

# DELPHI





# Parte I

Este artigo é parte de uma série onde será apresentada uma poderosa ferramenta bastante útil para o profissional da Eletrônica: **o ambiente de programação Delphi**. Pretende-se com isso desmistificar e tornar mais atraente e agradável a utilização de um computador PC para, por exemplo, controlar desde simples até complexos circuitos externos, fazer o acompanhamento de um determinado processo através da aquisição de dados do ambiente deste, ou ainda, simplesmente tornar o comportamento de uma montagem mais 'dinâmico': facilmente alterável/programável - como um jogo de luzes, por exemplo.

Eduardo Divino Dias Vilela  
eddv@mailbr.com.br

A eletrônica evolui muito rapidamente, e devido ao mercado ser cada vez maior, tem possibilitado tanto ao profissional como ao hobbysta, o acesso a tecnologia de ponta e a ferramentas que antes eram disponíveis apenas em laboratórios bem equipados de centros de pesquisa.

Uma das ferramentas que já faz parte do cotidiano do profissional da área é o computador, e a finalidade deste mini-curso é dar um maior poder de desenvolvimento ao amante da eletrônica, através da apresentação do Delphi e de alguns projetos práticos que o utilizam.

Note que a intenção aqui não é dar um curso de programação completo: não é esta a finalidade, entretanto, basta o leitor possuir algumas noções básicas de programação – mesmo que você não conheça programação, mas usa o *Windows* - certamente conseguirá fazer o seu PC controlar um circuito externo, e facilmente fará o seu primeiro programa utilizando o Delphi.

## A programação para Windows

Como você já deve ter observado, diferentemente do DOS, onde precisa digitar um comando e teclar ENTER para executá-lo, o ambiente Windows é totalmente *visual* e é *orientado a eventos*.

Como? O Windows não faz nada, a não ser que você lhe ordene, e estas ordens são dadas através de eventos, tais como dar um clique com o mouse em um botão, clicar em um menu, pressionar a tecla ENTER, ou seja, o Windows funciona respondendo aos estímulos dados pelo usuário - os chamados eventos, e não diretamente por comandos escritos.

Eventos são ações do usuário ou ocorrências de sistema que o componente pode reconhecer. Um exemplo de uma ação de usuário é um clique em um botão. Um exemplo de uma ocorrência de sistema é um alarme gerado em um intervalo de tempo pré-determinado.

Veja, a propósito (na figura 1), esta *janela de Aviso* que é apresentada pelo aplicativo 'Bloco de Notas', após o usuário tentar fechar um arquivo que está sendo editado, sem salvar as últimas modificações: existem 4 possibilidades de geração de eventos pelo usuário - o clique em um dos três botões, ou um clique no ícone [X] na parte superior da janela.

Observe que conforme o evento que você gera - o clique em um ou outro lugar - a aplicação 'reage' de formas totalmente diferentes.

A característica visual e a reação a eventos diferencia crucialmente a programação para ambiente Windows da programação para outros ambientes e linguagens (ambiente DOS, linguagens C e Assembly).

Na programação Delphi, que também é uma linguagem visual, o desenvolvimento de um aplicativo é composto de duas fases: a definição da interface e a codificação dos manipuladores de eventos.

Em outras palavras, isto significa que para se criar um programa, deve-se seguir uma sequência natural de passos:

- Cria-se uma janela (denomina-se **Form**), e insere-se os **componentes** necessários para fazer o que se deseja. Componente é cada um dos diferentes itens utilizados no Windows: botões, rótulos, menus, caixas de seleção etc. Criar? No Delphi, para criar uma janela ou um botão, basta apenas dar um clique com o mouse na paleta de componentes - veremos isto mais adiante.

- Ajusta-se as **propriedades** destes componentes inseridos: define-se a cor da janela, o tamanho, o que vai escrito em cada botão, o título da janela etc.

- Codifica-se os **manipuladores de eventos** (*event handlers*): consiste na escrita do código que instrui o seu programa a reagir aos eventos de forma a realizar a tarefa que determinar. Utiliza-se a linguagem *Object Pascal*, uma linguagem bem estruturada e de fácil aprendizado.

Todas estas definições são salvas em um conjunto de arquivos, os arquivos do seu **projeto**, e você é o responsável por dar um nome ao projeto (apesar do Delphi dar um nome

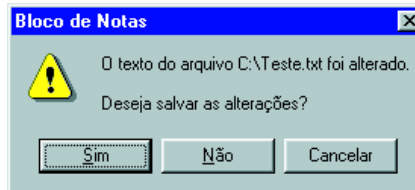


Fig. 1 - Eventos

padrão, você pode modificar para adequar ao seu propósito).

Tendo feito isto, é só **compilar** o seu programa e pronto: você o terá rodando no Windows, e fazendo o que desejar!

Vejamos onde isto se encaixa.

Para se criar um programa com o Delphi, é necessária a definição de vários itens, e o Delphi faz o gerenciamento de todas as informações de uma maneira muito interessante - ele cria uma estrutura de gerenciamento tratando o seu 'projeto de programa' da forma que ele é - um projeto.

Assim, um projeto é constituído fundamentalmente dos seguintes arquivos (que o Delphi cria para você automaticamente quando do início de um novo projeto):

*Project1.dpr* - Arquivo que armazena a descrição geral do seu projeto

*Unit1.dfm* - É a descrição textual da janela que você criou (o seu Form)

*Unit1.pas* - Contém o código básico e o código programado por você

*Unit1~pa* e *Unit1~df* - Cópias de backup dos respectivos arquivos.

E também os nomeados como *project*, com as seguintes extensões: *.cfg*, *.dof* e *.res* - que são arquivos de configurações do ambiente e arquivos de recursos - este último será abordado no momento oportuno.

Estes são os nomes *default*, ou seja, dados por padrão pelo Delphi, mas sugere-se que quando for salvá-los, faça a nomeação condizente com o seu trabalho, fornecendo nomes mais intuitivos.

Visualmente, o arquivo *Unit1.dfm* é o Form1 e seus componentes; e o arquivo *Unit1.pas* é o código que descreve em termos de *Object Pascal* o seu Form1, contendo as variáveis (cada componente: um botão, um rótulo, etc. do Form1 é uma variável que ocupará uma posição na memória do

PC) e contém também o código que você escrever para os eventos do seu programa.

## O ambiente de desenvolvimento Delphi - IDE

Quando se executa o Delphi, ele automaticamente abre um novo projeto de programa, e são mostradas as principais janelas do ambiente de desenvolvimento. Ver na figura 2.

O Ambiente de Desenvolvimento Integrado (*IDE - Integrated Development Environment*) é um conjunto de ferramentas que compõem a interface de desenvolvimento do Delphi, sendo que as principais ferramentas são

- Menus do Delphi
- Speed Bar
- Component Palette
- Object Inspector
- Form
- Code Editor
- Speed Menus
- On-Line Help

Segue-se uma sucinta descrição de cada uma destas partes:

**Menus do Delphi** : São constituídos de itens normais de um menu (abrir, salvar, editar, pesquisar etc), mais uma grande quantidade de itens próprios do ambiente de desenvolvimento Delphi. Os itens mais importantes são:

### File:

#### New application

Cria o esqueleto de um novo projeto Delphi.

#### New Form

Cria um novo *form* para o projeto atualmente aberto.

#### Open Project

Abre um projeto previamente salvo.

#### Reopen

Mostra uma lista dos últimos projetos abertos (rapidamente).

#### Save

Salva as modificações nas *units* atualmente abertas.

### Save Project As..

Salva as configurações do seu projeto

**Edit:** menu padrão do Windows, ressaltando a função.

### Align

Alinha os objetos selecionados no Form.

**Search:** faz pesquisas e substituições nas *units* do seu projeto - padrão do Windows.

**View:** (permite a visualização de diversas janelas/ferramentas do Delphi).

### Unit

Mostra a(s) Unit(s) que compõe(m) o seu projeto.

### Form

Mostra o(s) *Form*(s) que compõe(m) o seu projeto.

**Project:** (comandos do ambiente para manipulação do projeto).

### Compile Project1

Compila o seu projeto gerando o arquivo executável.

### Build Project1

Recompila o seu projeto gerando o arquivo executável.

### Syntax check Project1

Faz testes à procura de erros de sintaxe no código.

**Run:** (Executa o seu projeto)

### Run [F9]

Executa o projeto (execução do seu programa).

Os demais itens e menus serão abordados no momento oportuno.

**Speed Bars :** São ícones de atalho para alguns dos itens do menu do Delphi. A sua finalidade é dar maior agilidade ao usuário no uso do ambiente de programação. Os ícones são intuitivos, mas basta passar o cursor do mouse sobre cada botão, que aparece um *Hint* (pequeno texto de ajuda) indicando a função do botão.

**Component Palette:** Após abrir o ambiente Delphi, você encontrará um *Form* vazio e a respectiva *unit* com o código 'esqueleto' do seu projeto. Se você mandar o Delphi executar o seu projeto - 'rodar' - ele o fará, mas mostrará apenas uma janela vazia, e que

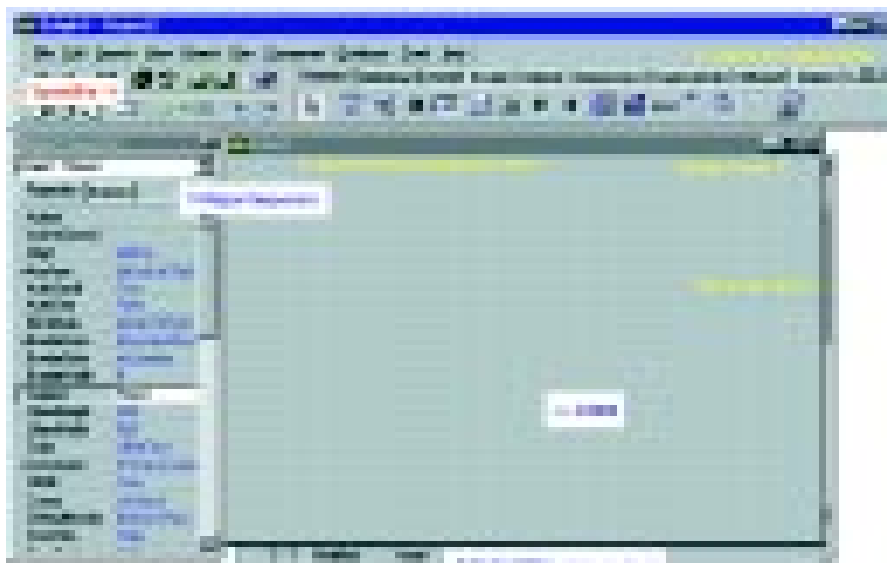


Fig. 2 - O ambiente de desenvolvimento Delphi.



Fig. 3 - A Component Palette

não faz nada. E deve ser assim, pois você ainda não programou nada e nem adicionou nenhum componente ao seu *Form*. Como já foi mencionado, um programa no estilo Windows geralmente começa pela definição da interface gráfica, ou seja, você deve 'desenhar' no seu *form* os itens que deseja que ele mostre, quando o seu programa for executado.

O próximo passo na *design* da interface é adicionar componentes ao *form*.

Componentes são elementos de programa das janelas, tais como rótulos, caixas de edição, botões etc. e estão localizados na *Component Palette*, sendo utilizados para construir a interface com o usuário. A Paleta de Componente exibe uma seleção de componentes agrupados por função ou utilização. Observe a figura 3.

Para definir o visual do programa, basta você arrastar o componente da paleta e soltar sobre o *form*, movendo e redimensionando-o livremente com o mouse.

### Object Inspector

Os componentes do Delphi (botões, *labels*, caixa de edição, ou mesmo um *form*) são caracterizados por propriedades e eventos. Propriedades,

como a própria palavra diz, são as características que definem o componente - exemplo: cor, largura, altura, nome etc.

E os eventos são todas as formas que o usuário pode interagir com o componente dinamicamente: o evento *Click* (quando o usuário dá um clique simples com o mouse sobre o componente), o *DoubleClick* (análogo, mas com um duplo clique), o *MouseMove* (quando se passa o mouse sobre o componente), etc.

Ao selecionar um componente do *form*, o *Object Inspector* automaticamente exibe as propriedades e eventos que podem ser utilizados para ele. Dessa forma, o *Object Inspector* permite personalizar os componentes visualmente, sem a necessidade de codificação.

A janela vista na figura 4 mostra o *Object Inspector* exibindo as propriedades do *Form*.

Vejamos algumas propriedades:

**Color :** é a cor de fundo do *Form* (geralmente, o *default* é cinza);

**Caption:** o texto de rótulo que aparece no componente (no caso do *Form*, aparece na barra de título);

**Border Style :** tipo de borda apresentado pelo *form* após quando em *run-time* (quando está sendo executado).

Altere-as, e perceba o resultado.

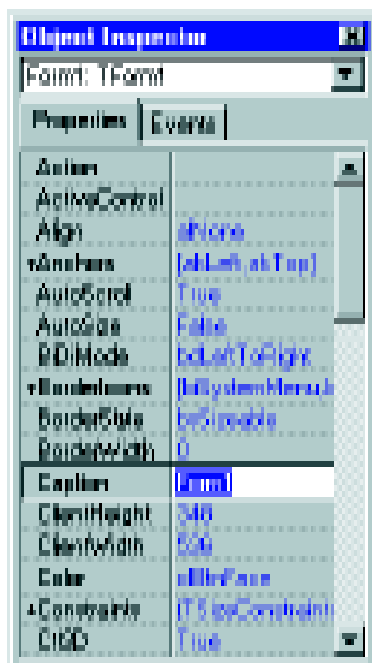


Fig. 4 - Propriedades

E alguns Eventos:

**onClick:** quando o objeto é clicado com o mouse (em *run-time*).

**onCreate:** evento que é acionado no momento em que o *form* é criado.

Para escrever um código que você quer que seja executado em um determinado evento, basta ir no *Object Inspector*, selecionar a aba *Events*, e dar um duplo clique no nome do evento, e o Delphi criará o procedimento (*procedure*) correspondente na janela do Editor de Código (*Code Editor*), e assim poderá escrever o código.

### Conheçamos o Code Editor

O Editor de Código na verdade é “apenas” um editor de textos bastante *incrementado*, e é nele que você irá codificar o seu programa, de forma a que ele reaja aos eventos da maneira que desejar. Quando você inicializa o Delphi, o *default* do *Code Editor* é aparecer aberto no arquivo *unit1*, que é posicionado atrás do *Form1*. Quando você dá um duplo clique em um evento na página *Events* do *Object Inspector*, o *Code Editor* automaticamente torna-se a janela ativa e é mostrado em primeiro plano para que possa digitar o código para manipular o evento. Uma *unit* é utilizada para or-

ganizar as declarações de programação do Delphi. Por exemplo, *procedures utilitárias* (as que você criar e que não sejam as *procedures* manipuladores de eventos) podem ser agrupadas como uma *unit*. Um *form* é organizado como uma *unit*.

Você também pode exibir um arquivo *unit* através do menu *View* selecionando *Toggle Form/Unit* ou via tecla F12. O arquivo *default*, *Unit1.pas*, aparece na janela do *Code Editor*, como na figura 5 a seguir.

Não se preocupe, por enquanto, com estes códigos que o Delphi gerou automaticamente. Na verdade, estas linhas de código são as instruções em *Object Pascal* para que o Delphi gere o arquivo executável assim que você compilar o projeto.

### Speed Menus e On-Line Help

Por fim, estas duas características do Delphi são ferramentas que auxiliam na tarefa criativa da programação: os menus de acesso rápido através do clique com o botão direito do mouse, que apresentam menus locais com atalhos para trabalhar melhor com o ambiente, tais como abrir/fechar uma *unit*, utilizar *bookmarks* no código para acelerar a busca por uma parte do código etc.

E o *on-line help* que é o acesso direto aos arquivos de ajuda a partir de qualquer parte do ambiente Delphi,

inclusive do código que você escreve. Para que possa observar a funcionalidade do *on-line help*, clique sobre a área do *form* para que ele fique todo visível e então, com o mouse sobre a área do *form*, pressione F1. Ao pressionar esta tecla, como está sobre o *form*, o Delphi entende que você deseja receber ajuda sobre o componente *form*, abrindo assim, o arquivo de ajuda onde descreve o componente *form*, inclusive versando sobre cada uma de suas propriedades e eventos. Utilize e navegue o máximo possível pelo *help* do Delphi, que certamente aprenderá muita coisa!

### Object Pascal

Para que o seu programa reaja convenientemente aos eventos, é preciso que você escreva o código que deverá ser executado ao ocorrer em aqueles. A codificação em Delphi é feita através da linguagem *Object Pascal*, uma linguagem com sintaxe e estrutura bastante rica. O *Object Pascal* é bastante semelhante ao Pascal comum, entretanto, fortalecido com a utilização de um poderoso paradigma da programação: a orientação a objeto.

Lembre-se que já foi apresentado sucintamente o componente *Form*, e antes de vermos um exemplo, você precisa conhecer um outro componente do Delphi: o *Edit*.

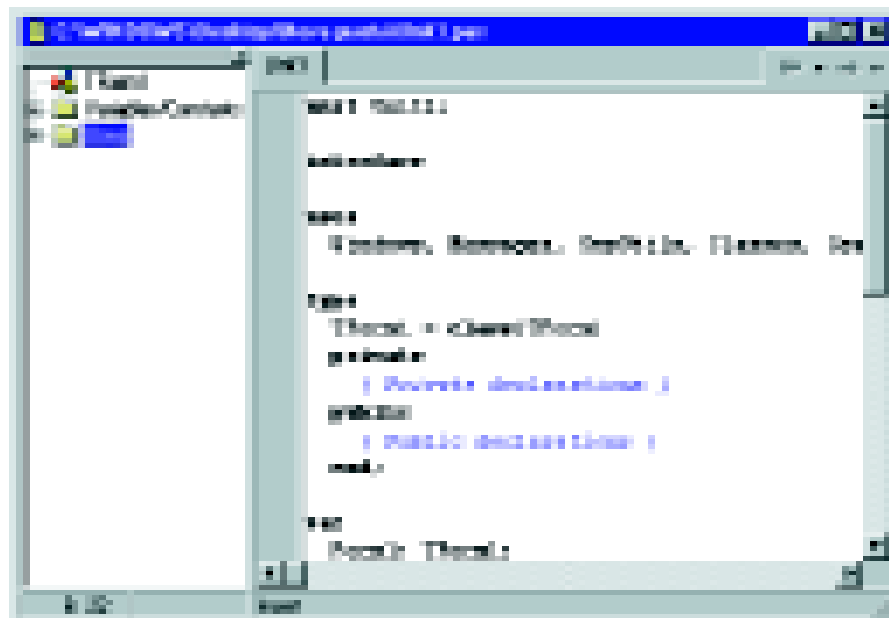


Fig. 5 - A janela Code Editor

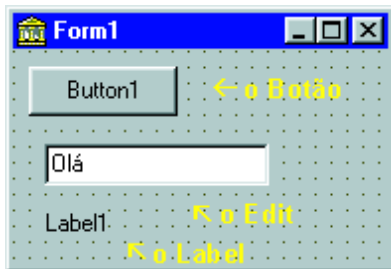


Fig. 6 - Alguns componentes.

A partir deste ponto, a figura 6 será o foco da discussão.

O *Edit* é aquele 'campo' geralmente usado para receber caracteres digitados pelo usuário. Com certeza você pode se lembrar de vários aplicativos Windows que o utilizam.

A propriedade mais importante deste componente é a chamada *Text*. É ela que contém o texto que é mostrado para o usuário. No caso da figura acima, a propriedade *Text* do *Edit1* mostrado contém a sequência de caracteres (*string*) 'Olá'.

Como você já viu, pode acessar as propriedades de um componente utilizando o *Object Inspector*, por exemplo, atribuir 'Olá' à propriedade *text* de um *Edit*, mas isto só é possível quando se está criando o programa (*design-time*). Quando se está executando (*run-time*), para acessar as propriedades de um componente, você deverá fazê-lo via codificação, na seguinte sintaxe:

*Nome\_do\_componente.Propriedade*

Assim, para se atribuir outra *string* à propriedade *Text* do *Edit1*, basta codificar a seguinte linha: *Edit1.Text := 'Eletrônica';*

Onde *Edit1* é o nome que foi dado àquele componente *Edit* (a sua propriedade *nome*, veja-a no *Object Inspector*), os caracteres '=' indicam atribuição: você está atribuindo o que está à direita ao que está à esquerda.

Para você se familiarizar um pouco com o *Object Pascal*, analisemos este pequeno trecho de código da figura 7.

Temos aqui um pequeno fragmento de código que é o que foi escrito para o manipulador de eventos de um botão, mais precisamente, no manipulador do evento *Click* do botão. Portanto, este código será ativado

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  if Edit1.Text <> '' then
  begin
    Label1.Caption := Edit1.text;
    Label1.Color := clRed;
    Edit1.Text := '';
  end;
end;
```

Fig. 7 - Código.

quando for dado um clique com o mouse sobre o botão *Button1*. A linha *procedure TForm1.Button1Click(Sender: TObject);* indica que o código a seguir, delimitado pelas palavras reservadas *begin* e *end*; (os que estão alinhados na mesma coluna da palavra 'procedure'), trata-se do manipulador do evento *Click* do botão *Button1* que está no *form Form1*. Os parâmetros (*Sender: TObject*); não são importantes neste ponto, oportunamente voltaremos a eles.

O código é semelhante ao da figura 8.

Esta *estrutura de seleção* é executada do seguinte modo: **se** a expressão for verdadeira, **então** as instruções dentro do bloco delimitado por **begin** e **end** são executadas.

Note que no trecho de código que foi dado, a expressão é: *Edit1.Text <> ''*. A palavra *Text* que segue ao nome *Edit1* indica que se está referindo à propriedade *Text* do componente *Edit*.

Portanto, a expressão que o Delphi avaliará quanto à veracidade é se a propriedade *text* do *Edit1* é 'nenhum\_texto'. Assim, a expressão será verdadeira e as instruções serão executadas quando existir qualquer texto na propriedade *Text* do *Edit1*; mas, se você selecionar o texto e apagá-lo, a expressão *Edit1.Text <> ''* não será verdadeira, logo, as instruções não serão executadas.

### Mas o que significa o código que será executado (se a expressão for verdadeira)?

A linha:

*Label1.Caption := Edit1.text;* Atribui o conteúdo da propriedade *Text* do componente *Edit1* à propriedade *caption* (o rótulo) do componente cujo nome é *Label1*, presente no *form* (veja

a figura 7). Assim, se o conteúdo do *Edit1.Text* for o texto 'Olá', este texto será também atribuído ao componente *label* (*Label1*, no caso) e este também passará a exibí-lo.

*Label1.Color := clRed;* Atribui a cor vermelha à propriedade *Color* do *label*, e '*clRed*' é uma constante do Delphi que simboliza o valor em hexadecimal da cor vermelha. Assim, você não precisa ter uma tabela de códigos hexadecimais para poder alterar as cores dos componentes.

*Edit1.Text := ''* atribui uma *string* 'em branco' à propriedade *text* do *Edit1*.

Lembre-se de que este código está escrito no manipulador de eventos do botão *Button1Click*.

Portanto, quando o programa estiver sendo executado, se você der um clique com o mouse, e supondo-se que os componentes estejam com as propriedades definidas como aparecem na figura 6, o código será executado, e você verá a janela ser alterada para a mostrada na figura 9.

```
if expressão then
begin
  Instrução1;
  Instrução2;
  ...
  InstruçãoN;
end;
```

Fig. 8 - Estrutura de Seleção.

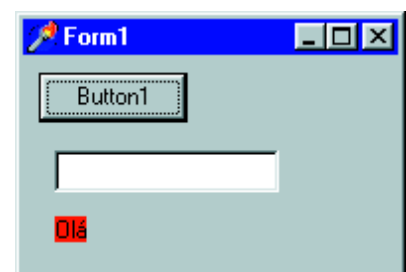


Fig. 9 - Após o clique.

Assim, tudo que você digitar no componente *Edit*, e depois pressionar o botão, vai ser passado para o *Caption* do *label*, juntamente com a 'limpeza' do conteúdo do *Edit*.

### O seu primeiro programa em Delphi

Tendo-se os conceitos básicos, o modo mais eficiente de aprender a programar, é programando. Logo, sugiro que você crie este mesmo programa em seu micro e faça experiências, de forma a compreender realmente o que foi abordado aqui.

Para fazer este exemplo em seu micro, bastará proceder da seguinte forma:

Abra o ambiente do Delphi (ao abrí-lo, ele cria automaticamente um novo projeto) e será exibido então um *form* 'limpo'; logo, o primeiro passo é criar a interface visual.

Adicione um componente botão ao seu *form*: para isso selecione a aba '*standard*' na Paleta de Componentes e clique no componente Botão, arrastando-o para o *form*.

Da mesma maneira, faça para o componente *Edit* e para o *Label*.

Para que a janela do seu projeto fique semelhante à mostrada na figura 6, basta redimensionar o *form* (da mesma forma que você redimensiona qualquer janela do Windows), e então selecionar o componente *Edit1*, pressionar F11 e aparecerá o *Object Inspector*, listando as propriedades do componente selecionado (o *Edit1*) - procure a propriedade *Text* dele e mude para 'Olá'. Pronto. A parte visual já está OK. Agora você deve codificar o manipulador do evento *click* do botão.

Para fazê-lo, basta dar um duplo clique no botão: o Delphi abre o *Code Editor* mostrando o seguinte esqueleto de código, figura 10.

Você deve então inserir o código que já discutimos (veja figura 7). Tendo codificado, é só salvar todo o projeto e executar.

Para isso, utilize os itens '*Save*' e o '*Save Project As..*' do menu '*File*'. E para rodar o programa (executá-lo), utilize o item '*Run*' do menu '*Run*', ou simplesmente utilize a tecla de atalho 'F9'.

### O que se pode fazer com o Delphi?

O Delphi é considerado a ferramenta de programação para *Windows* mais potente do mercado, permitindo ao programador desenvolver desde programas simples, até sistemas complexos.

Portanto, o que você imaginar para fazer em uma aplicação para *Windows*, com certeza pode ser feito utilizando o Delphi. Veremos progressivamente no decorrer deste curso, como utilizar o Delphi, dando ênfase às aplicações para eletrônica.

Faremos acessos ao *hardware* do PC, via *ports* de I/O (porta paralela e porta serial), de modo a fazer a interface 'de mão dupla' PC/mundo externo.

Como você deve ter notado, o Delphi é um ambiente complexo em termos de recursos, mas fácil de se trabalhar e de se programar, basta um pouco de paciência no início, e logo você se verá surpreso com o que conseguirá fazer.

Um conselho: não perca de maneira alguma as próximas edições!

A sua criatividade será o limite!

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    { insira aqui o código }
end;

end.
```

Figura 10