Projeto Dinâmico em Delphi

Mário Leite

...

Nos anos 1980 uma guerra não declarada entre programadores eclodiu no Brasil: era um embate intenso com os defensores do Visual Basic (VB) de um lado e os defensores do Delphi do outro; o primeiro da Microsoft Corporation e o segundo criado pela Borland International; ambos, ferramentas do tipo RAD (**R**apid **A**pplication **D**evelopment) para desenvolvimento rápido de programas orientados a eventos. Tanto o VB quanto o Delphi rodavam no Windows sob IDE's, criando sistemas bem inteligentes e muito interativos com o programador, mas com linguagens embarcadas bem diferentes; enquanto o VB usava o Quick Basic (Basic compilado) o Delphi usava o Object Pascal (Pascal orientado a objetos). Assim, essas duas ferramentas passaram a ser as preferidas pelos programadores que desejavam criar aplicativos com elementos em interfaces que executavam ações ao serem acionados através de algum tipo de evento: *click*, *duplo click*, *arrasto*, etc). Assim, para criar uma aplicação com eventos disparados sobre elementos de interfaces (geralmente formulários) o programador tinha que fazer isto em duas etapas bem distintas:

1) Criar a(s) interface(s) com os elementos que geravam os eventos.
2) Escrever os códigos das ações associadas aos eventos.

A primeira etapa era a criação das interfaces que interagiam diretamente com o usuário final do programa, e a segunda era a codificação dos eventos. Nesta segunda etapa era onde residia a briga entre os defensores dessas duas ferramentas. Enquanto os *VBistas* defendiam a ideia de que o VB era melhor, por ser mais fácil de usar (e por ser a primeira ferramenta RAD criada), os *Delphistas* "esnobavam" os adversários dizendo que o Delphi era melhor por ser orientado a objetos, o que o tornava mais profissional. Na verdade, as duas ferramentas eram (e são) excelentes, pois o VB também passou a ser orientado a eventos com o decorrer do tempo. No entanto, um problema persistia para ambas as ferramentas: a primeira etapa: muito trabalhosa; o que demandava um certo tempo (e paciência) do desenvolvedor para instanciar (colocar manualmente) os elementos (objetos) nas interfaces. Então, surgiu a possibilidade de se criar interfaces dinamicamente: sem precisar colocar manualmente os objetos diretamente no formulário, usando apenas código puro. Era esta a vantagem que o Delphi tinha sobre o VB, com mais facilidade. O projeto (muito simples) aqui apresentado é para criar uma interface de maneira dinâmica, sem precisar instanciar manualmente os elementos; com apenas um formulário, contendo dois botões de pressão (***TButton***) com evento ***Click*** associado a eles:

- Um botão rotulado como "Arredondar"
- Um botão rotulado como "Finalizar"

Para criar dinamicamente um formulário com componentes e eventos em tempo de execução no Delphi (sem usar o *designer* da IDE para arrastar e soltar os elementos) todo o código é colocado no método ***TForm1.FormCreate*** (o primeiro método embutido a ser executado). Neste exemplo o objetivo é calcular o tempo de processamento para realizar a contagem de 1 até 50 milhões com um *loop* ***for***, e arredondando o resultado para duas casas decimais. A **figura 1a** mostra a criação da interface ao rodar o programa, e a **figura 1b** a saída (resultado) numa janelinha da função **ShowMessage(**) ao clicar no botão **[Arredondar]**; e um *click* no botão **[Finalizar]** encerra o programa. A sequência para criar a interface e as procedures-eventos, dinamicamente, deve ser a seguinte:

1. Crie um novo projeto **VCL** no Delphi (*File → New → VCL Forms Application*).
2. Apague todos os componentes do formulário (se houver algo no *designer*).
3. Substitua o código da **Unit1.pas** pelo código acima.
4. Pressione **F9** para compilar e executar.

Embora a criação de interfaces dinamicamente pode não ser muito eficiente do ponto de vista prático, pois as ferramentas RAD foram criadas exatamente para minimizar a escrita de códigos, este é um recurso oferecido pelo Delphi que o torna uma das melhores ferramentas de desenvolvimento.

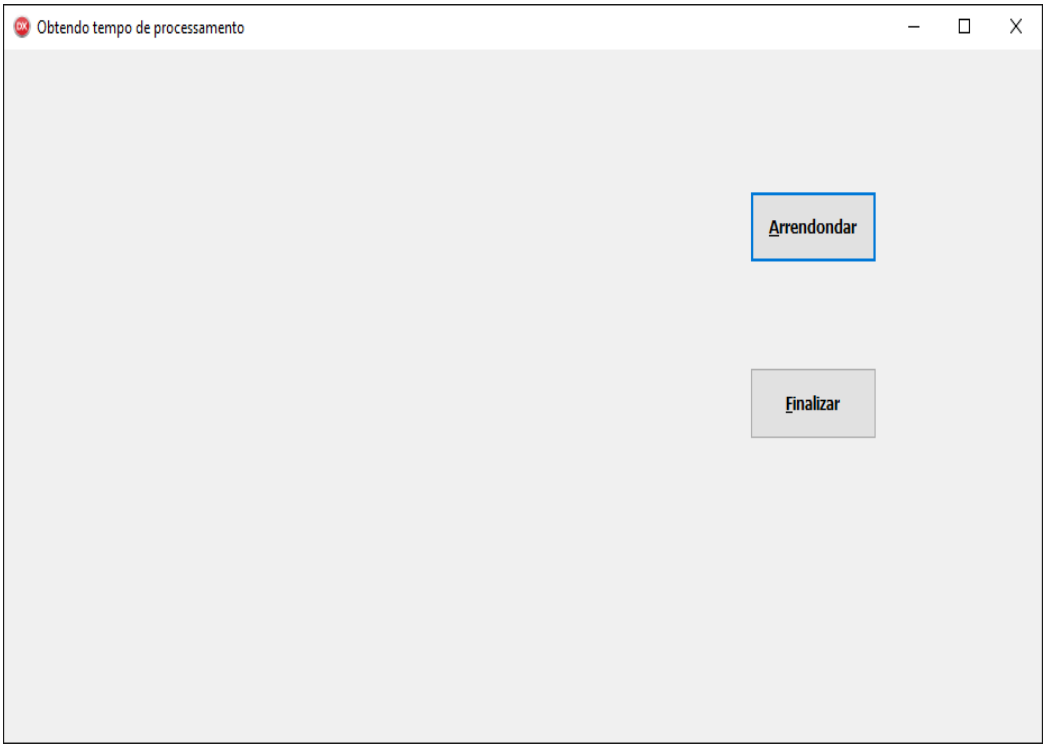

Figura 1a - Interface da aplicação criada dinamicamente ao rodar o programa

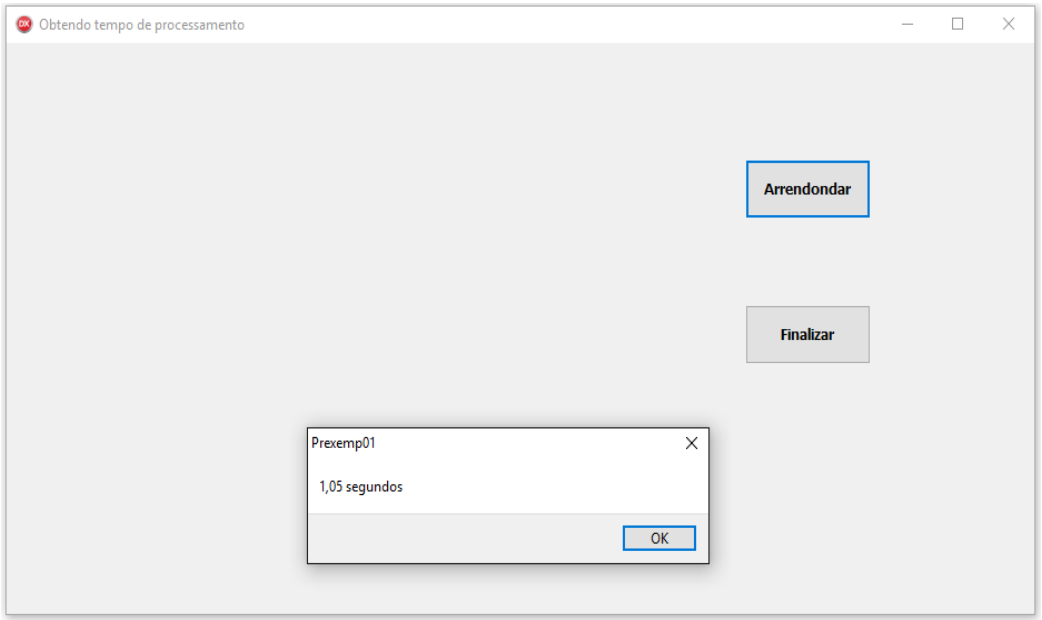

Figura 1b - Resultado: ao clicar no botão [Arredondar]

```
unit Unit1;


interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Math;

type
  TForm1 = class(TForm)
    procedure FormCreate(Sender: TObject); //cria a interface/eventos dinamicamente
  private
    { Private declarations }
    procedure BtnArredondarClick(Sender: TObject); //evento do botão [Arredondar]
    procedure BtnFinalizarClick(Sender: TObject);  //evento do botão [Finalizar]
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation
{$R *.dfm}

//============= CRIA OS COMPONENTES DINAMICAMENTE =============
procedure TForm1.FormCreate(Sender: TObject);
var
  BtnArredondar, BtnFinalizar: TButton;
begin
  //Configura o formulário principal
  Caption := 'Obtendo tempo de processamento';
  Width := 400;
  Height := 250;
  Position := poScreenCenter;

  //Cria o botão "Arredondar"
  BtnArredondar := TButton.Create(Self);
  with BtnArredondar do
     begin
       Parent := Self; //associa ao formulário
       Left := 120;
       Top := 50;
       Width := 150;
       Height := 40;
       Caption := 'Arredondar';
       OnClick := BtnArredondarClick; //associa o evento Click
     end;

  //Cria o botão "Finalizar"
  BtnFinalizar := TButton.Create(Self);
  with BtnFinalizar do
     begin
       Parent := Self;
       Left := 120;
       Top := 120;
       Width := 150;
       Height := 40;
       Caption := 'Finalizar';
       OnClick := BtnFinalizarClick; //associa o evento Click
  end;
end;
```

```
//============= EVENTO DO BOTÃO "ARREDONDAR" =============
procedure TForm1.BtnArredondarClick(Sender: TObject);
var
  T1, T2: Cardinal;
  j: Integer;
  TempoSegundos: Real;
begin
  T1 := GetTickCount(); //marca o tempo inicial (em milissegundos)

  //Loop de 1 a 50.000.000 (apenas para teste de processamento)
  for j := 1 to 50000000 do
  begin
    //nada é feito aqui, apenas conta o tempo
  end;

  T2 := GetTickCount(); //marca o tempo final
  TempoSegundos := (T2 - T1) / 1000; //converte para segundos
  TempoSegundos := RoundTo(TempoSegundos, -2); //arredonda para 2 casas decimais

  //Exibe o resultado
  ShowMessage(FloatToStr(TempoSegundos) + ' segundos');
end;

//============= EVENTO DO BOTÃO "FINALIZAR" =============
procedure TForm1.BtnFinalizarClick(Sender: TObject);
begin
  Application.Terminate; //encerra o programa
end;

end.
```