (ajuda sensível ao contexto);
- * Utilização de uma ferramenta que cria o arquivo de ajuda baseado nas

imagens capturadas das telas do SisCom. O usuário clica em qualquer parte das janelas capturadas e a ajuda referente àquela parte da janela é apresentada.



1 CD - 175 Minutos

- * Modificando o código do SisCom para que o mesmo funcione em rede;
- * Testando o funcionamento do SisCom em rede (máquina virtual);
- * Identificação das bibliotecas necessárias para a instalação numa máquina cliente;
- * Trabalhando com arquivos Ini (configuração do acesso aos dados através desses arquivos);
- * Desenvolvimento de um instalador personalizado no Delphi (o instalador se encarrega de copiar todos os arquivos necessários para o funcionamento do SisCom e instala o MySQL e o Firebird);
- * Utilização de ferramentas para criação de instaladores: Setup2Go, CreateInstall, InnoSetup);
- * Compactação de executáveis;
- * Criando uma biblioteca de funções e integrando a mesma ao SisCom;
- * Aperfeiçoamento da janela de Login;
- * Desenvolvimento de uma Splash Screen para o SisCom utilizando uma

imagem feita no CorelDraw e exportada para o SisCom. Utilização de uma barra de progresso.

Curso Delphi Avançado - Módulos Plus



2 CDs - 224 Minutos

- * Introdução a Orientação a Objetos
- * Histórico do Paradigma de Orientação a Objetos
- * Explicações sobre: Abstração; Objeto; Encapsulamento; Mensagem; Classe; Herança; Polimorfismo; Classificação; Associação; Agregação; Generalização; Especialização
- * Introdução a UML - Unified Modeling Language
- * Histórico da notação UML
- * Fases do desenvolvimento de um sistema em UML: Análise de requisitos; Análise; Design (projeto); Programação; Testes
- * Visões: Visão de Componentes; Visão Lógica; Visão de Use-Case; Visão de Organização; Visão de Concorrência
- * Modelos de Elementos: Classes; Objetos; Estado; Pacote; Componente; Relacionamentos (associação, generalização, dependência e refinamento)
- * Diagramas: Use-Case; Classes; Objetos; Estado; Seqüência;

Colaboração; Atividade; Componente; Execução

- * Estudos de caso: conta corrente e aplicações financeiras; locadora; sistema de matrícula; posto de gasolina; biblioteca.
- * Cada estudo de caso acima é feito em uma ferramenta diferente.

Obs: Este Módulo é pré-requisito para o Módulo Delphi OOP



1 CD - 220 Minutos

- * Introdução à Linguagem Object Pascal (Delphi Language);
- * Variáveis - declaração e exemplos; Variáveis Locais e Globais;
- * Constantes - declaração e exemplos;
- * Tipos de Dados - Inteiros, Reais, Texto, Ordinais;
- * Rotinas de conversão de tipos; TypeCasting;
- * Operadores - unários, multiplicativos, direção de bits, aditivos, relacionais;
- * Estruturas de Repetição - While, For, Repeat; Quebras de Laço (continue, break, exit, halt);
- * Tipos Definidos pelo Usuários - SubRange, Enumerações, Ponteiros, Records, Arrays, Sets;
- * Procedimentos, Funções e Métodos;
- * Passagem de parâmetros - explicações sobre passagem por valor e por referência; Uso do With;
- * Desenvolvimento de diversas rotinas para praticar os conceitos acima;
- * Explicações sobre a estrutura da Unit: Áreas: Unit, Interface,

Implementation, Var, Const, Uses, Referência Circular;

- * Sobrecarga de métodos: Overload - Exemplo de vários métodos de mesmo nome para mostrar o funcionamento;
- * Blocos protegidos - explicações sobre exceções (try, except, finally, raise);
- * Explicações sobre parâmetros formais e parâmetros reais;
- * Classes e Objetos; Desenvolvimento de diversas classes no Delphi;
- * Métodos construtores e destrutores; Métodos Getters e Setters;
- * Instanciação das classes - Desenvolvimento de uma aplicação Console para analisar o funcionamento;
- * Encapsulamento - Operadores de visibilidade (private, public, protected, published) - explicações e prática de cada um deles;
- * Herança entre classes; * Utilização de Property (propriedades);
- * Sobreposição de Métodos - Virtual/Override - Sobrescrever métodos em classes descendentes - Conceito de Polimorfismo;
- * Desenvolvimento do SisCom OO - explicações sobre o objetivo do SisCom (Sistema Comercial);
- * Exibição do DER e da UML do SisCom. Diferenças entre os dois diagramas;
- * Apresentação do SisCom desenvolvido durante o Curso Delphi Avançado;
- * Desenvolvimento passo a passo do Formulário Padrão de Cadastro. Através desse formulário, todos os outros forms de cadastro serão criados. Utilização do conceito de Herança Visual do Delphi;
- * Integração Delphi / ModelMaker; * Engenharia reversa do SisCom para o ModelMaker;
- * Alterações em modelo no ModelMaker e atualização em Tempo Real na aplicação no Delphi;
- * Explicações para desenvolvimento dos diagramas UML do SisCom no ModelMaker;

Obs: O Módulo Plus Entendendo UML é pré-requisito para este módulo

Curso Delphi Avançado - Módulos Plus



2 CDs - 339 Minutos

- * Explicações sobre ECF (Emissor de Cupom Fiscal) e TEF (Transferência Eletrônica de Fundos);
- * Instalação e utilização de Emulador de Impressora Fiscal para utilização junto com o sistema de Frente de Caixa;
- * Análise dos manuais das Impressoras Bematech, Yanco, Daruma e Zanthus;
- * Criação passo a passo da Interface do Sistema de Frente de Caixa;
- * Implementação dos comandos para utilização com a impressora fiscal: Leitura X, Redução Z, Relatórios de Memória Fiscal; Abertura e Fechamento de Cupons; Envio de Formas de Pagamento; Envio e Cancelamento de Itens; Etc;
- * Criação de um formulário de configurações onde o próprio usuário define as formas de pagamentos que vai utilizar, em quais ocasiões vai solicitar senhas, qual impressora está utilizando e o caminho onde as fotos dos produtos estão armazenadas (as fotos dos produtos são armazenadas

fora do banco de dados);

- * Criação de um formulário para localização de produtos, caso o sistema não consiga ler o código ou o produto não tenha código de barras;
- * Criação de um formulário para armazenar os dados da venda;
- * Estudo e implementação da Solução TEF;



2 CDs - 275 Minutos

- * SQL Server - Apresentação - Características;
- * Instalação passo a passo do SQL Server e Ferramentas Avançadas - Instalação do .NET Framework;
- * Apresentação da ferramenta de configuração (SQL Server Configuration Manager);
- * Apresentação e configuração do SQL Browser - ferramenta para aceitar conexões remotas;
- * Explicação e configuração dos tipos de conexões (Shared Memory, Named Pipes, TCP/IP, VIA);
- * Criação de aliases para o Banco de Dados;
- * Apresentação da ferramenta de configuração (SQL Server Surface Area Configuration);
- * Configuração e uso da ferramenta para gerenciamento de bancos de dados feitos em SQL Server (Microsoft SQL Server Management Studio Express);

- * Explicação sobre Collations; Detalhamento sobre os tipos de dados;
- * Criação e configuração de um banco de dados;
- * Criação das tabelas, índices, relacionamentos, constraints, etc do banco de dados utilizando ferramenta para modelagem;
- * Criação do campo Auto-Increment; Criação de Views atualizáveis;
- * Apresentação de ferramenta gráfica adicional para gerenciamento do banco de dados (SQL Manager 2005 for SQL Server);
- * Criação de Checks Constraints; Criação de Usuários/Roles para acesso ao banco;
- * Confecção de relatórios utilizando a ferramenta;
- * Exportando dados (Excel, Access, Word, RTF, HTML, PDF, TXT, CSV, DBF, XML);
- * Importando dados (Excel, Access, DBF, XML, TXT, CSV);
- * Explicação sobre Stored Procedures e Triggers;
- * Operações com o Banco (extração de dados, backup e restore);
- * Apresentação de ferramenta gráfica adicional para gerenciamento do banco de dados (SQL Lite);
- * Conhecendo as funções de sistema (agregação, matemáticas, data e hora, string, diversas...);
- * Desenvolvimento de uma aplicação do Delphi para acesso ao banco de dados (cadastros, vendas, etc). Utilização das tecnologias de acesso ADO e dbExpress;
- * Formulário mestre-detelhe e campos de lookup; Explicação e criação de Trigger para atualização do estoque;
- * Utilização de transações (início, cancelamento e confirmação);
- * Funcionamento da aplicação em rede através de máquina virtual W98;

Curso Delphi Avançado - Módulos Plus



1 CD - 129 Minutos

- * Introdução a Criptologia (Criptografia e Criptoanálise);
- * Utilizando na prática a Esteganografia (esconder arquivos em imagens);
- * Análise de vários componentes que implementam a Criptografia no Delphi;
- * Apresentação de vários métodos para Proteção de Programas (Nag-Screen, Período, Limitação de Registros, HardLocks, Identidade Única);
- * Implementando na prática várias técnicas para proteção de programas (Por Arquivo, Por Diretório, Por Registro, etc);
- * Desenvolvimento de um sistema que bloqueia o uso do sistema no usuário. O sistema grava as informações no registro do Windows. Pega informações da máquina do usuário (HD, BIOS) e combina tais informações com Constantes implementadas pelo programador (Phi e Pi). O desbloqueio do sistema se dá ao informar a Contra-Chave (Demonstrado passo a passo);
- * Apresentação de vários componentes para Delphi que implementam a

proteção de programas;

- * Dicas importantes para colocar em prática no momento da criação de uma proteção;
- * Apresentação de Engenharia Reversa (Descompiladores);
- * Utilização na prática de alguns descompiladores (DeDe, Source Rescuer, ResHacker);
- * Demonstração de técnica para evitar a Engenharia Reversa de programas feitos em Delphi;
- * Demonstração de como Crackear um programa feito em Delphi utilizando um Editor HexaDecimal (passo a passo);



1 CD - 229 Minutos

- * MSAccess - Apresentação;
- * Criação de Tabelas (explicação dos tipos de dados);
- * Especificações do Access - limitações do banco de dados;
- * Definindo relacionamentos e criando integridade referencial. Explicações sobre cardinalidade;
- * Criação de consultas (Visual e SQL);
- * Criação de formulários. Formulários prontos para impressão. Formulários vinculados a código em VBA. Subformulários;
- * Criação de uma consulta vinculada a dados de um formulário. Visualização dos dados de uma consulta em um subformulário;
- * Desenvolvimento de relatórios;
- * Desenvolvimento de uma página WEB que acessa e altera os dados da tabela;
- * Introdução a VBA (Visual Basic for Applications);
- * Acessando o banco de dados através do Delphi (tecnologia DAO);
- * Criação de um pequeno aplicativo no Delphi acessando o BD Access.

Utilização de Acion Lists;

- * Demonstração do Assistente de tabelas do Access que ajuda a normalizar o Banco de Dados;
- * Demonstração do Analisador de Desempenho do Access;
- * Definindo e alterando senhas no Access;
- * Acessando os relatórios desenvolvidos no Access através do Delphi: visualização em tela e envio para impressão sem que o usuário perceba que o relatório foi feito no Access;
- * Criação de índices primários e secundários;
- * Introdução ao conceito de boletos bancários;
- * Apresentação do Manual da Federação Brasileira dos Bancos;
- * Criação passo a passo de um registro de 240 posições definido no manual da Febraban;
- * Apresentação de uma solução pronta (componente open source) para a impressão de boletos bancários em Delphi, emissão do arquivo de remessa que é enviado pelas empresas aos bancos e leitura do arquivo de retorno que é enviado pelos bancos para as empresas;
- * Instalação e utilização do referido componente: impressão de boletos bancários em QuickReport (visualização personalizada e RaveReports, exportação do boleto para PDF, geração do arquivo de remessa);
- * Demonstrando a impressão de boletos bancários com dados vindos de uma tabela do banco de dados Access;
- * Orientações para migrar o componente para uma versão posterior do Delphi;