

## Short-term interest rates

3-month rates, in % p.a.



# Gráficos

## Dicas práticas para a criação de gráficos em aplicações

Criado por Descartes para desenhar um raciocínio, a utilização de representação por Gráficos (gráficos), possibilita transmitir um significado de planilhas ou tabelas complexas de uma forma mais eficiente e simples. Em nossos sistemas, com informações armazenadas em banco de dados, a necessidade de mostrar as informações de uma forma de fácil compreensão muitas vezes nos obriga a utilizarmos esses recursos.

Baseados nessa metodologia veremos no decorrer deste artigo como construir gráficos a partir de informações armazenadas em um banco de dados. Utilizaremos o Delphi 7, que fará acesso através dos componentes da paleta dbExpress à base de dados *Employee.fdb*, que acompanha a instalação do Firebird. Para criar os gráficos utilizaremos o componente *DBChart* da paleta *Data Controls*.

Nos gráficos que iremos criar, vamos utilizar duas formas distintas de exibir os registros. Na primeira iremos exibir os re-

gistros absolutos que serão retornados de uma consulta SQL. E na segunda iremos parametrizar a quantidade de registros a ser exibida, onde o restante das informações não atingidas pelo parâmetro de limite da busca será mostrado em uma categoria "OUTROS" criada no gráfico com a devida porcentagem de valores a quantidade correspondida.

### Criando a aplicação

Utilizaremos no decorrer deste artigo para criação do exemplo de gráficos o Delphi 7, mas qualquer outra versão poderá ser utilizada. Crie uma nova aplicação no menu *File\New>Application*. Altere o nome do formulário para "frmPrincipal" e defina o *Caption* do formulário principal para "Gráficos com Win32". Salve a Unit como "uGráficos.pas" e o projeto salve como "Gráficos.dpr".

Crie agora um novo formulário, a partir do menu *File\New>Form*, que utilizaremos para exibir todos os registros



### Maikel Marcelo Scheid

([maikelscheid@gmail.com](mailto:maikelscheid@gmail.com))

é técnico em Informática com ênfase em Análise e Programação de Sistemas. Atua na área de Desenvolvimento de Softwares em Delphi para plataforma Win32 e .NET com banco de dados Firebird e MSSQL. É membro da Equipe Editorial ClubeDelphi.

resultantes de uma pesquisa SQL à base de dados sem nenhum parâmetro de limitação de registros. Altere o nome para "frmCustomerCountry" e a propriedade *Caption* para "Clientes por País". Salve a Unit do formulário como "uCustomerCountry.pas".

Adicione também, através do menu *File|New>Data Module*, um Data Module ao projeto, que será utilizado para configuração do componente de conexão ao banco de dados e também dos componentes de consulta. Altere o nome do Data Module para "DM" e salve a *Unit* como "uDM.pas".

Adicione ao Data Module um componente *SQLConnection* ("sqlConexao") da paleta *dbExpress*. Altere a propriedade *LoginPrompt* para *False* e com um duplo clique abra o diálogo de configuração da conexão. Na caixa de diálogo que aparece clique no botão "+" e adicione uma nova conexão. Selecione, em *Driver Name*, a opção *Interbase* e em *Connection Name* digite "Employee" e confirme.

De volta ao diálogo faremos as configurações do caminho da base de dados e também do usuário e senha de acesso. Em *Database* informe o caminho para o banco de dados *Employee.fdb*, normalmente localizado em *C:\Arquivos de Programas\Firebird\Firebird\_2\_0\exemplos*, que acompanha a instalação do Firebird. Também é necessário incluir o nome de usuário em *User\_Name* e a senha padrão de acesso em *Password*. Caso não tenha modificado o usuário e senha do Firebird eles são *SYSDBA* e *masterkey*, respectivamente.

Também é de fundamental importân-

cia que se defina o dialeto de conexão em *SQLDialect* como dialeto 3. Defina também o *ServerCharSet* como *Win1252* (Figura 1).

## Criando e configurando a consulta de registros

Adicione ao Data Module um componente *SQLDataSet* ("sqlCustomerCountry") também da paleta *dbExpress*. Ligue a propriedade *SQLConnection* ao componente "sqlConexao" e altere a propriedade *Name* para "sqlCustomerCountry" e adicione a seguinte instrução *SQL* à propriedade *CommandText*:

```
select country.country as pais,
count(customer.cust_no) as clientes from
country, customer where customer.country =
country.country group by country.country
```

Na instrução *SQL* anterior estamos buscando dois campos dos registros do banco de dados. Um com o nome do país e outro com a contagem da quantidade de registros em cada país da seleção.

Com um duplo clique sobre o componente *SQLDataSet* abra o *Fields Editor* e com o botão direito do mouse selecione a opção *Add All Fields* para adicionar todos os campos retornados pela *SQL*.

## Criando o gráfico

Para criar o gráfico, volte ao *frmCustomerCountry* e adicione primeiramente um componente *Panel* da paleta *Standard*, alterando a propriedade *Align* para *alTop* e exclua o texto da sua propriedade *Caption*. Adicione ao formulário um componente *DBChart* da paleta *Data Controls* alterando

sua propriedade *Align* para *alClient*.

---

Nota: Não esqueça de adicionar o Data Module ao *Uses* do formulário para que possa ter acesso aos componentes de conexão com o banco. Para isso selecione *File|Use unit...* ou pressione *ALT + F11*.

---

Agora para realizar a configuração do componente, dê um duplo clique sobre o mesmo para que possa acessar suas propriedades de configuração. No diálogo de configuração do gráfico, a primeira escolha que temos a fazer é quanto ao tipo de gráfico. Utilizando o botão *Add*, selecione primeiramente o gráfico do tipo *Pizza* ("Pie") e confirme.

Aparecerá uma nova série selecionada, altere o nome da mesma para "Pie" utilizando o botão *Title*. Repetindo o processo selecione também o gráfico do tipo *Bar* e confirme. Altere seu nome para "Bar".

Pronto, você terá disponível agora dois tipos de gráficos (Figura 2).

Com o gráfico *Pie* selecionado, clique na aba *Series* ao lado da aba *Chart* e, dentre as abas de configuração, selecione *Data Source*. Nesta aba iremos configurar o acesso do componente de gráfico ao componente *sqlCustomerCountry* onde preparamos a instrução de consulta à base de dados.

Logo na primeira caixa de opções, selecione *DataSet*, a qual irá habilitar as demais opções de configuração. Na opção *DataSet* abaixo, ao clicar deverão aparecer os componentes aos quais o *DBChart* possui compatibilidade de conexão, no

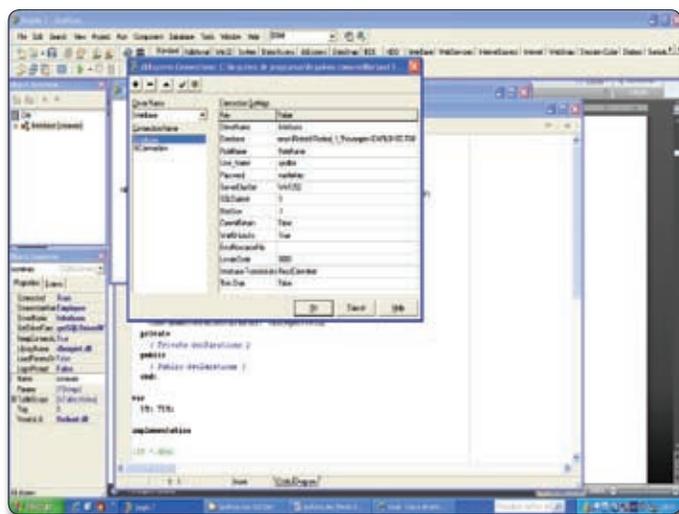


Figura 1. Configurando a conexão à base de dados

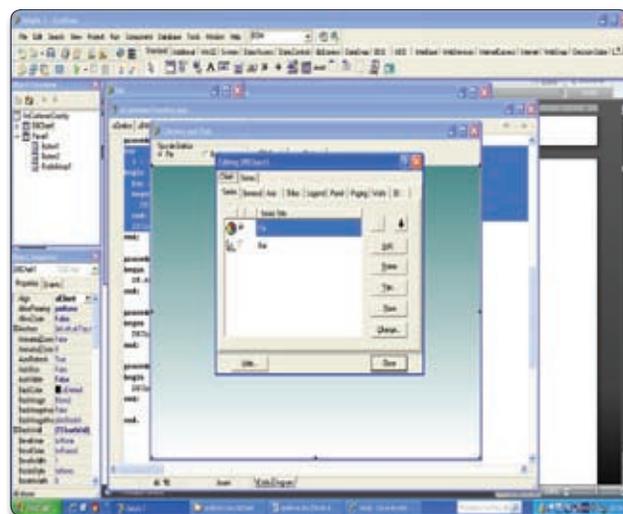


Figura 2. Adicionando os tipos de Gráficos

caso desse exemplo, irá aparecer somente o *sqlCustomerCountry* que configuramos no *DM*, que ao selecioná-lo irá habilitar outras duas configurações que precisam ser feitas.

Em *Labels* selecione *PAIS*, que corresponde ao campo onde serão informadas as descrições de países para exibição no gráfico. Na propriedade *Pie* selecione *CLIENTES*, que corresponde ao campo de contagem de ocorrências dos clientes para cada país que será exibido, de acordo com a instrução *SQL* configurada anteriormente (**Figura 3**).

Realizada a configuração da série *Pie*, selecione agora a série *Bar* nesta mesma aba *Series*. Na primeira opção de escolha

selecione novamente *DataSet* e ligue ao *sqlCustomerCountry* configurado no *Data Module*. As propriedades de configuração deste tipo de gráfico serão um pouco diferentes quanto à configuração do primeiro, pois este tipo de gráfico está solicitando um valor para a coluna *X* e outro valor para a coluna *Bar*.

Na propriedade *Labels* não se tem diferença, portanto selecione novamente o campo *PAIS*. Agora nas outras duas propriedades, iremos informar um valor apenas ao campo *Bar*, selecionando o campo *CLIENTES* (**Figura 4**).

Se fossemos informar o campo *CLIENTES* também na propriedade *X*, nosso gráfico iria agrupar os valores idênticos,

ou seja, se tivéssemos dois países com cinco clientes em cada um deles, uma única barra seria exibida ao valor correspondente e somente na legenda que apareceriam os nomes de ambos os países com o mesmo número de clientes.

### Configurações e aparência do Gráfico

Veremos agora como configurar algumas das muitas opções de configuração das propriedades do componente *DB-Chart*. Logo na aba *Marks*, dentro da aba *Series*, que corresponde especificamente ao tipo do gráfico selecionado, teremos as opções de configuração do corpo do gráfico, onde você poderá escolher de que forma serão exibidos os *Labels* para

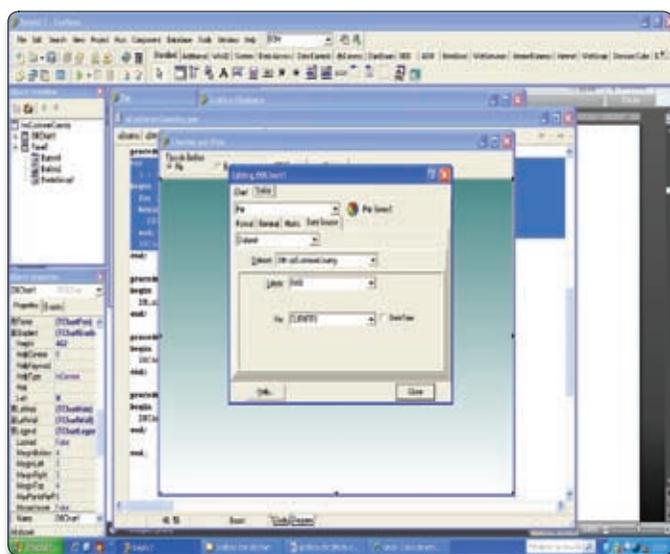


Figura 3. Configurando as fontes de informações do gráfico

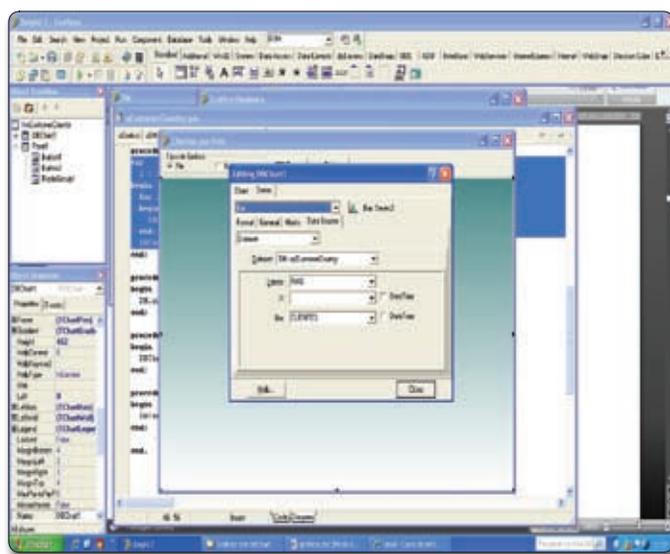


Figura 4. Configurando as fontes do gráfico de barras

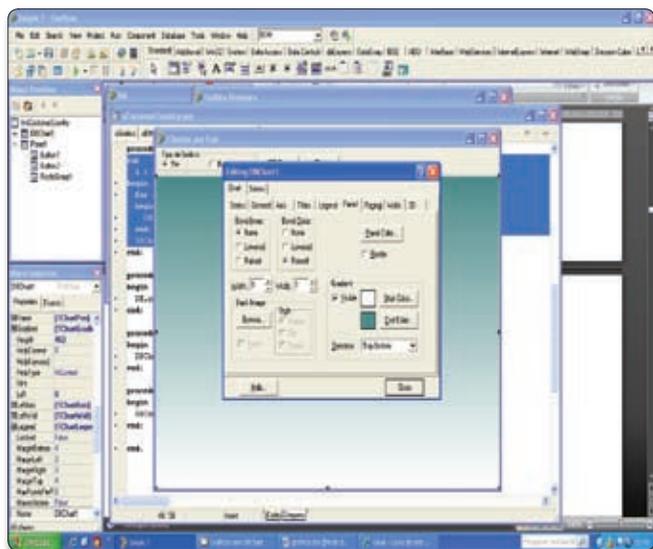


Figura 5. Configurações de fundo do corpo do gráfico

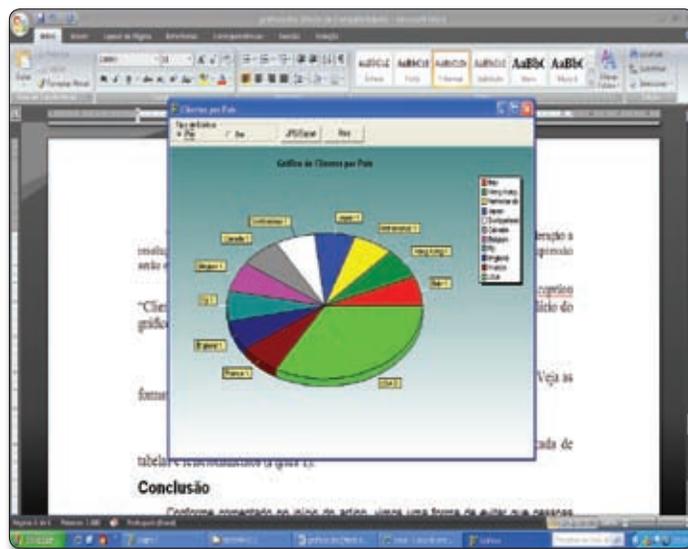


Figura 6. Exibição do gráfico de clientes por cidade

cada barra do gráfico. Se desejar, poderá exibir somente o valor (“Value”) ou a descrição e o valor (“Label and Value”), entre outras.

Poderá também alterar a cor de fundo (“Back Color”) dos Labels, tipo de fonte, tipo de borda, entre as demais configurações. Na aba *General* ao lado, você terá as opções para definir o estilo de formatação dos campos de valor. Na aba *Format* você encontrará inúmeras opções das configurações da aparência do gráfico individuais para cada tipo.

No gráfico do tipo *Pie* marque a opção *Circled Pie* que irá garantir um gráfico sempre em forma de círculo, independente da resolução em que o formulário que carrega o gráfico for aberto.

Voltando agora para a aba *Chart*, que corresponde às configurações gerais do gráfico independente do tipo selecionado, vá até a aba *Titles* e digite o texto

“Gráfico de Clientes por País” no campo *Memo* de edição do título. Altere também sua fonte e tamanho para destacar o título no gráfico.

Na aba *Panel* ative a opção *Visible no grupo Gradient do canto direito inferior*, a qual irá possibilitar que você configure uma combinação de duas cores para o fundo do gráfico.

Após selecionadas as cores, escolha a direção em que as mesmas serão exibidas (Figura 5). Todas as configurações já estarão valendo para o gráfico, utilize o botão *Close* para sair.

Você poderá ir testando todas as demais propriedades de configurações que não mencionei aqui neste artigo.

Ainda no mesmo formulário precisamos adicionar dentro do componente *Panel* um *RadioGroup* da paleta *Standard* para alternar de um tipo de gráfico para outro em tempo de execução.

Adicione a sua propriedade *Items* em duas linhas os valores “Pie” e “Bar”. Adicione também dentro do *Panel* dois componentes *Button* alterando o *Caption* do primeiro para “JPG Export” e do segundo para “Print”. Esses botões serão usados para exportar o gráfico para uma imagem no formato JPG e fazer a impressão direta do gráfico, respectivamente.

Adicione ao evento *OnClick* do *RadioGroup1* o código da **Listagem 1** que será o responsável por alterar os tipos de gráficos em tempo de execução. O código desabilitará todos os tipos de gráficos para que não ocorra a sobreposição de algum deles e após isso ativa um novo tipo de acordo com o índice selecionado.

No botão de exportação do gráfico para o formato JPG inclua a seguinte linha de código, onde chamamos o método *SaveToBitmapFile* do componente *DbChart*, passando um caminho onde a imagem será arquivada. Se preferir, poderá salvar também em outros formatos de imagens, como *Bitmap* por exemplo.

```
DBChart1.SaveToBitmapFile('C:\Grafico.jpg');
```

Ao componente de impressão, simplesmente adicione ao evento *OnClick* o código abaixo, que fará a impressão do gráfico na impressora padrão instalada.

```
DBChart1.Print;
```

---

**Nota:** Tanto na opção de exportar quanto para imprimir, leve em consideração a resolução de tela e do formulário onde o gráfico se encontra. As proporções de exportação e/ou impressão serão exatamente as mesmas as quais o gráfico se encontra no momento da requisição.

---

#### Listagem 1. Código para alteração do tipo de gráfico em runtime

```
procedure TfrmCustomerCountry.RadioGroup1Click(Sender: TObject);
var
  I : Integer;
begin
  for I:= 0 to DBChart1.SeriesCount -1 do
    DBChart1.Series[I].Active := False;
    DBChart1.Series[RadioGroup1.ItemIndex].Active := True
  end;
```

#### Listagem 2. Código exibição dos registros na montagem do gráfico dinâmico.

```
procedure TfrmGraficoDinamico.BitBtn1Click(Sender: TObject);
var
  soma: Integer;
begin
  if ((edtQntde.Text <> '') and (StrToInt(edtQntde.Text) > 0)) Then
  begin
    DM.cdsDinamicoGrafico.Open;
    with DM.cdsDinamicoGrafico do
      begin
        while not eof do Delete;
        DM.sqlDinamicoGrafico.Close;
        DM.sqlDinamicoGrafico.CommandText := 'select first ' + edtQntde.Text +
        ' skip 0 department.department ' + 'as departamento, ' +
        'count(employee.emp_no) as clientes ' + ' from department, employee ' +
        'where employee.dept_no = ' + ' department.dept_no group by ' +
        'department.department';
        DM.sqlDinamicoGrafico.Open;
        soma := 0;
        while not DM.sqlDinamicoGrafico.Eof do
          begin
            DM.cdsDinamicoGrafico.Append;
            DM.cdsDinamicoGraficoDEPARTAMENTO.Value := DM.sqlDinamicoGrafico.Fields[0].AsString;
            DM.cdsDinamicoGraficoCLIENTES.Value := DM.sqlDinamicoGrafico.Fields[1].Value;
            soma := soma + DM.sqlDinamicoGrafico.Fields[1].AsInteger;
            DM.cdsDinamicoGrafico.Post;
            DM.sqlDinamicoGrafico.Next;
          end;
        DM.sqlDinamicoGrafico.Close;
        DM.sqlDinamicoGrafico.CommandText := 'select cast(''OUTROS'' as varchar(70)) ' +
        ' as departamentos, ' + ' cast( (select (count(a.emp_no)) ' + 'from employee a) - ' +
        'IntToStr(Soma)+') as integer) as '+ 'CLIENTES from rdb$database';
        DM.sqlDinamicoGrafico.Open;
        DM.cdsDinamicoGrafico.Append;
        DM.cdsDinamicoGraficoDEPARTAMENTO.Value := DM.sqlDinamicoGrafico.Fields[0].AsString;
        DM.cdsDinamicoGraficoCLIENTES.Value := DM.sqlDinamicoGrafico.Fields[1].Value;
        DM.cdsDinamicoGrafico.Post;
      end;
    end;
  else
    ShowMessage('O valor deverá ser superior a 0.');
```

## Rodando a aplicação

Retornando agora até o formulário principal, adicione um *Button* com o *Caption* “Clientes por País” e adicione ao evento *OnClick* o código para chamar o formulário do gráfico conforme a seguir:

```
frmCustomerCountry.Show;
```

Execute agora sua aplicação, exibindo o formulário onde criamos o gráfico. Veja as formatações e valores do gráfico construído (Figura 6).

## Gráfico dinâmico

Chamo esta forma de gráfico como dinâmica devido ao fato de você ter a liberdade de definir a quantidade de registros que deseja exibir, sendo que o restante dos mesmos, que não são atingidos pela condição de limite da quantidade, serão somados e adicionados ao gráfico em uma categoria "OUTROS". Sua construção não é muito complicada, portanto mãos a obra.

Vá até o Data Module do projeto e adicione três novos componentes, sendo eles:

**SQLDataSet:** altere a propriedade *Name* para "sqlDinamicoGrafico", ligue sua propriedade *SQLConnection* ao componente "sqlConexao" e adicione a seguinte instrução SQL a propriedade *CommandText*:

```
select first 5 skip 0 department.department as departamento,
count(employee.emp_no) as clientes from
department, employee
where employee.dept_no = department.dept_no
group by department.department order by 2
desc
```

**DataSetProvider:** altere o *Name* para "dspDinamicoGrafico" e conecte a propriedade *DataSet* ao *sqlDinamicoGrafico*;

**ClientDataSet:** altere o *Name* para "cdsDinamicoGrafico" e ligue a propriedade *ProviderName* ao *dspDinamicoGrafico*. Abra o *Fields Editor* e adicione todos os campos retornados pela SQL.

Crie agora um novo formulário, altere o nome para "frmGraficoDinamico" e o *Caption* para "Gráfico de empregados por Departamento". Salve a *Unit* do formulário como "uGraficoDinamico.pas". Adicione

agora ao formulário um componente *Panel* com sua propriedade *Align* definida para *alTop* e dentro do *Panel* adicione um componente *Edit* ("edtQntde") para receber a quantidade de registros a ser exibida e um *Button* com o *Caption* "Exibir Gráfico" para ativar a pesquisa. Adicione ao formulário um componente *DBChart* definindo sua propriedade *Align* para *alClient*, deixando os componentes organizados de forma semelhante a **Figura 7**.

Com um duplo clique sobre o gráfico, adicione uma nova série do tipo *Pie*, acesse a aba *Series* ao lado de *Chart* e faça a configuração na aba *DataSource*, de forma semelhante ao exemplo de gráfico anterior, porém agora não ligando o *DataSet* ao *SQLDataSet*, mas sim ao componente *ClientDataSet* configurado no Data Module. Na instrução SQL que adotamos para este exemplo, estamos selecionando o número de empregados por departamento. Atribua à propriedade *Labels* o campo DEPARTAMENTOS e à propriedade *Pie* o campo CLIENTES. Faça as demais configurações do gráfico da mesma forma que no gráfico do exemplo anterior.

Atribua ao evento *OnClick* do botão "Exibir Gráfico" o código da **Listagem 2**, que será responsável por fazer a pesquisa dos registros no banco de dados e armazenar os valores no *ClientDataSet*.

Ao final da seleção de todos os registros, de acordo com o limite de informações a serem exibidas, um cálculo será realizado e designará uma nova busca no banco de dados, que trará como resultados os

registros não atingidos pela primeira condição de pesquisa, adicionando um novo registro ao *ClienteDataSet* com a descrição de OUTROS e o valor correspondente.

Volte agora para o formulário principal e adicione mais um botão, altere o *Caption* para "Empregados por Departamento" e adicione a seguinte linha de código ao evento *OnClick* do botão:

```
frmGraficoDinamico.Show;
```

Execute sua aplicação acessando o gráfico que irá trazer por padrão os 5 primeiros registros que deixamos configurados na instrução SQL, mas que ao receber um parâmetro de limite de valores exibirá o gráfico com registros selecionados e um último registro com o rótulo "OUTROS" e a quantidade a ele relacionado (**Figura 8**).

## Conclusão

Vimos nesse artigo exemplos práticos do uso e da empregabilidade de gráficos em nossos sistemas. Agora de uma forma fácil você poderá apresentar uma ferramenta a mais nos seus sistemas para seus clientes, possibilitando uma visualização mais entendível dos conteúdos e planilhas de informações. Até o próximo artigo. Abraços. ●

**ClubeDelphi PLUS** [www.devmedia.com.br/clubedelphi/portal.asp](http://www.devmedia.com.br/clubedelphi/portal.asp)

Acesse agora o mesmo o portal do assinante ClubeDelphi e assista a uma vídeo aula de Fabrício Desbessel que mostra como trabalhar com gráficos no Delphi.

[www.devmedia.com.br/articles/viewcomp.asp?comp=4906](http://www.devmedia.com.br/articles/viewcomp.asp?comp=4906)

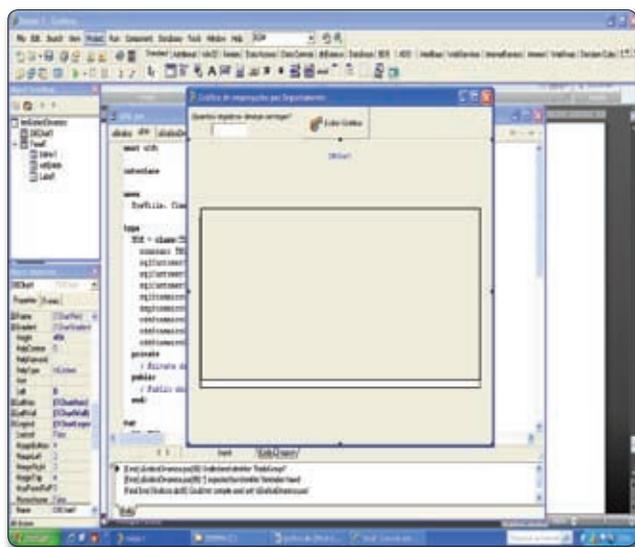


Figura 7. Componentes para montagem do gráfico dinâmico

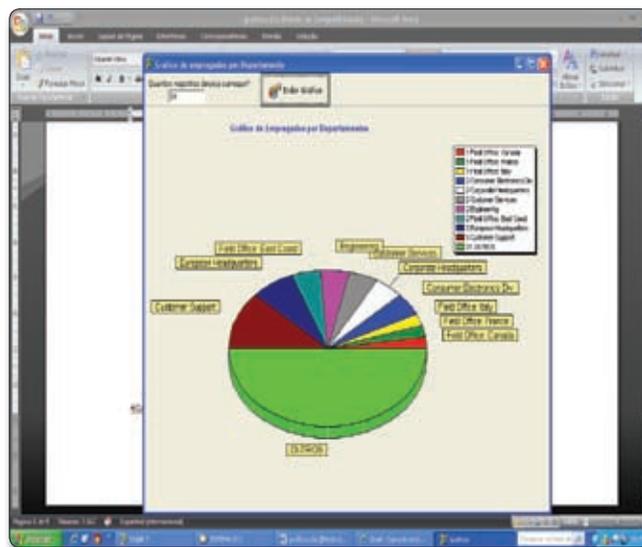


Figura 8. Gráfico dinâmico com limitação de registros