

Delphi 3

Introdução

Visão Geral do Delphi

Desde que a primeira versão do Delphi foi lançada, em 1995, esta ferramenta tem se mostrado como a melhor escolha no desenvolvimento para Windows. Numa relação com outros ambientes de programação, podemos dizer que o Delphi tem o poder do C++, e a facilidade do Visual Basic.

A principal vantagem do Delphi está na linguagem usada, Object Pascal, que é uma evolução do Pascal padrão. O Pascal surgiu no final dos anos 60 e, até hoje, é usada como uma das primeiras linguagens de programação para estudantes de computação. Em 1984, a Borland lançou o Turbo Pascal, que se firmou como o melhor compilador de Pascal do mercado e, a partir de então, passou a incluir novos recursos nesta linguagem, como Units e Objetos, até a ascensão do Windows, quando foi lançado o Turbo Pascal for Windows e, depois, o Borland Pascal, cuja linguagem é considerada a primeira versão da Object Pascal. Na sua atual versão, usada pelo Delphi, a Object Pascal é uma linguagem poderosa, sólida e respeitada, sem perder sua peculiar facilidade.

No Delphi, a criação de aplicativos começa com a montagem de componentes em janelas, como se fosse um programa gráfico, o usuário também pode utilizar componentes desenvolvidos por terceiros ou criar seus próprios componentes.

O Delphi vem com todas as ferramentas necessárias para a criação de bancos de dados dBase e Paradox, além de uma versão do Interbase, permitindo a criação de aplicativos com banco de dados sem a necessidade de aquisição de outro programa. O Delphi também tem acesso a bases de dados como Foxpro, Access, InFormix, SYBASE, Oracle, SQL Server e DB2, além de qualquer outro banco de dados para Windows compatível com ODBC.

Principais Características

- Compilador/otimizador de código mais rápido do mercado, gerando executáveis rápidos e puros, sem runtime
- Totalmente orientado a objetos e com suporte a threads e OLE Automation
- Baseado em componentes, com facilidade de criação de componentes nativos, além de controles ActiveX, inclusive com disponibilidade do código fonte dos componentes padrão
- Programação two-way, utilização de métodos visuais ou diretamente sobre o código
- Suporte a manipulação de exceções, que permite criar aplicações mais robustas e com maior segurança
- Acesso rápido e seguro a bancos de dados através do Borland Database Engine, com facilidades de manipulação
- Criação de relatórios no próprio executável, com utilização de componentes nativos
- Facilidade de upsizing para bancos de dados cliente/servidor
- Capacidade de criação de aplicações multi-tier, com objetos distribuídos
- Suporte a código in-line, em assembly
- Capacidade de criação de outros tipos de utilitários, como DLL's, Screen Saver's e aplicações CGI..
- Literatura diversificada
- Fluxo de programação baseado em eventos

Versões

- Delphi Standard, para estudantes, com poucas ferramentas de apoio
- Delphi Professional, com a maioria das ferramentas, mas sem suporte a arquitetura cliente/servidor
- Delphi Client/Server Suite, versão completa, com todas as ferramentas de apoio

Bibliografia

Borland Delphi 3 for Windows 95

Série Curso Básico & Rápido
Marcelo Leão
Axcel Books

Como Programar em Delphi 3

Frank Engo
Makron Books

Sites

<http://www.rn.senac.br/desenvolvimento>
<http://carbohyd.siobc.ras.ru/torry/>
<http://delphi32.com>
<http://rx.demo.ru/links.htm>
<http://super.sonic.net/ann/delphi>
<http://www.borland.com>

Delphi 3 Total: Aplicações para Bancos de Dados

Ronaldo Smith/Maurício Longo
Brasport

Dominando o Delphi 3

Marco Cantù
Makron Books

<http://www.chami.com/tips/>
<http://www.hyperact.com>
<http://www.intermid.com/delphi/>
<http://www.sunsite.icm.edu.pl/delphi/>
<http://www.teleport.com/~ol/dfiles.shtml>

Ambiente de Programação

Janela Principal

A janela principal do Delphi é composta pela barra de menus, barra de ferramentas e paleta de componentes. Para personalizar a barra de ferramentas, basta clicar em Properties no menu de contexto. Na paleta de componentes estão os controles usados nas aplicações, agrupados em guias por categorias.

Object Inspector

No Object Inspector podemos manipular, em tempo de projeto, as propriedades e eventos dos componentes. Você também pode selecionar um componente usando o Seletor de Objetos, no topo do Object Inspector.

Propriedades

São as características de um componente. Para mudar uma propriedade, selecione o componente no Form Designer ou no Object Selector, localize a propriedade, na guia Properties e mude o valor na coluna à direita.

A edição de propriedades pode ser simples, por lista suspensa, caixa de diálogo ou com propriedades aninhadas.

Eventos

Numa definição inicial, eventos podem ser vistos como chamadas a métodos em resposta a determinadas mensagens do Windows. Para criar um método para um evento, selecione o componente e clique duas vezes na coluna à direita do evento na guia Events do Object Inspector, o Delphi faz todas as declarações necessárias e mostra o método pronto para ser programado no Editor de Código. Para que um evento chame um método já definido, em vez de clicar duas vezes na coluna à direita do evento, você deve usar a lista suspensa.

Form Designer

O Form Designer é onde são desenhados os Forms das aplicações, com a inserção de componentes. No menu de contexto do Form, você pode clicar em View as Text para editar a descrição textual do Form e de seus componentes no Editor de Código, essas inFormações são gravadas em um arquivo binário com a extensão DFM, para voltar ao modo de exibição normal, escolha View as Form no menu de contexto do Editor de Código.

Manipulando Componentes

Incluir: Selecionar o componente na paleta e clicar no Form Designer.

Redimensionar: Clicar no componente e arrastar as alças de borda. Podemos usar SHIFT+SETAS para redimensionar o componente fora da grade de alinhamento do Form.

Mover: Arrastar o componente. Podem ser usadas também operações de recortar, copiar e colar, além de CTRL+SETAS para mover o componente para fora da grade de alinhamento do Form.

Selecionar: Segurar SHIFT para selecionar vários componentes individuais e CTRL para escolher uma área retangular do Form e selecionar todos os componentes nesta área.

Alinhamento: Para alinhar componentes selecione-os e escolha View / Alignment Palette.

Menu de Contexto: Bring To Front / Send To Back, para trazer o componente para frente ou enviar para trás, Tab Order para mudar a ordem de tabulação, além de Align To Grid, para alinhar os componentes selecionados à Grade do Form Designer.

Editor de Código

Para escrever o código, usamos o Editor de Código do Delphi. Para cada Form é criado um código, que é gravado em arquivos chamados Units, nesses arquivos é definida a classe do Form e seus métodos de eventos. Para alternar entre o Form e sua Unit podemos clicar em Toggle Form/Unit no menu View, ou no botão corresponde da Barra de Ferramentas. Para cada Form aberto é criado um Form Designer e uma nova guia no Editor de Código.

Configuração do Ambiente

Grande parte das opções de configuração do ambiente podem ser acessadas através do item Environment Options do menu Tools. A maioria das opções desse diálogo são bastante claras e através delas podemos definir, desde as opções do Form Designer, até o Editor de Código e o caminho das Livrarias. No menu Tools, podemos escolher também Configure Tools, para permitir abrir aplicações externas a partir do ambiente do Delphi, como o Image Editor e o Database Desktop.

Estrutura de Projetos

Um projeto em Delphi é dividido em módulos, chamados Units, seguindo a estrutura de arquivos descrita na tabela abaixo.

Extensão	Descrição
DPR	Arquivo de projeto, onde são indicados as Units e o código de inicialização do programa
PAS	Código fonte de uma Unit do projeto
DCU	Unit compilada
DFM	Definição visual de um Form. O código fonte está em uma Unit com o mesmo nome
DOF	Opções de configuração para o projeto
RES	Recursos do projeto, com o ícone do programa
~PA, ~DF, ~DP	Arquivos temporários

DSK	Configurações de Desktop
-----	--------------------------

Project Manager

Para ajudar no gerenciamento de projetos, podemos usar o Project Manager pelo menu View. O Project Manager lista as Units, os Forms existentes nessas Units e o path, se a Unit não estiver na pasta do projeto. Através dos botões do Project Manager você pode adicionar, excluir e visualizar Units e Forms que compõem o projeto.

Project Options

Através do item Options, do menu Project, podemos escolher diversos aspectos de um projeto.

Forms

Nessa página, podemos definir o Form principal da aplicação e a os Forms que serão criados automaticamente. Se um Form não for criado automaticamente, você terá que instanciar esse Form explicitamente.

Application

Nessa página podemos definir o título, o arquivo help e o ícone da aplicação.

Compiler

Usamos essa página para definir as opções de compilação, para o projeto atual. Essas opções irão interferir diretamente no executável gerado.

Linker

Essa página é muito pouco usada, mas somente através dela podem modificar a memória exigida por uma aplicação.

Directories/Conditionals

Aqui você pode especificar pastas de saída para os arquivos gerados na compilação do projeto e opções de compilação condicional.

Version InFormation

InFormações da versão do executável.

Packages

Nesta página você pode especificar parte do código para ser incluído em *Packages*, fora do executável, permitindo compartilhamento de componentes entre várias aplicações Delphi.

Gerenciamento de Projetos

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

File	
New	Abre um diálogo com novos itens que podem ser adicionados ao projeto
Open	Abrir projetos, pode abrir também Units, Forms e texto no editor de código
Save	Salva o arquivo aberto no editor de código
Save Project As	Salva o projeto com outro nome ou local
Use Unit	Faz com que a Unit atual possa usar outra Unit do projeto
Add to Project	Adiciona uma Unit em disco ao projeto
Remove from Project	Remove uma Unit do projeto
View	
Project Manager	Mostra o gerenciador de projeto
Project Source	Mostra o código do projeto
Object Inspector	Mostra o Object Inspector
Toggle Form/Unit	Alterna entre o Form e a Unit
Units	Mostra o código fonte de uma Unit ou do Projeto a partir de uma lista
Forms	Seleciona um Form a partir de uma lista
Project	
Compile	Compila o projeto
Options	Opções do projeto, como ícone do executável, nome da aplicação e opções de compilação
Run	
Run	Compila e executa o projeto

Ajuda

O sistema de ajuda do Delphi é a referência mais completa, seguida pelos manuais do usuário cedidos com o sistema. Se quiser ajuda sobre um componente, selecione-o e aperte F1, o mesmo pode ser feito com propriedades e eventos, no Object Inspector e comandos, no editor de código.

Biblioteca de Classes

Nomenclatura

Para nomear os componentes podemos usar uma convenção muito usada, onde as primeiras letras, minúsculas, identificam o tipo do componente e o restante identifica a função deste, assim, btnSair, seria o botão de sair.

Se a função do componente for um nome composto esse nome deve ser escrito com os primeiros nomes abreviados e com letras de caso variável, como em btnRelVendas, que seria o botão do relatório de vendas ou btnRelVenProduto, que seria o botão do relatório de vendas por produto.

Propriedades

As propriedades são características dos componentes, como foi mostrado anteriormente. Para alterar propriedades em código use a sintaxe de ponto, como mostrado abaixo.

Tipos de Propriedade

- Tipo String

```
Button1.Caption := 'Fechar';
Label1.Caption := Edit1.Text + '/' + Edit2.Text;
```

- Tipo Numérico

```
Button2.Height := Button2.Height * 2;
Width := Button1.Width + Button2.Width + 12;
```

- Tipo Enumerado

```
BorderStyle := bsDialog;
Panel1.Color := clWindow;
```

- Propriedades Aninhadas de Classe

```
Memor1.Lines.Text := 'E agora, José?';
Label1.Font.Color := clBlue;
```

- Propriedades Aninhadas de Conjunto

```
BorderIcons := [biSystemMenu, biMaximize];
Label1.Font.Style := [fsBold, fsItalic];
```

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
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 executados 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.ScrollBy(10, 10);
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 Padrão

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
l	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 TStringList 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 TStringList, essa classe permite armazenar e manipular uma lista de Strings. Toda propriedade do tipo TStringList 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	Insere uma nova string numa posição especificada
Move	Move 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

Utilizado para obter informações de checagem.

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 painel.

Propriedade	Descrição
BevelInner	Estilo da moldura interna do painel
BevelOuter	Estilo da moldura externa do painel
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 esta 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 ajustada 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
TabHeigth	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 milissegundos 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

TStatusPanel

Panel de um StatusBar.

Propriedade	Descrição
Text	Texto do panel
Width	Largura em pixels
Bevel	Moldura do panel
Alignment	Alinhamento do texto de um panel

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

Classes Não Visuais

TApplication

Todo programa tem um objeto global nomeado Application, do tipo TApplication, esse objeto representa a aplicação para o Windows.

Propriedade	Descrição
ExeName	Caminho e nome do arquivo executável
MainForm	Form principal da aplicação
Hint	Hint recebido pela aplicação
Title	Título da aplicação
HelpFile	Caminho e nome do arquivo help
Evento	Descrição
OnHint	Quando um hint é recebido pela aplicação
OnException	Quando ocorre uma exceção
OnHelp	Quando acontece uma solicitação de help
Método	Descrição

MessageBox	Apresenta um quadro de mensagem
Run	Executa a aplicação
Terminate	Finaliza a aplicação normalmente

Quadros de Mensagem

O método Application.MessageBox mostra quadros de mensagem com chamadas a funções da API do Windows. Os flags de mensagem mais usados e os valores de retorno desse método são mostrados abaixo.

Flag	Item Mostrado
MB_ABORTRETRYIGNORE	Botões de Abortar, Repetir e Ignorar
MB_ICONERROR	Ícone de erro
MB_ICONEXCLAMATION	Ícone com ponto de exclamação
MB_ICONINFORMATION	Ícone com letra i, usada para mostrar informações
MB_ICONQUESTION	Ícone de pergunta
MB_OK	Botão de Ok
MB_OKCANCEL	Botões de Ok e Cancelar
MB_RETRYCANCEL	Botões de Repetir e Cancelar
MB_SYSTEMMODAL	O Windows só poderá ser usado quando o quadro for fechado
MB_YESNO	Botões de Sim e Não
MB_YESNOCANCEL	Botões de Sim, Não e Cancelar
Valor de Retorno	Botão Escolhido
IDABORT	Abortar
IDCANCEL	Cancelar
IDIGNORE	Ignorar
IDNO	Não
IDOK	Ok
IDRETRY	Repetir
IDYES	Sim

Esses quadros são usados quando se deseja uma resposta simples do usuário, principalmente numa confirmação ou pergunta para o usuário, como o código abaixo, usado no evento OnCloseQuery do Form principal.

```
if Application.MessageBox('Deseja fechar a aplicação?', 'Sair do sistema', MB_ICONQUESTION +
    MB_YESNO) = IDNO then
    CanClose := False;
```

TScreen

O Delphi automaticamente cria a variável Screen do tipo Tscreen, essa variável guarda características do vídeo, como mostrado abaixo.

Propriedade	Descrição
ActiveForm	Form com o foco
FormCount	Número de Forms no vídeo
Cursor	Cursor do mouse
Forms	Lista dos Forms disponíveis
Fonts	Lista de Fontes de tela disponíveis
PixelsPerInch	Número de pixels por polegada da Fonte usada pelo sistema
Height	Altura da tela em pixels
Width	Largura da tela em pixels

TPrinter

Na Unit Printers é declarado um objeto do tipo TPrinter nomeado Printer que encapsula toda a interface de impressão do Windows e pode ser usado para imprimir diretamente, sem usar componentes de relatório, como o QuickReport.

Propriedade	Descrição
Canvas	Superfície de desenho, do tipo TCanvas, onde será desenhada a imagem a ser impressa
Printers	Lista de impressoras instaladas
Orientation	Retrato ou Paisagem
PageHeight	Altura da página
PageWidth	Largura da página
PageNumber	Página atual
Método	Descrição
BeginDoc	Inicia o processo de desenho
EndDoc	Finaliza o processo de desenho e envia a imagem do Canvas para a impressora
Abort	Aborta a impressão

TCanvas

Um objeto da classe TCanvas é uma superfície de desenho, onde podem ser usados vários métodos de plotagem gráfica. Todos os controles visíveis possuem uma propriedade Canvas, do tipo TCanvas, que geralmente é usada nos Forms e no objeto Printer.

Propriedade	Descrição
Brush	Padrão de preenchimento, propriedade do tipo TBrush
Pen	Estilo de linha, propriedade do tipo TPen
Font	Fonte usada nas plotagens de texto
Método	Descrição
TextOut	Desenha texto na superfície
Ellipse	Desenha uma elipse
Polygon	Desenha um polígono
Rectangle	Desenha um retângulo

TList

Estrutura de dados polimórfica que pode gerenciar uma lista de objetos de qualquer classe e possui métodos semelhantes aos de TStringList.

TStringList

Lista de strings descendente de TStringList usada para manter listas de strings independentes de qualquer componente.

TRegistry

Interface com a API de manipulação do Registry do Windows, banco de dados de configuração do sistema.

Exercícios

1. Quais os tipos de arquivo que essenciais para um projeto Delphi.
2. Como mudar o ícone de uma aplicação em tempo de projeto?
3. Quais propriedades pode se usar para:
 - Deixar o Form sempre acima das outras janelas do sistema.
 - Mostrar as barras de rolagem quando o Form sobrepor um componente.
 - Não mostrar os botões de maximizar e minimizar.
 - Fazer com que o Form não possa ser redimensionado, mas tenha todos os ícones de borda.
 - Deixar o Form maximizado.
4. Que comandos podem ser usados para:
 - Duplicar a altura do Form.
 - Alterar o Form para 120% de seu tamanho.
 - Fazer com que um Form fique desabilitado.
 - Centralizar o Form na tela quando ele for mostrado.
 - Minimizar um Form.
5. Como mudar o título de um Form para Delphi quando o usuário clicar nesse Form?
6. Crie um Form cujas propriedades possam ser mudadas em tempo de execução, como sugestão, tente inserir nesse Form controles que definam as seguintes propriedades:
 - Estilo de borda
 - Título da janela
 - Ícones visíveis na barra de título
 - Se o Form é sempre visível
 - Se surgirão barras de rolagem automaticamente quando algum controle for sobreposto
 - Fonte do Form, que será usada como aparência base de todos os controles
7. Crie um Form que mostre a hora, atualizada a cada segundo.
8. Crie um projeto de um editor de notas, semelhante ao Bloco de Notas do Windows.
9. Crie um projeto que funcione como um visualizador de imagens e permita abrir imagens dos tipos padrão. Este projeto deve suprir as seguintes exigências:
 - Barra de menus padrão, com teclas de atalho
 - Barra de ferramentas como atalhos para todos os itens de menu, que possa ser ocultada
 - Barra de status, que também possa ser ocultada, com o nome do arquivo atual e a data do sistema
 - O usuário deve escolher se a imagem vai se ajustar automaticamente à área livre da janela principal
10. Como fazer para que seja mostrado um quadro de *status* durante a compilação de um programa?