

# **APOSTILA**

## **DE**

### **DELPHI 7.0**



**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO.**  
**PROFa. GILENE BORGES GOMES.**  
**Home page: <http://www.gomeshp.com>**  
**E-mail: [gilene@gomeshp.com](mailto:gilene@gomeshp.com)**

## 1. INTRODUÇÃO

Delphi possui um ambiente de desenvolvimento fácil de usar, com uma grande Biblioteca de Componentes Visuais (VCL - Visual Component Library). A VCL contém código de botões, campos, rótulos, gráficos, caixas de diálogo e acesso a tabelas de bancos de dados, e foi desenvolvida levando em conta as velocidades no desenvolvimento de aplicativos e na execução destes aplicativos.

Vantagens do Delphi :

- Facilidade em alterações e implementações
- Melhor estruturação do código
- Velocidade
- Verdadeira orientação a objetos

O rápido desenvolvimento de aplicativos é possível graças aos vários controles disponíveis na paleta de componentes, onde o programador escolhe um destes componentes, e coloca-o diretamente no local desejado, dentro de um formulário. Formulário este que será a janela do aplicativo apresentada ao usuário.

O Delphi permite o uso de objetos, e sua criação. Ele trabalha com eventos que dão início à alguma rotina de trabalho, ou seja, o programa fica parado até que um evento ocorra.

Um programa tradicional, feito para ser executado em DOS, é organizado em torno de estruturas de dados com um loop principal e uma série de sub-rotinas constituindo o aplicativo, com procedimentos e funções separados para manipular os dados.

Um programa orientado a objetos e eventos é organizado em torno de um conjunto de objetos. Onde cada objeto possui propriedades que o definem, e vários códigos (eventos) dando funcionalidade a este objeto. Ou seja, objetos são estruturas que combinam dados e funções em uma mesma estrutura.

Um Objeto possui dados internos, que não podem ser acessados por outros objetos e dados externos, também chamados de propriedades, estas podendo ser acessadas de fora deste objeto. De maneira semelhante, um objeto possui rotinas internas que são usadas apenas internamente e rotinas externas, também chamadas de métodos, que podem ser acessadas externamente.

Um carro é um objeto que possui propriedades e métodos. A tabela abaixo lista algumas propriedades e comportamentos do objeto real **carro**.

Propriedades	Métodos
cor	dar partida
potência do motor	acelerar
tipo de pintura	frear

Um método é uma rotina própria do objeto que o dá funcionalidade, ou seja, torna-o vivo, e as propriedades fazem o intercâmbio entre o objeto e o programa.

## 2. AMBIENTE DELPHI

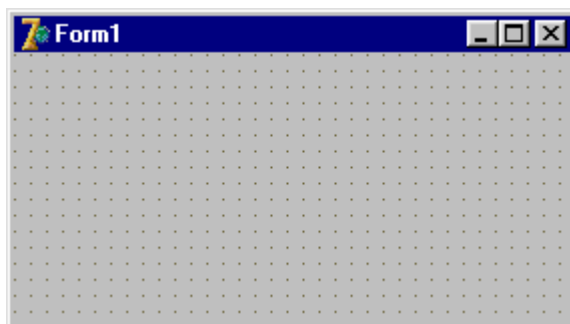
Neste item será estudado o **IDE** (*Integrated Developer Environment* – Ambiente de Desenvolvimento Integrado) do Delphi.

O Delphi possui um conjunto de ferramentas que permitem facilitar e agilizar a construção de programas, permitindo uma melhor interação entre o programador e o computador. Suas principais janelas são:

- Janela FORM
- Janela OBJECT INSPECTOR
- Janela CODE EDITOR
- Janela OBJECT TREEVIEW
- Janela MAIN

### 2.1. JANELA FORM

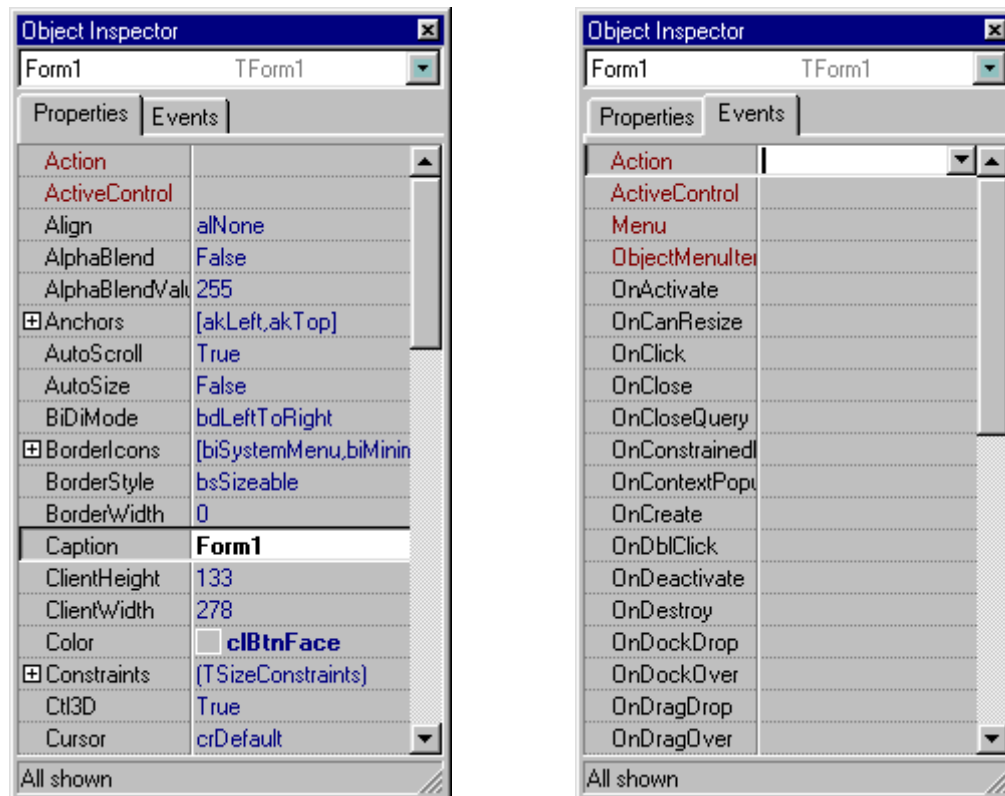
O FORM é a tela onde o desenvolvedor constrói sua aplicação. A partir de um FORM é que se estabelece a interação USUÁRIO-COMPUTADOR, através de botões, rótulos e outros componentes, estabelecendo-se funções, métodos ou eventos que serão ativados. Os componentes são dispostos dentro da área útil do FORM.



A Janela FORM: interface.

### 2.2. JANELA OBJECT INSPECTOR

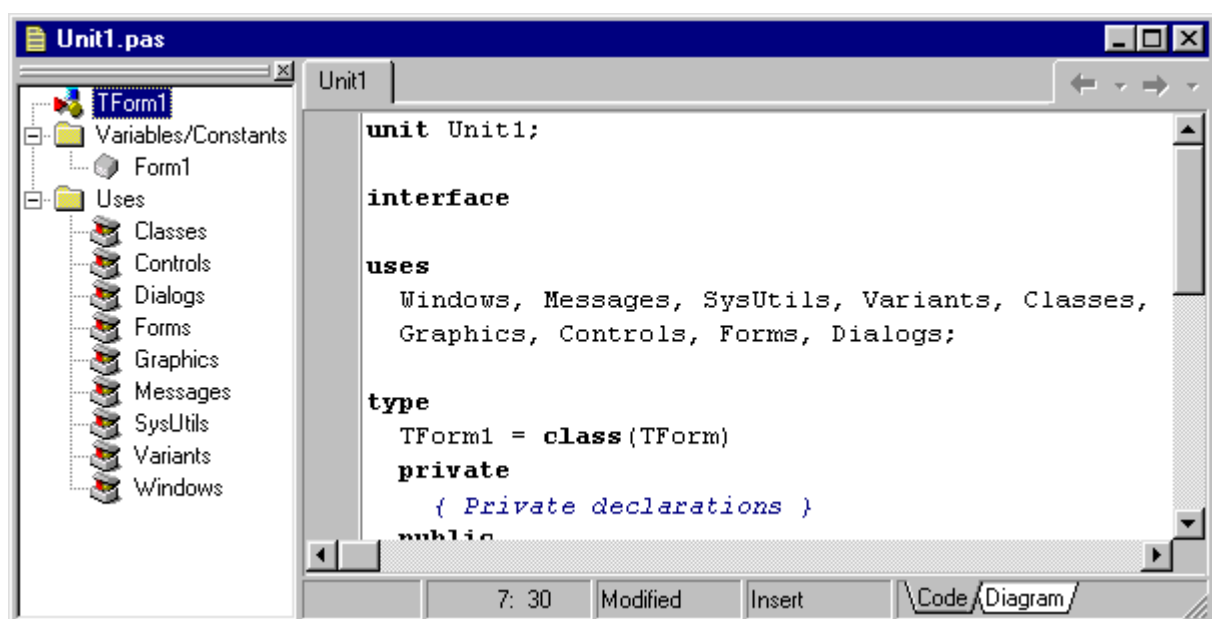
A janela OBJECT INSPECTOR contém propriedades e eventos dos componentes inseridos em um FORM, e do próprio FORM. É na guia *Properties* (**Propriedades**), por exemplo, que se estabelecem as características de cada componente, como nome, fonte, altura, largura, etc. Já na guia *Events* (**Eventos**) estabelecem-se ações a serem tomadas pelo componente a partir de um evento associado ao mouse, teclado, sistema operacional, etc.



A Janela OBJECT INSPECTOR: propriedades e eventos

## 2.3. JANELA CODE EDITOR

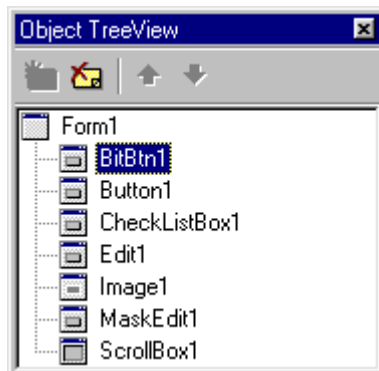
A janela CODE EDITOR, ou editor de código, é onde se desenvolve o programa fonte. É neste editor que se encontra a estrutura sintática propriamente dita da Linguagem Object Pascal, que é utilizada pelo Delphi. Cabe ressaltar, no entanto, que boa parte do código escrito é gerado automaticamente.



A Janela CODE EDITOR: código fonte da aplicação

## 2.4. JANELA OBJECT TREEVIEW

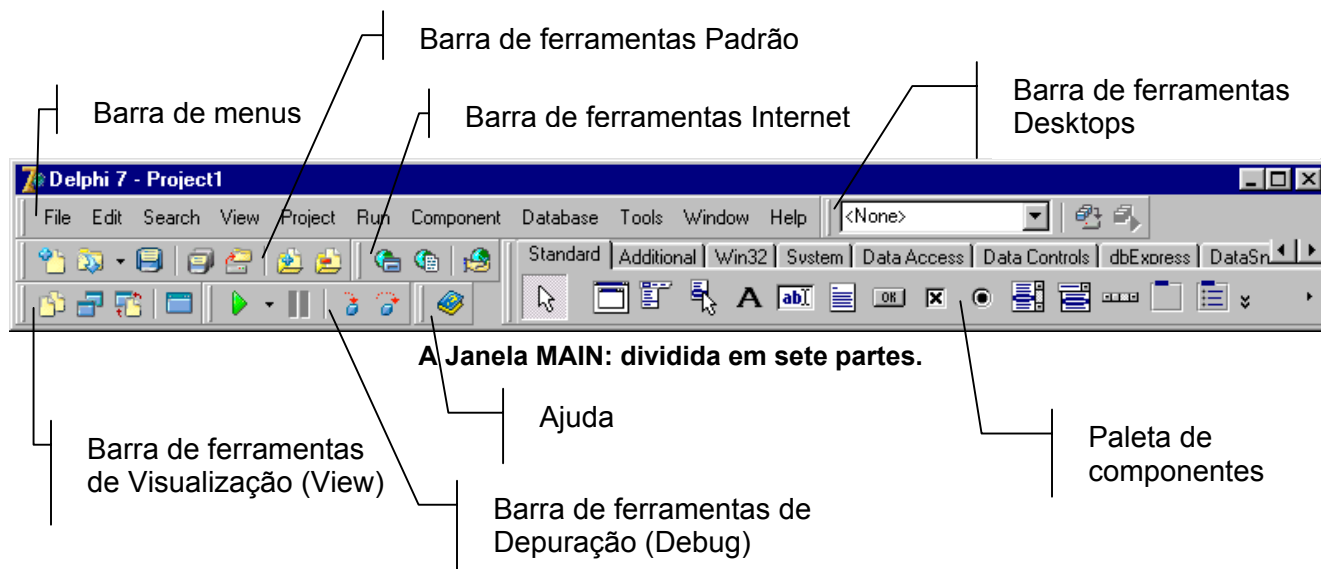
Object TreeView apresenta uma árvore do diagrama dos componentes visuais e não visuais colocados no formulário, no módulo de dados ou no frame.



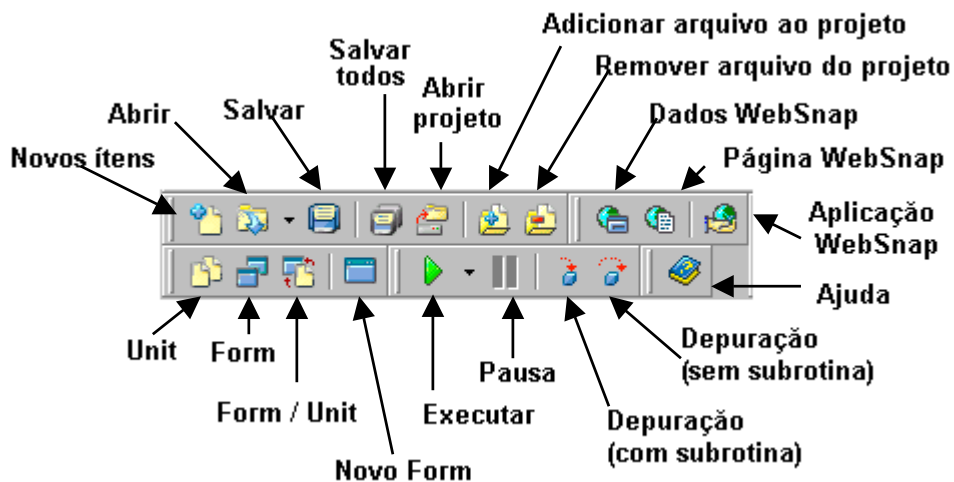
A Janela OBJECT TREEVIEW: Diagramas da aplicação

## 2.5. JANELA MAIN

A janela MAIN, ou janela principal, controla o funcionamento do Delphi. Esta janela pode ser dividida em sete partes:



### 2.5.1. Barra de Ferramentas



### 2.5.2. Menu Principal

O menu principal contém as opções de utilização do Delphi:

- **File:** permite a manipulação de arquivos do desenvolvedor (PAS, DPR, ...);
- **Edit:** apresenta opções de edição;
- **Search:** apresenta opções de pesquisa e localização;
- **View:** permite verificar detalhes do projeto;
- **Project:** permite adicionar ou remover partes em um projeto, bem como compilá-lo;
- **Run:** apresenta opções de execução e depuração do projeto;
- **Component:** permite a criação ou instalação de novos componentes no Delphi;
- **Database:** apresenta opções de uso de banco de dados;
- **Tools:** permite configurar o ambiente de trabalho, bem com acessar ferramentas externas ao Delphi;
- **Window:** permite alternar entre as principais janelas do Delphi;
- **Help:** ajuda do Delphi.

### 2.5.3. Paleta de Componentes

A Paleta de Componentes possui todos os controles necessários para desenharmos nossa janela - formulário - como um programa de desenho livre. Para incluir um controle no formulário, existem dois métodos:

1 - Click Duplo no ícone da paleta de componentes. Fará com que o controle seja inserido no centro do formulário com um tamanho padrão.

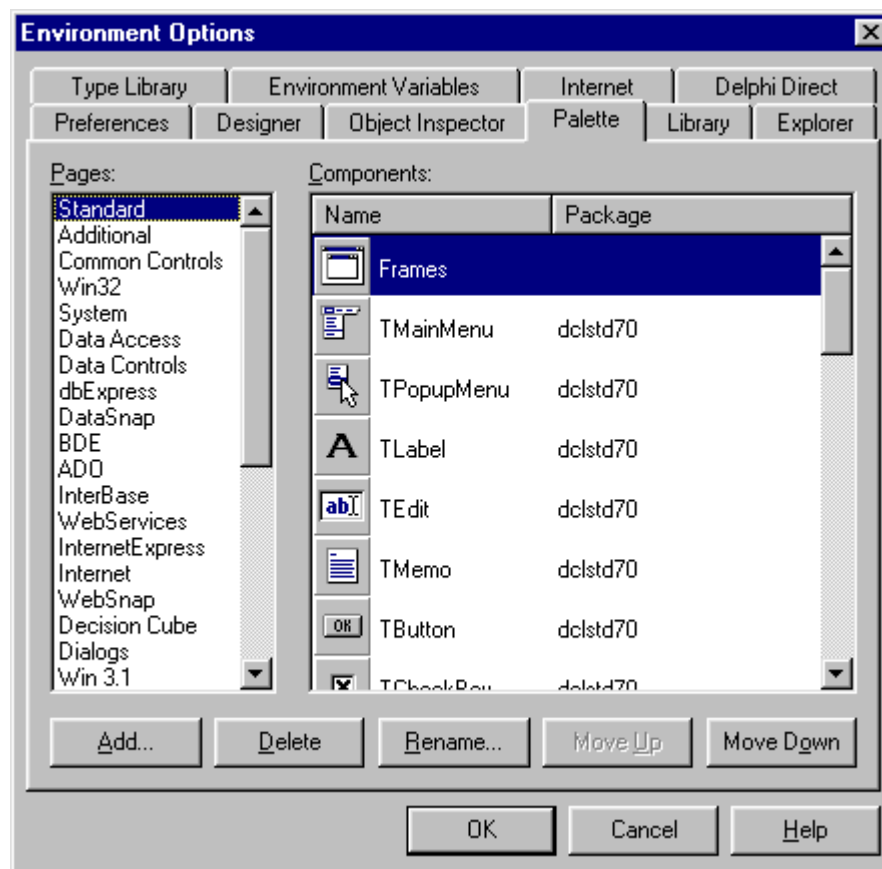
2 - Selecionar o ícone na caixa de ferramentas e depois dar um clique no formulário, na posição desejada para o objeto (canto superior esquerdo deste).

Podemos dimensionar estes controles, depois de inseridos, a qualquer momento durante o desenvolvimento. Primeiro seleciona o controle dando um clique em cima dele e depois o dimensionamos arrastando um dos oito botões dimensionadores que circundam este objeto.

A paleta de componentes é a biblioteca de classes que fornece recursos para o desenvolvimento visual em Delphi. As classes representadas na paleta de componentes estão separadas por tipos, as paletas que serão utilizadas neste curso são:

- **Standard**: componentes mais comuns e usados.
- **Additional**: componentes adicionais também de uso comum.
- **Win32**: componentes para acesso de controles comuns de interface de usuário do Windows 32-bits.
- **System**: componentes para aproveitar recursos de sistema operacional.

Os componentes podem ser incluídos ou excluídos da paleta de componentes. Basta abrir a caixa de diálogo *Environment Options* do menu *Tools* e selecionar a guia *Palette*.



Como configurar os componentes das paletas.

Os componentes disponíveis na VCL podem ser divididos entre:

**COMPONENTES VISUAIS** - podem ter sua forma e tamanho alterados no formulário (Form), além das propriedades e eventos no Object Inspector. Eles aparecem durante a execução do aplicativo exatamente como foram definidos durante o projeto.

**COMPONENTES NÃO-VISUAIS** - ficam apenas como a representação de um ícone no formulário (Form), mas suas propriedades e eventos podem ser alterados no Object Inspector. Eles não aparecem no formulário durante a execução do aplicativo, podendo ser ativados por comandos específicos (por exemplo, podemos citar a caixa de diálogo abrir arquivo).

### 2.5.3.1 Paleta STANDARD

A paleta Standard contém 16 componentes mais comuns para a construção de aplicações.



**Paleta Standard: componentes mais comuns**

Os componentes desta paleta, respectivamente, são:

Frames	Abre uma caixa de diálogo mostrando uma lista de frames incluídos no projeto corrente.
MainMenu	Permite a construção da barra de menus e de menus suspensos.
PopupMenu	Permite a construção de menus a partir do botão direito do mouse.
Label	Permite colocar textos que não podem ser selecionados ou alterados pelo usuário.
Edit	Permite a apresentação ou a entrada de dados pelo usuário.
Memo	Permite a introdução ou exibição de uma área de texto.
Button	Permite a colocação de botões para inicialização de ações por parte do usuário.
CheckBox	Permite a colocação de caixa de verificação para a seleção de diversas opções.
RadioButton	Permite a colocação de botões de seleção de onde pode ser selecionada apenas uma opção.
ListBox	Apresenta uma lista de itens que podem ser selecionados.
ComboBox	Apresenta uma lista de itens de onde pode ser selecionado apenas um. Este componente também permite que o usuário digite sua própria opção.
ScrollBar	Permite criar a barras de rolagem verticais ou horizontais, no padrão do Windows®.
GroupBox	Permite agrupar controles como <i>CheckBox</i> , <i>RadioButton</i> , etc.
RadioGroup	Permite agrupar <i>RadioButtons</i> para que se faça a seleção de uma opção.
Panel	Cria painéis que contém outros componentes num formulário. São utilizados para construir barra de <i>status</i> , barra de ferramentas, etc.
ActionList	Cria coleções de ações que centraliza as respostas da aplicação para as ações do usuário.

### 2.5.3.2 Paleta ADDITIONAL

A paleta Additional tem 25 componentes, também de uso comum, mas com algumas funções mais especializadas.



**Paleta Additional: mais componentes de uso comum**

Os primeiros componentes desta paleta, respectivamente, são:



BitBtn	Permite a colocação de botões com imagem bitmap.
SpeedButton	Permite a criação de barra de ferramentas e conjuntos de botões. Devem ser utilizados juntamente com o componente <i>Panel</i> .
MaskEdit	Permite a entrada de dados definindo-se máscaras de leitura.
StringGrid	Permite a apresentação de <i>strings</i> em colunas.
DrawGrid	Permite a apresentação de informações em colunas e linhas.
Image	Permite a apresentação de imagens gráficas.
Shape	Permite o desenho de figuras geométricas.
Bevel	Permite o desenho de retângulos em relevo.
ScrollBar	Cria áreas de exibição com barras de rolagem, quando necessário.
CheckListBox	Similar ao ListBox onde cada item tem um CheckBox.

### 2.5.3.3 Paleta WIN32

A paleta Win32 contém 18 componentes para criar aplicações que tenham a aparência do Windows95©.



**Paleta Win32: aplicativos com a aparência do Windows.**

### 2.5.3.4 Paleta SYSTEM

A paleta System contém 8 componentes que permitem utilizar em suas aplicações alguns recursos do sistema operacional.



**Paleta System: utilizar recursos do sistema operacional**

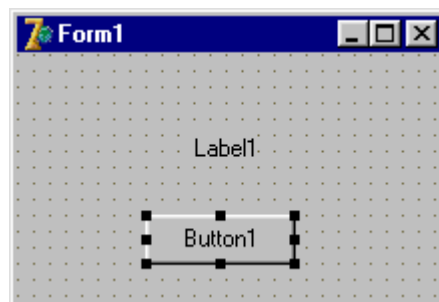
### 3. MEU PRIMEIRO PROGRAMA

Para iniciar, vamos construir um programa que quando for dado um clique no botão de comando, será mostrada uma mensagem. E posteriormente poderemos alterar a cor desta mensagem através de outros botões.

Existem três passos principais para a escrita de uma aplicação no Delphi que iremos seguir:

- **Desenhar as janelas que se deseja usar.**  
Inserir no formulário os controles que serão necessários.
- **Adaptar as propriedades dos objetos.**  
Alterar as propriedades dos controles às necessidades da aplicação.
- **Escrever o código para os eventos associados.**  
Esta é a parte mais complexa do desenvolvimento, é ela que dá a funcionalidade ao programa, são as rotinas que começam a ser executadas a partir de um evento.

#### 3.1. DESENHAR AS JANELAS QUE SE DESEJA USAR.

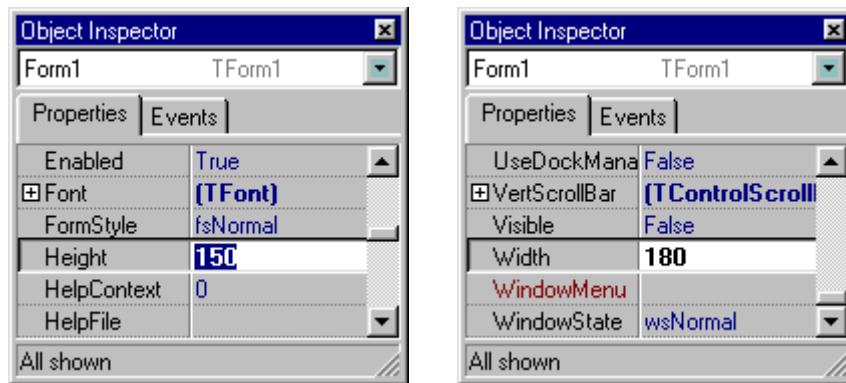


- 1 - Começamos inserindo um **Label** (Legenda) e um **Button** (Botão de Comando) no Formulário.
- 2 - Observe que, quando o controle estiver selecionado, poderemos arrastá-lo e dimensioná-lo dentro do formulário.

#### 3.2. ADAPTAR AS PROPRIEDADES DOS OBJETOS

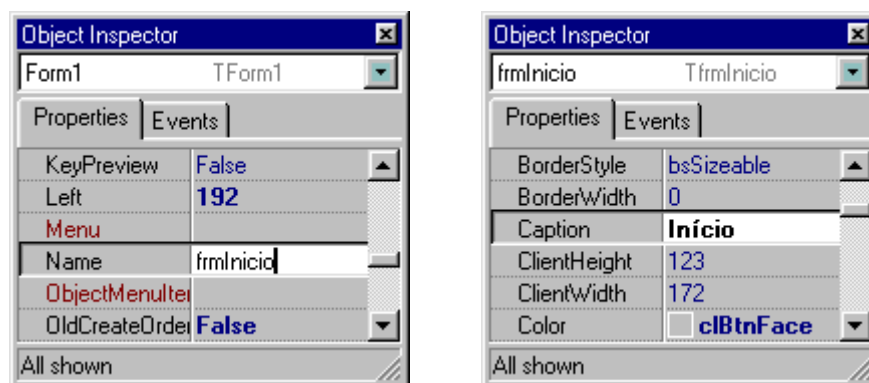
Para se alterar a propriedade de um objeto, ele tem que estar selecionado (com os oito pontos dimensionadores visíveis), depois procurar o nome da propriedade a ser alterada, na janela Object Inspector, e selecionar (no caso de valores padrão) o seu valor, ou então escrever um valor.

- 1 - Dimensione o formulário da seguinte maneira:  
Selecionar a propriedade **Height**, e atribuir a ela o valor de 150.  
Selecionar a propriedade **Width** e dar o valor de 180.



Estes números correspondem a Pixels, que é a quantidade de pontos do monitor.

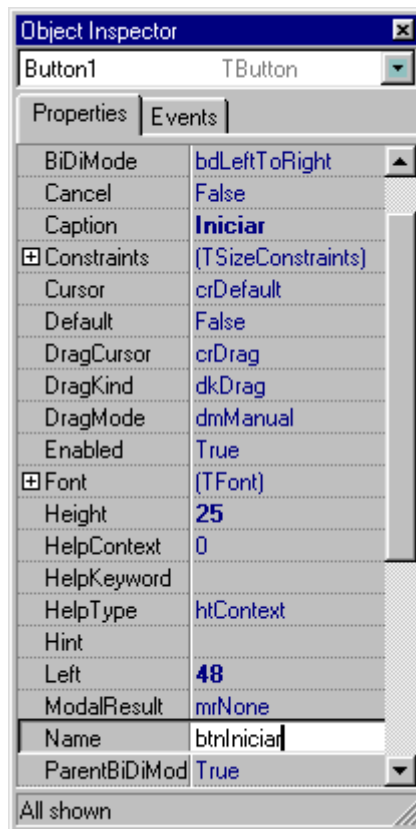
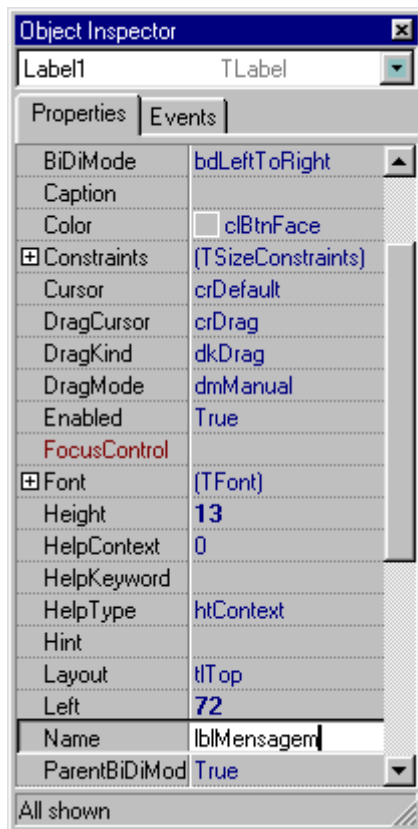
O mesmo deverá ser feito para as propriedades **Name** e **Caption**. A propriedade Name será a identificação do Objeto quando construirmos o código da aplicação. E a propriedade Caption é a palavra que aparecerá como título da janela.



Após você alterar estas quatro propriedades (Caption, Height, Name e Width) do formulário, ela estará assim:



Agora, altere as propriedades Caption e Name dos componentes **TLabel** e **TButton**.



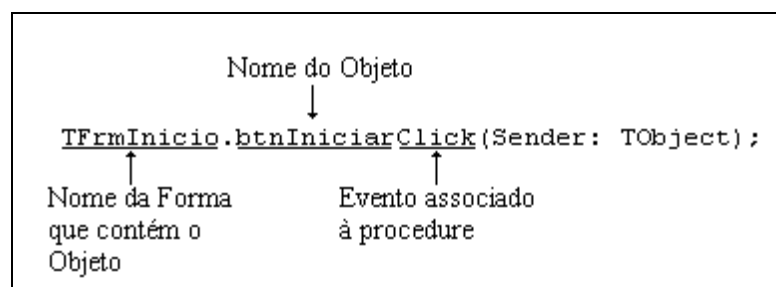
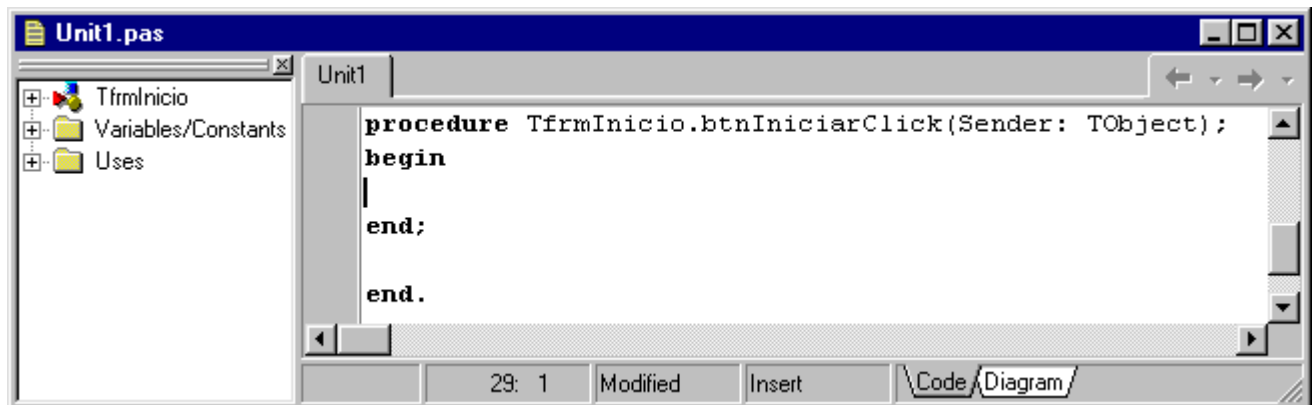
### 3.3. ESCREVER O CÓDIGO PARA OS EVENTOS ASSOCIADOS

O código é escrito na janela **Unit**, para acessá-la, selecione o botão **Iniciar** e na janela *Object Inspector*, selecione a guia *Events* e dê um duplo clique na parte direita da linha que contém o evento **OnClick** - a rotina escrita para este evento, será executada quando o botão **Iniciar** for clicado. Isto traz a janela *Unit* para a frente.



#### 3.3.1. Janela Unit

Nesta janela observamos o nome da procedure, identificando qual o objeto e o evento que dará início à execução do código, e onde está localizado este objeto.



Todas as instruções a serem executadas por um procedimento devem estar entre as palavras reservadas **begin** e **end**.

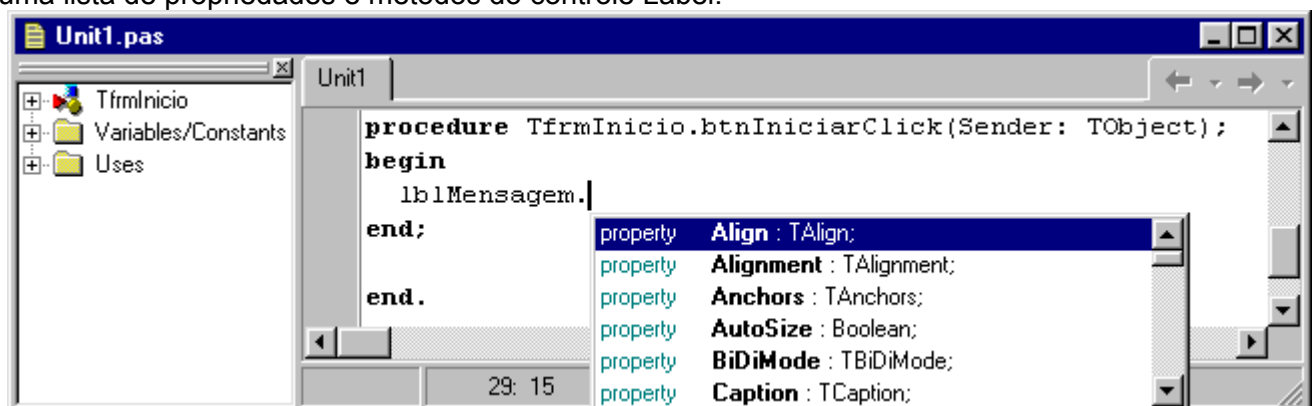
A Janela Unit também pode ser acessada dando-se um duplo clique no objeto que se quer criar um código. Cada objeto tem um evento que é mais comumente utilizado, e é com este evento que o Delphi iniciará a Janela Unit quando acessada desta forma, isto não impede que criemos outros códigos utilizando mais de um evento ao mesmo tempo.

O nosso projeto de Início, mostrará uma mensagem no Label (objeto) com um Click (evento) no Botão "Iniciar" (objeto). Ou seja, iremos alterar a propriedade Caption de lblMensagem, esta propriedade contém o que será mostrado ao usuário.

Atribuímos valores a uma propriedade de objeto seguindo o padrão:

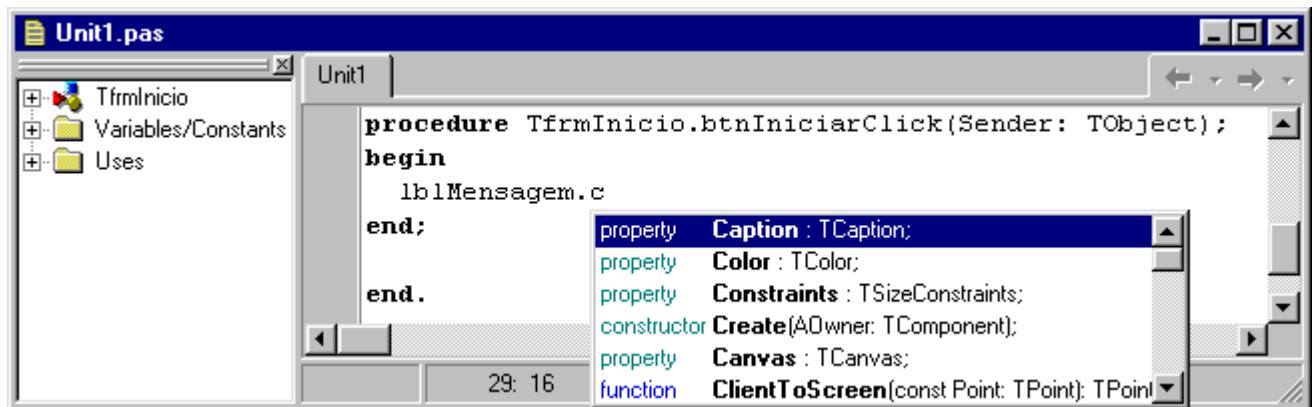
objeto + . + propriedade + := + valor da propriedade;

Abra a Janela **Unit** para o botão de comando e digite o código conforme a figura a seguir. Repare que ao digitar o ponto após lblMensagem, e aguardando alguns instantes, o Delphi exibirá uma lista de propriedades e métodos do controle Label.

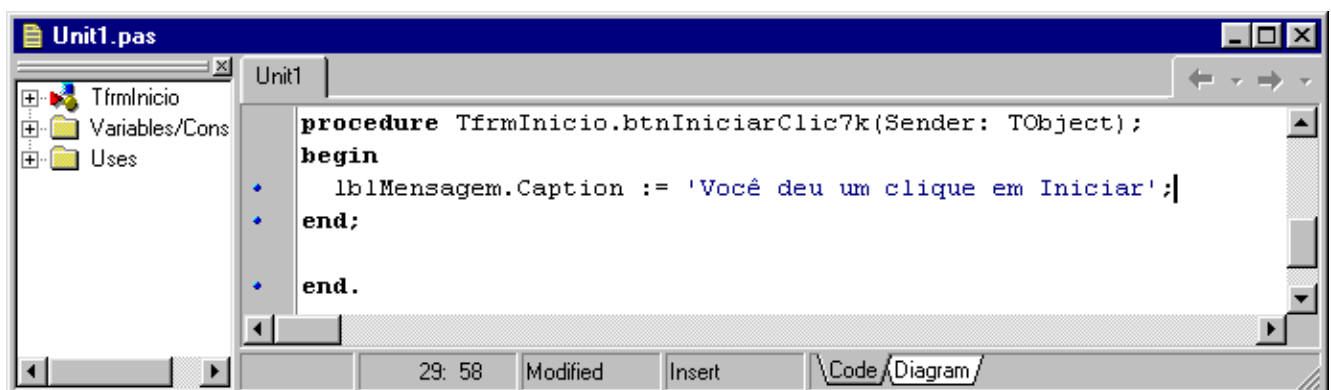


Esta ajuda do Delphi pode ser acionada para qualquer controle ou função, quando digitamos o nome de uma função, ele exibe os parâmetros necessários para a execução desta função.

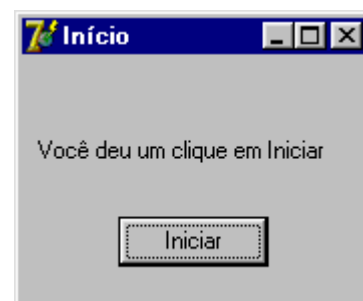
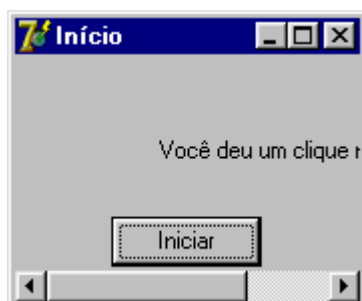
Para escolher uma propriedade do Label lblMensagem, selecione-a com as setas de direção e então pressione Enter, inserindo-a na linha de comando. Ou então, digite a primeira letra da propriedade, selecionando-a.




Continue com o código, seguindo a figura mostrada abaixo. Quando for dado um clique no botão Iniciar, será mostrada a mensagem "Você deu um clique em Iniciar".




Clique sobre o botão **Run** da barra de ferramentas (  ) para que o Delphi inicie a compilação do projeto. Em seguida, dê um clique no botão Iniciar para ver o resultado.

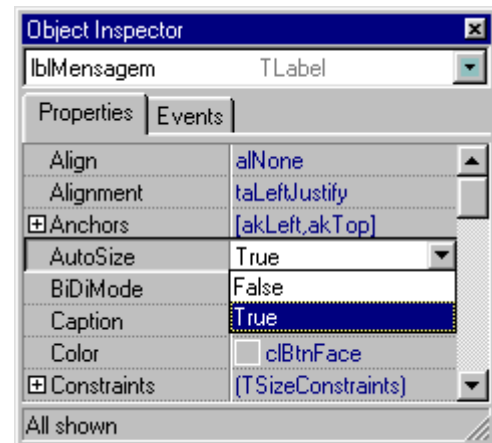
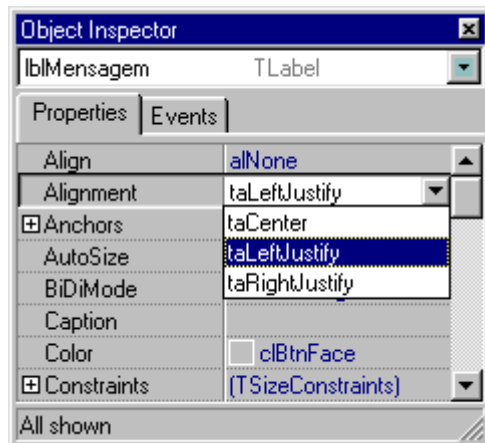


Se o seu formulário se parecer com o apresentado à esquerda, reposicione o seu componente Label. Seu formulário deve ficar como o apresentado a direita.

Finalize a execução do projeto teclando **Alt+F4** ou no botão Finalizar (  ) da barra de título da janela.

Para alternar a visualização entre o Formulário e a janela de código Unit, utilize o botão **Toggle Form/Unit** (  ) na barra de ferramentas, ou pressione a tecla F12.

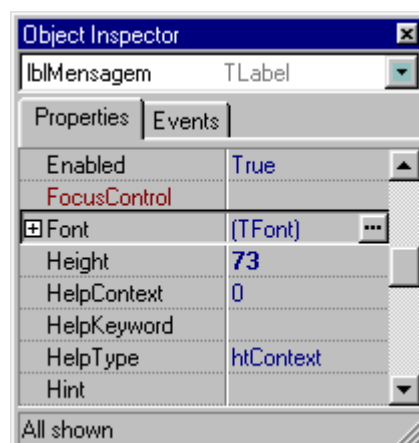
Existem propriedades que possuem valores predefinidos, quando escolhemos a propriedade *Alignment* e damos um clique na seta da caixa de valor, aparecem os tipos de alinhamento para o texto.



Selecione o objeto **IbIMensagem** através da Caixa de Objeto da janela Object Inspector, e altere a propriedade **Alignment** para **taCenter**, para que o texto no TLabel fique centralizado. Altere também a propriedade **AutoSize** para **False**, e no Formulário aumente a largura do TLabel.



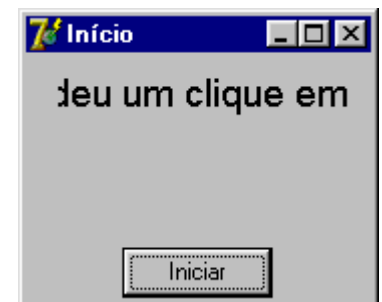
Além das propriedades descritas acima, com padrões pré-definidos, existem outras que possuem inúmeras escolhas, neste caso, ao invés de uma seta, observaremos três pontos, este é o caso da propriedade Font.



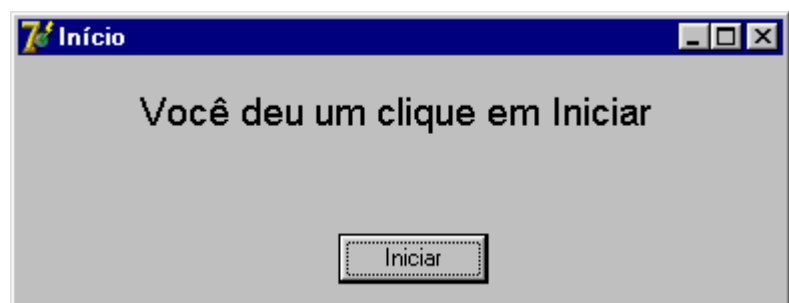
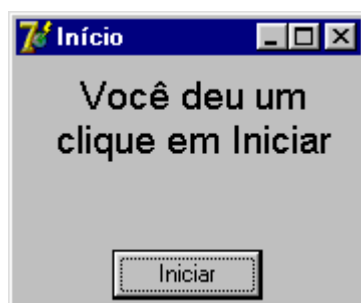
Quando selecionamos os três pontos, aparece uma caixa de diálogo onde escolheremos o formato da fonte que será apresentada a mensagem.



No seu projeto teste as alterações de fonte e observe as mudanças. Na figura ao lado, foi utilizada a fonte Arial com tamanho de 14 pontos. Observe que o texto não coube na área de exibição do TLabel e nem do Formulário, existem duas opções para que este texto apareça integralmente.



A primeira, é alterar para True, a propriedade **WordWrap** do TLabel, esta propriedade insere uma mudança de linha quando o texto atinge a margem direita do objeto. A segunda, é redimensionar os tamanhos da TLabel e do Formulário. Como mostram as figuras a seguir:



Salve o seu projeto, selecionando a opção *Save All* do menu *File*. Primeiramente, será solicitado o nome da Unit, dê o nome de **U\_1oprograma.pas**. Posteriormente, será solicitado o nome do Projeto, dê o nome de **P\_1oprograma.dpr**.