

Universidade Tecnológica Federal do Paraná

UTFPR – Campus Curitiba

Orientação a Objetos

Programação em C++

Grupos de Slides 17B:

Programação Visual - Orientada a Objetos
Gráficos (Formulários, Botões, Caixas de Texto etc) –
*Ambiente RAD - Exemplificado em **C++ Builder**.*

Prof. Jean Marcelo SIMÃO

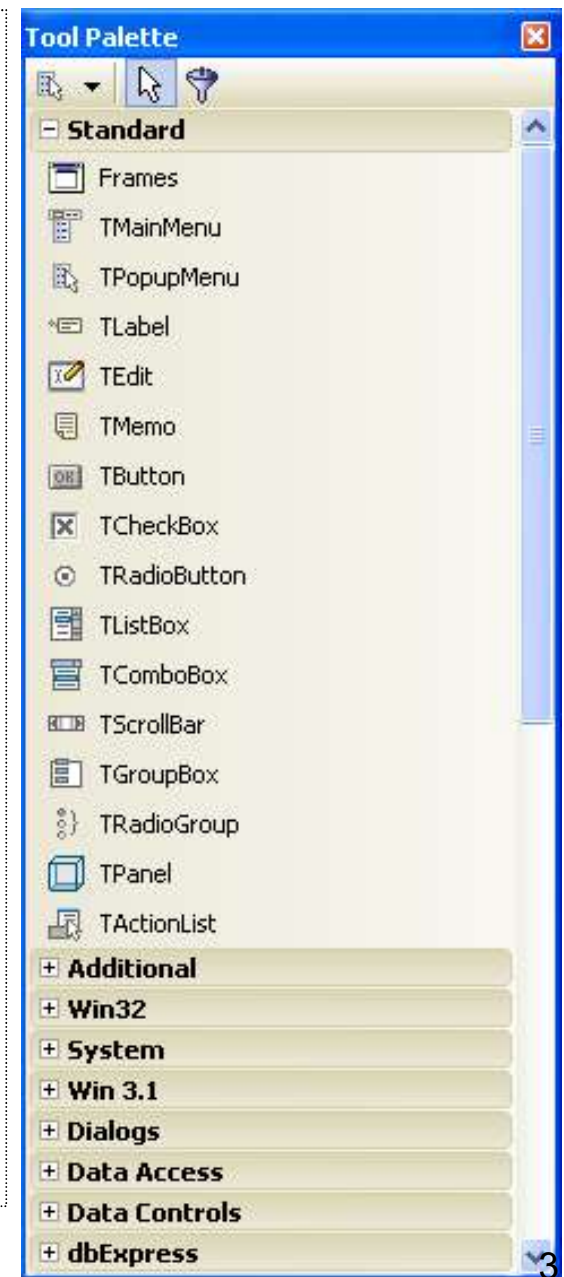
Então aluno monitor (em ~2011): Vagner Vengue

Programação Visual.

- O C++ Builder foi conhecido inicialmente como Borland C++ Builder, mas subsequentemente ele passou para a empresa Embarcadero.
- Estudar os código de exemplo.
 - Entender que objetos gráficos são relacionados com código ‘especial’ em C++.
 - Entender que este código C++ ‘especial’ serve para tratar os eventos sobre objetos gráficos.
 - Entender que, em um projeto correto, o código essencial do sistema é o mais independente possível do código relacionado a objetos gráficos.
- No ProjetoOOJanelaExemplo é tratada a classe Universidade e o seu relacionamento para com a classe Departamento. De forma análoga, tratar as demais classes existentes nas versões precedentes do ‘sistema de universidade’ (Disciplina, Alunos etc).

C++ Builder

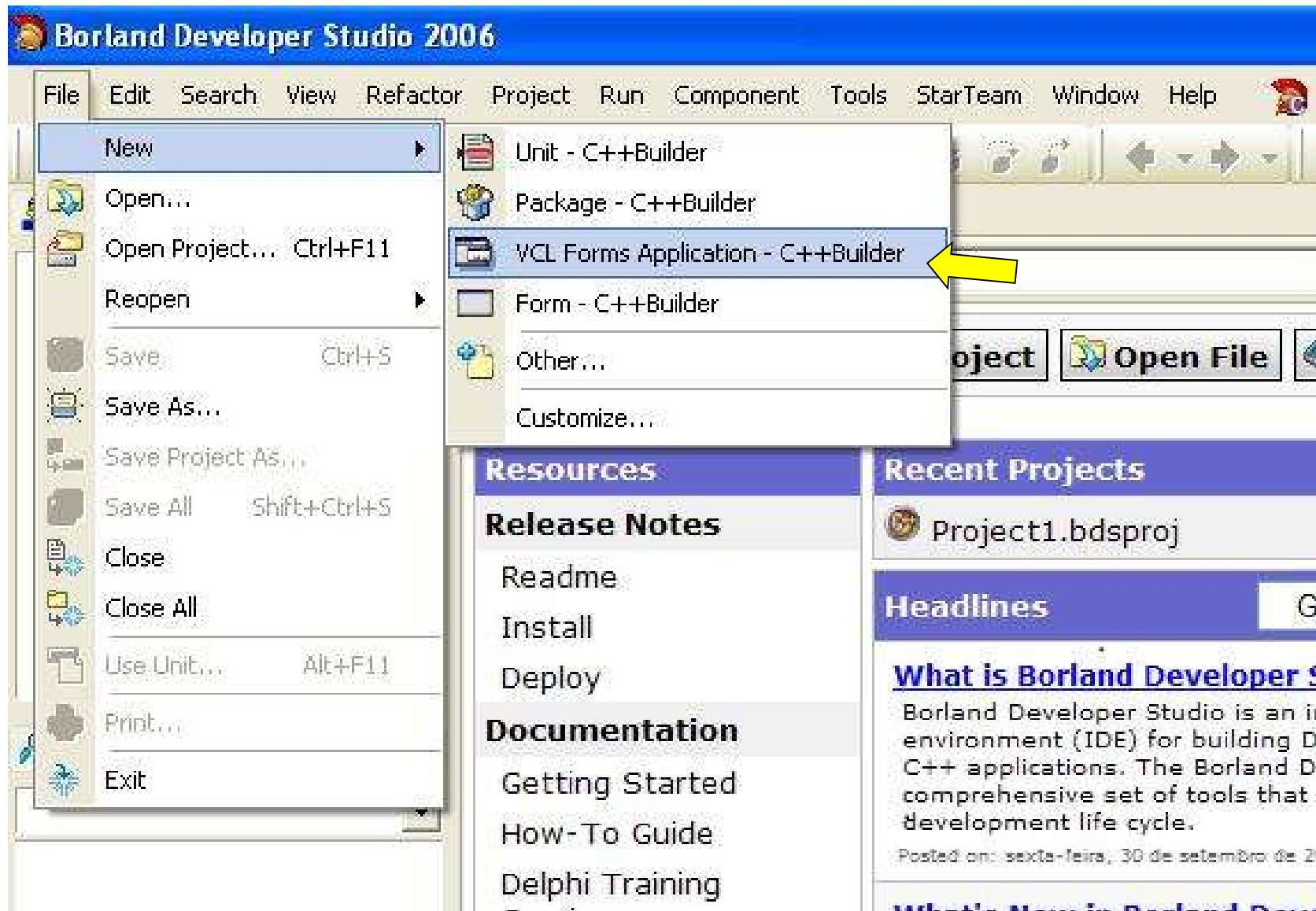
- Ambiente *Rapid Application Development* (RAD).
- Ambiente visual: *TButton*, *TEdit*, *TCheckBox*, *TComboBox*, *TLabel* etc...
- Utiliza o *Framework Visual Component Library* (VCL) para a criação de objetos gráficos. O VCL foi desenvolvido pela empresa Borland.
- Traz uma grande variedade de classes, tanto para o desenvolvimento visual, quanto para o não visual, como acesso a banco de dados, criação de *threads* e comunicação *TCP*.
- Existe uma grande quantidade de “componentes de terceiros” desenvolvidos para Delphi, que compartilha o mesmo framework, como o *JCL* e *JVCL*.



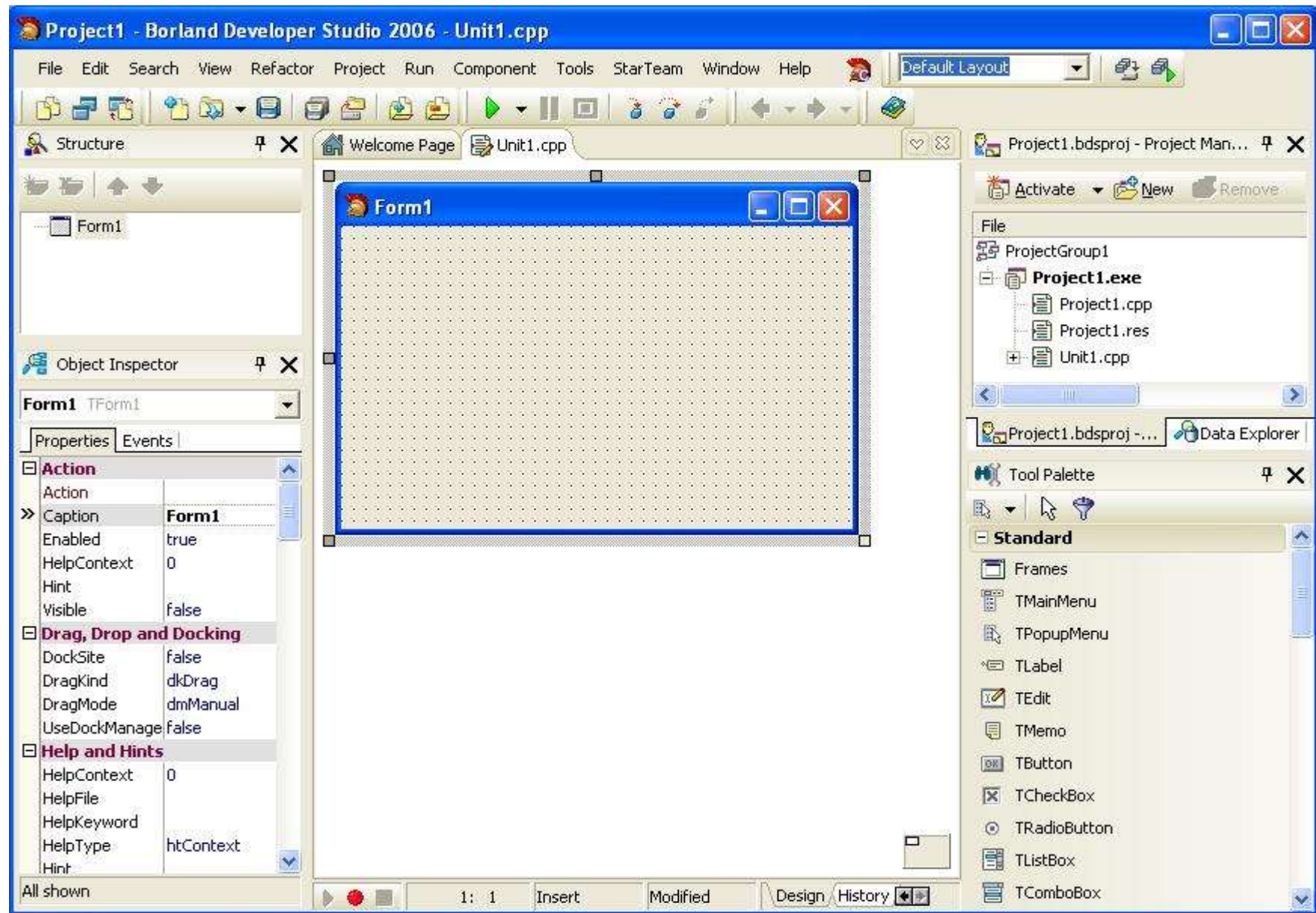
C++ Builder

Exemplo de Introdução

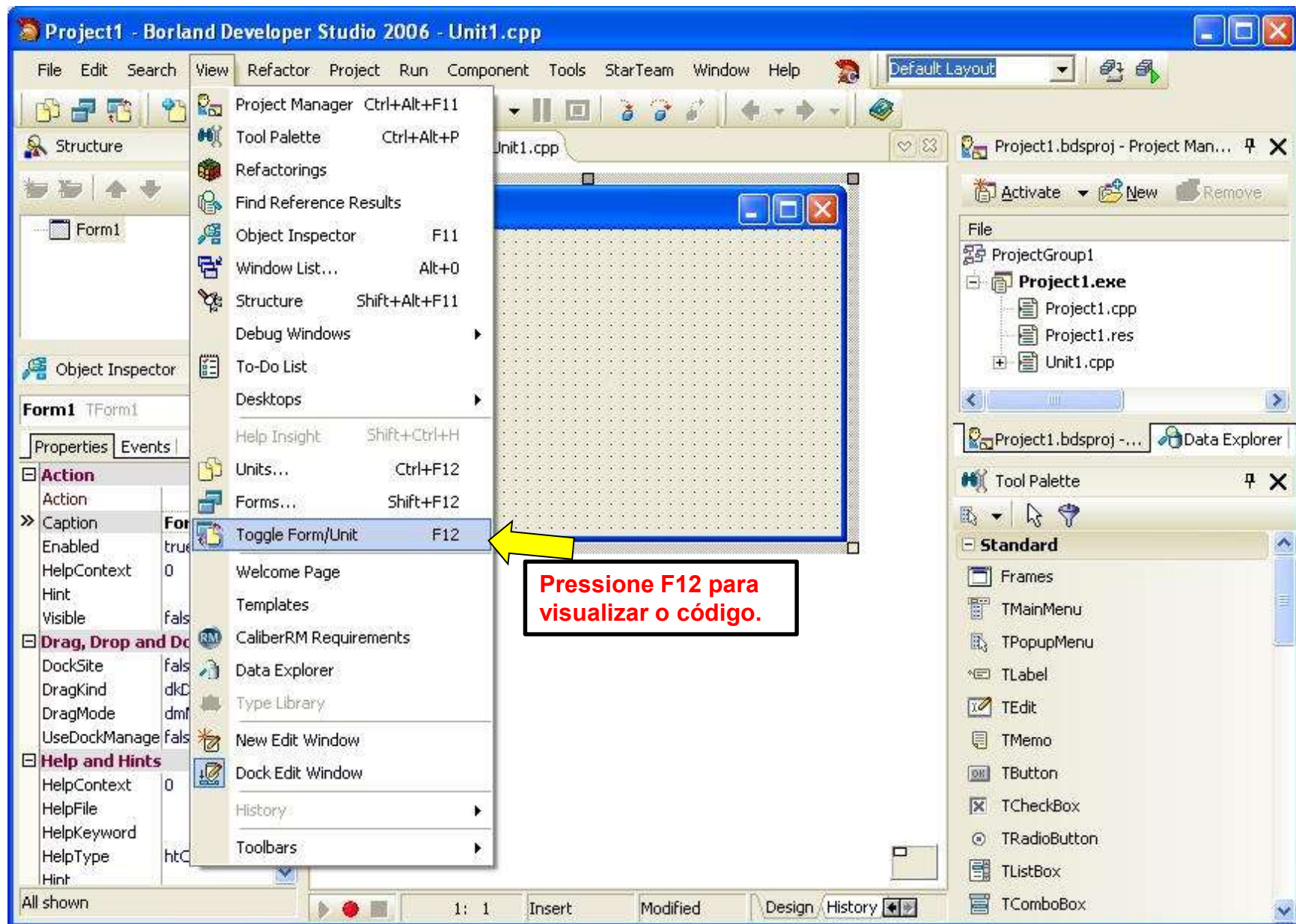
C++ Builder



C++ Builder



C++ Builder



C++ Builder

```
//$ $---- Form HDR ----  
//-----  
  
#ifndef Unit1H  
#define Unit1H  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
//-----  
class TForm1 : public TForm  
{  
    __published:           // IDE-managed Components  
private:                 // User declarations  
public:                  // User declarations  
    __fastcall TForm1(TComponent* Owner);  
};  
//-----  
extern PACKAGE TForm1 *Form1;  
//-----  
#endif
```

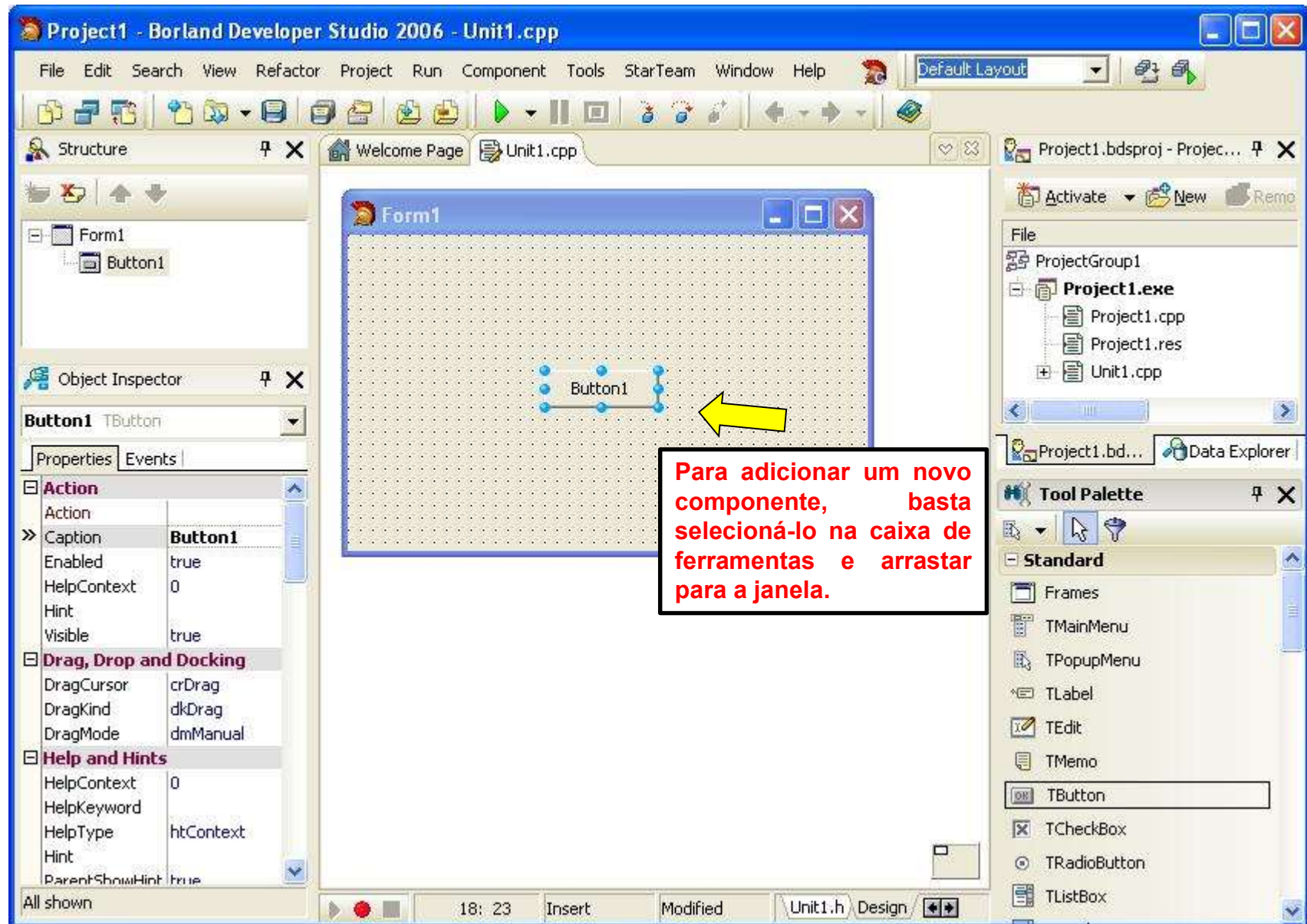
```
//$ $---- Form CPP ----  
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----
```

Código gerado
automaticamente pelo
Borland C++ Builder.

C++ Builder

Adicionando um botão...

C++ Builder



C++ Builder

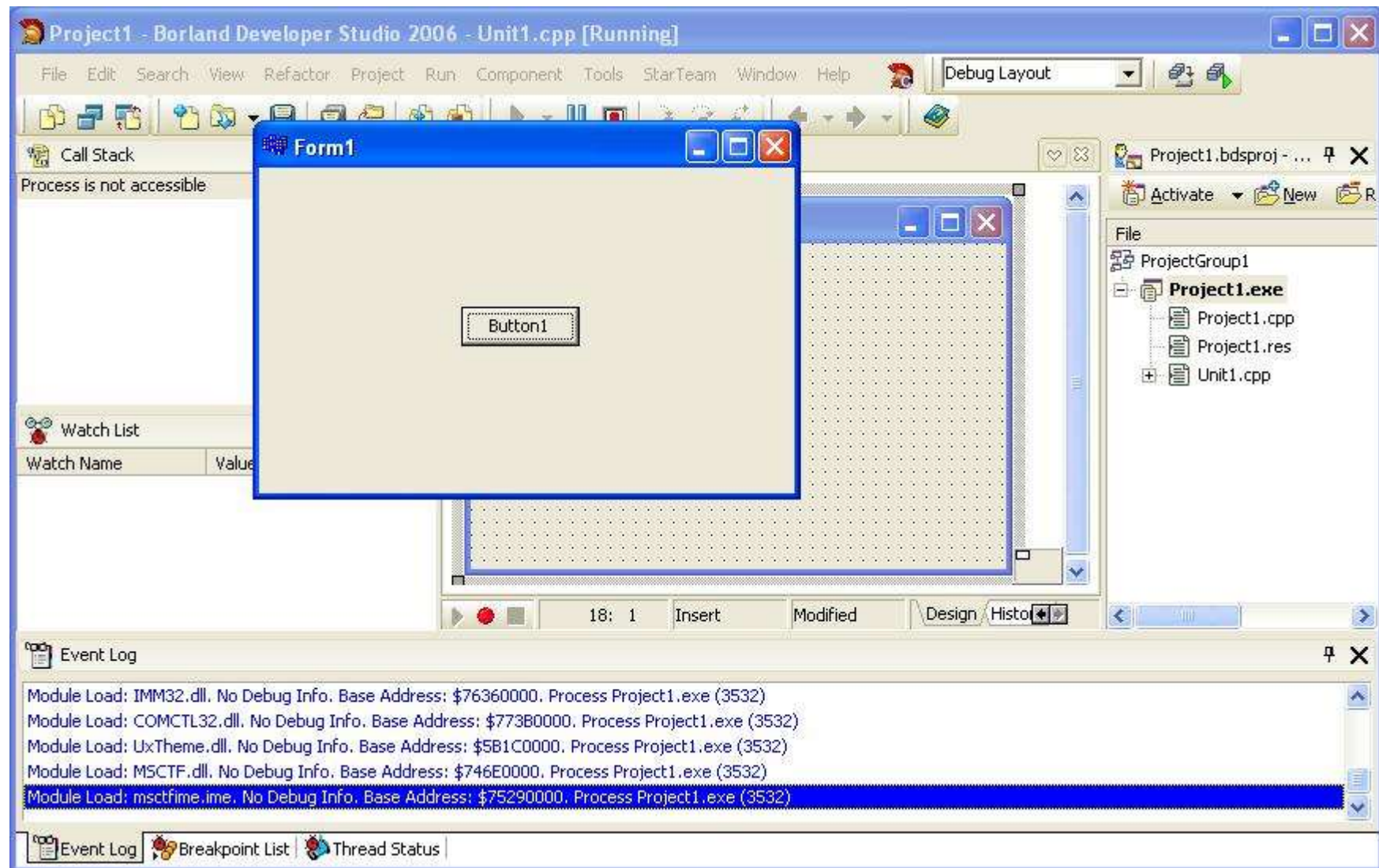
```
//$----- Form HDR ----  
//-----  
  
#ifndef Unit1H  
#define Unit1H  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
//-----  
class TForm1 : public TForm  
{  
  __published:           // IDE-managed Components  
    TButton *Button1;  
private:                // User declarations  
public:                 // User declarations  
  __fastcall TForm1(TComponent* Owner);  
};  
//-----  
extern PACKAGE TForm1 *Form1;  
//-----  
#endif
```

Código adicionado automaticamente para o botão.

```
//$----- Form CPP ----  
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
  : TForm(Owner)  
{  
}  
//-----
```

Código gerado automaticamente pelo Borland C++ Builder.

C++ Builder

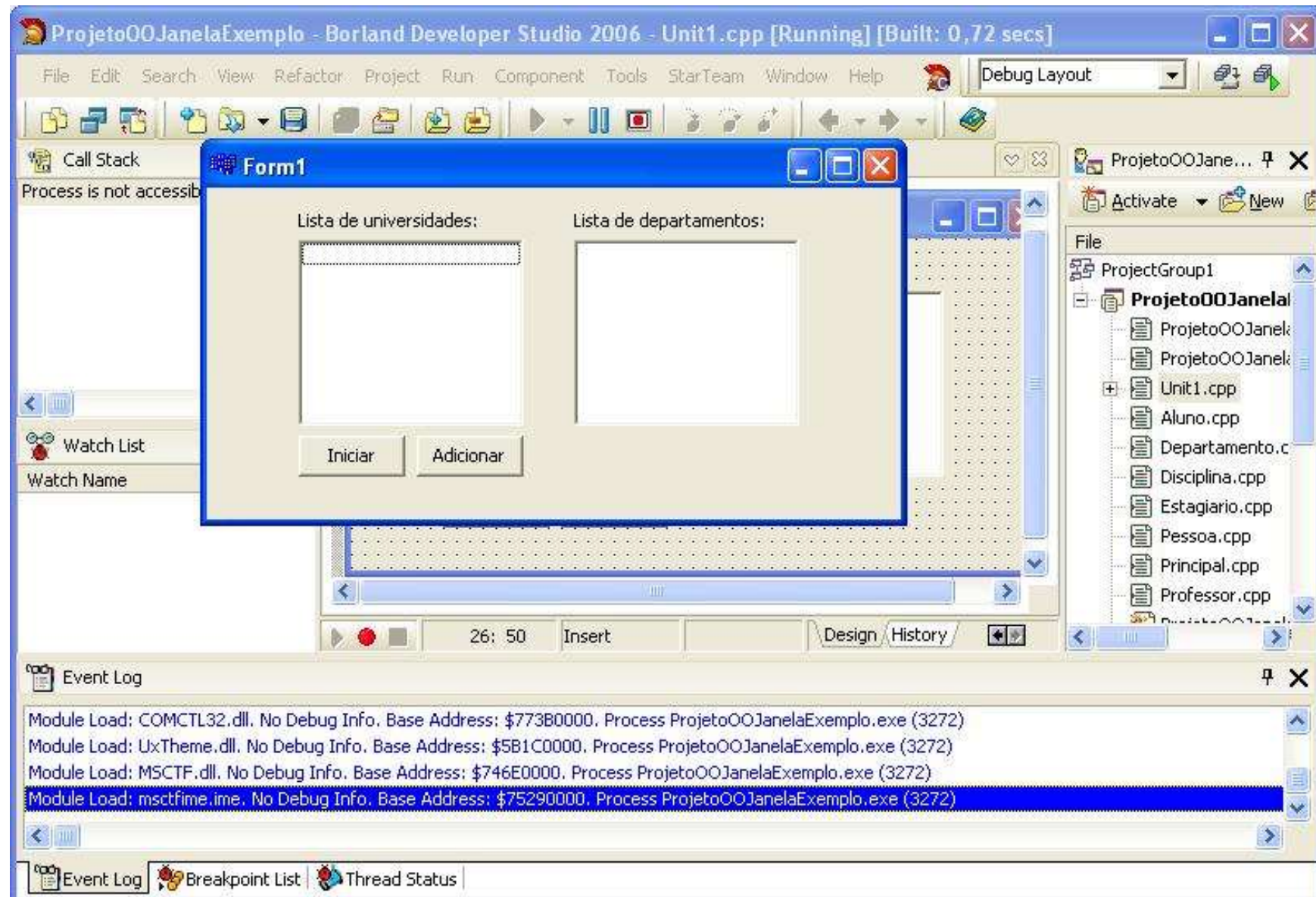


Para compilar e executar, pressione F9.

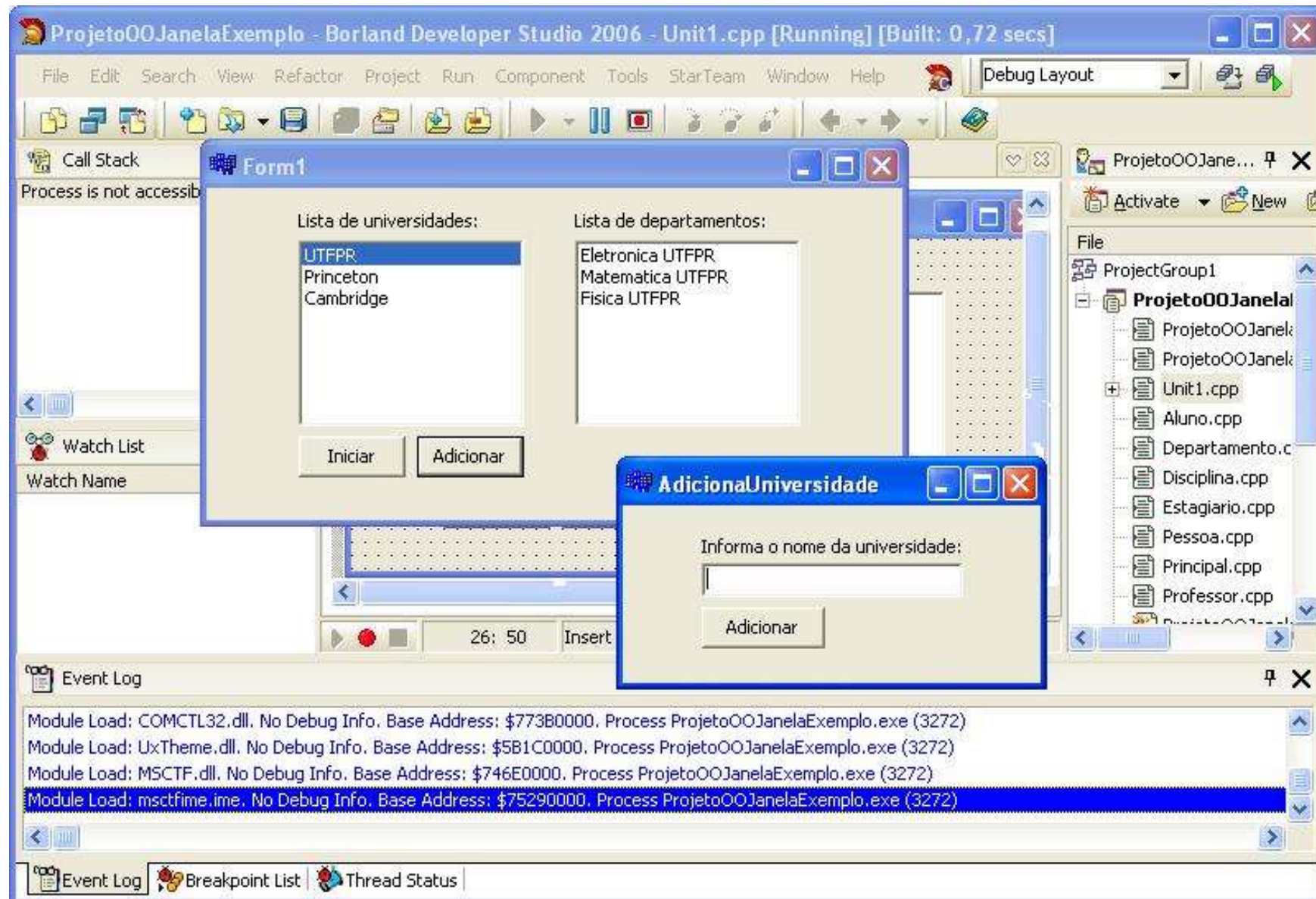
C++ Builder

ProjetoOOJanelaExemplo

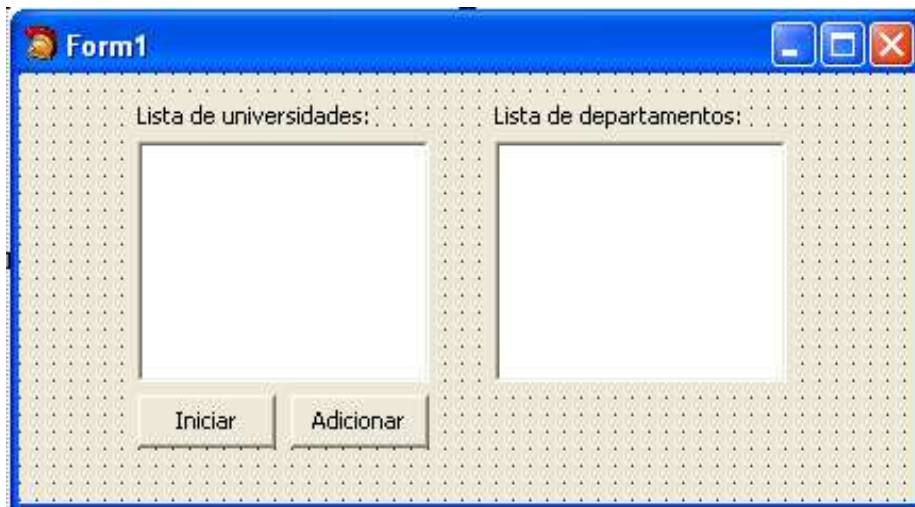
Propriedades



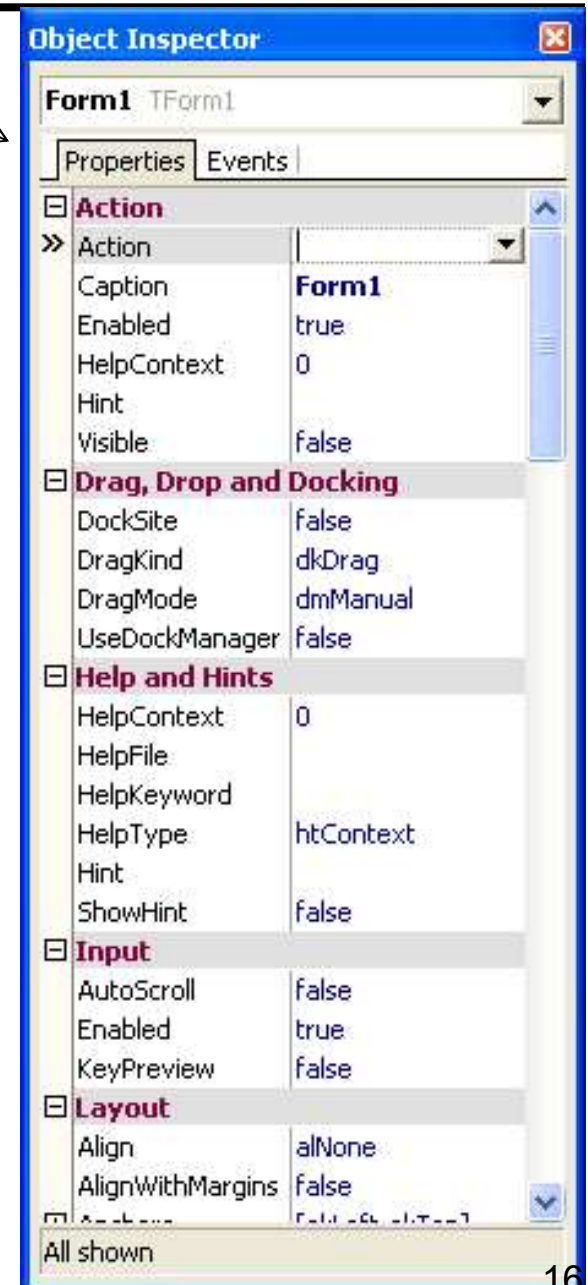
Propriedades



Propriedades



Propriedades



- Através da caixa de propriedades, pode-se definir as características de cada componente, tais como nome, cor, posição na tela, tamanho, imagem de fundo, etc...

- É uma boa prática de programação sempre definir características básicas, como o nome, o título (para telas) ou texto (para botões).

Eventos

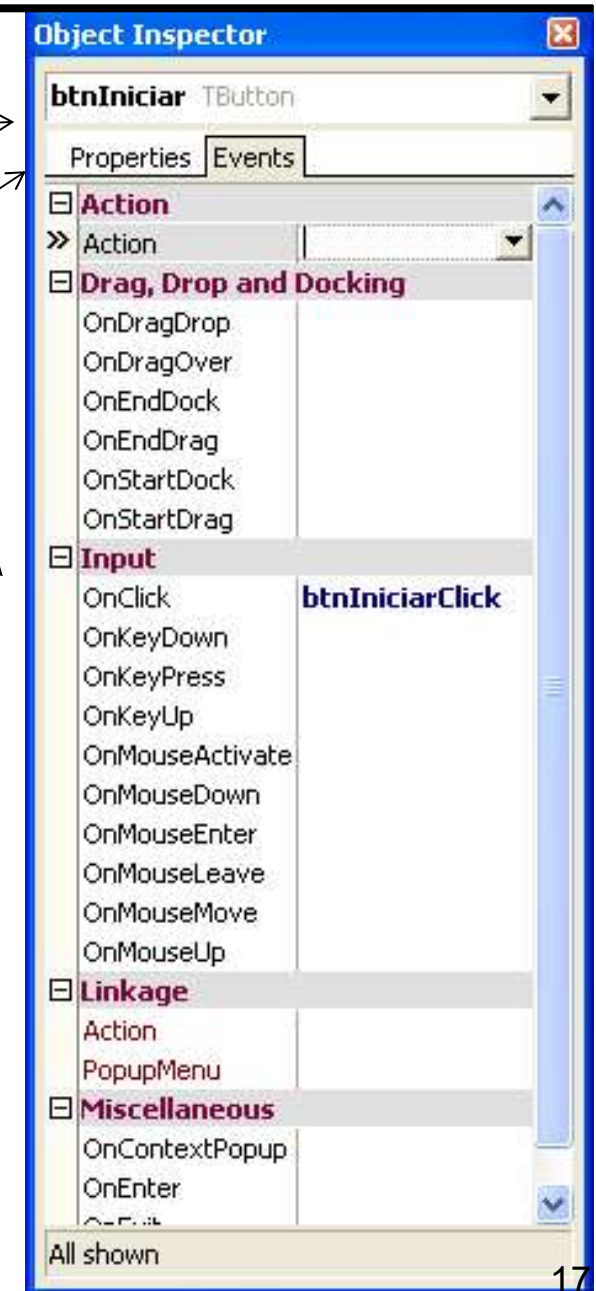


- É uma ação externa ao programa, por exemplo: um click de mouse, uma tecla pressionada, um movimento da tela, uma tela abrindo ou fechando.

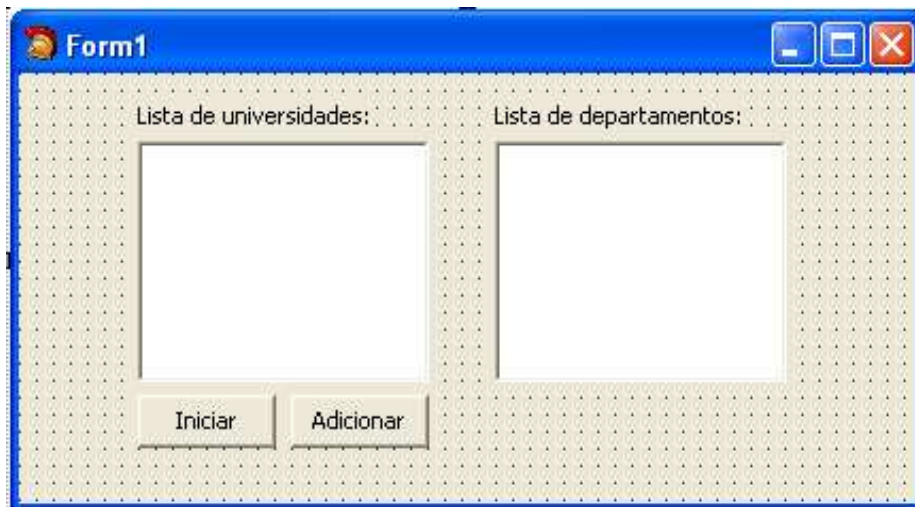
- Com um duplo click sobre o componente é selecionado o evento padrão, onde pode-se inserir o código a ser executado quando o evento ocorrer.

Selecionado

Eventos



Eventos



-Quando o evento é selecionado, o método que trata o evento é adicionado automaticamente à classe.

- O objeto 'Sender' que é passado como parâmetro traz informações sobre o evento ocorrido e pode ser utilizado no código. Um exemplo é a posição (x, y) do mouse quando ele sobrepõe um componente.

Click de mouse do button1

```
void __fastcall TForm1::btnIniciarClick(TObject *Sender)
{
    IstUniver->Items->Clear();
    objeto_principal->IteradorLUniversidades = objeto_principal->LUniversidades.begin();
    while (objeto_principal->IteradorLUniversidades!= objeto_principal->LUniversidades.end())
    {
        string aux;
        aux = (*(objeto_principal->IteradorLUniversidades)->getNome());

        AnsiString NomeUniv;
        NomeUniv = aux.data();

        IstUniver->Items->Add(NomeUniv);
        objeto_principal->IteradorLUniversidades++;
    }
}
```


Agregação de Janelas (objetos)

```
//-----  
class TForm1 : public TForm  
{  
    __published:           // IDE-managed Components  
        TListBox *lstUniver;  
        TListBox *lstDepart;  
        TLabel *Label1;  
        TLabel *Label2;  
        TButton *btnIniciar;  
        TButton *btnAdicionar;  
        void __fastcall btnAdicionarClick(TObject *Sender);  
        void __fastcall FormDestroy(TObject *Sender);  
        void __fastcall btnIniciarClick(TObject *Sender);  
        void __fastcall lstUniverClick(TObject *Sender);  
private: // User declarations  
        Principal* objeto_principal;  
public: // User declarations  
        __fastcall TForm1(TComponent* Owner);  
};  
//-----
```

```
void __fastcall TForm1::btnAdicionarClick(TObject *Sender)  
{  
    frmAdicionaUniver->limpaCampos();  
    frmAdicionaUniver->setPrincipal( objeto_principal );  
    frmAdicionaUniver->Show();  
}
```

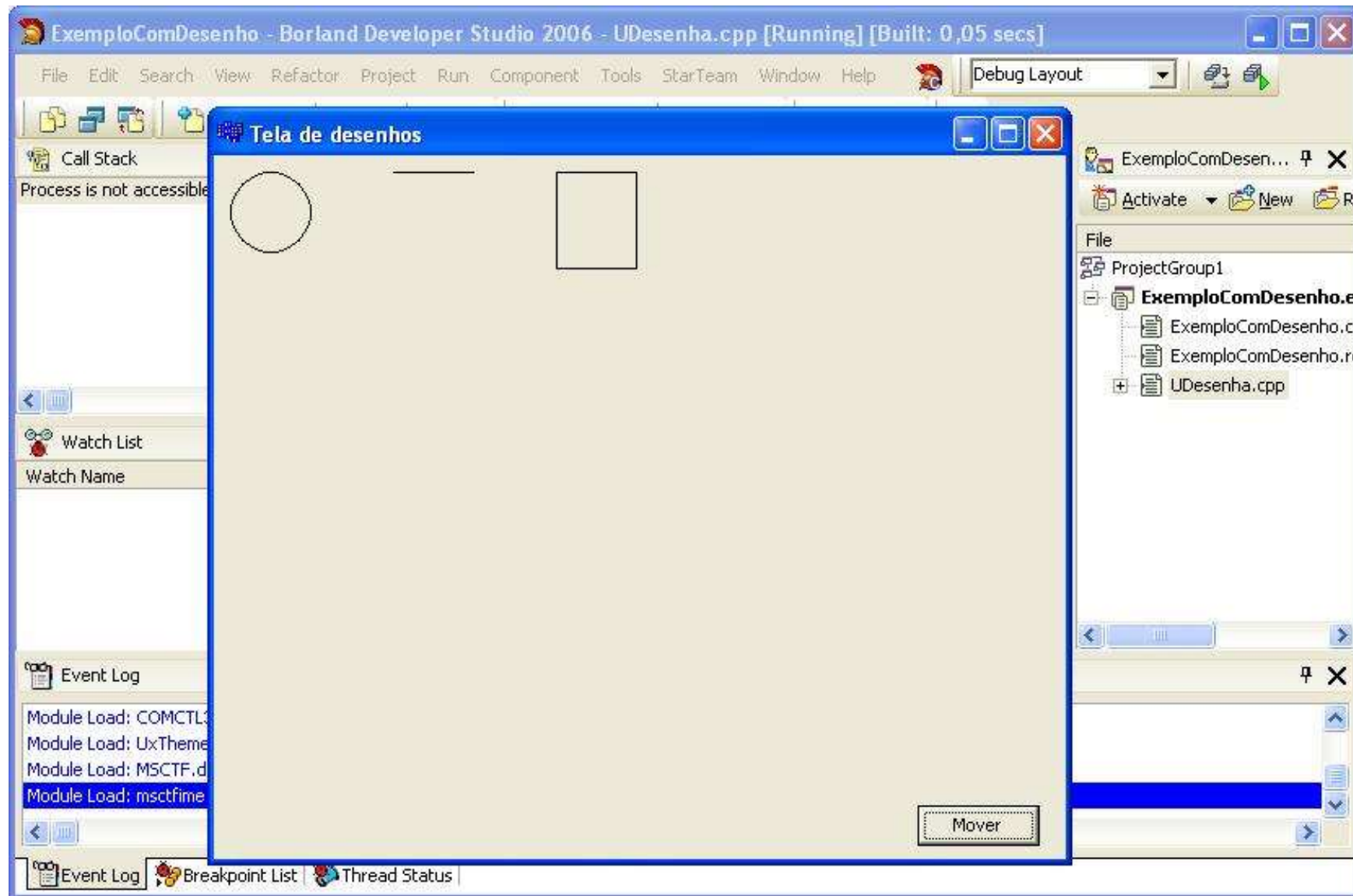
Por padrão, automaticamente é criado um objeto de cada formulário, portanto, só é preciso executar o método Show() de cada uma quando desejado.

Mostra a janela.

C++ Builder

Exemplo com figuras geométricas

Exemplo com desenhos



Exemplo com desenhos

```
///$$---- Form HDR ----  
//-----  
  
#ifndef UDesenhaH  
#define UDesenhaH  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
//-----  
class TfrmDesenho : public TForm  
{  
  __published:           // IDE-managed Components  
    TButton *btnMover;  
    void __fastcall FormPaint(TObject *Sender);  
    void __fastcall btnMoverClick(TObject *Sender);  
private:                   // User declarations  
  int valor_aumentar;  
public:                    // User declarations  
  __fastcall TfrmDesenho(TComponent* Owner);  
};  
//-----  
extern PACKAGE TfrmDesenho *frmDesenho;  
//-----  
#endif
```

Para realizar os desenhos, pode ser utilizado o evento OnPaint() do formulário criado.

```
//-----  
void __fastcall TfrmDesenho::FormPaint(TObject  
*Sender)  
{  
  int x, y;  
  x = 10;  
  y = 10 + valor_aumentar;  
  
  // desenha uma elipse;  
  this->Canvas->Ellipse(x, y, x+50, y+50);  
  x += 100;  
  
  // move para uma posição da tela;  
  this->Canvas->MoveTo(x, y);  
  // desenha uma reta;  
  this->Canvas->LineTo(x+50, y);  
  
  x += 100;  
  // desenha um retângulo;  
  this->Canvas->Rectangle(x, y, x+50, y+60);  
}  
//-----  
void __fastcall TfrmDesenho::btnMoverClick(TObject  
*Sender)  
{  
  valor_aumentar += 10;  
  this->Repaint();  
}  
//-----
```

Bibliografias

- Para mais detalhes sobre Borland C++ Builder, consulte:

<http://www.functionx.com/bcb/index.htm> (acesso em 02/08/2010).

<http://www.embarcadero.com/products/cbuilder> (acesso em 02/08/2010).

<http://en.wikipedia.org/wiki/C%2B%2BBuilder> (acesso em 02/08/2010).