

CONTROLE DE CHEQUES

Cadastro de Endereços

Cadastro de Endereços

Cidade	UF	CEP Cidade
BLUMENAU	SC	89100-000
Rua		CEP Rua
RUA AMAZONAS		89021-000
Bairro		
GARCIA		

Gravar Excluir Cancelar Fechar

```
unit UEnde;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls, Buttons, Grids, DBGrids, CurrEdit, Mask, ToolEdit, ExtCtrls,  
Db, DBTables;
```

```
type
```

```
TTelEnde = class(TForm)  
    pnDados: TPanel;  
    Label9: TLabel;  
    Label10: TLabel;  
    Label11: TLabel;  
    Label7: TLabel;  
    meNrCep: TMaskEdit;  
    PnBotao: TPanel;  
    BtnGrava: TBitBtn;  
    btnCance: TBitBtn;  
    btnSaida: TBitBtn;  
    ceCidad: TComboEdit;  
    EDUFCID: TEdit;  
    LBairr: TLabel;  
    edBairr: TEdit;  
    edEnde: TEdit;  
    Label1: TLabel;  
    edCepCi: TMaskEdit;  
    Table1: TTable;  
    dbgPesqu: TDBGrid;  
    tbPesqu: TTable;  
    dsPesqu: TDataSource;  
    BtnExclu: TBitBtn;
```

```

spPesRua: TSpeedButton;
procedure BtnGravaClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btnCanceClick(Sender: TObject);
procedure btnSaidaClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure ceCidadExit(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure ceCidadButtonClick(Sender: TObject);
procedure ceCidadEnter(Sender: TObject);
procedure ceCidadKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceCidadKeyPress(Sender: TObject; var Key: Char);
procedure ceCidadKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure dbgPesquDbClick(Sender: TObject);
procedure dbgPesquKeyPress(Sender: TObject; var Key: Char);
procedure spPesRuaClick(Sender: TObject);
procedure BtnExcluClick(Sender: TObject);
private
  { Private declarations }
  vEndNumer,
  vCidNumer : Longint;
  procedure pExcluirRua;
  procedure MostrarDbgPesqu(Sender : TObject);
public
  { Public declarations }
end;

var
  TelEnder: TTelEnder;

implementation
Uses
  UDataM,UProce, UPesquisa;
{$R *.DFM}

procedure TTelEnder.pExcluirRua;
begin
  dm.QDelete.Close;
  dm.QDelete.SQL.Clear;
  dm.QDelete.SQL.Add('Delete from ' + 'Le' + FormatFloat('00000',vCidNumer));
  dm.QDelete.SQL.Add('Where LogNumer = :p1');
  dm.QDelete.ParamByName('p1').AsInteger := vEndNumer;
  dm.QDelete.ExecSQL;

  edEnder.Clear;
  meNrCep.Clear;
  edBairr.Clear;
end;

```

```

procedure TTelEnder.MostrarDbgPesqu(Sender : TObject);
begin

    dbgPesqu.Top := (Sender as tComboEdit).Top + (Sender as tComboEdit).Height;
    dbgPesqu.Left := (Sender as tComboEdit).Left;
    dbgPesqu.Width := (Sender as tComboEdit).Width;
    dbgPesqu.DataSource := dsPesqu;
    dbgPesqu.Visible := true;

end;

procedure TTelEnder.BtnGravaClick(Sender: TObject);
var
    vCodigo : Longint;
    vEstado,
    vMumCep,
    vCidade : String;

begin
    if ceCidad.Text = '' then
        begin
            MessageDlg('Informe a cidade!',mtInformation,[mbOk],0);
            ceCidad.SetFocus;
            exit;
        end;

    if edUfCid.Text = '' then
        begin
            MessageDlg('Informe o estado!',mtInformation,[mbOk],0);
            edUfCid.SetFocus;
            exit;
        end;

    if edEnder.Text = '' then
        begin
            MessageDlg('Informe o nome da rua!',mtInformation,[mbOk],0);
            edEnder.SetFocus;
            exit;
        end;

    if meNrCep.Text = ' ' - ' ' then
        begin
            MessageDlg('Informe o CEP da rua!',mtInformation,[mbOk],0);
            meNrCep.SetFocus;
            exit;
        end;

    if edBairr.Text = '' then
        begin
            MessageDlg('Informe o Bairro!',mtInformation,[mbOk],0);
            edBairr.SetFocus;
            exit;
        end;

```

```

end;

if vCidNumer = 0 then
begin
  With Dm.QInserir do
  begin
    vCidNumer := BuscaNumero('LeMunic','MunNumer');
    Close;
    Sql.clear;
    Sql.Add('Insert Into LeMunic ');
    Sql.Add('(MunNumer,MunEstad,MunCidad,MunNrCep,BluDatas) ');
    Sql.Add('Values (:vCodig,:vEstad,:vCidad,:vNrCep,:vDatas)');
    ParamByName('vCodig').AsInteger := vCidNumer;
    ParamByName('vEstad').AsString := EDUFCID.Text;
    ParamByName('vCidad').AsString := ceCidad.Text;
    ParamByName('vNrCep').AsString := edCepCi.Text;
    ParamByName('vDatas').AsDateTime := date;
    ExecSql;

    end;
  end;

try
  Table1.Close;
  Table1.TableName := 'Le' + FormatFloat('00000',vCidNumer);
  Table1.Open;
except
  With Table1 do
  begin
    Table1.Active := false;
    Table1.DatabaseName := 'GmvChequ';
    Table1.TableName := 'Le' + FormatFloat('00000',vCidNumer);
    Table1.TableType := ttParadox;
    with FieldDefs do
    begin
      Clear;
      Add('logNumer',ftInteger,0,false);
      Add('logENDER',ftString,50,false);
      Add('logBairr',ftString,30,false);
      Add('logNrCep',ftString,08,false);
    end;
    try
      CreateTable;
      AddIndex('', 'LogNumer',[ixPrimary]);
      AddIndex('Rua','LogENDER',[ixCaseInsensitive]);
    except
      on E:Exception do
      begin
        MessageDlg(E.Message + ' Erro ao criar tabela de
logradouros.',mtInformation,[mbOk],0);
      end;
    end;
  end;

```

```

        end;
    end;
end;

With Dm.QInserir do
begin
    vEndNumer := BuscaNumero('Le' + FormatFloat('00000',vCidNumer),'LogNumer');
    Close;
    Sql.clear;
    Sql.Add('Insert Into ' + 'Le' + FormatFloat('00000',vCidNumer));
    Sql.Add('(LogNumer,logEnder,logBairr,logNrCep)');
    Sql.Add('Values (:vNumer,:vEnder,:vBairr,:vNrCep)');
    ParamByName('vNumer').AsInteger := vEndNumer;
    ParamByName('vEnder').AsString := edEnder.Text;
    Dm.QInserir.ParamByName('vNrCep').AsString := meNrCep.Text;
    ParamByName('vBairr').AsString := edBairr.Text;
    ExecSql;

    vEndereco := edEnder.Text;
    vCidade := ceCidad.Text;
    vCodigo := vCidNumer;
    vEstado := EDUFCID.Text;
    vMumCep := edCepCi.Text;
    LimparTela(telEnder);
    ceCidad.Text := vCidade;
    vCidNumer := vCodigo;
    EDUFCID.Text := vEstado;
    edCepCi.Text := vMumCep;
    edEnder.SetFocus;
end;

close;
end;

procedure TTelEnder.FormCreate(Sender: TObject);
begin
    vCidNumer := 0;

end;

procedure TTelEnder.btnCanceClick(Sender: TObject);
begin
    LimparTela(TelEnder);
    ceCidad.SetFocus;
    vCidNumer := 0;
end;

procedure TTelEnder.btnSaidaClick(Sender: TObject);
begin
    Close;
end;

```

```

procedure TTelEnder.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Action := caFree;
    TelEnder := nil;
end;

```

```

procedure TTelEnder.FormKeyPress(Sender: TObject; var Key: Char);
begin
    TestaTecla(Self,key);
end;

```

```

procedure TTelEnder.ceCidadExit(Sender: TObject);
begin

    if ( (ActiveControl = btnCance) or (ActiveControl = btnSaida) or
        (ActiveControl = dbgPesqu) ) then
        exit;

    dm.QPesquisa.Close;
    dm.QPesquisa.SQL.Clear;
    dm.QPesquisa.Sql.Add('Select * from leMunic ');
    dm.QPesquisa.sql.add('where MunCidad like ' + chr(39) + ceCidad.Text + '%' + chr(39));
    dm.QPesquisa.Open;
    if dm.QPesquisa.RecordCount > 0 then
        begin
            vCidNumer := dm.QPesquisa.FieldByName('MunNumer').AsInteger;
            ceCidad.Text := dm.QPesquisa.FieldByName('MunCidad').AsString;
            edUfCid.Text := dm.QPesquisa.FieldByName('MunEstad').AsString;
            edCepCi.EditMask := '';
            edCepCi.Text:= dm.QPesquisa.FieldByName('MunNrCep').AsString;
            edCepCi.EditMask := '00000-000;0;_';
            vEndNumer := 0;
            edEnder.Text := '';
            edBairr.Text := '';
        end
    else
        begin
            if messageDlg('Cidade não está cadastrada. Deseja cadastrar nova cidade?',
                mtCustom,[mbNo,mbYes],0) = mrNo then
                begin
                    ceCidadButtonClick(Sender);
                end;
        end;
    end;

end;

```

```

procedure TTelEnder.FormShow(Sender: TObject);
begin

```

```

TrazFiguras(TelEnder);
ceCidad.SetFocus;

end;

procedure TTelEnder.ceCidadButtonClick(Sender: TObject);
begin
    telEnder.KeyPreview := false;
    if not dbgPesqu.Visible then
        begin
            MostrarDbgPesqu(Sender);
            dbgPesqu.Height := 197;
        end;

    ceCidad.SetFocus;

end;

procedure TTelEnder.ceCidadEnter(Sender: TObject);
begin
    tbPesqu.close;
    tbPesqu.TableName := 'LeMunic';
    tbPesqu.IndexName := 'Cidad';
    tbPesqu.Open;
    tbPesqu.IndexName := 'Cidad';
    dsPesqu.DataSet := tbPesqu;
    dbgPesqu.Columns.Clear;
    dbgPesqu.Columns.Add.FieldName := 'MunNumer';
    dbgPesqu.Columns[0].Title.Caption := 'Código';

    dbgPesqu.Columns.Add.FieldName := 'MunCidad';
    dbgPesqu.Columns[1].Title.Caption := 'Município';

end;

procedure TTelEnder.ceCidadKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if not dbgPesqu.Visible then
        begin
            if Key = vk_f2 then
                begin
                    MostrarDbgPesqu(Sender);
                    dbgPesqu.Height := 197;
                    dbgPesqu.Visible := true;
                end;
            end
        else
            if Key = vk_f3 then
                dbgPesqu.Visible := false;
            end;
end;

```

```

procedure TTelEnder.ceCidadKeyPress(Sender: TObject; var Key: Char);
begin
  if key = #0 then
  begin
    if dbgPesqu.Visible then
      dbgPesqu.SetFocus;
    end;
  end;

```

```

end;

```

```

procedure TTelEnder.ceCidadKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if dbgPesqu.Visible then
  if ceCidad.Text <> '' then
    tbPesqu.Locate('MunCidad',ceCidad.Text,[loPartialKey]);

```

```

end;

```

```

procedure TTelEnder.dbgPesquDbClick(Sender: TObject);
begin
  vCidNumer := tbPesqu.FieldByName('MunNumer').AsInteger;
  ceCidad.Text := tbPesqu.FieldByName('MunCidad').AsString;
  edUfCid.Text := tbPesqu.FieldByName('MunEstad').AsString;
  edCepCi.EditMask := '';
  edCepCi.Text:= tbPesqu.FieldByName('MunNrCep').AsString;
  edCepCi.EditMask := '00000-000;0;_';
  edEnder.Text := '';
  edBairr.Text := '';
  dbgPesqu.Visible := false;
  edEnder.SetFocus;
  TelEnder.KeyPreview := true;

```

```

end;

```

```

procedure TTelEnder.dbgPesquKeyPress(Sender: TObject; var Key: Char);
begin
  dbgPesquDbClick(Sender);
end;

```

```

procedure TTelEnder.spPesRuaClick(Sender: TObject);
begin
  if vCidNumer <> 0 then
  begin
    if FPesquisa = nil then
      fPesquisa := TfPesquisa.Create(Self);

    fPesquisa.vTabela := 'Le' + FormatFloat('00000',vCidNumer);

```



```

fPesquisa.vCampo := 'LogEnder';

fPesquisa.ShowModal;

if fPesquisa.ModalResult = mrOk then
begin
    vEndNumer := dm.QPesquisa.fieldByName('LogNumer').AsInteger;
    edEnder.text := dm.QPesquisa.FieldByName('LogEnder').AsString;
    meNrCep.Text := dm.QPesquisa.FieldByName('LogNrCep').AsString;
    edBairr.Text := dm.QPesquisa.FieldByName('LogBairr').AsString;

end;

fPesquisa.Free;
fPesquisa := Nil;
end;

end;

procedure TTelEnder.BtnExcluClick(Sender: TObject);
begin
    if MessageDlg('Confirma a exclusao da rua? ', mtConfirmation, [mbYes,mbNO],0) = mrYes
    then
        begin
            dm.qPesquisa.Close;
            dm.qPesquisa.SQL.Clear;
            dm.qPesquisa.SQL.Add('Select CliEnder from grClien');
            dm.qPesquisa.SQL.Add('Where cliEnder = :p1');
            dm.qPesquisa.ParamByName('p1').AsInteger := vEndNumer;
            dm.QPesquisa.Open;
            if dm.QPesquisa.RecordCount > 0 then
                begin
                    if MessageDlg('Atenção! Esta rua está relacionada com cadastros de clientes. Confirma a
                    exclusão? ', mtConfirmation, [mbYes,mbNO],0) = mrYes then
                        begin
                            pExcluirRua;
                            dm.QDelete.Close;
                            dm.QDelete.SQL.Clear;
                            dm.QDelete.SQL.Add('UpDate grClien set cliEnder = 0 ');
                            dm.QDelete.SQL.Add('Where cliEnder = :p1');
                            dm.QDelete.ParamByName('p1').AsInteger := vEndNumer;
                            dm.QDelete.ExecSQL;

                            dm.QDelete.Close;
                            dm.QDelete.SQL.Clear;
                            dm.QDelete.SQL.Add('UpDate grForne set ForEnder = 0 ');
                            dm.QDelete.SQL.Add('Where ForEnder = :p1');
                            dm.QDelete.ParamByName('p1').AsInteger := vEndNumer;
                            dm.QDelete.ExecSQL;

                            dm.QDelete.Close;

```

```

dm.QDelete.SQL.Clear;
dm.QDelete.SQL.Add('UpDate grTerce set terEnde = 0 ');
dm.QDelete.SQL.Add('Where terEnde = :p1');
dm.QDelete.ParamByName('p1').AsInteger := vEndNumer;
dm.QDelete.ExecSQL;
end;
end
else
pExcluirRua;

end;
end;

end.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Cadastro de Fornecedores

```

Cadastro de Fornecedores

Fornecedor: J&J - ATACADISTA DE PRODUTOS ALIMENTÍCIOS S/A

Pessoa: **Jurídica** CPF/CGC: 22.522.522/0001-02 Inscrição Estadual: 545125222 Identidade:

Responsável: JOÃO DA SILVA ANTUNES

Cidade: BLUMENAU Estado: SC CEP: 89021-001

Rua: RUA AMAZONAS Número: 2562

Bairro: GARCIA

Telefone: 33255698 Fax: 33258898 E-mail: j_latacado@gmail.com

Data Cadastramento: 16/04/2013

Buttons: Gravar, Cancelar, Excluir, Fechar

unit UForne;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, ToolEdit, Mask, Buttons, ExtCtrls, DbGrids, DB, Grids, DBCtrls,
DBTables;

type

TTelForne = class(TForm)
PnBotao: TPanel;

```

BtnGrava: TBitBtn;
BtnCance: TBitBtn;
BtnExclu: TBitBtn;
BtnSaida: TBitBtn;
PnDados: TPanel;
CbTpPes: TComboBox;
Label1: TLabel;
CeNomes: TComboEdit;
Label2: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
EdBairr: TEdit;
Label9: TLabel;
Label10: TLabel;
Label12: TLabel;
Label13: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
EdEmail: TEdit;
Label18: TLabel;
Label20: TLabel;
DeDtCad: TDateEdit;
EdRespo: TEdit;
EdIdent: TEdit;
EdNrRua: TEdit;
EdInscr: TEdit;
Label3: TLabel;
meCpfCg: TMaskEdit;
meNrCep: TMaskEdit;
edNrTel: TEdit;
edNrFax: TEdit;
edUfMun: TEdit;
ceMunic: TComboEdit;
tbPesqu: TTable;
dsPesqu: TDataSource;
ceEnder: TComboEdit;
dbgPesqu: TDBGrid;
Table1: TTable;
procedure BtnGravaClick(Sender: TObject);
procedure BtnSaidaClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure CeNomesButtonClick(Sender: TObject);
procedure BtnExcluClick(Sender: TObject);
procedure BtnCanceClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure EdNrRuaKeyPress(Sender: TObject; var Key: Char);
procedure CbTpPesExit(Sender: TObject);
procedure CbEstadKeyPress(Sender: TObject; var Key: Char);

```

```

procedure CeNomesKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure CeNomesEnter(Sender: TObject);
procedure CeNomesKeyPress(Sender: TObject; var Key: Char);
procedure CeNomesKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure dbgPesquDbClick(Sender: TObject);
procedure dbgPesquKeyPress(Sender: TObject; var Key: Char);
procedure ceMunicButtonClick(Sender: TObject);
procedure ceMunicEnter(Sender: TObject);
procedure ceMunicExit(Sender: TObject);
procedure ceMunicKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceMunicKeyPress(Sender: TObject; var Key: Char);
procedure ceMunicKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceEnderButtonClick(Sender: TObject);
procedure ceEnderEnter(Sender: TObject);
procedure ceEnderExit(Sender: TObject);
procedure ceEnderKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceEnderKeyPress(Sender: TObject; var Key: Char);
procedure ceEnderKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
private
  { Private declarations }
  DbGForne : TDbGrid;
  Pesquisar : boolean;
  vMunNumer,
  vEndNumer,
  vForNumer : Longint;
  procedure CarregaDadosEndereco;
  procedure CarregaDadosCidade;
  procedure ConfereCidade(Sender : TObject);
  procedure CarregaDadosArquivoParaTela;
  procedure MostrarDbgPesqu(Sender : TObject);
  function IncluirFornecedor : Boolean;
  function AlterarFornecedor : Boolean;
  procedure LimparVariaveis;
  procedure HabilitaCampos;
public
  { Public declarations }
end;

var
  TelForne: TTelForne;

implementation

uses UDataM,UProce;

```

```
{ $R *.DFM }
```

```
procedure TTelForne.CarregaDadosEndereco;
begin
  try
    with dm.tbTempo do
      begin
        Close;
        TableName := 'Le' + FormatFloat('00000',vMunNumer);
        Open;
        IndexName := '';

        if FindKey([vEndNumer]) then
          begin
            ceEnder.text := FieldByName('LogEnder').AsString;
            meNrCep.Text := FieldByName('LogNrCep').AsString;
            edBairr.Text := FieldByName('LogBairr').AsString;
          end
        else
          begin
            ceEnder.text := '';
            meNrCep.Text := '';
            edBairr.Text := '';
          end;

        end;
      except

        end;
    end;
```

```
procedure TTelForne.CarregaDadosCidade;
begin
  with dm.TbMunic do
    begin
      IndexName := '';
      if FindKey([vMunNumer]) then
        begin
          ceMunic.text := FieldByName('MunCidad').AsString;
          edUfMun.Text := FieldByName('MunEstad').AsString;
          meNrCep.Text := FieldByName('MunNrCep').AsString;
        end
      else
        begin
          ceMunic.text := '';
          edUfMun.Text := '';
          meNrCep.Text := '';
        end;
    end;
```

```

    end;
end;

procedure TTelForne.ConfereCidade(Sender : TObject);
begin
    tbPesqu.close;
    tbPesqu.TableName := 'LeMunic';
    tbPesqu.IndexName := 'Cidad';
    tbPesqu.Open;
    with tbPesqu do
    begin
        if FindKey([ceMunic.Text]) then
        begin
            vMunNumer := FieldByName('MunNumer').AsInteger;
            ceMunic.Text := FieldByName('MunCidad').AsString;
            edUfMun.Text := FieldByName('MunEstad').AsString;
            meNrCep.EditMask := '';
            meNrCep.Text:= FieldByName('MunNrCep').AsString;
            meNrCep.EditMask := '00000-000;0;_';
            vEndNumer := 0;
            ceEnder.Text := '';
            edBairr.Text := '';

            end
        else
        begin
            vMunNumer := 0;
            edUfMun.Text := FieldByName('MunEstad').AsString;
            meNrCep.EditMask := '';
            meNrCep.Text:= '';
            meNrCep.EditMask := '00000-000;0;_';
            vEndNumer := 0;
            ceEnder.Text := '';
            edBairr.Text := '';

            end;
        end;
    end;

procedure TTelForne.CarregaDadosArquivoParaTela;
begin
    with tbPesqu do
    begin
        CbTpPes.ItemIndex := FieldByName('ForTpPes').AsInteger - 1;
        meCpfCg.Text := FieldByName('ForNrDoc').AsString;
        EdInscr.Text := FieldByName('ForInsEs').AsString;
        EdIdent.Text := FieldByName('ForIdent').AsString;
        EdRespo.Text := FieldByName('ForRespo').AsString;
        EdNrRua.Text := FieldByName('ForNrRua').AsString;
        edNrTel.Text := FieldByName('ForNrTel').AsString;
        edNrFax.Text := FieldByName('ForNrFax').AsString;
    end;
end;

```

```

EdEmail.Text    := FieldByName('ForEmail').AsString;
deDtCad.Date    := FieldByName('ForDtCad').AsDateTime;
vMunNumer      := FieldByName('ForMunic').AsInteger;
if vMunNumer > 0 then
    CarregaDadosCidade;
vEndNumer      := FieldByName('ForEnder').AsInteger;
if vEndNumer > 0 then
    CarregaDadosEndereco;

HabilitaCampos;

end;
end;

procedure TTelForne.MostrarDbgPesqu(Sender : TObject);
begin
    TelForne.KeyPreview := false;
    dbgPesqu.Top := (Sender as tComboEdit).Top + (Sender as tComboEdit).Height;
    dbgPesqu.Left := (Sender as tComboEdit).Left;
    dbgPesqu.Width := (Sender as tComboEdit).Width;
    dbgPesqu.DataSource := dsPesqu;
    dbgPesqu.Visible := true;
end;

function TTelForne.IncluirFornecedor : Boolean;
var
    vNumero : Longint;
begin
    try
        with Dm.QInserir do
            begin
                vNumero := BuscaNumero('GrForne','ForNumer');
                Close;
                Sql.Clear;
                Sql.Add('Insert Into GrForne ');
                Sql.Add('(ForNumer,ForNomes,ForTpPes,ForNrDoc,ForInsEs, ');
                Sql.Add('ForIdent,ForRespo,ForMunic,ForEnder,ForNrRua, ');
                Sql.Add('ForNrTel,ForNrFax,ForEmail,ForDtCad)');
                Sql.Add(' Values (:vNumer,:vNomes,:vTpPes,:vNrDoc,:vInsEs,');
                Sql.Add('      :vIdent,:vRespo,:vMunic,:vEnder,:vNrRua,');
                Sql.Add('      :vNrTel,:vNrFax,:vEMail,:vDtCad)');
                ParamByName('vNumer').AsInteger := vNumero;
                ParamByName('vNomes').AsString := ceNomes.Text;
                ParamByName('vTpPes').AsInteger := cbTpPes.ItemIndex + 1;
                ParamByName('vNrDoc').AsString := meCpfCg.Text;
                ParamByName('vIdent').AsString := edIdent.Text;
                ParamByName('vInsEs').AsString := EdInscr.Text;
                ParamByName('vRespo').AsString := EdRespo.Text;
                ParamByName('vMunic').AsInteger := vMunNumer;
                ParamByName('vEnder').AsInteger := vEndNumer;
                ParamByName('vNrRua').AsString := edNrRua.Text;
                ParamByName('vNrTel').AsString := EdNrTel.Text;
            end
        end
    except
        ;
    end
end;

```

```

        ParamByName('vNrFax').AsString := EdNrFax.Text;
        ParamByName('vEmail').AsString := EdEmail.Text;
        ParamByName('vDtCad').AsDateTime := deDtCad.Date;
        ExecSql;
        Result := true;
    end;
except
    On E:Exception do
        begin
            vMensagem := E.Message;
            Result := false;
        end;
    end;
end;

function TTelForne.AlterarFornecedor : Boolean;
begin
    try
        with Dm.QInserir do
            begin
                Close;
                Sql.Clear;
                Sql.Add('UpDate GrForne ');
                Sql.Add('Set ForNomes = :vNomes,ForTpPes = :vTpPes,ForNrDoc = :vNrDoc, ');
                Sql.Add(' ForInsEs = :vInsEs,ForIdent = :vIdent,ForRespo = :vRespo, ');
                Sql.Add(' ForMunic = :vMunic,ForEnder = :vEnder,ForNrRua = :vNrRua, ');
                Sql.Add(' ForNrTel = :vNrTel,ForNrFax = :vNrFax,ForEmail = :vEmail, ');
                Sql.Add(' ForDtCad = :vDtCad ');
                Sql.Add(' Where ForNumer = :vNumer ');
                ParamByName('vNumer').AsInteger := vForNumer;
                ParamByName('vNomes').AsString := ceNomes.Text;
                ParamByName('vTpPes').AsInteger := cbTpPes.ItemIndex + 1;
                ParamByName('vNrDoc').AsString := meCpfCg.Text;
                ParamByName('vIdent').AsString := edIdent.Text;
                ParamByName('vInsEs').AsString := EdInscr.Text;
                ParamByName('vRespo').AsString := EdRespo.Text;
                ParamByName('vMunic').AsInteger := vMunNumer;
                ParamByName('vEnder').AsInteger := vEndNumer;
                ParamByName('vNrRua').AsString := edNrRua.Text;
                ParamByName('vNrTel').AsString := EdNrTel.Text;
                ParamByName('vNrFax').AsString := EdNrFax.Text;
                ParamByName('vEmail').AsString := EdEmail.Text;
                ParamByName('vDtCad').AsDateTime := deDtCad.Date;
                ExecSql;
                Result := true;
            end;
        except
            On E:Exception do
                begin
                    vMensagem := E.Message;
                    Result := false;
                end;
            end;
        end;
    end;
end;

```



```

    end;
end;

procedure TTelForne.LimparVariaveis;
begin
    vMunNumer := 0;
    vEndNumer := 0;
    vForNumer := 0;
    DeDtCad.Text := DateToStr(Now);
    if dbgPesqu.Visible then
        dbgPesqu.Visible := false;

    TelForne.KeyPreview := true;
    ceNomes.SetFocus;

end;

procedure TTelForne.BtnGravaClick(Sender: TObject);
begin
    if CeNomes.Text = '' then
        begin
            ShowMessage('Fornecedor não foi informado. ');
            CeNomes.SetFocus;
            exit;
        end;

    if ceMunic.Text <> '' then
        if vMunNumer = 0 then
            begin
                vMunNumer := InserirDadosCidade(ceMunic.Text,
                    edUfMun.Text,
                    meNrCep.Text);
                if vMunNumer = 0 then
                    begin
                        MessageDlg('Erro ao incluir dados da cidade.',
                            mtInformation,[mbOk],0);
                    end;
                end;
            end;

    if ceEnder.Text <> '' then
        if vEndNumer = 0 then
            begin
                vEndNumer := InserirDadosLogradouro(vMunNumer,
                    ceEnder.Text,
                    meNrCep.Text,
                    edBairr.Text);
                if vEndNumer = 0 then
                    begin
                        MessageDlg('Erro ao incluir dados da cidade.',
                            mtInformation,[mbOk],0);
                    end;
                end;
            end;
        end;
    end;
end;

```

```

        end;

        if vForNumer = 0 then // incluir
            IncluirFornecedor
        else
            AlterarFornecedor;

        LimparTela(TelForne);
        LimparVariaveis;

    end;

    procedure TTelForne.BtnSaidaClick(Sender: TObject);
    begin
        Close;
    end;

    procedure TTelForne.FormClose(Sender: TObject; var Action: TCloseAction);
    begin
        TelForne := nil;
        Action := CaFree;
    end;

    procedure TTelForne.CeNomesButtonClick(Sender: TObject);
    begin
        if not dbgPesqu.Visible then
            begin
                MostrarDbgPesqu(Sender);
                dbgPesqu.Height := 197;
            end;
        ceNomes.SetFocus;
    end;

    procedure TTelForne.BtnExcluClick(Sender: TObject);
    begin
        if MessageDlg('Confirma a exclusão do fornecedor?',
            mtInformation,[mbYes,mbNo],0) = mrYes then
            begin
                if DeletarDados('GrForne','ForNumer',vForNumer) then
                    begin
                        LimparTela(TelForne);
                        LimparVariaveis;
                    end
                else
                    ShowMessage('Erro na exclusão do registro');
            end;
        end;

    end;

    procedure TTelForne.BtnCanceClick(Sender: TObject);

```

```

begin
  LimparTela(TelForne);
  LimparVariaveis;
end;

procedure TTelForne.FormShow(Sender: TObject);
begin
  TrazFiguras(TelForne);
  LimparVariaveis;

end;

procedure TTelForne.EdNrRuaKeyPress(Sender: TObject; var Key: Char);
begin
  if not (key in ['0'..'9',#8]) then
    key := #0;
end;

procedure TTelForne.HabilitaCampos;
begin
  if cbTpPes.ItemIndex = 1 then
    begin
      EdIdent.Enabled := false;
      EdIdent.Color := clScrollBar;
      EdInscr.Enabled := true;
      EdInscr.Color := clWindow;
      meCpfCg.EditMask := '00.000.000/0000-00;0;';

    end
  else
    begin
      EdIdent.Enabled := true;
      EdIdent.Color := clWindow;
      EdInscr.Enabled := false;
      EdInscr.Color := clScrollBar;
      meCpfCg.EditMask := '000.000.000-00;0;';
    end;

    meNrCep.EditMask := '99999-999;0;_';
end;

procedure TTelForne.CbTpPesExit(Sender: TObject);
begin
  try
    if cbTpPes.Text[1] in ['0'..'9'] then
      cbTpPes.ItemIndex := StrToInt(cbTpPes.Text) - 1;
    except
    end;

    HabilitaCampos;
end;

```

```

procedure TTelForne.CbEstadKeyPress(Sender: TObject; var Key: Char);
begin
    key := UpCase(Key);
    if not (Key in ['a'..'z', 'A'..'Z', #8]) then
        Key := #0;
end;

```

```

procedure TTelForne.CeNomesKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if not dbgPesqu.Visible then
        begin
            if Key = vk_f2 then
                begin
                    MostrarDbgPesqu(Sender);
                    dbgPesqu.Height := 197;
                    dbgPesqu.Visible := true;
                end;
            end
        else
            if Key = vk_f3 then
                dbgPesqu.Visible := false;
            end;
end;

```

```

procedure TTelForne.FormKeyPress(Sender: TObject; var Key: Char);
begin
    TestaTecla(Self, Key);
end;

```

```

procedure TTelForne.CeNomesEnter(Sender: TObject);
begin
    tbPesqu.close;
    tbPesqu.TableName := 'GrForne';
    tbPesqu.IndexName := 'Nomes';
    tbPesqu.Open;
    dsPesqu.DataSet := tbPesqu;
    dbgPesqu.Parent := PnDados;
    dbgPesqu.Columns.Clear;
    dbgPesqu.Columns.AddFieldName := 'ForNumer';
    dbgPesqu.Columns[0].Title.Caption := 'Código';
    dbgPesqu.Columns.AddFieldName := 'ForNomes';
    dbgPesqu.Columns[1].Title.Caption := 'Nome';

end;

```

```

procedure TTelForne.CeNomesKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #0 then
        begin

```

```

        if dbgPesqu.Visible then
            dbgPesqu.SetFocus;
        end;

end;

procedure TTelForne.CeNomesKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if dbgPesqu.Visible then
        if ceNomes.Text <> '' then
            tbPesqu.Locate('ForNomes',ceNomes.Text,[loPartialKey]);

end;

procedure TTelForne.dbgPesquDbClick(Sender: TObject);
begin
    if tbPesqu.TableName = 'GrForne' then
        begin
            LimparTela(TelForne);
            vForNumer := tbPesqu.FieldByName('ForNumer').AsInteger;
            ceNomes.Text := tbPesqu.FieldByName('ForNomes').AsString;
            CarregaDadosArquivoParaTela;
            cbTpPes.SetFocus;
        end
    else
        if tbPesqu.TableName = 'Le' + FormatFloat('00000',vMunNumer) then
            begin
                vEndNumer := tbPesqu.FieldByName('LogNumer').AsInteger;
                ceEnder.Text := tbPesqu.FieldByName('LogEnder').AsString;
                meNrCep.EditMask := '';
                meNrCep.Text:= tbPesqu.FieldByName('logNrCep').AsString;
                meNrCep.EditMask := '00000-000;0;_';
                edBairr.Text := tbPesqu.FieldByName('LogBairr').AsString;
                edNrRua.SetFocus;
            end
        else
            if tbPesqu.TableName = 'LeMunic' then
                begin
                    vMunNumer := tbPesqu.FieldByName('MunNumer').AsInteger;
                    ceMunic.Text := tbPesqu.FieldByName('MunCidad').AsString;
                    edUfMun.Text := tbPesqu.FieldByName('MunEstad').AsString;
                    meNrCep.EditMask := '';
                    meNrCep.Text:= tbPesqu.FieldByName('MunNrCep').AsString;
                    meNrCep.EditMask := '00000-000;0;_';
                    ceEnder.Text := '';
                    vEndNumer := 0;
                    edBairr.Text := '';
                    ceEnder.SetFocus;
                end;
            end;
        dbgPesqu.Visible := false;

```

```

telForne.KeyPreview := true;

end;

procedure TTelForne.dbgPesquKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #13 then
        dbgPesquDbClick(Sender);
    end;

procedure TTelForne.ceMunicButtonClick(Sender: TObject);
begin
    MostrarDbgPesqu(Sender);
    dbgPesqu.Height := 133;
    ceMunic.SetFocus;

end;

procedure TTelForne.ceMunicEnter(Sender: TObject);
begin
    tbPesqu.close;
    tbPesqu.TableName := 'LeMunic';
    tbPesqu.IndexName := 'Cidad';
    tbPesqu.Open;
    tbPesqu.IndexName := 'Cidad';
    dsPesqu.DataSet := tbPesqu;
    dbgPesqu.Columns.Clear;

    dbgPesqu.Columns.Add.FieldName := 'MunNumer';
    dbgPesqu.Columns[0].Title.Caption := 'Código';

    dbgPesqu.Columns.Add.FieldName := 'MunCidad';
    dbgPesqu.Columns[1].Title.Caption := 'Município';
end;

procedure TTelForne.ceMunicExit(Sender: TObject);
begin
    if ( ActiveControl = btnCance) or (ActiveControl = btnSaida) ) then
        exit;

    if not dbgPesqu.Visible then
        begin
            if ceMunic.Text <> '' then
                begin
                    try
                        ConfereCidade(ceMunic);
                    except
                        end;
                    end
                else
                    begin
                        vMunNumer := 0;

```

```

        vEndNumer := 0;
    end;
end;

dm.QPesquisa.Close;
dm.QPesquisa.SQL.Clear;
dm.QPesquisa.Sql.Add('Select * from leMunic ');
dm.QPesquisa.sql.add('where MunCidad like ' + chr(39) + ceMunic.Text + '%' + chr(39));
dm.QPesquisa.Open;
if dm.QPesquisa.RecordCount > 0 then
begin
    vMunNumer := dm.QPesquisa.FieldByName('MunNumer').AsInteger;
    ceMunic.Text := dm.QPesquisa.FieldByName('MunCidad').AsString;
    edUfMun.Text := dm.QPesquisa.FieldByName('MunEstad').AsString;
    meNrCep.EditMask := '';
    meNrCep.Text:= dm.QPesquisa.FieldByName('MunNrCep').AsString;
    meNrCep.EditMask := '00000-000;0;_';
    vEndNumer := 0;
    ceEnder.Text := '';
    edBairr.Text := '';

end;

end;

procedure TTelForne.ceMunicKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
    if Key = vk_f2 then
        begin
            if Not dbgPesqu.Visible then
                begin
                    MostrarDbgPesqu(Sender);
                    dbgPesqu.Height := 133;
                end;
            end
        else
            if Key = vk_f3 then
                dbgPesqu.Visible := false;

end;

procedure TTelForne.ceMunicKeyPress(Sender: TObject; var Key: Char);
begin
    if dbgPesqu.Visible then
        if key = #0 then
            dbgPesqu.SetFocus;

end;

procedure TTelForne.ceMunicKeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);

```

```

begin
  if ceMunic.Text <> '' then
    tbPesqu.Locate('MunCidad',ceMunic.Text,[loPartialKey]);

end;

procedure TTelForne.ceEnderButtonClick(Sender: TObject);
begin
  MostrarDbgPesqu(Sender);
  dbgPesqu.Height := 96;
  ceEnder.SetFocus;
end;

procedure TTelForne.ceEnderEnter(Sender: TObject);
begin
  try
    tbPesqu.close;
    if vMunNumer <> 0 then
      begin
        tbPesqu.TableName := 'Le' + FormatFloat('00000',vMunNumer);
        tbPesqu.IndexName := 'Rua';
        tbPesqu.Open;
        dbgPesqu.Columns.Clear;
        dsPesqu.DataSet := tbPesqu;
      end;

  except
    On E:Exception do
      begin
        MessageDlg(e.message + '. Tabela de logradouro inexistente.',
          mtInformation,[mbOk],0);
      end;
    end;

end;

procedure TTelForne.ceEnderExit(Sender: TObject);
begin

  if ( ActiveControl = btnSaida) or (ActiveControl = btnCance) ) then
    exit;

  if ceEnder.Text = '' then
    exit;

  if vMunNumer > 0 then
    vEndNumer := VerificaRuaJaCadastrada(vMunNumer,ceEnder.Text);

end;

procedure TTelForne.ceEnderKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);

```



```

begin
  if Key = vk_f2 then
    begin
      if not dbgPesqu.Visible then
        begin
          MostrarDbgPesqu(Sender);
          dbgPesqu.Height := 96;
        end;
      end
    else
      if Key = vk_f3 then
        dbgPesqu.Visible := false;
      end;
    end;

  procedure TTelForne.ceEnderKeyPress(Sender: TObject; var Key: Char);
  begin
    if vMunNumer > 0 then
      if not dbgPesqu.Visible then
        begin
          MostrarDbgPesqu(Sender);
          dbgPesqu.Height := 125;
          if vMunNumer <> 0 then
            dbgPesqu.Visible := true;
          end;
        end;

    if dbgPesqu.Visible then
      if key = #0 then
        dbgPesqu.SetFocus;
      end;

  procedure TTelForne.ceEnderKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
  begin
    if ceEnder.Text <> '' then
      if tbPesqu.State <> dsInactive then
        tbPesqu.Locate('LogEnder',ceEnder.Text,[loPartialKey]);
      end;
    end.

  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

Cadastro de Bancos

Banco		Cd. Agência	
BRASIL		696	
Cidade	UF	CEP	
IBIRAMA	SC	89140-__	
Logradouro		Número	
RUA DO BRASIL		100	
Bairro	Fone	Fax	
CENTRO	22522	5555	
Contato	Data Cadastramento		
ADRIANA	16/04/2013		

unit UBanco;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, ExtCtrls, Buttons, Mask, DB, DBTables, ToolEdit, Grids, DBGrids;

type

```
TTelBanco = class(TForm)
  pnDados: TPanel;
  Panel4: TPanel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Label12: TLabel;
  edUfMun: TEdit;
  edNrRua: TEdit;
  edbairr: TEdit;
  meNrCep: TMaskEdit;
  meNrFon: TMaskEdit;
  meNrFax: TMaskEdit;
  Label1: TLabel;
  edRespo: TEdit;
  Label2: TLabel;
  BtnGrava: TBitBtn;
  BtnCance: TBitBtn;
  BtnExclu: TBitBtn;
  BtnSaida: TBitBtn;
  edCdAge: TEdit;
  Label3: TLabel;
```

```

ceNomes: TComboEdit;
ceMunic: TComboEdit;
ceEnder: TComboEdit;
Query1: TQuery;
dsPesqu: TDataSource;
tbPesqu: TTable;
Label18: TLabel;
DeDtCad: TDateEdit;
dbgPesqu: TDBGrid;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure FormShow(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure ceNomesKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceMunicButtonClick(Sender: TObject);
procedure ceMunicExit(Sender: TObject);
procedure ceMunicKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceNomesButtonClick(Sender: TObject);
procedure ceEnderButtonClick(Sender: TObject);
procedure ceEnderExit(Sender: TObject);
procedure ceEnderKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure BtnExcluClick(Sender: TObject);
procedure ceNomesKeyPress(Sender: TObject; var Key: Char);
procedure ceNomesKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceMunicEnter(Sender: TObject);
procedure ceMunicKeyPress(Sender: TObject; var Key: Char);
procedure ceMunicKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceEnderEnter(Sender: TObject);
procedure ceEnderKeyPress(Sender: TObject; var Key: Char);
procedure ceEnderKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceNomesEnter(Sender: TObject);
procedure dbgPesquKeyPress(Sender: TObject; var Key: Char);
procedure dbgPesquDbClick(Sender: TObject);
procedure edUfMunKeyPress(Sender: TObject; var Key: Char);
procedure BtnCanceClick(Sender: TObject);
private
{ Private declarations }
vBanNumer,
vMunNumer,
vEndNumer : Longint;
procedure CarregaDadosEndereco;
procedure CarregaDadosCidade;
procedure MostrarDbgPesqu(Sender : TObject);
procedure DeletarBanco;
procedure InserirBanco;

```

```

    procedure AlterarBanco;
    procedure ConfereCidade(Sender : TObject);
    procedure BuscarCidade(Sender : TObject);
    Function  Consiste: Boolean;
    Procedure LimparVariaveis;
    procedure CarregaDadosArquivoParaTela;

public
    { Public declarations }

end;

var
    TelBanco :TTelBanco;

implementation

uses
    UProce,
    UDataM;
{$R *.DFM}

procedure TTelBanco.CarregaDadosEndereco;
begin

    try
        with dm.tbTempo do
            begin
                Close;
                TableName := 'Le' + FormatFloat('00000',vMunNumer);
                Open;
                IndexName := '';
                if FindKey([vEndNumer]) then
                    begin
                        ceEnde.text := FieldByName('LogEnde').AsString;
                        meNrCep.Text := FieldByName('LogNrCep').AsString;
                        edBairr.Text := FieldByName('LogBairr').AsString;
                    end
                else
                    begin
                        ceEnde.text := '';
                        meNrCep.Text := '';
                        edBairr.Text := '';
                    end;
            end;
        end;
    except

    end;
end;

procedure TTelBanco.CarregaDadosCidade;

```

```

begin
with dm.TbMunic do
begin
IndexName := '';
if FindKey([vMunNumer]) then
begin
ceMunic.text := FieldByName('MunCidad').AsString;
edUfMun.Text := FieldByName('MunEstad').AsString;
meNrCep.Text := FieldByName('MunNrCep').AsString;
end
else
begin
ceMunic.text := '';
edUfMun.Text := '';
meNrCep.Text := '';
end;

end;
end;

procedure TTelBanco.MostrarDbgPesqu(Sender : TObject);
begin
dbgPesqu.Top := (Sender as tComboEdit).Top + (Sender as tComboEdit).Height;
dbgPesqu.Left := (Sender as tComboEdit).Left;
dbgPesqu.Width := (Sender as tComboEdit).Width;
dbgPesqu.DataSource := dsPesqu;
dbgPesqu.Visible := true;
dbgPesqu.Columns.Clear;

end;

procedure TTelBanco.DeletarBanco;
begin
with Dm.QDelete do
begin
Close;
Sql.Clear;
Sql.Add('Delete From GrBanco ');
Sql.Add('Where BanNumer = :vNumer ');
ParamByName('vNumer').AsInteger := vBanNumer;
ExecSql;
Close;
end;
end;

procedure TTelBanco.InserirBanco;
begin
with dm.QInserir do
begin
vBanNumer := BuscaNumero('GrBanco', 'BanNumer');
Close;
Sql.Clear;

```

```

Sql.Add('Insert Into GrBanco ');
Sql.Add('(BanNumer, BanNomes, BanNrAge, BanMunic,BanEnder,');
Sql.Add(' BanNrRua, BanNrTel, BanNrFax, BanRespo,BanDtCad,BanUsuar) ');
Sql.Add('Values (:vNumer, :vNomes, :vNrAge, :vMunic, :vEnder, '+
        ':vNrRua, :vNrTel, :vNrFax, :vRespo, :vDtCad, :vUsuar) ');
ParamByName('vNumer').AsInteger := vBanNumer;
ParamByName('vNomes').AsString := ceNomes.Text;
ParamByName('vNrAge').AsString := edCdAge.Text;
ParamByName('vMunic').AsInteger := vMunNumer;
ParamByName('vEnder').AsInteger := vEndNumer;
ParamByName('vNrRua').AsString := edNrRua.Text;
ParamByName('vNrTel').AsString := meNrFon.Text;
ParamByName('vNrFax').AsString := meNrFax.Text;
ParamByName('vRespo').AsString := edRespo.Text;
ParamByName('vDtCad').AsDateTime := deDtCad.Date;
ParamByName('vUsuar').AsInteger := vUsuNumer;
ExecSql;
Close;
vRetorno := vBanNumer;
end;

end;

procedure TTelBanco.AlterarBanco;
begin
    with dm.QInserir do
        begin
            Close;
            Sql.Clear;
            Sql.Add('Update GrBanco ');
            Sql.Add('Set BanNomes = :vNomes, BanNrAge = :vNrAge, BanMunic = :vMunic, ');
            Sql.Add(' BanEnder = :vEnder, BanNrRua = :vNrRua, BanNrTel = :vNrTel, ');
            Sql.Add(' BanNrFax = :vNrFax, BanRespo = :vRespo, BanDtCad = :vDtCad, ');
            Sql.Add(' BanUsuar = :vUsuar ');
            Sql.Add('Where BanNumer = :vNumer ');
            ParamByName('vNomes').AsString := ceNomes.Text;
            ParamByName('vNrAge').AsString := edCdAge.Text;
            ParamByName('vMunic').AsInteger := vMunNumer;
            ParamByName('vEnder').AsInteger := vEndNumer;
            ParamByName('vNrRua').AsString := edNrRua.Text;
            ParamByName('vNrTel').AsString := meNrFon.Text;
            ParamByName('vNrFax').AsString := meNrFax.Text;
            ParamByName('vRespo').AsString := edRespo.Text;
            ParamByName('vDtCad').AsDateTime := deDtCad.Date;
            ParamByName('vUsuar').AsInteger := vUsuNumer;
            ParamByName('vNumer').AsInteger := vBanNumer;
            ExecSql;
            Close;

            end;

end;

```

```

procedure TTelBanco.ConfereCidade(Sender : TObject);
begin
  tbPesqu.close;
  tbPesqu.TableName := 'LeMunic';
  tbPesqu.IndexName := 'Cidad';
  tbPesqu.Open;
  with tbPesqu do
    begin
      if FindKey([ceMunic.Text]) then
        begin
          vMunNumer := FieldByName('MunNumer').AsInteger;
          ceMunic.Text := FieldByName('MunCidad').AsString;
          edUfMun.Text := FieldByName('MunEstad').AsString;
          meNrCep.EditMask := '';
          meNrCep.Text:= FieldByName('MunNrCep').AsString;
          meNrCep.EditMask := '00000-000;0;_';
          vEndNumer := 0;
          ceEnder.Text := '';
          edBairr.Text := '';

          end
        else
          begin
            vMunNumer := 0;
            edUfMun.Text := FieldByName('MunEstad').AsString;
            meNrCep.EditMask := '';
            meNrCep.Text:= '';
            meNrCep.EditMask := '00000-000;0;_';
            vEndNumer := 0;
            ceEnder.Text := '';
            edBairr.Text := '';

            end;
          end;
end;

```

```

procedure TTelBanco.BuscarCidade(Sender : TObject);
begin
end;

```

```

Function TTelBanco.Consiste: Boolean;
Begin
  Result := True;
  if ceNomes.Text = '' then
    Result := False;
end;

```

```

Procedure TTelBanco.LimparVariaveis;

```

```

Begin
  vBanNumer := 0;
  vMunNumer := 0;
  ceNomes.Setfocus;
  vMunNumer := 1;
  ceMunic.Text := 'BLUMENAU';
  meNrCep.EditMask := '';
  meNrCep.Text := '89100000';
  meNrCep.EditMask := '99999-999;1;_';
  edUfMun.Text := 'SC';
  dbgPesqu.Visible := false;
  deDtCad.Date := date;

end;

procedure TTelBanco.CarregaDadosArquivoParaTela;
begin
  with tbPesqu do
    begin
      ceNomes.Text := FieldByName('BanNomes').AsString;
      edCdAge.Text := FieldByName('BanNrAge').AsString;
      vMunNumer := FieldByName('BanMunic').AsInteger;
      vEndNumer := FieldByName('BanEnder').AsInteger;
      edNrRua.Text := FieldByName('BanNrRua').AsString;
      meNrFon.Text := FieldByName('BanNrTel').AsString;
      meNrFax.Text := FieldByName('BanNrFax').AsString;
      edRespo.Text := FieldByName('BanRespo').AsString;
      if vMunNumer > 0 then
        CarregaDadosCidade;

      if vEndNumer > 0 then
        CarregaDadosEndereco;

      meNrCep.EditMask := '99999-999;0;_';
    end;

end;

procedure TTelBanco.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action := caFree;
  TelBanco := Nil;
end;

procedure TTelBanco.FormKeyPress(Sender: TObject; var Key: Char);
begin
  TestaTecla(Self,Key);
end;

procedure TTelBanco.FormShow(Sender: TObject);
begin

```



```

TrazFiguras(TelBanco);
LimparVariaveis;
vMunNumer := 1;
ceMunic.Text := 'BLUMENAU';
meNrCep.EditMask := '';
meNrCep.Text := '89100000';
meNrCep.EditMask := '99999-999;1;_';
edUfMun.Text := 'SC';
end;

procedure TTelBanco.BitBtn2Click(Sender: TObject);
begin
    if ( ActiveControl = BtnSaida) or (ActiveControl = btnCance) ) then
        exit;

    if not Consiste then
        Begin
            MessageDlg('Nome deverá ser preenchido',mtInformation, [mbYes],0);
            ceNomes.SetFocus;
            Exit;
        end;

    if ceMunic.Text <> '' then
        if vMunNumer = 0 then
            begin
                vMunNumer := InserirDadosCidade(ceMunic.Text,
                    edUfMun.Text,
                    meNrCep.Text);
                if vMunNumer = 0 then
                    begin
                        MessageDlg('Erro ao incluir dados da cidade.',
                            mtInformation,[mbOk],0);
                    end;
                end;
            end;

    if ceEnder.Text <> '' then
        if vEndNumer = 0 then
            begin
                vEndNumer := InserirDadosLogradouro(vMunNumer,
                    ceEnder.Text,
                    meNrCep.Text,
                    edBairr.text);
                if vEndNumer = 0 then
                    begin
                        MessageDlg('Erro ao incluir dados da cidade.',
                            mtInformation,[mbOk],0);
                    end;
                end;
            end;

    if vBanNumer = 0 then
        InserirBanco

```

```

else
    AlterarBanco;

    LimparTela(TelBanco);
    LimparVariaveis;
    if not BtnSaida.Enabled then
        Close; // Foi chamado da tela de cadastro de cheques recebidos

end;

procedure TTelBanco.BitBtn8Click(Sender: TObject);
begin
    Close;
end;

procedure TTelBanco.ceNomesKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if not dbgPesqu.Visible then
        begin
            dbgPesqu.Height := 197;
            dbgPesqu.Width := ceNomes.Width;
            dbgPesqu.DataSource := dsPesqu;

            if Key = vk_f2 then
                dbgPesqu.Visible := true;

        end
    else
        if Key = vk_f3 then
            dbgPesqu.Visible := false;

        end
    end;

procedure TTelBanco.ceMunicButtonClick(Sender: TObject);
begin
    MostrarDbgPesqu(Sender);
    ceMunic.SetFocus;
    dbgPesqu.Height := 120;

end;

procedure TTelBanco.ceMunicExit(Sender: TObject);
begin
    if ( ActiveControl = btnCance ) or ( ActiveControl = btnSaida ) then
        exit;

```

```

if not dbgPesqu.Visible then
begin
  if ceMunic.Text <> '' then
    begin
      try
        ConfereCidade(ceMunic);
      except
      end;
    end
  else
    begin
      vMunNumer := 0;
      vEndNumer := 0;
    end;
  end;
end;

end;

procedure TTelBanco.ceMunicKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if Key = vk_f2 then
    begin
      if Not dbgPesqu.Visible then
        begin
          MostrarDbgPesqu(Sender);
          dbgPesqu.Height := 80;
        end;
      end
    else
      if Key = vk_f3 then
        dbgPesqu.Visible := false;

end;

procedure TTelBanco.ceNomesButtonClick(Sender: TObject);
begin
  TelBanco.KeyPreview := false;
  if not dbgPesqu.Visible then
    begin
      dbgPesqu.Height := 197;
      dbgPesqu.Width := ceNomes.Width;
      dbgPesqu.DataSource := dsPesqu;
    end;

  dbgPesqu.Visible := true;
  ceNomes.SetFocus;

end;

```

```

procedure TTelBanco.ceEnderButtonClick(Sender: TObject);
begin
  MostrarDbgPesqu(Sender);
  dbgPesqu.Height := 79;
  dbgPesqu.Top := ceEnder.Top + ceEnder.Height;
  ceEnder.SetFocus;

end;

procedure TTelBanco.ceEnderExit(Sender: TObject);

begin

  if ( ActiveControl = btnSaida) or (ActiveControl = btnCance) ) then
    exit;

  if ceEnder.Text = '' then
    exit;

  if vMunNumer > 0 then
    vEndNumer := VerificaRuaJaCadastrada(vMunNumer,ceEnder.Text);
end;

procedure TTelBanco.ceEnderKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if Key = vk_f2 then
    begin
      if not dbgPesqu.Visible then
        begin
          MostrarDbgPesqu(Sender);
          dbgPesqu.Height := 86;
        end;
    end
  else
    if Key = vk_f3 then
      dbgPesqu.Visible := false;
end;

procedure TTelBanco.BtnExcluClick(Sender: TObject);
begin
  if MessageDlg('Confirma a exclusão do banco?',
    mtInformation,[mbYes,mbNo],0) = mrYes then
    begin
      if DeletarDados('GrBanco','BanNumer',vBanNumer) then
        begin
          LimparTela(TelBanco);
          LimparVariaveis;
        end
      else
        ShowMessage('Erro na exclusão do registro');
    end;
end;

```

```

end;

end;

procedure TTelBanco.ceNomesKeyPress(Sender: TObject; var Key: Char);
begin
  if not dbgPesqu.Visible then
  begin
    dbgPesqu.Top := 44;
    dbgPesqu.Left := ceNomes.Left;
    dbgPesqu.Height := 197;
    dbgPesqu.Width := ceNomes.Width;
    dbgPesqu.DataSource := dsPesqu;
  end;

  if key = #0 then
  begin
    if dbgPesqu.Visible then
      dbgPesqu.SetFocus;
    if ceNomes.Text <> '' then
      tbPesqu.Locate('BanNomes',ceNomes.Text,[loPartialKey]);
    end;
  end;
end;

procedure TTelBanco.ceNomesKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if dbgPesqu.Visible then
  begin
    if ceNomes.Text <> '' then
      tbPesqu.Locate('BanNomes',ceNomes.Text,[loPartialKey]);
    end;
  end;
end;

procedure TTelBanco.ceMunicEnter(Sender: TObject);
begin
  tbPesqu.close;
  tbPesqu.TableName := 'LeMunic';
  tbPesqu.IndexName := 'Cidad';
  tbPesqu.Open;
  tbPesqu.IndexName := 'Cidad';
  dsPesqu.DataSet := tbPesqu;
  dbgPesqu.Columns.Clear;
  dbgPesqu.Columns.Add.FieldName := 'MunCidad';
  dbgPesqu.Columns[0].Title.Caption := 'Município';
  dbgPesqu.Height := 120;
end;

procedure TTelBanco.ceMunicKeyPress(Sender: TObject; var Key: Char);
begin
  if dbgPesqu.Visible then

```

```

    if key = #0 then
        dbgPesqu.SetFocus;
    end;

procedure TTelBanco.ceMunicKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if ceMunic.Text <> '' then
        tbPesqu.Locate('MunCidad',ceMunic.Text,[loPartialKey]);
    end;

procedure TTelBanco.ceEnderEnter(Sender: TObject);
begin
    try
        tbPesqu.close;
        if vMunNumer <> 0 then
            begin
                tbPesqu.TableName := 'Le' + FormatFloat('00000',vMunNumer);
                tbPesqu.IndexName := 'Rua';
                tbPesqu.Open;
                dbgPesqu.Columns.Clear;
                dsPesqu.DataSet := tbPesqu;
                dbgPesqu.Height := 85;
                dbgPesqu.Top := ceEnder.Top + ceEnder.Height;
            end;

        except
            On E:Exception do
                begin
                    MessageDlg(e.message + ' . Tabela de logradouro inexistente.',
                        mtInformation,[mbOk],0);
                end;
            end;
        end;

procedure TTelBanco.ceEnderKeyPress(Sender: TObject; var Key: Char);
begin
    if vMunNumer > 0 then
        if not dbgPesqu.Visible then
            begin
                MostrarDbgPesqu(Sender);
                dbgPesqu.Height := 125;
                if vMunNumer <> 0 then
                    dbgPesqu.Visible := true;
                end;

            if dbgPesqu.Visible then
                if key = #0 then
                    dbgPesqu.SetFocus;
                end;

procedure TTelBanco.ceEnderKeyUp(Sender: TObject; var Key: Word;

```

```

Shift: TShiftState);
begin
  if ceEnder.Text <> '' then
    if tbPesqu.State <> dsInactive then
      tbPesqu.Locate('LogEnder',ceEnder.Text,[loPartialKey]);

end;

procedure TTelBanco.ceNomesEnter(Sender: TObject);
begin
  dbgPesqu.Top := ceNomes.Top + ceNomes.Height;
  dbgPesqu.Left := ceNomes.Left;
  tbPesqu.close;
  tbPesqu.TableName := 'GrBanco';
  tbPesqu.IndexName := 'Nomes';
  tbPesqu.Open;
  dbgPesqu.Parent := pnDados;
  dbgPesqu.Columns.Clear;
  dbgPesqu.Columns.Add.FieldName := 'BanNrAge';
  dbgPesqu.Columns[0].Title.Caption := 'Agência';
  dbgPesqu.Columns.Add.FieldName := 'BanNomes';
  dbgPesqu.Columns[1].Title.Caption := 'Nome';

end;

procedure TTelBanco.dbgPesquKeyPress(Sender: TObject; var Key: Char);
begin
  if key = #13 then
    dbgPesquDbClick(Sender);

end;

procedure TTelBanco.dbgPesquDbClick(Sender: TObject);
begin
  if tbPesqu.TableName = 'GrBanco' then
    begin
      vBanNumer := tbPesqu.FieldByName('BanNumer').AsInteger;
      ceNomes.Text := tbPesqu.FieldByName('BanNomes').AsString;
      CarregaDadosArquivoParaTela;
    end
  else
    if tbPesqu.TableName = 'Le' + FormatFloat('00000',vMunNumer) then
      begin
        vEndNumer := tbPesqu.FieldByName('LogNumer').AsInteger;
        ceEnder.Text := tbPesqu.FieldByName('LogEnder').AsString;
        meNrCep.EditMask := '';
        meNrCep.Text:= tbPesqu.FieldByName('logNrCep').AsString;
        meNrCep.EditMask := '99999-999;0;_';
        edbairr.Text := tbPesqu.FieldByName('LogBairr').AsString;
        edNrRua.SetFocus;
      end
    end;
end;

```

```

end
else
    if tbPesqu.TableName = 'LeMunic' then
        begin
            vMunNumer := tbPesqu.FieldByName('MunNumer').AsInteger;
            ceMunic.Text := tbPesqu.FieldByName('MunCidad').AsString;
            edUfMun.Text := tbPesqu.FieldByName('MunEstad').AsString;
            meNrCep.EditMask := '';
            meNrCep.Text:= tbPesqu.FieldByName('MunNrCep').AsString;
            meNrCep.EditMask := '99999-999;1;_';
            ceEnder.SetFocus;
        end
    else
        begin
            vMunNumer := tbPesqu.FieldByName('MunNumer').AsInteger;
            ceMunic.Text := tbPesqu.FieldByName('MunCidad').AsString;
            edUfMun.Text := tbPesqu.FieldByName('MunEstad').AsString;
            meNrCep.EditMask := '';
            meNrCep.Text:= tbPesqu.FieldByName('MunNrCep').AsString;
            meNrCep.EditMask := '99999-999;1;_';
            ceEnder.SetFocus;
        end;

        dbgPesqu.Visible := false;
        telBanco.KeyPreview := true;

end;

procedure TTelBanco.edUfMunKeyPress(Sender: TObject; var Key: Char);
begin
    if dbgPesqu.Visible then
        dbgPesqu.Visible := false;

end;

procedure TTelBanco.BtnCanceClick(Sender: TObject);
begin
    LimparTela(TelBanco);
    LimparVariaveis;
end;

end.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```


unit UEmpresa;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, Buttons, ToolEdit, Mask, ExtCtrls;

type

```
TFEmpresa = class(TForm)
    pnDados: TPanel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label12: TLabel;
    Label1: TLabel;
    Label3: TLabel;
    edUfCid: TEdit;
    edNrRua: TEdit;
    edbairr: TEdit;
    meNrCep: TMaskEdit;
    edNrTel: TMaskEdit;
    meNrFax: TMaskEdit;
    pnBotao: TPanel;
    BtnGrava: TBitBtn;
    BtnCance: TBitBtn;
    BtnExclu: TBitBtn;
    BtnSaida: TBitBtn;
    Label2: TLabel;
    meCnpj: TMaskEdit;
    edInscEs: TEdit;
    edNomes: TEdit;
```

```

SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
edENDER: TEdit;
edCidade: TEdit;
SpeedButton3: TSpeedButton;
pnSenha: TPanel;
edSenha: TEdit;
Button2: TButton;
edControleAcesso: TEdit;
Label11: TLabel;
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure BtnSaidaClick(Sender: TObject);
procedure BtnGravaClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
private
  { Private declarations }
  vEndNumer,
  vEmpNumer,
  vMunNumer : Longint;
  procedure CarregaDadosCidade;
  procedure CarregaDadosEmpresa;
  procedure CarregaDadosEndereco;
public
  { Public declarations }
end;

var
  FEmpresa: TFEmpresa;

implementation

uses UPesquisa, UDataM, UProce;

{$R *.DFM}

procedure TFEmpresa.CarregaDadosCidade;
begin
  with dm.TbMunic do
    begin
      IndexName := '';
      if FindKey([vMunNumer]) then
        begin
          edCidade.text := FieldByName('MunCidad').AsString;
          edUfCid.Text := FieldByName('MunEstad').AsString;
          meNrCep.Text := FieldByName('MunNrCep').AsString;
        end
      else

```

```

begin
  edCidade.text := "";
  edUfCid.Text := "";
  meNrCep.Text := "";
end;

end;
end;

procedure TFEmpresa.CarregaDadosEmpresa;
begin

  edNomes.Text := dm.QPesquisa.FieldByName('EmpNomes').AsString;
  edNrTel.Text := dm.QPesquisa.FieldByName('EmpFones').AsString;
  meNrFax.Text := dm.QPesquisa.FieldByName('EmpFax').AsString;
  meCnpj.Text := dm.QPesquisa.FieldByName('EmpCnpj').AsString;
  edInscEs.Text := dm.QPesquisa.FieldByName('EmpInscEs').AsString;
  try
    edNrRua.Text := dm.QPesquisa.FieldByName('EmpNrCasa').AsString;
  except
    edNrRua.Clear;
  end;
  vMunNumer := dm.QPesquisa.FieldByName('EmpMunic').AsInteger;
  if vMunNumer > 0 then
    CarregaDadosCidade;
  vEndNumer := Dm.QPesquisa.FieldByName('EmpEnder').AsInteger;
  if vEndNumer > 0 then
    CarregaDadosEndereco;

end;

procedure TFEmpresa.CarregaDadosEndereco;
begin
  try
    with dm.tbTempo do
      begin
        Close;
        TableName := 'Le' + FormatFloat('00000',vMunNumer);
        Open;
        IndexName := "";
        if FindKey([vEndNumer]) then
          begin
            edEnder.text := FieldByName('LogEnder').AsString;
            meNrCep.Text := FieldByName('LogNrCep').AsString;
            edBairr.Text := FieldByName('LogBairr').AsString;
          end
        else
          begin
            edEnder.text := "";
            meNrCep.Text := "";

```

```

        edBairr.Text := '';
    end;

    end;
except

    end;
end;

procedure TFEmpresa.SpeedButton1Click(Sender: TObject);
begin

    if FPesquisa = nil then
        FPesquisa := TFPesquisa.Create(Self);

        fPesquisa.vTabela := 'grEmpre';
        fPesquisa.vCampo := 'EmpNomes';

        FPesquisa.ShowModal;

        if FPesquisa.ModalResult = MrOk then
            begin
                CarregaDadosEmpresa;

            end;

        FPesquisa.Free;
        FPesquisa := nil;

    end;

end;

procedure TFEmpresa.SpeedButton2Click(Sender: TObject);
begin

    if FPesquisa = nil then
        fPesquisa := TFPesquisa.Create(Self);

        fPesquisa.vTabela := 'leMunic';
        fPesquisa.vCampo := 'MunCidad';

        fPesquisa.ShowModal;

        if fPesquisa.ModalResult = mrOk then
            begin
                vMunNumer := dm.QPesquisa.fieldByName('MunNumer').AsInteger;
                edCidade.Text := dm.QPesquisa.fieldByName('MunCidad').asString;
            end;
        end;
    end;
end;

```

```

edUfCid.Text := dm.QPesquisa.fieldByName('MunEstad').asString;
meNrCep.Text := dm.QPesquisa.fieldByName('MunNrCep').asString;
edUfCid.SetFocus;
end;

fPesquisa.Free;
fPesquisa := Nil;

end;

procedure TFEmpresa.SpeedButton3Click(Sender: TObject);
begin
    if fPesquisa = nil then
        fPesquisa := TfPesquisa.Create(Self);

    fPesquisa.vTabela := 'Le' + FormatFloat('00000',vMunNumer);
    fPesquisa.vCampo := 'LogEnder';

    fPesquisa.ShowModal;

    if fPesquisa.ModalResult = mrOk then
        begin
            vEndNumer := dm.QPesquisa.fieldByName('LogNumer').AsInteger;

            edEnder.Text := '';
            CarregaDadosEndereco;
            EdNrRua.SetFocus;
        end;

    fPesquisa.Free;
    fPesquisa := Nil;

    try
        if vMunNumer <> 0 then
            begin
                dm.QConsulta.Close;
                dm.QConsulta.SQL.clear;
                dm.QConsulta.Sql.add('Select * from ' + 'Le' + FormatFloat('00000',vMunNumer));
                dm.QConsulta.Open;

                end;

    except
        On E:Exception do
            begin
                MessageDlg(e.message + ' . Tabela de logradouro inexistente.',
                    mtInformation,[mbOk],0);
            end;
        end;
    end;
end;

```

```

end;

procedure TFEmpresa.BtnSaidaClick(Sender: TObject);
begin
    Close;
end;

procedure TFEmpresa.BtnGravaClick(Sender: TObject);
var
    vCampo : String;
begin
    try
        with dm.tbTempo do
            begin
                Close;
                TableName := 'GrEmpre.db';
                Open;
                vCampo := FieldByName('EmpControle').AsString;
                Close;
            end;
        except
            with Dm.QCriaT do
                begin
                    try
                        dm.tbTempo.Close;
                        Close;
                        Sql.Clear;
                        Sql.Add('Alter Table GrEmpre ');
                        Sql.Add('Add EmpControle varchar(20)');
                        ExecSql;

                    except
                        ShowMessage('Erro ao criar campo de limite de compra na tabela de terceiros');
                    end;
                end;
            end;
        end;
    end;

    try
        if vEmpNumer = 0 then
            begin
                dm.QInserir.close;
                dm.QInserir.SQL.Clear;
                dm.QInserir.SQL.Add('Insert into grEmpre');
                dm.QInserir.SQL.Add('(EmpNumer,EmpNomes,EmpFones,EmpFax,EmpCnpj,EmpInsEs,');
                dm.QInserir.SQL.Add('EmpMunic,EmpEnder,EmpNrCasa, EmpControle)');
                dm.QInserir.SQL.Add(' Values ');
                dm.QInserir.SQL.Add('(:Numer,:Nomes,:Fones,:Fax,:Cnpj,:InsEs,');
                dm.QInserir.SQL.Add(':Munic,:Ender,:NrCasa,:pControle)');
            end
        end
    end

```

```

else
begin
    dm.QInserir.close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('Update grEmpre');
    dm.QInserir.SQL.Add('Set EmpNomes = :Nomes,EmpFones = :Fones,EmpFax = :Fax, ');
    dm.QInserir.SQL.Add('EmpCnpj = :Cnpj,EmpInsEs = :InsEs,');
    dm.QInserir.SQL.Add('EmpMunic = :Munic,EmpEnder = :Ender,EmpNrCasa = :NrCasa,
EmpControle = :pControle');
    dm.QInserir.SQL.Add('Where EmpNumer = :Numer');
end;

dm.QInserir.ParamByName('Numer').AsInteger := 1;
dm.QInserir.ParamByName('Nomes').AsString := edNomes.Text;
dm.QInserir.ParamByName('Fones').AsString := edNrTel.Text;
dm.QInserir.ParamByName('Fax').AsString := meNrFax.Text;
dm.QInserir.ParamByName('Cnpj').AsString := meCnpj.Text;
dm.QInserir.ParamByName('InsEs').AsString := edInscEs.Text;
dm.QInserir.ParamByName('Munic').AsInteger := vMunNumer;
dm.QInserir.ParamByName('Ender').AsInteger := vEndNumer;
dm.QInserir.ParamByName('NrCasa').AsString := edNrRua.Text;
dm.QInserir.ParamByName('pControle').AsString := edControleAcesso.Text;
dm.QInserir.ExecSQL;

vEmpresa := edNomes.Text;

except
end;

end;

procedure TFEmpresa.FormShow(Sender: TObject);
begin
    dm.QPesquisa.close;
    dm.QPesquisa.Sql.Clear;
    dm.QPesquisa.Sql.Add('Select * From GrEmpre');
    dm.QPesquisa.Open;
    vEmpNumer := dm.QPesquisa.FieldByName('EmpNumer').AsInteger;
    CarregaDadosEmpresa;
end;

procedure TFEmpresa.Button2Click(Sender: TObject);
begin
    if edSenha.Text = 'adoanita' then
    begin
        pnBotao.Enabled := true;
        pnSenha.Visible := false;
        edControleAcesso.Visible := true;
    end;
end;
end;

```

```
procedure TFEmpresa.FormKeyUp(Sender: TObject; var Key: Word;  
  Shift: TShiftState);  
begin  
  if key = vk_f4 then  
    pnSenha.Visible := true;  
end;  
  
end.
```

XX

Cadastro de Cheques Recebidos

Cadastro de Cheques Recebidos

Cheque Número: 2523 Pessoa: Física CPF/CNPJ: 111.111.111-11 Agência Banco: 696 BRASIL Data Lançamento: 16/04/2013

Tipo: Chequ Pré Data Entrada: 16/ 4/2013 Data Desconto: 16/ 5/2013 Valor: R\$ 360,80 Data Cadastro: 26/02/2013

Dados do Emissor
 Emitente: ADRIANA DA SILVA Data Nascimento: 12/12/1998 Inscrição Estadual: 542121
 Cidade: BLUMENAU Estado: SC CEP: 89055-060 Rua: RUA DO ENGENHO
 Número: 200 Bairro: FORTALEZA Telefone: 541212 Celular: 54154 E-mail: 2122121

Observação (Terceiro): JOÃO DA SILVA Fone: 454454545 - Identidade: 658222 CPF: 54455525225 RUA DO ENGENHO, 56422 BLUMENAU - FORTALEZA

Limite de Compra 0,00

Terceiros Visualizar cheques do Cliente

Gravar Cancelar Excluir Cheque Excluir Cliente Fechar

Compras de 26/2/2013 à 16/4/2013

Nr Cheque	Data Emissão	Data Desconto	Valor	Com Fundo	Repas	Comp
63252	16/4/2013	16/5/2013	300,2	Sim	Não	Não
2523	16/4/2013	16/5/2013	360,8	Sim	Não	Não
12522	16/3/2013	16/4/2013	526,8	Sim	Sim	Sim
12	26/2/2013	26/3/2013	100	Sim	Não	Sim
1	26/2/2013	26/2/2013	253,25	Sim	Não	Sim

Data Cadastro: 26/02/2013
 Total Compra Período: 1.841,05
 Total Compra à Vencer: 661,00

Cientes com cheques sem fundo

Cadastro de Cheques Recebidos

Cheque Número: 2523 Pessoa: Física CPF/CNPJ: 111.111.111-11 Agência Banco: 696 BRASIL Data Lançamento: 16/04/2013

Tipo: Chequ Pré Data Entrada: 16/ 4/2013 Data Desconto: 16/ 5/2013 Valor: R\$ 360,80 Data Cadastro: 26/02/2013

Dados do Emissor
 Emitente: ADRIANA DA SILVA Data Nascimento: 12/12/1998 Inscrição Estadual: 542121
 Cidade: BLUMENAU Estado: SC CEP: 89055-060 Rua: RUA DO ENGENHO
 Número: 200 Bairro: FORTALEZA Telefone: 541212 Celular: 54154 E-mail: 2122121

Observação (Terceiro): JOÃO DA SILVA Fone: 454454545 - Identidade: 658222 CPF: 54455525225 RUA DO ENGENHO, 56422 BLUMENAU - FORTALEZA

CLIENTE COM CHEQUE SEM FUNDO

Limite de Compra 0,00

Terceiros Visualizar cheques do Cliente

Gravar Cancelar Excluir Cheque Excluir Cliente Fechar

Compras de 26/2/2013 à 16/4/2013

Nr Cheque	Data Emissão	Data Desconto	Valor	Com Fundo	Repas	Comp
63252	16/4/2013	16/5/2013	300,2	Sim	Não	Não
2523	16/4/2013	16/5/2013	360,8	Sim	Não	Não
12522	16/3/2013	16/4/2013	526,8	Sim	Sim	Não
12	26/2/2013	26/3/2013	100	Não	Não	Sim
1	26/2/2013	26/2/2013	253,25	Sim	Não	Sim

Data Cadastro: 26/02/2013
 Total Compra Período: 1.841,05
 Total Compra à Vencer: 661,00

unit UCheRe;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
 Grids, DBGrids, StdCtrls, CurrEdit, ToolEdit, Mask, Buttons, ExtCtrls,
 Db, DBTables, Menus;

type

TTelCheRe = class(TForm)

pnDados: TPanel;

gbClien: TGroupBox;
Label3: TLabel;
Label7: TLabel;
ceEmite: TComboEdit;
meObser: TMemo;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label17: TLabel;
Label19: TLabel;
Label20: TLabel;
Label21: TLabel;
Label23: TLabel;
meNrCep: TMaskEdit;
EdBairr: TEdit;
EdEmail: TEdit;
EdIdent: TEdit;
EdNrRua: TEdit;
EdInscr: TEdit;
edNrTel: TEdit;
edNrFax: TEdit;
edUfMun: TEdit;
ceMunic: TComboEdit;
ceEnder: TComboEdit;
Label24: TLabel;
deDtNas: TDateEdit;
btnNovoE: TBitBtn;
btnAlter: TBitBtn;
btnTerce: TBitBtn;
Timer1: TTimer;
dsPesqu: TDataSource;
gbCompra: TGroupBox;
Panel1: TPanel;
LtotalCompra: TLabel;
LCompraVencer: TLabel;
LCadastro: TLabel;
PnBotao: TPanel;
BtnGrava: TBitBtn;
btnCance: TBitBtn;
btnSaida: TBitBtn;
btnExclu: TBitBtn;
tbnExcl2: TBitBtn;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
dbgRelatorio: TDBGrid;
btnFundo: TBitBtn;
PopupMenu1: TPopupMenu;
BaixarCheque1: TMenuItem;

```

ChequeSemFundo1: TMenuItem;
ChequeComFundo1: TMenuItem;
ChequeNoBaixado1: TMenuItem;
GroupBox1: TGroupBox;
Label1: TLabel;
Label2: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label8: TLabel;
Label22: TLabel;
Label16: TLabel;
Label9: TLabel;
Label25: TLabel;
IMensagem: TLabel;
Label18: TLabel;
ceBanco: TComboEdit;
cbTipos: TComboBox;
deDtDes: TDateEdit;
curValor: TCurrencyEdit;
deDtEmi: TDateEdit;
ceNrChe: TComboEdit;
deDtLan: TDateEdit;
meCpfCg: TMaskEdit;
CbTpPes: TComboBox;
edAgencia: TEdit;
DeDtCad: TDateEdit;
edLimite: TEdit;
ILimite: TLabel;
popupCor: TPopupMenu;
ConfigurarCor1: TMenuItem;
dbgPesqu: TDBGrid;
pnMensa: TPanel;
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure FormShow(Sender: TObject);
procedure ceBancoButtonClick(Sender: TObject);
procedure ceBancoEnter(Sender: TObject);
procedure ceBancoExit(Sender: TObject);
procedure ceBancoKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure btnSaidaClick(Sender: TObject);
procedure BtnGravaClick(Sender: TObject);
procedure ceNrCheExit(Sender: TObject);
procedure btnCanceClick(Sender: TObject);
procedure btnExcluClick(Sender: TObject);
procedure cbTiposExit(Sender: TObject);
procedure curValorKeyPress(Sender: TObject; var Key: Char);
procedure ceEmiteButtonClick(Sender: TObject);
procedure CbTpPesExit(Sender: TObject);
procedure ceMunicButtonClick(Sender: TObject);
procedure ceEmiteKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);

```

```

procedure ceMunicExit(Sender: TObject);
procedure ceMunicKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceEnderEnter(Sender: TObject);
procedure ceEnderExit(Sender: TObject);
procedure ceEnderKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure btnNovoEClick(Sender: TObject);
procedure btnExcl2Click(Sender: TObject);
procedure meCpfCgExit(Sender: TObject);
procedure btnAlterClick(Sender: TObject);
procedure btnTerceClick(Sender: TObject);
procedure EdIdentExit(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure edAgenciaExit(Sender: TObject);
procedure cbTiposEnter(Sender: TObject);
procedure ceMunicKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceEnderKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceBancoKeyPress(Sender: TObject; var Key: Char);
procedure ceBancoKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure dbgPesquDblClick(Sender: TObject);
procedure dbgPesquKeyPress(Sender: TObject; var Key: Char);
procedure edAgenciaEnter(Sender: TObject);
procedure ceEmiteEnter(Sender: TObject);
procedure ceEmiteKeyPress(Sender: TObject; var Key: Char);
procedure ceEmiteKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ceMunicEnter(Sender: TObject);
procedure ceMunicKeyPress(Sender: TObject; var Key: Char);
procedure ceEnderKeyPress(Sender: TObject; var Key: Char);
procedure EdNrRuaEnter(Sender: TObject);
procedure deDtDesExit(Sender: TObject);
procedure dbgRelatorioDrawColumnCell(Sender: TObject;
  const Rect: TRect; DataCol: Integer; Column: TColumn;
  State: TGridDrawState);
procedure btnFundoClick(Sender: TObject);
procedure ChequeSemFundo1Click(Sender: TObject);
procedure ChequeComFundo1Click(Sender: TObject);
procedure ChequeNoBaixado1Click(Sender: TObject);
procedure BaixarCheque1Click(Sender: TObject);
procedure edLimiteKeyPress(Sender: TObject; var Key: Char);
procedure ConfigurarCor1Click(Sender: TObject);
procedure curValorExit(Sender: TObject);

```

```

private
{ Private declarations }
vTerNumer,
vMunNumer,
vCliNumer,

```

```

vCheNumer,
vEndNumer,
vBanNumer  : Longint;
vAlterarDadosCliente : Boolean;
vPesquisa : String;

procedure cadastrarChequeSemFundo(Situacao : String);

procedure pVerificaSeClienteJaEmitiuChequeSemFundo;
procedure pMontaRelatorioCompras;
procedure pMontaConsulta;
procedure pCarregaDadosBanco(vBanco : Longint);
procedure pGravarCPFArquivoTexto;
function fVerificarClienteComChequeSemFundo : Boolean;
function ChamarTelaCadastroBanco : Boolean;
procedure ConfereCidade(Sender : TObject);
procedure MostrarDbgPesqu(Sender : TObject);
function InserirDadosCliente : Boolean;
function AlterarDadosCliente : Boolean;
procedure LimparDadosCliente;
procedure HabilitaCampos;
procedure CarregaDadosEndereco;
procedure CarregaDadosCidade;
procedure CarregaDadosArquivoClienteParaTela;
function VerificaClinteJaCadastrado : Boolean;
function Consiste : Boolean;
procedure AlterarCheque;
procedure InserirCheque;
procedure LimparVariaveis;
procedure CarregaDadosArquivoParaTela;
function VerificaChequeJaCadastrado : Boolean;

public
{ Public declarations }
end;

var
  TelCheRe: TTelCheRe;
  ArqTexto : Text;

implementation
Uses
  UProce, UDataM, UBanco, UCadTerceiros, UPesquisa,
  UCheFundoEmiCliente, GerCorClienteSemFundo;

{$R *.DFM}

procedure TTelCheRe.cadastrarChequeSemFundo(Situacao : String);
begin
  try

```

```

dm.QConsulta.close;
dm.QConsulta.Sql.Clear;
dm.QConsulta.Sql.Add('Select * from grChSFun ');
dm.QConsulta.Sql.Add('Where ConClien = :p1');
dm.QConsulta.Sql.Add('And ConNrChe = :p2');
dm.QConsulta.ParamByName('p1').AsInteger := vCliNumer;
dm.QConsulta.ParamByName('p2').AsString :=
dm.QPesCompras.FieldByName('CheNrChe').AsString;
dm.QConsulta.Open;
if dm.QConsulta.FieldByName('ConCodig').AsInteger = 0 then
begin
  dm.QInserir.Close;
  dm.QInserir.Sql.Clear;
  dm.QInserir.Sql.Add('Insert into grChSFun ');
  dm.QInserir.Sql.Add('(ConCodig,ConDatas,ConClien,ConChequ,
ConValor,ConTerce,ConNrChe,ConSitua)');
  dm.QInserir.Sql.Add('Values');
  dm.QInserir.Sql.Add('(:p1,:p2,:p3,:p4,:p5,:p6,:p7,:pSitua)');
  dm.QInserir.ParamByName('p1').AsInteger := BuscaNumero('GrChSFun','ConCodig');
  dm.QInserir.ParamByName('p2').AsDateTime := Date;
  dm.QInserir.ParamByName('p3').AsInteger := vCliNumer;
  dm.QInserir.ParamByName('p4').AsInteger :=
dm.QPesCompras.FieldByName('CheNumer').AsInteger;
  dm.QInserir.ParamByName('p5').AsCurrency :=
dm.QPesCompras.FieldByName('CheValor').AsCurrency;
  dm.QInserir.ParamByName('p6').AsInteger :=
dm.QPesCompras.FieldByName('cheTerce').AsInteger;
  dm.QInserir.ParamByName('p7').AsString :=
dm.QPesCompras.FieldByName('CheNrChe').AsString;
  dm.QInserir.ParamByName('pSitua').AsString := Situacao;
  dm.QInserir.ExecSql;
end
else
  alterarSituacao(vCliNumer, dm.QPesCompras.FieldByName('CheNrChe').AsString, 'Sem
Fundo');

except
  ShowMessage('Erro ao cadastrar cheque sem fundo - grChsFun')
end;

end;

procedure TTelCheRe.pVerificaSeClienteJaEmitiuChequeSemFundo;
begin
  dm.QConCheFundo.Close;
  dm.QConCheFundo.SQL.Clear;
  dm.QConCheFundo.SQL.Add('Select * from grChSFun ');
  dm.QConCheFundo.SQL.Add('where ConClien = :p1');
  dm.QConCheFundo.ParamByName('p1').AsInteger := vCliNumer;
  dm.QConCheFundo.Open;
  if not dm.QConCheFundo.IsEmpty then

```

```

    btnFundo.Visible := true
else
    btnFundo.Visible := false;

end;

procedure TTelCheRe.pMontaRelatorioCompras;
var
    vTotalCompra,vTotalAberto : Currency;
begin

    dm.QPesCompras.Close;
    dm.QPesCompras.SQL.Clear;
    dm.QPesCompras.SQL.Add('Select
CheNumer,CheNrChe,CheDtEmi,CheDtDes,CheValor,CheFundo,');
    dm.QPesCompras.SQL.Add('CheDtCad,CheRepas,CheBaixa,cheClien,cheTerce');
    dm.QPesCompras.SQL.Add('From grCheRe');
    dm.QPesCompras.SQL.Add('Where CheClien = :p1');
    dm.QPesCompras.SQL.Add('Order By CheDtDes desc');
    dm.QPesCompras.ParamByName('p1').AsInteger := vCliNumer;
    dm.QPesCompras.Open;

    vTotalCompra := 0;
    vTotalAberto := 0;

    while not dm.QPesCompras.EOF do
    begin
        vTotalCompra := vTotalCompra + dm.QPesCompras.fieldByName('CheValor').asCurrency;
        if dm.QPesCompras.fieldByName('CheDtDes').AsDateTime > Date then
            vTotalAberto := vTotalAberto + dm.QPesCompras.fieldByName('CheValor').asCurrency;

        dm.QPesCompras.Next;
    end;

    LttotalCompra.Caption := FormatFloat('###,##0.00',vTotalCompra);

    LCompraVencer.Caption := FormatFloat('###,##0.00',vTotalAberto);

    LCadastro.Caption := DeDtCad.Text;
    gbCompra.Caption := DateToStr(dm.QPesCompras.fieldByName('CheDtCad').AsDateTime);

    gbCompra.Caption :=
        'Compras de ' + DateToStr(dm.QPesCompras.fieldByName('CheDtCad').AsDateTime);

    dm.QPesCompras.First;
    gbCompra.Caption := gbCompra.Caption +
        ' à ' + DateToStr(dm.QPesCompras.fieldByName('CheDtCad').AsDateTime);

end;

```

```

procedure TTelCheRe.pMontaConsulta;
var
  vStr : String;
begin

  if vPesquisa = '' then
    exit;

  dbgPesqu.DataSource := dm.dsPesquisa;
  if vPesquisa = 'grBanco' then
    begin
      vStr := 'Select * From grBanco order By BanNomes';
    end
  else
    if vPesquisa = 'grClien' then
      vStr := 'Select * From grClien order By CliNomes'
    else
      if vPesquisa = 'leMunic' then
        vStr := 'Select * From leMunic order By MunCidad'
      else
        vStr := 'Select * From ' + vPesquisa + ' order By logEnder';

      with dm.QPesquisa do
        begin
          close;
          sql.Clear;
          sql.add(vStr);
          open;
        end;

      end;

procedure TTelCheRe.pCarregaDadosBanco(vBanco : Longint);
begin
  with dm.QConsulta do
    begin
      Close;
      Sql.Clear;
      Sql.Add('Select * from grBanco where BanNumer = :1');
      ParamByName('1').AsInteger := vBanco;
      Open;
      ceBanco.Text := FieldByName('BanNomes').AsString;
      edAgencia.Text := FieldByName('BanNrAge').AsString;

    end;

  end;

procedure TTelCheRe.pGravarCPFArquivoTexto;
begin

```



```

AssignFile(ArqTexto,vDiretorio + '\Cadastro.txt');
Append(ArqTexto);
write(ArqTexto,meCpfCg.Text + ';' + ceEmite.Text + ';0'); // nao tem cheque sem fundo
if vEndNumer > 0 then
  writeln(ArqTexto,';1') // 1 = tem endereço
else
  writeln(ArqTexto,';0'); // 0 = nao tem endereço

CloseFile(ArqTexto);

end;

function TTelCheRe.fVerificarClienteComChequeSemFundo : Boolean;
begin
  with dm.QConsulta do
    begin
      Close;
      Sql.Clear;
      Sql.Add('Select * from GrCheRe ');
      Sql.Add('Where CheClien = :vClien ');
      Sql.Add('and (CheFundo = true) ');
      ParamByName('vClien').AsInteger := vCliNumer;
      Open;
      if RecordCount > 0 then
        begin
          Result := true;
          if vMostraMensagemClientesSemFundo then
            begin
              pnMensa.Visible := true;
              Timer1.Enabled := true;
            end;
          gbClien.Color := vCorClienteSemFundo;
        end
      else
        begin
          Result := false;
          pnMensa.Visible := false;
          gbClien.Color := clBtnFace;
          Timer1.Enabled := false;
        end;
      end;
    end;

end;

end;

function TTelCheRe.ChamarTelaCadastroBanco : Boolean;
begin

  if TelBanco = nil then
    TelBanco := TTelBanco.Create(Self);

  TelBanco.ceNomes.Text := ceBanco.Text;

```

```

TelBanco.BtnSaida.Enabled := false;
TelBanco.btnExclu.Enabled := false;
vMensagem := 'Incluir Cheques';
TelBanco.ShowModal;
vBanNumer := vRetorno;
if vRetorno > 0 then
    Result := true
else
    Result := false;
vRetorno := 0;

end;

procedure TTelCheRe.ConfereCidade(Sender : TObject);
begin
    with dm.QPesquisa do
        begin
            close;
            sql.clear;
            sql.add('select * from grMunic where MunCidad = ' +
                chr(39) + ceMunic.text + chr(39));

            if fieldByName('MunCidad').asString = ceMunic.Text then
                begin
                    vMunNumer := FieldByName('MunNumer').AsInteger;
                    ceMunic.Text := FieldByName('MunCidad').AsString;
                    edUfMun.Text := FieldByName('MunEstad').AsString;
                    meNrCep.EditMask := '';
                    meNrCep.Text:= FieldByName('MunNrCep').AsString;
                    meNrCep.EditMask := '00000-000;0;_';
                    vEndNumer := 0;
                    ceEnder.Text := '';
                    edBairr.Text := '';

                    end
                else
                    begin
                        vMunNumer := 0;
                        edUfMun.Text := FieldByName('MunEstad').AsString;
                        meNrCep.EditMask := '';
                        meNrCep.Text:= '';
                        meNrCep.EditMask := '00000-000;0;_';
                        vEndNumer := 0;
                        ceEnder.Text := '';
                        edBairr.Text := '';

                        end;
                    end;
                end;

procedure TTelCheRe.MostrarDbgPesqu(Sender : TObject);

```

```

begin
    TelCheRe.KeyPreview := false;
    dbgPesqu.Top := (Sender as tComboEdit).Top + (Sender as tComboEdit).Height;
    dbgPesqu.Left := (Sender as tComboEdit).Left;
    dbgPesqu.Width := (Sender as tComboEdit).Width;
    dbgPesqu.DataSource := dsPesqu;
    dbgPesqu.Visible := true;

end;

function TTelCheRe.InserirDadosCliente : Boolean;
begin
    try
        with Dm.QInserir do
            begin
                vCliNumer := BuscaNumero('GrClien','CliNumer');
                Close;
                Sql.Clear;
                Sql.Add('Insert Into GrClien ');
                Sql.Add('(CliNumer,CliNomes,CliDtNas,CliTpPes,CliNrDoc,CliInsEs, ');
                Sql.Add('CliIdent,CliMunic,CliEnder,CliNrRua, ');
                Sql.Add('CliNrTel,CliNrFax,CliEmail,CliDtCad,CliLimite)');
                Sql.Add(' Values (:vNumer,:vNomes,:vDtNas,:vTpPes,:vNrDoc,:vInsEs,');
                Sql.Add('      :vIdent,:vMunic,:vEnder,:vNrRua,');
                Sql.Add('      :vNrTel,:vNrFax,:vEmail,:vDtCad,:vLimite)');
                ParamByName('vNumer').AsInteger := vCliNumer;
                ParamByName('vNomes').AsString := ceEmite.Text;
                ParamByName('vDtNas').AsDateTime := deDtNas.Date;
                ParamByName('vTpPes').AsInteger := cbTpPes.ItemIndex + 1;
                ParamByName('vNrDoc').AsString := DeixaSoNumero(meCpfCg.Text);
                ParamByName('vIdent').AsString := edIdent.Text;
                ParamByName('vInsEs').AsString := EdInscr.Text;
                ParamByName('vMunic').AsInteger := vMunNumer;
                ParamByName('vEnder').AsInteger := vEndNumer;
                ParamByName('vNrRua').AsString := edNrRua.Text;
                ParamByName('vNrTel').AsString := EdNrTel.Text;
                ParamByName('vNrFax').AsString := EdNrFax.Text;
                ParamByName('vEmail').AsString := EdEmail.Text;
                ParamByName('vDtCad').AsDateTime := deDtCad.Date;
                ParamByName('vLimite').AsCurrency := StrToFloat(edLimite.text);

                ExecSql;
                Result := true;
            end;
        except
            On E:Exception do
                begin
                    vMensagem := E.Message;
                    Result := false;
                end;
            end;
        end;
    end;
end;

```

```

function TTelCheRe.AlterarDadosCliente : Boolean;
begin
  try
    with Dm.QInserir do
      begin
        Close;
        Sql.Clear;
        Sql.Add('UpDate GrClien ');
        Sql.Add('Set CliNomes = :vNomes,CliDtNas = :vDtNas,CliTpPes = :vTpPes,CliNrDoc =
:vNrDoc, ');
        Sql.Add(' CliInsEs = :vInsEs,CliIdent = :vIdent, ');
        Sql.Add(' CliMunic = :vMunic,CliEnder = :vEnder,CliNrRua = :vNrRua, ');
        Sql.Add(' CliNrTel = :vNrTel,CliNrFax = :vNrFax,CliEmail = :vEmail, ');
        Sql.Add(' CliDtCad = :vDtCad, CliLimite = :vLimite ');
        Sql.Add(' Where CliNumer = :vNumer ');
        ParamByName('vNumer').AsInteger := vCliNumer;
        ParamByName('vNomes').AsString := ceEmite.Text;
        ParamByName('vDtNas').AsDateTime := deDtNas.Date;
        ParamByName('vTpPes').AsInteger := cbTpPes.ItemIndex + 1;
        ParamByName('vNrDoc').AsString := meCpfCg.Text;
        ParamByName('vIdent').AsString := edIdent.Text;
        ParamByName('vInsEs').AsString := EdInscr.Text;
        ParamByName('vMunic').AsInteger := vMunNumer;
        ParamByName('vEnder').AsInteger := vEndNumer;
        ParamByName('vNrRua').AsString := edNrRua.Text;
        ParamByName('vNrTel').AsString := EdNrTel.Text;
        ParamByName('vNrFax').AsString := EdNrFax.Text;
        ParamByName('vEmail').AsString := EdEmail.Text;
        ParamByName('vDtCad').AsDateTime := deDtCad.Date;
        ParamByName('vLimite').AsCurrency := StrToFloat(edLimite.text);

        ExecSql;
        Result := true;
      end;
    except
      On E:Exception do
        begin
          vMensagem := E.Message;
          Result := false;
        end;
      end;
    end;
  end;

procedure TTelCheRe.LimparDadosCliente;
begin
  vTerNumer := -1;
  vCliNumer := 0;
  ceEmite.Text := '';
  deDtNas.Clear;
  EdInscr.Text := '';
  EdIdent.Text := '';

```

```

EdNrRua.Text := "";
edNrTel.Text := "";
edNrFax.Text := "";
EdEmail.Text := "";
ceMunic.Text := "";
ceEnder.Text := "";
edBairr.Text := "";
meNrCep.Text := "";
edNrRua.Text := "";
meObser.Lines.Clear;
edLimite.Text := '0,00';
vMunNumer := 0;
vEndNumer := 0;
end;

procedure TTelCheRe.HabilitaCampos;
begin
  if cbTpPes.ItemIndex = 1 then
  begin
    EdIdent.Enabled := false;
    EdIdent.Color := clScrollBar;
    EdInscr.Enabled := true;
    EdInscr.Color := clWindow;
    meCpfCg.EditMask := '00.000.000/0000-00;0;';

  end
  else
  begin
    EdIdent.Enabled := true;
    EdIdent.Color := clWindow;
    EdInscr.Enabled := false;
    EdInscr.Color := clScrollBar;
    meCpfCg.EditMask := '000.000.000-00;0;';
  end;

  meNrCep.EditMask := '99999-999;0;_';
end;

procedure TTelCheRe.CarregaDadosEndereco;
begin
  try
    with dm.QConsulta do
      begin
        ceEnder.text := FieldByName('LogEnder').AsString;
        meNrCep.Text := FieldByName('LogNrCep').AsString;
        edBairr.Text := FieldByName('LogBairr').AsString;
      end;
    except
  end;
end;
end;

```

```

procedure TTelCheRe.CarregaDadosCidade;
begin
with dm.QConsulta do
begin
close;
sql.clear;
sql.add('Select * from leMunic where MunNumer = ' + InttoStr(vMunNumer));
open;
ceMunic.Text := FieldByName('MunCidad').AsString;
edUfMun.Text := FieldByName('MunEstad').AsString;
meNrCep.Text := FieldByName('MunNrCep').AsString;

end;

end;

procedure TTelCheRe.CarregaDadosArquivoClienteParaTela;
begin
with dm.QPesquisa do
begin
vCliNumer := FieldByName('CliNumer').AsInteger;
fVerificarClienteComChequeSemFundo;
ceEmite.Text := FieldByName('CliNomes').AsString;
deDtNas.Date := FieldByName('CliDtNas').AsDateTime;
CbTpPes.ItemIndex := FieldByName('CliTpPes').AsInteger - 1;
meCpfCg.Text := FieldByName('CliNrDoc').AsString;
EdInscr.Text := FieldByName('CliInsEs').AsString;
EdIdent.Text := FieldByName('CliIdent').AsString;
EdNrRua.Text := FieldByName('CliNrRua').AsString;
edNrTel.Text := FieldByName('CliNrTel').AsString;
edNrFax.Text := FieldByName('CliNrFax').AsString;
EdEmail.Text := FieldByName('CliEmail').AsString;
deDtCad.Date := FieldByName('CliDtCad').AsDateTime;
vMunNumer := FieldByName('CliMunic').AsInteger;
vEndNumer := FieldByName('CliEnder').AsInteger;
edLimite.Text := FormatFloat('###,##0.00',FieldByName('CliLimite').AsCurrency);

if vMunNumer > 0 then
CarregaDadosCidade;

if vEndNumer > 0 then
begin
dm.QConsulta.Close;
dm.QConsulta.SQL.Clear;
dm.QConsulta.SQL.Add('Select * from Le' + FormatFloat('00000',vMunNumer) +
' where LogNumer = ' + IntToStr(vEndNumer));
dm.QConsulta.Open;

CarregaDadosEndereco;
end;

```

```

    HabilitaCampos;

    btnAlter.Visible := true;

end;

end;

function TTelCheRe.VerificaClinteJaCadastrado : Boolean;
begin

    dm.QPesquisa.Close;
    dm.QPesquisa.sql.clear;
    dm.QPesquisa.sql.add('Select * from grClien ' +
        'where CliNrDoc = ' + chr(39) + meCpfCg.Text + chr(39));
    dm.QPesquisa.open;

    if dm.QPesquisa.FieldByName('CliNrDoc').AsString = meCpfCg.Text then
        begin
            Result := true;
            if vUsuario <> 1 then
                btnNovoE.Visible := true;
            end
        else
            begin

                btnNovoE.Visible := false;
                Result := false;
                pnMensa.Visible := false;
            end;

        end;

end;

function TTelCheRe.Consiste : Boolean;
begin
    Result := true;
    if not vAlterarDadosCliente then
        begin
            if ceNrChe.Text = '' then
                begin
                    MessageDlg('Informe o número do cheque.', mtInformation,[mbOk],0);
                    ceNrChe.SetFocus;
                    Result := false;
                    exit;
                end;

            if vBanNumer = 0 then
                begin
                    MessageDlg('Banco não foi Selecionado.', mtInformation,[mbOk],0);
                    ceBanco.SetFocus;
                    Result := false;
                end;
            end;
        end;
end;

```

```

        exit;
    end;

    if cbTipos.ItemIndex = -1 then
    begin
        MessageDlg('Informe o tipo do cheque (Bom/Pré).', mtInformation,[mbOk],0);
        cbTipos.SetFocus;
        Result := false;
        exit;
    end;

    if ( (deDtEmi.Text = ' / / ') or (deDtEmi.Text = ' / / ') ) then
    begin
        MessageDlg('Informe a data de emissão do cheque.', mtInformation,[mbOk],0);
        deDtEmi.SetFocus;
        Result := false;
        exit;
    end;

    if ( (deDtDes.Text = ' / / ') or (deDtDes.Text = ' / / ') ) then
    begin
        MessageDlg('Informe a data de desconto do cheque.', mtInformation,[mbOk],0);
        deDtDes.SetFocus;
        Result := false;
        exit;
    end;

    if curValor.Text = '0' then
    begin
        MessageDlg('Valor do cheque não foi informado.', mtInformation,[mbOk],0);
        curValor.SetFocus;
        Result := false;
        exit;
    end;
end;

if ceEmiter.Text = '' then
begin
begin
    MessageDlg('Emitente não foi informado.', mtInformation,[mbOk],0);
    ceEmiter.SetFocus;
    Result := false;
    exit;
end;
end;

end;
end;

procedure TTelCheRE.InserirCheque;
var
    vStr : String;

```



```

i : integer;
vNumero : Longint;
begin

    with dm.QInserir do
    begin
        vNumero := BuscaNumero('GrCheRe','CheNumer');
        Close;
        Sql.Clear;
        Sql.Add('Insert Into GrCheRe ');
        Sql.Add('(CheNumer, CheNrChe, CheBanco, Chetipos, CheDtEmi, ');
        Sql.Add(' CheDtDes, CheCeden, CheValor, CheObser, CheBaixa, ');
        Sql.Add('
CheDtCad,CheFundo,CheRepas,CheClien,CheDtMod,CheNrDoc,CheTerce,CheMesCo) ');
        Sql.Add('Values (:vNumer, :vNrChe, :vBanco, :vTipos, :vDtEmi, :vDtDes, ');
        Sql.Add('      :vCeden, :vValor, :vObser, :vBaixa, :vDtCad, :vFundo, ');
        Sql.Add('      :vRepas, :vClien, :vDtMod,:vNrDoc,:vTerce,:vMesCo) ');

        ParamByName('vNumer').AsInteger := vNumero;
        ParamByName('vNrChe').AsString := ceNrChe.Text;
        ParamByName('vBanco').AsInteger := vBanNumer;
        ParamByName('vTipos').AsInteger := cbTipos.ItemIndex + 1;
        ParamByName('vDtEmi').AsDate := StrToDate(deDtEmi.Text);
        ParamByName('vDtDes').AsDate := StrToDate(deDtDes.Text);
        ParamByName('vCeden').AsString := fBuscaCliente(vCliNumer);
        //      ceEmite.Text;

        ParamByName('vValor').AsCurrency := curValor.value;
        ParamByName('vObser').AsString := meObser.Text;
        ParamByName('vDtCad').AsDateTime := deDtLan.Date;
        ParamByName('vBaixa').AsBoolean := false;
        ParamByName('vFundo').AsBoolean := false;
        ParamByName('vRepas').AsBoolean := false;
        ParamByName('vClien').AsInteger := vCliNumer;
        ParamByName('vDtMod').AsDateTime := Date;
        ParamByName('vNrDoc').AsString := DeixaSoNumero(meCpfCg.Text);
        ParamByName('vTerce').AsInteger := vTerNumer;

        try
            vstr := FormatDateTime('dd/MM/yyyy',StrToDate(deDtDes.Text));
            i := pos('/',vstr);
            ParamByName('vMesCo').AsString := Copy(vStr,i + 1,Length(vStr));
        except
        end;

        ExecSql;
        Close;

    end;

end;

procedure TTelCheRe.AlterarCheque;

```

```

var
  vStr : String;
  i : integer;
begin
  with dm.QInserir do
    begin
      Close;
      Sql.Clear;
      Sql.Add('UpDate GrCheRe ');
      Sql.Add('Set CheNrChe = :vNrChe, CheBanco = :vBanco, Chetipos = :vTipos, '+
        'CheDtEmi = :vDtEmi, CheDtDes = :vDtDes, CheCeden = :vCeden, '+
        'CheValor = :vValor, CheObser = :vObser, CheDtCad = :vDtCad, ' +
        'CheClien = :vClien, CheDtMod = :vDtMod, CheNrDoc = :vNrDoc, ' +
        'CheTerce = :vTerce, CheMesCo = :vMesCo');
      Sql.Add('Where CheNumer = :vNumer ');
      ParamByName('vNrChe').AsString := ceNrChe.Text;
      ParamByName('vBanco').AsInteger := vBanNumer;
      ParamByName('vTipos').AsInteger := cbTipos.ItemIndex + 1;
      ParamByName('vDtEmi').AsDate := StrToDate(deDtEmi.Text);
      ParamByName('vDtDes').AsDate := StrToDate(deDtDes.Text);
      ParamByName('vCeden').AsString := ceEmite.Text;
      ParamByName('vValor').AsCurrency := curValor.value;
      ParamByName('vObser').AsString := meObser.Text;
      ParamByName('vDtCad').AsDateTime := deDtLan.Date;
      ParamByName('vNumer').AsInteger := vCheNumer;
      ParamByName('vClien').AsInteger := vCliNumer;
      ParamByName('vDtMod').AsDateTime := Date;
      ParamByName('vNrDoc').AsString := meCpfCg.Text;
      ParamByName('vTerce').AsInteger := vTerNumer;

      try
        vstr := FormatDateTime('dd/MM/yyyy',StrToDate(deDtDes.Text));
        i := pos('/',vStr);
        ParamByName('vMesCo').AsString := Copy(vStr,i + 1,Length(vStr));
      except
      end;

      ExecSql;
      Close;

    end;

  end;

end;

procedure TTelCheRe.LimparVariaveis;
begin
  vTerNumer := 0; // Controle de cheques de terceiros
  vCheNumer := 0;
  vBanNumer := 0;
  vCliNumer := 0;
  vMunNumer := 0;

```

```

vEndNumer := 0;
CbTpPes.ItemIndex := 0;
meCpfCg.Text := '';
deDtEmi.Date := Date;
deDtCad.Date := Date;
telCheRe.KeyPreview := true;
CbTpPes.ItemIndex := 0;
deDtLan.Date := date;
vAlterarDadosCliente := false;
btnAlter.Visible := false;
meCpfCg.EditMask := '999.999.999-99;0;';
pnMensa.Visible := false;
LtotalCompra.Caption := '';

LCompraVencer.Caption := '';

LCadastro.Caption := '';

gbCompra.Caption := '';
dm.QPesCompras.Close;
btnFundo.Visible := false;

end;

procedure TTelCheRe.CarregaDadosArquivoParaTela;
begin

With dm.QInserir do
begin
vCheNumer := FieldByName('CheNumer').AsInteger;
fVerificarClienteComChequeSemFundo;
vBanNumer := FieldByName('CheBanco').AsInteger;
ceNrChe.Text := FieldByName('CheNrChe').AsString;
cbTipos.ItemIndex := FieldByName('CheTipos').AsInteger - 1;
ceEmite.Text := fBuscaCliente(vCliNumer);
if FieldByName('CheDtEmi').AsDateTime <> 0 then
deDtEmi.Text := DateToStr(FieldByName('CheDtEmi').AsDateTime)
else
deDtEmi.Text := '';
if FieldByName('CheDtDes').AsDateTime <> 0 then
deDtDes.Text := DateToStr(FieldByName('CheDtDes').AsDateTime)
else
deDtDes.Text := '';
curValor.Text := currToStr(FieldByName('CheValor').AsCurrency);
meObser.Text := FieldByName('CheObser').AsString;
deDtLan.Date := FieldByName('CheDtCad').AsDateTime;
vCliNumer := FieldByName('CheClien').AsInteger;
pCarregaDadosBanco(FieldByName('CheBanco').AsInteger);
end;

```

```

end;

function TTelCheRe.VerificaChequeJaCadastrado : Boolean;
begin

with dm.QInserir do
begin
close;
sql.clear;
sql.add('select * from grCheRe where (CheNrChe = :1) and (CheNrdoc = :2)');
paramByName('1').AsString := ceNrChe.Text;
paramByName('2').AsString := meCpfCg.Text;
open;
if RecordCount > 0 then
Result := true
else
Result :=false;

end;

end;

end;

procedure TTelCheRe.FormKeyPress(Sender: TObject; var Key: Char);
begin
TestaTecla(Self,Key);
end;

procedure TTelCheRe.FormShow(Sender: TObject);
begin
LtotalCompra.Caption := '';

LCompraVencer.Caption := '';

LCadastro.Caption := '';
gbCompra.Caption := '';

gbCompra.Caption := '';

TrazFiguras(Self);
CarregaFiguraBitBtn(tbnExcl2, 'BTNEXCLU');
CarregaFiguraBitBtn(btnTerce, 'pesquisa');
LimparVariaveis;
deDtLan.Date := date;
cbTipos.ItemIndex := 1;

if vUtilizaLimiteCompra then
begin
edLimite.Visible := true;
lLimite.Visible := true;

```

```

end;

try
    dm.QPesquisa.Close;
    dm.QPesquisa.SQL.Clear;
    dm.QPesquisa.SQL.Add('Select * from leConfi where ConNumer = :p1');
    dm.QPesquisa.ParamByName('p1').AsInteger := 18;
    dm.QPesquisa.Open;
    if dm.QPesquisa.FieldName('ConLinha').AsString = '1' then
        ceNrChe.MaxLength := 8;
except
end;

if vUsuario = 1 then
begin
    btnExclu.Visible := false;
    btnExcl2.Visible := false;
    btnFundo.Visible := false;
    gbCompra.Visible := false;
    ceEmite.ReadOnly := true;
    btnAlter.Visible := false;
    btnNovoE.Visible := false;
    deDtNas.ReadOnly := true;
    EdInscr.ReadOnly := true;
    EdIdent.ReadOnly := true;
    ceMunic.ReadOnly := true;
    edUfMun.ReadOnly := true;
    meNrCep.ReadOnly := true;
    ceEnder.ReadOnly := true;
    EdNrRua.ReadOnly := true;
    EdBairr.ReadOnly := true;
    edNrTel.ReadOnly := true;
    edNrFax.ReadOnly := true;
    EdEmail.ReadOnly := true;
    meObser.ReadOnly := true;
    btnTerce.Visible := false;
    btnFundo.visible := false;

end;

TelCheRe.Refresh;

end;

procedure TTelCheRe.ceBancoButtonClick(Sender: TObject);
begin
    IMensagem.Visible := true;
    vPesquisa := 'grBanco';

```

```

TelCheRe.KeyPreview := false;
if not dbgPesqu.Visible then
begin
    dbgPesqu.Height := 168;
    dbgPesqu.Top := ceBanco.Top + ceBanco.Height;
    dbgPesqu.Left := ceBanco.Left;
    dbgPesqu.Width := ceBanco.Width;
    dbgPesqu.DataSource := dm.dsPesquisa;
    dbgPesqu.Parent := pnDados;
    dbgPesqu.Columns.Clear;
    dbgPesqu.Columns.Add.FieldName := 'BanNrAge';
    dbgPesqu.Columns[0].Title.Caption := 'Agência';
    dbgPesqu.Columns.Add.FieldName := 'BanNomes';
    dbgPesqu.Columns[1].Title.Caption := 'Nome';
end;

dbgPesqu.Visible := true;

end;

procedure TTelCheRe.ceBancoEnter(Sender: TObject);
begin

    ceBancoButtonClick(Sender);
    pMontaConsulta;

end;

procedure TTelCheRe.ceBancoExit(Sender: TObject);
var
    vNumChequ,
    vNumBanco : Longint;
    vNrChequ,
    vNomeBanco : String;
begin
    IMensagem.Visible := false;
    if ( ActiveControl = btnSaida ) or ( ActiveControl = btnCance ) ) then
        exit;

    if ceBanco.Text = '' then
        exit;

end;

procedure TTelCheRe.ceBancoKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin

    if key = vk_f4 then
        if ChamarTelaCadastroBanco then
            pMontaConsulta

```

```

else
    Exit; //Chama a tela de cadastro do banco para cadastrar um novo banco
    // se retornar false o banco não foi cadastrado

if not dbgPesqu.Visible then
begin
    dbgPesqu.Height := 168;
    dbgPesqu.Width := ceBanco.Width;
    dbgPesqu.DataSource := dm.dsPesquisa;

    if Key = vk_f2 then
        dbgPesqu.Visible := true;
    end
else
    if Key = vk_f3 then
        dbgPesqu.Visible := false;
    end;

procedure TTelCheRe.btnSaidaClick(Sender: TObject);
begin
    dm.QPesquisa.Close;
    Close;
end;

procedure TTelCheRe.BtnGravaClick(Sender: TObject);
begin

    if ( ActiveControl = BtnSaida) or (ActiveControl = btnCance) ) then
        exit;

    if meCpfCg.Text = '' then
    begin
        ShowMessage('Informe o CPF/CPNJ do cliente');
        meCpfCg.setFocus;
        exit;
    end;

    if not Consiste then
        exit;

    if ceMunic.Text <> '' then
        if vMunNumer = 0 then
            begin
                vMunNumer := InserirDadosCidade(ceMunic.Text,
                    edUfMun.Text,
                    meNrCep.Text);
            end
        if vMunNumer = 0 then
            begin
                MessageDlg('Erro ao incluir dados da cidade.',

```

```

        mtInformation,[mbOk],0);
    end;
end;

if ceEnder.Text <> " then
if vEndNumer = 0 then
begin
    vEndNumer := InserirDadosLogradouro(vMunNumer,
        ceEnder.Text,
        meNrCep.Text,
        edBairr.text);
    if vEndNumer = 0 then
        begin
            MessageDlg('Erro ao incluir dados da cidade.',
                mtInformation,[mbOk],0);
        end;
    end;
end;

if vCliNumer = 0 then
begin
    if not InserirDadosCliente then
        begin
            MessageDlg('Erro ao incluir dados do cliente.',
                mtInformation,[mbOk],0);
        end;
    end;

    pGravarCPFArquivoTexto;
end
else
if vAlterarDadosCliente then
begin
    if not AlterarDadosCliente then
        begin
            MessageDlg('Erro ao alterar dados do cliente.',
                mtInformation,[mbOk],0);
        end;
    end;
end;

if vCheNumer = 0 then
begin
    if VerificaChequeJaCadastrado then
        begin
            MessageDlg('Já existe cheque cadastrado com o banco e número informado.',
                mtInformation,[mbOk],0);
        end;
        exit;
    end;

    if not vAlterarDadosCliente then
        InserirCheque
    end
else

```



```

        AlterarCheque;

        LimparTela(TelCheRe);
        LimparVariaveis;
        ceNrChe.SetFocus;

end;

procedure TTelCheRe.ceNrCheExit(Sender: TObject);
var
    vNumChequ,
    vNumBanco : Longint;
    vNrChequ,
    vAgencia,
    vNomeBanco : String;
begin
    if ( ActiveControl = btnSaida) or (ActiveControl = btnCance) ) then
        exit;

    if vPesquisa = 'grBanco' then
        exit;

    if meCpfCg.Text <> '' then
        if VerificaChequeJaCadastrado then
            CarregaDadosArquivoParaTela;

end;

procedure TTelCheRe.btnCanceClick(Sender: TObject);
begin
    dbgPesqu.Visible := false;
    LimparTela(TelCheRe);
    LimparVariaveis;
    ceNrChe.SetFocus;
    pnMensa.Visible := false;
    gbClien.Color := clBtnFace;
    Timer1.Enabled := false;

    meObser.Color := clWindow;
    meObser.Lines.Clear;
    cbTipos.ItemIndex := 1;

end;

procedure TTelCheRe.btnExcluClick(Sender: TObject);
begin
    if MessageDlg('Confirma a exclusão do cheque?',
        mtInformation,[mbYes,mbNo],0) = mrYes then
        begin
            if DeletarDados('GrCheRe','CheNumer',vCheNumer) then
                begin
                    LimparTela(TelCheRe);

```

```

        LimparVariaveis;
    end
else
    ShowMessage('Erro na exclusão do registro');
end;
ceNrChe.SetFocus;

end;

procedure TTelCheRe.cbTiposExit(Sender: TObject);
begin
    try
        if cbTipos.Text <> '' then
            if cbTipos.Text[1] in ['0'..'9'] then
                cbTipos.ItemIndex := StrToInt(cbTipos.Text) - 1;
            except
            end;
        end;
    end;

end;

procedure TTelCheRe.curValorKeyPress(Sender: TObject; var Key: Char);
begin
    if key = '.' then
        key := ',';
    end;
end;

procedure TTelCheRe.ceEmiteButtonClick(Sender: TObject);
begin
    TelCheRe.KeyPreview := false;
    ceEmiteEnter(Sender);
    if not dbgPesqu.Visible then
        begin
            dbgPesqu.Height := 168;
            dbgPesqu.Top := ceEmite.Top + ceEmite.Height;
            dbgPesqu.Left := ceEmite.Left;
            dbgPesqu.Width := ceEmite.Width;
            dbgPesqu.DataSource := dsPesqu;
            dbgPesqu.Parent := gbClien;
        end;

        dbgPesqu.Visible := true;
        ceEmite.SetFocus;
        pMontaConsulta;
    end;

end;

procedure TTelCheRe.CbTpPesExit(Sender: TObject);
begin
    try
        if cbTpPes.Text[1] in ['0'..'9'] then

```

```

        cbTpPes.ItemIndex := StrToInt(cbTpPes.Text) - 1;
    except
    end;

    HabilitaCampos;

end;

procedure TTelCheRe.ceMunicButtonClick(Sender: TObject);
begin
    MostrarDbgPesqu(Sender);
    dbgPesqu.Height := 133;
    ceMunic.SetFocus;
    dbgPesqu.Parent := gbClien;
    vPesquisa := 'leMunic';

    pMontaConsulta;

end;

procedure TTelCheRe.ceEmiteKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if not dbgPesqu.Visible then
    begin
        if Key = vk_f2 then
        begin
            ceEmiteButtonClick(Sender);
        end;
    end
    else
        if Key = vk_f3 then
            dbgPesqu.Visible := false;

end;

end;

procedure TTelCheRe.ceMunicExit(Sender: TObject);
begin

    if ( (ActiveControl = btnCance) or (ActiveControl = btnSaida) or
        (ActiveControl = dbgPesqu) ) then
        exit;

    dm.QPesquisa.Close;
    dm.QPesquisa.SQL.Clear;
    dm.QPesquisa.Sql.Add('Select * from leMunic ');
    dm.QPesquisa.sql.add('where MunCidad like ' + chr(39) + ceMunic.Text + '%' + chr(39));
    dm.QPesquisa.Open;
    if dm.QPesquisa.RecordCount > 0 then
    begin
        vMunNumer := dm.QPesquisa.FieldName('MunNumer').AsInteger;

```

```

ceMunic.Text := dm.QPesquisa.FieldByName('MunCidad').AsString;
edUfMun.Text := dm.QPesquisa.FieldByName('MunEstad').AsString;
meNrCep.EditMask := '';
meNrCep.Text:= dm.QPesquisa.FieldByName('MunNrCep').AsString;
meNrCep.EditMask := '00000-000;0;_';
vEndNumer := 0;
ceEnder.Text := '';
edBairr.Text := '';
end
else
begin
  if messageDlg('Cidade não está cadastrada. Deseja cadastrar nova cidade?',
    mtCustom,[mbNo,mbYes],0) = mrNo then
    begin
      ceMunicButtonClick(Sender);
    end;
  end;
end;

end;

procedure TTelCheRe.ceMunicKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if dbgPesqu.Visible then
    if ceMunic.Text <> '' then
      dm.QPesquisa.Locate('MunCidad',ceMunic.Text,[loPartialKey]);

end;

end;

procedure TTelCheRe.ceEnderEnter(Sender: TObject);
begin
  if vMunNumer = 0 then
    exit;

  vPesquisa := 'Le' + FormatFloat('00000',vMunNumer);

  MostrarDbgPesqu(Sender);
  dbgPesqu.Height := 96;
  ceEnder.SetFocus;
  dbgPesqu.Parent := gbClien;
  dbgPesqu.Columns.Clear;

  dbgPesqu.Parent := gbClien;

  pMontaConsulta;

end;

procedure TTelCheRe.ceEnderExit(Sender: TObject);
begin
  if ( (ActiveControl = btnSaida) or (ActiveControl = btnCance) ) then

```

```

        exit;

    if ceEnder.Text = '' then
        exit;

    if vMunNumer > 0 then
        vEndNumer := VerificaRuaJaCadastrada(vMunNumer,ceEnder.Text);

end;

procedure TTelCheRe.ceEnderKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin

    if dbgPesqu.Visible then
        if ceEnder.Text <> '' then
            begin
                try
                    dm.QPesquisa.Locate('LogEnder',ceEnder.Text,[loPartialKey]);
                except
                    end;

            end;

end;

end;

procedure TTelCheRe.btnNovoEClick(Sender: TObject);
var
    vNomes : String;
begin
    vNomes := ceEmite.Text;
    LimparDadosCliente;
    btnNovoE.Visible := false;
    ceEmite.Text := vNomes;
    deDtNas.SetFocus;
    meCpfCg.Clear;
    meCpfCg.SetFocus;
end;

procedure TTelCheRe.tbnExcl2Click(Sender: TObject);
begin

    with dm.QConsulta do
        begin
            close;
            Sql.Clear;
            Sql.Add('Select * From grCheRe where CheClien = ' + IntToStr(vCliNumer));
            Open;
            if FieldByName('CheClien').AsInteger = vCliNumer then

```

```

begin
  if MessageDlg('Este cliente possui cheques cadastrados no sistema. ' +
    'Confirma a exclusão?',mtInformation,[mbYes,mbNo],0) = mrYes then
    begin
      if DeletarDados('GrClien','CliNumer',vCliNumer) then
        begin
          LimparTela(TelCheRe);
          LimparVariaveis;
        end
      else
        ShowMessage('Erro na exclusão do registro');
      end;
    end
  else
    if MessageDlg('Confirma a exclusão do cliente?',mtInformation,[mbYes,mbNo],0) = mrYes
    then
      begin
        if DeletarDados('GrClien','CliNumer',vCliNumer) then
          begin
            LimparTela(TelCheRe);
            LimparVariaveis;
          end;
        end;
      end;
    end;
  ceNrChe.SetFocus;

end;

```

```

procedure TTelCheRe.meCpfCgExit(Sender: TObject);
var
  vCliNomes : String;
begin
  if ( ActiveControl = btnSaida) or (ActiveControl = btnCance) ) then
    exit;
  if meCpfCg.Text <> '' then
    if VerificaClinteJaCadastrado then
      begin
        if Not vAlterarDadosCliente then
          begin
            CarregaDadosArquivoClienteParaTela;

            pMontaRelatorioCompras;
            pVerificaSeClienteJaEmitiuChequeSemFundo; {grChSFun}

            if ceNrChe.Text <> '' then
              if VerificaChequeJaCadastrado then
                begin
                  CarregaDadosArquivoParaTela;
                  cbTipos.SetFocus;
                end;
              end;
            end
          end
        end;
      end;
    end;
  end
end

```

```

        else
        begin
            // ShowMessage('Cpf já Cadastrado!');
            // meCpfCg.SetFocus;
        end;
    end
else
    begin
        vCliNomes := ceEmite.Text;
        LimparDadosCliente;
        ceEmite.Text := vCliNomes;
    end;

    if vCliNumer > 0 then
        btnAlter.Visible := True
    else
        begin
            btnAlter.Visible := false;
            vAlterarDadosCliente := false;
        end;

end;

procedure TTelCheRe.btnAlterClick(Sender: TObject);
begin
    vAlterarDadosCliente := true;
end;

procedure TTelCheRe.btnTerceClick(Sender: TObject);
begin

    if FCadTerceiros = nil then
        FCadTerceiros := TFCadTerceiros.Create(Self); {UCadTerceiros}
        FCadTerceiros.ShowModal;

    vTerNumer := fCadTerceiros.vTerNumer;

    meObser.Lines.Clear;
    if FCadTerceiros.ModalResult = mrOk then
        begin

            if fCadTerceiros.cbNegativar.Checked then
                begin
                    meObser.Color := clRed;
                    meObser.Alignment := taCenter;
                    meObser.Lines.Clear;
                    meObser.Lines.Add("");
                    meObser.Lines.Add("");
                    meObser.Lines.Add("");
                    meObser.Lines.Add('TERCEIRO NEGATIVADO');
                end;
            end;
        end;
    end;

```

```

if not(fCadTerceiros.cbNegativar.Checked) and (not fCadTerceiros.vAchouInadimplente) or
(fCadTerceiros.cbPermite.Checked) then
begin
    meObser.Text := '';
    meObser.Color := clWindow;
    meObser.Alignment := taLeftJustify;
    meObser.Lines.Add(FCadTerceiros.edNomes.Text);
    meObser.Lines.Add('Fone: ' + FCadTerceiros.edNrTel.Text +
        ' - Identidade: ' + FCadTerceiros.edIdent.Text);
    meObser.Lines.Add('CPF: ' + FCadTerceiros.meNrCpf.Text);
    meObser.Lines.Add(FCadTerceiros.edEnder.Text + ', ' + FCadTerceiros.edNrRua.Text);
    meObser.Lines.Add(FCadTerceiros.edCidade.Text + ' - ' + FCadTerceiros.EdBairr.Text);
end
else
begin
    if fCadTerceiros.vAchouInadimplente then
    begin
        meObser.Color := clRed;
        meObser.Alignment := taCenter;
        meObser.Lines.Clear;
        meObser.Lines.Add('');
        meObser.Lines.Add('');
        meObser.Lines.Add('');
        meObser.Lines.Add('TERCEIRO INADIMPLENTE');
    end;
end;

end;

FCadTerceiros := nil;
FCadTerceiros.Free;

end;

procedure TTelCheRe.EdIdentExit(Sender: TObject);
begin
    with dm.QPesquisa do
    begin
        Close;
        sql.Clear;
        sql.Add('Select * from GrClien ');
        Sql.Add('Where (CliIdent = :vIdent) or (CliNrDoc = :vNrDoc)');
        ParamByName('vIdent').AsString := edIdent.Text;
        ParamByName('vNrDoc').AsString := meCpfCg.Text;
        Open;
        if RecordCount > 0 then
        begin
            vCliNumer := FieldByName('CliNumer').AsInteger;

```



```

        fVerificarClienteComChequeSemFundo;
    end
    else
        begin
            pnMensa.Visible := false;
        end;
    end;
end;
end;

```

```

procedure TTelCheRe.Timer1Timer(Sender: TObject);
begin
    pnMensa.Visible := not pnMensa.Visible;
end;

```

```

procedure TTelCheRe.edAgenciaExit(Sender: TObject);
begin
    vPesquisa := 'grBanco';
    dbgPesqu.DataSource := dm.dsPesquisa;

```

```

with dm.QConsulta do
begin
    Close;
    Sql.Clear;
    Sql.Add('Select * from grBanco ');
    Sql.Add('Where BanNrAge = :vAgencia');
    ParamByName('vAgencia').AsString := edAgencia.Text;
    Open;
    if RecordCount = 1 then
        begin
            vBanNumer := FieldByName('BanNumer').AsInteger;
            ceBanco.Text := FieldByName('BanNomes').AsString;
            edAgencia.Text := FieldByName('BanNrAge').AsString;
            cbTipos.SetFocus;
        end
    else
        if RecordCount > 1 then
            begin

                ShowMessage('Tem mais de um banco com esse número da agência. ' + #10#13
                    + 'Por favor, realize a consulta de banco.');
            end
        end
    end
end;

dbgPesqu.Height := 168;
dbgPesqu.Top := ceBanco.Top + ceBanco.Height;
dbgPesqu.Left := ceBanco.Left;
dbgPesqu.Width := ceBanco.Width;
dbgPesqu.DataSource := dm.dsPesquisa;

```

```

    dbgPesqu.Parent := pnDados;
    dbgPesqu.Columns.Clear;
    dbgPesqu.Columns.Add.FieldName := 'BanNrAge';
    dbgPesqu.Columns[0].Title.Caption := 'Agência';
    dbgPesqu.Columns.Add.FieldName := 'BanNomes';
    dbgPesqu.Columns[1].Title.Caption := 'Nome';
    dbgPesqu.Visible := true;
    dbgPesqu.SetFocus;

    end;
end;
end;

procedure TTelCheRe.cbTiposEnter(Sender: TObject);
begin
    if ( ActiveControl = btnSaida) or (ActiveControl = btnCance) ) then
        exit;

    IMensagem.Visible := false;
    dbgPesqu.Visible := false;

    if ((vBanNumer = 0) and (ceBanco.Text <> '')) then
        begin
            vBanNumer := PesquisaBancoPorNome(ceBanco.Text);

            if sender = cbTipos then
                if vBanNumer = 0 then
                    end;

        end;

    end;

procedure TTelCheRe.ceMunicKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if Key = vk_f2 then
        begin
            if Not dbgPesqu.Visible then
                begin
                    MostrarDbgPesqu(Sender);
                    dbgPesqu.Height := 133;
                    vPesquisa := 'leMunic';
                    pMontaConsulta;
                end;
        end
    else
        if Key = vk_f3 then
            dbgPesqu.Visible := false;
        end;
end;

```

```

procedure TTelCheRe.ceEnderKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin

  if Key = vk_f2 then
    begin
      if not dbgPesqu.Visible then
        begin
          MostrarDbgPesqu(Sender);
          dbgPesqu.Height := 96;
        end;
      end
    else
      if Key = vk_f3 then
        dbgPesqu.Visible := false;
      end;
end;

```

```

procedure TTelCheRe.ceBancoKeyPress(Sender: TObject; var Key: Char);
begin

```

```

  if not dbgPesqu.Visible then
    begin
      dbgPesqu.Top := 44;
      dbgPesqu.Left := ceBanco.Left;
      dbgPesqu.Height := 168;
      dbgPesqu.Width := ceBanco.Width;
      dbgPesqu.DataSource := dsPesqu;
    end;

```

```

  if key = #0 then
    begin
      if dbgPesqu.Visible then
        dbgPesqu.SetFocus;
      end;

```

```

end;

```

```

procedure TTelCheRe.ceBancoKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if dbgPesqu.Visible then
    if ceBanco.Text <> '' then
      dm.QPesquisa.Locate('BanNomes',ceBanco.Text,[loPartialKey]);

```

```

end;

```

```

procedure TTelCheRe.dbgPesquDbClick(Sender: TObject);
begin
  telCheRe.KeyPreview := false;

```

```

if vPesquisa = 'grBanco' then
begin

    vBanNumer := dm.QPesquisa.FieldByName('BanNumer').AsInteger;
    ceBanco.Text := dm.QPesquisa.FieldByName('BanNomes').AsString;
    edAgencia.Text := dm.QPesquisa.FieldByName('BanNrAge').AsString;
    cbTipos.SetFocus;
    cbTipos.SetFocus;

end
else
if vPesquisa = 'grClien' then
begin
    LimparDadosCliente;
    CarregaDadosArquivoClienteParaTela;
    pMontaRelatorioCompras;
    pVerificaSeClienteJaEmitiuChequeSemFundo; {grChSFun}

    deDtNas.SetFocus;
    btnAlter.Visible := true;
end
else
if vPesquisa = 'Le' + FormatFloat('00000',vMunNumer) then
begin
    vEndNumer := dm.QPesquisa.FieldByName('LogNumer').AsInteger;
    ceEnder.Text := dm.QPesquisa.FieldByName('LogEnder').AsString;
    meNrCep.EditMask := '';
    meNrCep.Text:= dm.QPesquisa.FieldByName('logNrCep').AsString;
    meNrCep.EditMask := '00000-000;0;_';
    edBairr.Text := dm.QPesquisa.FieldByName('LogBairr').AsString;
    edNrRua.SetFocus;
end
else
if vPesquisa = 'leMunic' then
begin
    vMunNumer := dm.QPesquisa.FieldByName('MunNumer').AsInteger;
    ceMunic.Text := dm.QPesquisa.FieldByName('MunCidad').AsString;
    edUfMun.Text := dm.QPesquisa.FieldByName('MunEstat').AsString;
    meNrCep.EditMask := '';
    meNrCep.Text:= dm.QPesquisa.FieldByName('MunNrCep').AsString;
    meNrCep.EditMask := '00000-000;0;_';
    ceEnder.Text := '';
    vEndNumer := 0;
    edBairr.Text := '';
    edUfMun.SetFocus;
end;

dbgPesqu.Visible := false;
telCheRe.KeyPreview := true;

end;

```

```

procedure TTelCheRe.dbgPesquKeyPress(Sender: TObject; var Key: Char);
begin
  if key = #13 then
    dbgPesquDbClick(Sender);

```

```

end;

```

```

procedure TTelCheRe.edAgenciaEnter(Sender: TObject);
begin
  vPesquisa := '';
end;

```

```

procedure TTelCheRe.ceEmiteEnter(Sender: TObject);
begin

```

```

  vPesquisa := 'grClien';
  dbgPesqu.Top := ceEmite.Top + ceEmite.Height;
  dbgPesqu.Left := ceEmite.Left;
  dbgPesqu.Parent := pnDados;
  dbgPesqu.Columns.Clear;
  dbgPesqu.Columns.Add.FieldName := 'CliNumer';
  dbgPesqu.Columns[0].Title.Caption := 'Código';
  dbgPesqu.Columns.Add.FieldName := 'CliNomes';
  dbgPesqu.Columns[1].Title.Caption := 'Nome';
  dbgPesqu.Height := 168;
  dbgPesqu.Top := ceEmite.Top + ceEmite.Height;
  dbgPesqu.Left := ceEmite.Left;
  dbgPesqu.Width := ceEmite.Width;
  dbgPesqu.DataSource := Dm.dsPesquisa;
  dbgPesqu.Parent := gbClien;

```

```

end;

```

```

procedure TTelCheRe.ceEmiteKeyPress(Sender: TObject; var Key: Char);
begin

```

```

  if key = #0 then
    begin
      if dbgPesqu.Visible then
        dbgPesqu.SetFocus;
    end;

```

```

end;

```

```

procedure TTelCheRe.ceEmiteKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);

```

```

begin
  if dbgPesqu.Visible then
    if ceEmite.Text <> '' then
      dm.QPesquisa.Locate('CliNomes',ceEmite.Text,[loPartialKey]);

```

```

end;

procedure TTelCheRe.ceMunicEnter(Sender: TObject);
begin
    dbgPesqu.Columns.Clear;
    dbgPesqu.Columns.AddFieldName := 'MunNumer';
    dbgPesqu.Columns[0].Title.Caption := 'Código';

    dbgPesqu.Columns.AddFieldName := 'MunCidad';
    dbgPesqu.Columns[1].Title.Caption := 'Município';
    dbgPesqu.Parent := gbClien;
    dbgPesqu.Height := 133;

end;

procedure TTelCheRe.ceMunicKeyPress(Sender: TObject; var Key: Char);
begin
    if dbgPesqu.Visible then
        if key = #0 then
            dbgPesqu.SetFocus;

end;

procedure TTelCheRe.ceEnderKeyPress(Sender: TObject; var Key: Char);
begin
    if vMunNumer > 0 then
        if not dbgPesqu.Visible then
            begin
                MostrarDbgPesqu(Sender);
                dbgPesqu.Height := 125;
                if vMunNumer <> 0 then
                    dbgPesqu.Visible := true;
                end;

            if dbgPesqu.Visible then
                if key = #0 then
                    dbgPesqu.SetFocus;

end;

procedure TTelCheRe.EdNrRuaEnter(Sender: TObject);
begin
    dbgPesqu.Visible := false;
end;

procedure TTelCheRe.deDtDesExit(Sender: TObject);
begin

```

```

try
    if StrToDate(deDtDes.Text) < StrToDate(deDtEmi.Text) then
        ShowMessage('Data de Desconto do cheque é menor que a data de emissão! Verifique.');
```

```

except
end;
```

```

end;
```

```

procedure TTelCheRe.dbgRelatorioDrawColumnCell(Sender: TObject;
const Rect: TRect; DataCol: Integer; Column: TColumn;
State: TGridDrawState);
```

```

begin
```

```

    if dm.QPesCompras.FieldName('CheDtDes').AsDateTime > Date then
```

```

    begin
```

```

        dbgRelatorio.Canvas.Font.Color := clRed;
```

```

        dbgRelatorio.Canvas.Brush.Color := clInfoBk;
```

```

    end
```

```

    else
```

```

        begin
```

```

        // dbgRelatorio.Canvas.Font.Color := clWhite;
```

```

        // dbgRelatorio.Canvas.Brush.Color := clTeal;
```

```

        end;
```

```

    if dm.QPesCompras.FieldName('CheFundo').AsBoolean then
```

```

    begin
```

```

        dbgRelatorio.Canvas.Font.Color := clWhite;
```

```

        dbgRelatorio.Canvas.Brush.Color := clRed;
```

```

    end
```

```

    else
```

```

        begin
```

```

        // dbgRelatorio.Canvas.Font.Color := clWhite;
```

```

        // dbgRelatorio.Canvas.Brush.Color := clTeal;
```

```

        end;
```

```

    dbgRelatorio.Canvas.FillRect(Rect);
```

```

    dbgRelatorio.Canvas.TextOut(Rect.left + 2, Rect.top, Column.Field.AsString);
```

```

end;
```

```

procedure TTelCheRe.btnFundoClick(Sender: TObject);
```

```

begin
```

```

    if FCheFundoEmiCliente = nil then
```

```

        FCheFundoEmiCliente := TFcheFundoEmiCliente.Create(Self);
```

```

    FCheFundoEmiCliente.ShowModal;
```

```

    FCheFundoEmiCliente := nil;
```

```

    FCheFundoEmiCliente.Free;
```

end;

```
procedure TTelCheRe.ChequeSemFundo1Click(Sender: TObject);
var
  vNumero : Longint;

begin
  if dm.QPesCompras.FieldName('CheNumer').AsInteger = 0 then
    exit;

  dm.QInserir.Close;
  dm.QInserir.SQL.Clear;
  dm.QInserir.SQL.Add('Update grCheRe set CheFundo = true ');
  dm.QInserir.SQL.Add('Where CheNumer = :p1');
  dm.QInserir.ParamByName('p1').asInteger :=
    dm.QPesCompras.FieldName('CheNumer').AsInteger;

  vNumero := dm.QPesCompras.FieldName('CheNumer').AsInteger;

  dm.QInserir.ExecSQL;

  cadastrarChequeSemFundo('Sem Fundo');

  LttotalCompra.Visible := true;

  pMontaRelatorioCompras;

  dm.QPesCompras.Locate('CheNrChe',vNumero,[LoPartialkey]);
end;
```

```
procedure TTelCheRe.ChequeComFundo1Click(Sender: TObject);
var
  vNumero : Longint;

begin
  if dm.QPesCompras.FieldName('CheNumer').AsInteger = 0 then
    exit;

  dm.QInserir.Close;
  dm.QInserir.SQL.Clear;
  dm.QInserir.SQL.Add('Update grCheRe set CheFundo = false ');
  dm.QInserir.SQL.Add('Where CheNumer = :p1');
  dm.QInserir.ParamByName('p1').asInteger :=
    dm.QPesCompras.FieldName('CheNumer').AsInteger;

  vNumero := dm.QPesCompras.FieldName('CheNumer').AsInteger;

  dm.QInserir.ExecSQL;
```



```

    alterarSituacao(vCliNumer, dm.QPesCompras.FieldName('CheNrChe').AsString, 'A
Descontar');

    pMontaRelatorioCompras;

    dm.QPesCompras.Locate('CheNrChe',vNumero,[LoPartialkey]);

end;

procedure TTelCheRe.ChequeNoBaixado1Click(Sender: TObject);
var
    vNumero : Longint;
begin
    if dm.QPesCompras.FieldName('CheNumer').AsInteger = 0 then
        exit;

    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('UpDate grCheRe set CheBaixa = false ');
    dm.QInserir.SQL.Add('Where CheNumer = :p1');
    dm.QInserir.ParamByName('p1').asInteger :=
        dm.QPesCompras.FieldName('CheNumer').AsInteger;

    vNumero := dm.QPesCompras.FieldName('CheNumer').AsInteger;

    dm.QInserir.ExecSQL;

    alterarSituacao(vCliNumer, dm.QPesCompras.FieldName('CheNrChe').AsString, 'Nao
Compensado');

    pMontaRelatorioCompras;

    dm.QPesCompras.Locate('CheNrChe',vNumero,[LoPartialkey]);

end;

procedure TTelCheRe.BaixarCheque1Click(Sender: TObject);
var
    vNumero : Longint;
begin
    if dm.QPesCompras.FieldName('CheNumer').AsInteger = 0 then
        exit;

    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('UpDate grCheRe set CheBaixa = true ');
    dm.QInserir.SQL.Add('Where CheNumer = :p1');
    dm.QInserir.ParamByName('p1').asInteger :=
        dm.QPesCompras.FieldName('CheNumer').AsInteger;

    vNumero := dm.QPesCompras.FieldName('CheNumer').AsInteger;

```

```

dm.QInserir.ExecSQL;

alterarSituacao(vCliNumer,
dm.QPesCompras.FieldName('CheNrChe').AsString,'Compensado');

pMontaRelatorioCompras;

dm.QPesCompras.Locate('CheNrChe',vNumero,[LoPartialkey]);

end;

procedure TTelCheRe.edLimiteKeyPress(Sender: TObject; var Key: Char);
begin
if key = '.' then
    key := ',';
end;

procedure TTelCheRe.ConfigurarCor1Click(Sender: TObject);
begin
if FCorClienteSemFundo = nil then
    FCorClienteSemFundo := TFCorClienteSemFundo.Create(Self);
    FCorClienteSemFundo.ShowModal;

gbClien.Color := vCorClienteSemFundo;

FCorClienteSemFundo.free;
FCorClienteSemFundo := nil;

end;

procedure TTelCheRe.curValorExit(Sender: TObject);
begin
if vUsuario = 1 then
    BtnGrava.SetFocus;
end;
end.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Cadastro de Terceiros

```

unit UCadTerceiros;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, Buttons, ExtCtrls, Mask, Grids, DBGrids, Db, DBTables, ToolEdit;

type

```
TFCadTerceiros = class(TForm)
  PnBotao: TPanel;
  BtnGrava: TBitBtn;
  btnCance: TBitBtn;
  btnSaida: TBitBtn;
  btnExclu: TBitBtn;
  edNomes: TEdit;
  SpeedButton1: TSpeedButton;
  EdIdent: TEdit;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  edNrTel: TEdit;
  Label4: TLabel;
  meNrCPF: TMaskEdit;
  edEnder: TEdit;
  SpeedButton2: TSpeedButton;
  Label5: TLabel;
```

```

Label10: TLabel;
Label11: TLabel;
Label15: TLabel;
meNrCep: TMaskEdit;
edUfCid: TEdit;
Label13: TLabel;
EdNrRua: TEdit;
EdBairr: TEdit;
Label14: TLabel;
EdEmail: TEdit;
Label21: TLabel;
edCidade: TEdit;
SpeedButton3: TSpeedButton;
dsPesqu: TDataSource;
Label18: TLabel;
DeDtCad: TDateEdit;
LShowMensagemPesquisa: TLabel;
cbPermite: TCheckBox;
btnFundo: TBitBtn;
Timer1: TTimer;
edLimite: TEdit;
ILimite: TLabel;
Panel1: TPanel;
ILimiteCompra: TLabel;
cbNegativar: TCheckBox;
CheckBox1: TCheckBox;
procedure btnSaidaClick(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure edCidadeEnter(Sender: TObject);
procedure edCidadeExit(Sender: TObject);
procedure edCidadeKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure SpeedButton3Click(Sender: TObject);
procedure edEnderEnter(Sender: TObject);
procedure edEnderExit(Sender: TObject);
procedure edEnderKeyPress(Sender: TObject; var Key: Char);
procedure BtnGravaClick(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure btnExcluClick(Sender: TObject);
procedure btnCanceClick(Sender: TObject);
procedure EdIdentExit(Sender: TObject);
procedure meNrCPFExit(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure edNomesKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure edNomesEnter(Sender: TObject);
procedure edNomesExit(Sender: TObject);
procedure edEnderKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure btnFundoClick(Sender: TObject);

```

```

    procedure edLimiteKeyPress(Sender: TObject; var Key: Char);
    procedure cbNegativarClick(Sender: TObject);
private
    { Private declarations }

    vEndNumer,
    vMunNumer : Longint;
    vPesquisou : Boolean;
    procedure pVerificaSeClienteJaEmitiuChequeSemFundo;
    function fVerificarTerceiroComChequeSemFundo : Boolean;
    procedure pCarregarDadosTerceiro;
    procedure pPesquisarTerceiro(Sender : TObject);
    procedure pLimparTela;
    function fBuscaCodigo : Longint;
    procedure pInserirDados;
    procedure pAlterarDados;
    procedure CarregaDadosEndereco;
    procedure CarregaDadosCidade;
    procedure CarregaDadosArquivoTerceiroaraTela(vTabela : TTable);
public
    { Public declarations }
    vTerNumer : Longint;
    vAchouInadimplente : boolean;
end;

var
    FCadTerceiros: TFCadTerceiros;

implementation
Uses
    UProce, UDataM, UPesTerceiros, UPesquisa, UCheFundoEmiCliente;
{$R *.DFM}

procedure TFCadTerceiros.pVerificaSeClienteJaEmitiuChequeSemFundo;
begin
    dm.QPesquisa.Close;
    dm.QPesquisa.SQL.Clear;
    dm.QPesquisa.SQL.Add('Select * from grChSFun ');
    dm.QPesquisa.SQL.Add('where ConTerce = :p1');
    dm.QPesquisa.ParamByName('p1').AsInteger := vTerNumer;
    dm.QPesquisa.Open;

end;

function TFCadTerceiros.fVerificarTerceiroComChequeSemFundo : Boolean;
begin

    dm.QConCheFundoTerce.Close;
    dm.QConCheFundoTerce.SQL.Clear;
    dm.QConCheFundoTerce.SQL.Add('Select * from grChSFun ');

```

```

dm.QConCheFundoTerce.SQL.Add('where ConTerce = :p1');
dm.QConCheFundoTerce.ParamByName('p1').AsInteger := vTerNumer;
dm.QConCheFundoTerce.Open;
if not dm.QConCheFundoTerce.IsEmpty then
    btnFundo.Visible := true
else
    btnFundo.Visible := false;

with dm.QPesquisa do
begin
    Close;
    Sql.Clear;
    Sql.Add('Select * from GrCheRe ');
    Sql.Add('Where CheTerce = :vTerce ');
    Sql.Add('and (CheFundo = true) ');
    ParamByName('vTerce').AsInteger := vTerNumer;
    Open;
    if dm.QPesquisa.FieldByName('CheNumer').AsInteger > 0 then
        begin
            Result := true;
            vPesquisou := true;
            vAchouInadimplente := true;
            Timer1.Enabled := true;

            if cbNegativar.Checked then
                cbNegativar.Font.Color := clRed;
            end
        else
            begin
                Result := false;
                vPesquisou := false;
                vAchouInadimplente := false;
                Timer1.Enabled := false;
                if cbNegativar.Checked then
                    begin
                        cbNegativar.Font.Color := clYellow;
                        cbNegativar.Color := clBackground;

                    end;

                end;

            end;

        end;

    end;

end;

procedure TFCadTerceiros.pCarregarDadosTerceiro;
begin
    vTerNumer := dm.QPesquisa.FieldByName('TerNumer').AsInteger;
    edNomes.Text := dm.QPesquisa.FieldByName('TerNomes').AsString;
    edNrTel.Text := dm.QPesquisa.FieldByName('TerNrTel').AsString;
    meNrCpf.Text := dm.QPesquisa.FieldByName('TerNrCpf').AsString;

```

```

edIdent.Text := dm.QPesquisa.FieldByName('TerIdent').AsString;

if cbNegativar.Visible then
begin
    cbNegativar.Font.Color := clblack;
    cbNegativar.Color := clBtnFace;

    cbNegativar.Checked := dm.QPesquisa.FieldByName('TerSitua').AsBoolean;

end;

edLimite.Text := formatFloat('###,##0.00',dm.QPesquisa.FieldByName('TerLimit').AsFloat);
try
    edNrRua.Text := dm.QPesquisa.FieldByName('TerNrCasa').AsString;
except
    edNrRua.Clear;
end;
edEmail.Text := dm.QPesquisa.FieldByName('TerEmail').AsString;
try
    DeDtCad.Text := DateToStr(dm.QPesquisa.FieldByName('TerDtCad').AsDateTime);
except
    DeDtCad.Clear;
end;
vMunNumer := dm.QPesquisa.FieldByName('TerMunic').AsInteger;
if vMunNumer > 0 then
    CarregaDadosCidade;
vEndNumer := Dm.QPesquisa.FieldByName('TerEnder').AsInteger;
if vEndNumer > 0 then
    CarregaDadosEndereco;

end;

procedure TFCadTerceiros.pPesquisarTerceiro(Sender : TObject);
begin
    with dm.QPesquisa do
    begin
        Close;
        Sql.Clear;
        Sql.Add('Select * from GrTerce ');
        Sql.Add('Where ');
        if sender.ClassType = TEdit then
            begin
                Sql.Add('TerIdent = :vIdent');
                ParamByName('vIdent').AsString := edIdent.Text;
            end
        else
            begin
                Sql.Add('TerNrCpf = :vNrCpf');
                ParamByName('vNrCpf').AsString := meNrCPF.Text;
            end;
        Open;
    end;
end;

```

```

if RecordCount > 0 then
begin
    pCarregarDadosTerceiro;
    if fVerificarTerceiroComChequeSemFundo then
        ShowMessage('Terceiro possui cheque sem fundo...');

    pVerificaSeClienteJaEmitiuChequeSemFundo;
end
else
begin

    Timer1.Enabled := false;
    Panel1.Visible := false;
    vPesquisou := false;
end;

end;
end;

procedure TFCadTerceiros.pLimparTela;
begin
    edLimite.Text := '0,00';
    Timer1.Enabled := false;
    vPesquisou := false;
    vTerNumer := 0;
    vEndNumer := 0;
    vMunNumer := 0;
    edNomes.Clear;
    EdIdent.Clear;
    meNrCPF.Clear;
    edNrTel.Clear;
    EdEmail.Clear;
    edCidade.Clear;
    edUfCid.Clear;
    meNrCep.Clear;
    edEnder.Clear;
    EdNrRua.Text := '0';
    EdBairr.Clear;
    DeDtCad.Date := date;
    edIdent.SetFocus;

    Panel1.Visible := false;
    btnFundo.Visible := false;

    cbNegativar.Checked := false;
    cbNegativar.Font.Color := clblack;
    cbNegativar.Color := clBtnFace;

end;

function TFCadTerceiros.fBuscaCodigo : Longint;
begin

```



```

with dm.QPesquisa do
begin
    Close;
    Sql.Clear;
    sql.Add('Select Max(TerNumer) Maximo From GrTerce');
    Open;
    Result := FieldByName('Maximo').AsInteger + 1;
end;

end;

procedure TFCadTerceiros.pAlterarDados;
begin
    with dm.QInserir do
    begin
        Close;
        sql.Clear;
        sql.Add('UpDate GrTerce Set');
        Sql.Add('TerNomes = :pNomes,TerNrCpf = :pNrCpf,');
        Sql.Add('TerDtCad = :pDtCad,TerIdent = :pIdent,TerNrTel = :pNrTel,');
        Sql.Add('TerMunic = :pMunic,TerEnder = :pEnder,TerNrCasa = :pNrCasa,');
        Sql.Add('TerEmail = :pEmail, TerLimit = :pLimit');
        Sql.Add('Where TerNumer = :pCodigo');
        ParamByName('pCodigo').AsInteger := vTerNumer;
        ParamByName('pNomes').AsString := edNomes.Text;
        ParamByName('pNrCpf').AsString := meNrCpf.Text;
        try
            ParamByName('pDtCad').DataType := ftDate;
            ParamByName('pDtCad').AsDateTime := StrToDate(DeDtCad.Text);
        except
            ParamByName('pDtCad').Clear;
        end;
        ParamByName('pIdent').AsString := edIdent.Text;
        ParamByName('pNrTel').AsString := edNrTel.Text;
        ParamByName('pMunic').AsInteger := vMunNumer;
        ParamByName('pEnder').AsInteger := vEndNumer;
        try
            ParamByName('pNrCasa').AsString := EdNrRua.Text;
        except
            ParamByName('pNrCasa').Clear;
        end;
        ParamByName('pEmail').AsString := edEmail.Text;
        try
            ParamByName('pLimit').AsFloat := StrToFloat(edLimite.Text);
        except
            ParamByName('pLimit').AsFloat := 0;
        end;

    Try
        ExecSql;
    except

```

```

        on E:Exception do
        begin
            ShowMessage(e.message + ' Erro ao gravar dados do Terceiro...');
        end;
    end;

end;

end;

procedure TFCadTerceiros.pInserirDados;
begin
    with dm.QInserir do
    begin
        vTerNumer := fBuscaCodigo;
        Close;
        sql.Clear;
        sql.Add('Insert into GrTerce ');
        Sql.Add('(TerNumer,TerNomes,TerNrCpf,');
        Sql.Add('TerDtCad,TerIdent,TerNrTel,');
        Sql.Add('TerMunic,TerEnder,TerNrCasa,TerEmail,TerLimit)');
        Sql.Add('Values');
        Sql.Add('(:pCodigo,:pNomes,:pNrCpf,:pDtCad,:pIdent,:pNrTel,');
        Sql.Add(':pMunic,:pEnder,:pNrCasa,:pEmail, :pLimit)');
        ParamByName('pCodigo').AsInteger := vTerNumer;
        ParamByName('pNomes').AsString := edNomes.Text;
        ParamByName('pNrCpf').AsString := meNrCpf.Text;
        try
            ParamByName('pDtCad').DataType := ftDate;
            ParamByName('pDtCad').AsDateTime := StrToDate(DeDtCad.Text);
        except
            ParamByName('pDtCad').Clear;
        end;
        ParamByName('pIdent').AsString := edIdent.Text;
        ParamByName('pNrTel').AsString := edNrTel.Text;
        ParamByName('pMunic').AsInteger := vMunNumer;
        ParamByName('pEnder').AsInteger := vEndNumer;
        try
            ParamByName('pNrCasa').AsString := EdNrRua.Text;
        except
            ParamByName('pNrCasa').Clear;
        end;
        ParamByName('pEmail').AsString := edEmail.Text;
        try
            ParamByName('pLimit').AsFloat := StrToFloat(edLimite.Text);
        except
            ParamByName('pLimit').AsFloat := 0;
        end;
    end;

    Try
        ExecSql;
    
```

```

except
  on E:Exception do
  begin
    ShowMessage(e.message + ' . Erro ao gravar dados do Terceiro...');
  end;
end;
end;
end;
end;

```

```

procedure TFCadTerceiros.CarregaDadosEndereco;
begin
  try
    with dm.tbTempo do
    begin
      Close;
      TableName := 'Le' + FormatFloat('00000',vMunNumer);
      Open;
      IndexName := '';
      if FindKey([vEndNumer]) then
      begin
        edEnder.text := FieldByName('LogEnder').AsString;
        meNrCep.Text := FieldByName('LogNrCep').AsString;
        edBairr.Text := FieldByName('LogBairr').AsString;
      end
      else
      begin
        edEnder.text := '';
        meNrCep.Text := '';
        edBairr.Text := '';
      end;
    end;
  end;
except

end;
end;

```

```

procedure TFCadTerceiros.CarregaDadosCidade;
begin
  with dm.TbMunic do
  begin
    IndexName := '';
    if FindKey([vMunNumer]) then
    begin
      edCidade.text := FieldByName('MunCidad').AsString;
      edUfCid.Text := FieldByName('MunEstad').AsString;
      meNrCep.Text := FieldByName('MunNrCep').AsString;
    end
    else
    begin
      edCidade.text := '';
    end;
  end;
end;

```

```

        edUfCid.Text := '';
        meNrCep.Text := '';
    end;

end;

end;

procedure TFCadTerceiros.CarregaDadosArquivoTerceiroaraTela(vTabela : TTable);
begin
    with vTabela do
        begin
            vTerNumer := FieldByName('TerNumer').AsInteger;
            edNomes.Text := FieldByName('TerNomes').AsString;
            meNrCPF.Text := FieldByName('TerNrDoc').AsString;
            EdIdent.Text := FieldByName('TerIdent').AsString;
            edNrTel.Text := FieldByName('TerNrTel').AsString;
            EdEmail.Text := FieldByName('TerEmail').AsString;
            deDtCad.Date := FieldByName('CliDtCad').AsDateTime;
            try
                edNrRua.Text := FieldByName('TerNrCasa').AsString;
            except
                edNrRua.Clear;
            end;
            vMunNumer := FieldByName('CliMunic').AsInteger;
            if vMunNumer > 0 then
                CarregaDadosCidade;
            vEndNumer := FieldByName('CliEnder').AsInteger;
            if vEndNumer > 0 then
                CarregaDadosEndereco;

            end;

        end;

end;

procedure TFCadTerceiros.btnSaidaClick(Sender: TObject);
begin
    ModalResult := mrCancel;
end;

procedure TFCadTerceiros.SpeedButton2Click(Sender: TObject);
begin
    if FPesquisa = nil then
        fPesquisa := TfPesquisa.Create(Self);

    fPesquisa.vTabela := 'leMunic';
    fPesquisa.vCampo := 'MunCidad';
    fPesquisa.vCampo2 := 'MunNumer';

    fPesquisa.ShowModal;

```

```

if fPesquisa.ModalResult = mrOk then
begin
    vMunNumer := dm.QPesquisa.fieldByName('MunNumer').AsInteger;
    edCidade.Text := dm.QPesquisa.fieldByName('MunCidad').asString;
    edUfCid.Text := dm.QPesquisa.fieldByName('MunEstad').asString;
    meNrCep.Text := dm.QPesquisa.fieldByName('MunNrCep').asString;
    edUfCid.SetFocus;
end;

fPesquisa.Free;
fPesquisa := Nil;
end;

procedure TFCadTerceiros.edCidadeEnter(Sender: TObject);
begin
    LShowMensagemPesquisa.Caption := LShowMensagemPesquisa.Caption + ' ' +
Label10.Caption;
    LShowMensagemPesquisa.Top := Label10.Top;
    LShowMensagemPesquisa.Left := Label10.Left + Label10.Width + 10;
    LShowMensagemPesquisa.Visible := true;

end;

procedure TFCadTerceiros.edCidadeExit(Sender: TObject);
begin

    LShowMensagemPesquisa.Visible := false;
    LShowMensagemPesquisa.Caption := 'Clique F2 para abrir pesquisa de';
    if ( ActiveControl = btnCancel) or (ActiveControl = btnSaida) ) then
        exit;

    if ( ActiveControl = btnCancel) or (ActiveControl = btnSaida) ) then
        exit;

    dm.QPesquisa.Close;
    dm.QPesquisa.SQL.Clear;
    dm.QPesquisa.Sql.Add('Select * from leMunic ');
    dm.QPesquisa.sql.add('where MunCidad like ' + chr(39) + edCidade.Text + '%' + chr(39));
    dm.QPesquisa.Open;
    if dm.QPesquisa.RecordCount > 0 then
    begin
        vMunNumer := dm.QPesquisa.FieldByName('MunNumer').AsInteger;
        edCidade.Text := dm.QPesquisa.FieldByName('MunCidad').AsString;
        edUfCid.Text := dm.QPesquisa.FieldByName('MunEstad').AsString;
        meNrCep.EditMask := '';
        meNrCep.Text:= dm.QPesquisa.FieldByName('MunNrCep').AsString;
        meNrCep.EditMask := '00000-000;0;_';
        vEndNumer := 0;
        edEnder.Text := '';
        edBairr.Text := '';
    end;
end;

```

```

end;

procedure TFCadTerceiros.edCidadeKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if key = vk_f2 then
    SpeedButton2Click(Sender);
end;

procedure TFCadTerceiros.SpeedButton3Click(Sender: TObject);
begin
  if FPesquisa = nil then
    fPesquisa := TfPesquisa.Create(Self);

  fPesquisa.vTabela := 'Le' + FormatFloat('00000',vMunNumer);
  fPesquisa.vCampo := 'LogEnder';

  fPesquisa.ShowModal;

  if fPesquisa.ModalResult = mrOk then
    begin
      vEndNumer := dm.QPesquisa.fieldByName('LogNumer').AsInteger;

      edEnder.Text := '';
      CarregaDadosEndereco;
      EdNrRua.SetFocus;
    end;

  fPesquisa.Free;
  fPesquisa := Nil;

  try
    if vMunNumer <> 0 then
      begin
        dm.QConsulta.Close;
        dm.QConsulta.SQL.clear;
        dm.QConsulta.Sql.add('Select * from ' + 'Le' + FormatFloat('00000',vMunNumer));
        dm.QConsulta.Open;
        dsPesqu.DataSet := dm.QConsulta;
      end;

  except
    On E:Exception do
      begin
        MessageDlg(e.message + '. Tabela de logradouro inexistente.',
          mtInformation,[mbOk],0);
      end;
    end;

  end;

end;

procedure TFCadTerceiros.edEnderEnter(Sender: TObject);

```

```

begin
  LShowMensagemPesquisa.Caption := LShowMensagemPesquisa.Caption + ' ' +
Label5.Caption;
  LShowMensagemPesquisa.Top := Label5.Top;
  LShowMensagemPesquisa.Left := Label5.Left + Label5.Width + 10;
  LShowMensagemPesquisa.Visible := true;

end;

procedure TFCadTerceiros.edEnderExit(Sender: TObject);
begin
  LShowMensagemPesquisa.Visible := false;
  LShowMensagemPesquisa.Caption := 'Clique F2 para abrir pesquisa de';
  if ( (ActiveControl = btnSaida) or (ActiveControl = btnCancel) ) then
    exit;
  if edEnder.Text = '' then
    exit;

  if vMunNumer > 0 then
    vEndNumer := VerificaRuaJaCadastrada(vMunNumer,edEnder.Text);

end;

procedure TFCadTerceiros.edEnderKeyPress(Sender: TObject; var Key: Char);
begin
  SpeedButton3Click(Sender);
end;

procedure TFCadTerceiros.BtnGravaClick(Sender: TObject);
begin

  if meNrCpf.Text = '' then
    begin
      ShowMessage('Informe o CPF do cliente');
      meNrCpf.setFocus;
      exit;
    end;

  if edNomes.Text = '' then
    begin
      ShowMessage('Informe o Terceiro');
      exit;
    end;
  if vTerNumer = 0 then
    pInserirDados
  else
    pAlterarDados;

  ModalResult := mrOk;
end;

```

```

procedure TFCadTerceiros.SpeedButton1Click(Sender: TObject);
begin
    Panel1.Visible := false;
    cbPermite.Visible := false;
    cbPermite.Checked := false;
    if FPesTerceiros = nil then
        FPesTerceiros := TFPesTerceiros.Create(Self);

    FPesTerceiros.ShowModal;
    Panel1.Visible := false;
    if FPesTerceiros.ModalResult = MrOk then
        begin
            edNrTel.SetFocus;
            pCarregarDadosTerceiro;

            if fVerificarTerceiroComChequeSemFundo then
                begin
                    cbPermite.Visible := true;
                    ShowMessage('Terceiro possui cheque sem fundo...');
                    Panel1.Visible := true;
                end;

            end;

        end;

    FPesTerceiros.Free;
    FPesterceiros := nil;

end;

procedure TFCadTerceiros.FormShow(Sender: TObject);
begin
    TrazFiguras(Self);
    pLimparTela;
    deDtCad.Date := date;
    vAchouInadimplente := false;
    if vUtilizaLimiteCompraTerceiro then
        begin
            edLimite.Visible := true;
            lLimiteCompra.Visible := true;
        end;

    if vNegativarTerceiro then
        cbNegativar.Visible := true;

end;

procedure TFCadTerceiros.btnExcluClick(Sender: TObject);
begin
    if MessageDlg('Confirma a exclusão do registro? ', mtConfirmation,
        [mbYes,mbNo],0) = mrYes then
        begin

```



```

with dm.QDelete do
begin
  Close;
  Sql.Clear;
  Sql.Add('Delete from GrTerce ');
  sql.Add('Where TerNumer = :vNumer ');
  Params[0].AsInteger := vTerNumer;
  try
    ExecSql;
    pLimparTela;
  Except
    ShowMessage('Erro ao deletar registro!');
  end;
end;
end;
end;

procedure TFCadTerceiros.btnCanceClick(Sender: TObject);
begin
  pLimparTela;
end;

procedure TFCadTerceiros.EdIdentExit(Sender: TObject);
begin
  if (ActiveControl <> btnGrava) and
    (ActiveControl <> btnExclu) and
    (ActiveControl <> btnCance) and
    (ActiveControl <> btnSaida) then
    if edIdent.Text <> '' then
      pPesquisarTerceiro(Sender);

end;

procedure TFCadTerceiros.meNrCPFExit(Sender: TObject);
begin
  if (ActiveControl <> btnGrava) and
    (ActiveControl <> btnExclu) and
    (ActiveControl <> btnCance) and
    (ActiveControl <> btnSaida) then if meNrCpf.Text <> '' then
    pPesquisarTerceiro(Sender);
end;

procedure TFCadTerceiros.Timer1Timer(Sender: TObject);
begin
  Panel1.Visible := not Panel1.Visible;
end;

procedure TFCadTerceiros.FormKeyPress(Sender: TObject; var Key: Char);
begin
  TestaTecla(FCadTerceiros,Key);
end;

```

```

procedure TFCadTerceiros.edNomesKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if key = vk_f2 then
    SpeedButton1Click(Sender);

end;

procedure TFCadTerceiros.edNomesEnter(Sender: TObject);
begin
  LShowMensagemPesquisa.Caption := LShowMensagemPesquisa.Caption + ' ' +
Label1.Caption;
  LShowMensagemPesquisa.Top := Label1.Top;
  LShowMensagemPesquisa.Left := Label1.Left + Label1.Width + 10;
  LShowMensagemPesquisa.Visible := true;

end;

procedure TFCadTerceiros.edNomesExit(Sender: TObject);
begin
  LShowMensagemPesquisa.Visible := false;
  LShowMensagemPesquisa.Caption := 'Clique F2 para abrir pesquisa de';

end;

procedure TFCadTerceiros.edEnderKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if key = vk_f2 then
    SpeedButton3Click(Sender);
end;

procedure TFCadTerceiros.btnFundoClick(Sender: TObject);
begin
  if FCheFundoEmiCliente = nil then
    FCheFundoEmiCliente := TFCheFundoEmiCliente.Create(Self);

  FCheFundoEmiCliente.DBGrid1.DataSource := dm.dsConCheFundoTerce;
  FCheFundoEmiCliente.ShowModal;
  FCheFundoEmiCliente.Free;
  FCheFundoEmiCliente := nil;

end;

procedure TFCadTerceiros.edLimiteKeyPress(Sender: TObject; var Key: Char);
begin
  if key = '.' then
    key := ',';

end;

```

```

procedure TFCadTerceiros.cbNegativarClick(Sender: TObject);
begin
  if vTerNumer > 0 then
  begin
    if cbNegativar.Checked then
    begin
      negativarTerceiro(vTerNumer,true);
      cbNegativar.Font.Color := clred;
      cbNegativar.Caption := 'Terceiro Negativado';
    end
    else
    begin
      negativarTerceiro(vTerNumer,false);
      cbNegativar.Font.Color := clblack;
      cbNegativar.Color := clBtnFace;
      cbNegativar.Caption := 'Negativar Terceiro';

      end;

    end;

  end;
end;
end.

```

XX

Cadastro de Cheques Emitidos

Cadastro de Cheques Emitidos

Cheque Número	Banco		
586522	BRASIL		
Tipo	Data Entrada	Data Desconto	Valor
Chequ Pré	16/04/2013	15/06/2013	R\$ 1.569,40
Nr. Nota	Fornecedor		
5422522	J&J - ATACADISTA DE PRODUTOS ALIMENTÍCIOS S/		
Observação			
COMPRA DE PRODUTOS PARA O SUPERMERCADO			
<input type="checkbox"/> Despesa	Data Cadastramento		
	16/04/2013		

Gravar Cancelar Excluir Fechar

```

unit UCheEm;
interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  CurrEdit, StdCtrls, ToolEdit, Mask, Buttons, ExtCtrls, Grids, DBGrids,

```

Db, DBTables;

type

```
TTelCheEm = class(TForm)
  pnDados: TPanel;
  PnBotao: TPanel;
  BtnGrava: TBitBtn;
  btnCance: TBitBtn;
  btnSaida: TBitBtn;
  btnExclu: TBitBtn;
  Label1: TLabel;
  ceBanco: TComboEdit;
  Label2: TLabel;
  cbTipos: TComboBox;
  Label4: TLabel;
  Label5: TLabel;
  deDtDes: TDateEdit;
  curValor: TCurrencyEdit;
  Label6: TLabel;
  Label3: TLabel;
  Label7: TLabel;
  meObser: TMemo;
  Label8: TLabel;
  deDtEmi: TDateEdit;
  ceNrChe: TComboEdit;
  tbPesqu: TTable;
  dsPesqu: TDataSource;
  tbChequ: TTable;
  Label18: TLabel;
  DeDtCad: TDateEdit;
  cbDespe: TCheckBox;
  cbDefault: TCheckBox;
  edNrNota: TEdit;
  Label9: TLabel;
  edForne: TEdit;
  spPesquForne: TSpeedButton;
  dbgPesqu: TDBGrid;
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure btnSaidaClick(Sender: TObject);
  procedure BtnGravaClick(Sender: TObject);
  procedure ceBancoKeyPress(Sender: TObject; var Key: Char);
  procedure FormShow(Sender: TObject);
  procedure meObserEnter(Sender: TObject);
  procedure meObserExit(Sender: TObject);
  procedure btnExcluClick(Sender: TObject);
  procedure btnCanceClick(Sender: TObject);
  procedure ceBancoButtonClick(Sender: TObject);
  procedure ceBancoEnter(Sender: TObject);
  procedure ceBancoKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
  procedure ceBancoKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
```

```

procedure dbgPesquDbfClick(Sender: TObject);
procedure dbgPesquKeyPress(Sender: TObject; var Key: Char);
procedure ceNrCheExit(Sender: TObject);
procedure ceBancoExit(Sender: TObject);
procedure meObserKeyPress(Sender: TObject; var Key: Char);
procedure cbTiposExit(Sender: TObject);
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure curValorKeyPress(Sender: TObject; var Key: Char);
procedure FormCreate(Sender: TObject);
procedure cbDefaultClick(Sender: TObject);
procedure spPesquForneclick(Sender: TObject);
procedure edFornecKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
private
  { Private declarations }
  vForNumer,
  vCheNumer,
  vBanNumer : Longint;

  function ChamarTelaCadastroBanco : Boolean;
  function VerificaChequeJaCadastrado : Boolean;
  function PesquisaCheques : boolean;
  procedure CarregaDadosArquivoParaTela;
  function Consiste : Boolean;
  procedure LimparVariaveis;
  procedure InserirCheque;
  procedure AlterarCheque;
public
  { Public declarations }
  vTipoEmissao : Byte;
end;

var
  TelCheEm: TTelCheEm;

implementation

uses UDataM, UProce, UBanco, UPesquisa;

{$R *.DFM}

function TTelCheEm.ChamarTelaCadastroBanco : Boolean;
begin
  if TelBanco = nil then
    TelBanco := TTelBanco.Create(Self);

  TelBanco.ceNomes.Text := ceBanco.Text;
  TelBanco.BtnSaida.Enabled := false;
  TelBanco.btnExclu.Enabled := false;
  vMensagem := 'Incluir Cheques';
  TelBanco.ShowModal;

```

```

vBanNumer := vRetorno;
if vRetorno > 0 then
    Result := true
else
    Result := false;
vRetorno := 0;
end;

```

```

function TTelCheEm.VerificaChequeJaCadastrado : Boolean;
begin
    tbChequ.IndexName := 'NrChequBanco';
    if tbChequ.FindKey([ceNrChe.Text,vBanNumer]) then
        Result := true
    else
        Result := false;
end;

```

```

function TTelCheEm.PesquisaCheques : boolean;
begin
    Result := false;
    with Dm.QPesquisa do
        begin
            Close;
            Sql.Clear;
            Sql.Add('Select * From GrChequ ');
            Sql.Add('Where CheNCheq = :vChequ ');
            Sql.Add('and CheBanco = :vBanco ');
            ParamByName('vChequ').AsString:= ceNrChe.Text;
            ParamByName('vBanco').AsInteger := vBanNumer;
            Open;
            if FieldByName('CheNumer').AsInteger <> 0 then
                begin
                    vCheNumer      := FieldByName('CheNumer').AsInteger;
                    cbTipos.ItemIndex := FieldByName('CheTipos').AsInteger;
                    if FieldByName('CheDtEmi').AsDateTime <> 0 then
                        deDtEmi.Text  := DateToStr(FieldByName('CheDtEmi').AsDateTime)
                    else
                        deDtEmi.Text  := '';
                    if FieldByName('CheDtDes').AsDateTime <> 0 then
                        deDtDes.Text  := DateToStr(FieldByName('CheDtDes').AsDateTime)
                    else
                        deDtDes.Text  := '';
                    curValor.Text    := currToStr(FieldByName('CheValor').AsCurrency);
                    meObser.Text     := FieldByName('CheObser').AsString;
                    Result := true;
                end;
            end;
        end;
end;

```

```

procedure TTelCheEm.CarregaDadosArquivoParaTela;

```

```

begin
  With TbChequ do
    begin
      vCheNumer      := FieldByName('CheNumer').AsInteger;
      vBanNumer      := FieldByName('CheBanco').AsInteger;
      ceNrChe.Text    := FieldByName('CheNrChe').AsString;
      cbTipos.ItemIndex := FieldByName('CheTipos').AsInteger - 1;
      edForne.Text    := FieldByName('CheClien').AsString;
      if FieldByName('CheDtEmi').AsDateTime <> 0 then
        deDtEmi.Text  := DateToStr(FieldByName('CheDtEmi').AsDateTime)
      else
        deDtEmi.Text  := '';
      if FieldByName('CheDtDes').AsDateTime <> 0 then
        deDtDes.Text  := DateToStr(FieldByName('CheDtDes').AsDateTime)
      else
        deDtDes.Text  := '';
      curValor.Text   := currToStr(FieldByName('CheValor').AsCurrency);
      meObser.Text    := FieldByName('CheObser').AsString;
      DeDtCad.Date    := FieldByName('CheDtCad').AsDateTime;
      ceBanco.Text    :=
        BuscaNome('GrBanco','BanNomes','BanNumer',FieldByName('CheBanco').AsInteger);
      if FieldByName('CheDespe').AsString = 'Despesa' then
        cbDespe.Checked := true
      else
        cbDespe.Checked := false;
    end;
  end;

  function TTelCheEm.Consiste : Boolean;
  begin
    Result := true;
    if ceNrChe.Text = '' then
      begin
        MessageDlg('Informe o número do cheque.', mtInformation,[mbOk],0);
        ceNrChe.SetFocus;
        Result := false;
        exit;
      end;

    if vBanNumer = 0 then
      begin
        MessageDlg('Banco não foi Selecionado.', mtInformation,[mbOk],0);
        ceBanco.SetFocus;
        Result := false;
        exit;
      end;

    if cbTipos.ItemIndex = -1 then
      begin
        MessageDlg('Informe o tipo do cheque (Bom/Pré).', mtInformation,[mbOk],0);
        cbTipos.SetFocus;
        Result := false;
      end;
    end;
  end;

```

```

        exit;
    end;

    if ( (deDtEmi.Text = ' / / ') or (deDtEmi.Text = ' / / ') ) then
    begin
        MessageDlg('Informe a data de emissão do cheque.', mtInformation,[mbOk],0);
        deDtEmi.SetFocus;
        Result := false;
        exit;
    end;

    if ( (deDtDes.Text = ' / / ') or (deDtDes.Text = ' / / ') ) then
    begin
        MessageDlg('Informe a data de desconto do cheque.', mtInformation,[mbOk],0);
        deDtDes.SetFocus;
        Result := false;
        exit;
    end;

    if curValor.Text = '0' then
    begin
        MessageDlg('Valor do cheque não foi informado.', mtInformation,[mbOk],0);
        curValor.SetFocus;
        Result := false;
        exit;
    end;

end;

procedure TTelCheEm.LimparVariaveis;
begin
    vCheNumer := 0;
    deDtEmi.Date := Date;
    deDtCad.Date := Date;
    telCheEm.KeyPreview := true;

end;

procedure TTelCheEm.InserirCheque;
var
    vNumero : Longint;
begin
    with dm.QInserir do
    begin
        vNumero := BuscaNumero('GrCheEm','CheNumer');
        Close;
        Sql.Clear;
        Sql.Add('Insert Into GrCheEm ');
        Sql.Add('(CheNumer, CheNrChe, CheBanco, Chetipos, CheDtEmi, ');

```



```

Sql.Add(' CheDtDes, CheClien, CheValor, CheObser, CheBaixa, CheDtCad, ');
Sql.Add(' CheFundo, CheDespe, CheDtMod,CheNrNot,CheForne ');
Sql.Add('Values (:vNumer, :vNrChe, :vBanco, :vTipos, :vDtEmi, :vDtDes, ');
Sql.Add('      :vClien, :vValor, :vObser, :vBaixa, :vDtCad, :vFundo, ');
Sql.Add('      :vDespe, :vDtMod,:vNrNot,:vForne)');
ParamByName('vNumer').AsInteger := vNumero;
ParamByName('vNrChe').AsString := ceNrChe.Text;
ParamByName('vBanco').AsInteger := vBanNumber;
ParamByName('vTipos').AsInteger := cbTipos.ItemIndex + 1;
ParamByName('vDtEmi').AsDate := StrToDate(deDtEmi.Text);
ParamByName('vDtDes').AsDate := StrToDate(deDtDes.Text);
ParamByName('vClien').AsString := edForne.Text;
ParamByName('vValor').AsCurrency := curValor.value;
ParamByName('vObser').AsString := meObser.Text;
ParamByName('vDtCad').AsDateTime := Date;
ParamByName('vBaixa').AsBoolean := false;
ParamByName('vFundo').AsBoolean := false;
ParamByName('vDtMod').AsDateTime := Date;
if cbDespe.Checked then
    ParamByName('vDespe').AsString := 'Despesa'
else
    ParamByName('vDespe').AsString := 'Não';

ParamByName('vNrNot').AsString := edNrNota.Text;
ParamByName('vForne').AsInteger := vForNumer;

ExecSql;
Close;

end;

end;

procedure TTelCheEm.AlterarCheque;
begin
    with dm.QInserir do
        begin
            Close;
            Sql.Clear;
            Sql.Add('UpDate GrCheEm ');
            Sql.Add('Set CheNrChe = :vNrChe, CheBanco = :vBanco, Chetipos = :vTipos, '+
                'CheDtEmi = :vDtEmi, CheDtDes = :vDtDes, CheClien = :vClien, '+
                'CheValor = :vValor, CheObser = :vObser, CheDtCad = :vDtCad, '+
                'CheDespe = :vDespe, CheDtMod = :vDtMod');

            Sql.Add('Where CheNumer = :vNumer ');
            ParamByName('vNrChe').AsString := ceNrChe.Text;
            ParamByName('vBanco').AsInteger := vBanNumber;
            ParamByName('vTipos').AsInteger := cbTipos.ItemIndex + 1;
            ParamByName('vDtEmi').AsDate := StrToDate(deDtEmi.Text);
            ParamByName('vDtDes').AsDate := StrToDate(deDtDes.Text);
            ParamByName('vClien').AsString := edForne.Text;

```

```

ParamByName('vValor').AsCurrency := curValor.value;
ParamByName('vObser').AsString := meObser.Text;
ParamByName('vDtCad').AsDateTime := deDtCad.Date;
ParamByName('vNumer').AsInteger := vCheNumer;
if cbDespe.Checked then
    ParamByName('vDespe').AsString := 'Despesa'
else
    ParamByName('vDespe').AsString := 'Não';

ParamByName('vDtMod').AsDateTime := Date;
ExecSql;
Close;

end;

end;

procedure TTelCheEm.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    tbChequ.Close;
    Action := caFree;
    TelCheEm := nil;
end;

procedure TTelCheEm.btnSaidaClick(Sender: TObject);
begin
    Close;
end;

procedure TTelCheEm.BtnGravaClick(Sender: TObject);
var
    vStr : String;
begin
    if ( ActiveControl = BtnSaida) or (ActiveControl = btnCance) ) then
        exit;

    if not Consiste then
        exit;

    if vCheNumer = 0 then
        begin
            if VerificaChequeJaCadastrado then
                begin
                    MessageDlg('Já existe cheque cadastrado com o banco e número informado.',
                        mtInformation,[mbOk],0);
                    exit;
                end;

            InserirCheque
        end
    end;

```

```

else
    AlterarCheque;

vStr := ceBanco.Text;
LimparTela(TelCheEm);
LimparVariaveis;
ceBanco.Text := vStr;
ceNrChe.SetFocus;

end;

procedure TTelCheEm.ceBancoKeyPress(Sender: TObject; var Key: Char);
begin
    if not dbgPesqu.Visible then
        begin
            dbgPesqu.Top := 44;
            dbgPesqu.Left := ceBanco.Left;
            dbgPesqu.Height := 168;
            dbgPesqu.Width := ceBanco.Width;
            dbgPesqu.DataSource := dsPesqu;
        end;

    if key = #0 then
        begin
            if dbgPesqu.Visible then
                dbgPesqu.SetFocus;
            if ceBanco.Text <> '' then
                tbPesqu.Locate('BanNomes',ceBanco.Text,[loPartialKey]);
        end;

end;

end;

procedure TTelCheEm.FormShow(Sender: TObject);
begin

    TrazFiguras(TelCheEm);
    LimparTela(TelCheEm);
    LimparVariaveis;
    tbChequ.Open;
    dm.QPesquisa.close;
    dm.QPesquisa.SQL.Clear;
    dm.QPesquisa.SQL.Add('Select * from GrBanco where BanDefau <> 0');
    dm.QPesquisa.Open;
    ceBanco.Text := dm.QPesquisa.fieldByName('BanNomes').AsString;
    vBanNumer := dm.QPesquisa.fieldByName('BanNumer').AsInteger;
    dm.QPesquisa.Close;

end;

```

```

procedure TTelCheEm.meObserEnter(Sender: TObject);
begin
    TelCheEm.KeyPreview := false;
end;

procedure TTelCheEm.meObserExit(Sender: TObject);
begin
    TelCheEm.KeyPreview := true;
end;

procedure TTelCheEm.btnExcluClick(Sender: TObject);
begin
    if MessageDlg('Confirma a exclusão do cheque?',
        mtInformation,[mbYes,mbNo],0) = mrYes then
    begin
        if DeletarDados('GrCheEm','CheNumer',vCheNumer) then
        begin
            LimparTela(TelCheEm);
            LimparVariaveis;
        end
    else
        ShowMessage('Erro na exclusão do registro');
    end;
    ceNrChe.SetFocus;

end;

procedure TTelCheEm.btnCanceClick(Sender: TObject);
begin
    LimparTela(TelCheEm);
    LimparVariaveis;
    ceNrChe.SetFocus;
end;

procedure TTelCheEm.ceBancoButtonClick(Sender: TObject);
begin
    TelCheEm.KeyPreview := false;
    if not dbgPesqu.Visible then
    begin
        tbPesqu.Close;
        tbPesqu.TableName := 'GrBanco';
        tbPesqu.IndexName := 'Nomes';
        tbPesqu.Open;
        dbgPesqu.Height := 168;
        dbgPesqu.Top := ceBanco.Top + ceBanco.Height;
        dbgPesqu.Left := ceBanco.Left;
        dbgPesqu.Width := ceBanco.Width;
        dbgPesqu.DataSource := dsPesqu;
    end;

    dbgPesqu.Visible := true;

```

end;

procedure TTelCheEm.ceBancoEnter(Sender: TObject);

begin

```
    dbgPesqu.Top := ceBanco.Top + ceBanco.Height;
    dbgPesqu.Left := ceBanco.Left;
    tbPesqu.close;
    tbPesqu.TableName := 'GrBanco';
    tbPesqu.IndexName := 'Nomes';
    tbPesqu.Open;
    dbgPesqu.Parent := pnDados;
    dbgPesqu.Columns.Clear;
    dbgPesqu.Columns.Add.FieldName := 'BanNrAge';
    dbgPesqu.Columns[0].Title.Caption := 'Agência';
    dbgPesqu.Columns.Add.FieldName := 'BanNomes';
    dbgPesqu.Columns[1].Title.Caption := 'Nome';
```

end;

procedure TTelCheEm.ceBancoKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);

begin

```
    if not dbgPesqu.Visible then
    begin
        dbgPesqu.Height := 168;
        dbgPesqu.Width := ceBanco.Width;
        dbgPesqu.DataSource := dsPesqu;
```

```
        if Key = vk_f2 then
            dbgPesqu.Visible := true;
```

```
    end
```

```
    else
```

```
        if Key = vk_f3 then
            dbgPesqu.Visible := false;
```

end;

procedure TTelCheEm.ceBancoKeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);

begin

```
    if dbgPesqu.Visible then
        if ceBanco.Text <> '' then
            tbPesqu.Locate('BanNomes',ceBanco.Text,[loPartialKey]);
```

end;

procedure TTelCheEm.dbgPesquDbClick(Sender: TObject);

begin

```
    vBanNumer := tbPesqu.FieldByName('BanNumer').AsInteger;
    ceBanco.Text := tbPesqu.FieldByName('BanNomes').AsString;
    dbgPesqu.Visible := false;
```

```

telCheEm.KeyPreview := true;
ceBancoExit(Sender);
cbTipos.SetFocus;
end;

procedure TTelCheEm.dbgPesquKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #13 then
        dbgPesquDbClick(Sender);

end;

procedure TTelCheEm.ceNrCheExit(Sender: TObject);
var
    vNumChequ,
    vNumBanco : Longint;
    vNrChequ,
    vNomeBanco : String;

begin
    if ( ActiveControl = btnSaida) or (ActiveControl = btnCance) ) then
        exit;

    if vBanNumer > 0 then
        if VerificaChequeJaCadastrado then
            CarregaDadosArquivoParaTela
        else
            begin

                vNrChequ := ceNrChe.text;
                vNumBanco := vBanNumer;
                vNomeBanco := ceBanco.Text;
                LimparTela(TelCheEm);
                LimparVariaveis;
                ceNrChe.text := vNrChequ;
                vBanNumer := vNumBanco;
                ceBanco.Text := vNomeBanco;
                ceBanco.SetFocus;
                ceBanco.SetFocus;
            end;
end;

procedure TTelCheEm.ceBancoExit(Sender: TObject);
var
    vNumChequ,
    vNumBanco : Longint;
    vNrChequ,
    vNomeBanco : String;

begin
    if ( ActiveControl = btnSaida) or (ActiveControl = btnCance) ) then
        exit;

```

```

if ceBanco.Text = "" then
    exit;

if ((vBanNumer = 0) and (ceBanco.Text <> "")) then
    vBanNumer := PesquisaBancoPorNome(ceBanco.Text);

if dbgPesqu.Visible then
    dbgPesqu.SetFocus
else
    if vBanNumer = 0 then
        if not ChamarTelaCadastroBanco then
            Exit; //Chama a tela de cadastro do banco para cadastrar um novo banco
            // se retornar false o banco não foi cadastrado

if ceNrChe.Text <> "" then
    if VerificaChequeJaCadastrado then
        CarregaDadosArquivoParaTela
    else
        begin
            vNrChequ := ceNrChe.text;
            vNumBanco := vBanNumer;
            vNomeBanco := ceBanco.Text;
            LimparTela(TelCheEm);
            LimparVariaveis;
            ceNrChe.text := vNrChequ;
            vBanNumer := vNumBanco;
            ceBanco.Text := vNomeBanco;

        end;
end;

procedure TTelCheEm.meObserKeyPress(Sender: TObject; var Key: Char);
begin
    Key := UpCase(Key);
end;

procedure TTelCheEm.cbTiposExit(Sender: TObject);
begin
    try
        if cbTipos.Text <> "" then
            if cbTipos.Text[1] in ['0'..'9'] then
                cbTipos.ItemIndex := StrToInt(cbTipos.Text) - 1;
    except
        end;
end;

procedure TTelCheEm.FormKeyPress(Sender: TObject; var Key: Char);
begin
    TestaTecla(Self,key);

```

```

end;

procedure TTelCheEm.curValorKeyPress(Sender: TObject; var Key: Char);
begin
    if key = '.' then
        key := ',';
end;

procedure TTelCheEm.FormCreate(Sender: TObject);
begin
    try
        dm.QPesquisa.close;
        dm.QPesquisa.SQL.Clear;
        dm.QPesquisa.SQL.Add('Select BanDefau from GrBanco');
        dm.QPesquisa.Open;
        dm.QPesquisa.Close;
    except
        dm.QPesquisa.close;
        dm.QPesquisa.SQL.Clear;
        dm.QPesquisa.SQL.Add('Alter table GrBanco add BanDefau Integer');
        dm.QPesquisa.ExecSql;
    end;
end;

procedure TTelCheEm.cbDefaultClick(Sender: TObject);
begin
    if vBanNumer = 0 then
        exit;

    try

        dm.QPesquisa.close;
        dm.QPesquisa.SQL.Clear;
        dm.QPesquisa.SQL.Add('UpDate GrBanco set BanDefau = :p1');
        dm.QPesquisa.ParamByName('p1').AsInteger := 0;
        dm.QPesquisa.ExecSql;

        dm.QPesquisa.close;
        dm.QPesquisa.SQL.Clear;
        dm.QPesquisa.SQL.Add('UpDate GrBanco set BanDefau = :p1');
        dm.QPesquisa.SQL.Add('where BanNumer = :p2');
        dm.QPesquisa.ParamByName('p1').AsInteger := vBanNumer;
        dm.QPesquisa.ParamByName('p2').AsInteger := vBanNumer;
        dm.QPesquisa.ExecSql;
    except

    end;

end;

procedure TTelCheEm.spPesquForneClick(Sender: TObject);
begin

```



```

if fPesquisa = nil then
    fPesquisa := TfPesquisa.Create(Self);

fPesquisa.vTabela := 'GrForne';
fPesquisa.vCampo := 'ForNomes';

fPesquisa.ShowModal;

if fPesquisa.ModalResult = mrOk then
begin
    edForne.Text := dm.QPesquisa.fieldByName('ForNomes').AsString;
    vForNumer := dm.QPesquisa.fieldByName('ForNumer').AsInteger;

end;

fPesquisa.Free;
fPesquisa := Nil;

end;

procedure TTelCheEm.edForneKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if Key = vk_f2 then
        spPesquForneClick(Sender);
end;

end.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Controle de Baixa de Cheques

```

Baixa de Cheques

Período
 Data Início: 01/01/2013 Data Fim: 25/04/2013

Cheques: ☒ Recebidos ☐ Emitidos

Selecionar por: ☒ Data Desconto ☐ Data Emissão

Mostrar: ☒ Todos ☐ Cheques não Baixados

Nº do Cheque: Cedente:

Ordenar por: ☐ Data Desconto ☒ Cliente
☐ Data Emissão ☐ Número Cheque

☒ Descontar ☒ Sem Fundo ☒ Retornar ☒ Cancelar ☒ Fechar

Baixado	Número	Dt Desconto	Dt Emissão	Cedente	Agência	Banco	Valor	Sem Fi
Sim	1	26/2/2013	26/2/2013	ADRIANA DA SILVA	696	BRASIL	R\$ 253,25	Não
Sim	12	26/3/2013	26/2/2013	ADRIANA DA SILVA	696	BRASIL	R\$ 100,00	Sim
Não	12522	16/4/2013	16/3/2013	ADRIANA DA SILVA	696	BRASIL	R\$ 526,80	Não
Sim	3	26/2/2013	26/2/2013	ADRIANA DA SILVA	696	BRASIL	R\$ 300,00	Não

unit UBCheq;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, ExtCtrls, Grids, DBGrids, Buttons, Db, DBTables, Mask, ToolEdit;

type

```
TTelBCheq = class(TForm)
  Panel7: TPanel;
  Panel8: TPanel;
  Label15: TLabel;
  rgChequ: TRadioGroup;
  DSBaixa: TDataSource;
  Label2: TLabel;
  ceNomes: TComboEdit;
  QCheRe: TQuery;
  ceNCheq: TComboEdit;
  QCheEm: TQuery;
  GroupBox1: TGroupBox;
  DeDtIni: TDateEdit;
  Label18: TLabel;
  deDtFim: TDateEdit;
  Label1: TLabel;
  QCheReCheNumer: TIntegerField;
  QCheReCheBanco: TIntegerField;
  QCheReCheBaixa: TBooleanField;
  QCheReCheDtEmi: TDateField;
```

```

QCheReCheDtDes: TDateField;
QCheReCheValor: TCurrencyField;
QCheReCheNrChe: TStringField;
QCheReCheObser: TStringField;
QCheReBanNomes: TStringField;
QCheEmCheNumer: TIntegerField;
QCheEmCheBanco: TIntegerField;
QCheEmCheBaixa: TBooleanField;
QCheEmCheDtEmi: TDateField;
QCheEmCheDtDes: TDateField;
QCheEmCheValor: TCurrencyField;
QCheEmCheNrChe: TStringField;
QCheEmCheObser: TStringField;
QCheEmBanNomes: TStringField;
DBGrid2: TDBGrid;
QCheEmCheClien: TStringField;
QCheReCheCeden: TStringField;
QCheEmCheFundo: TBooleanField;
QCheReCheFundo: TBooleanField;
QCheEmBanNrAge: TStringField;
QCheReBanNrAge: TStringField;
rgSelec: TRadioGroup;
rgOrdem: TRadioGroup;
rgMostr: TRadioGroup;
pnMensa: TPanel;
QCheReCheClien: TIntegerField;
QCheReCheTerce: TIntegerField;
Panel10: TPanel;
btnSaida: TBitBtn;
BtnGrava: TBitBtn;
btncance: TBitBtn;
BtnFundo: TBitBtn;
btnTodos: TBitBtn;
BtnRetor: TBitBtn;
QCheReCheNrDoc: TStringField;
procedure rgChequClick(Sender: TObject);
procedure btnSaidaClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure ceNomesExit(Sender: TObject);
procedure btncanceClick(Sender: TObject);
procedure cbContrExit(Sender: TObject);
procedure BtnGravaClick(Sender: TObject);
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure ceNCheqKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure FormShow(Sender: TObject);
procedure BtnFundoClick(Sender: TObject);
procedure ceNomesKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure DBGrid1CellClick(Column: TColumn);
procedure DBGrid2CellClick(Column: TColumn);
procedure ceNCheqKeyPress(Sender: TObject; var Key: Char);

```

```

procedure DBGrid2Enter(Sender: TObject);
procedure deDtFimAcceptDate(Sender: TObject; var Date: TDateTime;
    var Action: Boolean);
procedure btnTodosClick(Sender: TObject);
procedure deDtFimEnter(Sender: TObject);
procedure deDtFimExit(Sender: TObject);
procedure BtnRetorClick(Sender: TObject);
procedure ceNomesKeyPress(Sender: TObject; var Key: Char);
private
    { Private declarations }
    ArqTexto2,
    ArqTexto : Text;
    vCheNumer,
    vBanNumer : Longint;
    vTipoEmissao : Byte;
    vTemChequeSemFundo,
    vJaPesquisou : Boolean;
    procedure pGravarArquivoTxt(vDocumento : String; vNome : String;
        vFundo : Boolean);
    procedure statusBar;
    procedure HabilitaDesabilitaBotoes;
    procedure RetornarChequeRecebidos;
    procedure RetornarChequeEmitidos;
    procedure AlterarChequeEmitidosSemFundo;
    procedure AlterarChequeRecebidosSemFundo;
    procedure BaixaChequeRecebidos;
    procedure BaixaChequeEmitidos;
    procedure MontaSqlChequeEmitidos;
    procedure MontaSqlChequRecebidos;
    procedure LimparTela;

public
    { Public declarations }
end;

var
    TelBCheq: TTelBCheq;

implementation

uses UDataM,
    UProce, UEntra;
{$R *.DFM}

procedure TTelBCheq.pGravarArquivoTxt(vDocumento : String; vNome : String;
    vFundo : Boolean);
var
    vTexto,
    vStr,
    vStr2 : String;
    i : word;
begin

```

```

AssignFile(ArqTexto,vDiretorio + '\Cadastro.txt');
AssignFile(ArqTexto2,vDiretorio + '\Cadastro2.txt');
{$I-}
    Reset(ArqTexto);
    rewrite(ArqTexto2);
{$I+}
    // Gravar arquivo txt;

While not eof(arqTexto) do
begin
    ReadLn(ArqTexto,vStr);
    vTexto := vStr;
    i := pos(';',vStr);
    vStr2 := copy(vStr,1,i-1);
    try
        if vStr2 = vDocumento then
            begin
                vStr2 := vStr2 + ';';
                delete(vStr,1,i);
                i := pos(';',vStr);
                vStr2 := vStr2 + copy(vStr,1,i);
                delete(vStr,1,i + 2);
                if vFundo then
                    vStr2 :=vStr2 + '1' + ';';
                else
                    vStr2 := vStr2 + '0' + ';';

                vStr2 := vStr2 + vStr;
                writeln(ArqTexto2,vStr2);
            end
        else
            WriteLn(ArqTexto2,vStr);
    except
    end;
end;

CloseFile(ArqTexto);
CloseFile(ArqTexto2);
try
    erase(arqTexto);
    rename(arqTexto2, vDiretorio + '\Cadastro.txt');

except
end;

end;

procedure TTelBCheq.statusBar;
begin
    FEntra.ProgressBar1.Position := FEntra.ProgressBar1.Position + 1;
    if FEntra.ProgressBar1.Position = 100 then

```

```
FEntra.ProgressBar1.Position := 0;  
end;
```

```
procedure TTelBCheq.HabilitaDesabilitaBotoes;
```

```
begin
```

```
    BtnGrava.Enabled := not BtnGrava.Enabled;  
    btnTodos.Enabled := not btnTodos.Enabled;  
    BtnFundo.Enabled := not BtnFundo.Enabled;  
    BtnRetor.Enabled := not BtnRetor.Enabled;  
    btncance.Enabled := not btncance.Enabled;  
    btnSaida.Enabled := not btnSaida.Enabled;
```

```
end;
```

```
procedure TTelBCheq.RetornarChequeEmitidos;
```

```
begin
```

```
    with Dm.QInserir do
```

```
    begin
```

```
        Close;  
        Sql.Clear;  
        Sql.Add('UpDate GrCheEm ');  
        Sql.Add('Set CheBaixa = :vBaixa, CheFundo = :vFundo,CheDtMod = :vDatas ');  
        Sql.Add('Where CheNumer = :vNumer ');  
        ParamByName('vBaixa').AsBoolean := false;  
        ParamByName('vFundo').AsBoolean := false;  
        ParamByName('vDatas').AsDateTime := Date;  
        ParamByName('vNumer').AsInteger := QCheEm.FieldByName('CheNumer').AsInteger;  
        ExecSql;
```

```
    end;
```

```
end;
```

```
procedure TTelBCheq.RetornarChequeRecebidos;
```

```
begin
```

```
    with Dm.QInserir do
```

```
    begin
```

```
        Close;  
        Sql.Clear;  
        Sql.Add('UpDate GrCheRe ');  
        Sql.Add('Set CheBaixa = :vBaixa, CheFundo = :vFundo,CheDtMod = :vDatas ');  
        Sql.Add('Where CheNumer = :vNumer ');  
        ParamByName('vBaixa').AsBoolean := false;  
        ParamByName('vFundo').AsBoolean := false;  
        ParamByName('vDatas').AsDateTime := Date;  
        ParamByName('vNumer').AsInteger := QCheRe.FieldByName('CheNumer').AsInteger;  
        ExecSql;
```

```
    end;
```

```
end;
```

```
procedure TTelBCheq.AlterarChequeEmitidosSemFundo;
```

```
begin
```

```
    with Dm.QInserir do
```

```

begin
    Close;
    Sql.Clear;
    Sql.Add('UpDate GrCheEm ');
    Sql.Add('Set CheBaixa = :vBaixa, CheFundo = :vFundo,CheDtMod = :vDatas ');
    Sql.Add('Where CheNumer = :vNumer ');
    ParamByName('vBaixa').AsBoolean := false;
    ParamByName('vFundo').AsBoolean := true;
    ParamByName('vDatas').AsDateTime := Date;
    ParamByName('vNumer').AsInteger := QCheEm.FieldByName('CheNumer').AsInteger;
    ExecSql;

    end;
end;

procedure TTelBCheq.AlterarChequeRecebidosSemFundo;
var
    vStr : String;
    vPosicao : integer;

begin
    with Dm.QInserir do
        begin
            Close;
            Sql.Clear;
            Sql.Add('UpDate GrCheRe ');
            Sql.Add('Set CheBaixa = :vBaixa, CheFundo = :vFundo,CheDtMod = :vDatas ');
            Sql.Add('Where CheNumer = :vNumer ');
            ParamByName('vBaixa').AsBoolean := false;
            ParamByName('vFundo').AsBoolean := true; // não tem fundo
            ParamByName('vDatas').AsDateTime := Date;
            ParamByName('vNumer').AsInteger := QCheRe.FieldByName('CheNumer').AsInteger;
            ExecSql;

            Close;
            Sql.Clear;
            Sql.Add('Select * from grChSFun ');
            Sql.Add('Where ConClien = :p1');
            Sql.Add('And ConNrChe = :p2');
            ParamByName('p1').AsInteger := QCheRe.FieldByName('CheClien').AsInteger;
            ParamByName('p2').AsString := QCheRe.FieldByName('CheNrChe').AsString;
            Open;
            if FieldByName('ConCodig').AsInteger = 0 then
                begin
                    Close;
                    Sql.Clear;
                    Sql.Add('Insert into grChSFun ');
                    Sql.Add('(ConCodig,ConDatas,ConClien,ConChequ,
ConValor,ConTerce,ConNrChe,conSitua)');
                    Sql.Add('Values');
                    Sql.Add('(:p1,:p2,:p3,:p4,:p5,:p6,:p7,pSitua)');
                end
            end
        end
    end
end

```

```

ParamByName('p1').AsInteger := BuscaNumero('GrChSFun','ConCodig');
ParamByName('p2').AsDateTime := Date;
ParamByName('p3').AsInteger := QCheRe.FieldByName('CheClien').AsInteger;
ParamByName('p4').AsInteger := QCheRe.FieldByName('CheNumer').AsInteger;
ParamByName('p5').AsCurrency := QCheRe.FieldByName('CheValor').AsCurrency;
ParamByName('p6').AsInteger := QCheRe.FieldByName('CheTerce').AsInteger;
dm.QInserir.ParamByName('p7').AsString := QCheRe.FieldByName('CheNrChe').AsString;
dm.QInserir.ParamByName('p7').AsString := 'Sem Fundo';
ExecSql;
end
else
    alterarSituacao(QCheRe.FieldByName('CheClien').AsInteger,
QCheRe.FieldByName('CheNrChe').AsString, 'Sem Fundo');

```

```

vJaPesquisou := false;
vTemChequeSemFundo := true;
end;

```

```
end;
```

```

procedure TTelBCheq.BaixaChequeRecebidos;
begin
    with Dm.QInserir do
        begin
            Close;
            Sql.Clear;
            Sql.Add('UpDate GrCheRe ');
            Sql.Add('Set CheBaixa = :vBaixa,CheFundo = :vFundo,CheDtMod = :vDatas ');
            Sql.Add('Where CheNumer = :vNumer ');
            ParamByName('vBaixa').AsBoolean := true;
            ParamByName('vFundo').AsBoolean := false;
            ParamByName('vDatas').AsDateTime := Date;
            ParamByName('vNumer').AsInteger := QCheRe.FieldByName('CheNumer').AsInteger;
            ExecSql;

            end;
        end;

```

```

procedure TTelBCheq.BaixaChequeEmitidos;
begin
    with Dm.QInserir do
        begin
            Close;
            Sql.Clear;
            Sql.Add('UpDate GrCheEm ');
            Sql.Add('Set CheBaixa = :vBaixa,CheFundo = :vFundo,CheDtMod = :vDatas ');
            Sql.Add('Where CheNumer = :vNumer ');
            ParamByName('vBaixa').AsBoolean := true;
            ParamByName('vFundo').AsBoolean := false;
            ParamByName('vDatas').AsDateTime := Date;
            ParamByName('vNumer').AsInteger := QCheEm.FieldByName('CheNumer').AsInteger;

```



```

    ExecSql;

    end;
end;

procedure TTelBCheq.MontaSqlChequeEmitidos;
begin

    with QCheEm do
        begin
            Close;
            Sql.Clear;
            Sql.Add('Select  C.CheNumer,C.CheClien,C.CheBanco,C.CheBaixa,C.CheDtEmi, C.CheDtDes,
C.CheValor, ');
            Sql.Add('C.CheNrChe, C.CheObser,C.CheFundo, B.BanNomes,B.BanNrAge ');
            Sql.Add('From GrCheEm C, GrBanco B');
            Sql.Add('Where C.CheBanco = B.BanNumer ');
            if DeDtIni.Date <> 0 then
                begin
                    Sql.Add('And C.CheDtEmi >= :vDtEmi ');
                    ParamByName('vDtEmi').AsDateTime := DeDtIni.Date;
                end;
            if deDtFim.Date <> 0 then
                begin
                    Sql.Add('And C.CheDtEmi <= :vDtFim ');
                    ParamByName('vDtFim').AsDateTime := deDtFim.Date;
                end;

            if rgOrdem.ItemIndex = 0 then
                Sql.Add('Order By C.CheDtDes ');
            else
                if rgOrdem.ItemIndex = 1 then
                    Sql.Add('Order By C.CheDtEmi ');
                else
                    Sql.Add('Order By C.CheClien ');
                end;

            Open;
        end;

    end;

    procedure TTelBCheq.MontaSqlChequRecebidos;
    begin
        if vJaPesquisou then
            exit;

        with QCheRe do
            begin
                Close;
                Sql.Clear;
            end;
        end;
    end;
end;

```

```

    Sql.Add('Select
C.CheNumer,C.CheClien,C.CheNrDoc,C.CheBanco,C.CheBaixa,C.CheDtEmi,');
    Sql.Add('C.CheDtDes, C.CheValor,C.CheTerce, ');
    Sql.Add('C.CheNrChe, C.CheCeden,C.CheObser,C.CheFundo, B.BanNomes,B.BanNrAge ');
    Sql.Add('From GrCheRe C, GrBanco B');
    Sql.Add('Where C.CheBanco = B.BanNumer ');
    if rgMostr.ItemIndex = 1 then
        Sql.Add('and C.CheBaixa = false ');

    if DeDtIni.Date <> 0 then
        begin
            if rgSelec.ItemIndex = 1 then
                Sql.Add('And C.CheDtEmi >= :vDtEmi ');
            else
                Sql.Add('And C.CheDtDes >= :vDtEmi ');
                ParamByName('vDtEmi').AsDateTime := DeDtIni.Date;
            end;
        if deDtFim.Date <> 0 then
            begin
                if rgSelec.ItemIndex = 1 then
                    Sql.Add('And C.CheDtEmi <= :vDtFim ');
                else
                    Sql.Add('And C.CheDtDes <= :vDtFim ');
                    ParamByName('vDtFim').AsDateTime := deDtFim.Date;
                end;

            if rgOrdem.ItemIndex = 0 then
                Sql.Add('Order By C.CheDtDes ');
            else
                if rgOrdem.ItemIndex = 1 then
                    Sql.Add('Order By C.CheDtEmi ');
                else
                    if rgOrdem.ItemIndex = 3 then
                        Sql.Add('Order By C.CheNrChe,C.CheCeden ');
                    else
                        Sql.Add('Order By C.CheCeden,C.CheNrChe ');

            Open;
            vJaPesquisou := true;
        end;
    end;

procedure TTelBCheq.LimparTela;
begin
    ceNomes.Text := '';
    ceNCheq.Text := '';
    QCheRe.Close;
    QCheRe.SQL.Clear;
    QCheEm.Close;
    QCheEm.SQL.Clear;

end;

```

```

procedure TTelBCheq.rgChequClick(Sender: TObject);
begin
    vJaPesquisou := false;
    pnMensa.Visible := true;
    TelBCheq.Refresh;
    QCheEm.Close;
    QCheRe.Close;
    DBGrid2.Refresh;

    if rgChequ.ItemIndex = 1 then
    begin
        DBGrid2.Columns[2].FieldName := 'CheClien';
        DBGrid2.Columns[2].Title.Caption := 'Emitido Para';
        rgOrdem.Items[2] := 'Cedente';
        MontaSqlChequeEmitidos;
        DSBaixa.DataSet := QCheEm;
        QCheRe.Close;
        ceNomes.Text := '';
        DBGrid2.Columns[2].Width := 170;
        DBGrid2.Columns[4].Width := 0;
    end
    else
    begin
        rgOrdem.Items[2] := 'Cliente';
        DBGrid2.Columns[4].FieldName := 'CheCeden';
        dbGrid2.Columns[2].Title.Caption := 'Dt Desconto';
        DBGrid2.Columns[2].FieldName := 'CheDtDes';
        dbGrid2.Columns[2].Width := 70;
        MontaSqlChequRecebidos;
        ceNomes.ReadOnly := false;
        DSBaixa.DataSet := QCheRe;
        QCheEm.Close;
        ceNomes.Text := '';
        dbGrid2.Columns[4].Width := 170;
    end;
    dbGrid2.Columns[0].Width := 40;
    dbGrid2.Columns[1].Width := 63;
    dbGrid2.Columns[3].Width := 70;
    dbGrid2.Columns[5].Width := 50;
    dbGrid2.Columns[6].Width := 150;
    pnMensa.Visible := false;
end;

procedure TTelBCheq.btnSaidaClick(Sender: TObject);
begin
    Close;
end;

procedure TTelBCheq.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    if vTemChequeSemFundo then

```

```

begin
    FEntra := TFEntra.Create(Self);
    FEntra.Label1.visible := false;
    FEntra.Label2.visible := false;
    FEntra.Label3.Caption := 'Aguarde!';
    FEntra.show;
    FEntra.Refresh;

    FEntra.Label4.Visible := true;
    FEntra.Refresh;
    AssignFile(ArqTexto,vDiretorio + '\Cadastro.txt');
    {$I-}
    Reset(ArqTexto);
    {$I+}
    // Gravar arquivo txt;

    try
        closeFile(ArqTexto);
    except
    end;

    FEntra.Close;

end;

TelBCheq := nil;
Action := CaFree;
end;

procedure TTelBCheq.ceNomesExit(Sender: TObject);
begin
    if ( (ActiveControl = btnSaida) or (ActiveControl = btnCance) ) then
        exit;

end;

procedure TTelBCheq.btncanceClick(Sender: TObject);
begin
    LimparTela;
end;

procedure TTelBCheq.cbContrExit(Sender: TObject);
begin
    if rgChequ.ItemIndex <> -1 then
        MontaSqlChequRecebidos;

end;

procedure TTelBCheq.BtnGravaClick(Sender: TObject);

```

```

var
  vNumero : Longint;
begin
  if vUtilizaTXTNosCaixa then
    vTemChequeSemFundo := true;

  HabilitaDesabilitaBotoes;

  vJaPesquisou := false;
  BtnGrava.Enabled := false;
  if rgChequ.ItemIndex = 1 then
    begin
      BaixaChequeEmitidos;
      MontaSqlChequeEmitidos;

    end
  else
    begin

      vNumero := QCheRe.FieldName('CheNumer').AsInteger;
      BaixaChequeRecebidos;
      alterarSituacao(QCheRe.FieldName('CheClien').AsInteger,
        QCheRe.FieldName('CheNrChe').AsString, 'Compensado');
      MontaSqlChequeRecebidos;

      QCheRe.Locate('CheNumer',IntToStr(vNumero),[LoCaseInsensitive]);
      pGravarArquivoTxt(QCheRe.FieldName('CheNrDoc').AsString,
        QCheRe.FieldName('CheCeden').AsString,
        QCheRe.FieldName('CheFundo').AsBoolean);

    end;

  try
    if rgChequ.ItemIndex = 0 then
      QCheRe.Locate('CheNrChe',ceNCheq.Text,[LoCaseInsensitive])
    else
      QCheEm.Locate('CheNrChe',ceNCheq.Text,[LoCaseInsensitive]);
    except
    end;

    HabilitaDesabilitaBotoes;
    BtnGrava.Enabled := true;
  end;

  procedure TTelBCheq.FormKeyPress(Sender: TObject; var Key: Char);
  begin
    TestaTecla(Self,Key);
  end;

  procedure TTelBCheq.ceNCheqKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
  begin

```

```

try
  if rgChequ.ItemIndex = 0 then
    begin
      if ceNCheq.Text <> '' then
        QCheRe.Locate('CheNrChe',ceNCheq.Text,[LoPartialKey]);
      end
    end
  else
    begin
      if ceNCheq.Text <> '' then
        QCheEm.Locate('CheNrChe',ceNCheq.Text,[LoPartialKey]);
      end;
    end;
  except
  end;

```

```

end;

```

```

procedure TTelBCheq.FormShow(Sender: TObject);
begin
  TrazFiguras(TelBCheq);
  CarregaFiguraBitBtn(btnFundo,'BTNGRAVA');
  CarregaFiguraBitBtn(btnTodos,'BTNGRAVA');
  CarregaFiguraBitBtn(btnRetor,'BTNGRAVA');
  DeDtIni.SetFocus;

  dm.QInserir.Close;
  dm.QInserir.SQL.Clear;
  dm.QInserir.SQL.Add('Select * from leConfi');
  dm.QInserir.SQL.Add('Where ConNumer = :p1');
  dm.QInserir.ParamByName('p1').AsInteger := 9;
  dm.QInserir.Open;
  if dm.QInserir.FieldByName('ConLinha').AsInteger = 1 then
    btnTodos.Visible := false
  else
    btnTodos.Visible := true;
end;

```

```

procedure TTelBCheq.BtnFundoClick(Sender: TObject);
var
  vNumero : Longint;

begin
  if vUtilizaTXTNosCaixa then
    vTemChequeSemFundo := true;

  HabilitaDesabilitaBotoes;
  if rgChequ.ItemIndex = 1 then
    begin
      if QCheEm.FieldByName('CheBaixa').AsBoolean then
        exit;
    end;

```

```

vNumero := QCheEm.FieldName('CheNumer').AsInteger;
AlterarChequeEmitidosSemFundo;

MontaSqlChequeEmitidos;
QCheEm.Locate('CheNumer',IntToStr(vNumero),[LoCaseInsensitive]);

end
else
begin
vNumero := QCheRe.FieldName('CheNumer').AsInteger;
AlterarChequeRecebidosSemFundo;
MontaSqlChequeRecebidos;
QCheRe.Locate('CheNumer',IntToStr(vNumero),[LoCaseInsensitive]);

pGravarArquivoTxt(QCheRe.FieldName('CheNrDoc').AsString,
    QCheRe.FieldName('CheCeden').AsString,
    QCheRe.FieldName('CheFundo').AsBoolean);

end;
HabilitaDesabilitaBotoes;
BtnFundo.Enabled := true;

end;

procedure TTelBCheq.ceNomesKeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
try
if rgChequ.ItemIndex = 0 then
begin
if ceNomes.Text <> '' then
QCheRe.Locate('CheCeden',ceNomes.Text,[loPartialKey]);
end
else
begin
if ceNomes.Text <> '' then
QCheEm.Locate('CheClien',ceNomes.Text,[loPartialKey]);
end;
except
end;
end;

procedure TTelBCheq.DBGrid1CellClick(Column: TColumn);
begin
try
if rgChequ.ItemIndex = 0 then
begin
ceNomes.Text := QCheRe.FieldName('CheClien').AsString;
ceNCheq.Text := QCheRe.FieldName('CheNrChe').AsString;
end

```

```

else
    ceNomes.Text := QCheEm.FieldByName('CheCeden').AsString;
    ceNChEq.Text := QCheEm.FieldByName('CheNrChe').AsString;
except
end;
end;

procedure TTelBCheq.DBGrid2CellClick(Column: TColumn);
begin
    try
        if rgChequ.ItemIndex = 0 then
            begin
                ceNChEq.Text := QCheRe.FieldByName('CheNrChe').AsString;
                ceNomes.Text := QCheRe.FieldByName('CheCeden').AsString;
            end
        else
            begin
                ceNChEq.Text := QCheEm.FieldByName('CheNrChe').AsString;
                ceNomes.Text := QCheEm.FieldByName('CheClien').AsString;
            end;
        except
        end;

    end;

procedure TTelBCheq.ceNChEqKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #0 then
        DBGrid2.SetFocus;

    end;

procedure TTelBCheq.DBGrid2Enter(Sender: TObject);
begin
    try
        if rgChequ.ItemIndex = 0 then
            begin
                ceNChEq.Text := QCheRe.FieldByName('CheNrChe').AsString;
                ceNomes.Text := QCheRe.FieldByName('CheCeden').AsString;
            end
        else
            begin
                ceNChEq.Text := QCheEm.FieldByName('CheNrChe').AsString;
                ceNomes.Text := QCheEm.FieldByName('CheClien').AsString;
            end;
        except
        end;

    end;

procedure TTelBCheq.deDtFimAcceptDate(Sender: TObject; var Date: TDateTime;
var Action: Boolean);

```



```

begin
  rgChequ.ItemIndex := -1;
end;

procedure TTelBCheq.btnTodosClick(Sender: TObject);
begin
  if vUtilizaTXTNosCaixa then
    vTemChequeSemFundo := true;

  if messageDlg('Confirma a baixa de todos os cheques selecionados?',
    mtInformation,[mbYes,mbNo],0) = mrYes then
  begin
    btnTodos.Enabled := false;

    if rgChequ.ItemIndex = 1 then
    begin
      with QCheEm do
      begin
        while not Eof do
        begin
          if not QCheEm.FieldName('CheBaixa').AsBoolean then
            BaixaChequeEmitidos;

          Next;
        end;
        MontaSqlChequeEmitidos;

      end;
    end
    else
    begin
      with QCheRe do
      begin
        while not Eof do
        begin
          if Not QCheRe.FieldName('CheBaixa').AsBoolean then
            BaixaChequeRecebidos;

          Next;
        end;
        MontaSqlChequeRecebidos;

      end;
    end;
  end;
  btnTodos.Enabled := true;

end;

procedure TTelBCheq.deDtFimEnter(Sender: TObject);
begin
  rgChequ.ItemIndex := -1;

```

```

end;

procedure TTelBCheq.deDtFimExit(Sender: TObject);
begin

    if rgChequ.ItemIndex = -1 then
    begin
        rgChequ.ItemIndex := 0;
        if not vJaPesquisou then
            rgChequClick(Sender);
        end;
    end;

end;

procedure TTelBCheq.BtnRetorClick(Sender: TObject);
var
    vNomes : String;
    vNumero : Longint;
begin
    if vUtilizaTXTNosCaixa then
        vTemChequeSemFundo := true;

    HabilitaDesabilitaBotoes;
    BtnGrava.Enabled := false;
    vJaPesquisou := false;
    if rgChequ.ItemIndex = 1 then
    begin
        vNomes := QCheEm.FieldName('CheClien').AsString;
        RetornarChequeEmitidos;
        MontaSqlChequeEmitidos;
    end
    else
    begin
        vNomes := QCheRe.FieldName('CheCeden').AsString;
        vNumero := QCheRe.FieldName('CheNumer').AsInteger;
        RetornarChequeRecebidos;
        alterarSituacao(QCheRe.FieldName('CheClien').AsInteger,
            QCheRe.FieldName('CheNrChe').AsString, 'A Descontar');
        MontaSqlChequeRecebidos;

        QCheRe.Locate('CheNumer',IntToStr(vNumero),[LoCaseInsensitive]);
        pGravarArquivoTxt(QCheRe.FieldName('CheNrDoc').AsString,
            QCheRe.FieldName('CheCeden').AsString,
            QCheRe.FieldName('CheFundo').AsBoolean);

    end;

    HabilitaDesabilitaBotoes;

end;

procedure TTelBCheq.ceNomesKeyPress(Sender: TObject; var Key: Char);

```

```
begin
  if key = #0 then
    dbGrid2.SetFocus;
  end;
```

```
end.
```

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Controle de Cheques Repassados Para Terceiros

```
unit UTerce;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
ExtCtrls, StdCtrls, Mask, ToolEdit, Buttons, Db, Grids, DBGrids, DBTables,
CurrEdit;
```

```
type
```

```
TTelTerce = class(TForm)
  pnDados: TPanel;
  CeNomes: TComboEdit;
  Label1: TLabel;
  grCheRe: TGroupBox;
  grCheTe: TGroupBox;
  pnBotao: TPanel;
  btnSaida: TBitBtn;
  pnAdici: TPanel;
  btnGrava: TBitBtn;
  btnCance: TBitBtn;
```

```

dsCheRe: TDataSource;
dbgCheRe: TDBGrid;
rgOrdem: TRadioGroup;
edNrChe: TEdit;
dbgCheTe: TDBGrid;
dsCheTe: TDataSource;
QCheTe: TQuery;
LPesqu: TLabel;
tbPesqu: TTable;
dsPesqu: TDataSource;
dbgPesqu: TDBGrid;
QCheTeCheNrChe: TStringField;
QCheTeCheCeden: TStringField;
QCheTeCheValor: TCurrencyField;
QCheTeCheDtEmi: TDateField;
QCheTeCheDtDes: TDateField;
QCheTeBanNomes: TStringField;
QCheTeCheChequ: TIntegerField;
Panel1: TPanel;
rgTipos: TRadioGroup;
curValor: TCurrencyEdit;
Label2: TLabel;
btnGrav2: TBitBtn;
meObser: TMemo;
Label3: TLabel;
deDatas: TDateEdit;
Label4: TLabel;
cbDespe: TCheckBox;
QCheRe: TQuery;
QCheReCheNrChe: TStringField;
QCheReCheCeden: TStringField;
QCheReCheDtDes: TDateField;
QCheReBanNomes: TStringField;
QCheReCheValor: TCurrencyField;
QCheReCheFundo: TBooleanField;
QCheReCheObser: TStringField;
QCheReCheNumer: TIntegerField;
cbMostra: TCheckBox;
QCheReCheRepas: TBooleanField;
BtnCorrige: TBitBtn;
cbTodosCheques: TCheckBox;
BitBtn1: TBitBtn;
deDtDesp: TDateEdit;
Label5: TLabel;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormShow(Sender: TObject);
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure btnSaidaClick(Sender: TObject);
procedure edNrCheKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure edNrCheKeyPress(Sender: TObject; var Key: Char);
procedure rgOrdemClick(Sender: TObject);

```

```

procedure btnGravaClick(Sender: TObject);
procedure dbgCheReCellClick(Column: TColumn);
procedure dbgCheReEnter(Sender: TObject);
procedure CeNomesButtonClick(Sender: TObject);
procedure CeNomesEnter(Sender: TObject);
procedure CeNomesKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure CeNomesKeyPress(Sender: TObject; var Key: Char);
procedure CeNomesKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure dbgPesquDblClick(Sender: TObject);
procedure dbgPesquKeyPress(Sender: TObject; var Key: Char);
procedure CeNomesExit(Sender: TObject);
procedure btnCanceClick(Sender: TObject);
procedure btnGrav2Click(Sender: TObject);
procedure curValorKeyPress(Sender: TObject; var Key: Char);
procedure dbgCheReDrawColumnCell(Sender: TObject; const Rect: TRect;
  DataCol: Integer; Column: TColumn; State: TGridDrawState);
procedure BtnCorrigeClick(Sender: TObject);
procedure cbMostraClick(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
private
  { Private declarations }
  vForNumer : Longint;
  procedure IncluirFornecedor;
  procedure MostrarDbgPesqu(Sender : TObject);
  procedure MarcarChequeRepassadoParaTerceiro(vNumero : Longint;vValor :
Boolean;vObser : String;
                                vFornecedor : Longint);
  procedure MontaQueryChequesRepassadosParaTerceiros;
  procedure MontaQueryChequesRecebidos;
public
  { Public declarations }
end;

var
  TelTerce: TTelTerce;

implementation
Uses
  UProce, UDataM, UCorrigeTroco, UForne;
{$R *.DFM}

procedure TTelTerce.IncluirFornecedor;
begin
  with dm.QInserir do
  begin
    vForNumer := BuscaNumero('GrForne','ForNumer');
    Close;
    Sql.Clear;
    Sql.Add('Insert Into GrForne ');
    Sql.Add('(ForNumer,ForNomes)');
  end;
end;

```

```

    Sql.Add(' Values (:vNumer,:vNomes)');
    ParamByName('vNumer').AsInteger := vForNumer;
    ParamByName('vNomes').AsString := ceNomes.Text;
    ExecSql;
end;
end;

procedure TTelTerce.MostrarDbgPesqu(Sender : TObject);
begin

    TelTerce.KeyPreview := false;
    dbgPesqu.Top := (Sender as tComboEdit).Top + (Sender as tComboEdit).Height;
    dbgPesqu.Left := (Sender as tComboEdit).Left;
    dbgPesqu.Width := (Sender as tComboEdit).Width;
    dbgPesqu.DataSource := dsPesqu;
    dbgPesqu.Visible := true;
end;

procedure TTelTerce.MarcasChequeRepassadoParaTerceiro(vNumero : Longint;vValor :
Boolean;vObser : String;
                                vFornecedor : Longint);
var
    i : word;
begin
    i := pos(';',vObser);

    if i = 0 then
        vObser := 'Repassado para ' + CeNomes.Text + ' em ' +
            DateToStr(Date) + ';; ' +
            QCheRe.FieldByName('CheObser').AsString

    else
        delete(vObser,1,i + 2);

    try
        with dm.QInserir do
            begin
                Close;
                Sql.Clear;
                Sql.Add('UpDate GrCheRe ');
                Sql.Add('Set CheRepas = :vValor,CheDtMod = :vDatas, CheObser = :vObser,CheTerce =
:vTerce ');
                Sql.Add('Where CheNumer = :vNumer ');
                ParamByName('vNumer').AsInteger := vNumero;
                ParamByName('vValor').AsBoolean := vValor;
                ParamByName('vDatas').AsDateTime := Date;
                ParamByName('vObser').AsString := vObser;
                ParamByName('vTerce').AsInteger := vFornecedor;//vForNumer;
                ExecSql;
            end;
        except

```

```

On E:Exception do
    MessageDlg('Erro ao marcar cheque repassado para terceiros.',
        mtInformation,[mbOk],0);
end;
end;

procedure TTelTerce.MontaQueryChequesRepassadosParaTerceiros;
begin
    with QCheTe do
        begin
            Close;
            Sql.Clear;
            Sql.Add('Select T.CheChequ,C.CheNrChe, Cli.CliNomes CheCeden,
C.CheValor,C.CheDtEmi,');
            Sql.Add('    C.CheDtDes, B.BanNomes ');
            Sql.Add(' From GrCheTe T, GrCheRe C,GrClien cli, GrBanco B ');
            Sql.Add(' Where (T.CheForne = :vForne) ');
            Sql.Add(' And  (T.CheChequ = C.CheNumer) ');
            Sql.Add(' And  (C.CheClien = cli.CliNumer) ');
            Sql.Add(' And  (C.CheBanco = B.BanNumer) ');
            ParamByName('vForne').AsInteger := vForNumer;
            if rgOrdem.ItemIndex = 1 then
                Sql.Add(' Order by C.CheDtDes ');
            Open;

            end;
        end;

procedure TTelTerce.MontaQueryChequesRecebidos;
begin
    with QCheRe do
        begin
            Close;
            Sql.Clear;
            Sql.Add('Select C.CheNumer, C.CheNrChe,Cli.CliNomes cheCeden,
C.CheValor,C.CheDtEmi,');
            Sql.Add('    C.CheDtDes,C.CheObser,C.CheFundo,C.CheRepas, B.BanNomes ');
            Sql.Add(' From GrCheRe C, GrClien Cli, GrBanco B ');
            Sql.Add(' Where  (C.CheClien = Cli.CliNumer) ');
            Sql.Add(' And  (C.CheBanco = B.BanNumer) ');
            if not cbTodosCheques.Checked then
                begin // Mostra somente os que não foram baixados.
                    Sql.Add(' and (C.CheBaixa = :vBaixa) ');
                    ParamByName('vBaixa').AsBoolean := false;
                end;

            if not cbMostra.Checked then
                begin
                    Sql.Add(' And  (C.CheRepas = :vRepas) ');
                    ParamByName('vRepas').AsBoolean := false;
                end;
            end;
        end;
    end;
end;

```

```

if rgOrdem.ItemIndex = 0 then
begin
    Sql.Add(' Order by C.CheNrChe ');
end
else
if rgOrdem.ItemIndex = 1 then
begin
    Sql.Add(' Order by Cli.CliNomes ');
end
else
begin
    Sql.Add(' Order by C.CheDtDes ');
end;

Open;
end;

dbgCheRe.Columns[0].Width := 50;
dbgCheRe.Columns[1].Width := 190;
dbgCheRe.Columns[2].Width := 63;
dbgCheRe.Columns[3].Width := 120;
dbgCheRe.Columns[4].Width := 80;
dbgCheRe.Columns[5].Width := 60;

end;

procedure TTelTerce.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Action := caFree;
    TelTerce := nil;
end;

procedure TTelTerce.FormShow(Sender: TObject);
begin
    dm.QPesquisa.Close;
    dm.QPesquisa.SQL.Clear;
    dm.QPesquisa.SQL.Add('Select * from leConfi where ConNumer = 20');
    dm.QPesquisa.Open;

    if dm.QPesquisa.FieldByName('ConNumer').AsInteger = 20 then
    begin
        if dm.QPesquisa.FieldByName('ConLinha').AsString = '1' then
            cbTodosCheques.Checked := true
        else
            cbTodosCheques.Checked := false;
    end;

    TrazFiguras(TelTerce);
    CarregaFiguraBitBtn(btnGrav2, 'BTNGRAVA');
    CarregaFiguraBitBtn(btnCorrige, 'PESQUISA');
    rgOrdemClick(Sender);

```



```

deDatas.Date := date;
deDtDesp.Date := date;
end;

procedure TTelTerce.FormKeyPress(Sender: TObject; var Key: Char);
begin
    TestaTecla(Self,Key);
end;

procedure TTelTerce.btnSaidaClick(Sender: TObject);
begin
    Close;
end;

procedure TTelTerce.edNrCheKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    try
        if edNrChe.Text <> '' then
            begin
                if edNrChe.Text[1] in ['0'..'9'] then
                    QCheRe.Locate('CheNrChe',edNrChe.Text,[LoPartialKey])
                else
                    QCheRe.Locate('CheCeden',edNrChe.Text,[LoPartialKey])
                end;
            end;
    except
        end;
end;

procedure TTelTerce.edNrCheKeyPress(Sender: TObject; var Key: Char);
begin
    if Key = #13 then
        dbgCheRe.SetFocus;
end;

procedure TTelTerce.rgOrdemClick(Sender: TObject);
begin
    MontaQueryChequesRecebidos;
    dbgCheRe.ShowHint := cbMostra.Checked;

end;

procedure TTelTerce.btnGravaClick(Sender: TObject);
var
    vNumero : Longint;
begin
    if ActiveControl = btnSaida then
        exit;

    if QCheRe.FieldByName('CheRepas').AsBoolean then
        begin

```

```

    ShowMessage('Cheque já foi repassado');
    exit;
end;

if vForNumer <= 0 then
begin
    showMessage('Informe o terceiro....');
    CeNomes.SetFocus;
    exit;
end;

if QCheRe.RecordCount = 0 then
begin
    showMessage('Não existe cheques para repassar...');
    exit;
end;

if CeNomes.Text = '' then
begin
    MessageDlg('Atenção! Informe o nome da pessoa a quem vai ser repassado o cheque.',
        mtInformation,[mbOk],0);
    ceNomes.SetFocus;
    exit;
end;
try
    pnAdici.Enabled := false;

    vNumero := BuscaNumero('GrCheTe','CheNumer');

    with Dm.QInserir do
    begin
        Close;
        Sql.Clear;
        Sql.Add('Insert Into GrCheTe ');
        Sql.Add(' (CheNumer, CheChequ,CheNomes,CheForne,CheDatas,CheDtMod,CheDespe) ');
        Sql.Add(' Values (:vNumer, :vChequ, :vNomes, :vForne, :vDatas,:vDtMod,:vDespe) ');
        ParamByName('vNumer').AsInteger := vNumero;
        ParamByName('vChequ').AsInteger := QCheRe.FieldByName('CheNumer').AsInteger;
        ParamByName('vNomes').AsString := ceNomes.Text;
        ParamByName('vForne').AsInteger := vForNumer;
        ParamByName('vDatas').DataType := ftDate;
        ParamByName('vDatas').AsDateTime := deDatas.Date;
        ParamByName('vDtMod').AsDateTime := Date;
        ParamByName('vDespe').AsBoolean := cbDespe.Checked;
        ExecSql;

        MarcarChequeRepassadoParaTerceiro(QCheRe.FieldByName('CheNumer').AsInteger,True,
            '',vForNumer);

    end;
    MontaQueryChequesRecebidos;
    MontaQueryChequesRepassadosParaTerceiros;

```

```

except
  On E:Exception do
    MessageDlg('Erro ao gravar cheque repassado para terceiros.',
      mtInformation,[mbOk],0);
  end;
  pnAdici.Enabled := true;
end;

procedure TTelTerce.dbgCheReCellClick(Column: TColumn);
begin
  if rgOrdem.ItemIndex = 1 then
    edNrChe.Text := QCheRe.FieldByName('CheNrChe').AsString
  else
    edNrChe.Text := QCheRe.FieldByName('CheCeden').AsString;

    btnGrava.Enabled := not btnGrava.Enabled;
  end;

procedure TTelTerce.dbgCheReEnter(Sender: TObject);
begin
  if rgOrdem.ItemIndex = 1 then
    edNrChe.Text := QCheRe.FieldByName('CheNrChe').AsString
  else
    edNrChe.Text := QCheRe.FieldByName('CheCeden').AsString;
end;

procedure TTelTerce.CeNomesButtonClick(Sender: TObject);
begin
  if not dbgPesqu.Visible then
    begin
      MostrarDbgPesqu(Sender);
      dbgPesqu.Height := 197;
      dbgPesqu.Width := 568;
    end;
  ceNomes.SetFocus;

end;

procedure TTelTerce.CeNomesEnter(Sender: TObject);
begin
  tbPesqu.close;
  tbPesqu.TableName := 'GrForne';
  tbPesqu.IndexName := 'Nomes';
  tbPesqu.Open;
  dsPesqu.DataSet := tbPesqu;
  dbgPesqu.Parent := PnDados;
  dbgPesqu.Columns.Clear;
  dbgPesqu.Columns.Add.FieldName := 'ForNumer';
  dbgPesqu.Columns[0].Title.Caption := 'Código';
  dbgPesqu.Columns.Add.FieldName := 'ForNomes';

```

```

dbgPesqu.Columns[0].Width := 50;
dbgPesqu.Columns[1].Title.Caption := 'Nome';
dbgPesqu.Columns[1].Width := 288;
dbgPesqu.Columns.Add.FieldName := 'ForNrDoc';
dbgPesqu.Columns[2].Title.Caption := 'CPF/CGC';
dbgPesqu.Columns.Add.FieldName := 'ForIdent';
dbgPesqu.Columns[3].Title.Caption := 'Identidade';
dbgPesqu.Width := 568;

CeNomesButtonClick(Sender);

end;

procedure TTelTerce.CeNomesKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
if not dbgPesqu.Visible then
begin
if Key = vk_f2 then
begin
MostrarDbgPesqu(Sender);
dbgPesqu.Width := 568;
dbgPesqu.Height := 197;
dbgPesqu.Visible := true;
end;
end;
end;

end;

procedure TTelTerce.CeNomesKeyPress(Sender: TObject; var Key: Char);
begin
if key = #0 then
begin
if dbgPesqu.Visible then
dbgPesqu.SetFocus;
end;
end;

end;

procedure TTelTerce.CeNomesKeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
if dbgPesqu.Visible then
if ceNomes.Text <> '' then
tbPesqu.Locate('ForNomes',ceNomes.Text,[loPartialKey]);
end;

end;

procedure TTelTerce.dbgPesquDbClick(Sender: TObject);
begin
vForNumer := tbPesqu.FieldByName('ForNumer').AsInteger;
ceNomes.Text := tbPesqu.FieldByName('ForNomes').AsString;

```

```

TelTerce.KeyPreview := true;
dbgPesqu.Visible := false;
edNrChe.SetFocus;
MontaQueryChequesRepassadosParaTerceiros;
grCheTe.Caption := grCheTe.Caption + ceNomes.Text;

end;

procedure TTelTerce.dbgPesquKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #13 then
        dbgPesquDbClick(Sender);
end;

procedure TTelTerce.CeNomesExit(Sender: TObject);
begin
    if ActiveControl = btnSaida then
        exit;

    if ceNomes.Text = '' then
        exit;

end;

procedure TTelTerce.btnCanceClick(Sender: TObject);
var
    vObser : String;
begin
    if MessageDlg('Confirma a Exclusão do ' + grCheTe.Caption + '?',
        mtInformation,[mbYes,mbNO],0) = mrYes then
        begin
            dm.QPesquisa.Close;
            dm.QPesquisa.SQL.Clear;
            dm.QPesquisa.Sql.Add('Select CheObser from grCheRe ');
            dm.QPesquisa.Sql.Add('Where CheNumer = :vChequ ');
            dm.QPesquisa.ParamByName('vChequ').AsInteger :=
                QCheTe.FieldByName('CheChequ').AsInteger;
            dm.QPesquisa.Open;
            vObser := dm.QPesquisa.FieldByName('CheObser').AsString;
            dm.QPesquisa.Close;

            DeletarDados('GrCheTe',
                'CheChequ',
                QCheTe.FieldByName('CheChequ').AsInteger);

            MarcarChequeRepassadoParaTerceiro(QCheTe.FieldByName('CheChequ').AsInteger,False,vObs
er,0);
            MontaQueryChequesRecebidos;
            MontaQueryChequesRepassadosParaTerceiros;

```

```

        end;

end;

procedure TTelTerce.btnGrav2Click(Sender: TObject);
var
    vNumero,vNumeroTroco : Longint;
    vStr : String;
    vData : TDate;
begin
    try
        vStr := deDtDesp.Text;
        vData := StrToDate(vStr);
    except
        MessageDlg('Data Troco/Valor inválida.',mtInformation,[mbOk],0);
        exit;
    end;

    if meObser.Lines.Text = '' then
    begin
        MessageDlg('Informe a observação.',mtInformation,[mbOk],0);
        meObser.SetFocus;
        exit;
    end;

    if curValor.Value > 0 then
    begin
        if rgTipos.ItemIndex = -1 then
        begin
            MessageDlg('Informe o tipo.',mtInformation,[mbOk],0);
            exit;
        end;
        vNumeroTroco := BuscaNumero('GrTroco','TroNumer');
        with dm.QInserir do
        begin
            Close;
            Sql.Clear;
            Sql.Add('Insert Into GrTroco ');
            Sql.Add('Values (:vNumer,:vTerce,:vTipos,:vDatas,:vValor,:vObser)');
            ParamByName('vNumer').AsInteger := vNumeroTroco;
            ParamByName('vTerce').AsInteger := vForNumer;
            ParamByName('vTipos').AsInteger := rgTipos.ItemIndex + 1;
            ParamByName('vDatas').DataType := ftDate;
            ParamByName('vDatas').AsDateTime := vData; // Solicitado por Silene
            ParamByName('vValor').AsCurrency := curValor.Value;
            ParamByName('vObser').AsString := meObser.Text;
            ExecSql;
        end;

        vNumero := BuscaNumero('GrCader','CadNumer');
    end;

```

```

with dm.QInserir do
begin
  Close;
  Sql.Clear;
  Sql.Add('Insert Into GrCader ');
  Sql.Add('Values (:vNumer,:vTipos,:vDadas,:vValor,:vHisto,:vTroco)');
  ParamByName('vNumer').AsInteger := vNumero;
  ParamByName('vTipos').AsInteger := rgTipos.ItemIndex + 1;
  ParamByName('vDadas').DataType := ftDate;
  ParamByName('vDadas').AsDateTime := vData; // Solicitado por Silene
  ParamByName('vValor').AsCurrency := curValor.Value;
  ParamByName('vHisto').AsString := meObser.Text;
  ParamByName('vTroco').AsInteger := vNumeroTroco;
  ExecSql;
end;

rgTipos.ItemIndex := -1;
curValor.Value := 0;
meObser.Clear;
end
else
begin
  MessageDlg('Informe o valor.',mtInformation,[mbOk],0);
  curValor.SetFocus;
end;

end;

procedure TTelTerce.curValorKeyPress(Sender: TObject; var Key: Char);
begin
  if key = '.' then
    key := ',';

end;

procedure TTelTerce.dbgCheReDrawColumnCell(Sender: TObject;
const Rect: TRect; DataCol: Integer; Column: TColumn;
State: TGridDrawState);
begin
  if cbMostra.Checked then
  begin
    if not QCheRe.FieldByName('CheRepas').AsBoolean then
    begin
      dbgCheRe.Canvas.Font.Color := clBlack;
      dbgCheRe.Canvas.Brush.Color := clWhite;
    end
    else
    begin
      dbgCheRe.Canvas.Font.Color := clBlack;

```

```

        dbgCheRe.Canvas.Brush.Color := clYellow;
    end;

    dbgCheRe.Canvas.FillRect(Rect);
    dbgCheRe.Canvas.TextOut(Rect.left + 2, Rect.top, Column.Field.AsString);
end;

end;

procedure TTelTerce.BtnCorrigeClick(Sender: TObject);
begin
    if FCorrigeTroco = nil then
        FCorrigeTroco := TFCorrigeTroco.Create(Self);

    FCorrigeTroco.vCorrecao := 1;
    FCorrigeTroco.ShowModal;
    FCorrigeTroco.Free;
    FCorrigeTroco := nil;
end;

procedure TTelTerce.cbMostraClick(Sender: TObject);
begin
    rgOrdemClick(Sender);
    btnGrava.Enabled := not btnGrava.Enabled;

end;

procedure TTelTerce.BitBtn1Click(Sender: TObject);
begin
    DbgPesqu.Visible := false;
    if TelForne = nil then
        TelForne := TTelForne.Create(Self); {UForne}
    TelForne.ShowModal;
    CeNomes.Clear;
    CeNomesEnter(ceNomes);
end;
end.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```


Lista de Cheques Para Banco

Relação de Cheques Para Enviar para Bancos

Cheques Recebidos

Número	Dt Desconto	Valor	Cedente
63252	16/5/2013	R\$ 300,20	ADRIANA DA SILVA
2523	16/5/2013	R\$ 360,80	ADRIANA DA SILVA

Cheques para enviar

Número	Valor	Dt Desconto	Cedente
--------	-------	-------------	---------

Banco: BRASIL Cedente: ADRIANA DA SILVA Valor: R\$ 300,20

Sem Fundo: Não Dt Emissão: 16/4/2013 Dt Desconto: 16/5/2013

Impressora Visualizar Imprimir Cancelar Saída

unit ULista;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
ExtCtrls, StdCtrls, Buttons, Grids, DBGrids, Mask, ToolEdit, Db, DBTables,
CurrEdit;

type

```
TTelLista = class(TForm)
    pnDados: TPanel;
    Panel2: TPanel;
    btnSaída: TBitBtn;
    btnCance: TBitBtn;
    GroupBox1: TGroupBox;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    GroupBox2: TGroupBox;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    dsCheRe: TDataSource;
    dsTempo: TDataSource;
    QCheRe: TQuery;
    QCheReCheNumer: TIntegerField;
    QCheReCheNrChe: TStringField;
    QCheReCheDtEmi: TDateField;
    QCheReCheDtDes: TDateField;
```

```

QCheReCheValor: TCurrencyField;
QCheReCheCeden: TStringField;
QCheReCheBanco: TIntegerField;
QCheReBanNomes: TStringField;
Panel1: TPanel;
Label1: TLabel;
edCeden: TEdit;
Label2: TLabel;
edBanco: TEdit;
Label3: TLabel;
curValor: TCurrencyEdit;
edFundo: TEdit;
Label4: TLabel;
edDtEmi: TEdit;
Label5: TLabel;
edDtDes: TEdit;
Label6: TLabel;
QCheReCheFundo: TBooleanField;
tbTempo: TTable;
tbTempoCheNumer: TIntegerField;
tbTempoCheNrChe: TStringField;
tbTempoCheBanco: TStringField;
tbTempoCheDtDes: TDateField;
tbTempoCheValor: TCurrencyField;
tbTempoCheCeden: TStringField;
tbTempoCheDtEmi: TDateField;
edNrChe: TEdit;
btnImpri: TBitBtn;
btnVisua: TBitBtn;
btnImpre: TBitBtn;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure btnSaidaClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure QCheReBeforeScroll(DataSet: TDataSet);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure edNrCheKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure btnImpreClick(Sender: TObject);
procedure btnImpriClick(Sender: TObject);
procedure btnCanceClick(Sender: TObject);
private
  { Private declarations }
  vBanNumer : Longint;
  function CriarTabelaTemporaria : Boolean;
  procedure MostraDadosTabela;
  procedure MontaQueryChequesRecebidos;
public
  { Public declarations }
end;

```

```

var
  TelLista: TTelLista;

implementation
Uses
  UProce, UDataM, QriRList;
{$R *.DFM}

function TTelLista.CriarTabelaTemporaria : Boolean;
var
  vNome : String;
begin
  try
    with dm.QCriaT do
      begin
        inc(vSequencia);
        vNome := 'tmp' + FormatFloat('0000', vControle) + FormatFloat('0000', vSequencia);
        Close;
        Sql.Clear;
        Sql.Add('Create Table ' + vNome );
        Sql.Add('(CheNumer Integer,CheNrChe VarChar(15),CheBanco VarChar(60),');
        Sql.Add(' CheDtDes Date,CheDtEmi Date, CheValor Money, CheCeden VarChar(60),');
        Sql.Add('Primary Key (CheNumer))');
        ExecSql;
        Close;
        Close;
        tbTempo.Close;
        tbTempo.TableName := vNome;
        tbTempo.Open;
        Result := true
      end;
    except
      On E:Exception do
        begin
          vMensagem := e.Message;
          Result := false;
        end;
      end;
    end;
end;

procedure TTelLista.MostraDadosTabela;
begin
  with QCheRe do
    begin
      edBanco.Text := QCheRe.FieldName('BanNomes').AsString;
      edCeden.Text := QCheRe.FieldName('CheCeden').AsString;
      curValor.Value := QCheRe.FieldName('CheValor').AsCurrency;
      if Not QCheRe.FieldName('CheFundo').AsBoolean then
        begin
          edFundo.Text := 'Não';
          edFundo.Color := clScrollBar;
        end;
    end;
  end;
end;

```

```

        end
    else
        begin
            edFundo.Text := 'Sim';
            edFundo.Color := clRed;
        end;
        edDtEmi.Text := QCheRe.FieldName('CheDtEmi').AsString;
        edDtDes.Text := QCheRe.FieldName('CheDtDes').AsString;
    end;
end;

procedure TTelLista.MontaQueryChequesRecebidos;
begin
    With QCheRe do
        begin
            Close;
            Sql.Clear;
            Sql.Add('Select C.CheNumer,C.CheNrChe,C.CheDtEmi,C.CheDtDes,');
            Sql.Add('C.CheValor,C.CheCeden,C.CheBanco,C.CheFundo, B.BanNomes ');
            Sql.Add(' From GrCheRe C, GrBanco B');
            Sql.Add('Where (CheRepas = False) ');
            Sql.Add(' And (CheBaixa = False) ');
            Sql.Add(' And (C.CheBanco = B.BanNumer) ');
            Sql.Add(' Order By CheDtDes,CheCeden ');
            Open;
        end;
    end;

end;

procedure TTelLista.FormClose(Sender: TObject; var Action: TCloseAction);
begin

    try
        with tbTempo do
            begin
                Close;
                DeleteTable;
            end;
    except
    end;

    Action := caFree;
    TelLista := Nil;
end;

procedure TTelLista.btnSaidaClick(Sender: TObject);
begin
    Close;
end;

```

```

procedure TTellista.FormShow(Sender: TObject);
begin
  TrazFiguras(Self);
  MontaQueryChequesRecebidos;
  MostraDadosTabela;
  if not CriarTabelaTemporaria then
  begin
    MessageDlg('Erro ao criar tabela temporária.',
      mtInformation,[mbOk],0);
    Close;
  end;
end;

procedure TTellista.FormKeyPress(Sender: TObject; var Key: Char);
begin
  TestaTecla(Self,key);
end;

procedure TTellista.QCheReBeforeScroll(DataSet: TDataSet);
begin
  MostraDadosTabela;
end;

procedure TTellista.BitBtn1Click(Sender: TObject);
begin
  with Tbtempo do
  begin
    tbTempo.IndexName := '';
    if Not FindKey([QCheRe.FieldByName('CheNumer').AsInteger]) then
    begin
      Append;
      FieldByName('CheNumer').AsInteger := QCheRe.FieldByName('CheNumer').AsInteger;
      FieldByName('CheNrChe').AsString := QCheRe.FieldByName('CheNrChe').AsString;
      FieldByName('CheCeden').AsString := QCheRe.FieldByName('CheCeden').AsString;
      FieldByName('CheBanco').AsString :=

BuscaNome('GrBanco','BanNomes','BanNumer',QCheRe.FieldByName('CheBanco').AsInteger);
      FieldByName('CheValor').AsCurrency := QCheRe.FieldByName('CheValor').AsCurrency;
      FieldByName('CheDtDes').AsDateTime := QCheRe.FieldByName('CheDtDes').AsDateTime;
      FieldByName('CheDtEmi').AsDateTime :=
      QCheRe.FieldByName('CheDtEmi').AsDateTime;
      Post;

    end;
  end;
  QCheRe.Next;
end;

procedure TTellista.BitBtn2Click(Sender: TObject);
begin
  TbTempo.Delete;
end;

```

```

procedure TTellLista.edNrCheKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if edNrChe.Text <> '' then
    QCheRe.Locate('CheNrChe',edNrChe.Text,[loPartialKey]);
end;

procedure TTellLista.btnImpreClick(Sender: TObject);
begin
  PSetup;
end;

procedure TTellLista.btnImpriClick(Sender: TObject);
begin
  if TQiRList = nil then
    TQiRList := TTQiRList.Create(Self); {GerRLis}

  TQiRList.AReport := TQiRList.QuickReport;
  TQiRList.QRLTitul01.Caption := TQiRList.QRLTitul01.Caption + ' ' + DateToStr(Date);
  if Sender = BtnVisua then
    TQiRList.Areport.Preview
  else
    TQiRList.Areport.Print;

  TQiRList.Free;
  TQiRList := nil;

end;

procedure TTellLista.btnCanceClick(Sender: TObject);
begin
  With TbTempo do
    begin
      tbTempo.Close;
      tbTempo.EmptyTable;
      tbTempo.Open;
    end;
  QCheRe.First;
end;
end.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

Controle Mensal

Fechamento Mês

Mês: **ABRIL** Ano: **2013** **1 = Vendas; 2 = Compras; 3 = Despesas**

Descrição	Valor	1 - 2 - 3
PAGAMENTO LUZ	R\$ 150,36	3
PAGAMENTO AGUA	R\$ 100,00	3
COMPRA DE PRODUTOS	R\$ 1.500,00	2
VENDAS MES	R\$ 3.052,60	1
*		

Total

Venda: **R\$ 3.052,60**

Compra: **R\$ 1.500,00**

Despesas: **R\$ 250,36**

Lucro Líquido Obtido: **R\$ 1.302,24**

Controle Bancário

Saídas = Cheques Emitidos: **R\$ 0,00**

Entradas = Depósitos: **R\$ 0,00**

Saldo: **R\$ 0,00**

Mês: ABRIL 2013 ☒ Gravar ☐ Excluir ☐ Fechar

unit UFecha;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, Buttons, ExtCtrls, Grids, DBGrids, Db, DBTables, Mask, CurrEdit;

type

```
TTelFecha = class(TForm)
    PnDados: TPanel;
    PnBotao: TPanel;
    BtnCance: TBitBtn;
    BtnSaida: TBitBtn;
    tbTempo: TTable;
    dsTempo: TDataSource;
    tbTempoFecNumer: TIntegerField;
    tbTempoFecAnoFe: TIntegerField;
    tbTempoFecMesFe: TIntegerField;
    tbTempoFecDescr: TStringField;
    tbTempoFecValor: TCurrencyField;
    tbTempoFecTipos: TStringField;
    GroupBox1: TGroupBox;
```

```

Label18: TLabel;
cbFeMes: TComboBox;
edFeAno: TEdit;
Label1: TLabel;
Label2: TLabel;
Panel1: TPanel;
GroupBox2: TGroupBox;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
curVenda: TCurrencyEdit;
curCompra: TCurrencyEdit;
curDespe: TCurrencyEdit;
curLucro: TCurrencyEdit;
GroupBox3: TGroupBox;
LSaida: TLabel;
LEntra: TLabel;
Label9: TLabel;
curSaida: TCurrencyEdit;
curEntra: TCurrencyEdit;
curSaldo: TCurrencyEdit;
dbgFecha: TDBGrid;
tbAux: TTable;
StringField1: TStringField;
CurrencyField1: TCurrencyField;
IntegerField1: TIntegerField;
IntegerField2: TIntegerField;
IntegerField3: TIntegerField;
StringField2: TStringField;
btnGrava: TBitBtn;
QFeMes: TQuery;
btnExclu: TBitBtn;
tbAux2: TTable;
IMes: TLabel;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure FormShow(Sender: TObject);
procedure BtnSaidaClick(Sender: TObject);
procedure edFeAnoKeyPress(Sender: TObject; var Key: Char);
procedure tbTempoBeforePost(DataSet: TDataSet);
procedure dbgFechaKeyPress(Sender: TObject; var Key: Char);
procedure tbTempoAfterPost(DataSet: TDataSet);
procedure btnGravaClick(Sender: TObject);
procedure cbFeMesClick(Sender: TObject);
procedure edFeAnoExit(Sender: TObject);
procedure curSaidaKeyPress(Sender: TObject; var Key: Char);
procedure curSaidaExit(Sender: TObject);
procedure btnExcluClick(Sender: TObject);
procedure BtnCanceClick(Sender: TObject);
private
{ Private declarations }

```



```

vGravar : Boolean;
procedure CarregaTabelaTemporaria;
procedure SomaValores;
function CriarTabelaTemporaria : Boolean;
public
{ Public declarations }
end;

var
  TelFecha: TTelFecha;

implementation
Uses
  UProce, UDataM;
{$R *.DFM}

procedure TTelFecha.CarregaTabelaTemporaria;
var
  vAno, vMes,
  vNumero : Longint;
begin
  curVenda.Value := 0;
  curCompra.Value := 0;
  curDespe.Value := 0;
  curLucro.Value := 0;
  curSaida.Value := 0;
  curEntra.Value := 0;
  curSaldo.Value := 0;

  vGravar := false;
  curSaida.Value := 0;
  curEntra.Value := 0;
  curSaldo.Value := 0;
  with tbTempo do
    begin
      Close;
      EmptyTable;
      Open;
    end;
  with QFeMes do
    begin
      Close;
      Sql.Clear;
      Sql.Add('Select * From GrFeMes ');
      Sql.Add('Where (FecAnoFe = :vAnoFe) ');
      Sql.Add(' And (FecMesFe = :vMesFe) ');
      Sql.Add(' Order By FecTipos, FecNumer ');
      ParamByName('vAnoFe').AsInteger := StrToInt(edFeAno.text);
      ParamByName('vMesFe').AsInteger := cbFeMes.ItemIndex + 1;
      Open;
      First;
      if RecordCount > 0 then

```

```

begin
while not Eof do
begin
if FieldByName('FecTipos').AsInteger <> 0 then
begin
vNumero := BuscaNumero(tbTempo.TableName,'FecNumer');
tbTempo.Append;
TbTempo.FieldByName('FecNumer').AsInteger := vNumero;
TbTempo.FieldByName('FecDescr').AsString :=
FieldByName('FecDescr').AsString;
TbTempo.FieldByName('FecValor').AsCurrency :=
FieldByName('FecValor').AsCurrency;
TbTempo.FieldByName('FecTipos').AsInteger :=
FieldByName('FecTipos').AsInteger;

tbTempo.Post;
end
else
if FieldByName('FecDescr').AsString = LSaida.Caption then
curSaida.Value := FieldByName('FecValor').AsCurrency
else
if FieldByName('FecDescr').AsString = LEntrada.Caption then
curEntrada.Value := FieldByName('FecValor').AsCurrency;

Next;
end;
end
else
begin
tbAux2.Close;
tbAux2.TableName := 'GrFeMes';
tbAux2.IndexName := '';
TbAux2.Open;
tbAux2.Last;
Close;
Sql.Clear;
Sql.Add('Select * From GrFeMes ');
Sql.Add('Where (FecAnoFe = :vAnoFe) ');
Sql.Add(' And (FecMesFe = :vMesFe) ');
Sql.Add(' Order By FecTipos, FecNumer ');
ParamByName('vAnoFe').AsInteger := tbAux2.FieldByName('FecAnoFe').AsInteger;
ParamByName('vMesFe').AsInteger := tbAux2.FieldByName('FecMesFe').AsInteger;
Open;

First;
vAno := QFeMes.FieldByName('FecAnoFe').AsInteger;
vMes := QFeMes.FieldByName('FecMesFe').AsInteger;
while ((not Eof) and ((vAno = QFeMes.FieldByName('FecAnoFe').AsInteger) and
(vMes = QFeMes.FieldByName('FecMesFe').AsInteger) ) ) do
begin
if FieldByName('FecTipos').AsInteger <> 0 then
begin

```

```

        vNumero := BuscaNumero(tbTempo.TableName,'FecNumer');
        tbTempo.Append;
        TbTempo.FieldName('FecNumer').AsInteger := vNumero;
        TbTempo.FieldName('FecDescr').AsString :=
            FieldByName('FecDescr').AsString;
        TbTempo.FieldName('FecTipos').AsInteger :=
            FieldByName('FecTipos').AsInteger;

        tbTempo.Post;
    end;
    Next;
end;
end;
end;
dbgFecha.Columns[0].Width := 230;
dbgFecha.Columns[1].Width := 80;
dbgFecha.Columns[2].Width := 50;

curSaldo.Value := curEntra.Value - curSaida.Value;
tbTempo.First;

IMes.Caption := 'Mês: ' + cbFeMes.Text + ' ' + edFeAno.Text;
btnGrava.Enabled := true;
end;

procedure TTelFecha.SomaValores;
begin
    curVenda.Value := 0;
    curCompra.Value := 0;
    curDespe.Value := 0;
    tbAux.close;
    tbAux.TableName := tbTempo.TableName;
    tbAux.Open;
    tbAux.First;
    While not tbAux.Eof do
        begin
            if tbAux.FieldName('FecTipos').AsInteger = 1 then
                curVenda.Value := curVenda.Value + tbAux.FieldName('FecValor').AsCurrency
            else
                if tbAux.FieldName('FecTipos').AsInteger = 2 then
                    curCompra.Value := curCompra.Value + tbAux.FieldName('FecValor').AsCurrency
                else
                    if tbAux.FieldName('FecTipos').AsInteger = 3 then
                        curDespe.Value := curDespe.Value + tbAux.FieldName('FecValor').AsCurrency;

                    tbAux.Next;
                end;
            end;

        curLucro.Value := curVenda.Value - (curCompra.Value + curDespe.Value);

        curSaldo.Value := curEntra.Value - curSaida.Value;
    end;

```

```

function TTelFecha.CriarTabelaTemporaria : Boolean;
var
  vNome : String;
begin
  try
    with dm.QCriaT do
      begin
        inc(vSequencia);
        vNome := 'tmp' + FormatFloat('0000', vControle) + FormatFloat('0000', vSequencia);
        Close;
        Sql.Clear;
        Sql.Add('Create Table ' + vNome );
        Sql.Add('(FecNumer Integer,FecAnoFe Integer, FecMesFe Integer, ');
        Sql.Add(' FecDescr VarChar(50),FecValor Money,FecTipos VarChar(1), ');
        Sql.Add('Primary Key (FecNumer))');
        ExecSql;
        Close;

        tbTempo.Close;
        tbTempo.TableName := vNome;
        tbTempo.Open;
        dbgFecha.Columns[0].Width := 270;
        Result := true
      end;
    except
      On E:Exception do
        begin
          vMensagem := e.Message;
          Result := false;
        end;
      end;
    end;
end;

procedure TTelFecha.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  try
    with tbTempo do
      begin
        tbAux.Close;
        Close;
        DeleteTable;
      end;
    except
      end;
    Action := caFree;
    TelFecha := nil;
  end;

  procedure TTelFecha.FormKeyPress(Sender: TObject; var Key: Char);
  begin
    TestaTecla(Self,key);
  end;
end;

```

end;

procedure TTelFecha.FormShow(Sender: TObject);

var

vAno,vMes,vDia : Word;

begin

TrazFiguras(Self);

if not CriarTabelaTemporaria then

begin

MessageDlg('Erro ao criar tabela temporaria.',
mtInformation,[mbOk],0);

Close;

end;

decodeDate(date,vAno,vMes,vDia);

cbFeMes.ItemIndex := vMes - 1;

edFeAno.Text := IntToStr(vAno);

CarregaTabelaTemporaria;

tbTempo.First;

dbgFecha.SetFocus;

end;

procedure TTelFecha.BtnSaidaClick(Sender: TObject);

begin

if vGravar then

begin

if MessageDlg('Deseja gravar o fechamento do mês?',
mtInformation,[mbYes,mbNo],0) = mrYes then
btnGravaClick(Sender);

end;

Close;

end;

procedure TTelFecha.edFeAnoKeyPress(Sender: TObject; var Key: Char);

begin

if not (key in ['0'..'9',#8]) then

key := #0;

end;

procedure TTelFecha.tbTempoBeforePost(DataSet: TDataSet);

Var

vNumero : Longint;

begin

if tbTempo.State = dsInsert then

begin

vNumero := BuscaNumero(tbTempo.TableName,'FecNumer');

tbTempo.FieldByName('FecNumer').AsInteger := vNumero;

end;

```

end;

procedure TTelFecha.dbgFechaKeyPress(Sender: TObject; var Key: Char);
begin
    vGravar := true;

    if key = #13 then
        key := #0;
    if key = '.' then
        key := ',';
    else
        Key := UpCase(key);

end;

procedure TTelFecha.tbTempoAfterPost(DataSet: TDataSet);
begin
    SomaValores;
end;

procedure TTelFecha.btnGravaClick(Sender: TObject);
var
    vNumero : Longint;
    vContador : Integer;
begin
    btnGrava.Enabled := false;

    if tbTempo.RecordCount = 0 then
        begin
            if MessageDlg('Somente os campos referentes a Total e Controle Bancário referente ao mês
'+ #13 +
                'indicado possuem valores! Confirma a gravação?',MtInformation,[mbyes,mbNo],0) =
mrNo then
                Exit;
            end;

            with Dm.QDelete do
                begin
                    Close;
                    Sql.clear;
                    Sql.Add('Delete From GrFeMes ');
                    Sql.Add('Where (FecAnoFe = :vAnoFe) and (FecMesFe = :vMesFe) ');
                    ParamByName('vAnoFe').AsInteger := StrToInt(edFeAno.Text);
                    ParamByName('vMesFe').AsInteger := cbFeMes.ItemIndex + 1;
                    ExecSql;
                end;

            vContador := 0;
            with tbTempo do
                begin

```

```

First;
Dm.QInserir.Close;
Dm.QInserir.Sql.Clear;
Dm.QInserir.Sql.Add('Insert Into GrFeMes ');
Dm.QInserir.Sql.Add(' (FecNumer,FecAnoFe,FecMesFe,FecDescr,FecValor,FecTipos )');
Dm.QInserir.Sql.Add(' Values (:vNumer,:vAnoFe,:vMesFe,:vDescr,:vValor,:vTipos )');
Dm.QInserir.Prepare;

while not Eof do
begin
    vNumero := BuscaNumero('GrFeMes','FecNumer');
    Dm.QInserir.ParamByName('vNumer').AsInteger := vNumero;
    Dm.QInserir.ParamByName('vAnoFe').AsInteger := StrToInt(edFeAno.Text);
    Dm.QInserir.ParamByName('vMesFe').AsInteger := cbFeMes.ItemIndex + 1;
    Dm.QInserir.ParamByName('vDescr').AsString :=
        TbTempo.FieldByName('FecDescr').AsString;
    Dm.QInserir.ParamByName('vValor').AsCurrency :=
        TbTempo.FieldByName('FecValor').AsCurrency;
    Dm.QInserir.ParamByName('vTipos').AsInteger :=
        TbTempo.FieldByName('FecTipos').AsInteger;

    Dm.QInserir.ExecSql;
    Next;
end;
end;

with Dm.QDelete do
begin
    Close;
    Sql.clear;
    Sql.Add('Select * From GrFeMes ');
    Sql.Add('Where (FecAnoFe = :vAnoFe) and (FecMesFe = :vMesFe) ');
    ParamByName('vAnoFe').AsInteger := StrToInt(edFeAno.Text);
    ParamByName('vMesFe').AsInteger := cbFeMes.ItemIndex + 1;
    Open;
    if RecordCount = 0 then
        if MessageDlg('Não foi gravado nenhum registro referente ao fechamento do mes. '+ #13
+
        'Pode ter havido um erro no momento da gravação, ou todos as contas do mês
anterior foram exluídas. '+ #13 +
        'Deseja continuar a gravação mesmo assim?',MtInformation,[mbyes,mbNo],0) =
mrNo then
            Exit;

    end;
    // Grava Controle Bancario - Inicio

    // Saída = Cheques Emitidos
Dm.QInserir.Close;
Dm.QInserir.Sql.Clear;

```

```

vNumero := BuscaNumero('GrFeMes','FecNumer');
Dm.QInserir.Sql.Add('Insert Into GrFeMes ');
Dm.QInserir.Sql.Add(' (FecNumer,FecAnoFe,FecMesFe,FecDescr,FecValor,FecTipos)');
Dm.QInserir.Sql.Add(' Values (:vNumer,:vAnoFe,:vMesFe,:vDescr,:vValor,:vTipos)');
Dm.QInserir.ParamByName('vNumer').AsInteger := vNumero;
Dm.QInserir.ParamByName('vAnoFe').AsInteger := StrToInt(edFeAno.Text);
Dm.QInserir.ParamByName('vMesFe').AsInteger := cbFeMes.ItemIndex + 1;
Dm.QInserir.ParamByName('vDescr').AsString := LSaida.Caption;
Dm.QInserir.ParamByName('vValor').AsCurrency := curSaida.Value;
Dm.QInserir.ParamByName('vTipos').AsInteger := 0;
Dm.QInserir.ExecSql;

```

// Entradas = Depósito

```

Dm.QInserir.Close;
Dm.QInserir.Sql.Clear;
vNumero := BuscaNumero('GrFeMes','FecNumer');
Dm.QInserir.Sql.Add('Insert Into GrFeMes ');
Dm.QInserir.Sql.Add(' (FecNumer,FecAnoFe,FecMesFe,FecDescr,FecValor,FecTipos)');
Dm.QInserir.Sql.Add(' Values (:vNumer,:vAnoFe,:vMesFe,:vDescr,:vValor,:vTipos)');
Dm.QInserir.ParamByName('vNumer').AsInteger := vNumero;
Dm.QInserir.ParamByName('vAnoFe').AsInteger := StrToInt(edFeAno.Text);
Dm.QInserir.ParamByName('vMesFe').AsInteger := cbFeMes.ItemIndex + 1;
Dm.QInserir.ParamByName('vDescr').AsString := LEntra.Caption;
Dm.QInserir.ParamByName('vValor').AsCurrency := curEntra.Value;
Dm.QInserir.ParamByName('vTipos').AsInteger := 0;
Dm.QInserir.ExecSql;
// Grava Controle Bancario - Fim

```

```

with tbTempo do
begin
  Close;
  EmptyTable;
  Open;
end;
vGravar := false;

```

end;

```

procedure TTelFecha.cbFeMesClick(Sender: TObject);
begin
  CarregaTabelaTemporaria;
end;

```

```

procedure TTelFecha.edFeAnoExit(Sender: TObject);
begin
  CarregaTabelaTemporaria;
end;

```

```

procedure TTelFecha.curSaidaKeyPress(Sender: TObject; var Key: Char);
begin

```



```

    if Key = '.' then
        key := ',';
    end;

    procedure TTelFecha.curSaidaExit(Sender: TObject);
    begin
        curSaldo.Value := CurEntra.Value - CurSaida.Value;
    end;

    procedure TTelFecha.btnExcluClick(Sender: TObject);
    begin
        if MessageDlg('Confirma a exclusão do registro?',
            mtInformation,[mbYes,mbNo],0) = mrYes then
            begin
                TbTempo.Delete;
            end;
        end;

    end;

    procedure TTelFecha.BtnCanceClick(Sender: TObject);
    begin
        with tbTempo do
            begin
                Close;
                EmptyTable;
                Open;
            end;
        end;
    end;

end.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

Controle Caderno

```
unit FrmContr;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls, Buttons, ExtCtrls, CurrEdit, Mask, ToolEdit, Db;
```

```
type
```

```
TTelContr = class(TForm)  
    PnBotao: TPanel;  
    BtnGrava: TBitBtn;  
    btnCance: TBitBtn;  
    btnSaida: TBitBtn;  
    btnExclu: TBitBtn;  
    PnDados: TPanel;  
    deDatas: TDateEdit;  
    Label1: TLabel;  
    Label2: TLabel;  
    curValor: TCurrencyEdit;  
    Label3: TLabel;  
    rgTipos: TRadioGroup;  
    meHisto: TMemo;  
    procedure FormClose(Sender: TObject; var Action: TCloseAction);  
    procedure btnSaidaClick(Sender: TObject);  
    procedure FormShow(Sender: TObject);  
    procedure BtnGravaClick(Sender: TObject);  
    procedure btnCanceClick(Sender: TObject);  
    procedure btnExcluClick(Sender: TObject);  
    procedure FormKeyPress(Sender: TObject; var Key: Char);  
    procedure curValorKeyPress(Sender: TObject; var Key: Char);  
private  
    { Private declarations }  
    procedure LimparVariaveis;  
public  
    { Public declarations }  
end;
```

```
var
```

```

TelContr: TTelContr;

implementation
Uses
    UProce, UDataM, UCorrigeTroco;
{$R *.DFM}

procedure TTelContr.LimparVariaveis;
begin
    rgTipos.ItemIndex := -1;
    LimparTela(TelContr);
    deDatas.Date := date;
    deDatas.SetFocus;
end;

procedure TTelContr.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Action := caFree;
    TelContr := nil;
end;

procedure TTelContr.btnSaidaClick(Sender: TObject);
begin
    Close;
end;

procedure TTelContr.FormShow(Sender: TObject);
begin
    TrazFiguras(TelContr);
    deDatas.Date := date;
end;

procedure TTelContr.BtnGravaClick(Sender: TObject);
var
    vNumero : Longint;
begin
    if deDatas.Date = 0 then
    begin
        MessageDlg('Informe a Data!',mtInformation,[mbOk],0);
        deDatas.Setfocus;
        exit;
    end;

    if rgTipos.ItemIndex = -1 then
    begin
        MessageDlg('Informe o Tipo!',mtInformation,[mbOk],0);
        exit;
    end;

    if curValor.Value <= 0 then

```

```

begin
    MessageDlg('Informe o Valor!',mtInformation,[mbOk],0);
    curValor.Setfocus;
    exit;
end;

if meHisto.Text = '' then
begin
    MessageDlg('Informe o Histórico!',mtInformation,[mbOk],0);
    meHisto.Setfocus;
    exit;
end;

if rgTipos.ItemIndex = -1 then
begin
    MessageDlg('Informe o tipo.',mtInformation,[mbOk],0);
    exit;
end;

try
    vNumero := BuscaNumero('GrCader','CadNumer');
    with dm.QInserir do
    begin
        Close;
        Sql.Clear;
        Sql.Add('Insert Into GrCader ');
        Sql.Add('Values (:vNumer,:vTipos,:vDatas,:vValor,:vHisto,:vTroco)');
        ParamByName('vNumer').AsInteger := vNumero;
        ParamByName('vTipos').AsInteger := rgTipos.ItemIndex + 1;
        ParamByName('vDatas').DataType := ftDate;
        ParamByName('vDatas').AsDateTime := deDatas.Date;
        ParamByName('vValor').AsCurrency := curValor.Value;
        ParamByName('vHisto').AsString := meHisto.Text;
        ParamByName('vTroco').AsInteger := 0;
        ExecSql;
    end;
except
    on E:exception do
    begin
        showMessage(e.message);
    end;
end;

LimpVarVariaveis;
end;

procedure TTelContr.btnCanceClick(Sender: TObject);
begin
    LimpVarVariaveis;
end;

```

```
procedure TTelContr.btnExcluClick(Sender: TObject);
begin
```

```
  if FCorrigeTroco = nil then
    FCorrigeTroco := TFCorrigeTroco.Create(Self);
```

```
  FCorrigeTroco.vCorrecao := 2;
```

```
  FCorrigeTroco.ShowModal;
```

```
  FCorrigeTroco.Free;
```

```
  FCorrigeTroco := nil;
```

```
end;
```

```
procedure TTelContr.FormKeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
  TestaTecla(Self,Key);
```

```
end;
```

```
procedure TTelContr.curValorKeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
  if key = '.' then
```

```
    key := ',';
```

```
end;
```

```
end.
```

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Contas à pagar

Nr. Documento	Histórico	Credor	Data Lançamento	Data Vencimento	Valor	Data Pagto
52222	Compra de produtos	J&J - ATACADISTA DE PRODUTOS AL	16/4/2013	15/6/2013	R\$ 1.600,50	

```
unit UCAPag;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
```

```
  StdCtrls, Buttons, ExtCtrls, CurrEdit, ToolEdit, Mask, Grids, DBGrids,
```

```
  Db, DBTables;
```

```

type
TTelCAPag = class(TForm)
  pnDados: TPanel;
  Label1: TLabel;
  Label2: TLabel;
  deDtVto: TDateEdit;
  Label3: TLabel;
  curValor: TCurrencyEdit;
  Label4: TLabel;
  deDtLan: TDateEdit;
  Label5: TLabel;
  meHisto: TMemo;
  tbPesqu: TTable;
  PnBotao: TPanel;
  BtnGrava: TBitBtn;
  btnCance: TBitBtn;
  btnSaida: TBitBtn;
  btnExclu: TBitBtn;
  Panel2: TPanel;
  dbContas: TDBGrid;
  Panel1: TPanel;
  dsContas: TDataSource;
  QContas: TQuery;
  dsPesqu: TDataSource;
  edPesqu: TEdit;
  Label6: TLabel;
  btnPagar: TBitBtn;
  edNrDoc: TEdit;
  Label7: TLabel;
  edPNrDo: TEdit;
  Label8: TLabel;
  ceForne: TComboEdit;
  dbgPesqu: TDBGrid;
  btnImpre: TBitBtn;
  cbContas: TCheckBox;
  deDtPag: TDateEdit;
  Label9: TLabel;
  Panel3: TPanel;
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure btnCanceClick(Sender: TObject);
  procedure btnSaidaClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure ceNomesButtonClick(Sender: TObject);
  procedure ceNomesKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
  procedure ceNomesKeyPress(Sender: TObject; var Key: Char);
  procedure ceNomesKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
  procedure dbgPesquDbClick(Sender: TObject);
  procedure dbgPesquKeyPress(Sender: TObject; var Key: Char);
  procedure BtnGravaClick(Sender: TObject);

```

```

procedure FormKeyPress(Sender: TObject; var Key: Char);
procedure btnExcluClick(Sender: TObject);
procedure cbHistoKeyPress(Sender: TObject; var Key: Char);
procedure curValorKeyPress(Sender: TObject; var Key: Char);
procedure FormShow(Sender: TObject);
procedure ceNomesEnter(Sender: TObject);
procedure edPesquKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure edPesquKeyPress(Sender: TObject; var Key: Char);
procedure edPNrDoKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure dbContasCellClick(Column: TColumn);
procedure dbContasKeyPress(Sender: TObject; var Key: Char);
procedure btnPagarClick(Sender: TObject);
procedure QContasAfterScroll(DataSet: TDataSet);
procedure dbContasEnter(Sender: TObject);
procedure btnImpreClick(Sender: TObject);
procedure cbContasClick(Sender: TObject);
procedure dbPagarTodasKeyPress(Sender: TObject; var Key: Char);
procedure dbPagarTodasCellClick(Column: TColumn);
procedure dbPagarTodasEnter(Sender: TObject);
procedure dbContasDbClick(Sender: TObject);
private
  { Private declarations }
  vtop : longint;
  vCampo : TCheckBox;
  vPesquisarContas : boolean;
  vConNumer,
  vCodigo,
  vForNumer : Longint;
  vNomeTabela : String;
  procedure CarregaDados;
  procedure MontaQuery;
  function GravarDados : Boolean;
  procedure MostrarDbgPesqu(Sender : TObject);
  procedure LimparVariaveis;

  //Evento
  procedure clickCampo(Sender: TObject);

public
  { Public declarations }
end;

var
  TelCAPag: TTelCAPag;

implementation
Uses
  UProce, UDataM, URConP;

```

```
{SR *.DFM}
```

```
procedure TTelCaPag.clickCampo(Sender: TObject);  
begin
```

```
end;
```

```
procedure TTelCaPag.CarregaDados;  
begin  
  if vPesquisarContas then  
  begin  
    ceForne.Text := QContas.FieldByName('ForNomes').AsString;  
    vForNumer := QContas.FieldByName('ForNumer').AsInteger;  
    edNrDoc.Text := QContas.FieldByName('ConNrDoc').AsString;  
    deDtLan.Date := QContas.FieldByName('ConDtLan').AsDateTime;  
    deDtVto.Date := QContas.FieldByName('ConDtVen').AsDateTime;  
    curValor.Value := QContas.FieldByName('ConValor').AsCurrency;  
    meHisto.Lines.Text := QContas.FieldByName('ConHisto').AsString;  
    vConNumer := QContas.FieldByName('ConNumer').AsInteger;
```

```
  end;
```

```
end;
```

```
procedure TTelCaPag.MontaQuery;  
begin
```

```
  with QContas do  
  begin  
    Close;  
    Sql.Clear;  
    Sql.Add('Select F.ForNomes,F.ForNumer,C.ConNumer,C.ConDtLan,C.ConDtVen,' +  
            'C.ConNrDoc,C.ConValor,C.ConHisto,C.ConDtPag ' +  
            'From GrConPa C,GrForne F ' +  
            'Where C.ConForne= F.ForNumer ');  
    if cbContas.Checked then  
      Sql.Add('and c.ConDtPag is null ');  
    Sql.Add('Order By ConDtLan Desc ');
```

```
  Open;  
end;  
end;
```

```
function TTelCAPag.GravarDados : Boolean;  
begin  
  with dm.QInserir do  
  begin  
    Close;  
    Sql.Clear;  
    if vConNumer = 0 then  
    begin
```



```

vConNumer := BuscaNumero('GrConPa','ConNumer');
Sql.Add('Insert Into GrConPa ');

Sql.Add('(ConNumer,ConForne,ConNrDoc,ConDtLan,ConDtVen,ConValor,ConHisto,ConDtPag)
');
  Sql.Add(' Values ');
  Sql.Add('(:vNumer,:vForne,:vNrDoc,:vDtLan,:vDtVen,:vValor,:vHisto,:vDatas)');
  ParamByName('vDatas').DataType := ftDate;
  ParamByName('vDatas').Clear ;
end
else
begin
  Sql.Add('UpDate GrConPa');
  Sql.Add('Set ConForne = :vForne,ConNrDoc = :vNrDoc,ConDtLan = :vDtLan,');
  Sql.Add('ConDtVen = :vDtVen,ConValor = :vValor,ConHisto = :vHisto');
  Sql.Add('Where ConNumer = :vNumer ');
end;

ParamByName('vNumer').AsInteger := vConNumer;
ParamByName('vForne').AsInteger := vForNumer;
ParamByName('vNrDoc').AsString := edNrDoc.Text;
ParamByName('vDtLan').AsDateTime := deDtLan.Date;
ParamByName('vDtVen').AsDateTime := deDtVto.Date;
ParamByName('vValor').AsCurrency := CurValor.Value;
ParamByName('vHisto').AsString := meHisto.Text;

ExecSql;
end;

end;
procedure TTelCAPag.MostrarDbgPesqu(Sender : TObject);
begin
  dbgPesqu.Parent := TelCaPag;
  TelCAPag.KeyPreview := false;
  dbgPesqu.Top := (Sender as tComboEdit).Top + (Sender as tComboEdit).Height;
  dbgPesqu.Left := (Sender as tComboEdit).Left;
  dbgPesqu.Width := (Sender as tComboEdit).Width;
  dbgPesqu.DataSource := dsPesqu;
  dbgPesqu.Visible := true;

end;

procedure TTelCAPag.LimparVariaveis;
begin
  deDtLan.Date := Date;
  vForNumer := 0;
  ceForne.SetFocus;
  dbgPesqu.Visible := false;
  telCaPag.KeyPreview := true;
  vConNumer := 0;

```

```

deDtPag.Text := DateToStr(date);

end;

procedure TTelCAPag.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Action := caFree;
    TelCAPag := nil;
end;

procedure TTelCAPag.btnCanceClick(Sender: TObject);
begin
    LimparTela(TelCAPag);
    LimparVariaveis;
end;

procedure TTelCAPag.btnSaidaClick(Sender: TObject);
begin
    Close;
end;

procedure TTelCAPag.FormCreate(Sender: TObject);
begin
    deDtLan.Date := date;
end;

procedure TTelCAPag.ceNomesButtonClick(Sender: TObject);
begin
    if not dbgPesqu.Visible then
    begin
        MostrarDbgPesqu(Sender);
        dbgPesqu.Height := 197;
    end;
    ceForne.SetFocus;
    dbgPesqu.Parent := TelCaPag;

end;

procedure TTelCAPag.ceNomesKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if Key = vk_f3 then
        dbgPesqu.Visible := false
    else
        if Key <> vk_f4 then
            begin
                MostrarDbgPesqu(Sender);
                dbgPesqu.Height := 197;
                dbgPesqu.Visible := true;
            end;
        end;
end;

end;

```

```

procedure TTelCAPag.ceNomesKeyPress(Sender: TObject; var Key: Char);
begin
  if key = #0 then
    begin
      if dbgPesqu.Visible then
        dbgPesqu.SetFocus;
      end;
    end;
end;

```

```

procedure TTelCAPag.ceNomesKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if dbgPesqu.Visible then
    if ceForne.Text <> '' then
      tbPesqu.Locate('ForNomes',ceForne.Text,[loPartialKey]);

end;

```

```

procedure TTelCAPag.dbgPesquDbClick(Sender: TObject);
var
  vNomes : String;
begin
  vPesquisarContas := false;
  if vConNumer = 0 then
    begin
      LimparTela(TelCAPag);
      deDtLan.Date := date;
    end;
  dbgPesqu.Visible := false;
  telCAPag.KeyPreview := true;

  edNrDoc.SetFocus;
  btnPagar.Enabled := false;
  vForNumer := tbPesqu.FieldName('ForNumer').AsInteger;
  ceForne.Text := tbPesqu.FieldName('ForNomes').AsString;

end;

```

```

procedure TTelCAPag.dbgPesquKeyPress(Sender: TObject; var Key: Char);
begin
  if key = #13 then
    dbgPesquDbClick(Sender);

end;

```

```

procedure TTelCAPag.BtnGravaClick(Sender: TObject);
var
  vLocal : TBookMark;
begin

```

```

vLocal := QContas.GetBookmark;

if ceForne.Text = '' then
begin
    MessageDlg('Informe o Credor.',mtInformation,[mbOk],0);
    ceForne.SetFocus;
    exit;
end;

if ( (deDtLan.Text = ' / / ') or (deDtLan.Text = ' / / ') ) then
begin
    MessageDlg('Informe a data de Lançamento.',mtInformation,[mbOk],0);
    deDtLan.SetFocus;
    exit;
end;

if ( (deDtVto.Text = ' / / ') or (deDtVto.Text = ' / / ') ) then
begin
    MessageDlg('Informe a data do vencimento.',mtInformation,[mbOk],0);
    deDtVto.SetFocus;
    exit;
end;

if curValor.Value <= 0 then
begin
    MessageDlg('Informe o valor do pagamento.',mtInformation,[mbOk],0);
    curValor.SetFocus;
    exit;
end;

if not GravarDados then
    ShowMessage('Erro ao gravar tabela contas a pagar!');

LimparTela(TelCAPag);
LimparVariaveis;
MontaQuery;

QContas.GotoBookmark(vLocal);
end;

procedure TTelCAPag.FormKeyPress(Sender: TObject; var Key: Char);
begin
    TestaTecla(Self,Key);
end;

procedure TTelCAPag.btnExcluClick(Sender: TObject);
begin
    if vConNumer = 0 then
    begin
        MessageDlg('Clique no botão "Pesquisar" para selecionar uma conta para excluir.',mtInformation,[mbOk],0);
        exit;
    end;
end;

```

```

end;
if MessageDlg('Confirma a exclusão da conta a pagar?',mtInformation,[mbYes,mbNo],0) =
mrYes then
begin
if DeletarDados('GrConPa','ConNumer',vConNumer) then
begin
MontaQuery;
LimparTela(TelCAPag);
LimparVariaveis;
end
else
ShowMessage('Erro na exclusão do registro');
end;

```

```

end;

```

```

procedure TTelCAPag.cbHistoKeyPress(Sender: TObject; var Key: Char);
begin
key := #0;
end;

```

```

procedure TTelCAPag.curValorKeyPress(Sender: TObject; var Key: Char);
begin
if key = '.' then
key := ',';

```

```

end;

```

```

procedure TTelCAPag.FormShow(Sender: TObject);
begin
TrazFiguras(TelCAPag);
MontaQuery;
CarregaFiguraBitBtn(btnPagar,'BTNGRAVA');

```

```

deDtPag.Text := DateToStr(date);

```

```

end;

```

```

procedure TTelCAPag.ceNomesEnter(Sender: TObject);
begin
tbPesqu.close;
tbPesqu.TableName := 'GrForne';
tbPesqu.IndexName := 'Nomes';
tbPesqu.Open;
dsPesqu.DataSet := tbPesqu;
dbgPesqu.Parent := PnDados;
dbgPesqu.Columns.Clear;
dbgPesqu.Columns.Add.FieldName := 'ForNumer';
dbgPesqu.Columns[0].Title.Caption := 'Código';
dbgPesqu.Columns.Add.FieldName := 'ForNomes';
dbgPesqu.Columns[1].Title.Caption := 'Nome';

```

```

end;

procedure TTelCAPag.edPesquKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if edPesqu.Text <> '' then
    QContas.Locate('ForNomes',EdPesqu.Text,[LoPartialKey]);
end;

procedure TTelCAPag.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if key = vk_F4 then
    edPesqu.SetFocus;

end;

procedure TTelCAPag.edPesquKeyPress(Sender: TObject; var Key: Char);
begin
  if key = #13 then
    dbContas.SetFocus

end;

procedure TTelCAPag.edPNrDoKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if edPNrDo.Text <> '' then
    QContas.Locate('ConNrDoc',EdPNrDo.Text,[LoPartialKey])

end;

procedure TTelCAPag.dbContasCellClick(Column: TColumn);
begin
  vPesquisarContas := true;
  CarregaDados;
  btnPagar.Enabled := true;

end;

procedure TTelCAPag.dbContasKeyPress(Sender: TObject; var Key: Char);
begin
  if key = #13 then
    begin
      vPesquisarContas := true;
      CarregaDados;
      btnPagar.Enabled := true;
    end;

end;

```

```

procedure TTelCAPag.btnPagarClick(Sender: TObject);
var
  vNumero : Longint;
begin
  vCodigo := QContas.fieldByName('ConNumer').AsInteger;
  with dm.QInserir do
    begin
      if vConNumer <> 0 then
        begin
          Close;
          Sql.Clear;
          Sql.Add('UpDate GrConPa');
          Sql.Add('Set ConDtPag = :vDatas');
          Sql.Add('Where ConNumer = :vNumer ');

          ParamByName('vNumer').AsInteger := vConNumer;
          try
            ParamByName('vDatas').AsDateTime := StrToDate(deDtPag.Text);
          except
            ParamByName('vDatas').AsDateTime := Date;
          end;
          ExecSql;
          if sender = btnPagar then
            MontaQuery;

          QContas.Locate('ConNumer',vCodigo,[LoPartialKey]);
          dbContas.SetFocus;
        end;
      end;
    end;

procedure TTelCAPag.QContasAfterScroll(DataSet: TDataSet);
begin
  CarregaDados;
end;

procedure TTelCAPag.dbContasEnter(Sender: TObject);
begin
  vPesquisarContas := true;
end;

procedure TTelCAPag.btnImpreClick(Sender: TObject);
begin
  if TelRConP = nil then
    TelRConP := TTelRConP.Create(Self);

    TelRConP.Show;
  end;

procedure TTelCAPag.cbContasClick(Sender: TObject);
begin
  MontaQuery;

```

end;

```
procedure TTelCAPag.dbPagarTodasKeyPress(Sender: TObject; var Key: Char);
begin
  if key = #13 then
  begin
    vPesquisarContas := true;
    CarregaDados;
    btnPagar.Enabled := true;
  end;
end;
```

```
procedure TTelCAPag.dbPagarTodasCellClick(Column: TColumn);
begin
  vPesquisarContas := true;
  CarregaDados;
  btnPagar.Enabled := true;
end;
```

end;

```
procedure TTelCAPag.dbPagarTodasEnter(Sender: TObject);
begin
  vPesquisarContas := true;
end;
```

```
procedure TTelCAPag.dbContasDbClick(Sender: TObject);
begin
  if vPagarContasComDuploCliqueDoMouse then
    btnPagarClick(btnPagar);
end;
```

XX

Menu Principal

Controle de Cheques

Cadastro

Controle Cheques

Fechamento

Caixa

Relatórios

Cálculo Juros

Manutenção

Sobre

Fechar

Cheques Pré Recebidos Para Hoje: 16/4/2013

Número	Banco	Dt Emissão	Valor	Recebido de	Repassado
12522	BRASIL	16/3/2013	526,80	ADRIANA DA SILVA	SIM

Cheques Pré Emitidos Para Hoje: 16/4/2013

Número	Banco	Dt Emissão	Valor	Emitido para
4545522	BRASIL	16/3/2013	5.630,90	J&J - ATACADISTA DE PRODUTOS ALIMENTÍCIOS S/A

unit GerMenus;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
Menus, StdCtrls, Buttons, Db, Grids, DBGrids, DBTables, ExtCtrls,
ComCtrls;

type

TTelMenus = class(TForm)
 MainMenu1: TMainMenu;
 Cadastro1: TMenuItem;
 Fornecedores1: TMenuItem;
 ControleCheques1: TMenuItem;
 ChequesRecebidos1: TMenuItem;
 ChequesEmitidos1: TMenuItem;
 BaixadeCheques1: TMenuItem;
 Relatrios1: TMenuItem;
 ChequesRecebidos2: TMenuItem;
 Terceiros1: TMenuItem;
 Sada1: TMenuItem;
 Sobre1: TMenuItem;
 Bancos1: TMenuItem;
 FechamentoMs1: TMenuItem;
 FechamentoMs2: TMenuItem;
 FechamentoMs3: TMenuItem;
 Endereos1: TMenuItem;
 ListaParaBanco1: TMenuItem;
 GroupBox1: TGroupBox;
 GroupBox2: TGroupBox;
 PopupMenu1: TPopupMenu;
 EsconderChequesPreRecebidos1: TMenuItem;
 EsconderChequePrRecebidos1: TMenuItem;
 tbTempo: TTable;
 Caixa1: TMenuItem;
 ControleCaderno1: TMenuItem;
 LanamentosCaderno1: TMenuItem;
 ContasaPagar1: TMenuItem;
 ContasAPagar2: TMenuItem;
 ContasPagar1: TMenuItem;
 Panel1: TPanel;
 Edit1: TEdit;
 ChequesSemFundo1: TMenuItem;
 Chequesrecebidosdeterceiros1: TMenuItem;
 Manuteno1: TMenuItem;
 TabeladeChequesRecebidos1: TMenuItem;
 ChequesRecebidos3: TMenuItem;
 SQL1: TMenuItem;
 ClculoJuros1: TMenuItem;
 Juros1: TMenuItem;

```

pnMensa: TPanel;
sgCheRe: TStringGrid;
sgCheEm: TStringGrid;
Configurao1: TMenuItem;
Relatriodechequesparacobrana1: TMenuItem;
Empresa1: TMenuItem;
CorrigirTroco1: TMenuItem;
ConfigurarTelas1: TMenuItem;
Panel2: TPanel;
SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
SpeedButton4: TSpeedButton;
SpeedButton5: TSpeedButton;
Geraarquivoparaconsultanoscaixas: TMenuItem;
LimiteCliente: TMenuItem;
GerararquivoparaconsultadeSaldodeClientes: TMenuItem;
buCadastro: TButton;
buFechar: TButton;

procedure FormCreate(Sender: TObject);
procedure Fornecedores1Click(Sender: TObject);
procedure Sada1Click(Sender: TObject);
procedure Sobre1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Bancos1Click(Sender: TObject);
procedure ChequesEmitidos1Click(Sender: TObject);
procedure ChequesRecebidos1Click(Sender: TObject);
procedure BaixadeCheques1Click(Sender: TObject);
procedure Terceiros1Click(Sender: TObject);
procedure FechamentoMs2Click(Sender: TObject);
procedure ChequesRecebidos2Click(Sender: TObject);
procedure FechamentoMs3Click(Sender: TObject);
procedure Endereos1Click(Sender: TObject);
procedure ListaParaBanco1Click(Sender: TObject);
procedure EsconderChequesPreRecebidos1Click(Sender: TObject);
procedure EsconderChequePrRecebidos1Click(Sender: TObject);
procedure ControleCaderno1Click(Sender: TObject);
procedure LanamentosCaderno1Click(Sender: TObject);
procedure ContasAPagar2Click(Sender: TObject);
procedure ContasPagar1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure ChequesSemFundo1Click(Sender: TObject);
procedure Chequesrecebidosdeterceiros1Click(Sender: TObject);
procedure TabelaDeChequesRecebidos1Click(Sender: TObject);
procedure ChequesRecebidos3Click(Sender: TObject);
procedure SQL1Click(Sender: TObject);
procedure Juros1Click(Sender: TObject);
procedure Configurao1Click(Sender: TObject);
procedure Relatriodechequesparacobrana1Click(Sender: TObject);
procedure Empresa1Click(Sender: TObject);
procedure CorrigirTroco1Click(Sender: TObject);

```

```

procedure ConfigurarTelas1Click(Sender: TObject);
procedure GeraarquivoparaconsultanoscaixasClick(Sender: TObject);
procedure LimiteClienteClick(Sender: TObject);
procedure GerararquivoparaconsultadeSaldodeClientesClick(
  Sender: TObject);
procedure FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure buCadastroClick(Sender: TObject);
procedure buFecharClick(Sender: TObject);
private
  { Private declarations }

procedure criarCampoSituacaoTerceiro;
procedure criarCampoSituacaoChequeGrChSFun;

function verificaAcesso : boolean;
procedure gravarArquivoLimiteSaldoTerceiro;
procedure pBaixarChequesAutomaticamente;
procedure criarCampoTrocoCaderno;
procedure criarCampoLimiteTerceiro;
procedure gravarArquivoLimiteSaldoCliente;
// procedure CriarTabelaCor;
procedure CriarCampoTabelaFornecedor;
procedure CriarTabelaEmpresa;
procedure CriarTabelaControleChequeSemFundo;
procedure CriarTabelaChequeEnviadosParaCobranca;
procedure pCriarTabelaControle;
procedure statusBar;
function fGravarArquivoTxt : boolean;
function fCriarArquivoTexto : boolean;
procedure pCriarArquivoConsultaCPFCliente;
function CriarTabelaTerceiros: Boolean;
function CriarTabelaParaContasAPagar: Boolean;
function CriarCampoControle: Boolean;
function CriarTabelaParaControleCaderno: Boolean;
function CriarTabelaParaControleDeTroco: Boolean;
procedure CriarCampoDataChequeRepasadosParaTerceiros;
procedure CriarIndiceTabelaChequeReceber;
procedure CriarCampoDespesaTabelaChequesRecebidos;
public
  { Public declarations }
end;

var
  TelMenus: TTelMenus;
  ArqLimite,
  ArqUsuario,
  ArqTexto : Text;

implementation
Uses

```

UProce,UConec, USobre, UForne, UDataM, UCriaT, UBanco, UCheEm,
 UCheRe, UBChcq, UTerce, UFecha, URCher, URFech, UENDER,
 ULista, FrmContr, URCade, UCApag, URConP, URSemF,
 URChequeRecebidosTerceiros, UManutencaoChequesRecebidos, URCheque,
 UEntra, URSQL, UJuros, UConfi, URelCheqCobranca, UEmpresa,
 UCorrigeTroco, UConfiTela, ULimiteCliente;

{ \$R *.DFM }

```

procedure TTelMenus.criarCampoSituacaoTerceiro;
var
  vCampo : boolean;
begin
  try
    with tbTempo do
      begin
        Close;
        TableName := 'GrTerce.db';
        Open;
        vCampo := FieldByName('TerSitu').AsBoolean;
        Close;
      end;
    except
      with Dm.QCriaT do
        begin
          try
            tbTempo.Close;
            Close;
            Sql.Clear;
            Sql.Add('Alter Table GrTerce ');
            Sql.Add('Add TerSitu boolean');
            ExecSql;

            Close;
            Sql.Clear;
            Sql.Add('UpDate GrTerce ');
            Sql.Add('set TerSitu = false');
            ExecSql;

          except
            on E:Exception do
              begin
                ShowMessage('Erro ao criar campo de situação do terceiro. ' + e.message);
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;
end;

```

```

procedure TTelMenus.criarCampoSituacaoChequeGrChSFun;
var
  vCampo : String;
begin
  try
    with tbTempo do
      begin
        Close;
        TableName := 'GrChSFun.db';
        Open;
        vCampo := FieldByName('ConSitua').AsString;
        Close;
      end;
  except
    with Dm.QCriaT do
      begin
        try
          tbTempo.Close;
          Close;
          Sql.Clear;
          Sql.Add('Alter Table GrChSFun ');
          Sql.Add('Add ConSitua varchar(15)');
          ExecSql;

          dm.QConsulta.Close;
          dm.QConsulta.sql.clear;
          dm.QConsulta.SQL.Add('Select * from grChSFun');
          dm.QConsulta.open;

          dm.QCriaT.close;
          dm.QCriaT.SQL.Clear;
          dm.QCriaT.SQL.Add('UpDate grChSFun ');
          dm.QCriaT.SQL.Add('Set ConSitua = :pSitua ');
          dm.QCriaT.SQL.Add('Where ConClien = :pClien ');
          dm.QCriaT.SQL.Add('And ConNrChe = :pCheque ');
          dm.QCriaT.Prepare;

          while not dm.QConsulta.EOF do
            begin
              dm.QInserir.close;
              dm.QInserir.SQL.Clear;
              dm.QInserir.SQL.Add('Select * from grCheRe ');
              dm.QInserir.SQL.Add('Where CheClien = :pCliente ');
              dm.QInserir.SQL.Add('And CheNrChe = :pNrCheque ');
              dm.QInserir.ParamByName('pCliente').AsInteger :=
dm.QConsulta.FieldByName('ConClien').AsInteger;
              dm.QInserir.ParamByName('pNrCheque').AsString :=
dm.QConsulta.FieldByName('ConNrChe').AsString;
              dm.QInserir.Open;

```

```

        if dm.QInserir.RecordCount > 0 then
        begin
            if dm.QInserir.FieldName('CheFundo').AsBoolean then
            begin
                dm.QCriaT.ParamByName('pSitua').AsString := 'Sem Fundo';
                dm.QCriaT.ParamByName('pClien').AsInteger :=
dm.QConsulta.FieldName('ConClien').AsInteger;
                dm.QCriaT.ParamByName('pCheque').AsString :=
dm.QConsulta.FieldName('ConNrChe').AsString;
                DM.QCriaT.ExecSQL;
            end
            else
            begin
                dm.QCriaT.ParamByName('pSitua').AsString := 'PAGO';
                dm.QCriaT.ParamByName('pClien').AsInteger :=
dm.QConsulta.FieldName('ConClien').AsInteger;
                dm.QCriaT.ParamByName('pCheque').AsString :=
dm.QConsulta.FieldName('ConNrChe').AsString;
                DM.QCriaT.ExecSQL;
            end;
        end;
        statusBar;
        dm.QConsulta.Next;
    end;

except
    on E:Exception do
    begin
        ShowMessage('Erro ao criar campo de situação do cheque GrChSFun. ' + e.message);
    end;
end;
end;
end;

end;

function TTelMenus.verificaAcesso : boolean;
begin

    vUsuario := 0;

    try
        AssignFile(ArqUsuario,'C:\GmvChequ\Programa\Usuario.txt');
    //  {$I-}
        Append(ArqUsuario);
    //  {$I+}

    cadastro1.Visible := false;
    ControleCheques1.Visible := false;
    FechamentoMs1.Visible := false;

```

```

Caixa1.Visible := false;
Relatrios1.Visible := false;
ClculoJuros1.Visible := false;
Manuteno1.Visible := false;
FechamentoMs1.Visible := false;
sobre1.Visible := false;
sada1.Visible := false;
vUsuario := 1;
buCadastro.Visible := true;
buFechar.Visible := true;
buCadastro.Parent := TelMenus;
buFechar.Parent := TelMenus;
buFechar.Top := buCadastro.top + buCadastro.Height + 40;
GroupBox2.Visible := false;
GroupBox1.Visible := false;

    result := true;
except
    result := false;
end;

end;

procedure TTelMenus.gravarArquivoLimiteSaldoTerceiro;
var
    vDocumento,vCliente : String;
    vLimiteCliente : Currency;
    vGravou : boolean;

begin

    dm.QPesquisa.close;
    dm.QPesquisa.SQL.Clear;
    dm.QPesquisa.SQL.add(' Select terNrCpf,Sum(CheValor) CheValor,TerNomes,TerLimit,');
    dm.QPesquisa.SQL.add('TerLimit - Sum(CheValor) Saldo_Disponível,CheMesCo,CheBaixa ');
    dm.QPesquisa.SQL.Add(' from grCheRe,grTerce');
    dm.QPesquisa.SQL.add('Where CheTerce = TerNumer');

    dm.QPesquisa.SQL.add('And TerLimit > 0');
    dm.QPesquisa.SQL.add('And CheBaixa = false');

    dm.QPesquisa.SQL.add('Group By TerNrCpf,cheMesCo,TerNomes,TerLimit,CheBaixa');
    dm.QPesquisa.SQL.add('Order by terNrCpf,TerNomes');

    dm.QPesquisa.open;

    AssignFile(ArqLimite,vDiretorio + '\Limite.txt');
    {$I-}
    Append(ArqLimite);
    {$I+}
    // Gravar arquivo txt;

```

```

if FEntra = nil then
    FEntra := TFEntra.Create(Self);

FEntra.Show;

FEntra.Label4.Visible := true;
FEntra.Refresh;

try
    While not dm.QPesquisa.EOF do
        begin

            if (dm.QPesquisa.FieldName('TerNrCpf').AsString <> '') then
                begin
                    statusBar;

                    write(ArqLimite,dm.QPesquisa.FieldName('TerNrCpf').AsString + ';');
                    write(ArqLimite,dm.QPesquisa.FieldName('TerNomes').AsString + ';');
                    write(ArqLimite,
FormatFloat('##,##0.00',dm.QPesquisa.FieldName('TerLimit').AsCurrency) + ';');
                    write(ArqLimite, dm.QPesquisa.FieldName('CheMesCo').AsString + ';');
                    write(ArqLimite,
FormatFloat('##,##0.00',dm.QPesquisa.FieldName('CheValor').AsCurrency) + ';');

writeLn(ArqLimite,FormatFloat('##,##0.00',dm.QPesquisa.FieldName('Saldo_Disponível').As
Currency));
                    vGravou := true;

                end;
                vDocumento := dm.QPesquisa.FieldName('TerNrCpf').AsString;
                vCliente := dm.QPesquisa.FieldName('TerNomes').AsString;
                vLimiteCliente := dm.QPesquisa.FieldName('TerLimit').AsCurrency;

                dm.QPesquisa.Next;

                if vDocumento <> dm.QPesquisa.FieldName('TerNrCpf').AsString then
                    begin
                        if not vGravou then
                            begin
                                write(ArqLimite,vDocumento + ';');
                                write(ArqLimite,vCliente + ';');
                                write(ArqLimite,FormatFloat('##,##0.00',vLimiteCliente) + ';');
                                write(ArqLimite,' 0000 ' + ';');
                                write(ArqLimite,' 0000 ' + ';');
                                writeLn(ArqLimite, FormatFloat('##,##0.00',vLimiteCliente));
                            end;
                            vGravou := false;
                        end;
                    end;

                end;
            end;

```



```

Finally
  // gravar o ultimo registro;

  if not vGravou then
  begin
    write(ArqLimite,vDocumento + ';');
    write(ArqLimite,vCliente + ';');
    write(ArqLimite, FormatFloat('##,##0.00',vLimiteCliente) + ';');
    write(ArqLimite,' 0000 ' + ';');
    write(ArqLimite,' 0000 ' + ';');
    writeLn(ArqLimite, FormatFloat('##,##0.00',vLimiteCliente));
  end;

  closeFile(ArqLimite);
end;

end;

procedure TTelMenus.pBaixarChequesAutomaticamente;
begin
  try
    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('Update grCheRe ');
    dm.QInserir.SQL.Add('set CheBaixa = true ');
    dm.QInserir.SQL.Add('Where CheDtDes < :vData ');
    dm.QInserir.ParamByName('vData').AsDateTime := Date;
    dm.QInserir.ExecSQL;
  except
  end;
end;

procedure TTelMenus.criarCampoTrocoCaderno;
var
  vCampo : Real;
begin
  try
    with tbTempo do
    begin
      Close;
      TableName := 'GrCader.db';
      Open;
      vCampo := FieldByName('CadTroco').AsInteger;
      Close;
    end;
  except
    with Dm.QCriaT do
    begin
      try
        tbTempo.Close;

```

```

        Close;
        Sql.Clear;
        Sql.Add('Alter Table GrCader ');
        Sql.Add('Add CadTroco Integer');
        ExecSql;

    except
        ShowMessage('Erro ao criar campo de Troco na tabela do Caderno');
    end;
end;
end;

end;

procedure TTelMenus.criarCampoLimiteTerceiro;
var
    vCampo : Real;
begin
    try
        with tbTempo do
            begin
                Close;
                TableName := 'GrTerce.db';
                Open;
                vCampo := FieldByName('TerLimit').AsFloat;
                Close;
            end;
    except
        with Dm.QCriaT do
            begin
                try
                    tbTempo.Close;
                    Close;
                    Sql.Clear;
                    Sql.Add('Alter Table GrTerce ');
                    Sql.Add('Add TerLimit numeric(10,2)');
                    ExecSql;

                    Close;
                    Sql.Clear;
                    Sql.Add('UpDate GrTerce ');
                    Sql.Add('set TerLimit = 0 ');
                    ExecSql;

                except
                    ShowMessage('Erro ao criar campo de limite de compra na tabela de terceiros');
                end;
            end;
    end;
end;

```

```

end;

end;

procedure TTelMenus.gravarArquivoLimiteSaldoCliente;
var
  vDocumento,vCliente : String;
  vLimiteCliente : Currency;
  vGravou : boolean;

begin

  dm.QPesquisa.close;
  dm.QPesquisa.SQL.Clear;
  dm.QPesquisa.SQL.add(' Select CheNrDoc,Sum(CheValor)
CheValor,CliNomes,CliNrDoc,CliLimite,');
  dm.QPesquisa.SQL.add('CliLimite - Sum(CheValor) Saldo_Disponível,CheMesCo,CheBaixa ');
  dm.QPesquisa.SQL.Add(' from grCheRe,grClien');
  dm.QPesquisa.SQL.add('Where CheNrDoc = CliNrDoc');

  dm.QPesquisa.SQL.add('And CliLimite > 0');
  dm.QPesquisa.SQL.add('And CheBaixa = false');

  dm.QPesquisa.SQL.add('Group By CheNrDoc,CliNomes,CliLimite,cheMesCo, CheBaixa,
CliNrDoc');
  dm.QPesquisa.SQL.add('Order by CheNrDoc, CliNomes');

  dm.QPesquisa.open;

  AssignFile(ArqLimite,vDiretorio + '\Limite.txt');
  {$I-}
  Rewrite(ArqLimite);
  {$I+}
  // Gravar arquivo txt;

  if FEntra = nil then
    FEntra := TFEntra.Create(Self);

  FEntra.Show;

  FEntra.Label4.Visible := true;
  FEntra.Refresh;

try
  While not dm.QPesquisa.EOF do
    begin

      if (dm.QPesquisa.FieldByName('CheNrDoc').AsString <> '') then
        begin

```

```

statusBar;

write(ArqLimite,dm.QPesquisa.FieldByName('CliNrDoc').AsString + ');');
write(ArqLimite,dm.QPesquisa.FieldByName('CliNomes').AsString + ');');
write(ArqLimite,
FormatFloat('##,##0.00',dm.QPesquisa.FieldByName('CliLimite').AsCurrency) + ');');
write(ArqLimite, dm.QPesquisa.FieldByName('CheMesCo').AsString + ');');
write(ArqLimite,
FormatFloat('##,##0.00',dm.QPesquisa.FieldByName('CheValor').AsCurrency) + ');');
writeln(ArqLimite,
FormatFloat('##,##0.00',dm.QPesquisa.FieldByName('Saldo_Disponível').AsCurrency));
vGravou := true;

end;
vDocumento := dm.QPesquisa.FieldByName('CheNrDoc').AsString;
vCliente := dm.QPesquisa.FieldByName('CliNomes').AsString;
vLimiteCliente := dm.QPesquisa.FieldByName('CliLimite').AsCurrency;

dm.QPesquisa.Next;

if vDocumento <> dm.QPesquisa.FieldByName('CheNrDoc').AsString then
begin
if not vGravou then
begin
write(ArqLimite,vDocumento + ');');
write(ArqLimite,vCliente + ');');
write(ArqLimite, FormatFloat('##,##0.00',vLimiteCliente) + ');');
write(ArqLimite, ' 0000 ' + ');');
write(ArqLimite, ' 0000 ' + ');');
writeln(ArqLimite, FormatFloat('##,##0.00',vLimiteCliente));
end;
vGravou := false;
end;

end;
Finally
// gravar o ultimo registro;

if not vGravou then
begin
write(ArqLimite,vDocumento + ');');
write(ArqLimite,vCliente + ');');
write(ArqLimite,FormatFloat('##,##0.00',vLimiteCliente) + ');');
write(ArqLimite, ' 0000 ' + ');');
write(ArqLimite, ' 0000 ' + ');');
writeln(ArqLimite, FormatFloat('##,##0.00',vLimiteCliente));
end;

closeFile(ArqLimite);
end;

```

end;

procedure TTelMenus.CriarCampoTabelaFornecedor;

var

vDia,vMes,vAno : word;

vStr,

vCampo : String;

vLimite : Currency;

i : integer;

begin

try

with tbTempo do

begin

Close;

TableName := 'GrClien.db';

Open;

vLimite := FieldByName('CliLimite').AsCurrency;

Close;

end;

except

with Dm.QCriaT do

begin

try

statusBar;

tbTempo.Close;

Close;

Sql.Clear;

Sql.Add('Alter Table GrClien ');

Sql.Add('Add CliLimite money');

ExecSql;

except

end;

end;

end;

try

with tbTempo do

begin

Close;

TableName := 'GrCheRe.db';

Open;

vCampo := FieldByName('CheMesCo').AsString;

Close;

end;

except

with Dm.QCriaT do

```

begin
try
  tbTempo.Close;
  Sql.Clear;
  Sql.Add('Alter Table GrCheRe ');
  Sql.Add('Add CheMesCo varchar(7)');
  ExecSql;
  Close;
  tbTempo.Open;
  While not tbTempo.EOF do
  begin
    statusBar;
    vstr := FormatDateTime('dd/MM/yyyy',tbTempo.fieldByName('CheDtDes').AsDateTime);

    tbTempo.Edit;
    i := pos('/',vStr);
    tbTempo.fieldByName('CheMesCo').AsString := Copy(vStr,i + 1,Length(vStr));
    tbTempo.Post;
    tbTempo.Next;
  end;

except
end;
end;
end;

try
with tbTempo do
begin
  Close;
  TableName := 'GrCheEm.db';
  Open;
  vCampo := FieldByName('CheNrNot').AsString;
  Close;
end;
except
with Dm.QCriaT do
begin
  try
    tbTempo.Close;
    Close;
    Sql.Clear;
    Sql.Add('Alter Table GrCheEm ');
    Sql.Add('Add CheNrNot varchar(15)');
    ExecSql;

    Close;
    Sql.Clear;
    Sql.Add('Alter Table GrCheEm ');
    Sql.Add('Add CheForne Integer');
    ExecSql;

```

```

        Close;
        Sql.Clear;
        Sql.Add('UpDate GrCheEm ');
        Sql.Add('set CheForne = 0 ');
        ExecSql;

    except
        ShowMessage('Erro ao criar campo na tabela de cheques recebidos para nr. nota fiscal');
    end;
end;
end;
end;

procedure TTelMenus.CriarTabelaEmpresa;
begin
    try
        dm.QInserir.Close;
        dm.QInserir.SQL.Clear;
        dm.QInserir.SQL.Add('Select * from grEmpre');
        dm.QInserir.Open;
        dm.QInserir.Close;
    except
        dm.QInserir.Close;
        dm.QInserir.SQL.Clear;
        dm.QInserir.SQL.Add('Create Table grEmpre'); //CheNumer = chave primaria tabela grCheRe
        dm.QInserir.SQL.Add('(EmpNumer integer, EmpNomes varchar(100),');
        dm.QInserir.SQL.Add('EmpFones varchar(14),EmpFax varchar(14),EmpCnpj
varchar(18),EmpInsEs varchar(11),');
        dm.QInserir.SQL.Add('EmpMunic integer,EmpENDER integer,EmpNrCasa varchar(6),');
        dm.QInserir.SQL.Add('primary key (EmpNumer))');
        dm.QInserir.ExecSql;

    end;

end;

procedure TTelMenus.CriarTabelaChequeEnviadosParaCobranca;
begin
    try
        dm.QInserir.Close;
        dm.QInserir.SQL.Clear;
        dm.QInserir.SQL.Add('Select * from grChCob');
        dm.QInserir.Open;
        dm.QInserir.Close;
    except
        dm.QInserir.Close;
        dm.QInserir.SQL.Clear;
        dm.QInserir.SQL.Add('Create Table grChCob'); //CheNumer = chave primaria tabela grCheRe
        dm.QInserir.SQL.Add('(CheNumer integer, CheData Date,');

```

```

dm.QInserir.SQL.Add('primary key (CheNumer))');
dm.QInserir.ExecSql;

dm.QInserir.Close;
dm.QInserir.SQL.Clear;
dm.QInserir.SQL.Add('Alter Table grCheRe'); //CheNumer = chave primaria tabela grCheRe
dm.QInserir.SQL.Add('Add CheCobra boolean');
dm.QInserir.ExecSql;

dm.QInserir.Close;
dm.QInserir.SQL.Clear;
dm.QInserir.SQL.Add('UpDate grCheRe'); //CheNumer = chave primaria tabela grCheRe
dm.QInserir.SQL.Add('Set CheCobra = false');
dm.QInserir.ExecSql;

end;

end;

procedure TTelMenus.CriarTabelaControleChequeSemFundo;
var
    vNumero : Longint;
begin

    Edit1.Visible := false;
    try
        dm.QInserir.Close;
        dm.QInserir.SQL.Clear;
        dm.QInserir.SQL.Add('Select * from grChSFun');
        dm.QInserir.Open;
        dm.QInserir.Close;
    except
        dm.QInserir.Close;
        dm.QInserir.SQL.Clear;
        dm.QInserir.SQL.Add('Create Table grChSFun');
        dm.QInserir.SQL.Add('(ConCodig integer, ConDatas Date, ConClien integer,');
        dm.QInserir.SQL.Add('ConChequ integer, ConValor Money, ConTerce integer, ConNrChe
varchar(15),');
        dm.QInserir.SQL.Add('primary key (ConCodig))');
        dm.QInserir.ExecSql;

        dm.QPesquisa.Close;
        dm.QPesquisa.SQL.Clear;
        dm.QPesquisa.SQL.Add('Select * from grCheRe ');
        dm.QPesquisa.SQL.Add('where CheFundo = true');
        dm.QPesquisa.Open;
        vNumero := 1;
        statusBar;
        Panel1.Visible := true;
        FEntra.Label2.Caption := 'Atualizando tabela';
    end;
end;

```



```

FEntra.Refresh;

while not Dm.QPesquisa.EOF do
begin
    statusBar;

    dm.QInserir.Close;
    dm.QInserir.Sql.Clear;
    dm.QInserir.Sql.Add('Insert into grChSFun ');
    dm.QInserir.Sql.Add('(ConCodig,ConDatas,ConClien,ConChequ,
ConValor,ConTerce,ConNrChe)');
    dm.QInserir.Sql.Add('Values');
    dm.QInserir.Sql.Add('(:p1,:p2,:p3,:p4,:p5,:p6,:p7)');
    dm.QInserir.ParamByName('p1').AsInteger := vNumero;
    dm.QInserir.ParamByName('p2').AsDateTime := Date;
    dm.QInserir.ParamByName('p3').AsInteger :=
dm.QPesquisa.FieldByName('CheClien').AsInteger;
    dm.QInserir.ParamByName('p4').AsInteger :=
dm.QPesquisa.FieldByName('CheNumer').AsInteger;
    dm.QInserir.ParamByName('p5').AsCurrency :=
dm.QPesquisa.FieldByName('CheValor').AsCurrency;
    dm.QInserir.ParamByName('p6').AsInteger :=
dm.QPesquisa.FieldByName('CheTerce').AsInteger;
    dm.QInserir.ParamByName('p7').AsString :=
dm.QPesquisa.FieldByName('CheNrChe').AsString;
    dm.QInserir.ExecSql;
    inc(vNumero);
    Dm.QPesquisa.Next;

end;

end;

Panel1.Visible := false;
end;

procedure TTelMenus.pCriarTabelaControle;
begin
    try
        dm.QInserir.Close;
        dm.QInserir.SQL.Clear;
        dm.QInserir.SQL.Add('Select * from grControle');
        dm.QInserir.Open;
        dm.QInserir.Close;
    except
        dm.QInserir.Close;
        dm.QInserir.SQL.Clear;
        dm.QInserir.SQL.Add('Create Table grControle');
        dm.QInserir.SQL.Add('(ConCodig integer, ConDatas Date,');
        dm.QInserir.SQL.Add('primary key (ConCodig))');
        dm.QInserir.ExecSql;
    end;
end;

```

```

dm.QInserir.Close;
dm.QInserir.SQL.Clear;
dm.QInserir.SQL.Add('Insert Into grControle');
dm.QInserir.SQL.Add('(ConCodig,ConDatas)');
dm.QInserir.SQL.Add('values');
dm.QInserir.SQL.Add('(1,:p1)');
dm.QInserir.ParamByName('p1').AsDateTime := date;
dm.QInserir.ExecSql;
end;

end;

procedure TTelMenus.statusBar;
begin
  FEntra.ProgressBar1.Position := FEntra.ProgressBar1.Position + 1;
  if FEntra.ProgressBar1.Position = 100 then
    FEntra.ProgressBar1.Position := 0;

end;

function TTelMenus.fGravarArquivoTxt : boolean;
var
  conta : integer;
begin

  conta := 0;

  FEntra.Label4.Visible := true;
  FEntra.Refresh;

  with dm.QPesquisa do
  begin
    close;
    sql.Clear;
    sql.add('Select CliNumer, CliNrDoc,CliNomes,CliEnder from grClien');
    sql.add('Order by CliNomes');

    Open;

    if vMostraMensagemUsuarioRestricoes then
    begin
      While not eof do
      begin
        write(ArqTexto,FieldByName('CliNrDoc').AsString + ';');
        write(ArqTexto,FieldByName('CliNomes').AsString + ';');

        statusBar;
        dm.QConsulta.Close;
        dm.QConsulta.SQL.Clear;
        dm.QConsulta.SQL.Add('Select CheFundo,CheClien from GrCheRe where CheClien = ' +
          IntToStr(dm.QPesquisa.FieldByName('CliNumer').AsInteger));
        dm.QConsulta.SQL.Add('And CheFundo = true');

```

```

dm.QConsulta.Open;
if dm.QConsulta.FieldName('CheClien').AsInteger > 0 then
begin

    if dm.QConsulta.FieldName('CheFundo').AsBoolean then
        write(ArqTexto,'1;') //Cliente inadimplente
    else
        write(ArqTexto,'0;'); // Cliente ok
    end
else
    write(ArqTexto,'0;');

if not (vBloquearClienteSemCompraSeisMeses) then
begin
    if FieldByName('CliEnder').AsInteger > 0 then
        writeLn(ArqTexto,'1') // 1 = tem endereço

    else
        writeLn(ArqTexto,'0'); // 0 = nao tem endereço

    end
else
begin
    if FieldByName('CliEnder').AsInteger > 0 then
        write(ArqTexto,'1;') // 1 = tem endereço
    else
        write(ArqTexto,'0;');
    end;

// bloquear cliente com mais de 6 meses sem comprar - inicio
if vBloquearClienteSemCompraSeisMeses then
begin
    dm.QConsulta.Close;
    dm.QConsulta.SQL.Clear;
    dm.QConsulta.SQL.Add('Select CheClien from GrCheRe where CheClien = ' +
        IntToStr(dm.QPesquisa.FieldName('CliNumer').AsInteger));
    dm.QConsulta.SQL.Add('And CheDtEmi >= :p1');
    dm.QConsulta.ParamByName('p1').AsDateTime := Date - 183;
    dm.QConsulta.Open;

    // Cliente sem comprar a mais de 6 meses
    if dm.QConsulta.FieldName('CheClien').AsInteger > 0 then
        writeLn(ArqTexto,'OK')
    else
        writeLn(ArqTexto,'BL');
    end;

dm.QConsulta.Close;
statusBar;
Next;

end;

```

```

end
else
  While not eof do
  begin
    write(ArqTexto,FieldByName('CliNrDoc').AsString + ';');
    writeln(ArqTexto,FieldByName('CliNomes').AsString + ';');
    statusBar;
    Next;
  end;

  // Grava Clientes terceiros

  fEntra.Label4.Caption := 'Gravando arquivo para consulta nos Caixas';
  FEntra.Refresh;

  close;
  sql.Clear;
  sql.add('Select distinct TerNumer, TerNrCpf, TerNomes, TerEnder, TerSitua from grTerce');
  sql.add('Order by TerNomes');

  Open;

  if vMostraMensagemUsuarioRestricoes then
  begin
    While not eof do
    begin
      write(ArqTexto,FieldByName('TerNrCpf').AsString + ';');
      write(ArqTexto,FieldByName('TerNomes').AsString + ';');

      statusBar;
      dm.QConsulta.Close;
      dm.QConsulta.SQL.Clear;
      dm.QConsulta.SQL.Add('Select CheFundo from GrCheRe where CheTerce = ' +
        IntToStr(dm.QPesquisa.FieldByName('TerNumer').AsInteger));
      dm.QConsulta.SQL.Add('And CheFundo = true'); // cliente com cheques sem fundo
      dm.QConsulta.Open;
      if ((dm.QConsulta.FieldByName('CheFundo').AsBoolean) or
        (dm.QPesquisa.FieldByName('TerSitua').AsBoolean)) then
        write(ArqTexto,'1;') //Cliente inadimplente
      else
        write(ArqTexto,'0;') // Cliente ok

      if dm.QPesquisa.FieldByName('TerEnder').AsInteger > 0 then
        writeln(ArqTexto,'1') // 1 = tem endereço
      else
        writeln(ArqTexto,'0'); // 0 = nao tem endereço

      dm.QConsulta.Close;
      statusBar;
      Next;
    end;
  end;

```

```

end;

result := true;
end;
closeFile(ArqTexto);

end;

function TTelMenus.fCriarArquivoTexto : boolean;
begin

try
dm.QPesquisa.Close;
dm.QPesquisa.SQL.clear;
dm.QPesquisa.Sql.Add('Select * From grGerTxt');
dm.QPesquisa.Open;
if dm.QPesquisa.FieldByName('Data').asDateTime <> date then
result := true
else
result := false;
except
showMessage('erro 004 - acesso tabela txt. ');
end;

end;

procedure TTelMenus.pCriarArquivoConsultaCPFCliente;
begin
statusBar;
if fCriarArquivoTexto then
begin

AssignFile(ArqTexto,vDiretorio + '\Cadastro.txt');
{$I-}
rewrite(ArqTexto);
{$I+}
// Gravar arquivo txt;

statusBar;
if fGravarArquivoTxt then
begin
// Grava data
dm.QPesquisa.Close;
dm.QPesquisa.SQL.clear;
dm.QPesquisa.Sql.Add('UpDate grGerTxt');
dm.QPesquisa.SQL.Add('Set data = :1');
dm.QPesquisa.Sql.Add('Where codigo = 1');
dm.QPesquisa.ParamByName('1').AsDateTime := date;
dm.QPesquisa.ExecSql;
end;

```

```

end;
statusBar;

end;

function TTelMenus.CriarTabelaTerceiros: Boolean;
var
  vCampo : String;
begin
  try
    with tbTempo do
      begin
        Close;
        TableName := 'GrTerce.db';
        Open;

        if tbTempo.FieldByName('TerNrCasa').Size <> 10 then
          begin
            Close;
            Dm.QCriaT.Close;
            Dm.QCriaT.Sql.Clear;
            Dm.QCriaT.Sql.Add('Alter Table GrTerce ');
            Dm.QCriaT.Sql.Add('alter column TerNrCasa to varchar(10)');
            try
              Dm.QCriaT.ExecSql;
            except
              on e:Exception do
                begin
                  ShowMessage(e.message);
                end;
              end;
            end;
          end;
        except
          end;
      end;
    end;

function TTelMenus.CriarTabelaParaContasAPagar: Boolean;
begin
  try
    with tbTempo do
      begin
        Close;
        TableName := 'GrConPa.db';
        Open;
        Close;
        Result := true;
      end;
    except
      with Dm.QCriaT do

```

```

begin
  try
    Close;
    Sql.Clear;
    Sql.Add('Create Table GrConPa ');
    Sql.Add('(ConNumer Integer, ConForne Integer, ConNrDoc VarChar(20),');
    Sql.Add('ConDtLan Date, ConDtVen Date, ConValor Money, ConHisto VarChar(100),');
    Sql.Add('ConDtPag Date,');
    Sql.Add('Primary Key (ConNumer))');
    ExecSql;
    Result := true;

    Close;
    Sql.Clear;
    Sql.Add('Create Index NrDocumento on GrConPa (ConNrDoc) ');
    ExecSql;
  except
    Result := false;
  end;
end;
end;
end;

```

function TTelMenus.CriarCampoControle: Boolean;

var

vCampo : TDate;
vDespe : Boolean;
vStr : String;

begin

try

with tbTempo do

begin

Close;

TableName := 'GrCheTe.db';

Open;

vDespe := FieldByName('CheDespe').AsBoolean;

Close;

Result := true;

end;

except

with Dm.QCriaT do

begin

try

tbTempo.Close;

Close;

Sql.Clear;

Sql.Add('Alter Table GrCheTe ');

Sql.Add('Add CheDespe Boolean');

ExecSql;

```

        Close;
        Sql.Clear;
        Sql.Add('UpDate GrCheTe ');
        Sql.Add('Set CheDespe = False');
        ExecSql;

        Result := true;
    except
        Result := false;
    end;
end;
end;

try
    with tbTempo do
    begin
        Close;
        TableName := 'GrCheRe.db';
        Open;
        vCampo := FieldByName('CheDtMod').AsDateTime;
        Close;
        Result := true;
    end;
except
    with Dm.QCriaT do
    begin
        try
            tbTempo.Close;
            Close;
            Sql.Clear;
            Sql.Add('Alter Table GrCheRe ');
            Sql.Add('Add CheDtMod Date');
            ExecSql;
            Result := true;
        except
            Result := false;
        end;
    end;
end;

try
    with tbTempo do
    begin
        tbTempo.Close;
        Close;
        TableName := 'GrCheEm.db';
        Open;
        vCampo := FieldByName('CheDtMod').AsDateTime;
        Close;
        Result := true;
    end;
end;

```



```

end;
except
with Dm.QCriaT do
begin
tbTempo.Close;
try
Close;
Sql.Clear;
Sql.Add('Alter Table GrCheEm ');
Sql.Add('Add CheDtMod Date');
ExecSql;
Result := true;
except
Result := false;
end;
end;
end;

try
with tbTempo do
begin
tbTempo.Close;
Close;
TableName := 'GrCheTe.db';
Open;
vCampo := FieldByName('CheDtMod').AsDateTime;
Close;
Result := true;
end;
except
with Dm.QCriaT do
begin
tbTempo.Close;
try
Close;
Sql.Clear;
Sql.Add('Alter Table GrCheTe ');
Sql.Add('Add CheDtMod Date');
ExecSql;
Result := true;
except
Result := false;
end;
end;
end;
end;

function TTelMenus.CriarTabelaParaControleCaderno: Boolean;
begin
try
with tbTempo do
begin

```

```

Close;
TableName := 'GrCader.db';
Open;
Close;
Result := true;
end;
except
with Dm.QCriaT do
begin
try
Close;
Sql.Clear;
Sql.Add('Create Table GrCader ');
Sql.Add('(CadNumer Integer,CadTipos Integer,');
Sql.Add('CadDatas Date,CadValor Money,CadHisto VarChar(100),');
Sql.Add('Primary Key (CadNumer))');
ExecSql;
Result := true;
except
Result := false;
end;
end;
end;
end;

```

```

function TTelMenus.CriarTabelaParaControleDeTroco: Boolean;
begin
try
with tbTempo do
begin
Close;
TableName := 'GrTroco.db';
Open;
Close;
Result := true;
end;
except
with Dm.QCriaT do
begin
try
Close;
Sql.Clear;
Sql.Add('Create Table GrTroco ');
Sql.Add('(TroNumer Integer, TroTerce Integer,TroTipos Integer,');
Sql.Add('TroDatas Date,TroValor Money,TroObser VarChar(100),');
Sql.Add('Primary Key (TroNumer))');
ExecSql;
Result := true;
except
Result := false;
end;
end;

```

```

        end;
    end;
end;

procedure TTelMenus.CriarCampoDataChequeRepassadosParaTerceiros;
var
    vCampo : TDateTime;
begin
    try
        tbTempo.Close;
        tbTempo.TableName := 'GrCheTe';
        TbTempo.Open;
        vCampo := TbTempo.FieldByName('CheDatas').AsDateTime
    except
        tbTempo.Close;
        with dm.QCriaT do
            begin
                Close;
                Sql.Clear;
                Sql.Add('Alter Table GrCheTe Add CheDatas Date ');
                ExecSql;
                Close;
                Sql.Clear;
                Sql.Add('UpDate GrCheTe Set CheDatas = :vDatas ');
                ParamByName('vDatas').DataType := ftDate;
                ParamByName('vDatas').Clear;
                ExecSql;
            end;
        end;
    end;
end;

procedure TTelMenus.CriarIndiceTabelaChequeReceber;
begin
    try
        dm.tbCheRe.Open;
        dm.tbCheRe.IndexName := 'Clien';
        dm.TbCheRe.Close;
    except
        with dm.QInserir do
            begin
                dm.tbCheRe.Close;
                Close;
                Sql.Clear;
                Sql.Add('Create Index Clien on GrCheRe (CheClien)');
                ExecSql;
            end;
        end;
    end;
end;

procedure TTelMenus.CriarCampoDespesaTabelaChequesRecebidos;
var

```

```

vCampo : String;
begin
try
    tbTempo.Close;
    tbTempo.TableName := 'GrCheEm';
    TbTempo.Open;
    vCampo := TbTempo.FieldName('CheDespe').AsString;
except
    tbTempo.Close;
    with dm.QCriaT do
        begin
            Close;
            Sql.Clear;
            Sql.Add('Alter Table GrCheEm Add CheDespe VarChar(7) ');
            ExecSql;
            Close;
            Sql.Clear;
            Sql.Add('UpDate GrCheEm Set CheDespe = :vDespe ');
            ParamByName('vDespe').AsString := 'Não';
            ExecSql;
        end;
    end;
end;

procedure TTelMenus.FormCreate(Sender: TObject);
var
    vStr : String;
begin

    FEntra := TFEntra.Create(Self);
    FEntra.show;
    FEntra.Refresh;

    statusBar;
    vSequencia := 0;

    if ConectarBanco <> 'OK' then // Devolve a mensagem de erro
        begin
            MessageDlg('Erro ao conectar banco de dados.',
                mtInformation,[mbOk],0);

        try

            dm.TbBitMa.Close;

            dm.TbMunic.Close;
        except
            on E:Exception do
                begin
                    ShowMessage(e.message);
                end;
            end;
        end;
    end;
end;

```

```

    end;
end;

statusBar;

if TelSobre = nil then
    TelSobre := tTelSobre.Create(Self);
    TelSobre.ShowModal;
    Application.Terminate;
    Exit;
end;

if verificaAcesso then
    exit;

statusBar;

vControle := BuscaNumero('GrAcess','AceNumer');
try
    dm.TbAcess.Open;
    dm.TbAcess.Append;
    dm.TbAcess.FieldName('AceNumer').AsInteger := vControle;
    dm.TbAcess.FieldName('AceUsuar').AsInteger := vUsuNumer;
    dm.TbAcess.Post;
    dm.TbAcess.Close;
except
    showMessage('erro ao acessar tabela grAcess. 002');
end;

statusBar;

try

    dm.QPesquisa.Close;
    dm.QPesquisa.Sql.Clear;
    dm.QPesquisa.Sql.Add('Select C.CheNrChe,C.CheDtEmi,C.CheValor,C.CheRepas,' +
        'B.BanNomes,A.CliNomes From GrCheRe C,GrClie A,GrBanco B ' +
        ' Where (C.CheTipos = 2) and (C.CheBaixa = False) ' +
        ' and (C.CheClie = A.CliNumer) And (C.CheBanco = B.BanNumer) ' +
        ' And (CheDtDes = :1) ');

    dm.QPesquisa.ParamByName('1').AsDateTime := Date;
    dm.QPesquisa.Open;

    sgCheRe.Cells[0,0] := 'Número';
    sgCheRe.Cells[1,0] := 'Banco';
    sgCheRe.Cells[2,0] := 'Dt Emissão';

```

```

sgCheRe.Cells[3,0] := 'Valor';
sgCheRe.Cells[4,0] := 'Recebido de';
sgCheRe.Cells[5,0] := 'Repassado';
while not dm.QPesquisa.EOF do
begin
sgCheRe.Cells[0,sgCheRe.RowCount - 1] :=
    dm.QPesquisa.FieldName('CheNrChe').AsString;
sgCheRe.Cells[1,sgCheRe.RowCount - 1] :=
    dm.QPesquisa.FieldName('BanNomes').AsString;
sgCheRe.Cells[2,sgCheRe.RowCount - 1] :=
    DateToStr(dm.QPesquisa.FieldName('CheDtEmi').AsDateTime);
sgCheRe.Cells[3,sgCheRe.RowCount - 1] :=
    FormatFloat('##,###,##0.00', dm.QPesquisa.FieldName('CheValor').AsCurrency);
sgCheRe.Cells[4,sgCheRe.RowCount - 1] :=
    dm.QPesquisa.FieldName('CliNomes').AsString;
if dm.QPesquisa.FieldName('CheRepas').AsBoolean then
    sgCheRe.Cells[5,sgCheRe.RowCount - 1] := 'SIM'
else
    sgCheRe.Cells[5,sgCheRe.RowCount - 1] := 'NÃO';

sgCheRe.RowCount := sgCheRe.RowCount + 1;

dm.QPesquisa.Next;
end;

```

```

dm.QPesquisa.Close;
dm.QPesquisa.Sql.Clear;
dm.QPesquisa.Sql.Add('Select C.CheNrChe,C.CheDtEmi,C.CheValor, '+
    ' C.CheClien,B.BanNomes From GrCheEm C,GrBanco B ' +
    ' Where (C.CheBaixa = False) ' +
    ' and (C.CheBanco = B.BanNumer) and (C.CheDtDes = :1) ');

```

```

dm.QPesquisa.ParamByName('1').AsDateTime := Date;
dm.QPesquisa.Open;

```

```

sgCheEm.Cells[0,0] := 'Número';
sgCheEm.Cells[1,0] := 'Banco';
sgCheEm.Cells[2,0] := 'Dt Emissão';
sgCheEm.Cells[3,0] := 'Valor';
sgCheEm.Cells[4,0] := 'Emitido para';

```

```

while not dm.QPesquisa.EOF do
begin
sgCheEm.Cells[0,sgCheEm.RowCount - 1] :=
    dm.QPesquisa.FieldName('CheNrChe').AsString;
sgCheEm.Cells[1,sgCheEm.RowCount - 1] :=
    dm.QPesquisa.FieldName('BanNomes').AsString;
sgCheEm.Cells[2,sgCheEm.RowCount - 1] :=
    DateToStr(dm.QPesquisa.FieldName('CheDtEmi').AsDateTime);

```

```

sgCheEm.Cells[3,sgCheEm.RowCount - 1] :=
    FormatFloat('###,###,##0.00', dm.QPesquisa.FieldByName('CheValor').AsCurrency);
sgCheEm.Cells[4,sgCheEm.RowCount - 1] :=
    dm.QPesquisa.FieldByName('CheClien').AsString;

sgCheEm.RowCount := sgCheEm.RowCount + 1;

dm.QPesquisa.Next;
end;

dm.QPesquisa.Close;

GroupBox1.Caption := GroupBox1.Caption + ': ' + DateToStr(Date);
GroupBox2.Caption := GroupBox2.Caption + ': ' + DateToStr(Date);
except
On E:Exception do
begin
    showMessage(' Erro ao selecionar cheques recebidos e emitidos...');
end;

end;

statusBar;

try
    dm.QPesquisa.Close;
    dm.QPesquisa.sql.clear;
    dm.QPesquisa.sql.add('Select * from grGerTxt ');
    dm.QPesquisa.Open;
except

try
    with dm.QCriaT do
    begin
        close;
        sql.clear;
        sql.add('Create Table grGerTxt ');
        Sql.add('(codigo integer,data date,');
        Sql.Add('Primary key (codigo)');
        ExecSql;
    end;

    with dm.QInserir do
    begin
        close;
        sql.clear;
        sql.Add('Insert into grGerTxt values (1,:1)');
        ParamByName('1').AsDateTime := date - 1;
        ExecSql;
    end;
except
    showMessage('erro 005 - Gravar tabela Txt.');
```

```

end;

statusBar;
end;

criarCampoSituacaoTerceiro;

CriarTabelaControleChequeSemFundo;

CriarTabelaChequeEnviadosParaCobranca;
CriarTabelaEmpresa;

CriarCampoTabelaFornecedor; // Solicitação Airton em 18/07/2007
try
  dm.QInserir.Close;
  dm.QInserir.SQL.Clear;
  dm.QInserir.SQL.Add('Select * from leConfi');
  dm.QInserir.SQL.Add('Where ConNumer = :p1');
  dm.QInserir.ParamByName('p1').AsInteger := 10;
  dm.QInserir.Open;
  if dm.QInserir.FieldName('ConLinha').AsInteger = 1 then
    begin
      vUtilizaTXTNosCaixa := true;

    end
  else
    begin
      vUtilizaTXTNosCaixa := false;

    end;
except
end;

try
  dm.QInserir.Close;
  dm.QInserir.SQL.Clear;
  dm.QInserir.SQL.Add('Select * from leConfi');
  dm.QInserir.SQL.Add('Where ConNumer = :p1');
  dm.QInserir.ParamByName('p1').AsInteger := 11;
  dm.QInserir.Open;
  if dm.QInserir.FieldName('ConLinha').AsInteger = 1 then
    vMostraMensagemUsuarioRestricoes := true
  else
    vMostraMensagemUsuarioRestricoes := false;
except
end;

try
  dm.QInserir.Close;
  dm.QInserir.SQL.Clear;
  dm.QInserir.SQL.Add('Select * from leConfi');

```



```

dm.QInserir.SQL.Add('Where ConNumer = :p1');
dm.QInserir.ParamByName('p1').AsInteger := 12;
dm.QInserir.Open;
vMostraMensagemClientesSemFundo := true;
if dm.QInserir.FieldName('ConLinha').AsInteger = 1 then
    vMostraMensagemClientesSemFundo := true
else
    vMostraMensagemClientesSemFundo := false;
except
    dm.QInserir.Close;
end;

try
    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('Select * from leConfi');
    dm.QInserir.SQL.Add('Where ConNumer = :p1');
    dm.QInserir.ParamByName('p1').AsInteger := 13;
    dm.QInserir.Open;
    if dm.QInserir.FieldName('ConLinha').AsInteger = 1 then
        vPagarContasComDuploCliqueDoMouse := true
    else
        vPagarContasComDuploCliqueDoMouse := false;
except
end;

// Bloquear cliente com mais de 6 meses sem comprar
try
    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('Select * from leConfi');
    dm.QInserir.SQL.Add('Where ConNumer = :p1');
    dm.QInserir.ParamByName('p1').AsInteger := 19;
    dm.QInserir.Open;
    if dm.QInserir.FieldName('ConLinha').AsInteger = 1 then
        begin
            vBloquearClienteSemCompraSeisMeses := true;

        end
    else
        begin
            vBloquearClienteSemCompraSeisMeses := false;

        end;
except
end;
// fim

try
    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;

```

```

dm.QInserir.SQL.Add('Select * from leConfi');
dm.QInserir.SQL.Add('Where ConNumer = :p1');
dm.QInserir.ParamByName('p1').AsInteger := 1000;
dm.QInserir.Open;
if dm.QInserir.FieldByName('ConNumer').AsInteger = 0 then
begin
  dm.QInserir.Close;
  dm.QInserir.SQL.Clear;
  dm.QInserir.SQL.Add('Insert into leConfi');
  dm.QInserir.SQL.Add('(ConNumer,ConLinha)');
  dm.QInserir.SQL.Add('Values');
  dm.QInserir.SQL.Add('(:p1,:p2)');
  dm.QInserir.ParamByName('p1').AsInteger := 1000;
  dm.QInserir.ParamByName('p2').AsString := '1';
  dm.QInserir.ExecSql;
end
else
begin
  if StrToInt(dm.QInserir.FieldByName('ConLinha').AsString) > 10000 then
  begin
    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;

    try
      vSequencia := fBuscaSequencia;
    except
      inc(vSequencia);
    end;

    dm.QInserir.SQL.Add('UpDate leConfi');
    dm.QInserir.SQL.Add('Set ConLinha = :p2 where ConNumer = :p1');
    dm.QInserir.ParamByName('p1').AsInteger := 1000;
    dm.QInserir.ParamByName('p2').AsString := '1';
    dm.QInserir.ExecSql;
  end;

  end;
except
end;

try
dm.QInserir.Close;
dm.QInserir.SQL.Clear;
dm.QInserir.SQL.Add('Select * from leConfi');
dm.QInserir.SQL.Add('Where ConNumer = :p1');
dm.QInserir.ParamByName('p1').AsInteger := 14;
dm.QInserir.Open;
vMostraMensagemClientesSemFundo := true;
if dm.QInserir.FieldByName('ConLinha').AsInteger = 1 then
begin
  vUtilizaLimiteCompra := true;
  LimiteCliente.Visible := true;

```

```

FEntra.Label4.Caption := 'Gravando arquivo Limite Crédito Cliente';
FEntra.Refresh;

gravarArquivoLimiteSaldoCliente;
gravarArquivoLimiteSaldoTerceiro;

end
else
begin
    vUtilizaLimiteCompra := false;
    LimiteCliente.Visible := false;
end;
except
    dm.QInserir.Close;
end;

if vUtilizaTXTNosCaixa then
begin
    pCriarArquivoConsultaCPFCliente;
    Geraarquivoparaconsultanoscaixas.Visible := true;
end
else
    Geraarquivoparaconsultanoscaixas.Visible := false;

try
    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('Select * from leConfi');
    dm.QInserir.SQL.Add('Where ConNumer = :p1');
    dm.QInserir.ParamByName('p1').AsInteger := 15;
    dm.QInserir.Open;

    try
        if dm.QInserir.FieldName('ConLinha').AsInteger <> 0 then
            vCorClienteSemFundo := StrToInt(dm.QInserir.FieldName('ConLinha').AsString)
        else
            vCorClienteSemFundo:= 255;
        except
            vCorClienteSemFundo:= 255;
        end;

    except
        dm.QInserir.Close;
    end;

// Limite de compra para Terceiro
try
    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('Select * from leConfi');

```

```

dm.QInserir.SQL.Add('Where ConNumer = :p1');
dm.QInserir.ParamByName('p1').AsInteger := 16;
dm.QInserir.Open;
vMostraMensagemClientesSemFundo := true;
if dm.QInserir.FieldByName('ConLinha').AsInteger = 1 then
begin
    vUtilizaLimiteCompraTerceiro := true;
end
else
begin
    vUtilizaLimiteCompraTerceiro := false;
end;
except
    dm.QInserir.Close;
end;

// Baixar cheques automaticamente
try
    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('Select * from leConfi');
    dm.QInserir.SQL.Add('Where ConNumer = :p1');
    dm.QInserir.ParamByName('p1').AsInteger := 17;
    dm.QInserir.Open;
    vMostraMensagemClientesSemFundo := true;
    if dm.QInserir.FieldByName('ConLinha').AsInteger = 1 then
    begin
        pBaixarChequesAutomaticamente;
    end;
except
    dm.QInserir.Close;
end;

try
    vNegativarTerceiro := false;
    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('Select * from leConfi');
    dm.QInserir.SQL.Add('Where ConNumer = :p1');
    dm.QInserir.ParamByName('p1').AsInteger := 21;
    dm.QInserir.Open;
    if dm.QInserir.FieldByName('ConLinha').AsInteger = 1 then
        vNegativarTerceiro := true
    else
        vNegativarTerceiro := false;
except
end;

criarCampoTrocoCaderno;

criarCampoSituacaoChequeGrChSFun;

```

```

end;

procedure TTelMenus.Fornecedores1Click(Sender: TObject);
begin
    if TelForne = nil then
        TelForne := TTelForne.Create(Self); {UForne}
        TelForne.Show;
    end;

procedure TTelMenus.Sada1Click(Sender: TObject);
begin
    try
        Close;
    except
        end;
    end;

procedure TTelMenus.Sobre1Click(Sender: TObject);
begin
    if TelSobre = nil then
        TelSobre := tTelSobre.Create(Self); {USobre}
        TelSobre.ShowModal;

end;

procedure TTelMenus.FormClose(Sender: TObject; var Action: TCloseAction);
begin

    try
        dm.TbAcess.Open;
        if dm.TbAcess.RecordCount > 1000 then
            begin
                dm.TbAcess.Close;
                dm.TbAcess.EmptyTable;
            end;
        except
            end;

    if DesconectarBanco <> 'OK' then // Devolve a mensagem de erro
        begin
            MessageDlg('Erro ao desconectar banco de dados.',
                mtInformation,[mbOk],0);
            if TelSobre = nil then
                TelSobre := tTelSobre.Create(Self); {USobre}
                TelSobre.ShowModal;
            end;

            Application.Terminate;

end;

```

```

procedure TTelMenus.Bancos1Click(Sender: TObject);
begin
  if TelBanco = nil then
    TelBanco := tTelBanco.Create(Self); {UBanco}
    TelBanco.ShowModal;
end;

procedure TTelMenus.ChequesEmitidos1Click(Sender: TObject);
begin
  if TelCheEm = nil then
    TelCheEm := tTelCheEm.Create(Self); {UCheEm}
    TelCheEm.ShowModal;

end;

procedure TTelMenus.ChequesRecebidos1Click(Sender: TObject);
begin
  if TelCheRe = nil then
    TelCheRe := tTelCheRe.Create(Self); {UCheRe}
    TelCheRe.ShowModal;
    TelCheRe.Free;
    TelCheRe := nil;
end;

procedure TTelMenus.BaixadeCheques1Click(Sender: TObject);
begin
  if TelBCheq = nil then
    TelBCheq := tTelBCheq.Create(Self); {UBCheq}
    TelBCheq.ShowModal;
end;

procedure TTelMenus.Terceiros1Click(Sender: TObject);
begin
  pnMensa.Visible := true;
  pnMensa.Refresh;
  if TelTerce = nil then
    TelTerce := tTelTerce.Create(Self); {UTerce}
    TelTerce.ShowModal;
  pnMensa.Visible := false;
end;

procedure TTelMenus.FechamentoMs2Click(Sender: TObject);
begin
  if TelFecha = nil then
    TelFecha := TTelFecha.Create(Self); {UFecha}

    TelFecha.ShowModal;

end;

procedure TTelMenus.ChequesRecebidos2Click(Sender: TObject);
begin

```

```

if TelRCher = nil then
    TelRCher := TTelRCher.Create(Self); {URCher}

    TelRCher.ShowModal;
end;

procedure TTelMenus.FechamentoMs3Click(Sender: TObject);
begin
    if TelRFech = nil then
        TelRFech := TTelRFech.Create(Self); {URFech}

        TelRFech.ShowModal;

    end;

procedure TTelMenus.Endereos1Click(Sender: TObject);
begin
    if TelEnder = nil then
        TelEnder := TTelEnder.Create(Self); {UEnder}

        TelEnder.Show;

    end;

procedure TTelMenus.ListaParaBanco1Click(Sender: TObject);
begin
    if TelLista = nil then
        TelLista := TTelLista.Create(Self); {ULista}

        TelLista.Show;

    end;

procedure TTelMenus.EsconderChequesPreRecebidos1Click(Sender: TObject);
begin
    GroupBox2.Visible := false;
end;

procedure TTelMenus.EsconderChequePrRecebidos1Click(Sender: TObject);
begin
    GroupBox1.Visible := false;
end;

procedure TTelMenus.ControleCaderno1Click(Sender: TObject);
begin
    if TelContr = nil then
        TelContr := TTelContr.Create(Self);

        TelContr.Show;

    end;

```

```

procedure TTelMenus.LanamentosCaderno1Click(Sender: TObject);
begin
  if TelRCade = nil then
    TelRCade := TTelRCade.Create(Self); {URCade}
    TelRCade.ShowModal;

end;

procedure TTelMenus.ContasAPagar2Click(Sender: TObject);
begin
  if TelCAPag = nil then
    TelCAPag := TTelCAPag.Create(Self); {UCApag}
    TelCAPag.Show;

end;

procedure TTelMenus.ContasPagar1Click(Sender: TObject);
begin
  if TelRConP = nil then
    TelRConP := TTelRConP.Create(Self); {URConP}

    TelRConP.Show;

end;

procedure TTelMenus.FormShow(Sender: TObject);
var
  vStr : String;
  vData : TDate;

begin

  if vUsuario = 1 then
    begin
      ChequesRecebidos1Click(Sender);
      exit;
    end;

  try
    with tbTempo do
      begin
        Close;
        TableName := 'GrCheRe.db';
        Open;
        vStr := FieldByName('CheNrDoc').AsString;
        Close;
      end;
    except
    end;

    vData := date - 5;

```



```

try
    pCriarTabelaControle;

    dm.QInserir.Close;
    dm.QInserir.SQL.Clear;
    dm.QInserir.SQL.Add('Select * from grControle');
    dm.QInserir.Open;
    if dm.QInserir.FieldByName('ConDatas').AsDateTime < Date - 10 then
    begin
        dm.QPesquisa.close;
        dm.QPesquisa.Sql.Clear;
        dm.QPesquisa.Sql.Add('Select * from GrCheRe ' +
            'Where cheDtDes < :vData and cheBaixa = false');
        dm.QPesquisa.ParamByName('vData').AsDateTime := vData;
        dm.QPesquisa.open;

        if dm.QPesquisa.RecordCount > 0 then
        begin
            if MessageDlg('Existe cheques com data de desconto maior/igual a 5 dias passados' +
                chr(13) +
                'DESEJA PASSAR ESTES CHEQUES PARA DESCONTADOS?',
                mtConfirmation,[mbYes,mbNo],0) = mrYes then
            begin
                dm.QInserir.Close;
                dm.QInserir.Sql.Clear;
                dm.QInserir.SQL.Add('UpDate GrCheRe set CheBaixa = true where cheDtDes <= :vData
and cheBaixa = false');
                dm.QInserir.ParamByName('vData').AsDateTime := vData;
                dm.QInserir.ExecSQL;
                dm.QInserir.close;

                end;
            end;

            dm.QInserir.Close;
            dm.QInserir.SQL.Clear;
            dm.QInserir.SQL.Add('UpDate grControle');
            dm.QInserir.SQL.Add('Set ConDatas = :p1');
            dm.QInserir.ParamByName('p1').AsDateTime := date;
            dm.QInserir.ExecSql;

            end;
        except
        on e:exception do
            showMessage(e.message);

        end;

        with dm.QPesquisa do
        begin
            close;
            sql.clear;

```

```

    sql.Add('Select * from grEmpre');
    open;
    vEmpresa := fieldByName('EmpNomes').AsString;
    vCnpjEmpresa := fieldByName('EmpCnpj').AsString;
end;

CriarCampoLimiteTerceiro;

FEntra.Close;

end;

procedure TTelMenus.ChequesSemFundo1Click(Sender: TObject);
begin
    if TelRSemF = nil then
        TelRSemF := TTelRSemF.Create(Self); {URSemF}

        TelRSemF.Show;
    end;

procedure TTelMenus.Chequesrecebidosdeterceiros1Click(Sender: TObject);
begin

    if FRChequeRecebidosTerceiros = nil then
        FRChequeRecebidosTerceiros := TFRChequeRecebidosTerceiros.Create(Self);
        FRChequeRecebidosTerceiros.Show;
    end;

procedure TTelMenus.TabeladeChequesRecebidos1Click(Sender: TObject);
begin
    if FManutencaoChequeRecebidos = nil then
        FManutencaoChequeRecebidos := TManutencaoChequeRecebidos.Create(Self);
        FManutencaoChequeRecebidos.ShowModal;
        FManutencaoChequeRecebidos.Free;
        FManutencaoChequeRecebidos := Nil;

    end;

procedure TTelMenus.ChequesRecebidos3Click(Sender: TObject);
begin
    if TelRCheque = nil then
        TelRCheque := TTelRCheque.create(Self);

        TelRCheque.show;
    end;

```

```

procedure TTelMenus.SQL1Click(Sender: TObject);
begin
    if FRelSQL = nil then
        FRelSQL := TRelSQL.create(self);

    FRelSQL.showModal;
end;

```

```

procedure TTelMenus.Juros1Click(Sender: TObject);
begin
    if TelJuros = nil then
        TelJuros := TTelJuros.create(self);

    TelJuros.showModal;
end;

```

```

procedure TTelMenus.Configurao1Click(Sender: TObject);
begin
    if FConfig = nil then
        FConfig := TConfig.Create(Self); {UConfi}

    FConfig.ShowModal;
    FConfig.Free;
    FConfig := Nil;

    if vUtilizaTXTNosCaixa then
        Geraarquivoparaconsultanoscaixas.Visible := true
    else
        Geraarquivoparaconsultanoscaixas.Visible := false;

end;

```

```

procedure TTelMenus.Relatriodechequesparacobranca1Click(Sender: TObject);
begin
    if FRelCheqCobranca = nil then
        FRelCheqCobranca := TRelCheqCobranca.Create(Self);

    FRelCheqCobranca.ShowModal;
    FRelCheqCobranca.Free;
    FRelCheqCobranca := nil;
end;

```

```

procedure TTelMenus.Empresa1Click(Sender: TObject);
begin
    if FEmpresa = nil then
        FEmpresa := TEmpresa.Create(Self);

    FEmpresa.ShowModal;
    FEmpresa.Free;

```

```

FEmpresa := nil;

end;

procedure TTelMenus.CorrirTroco1Click(Sender: TObject);
begin
    if FCorrigeTroco = nil then
        FCorrigeTroco := TFCorrigeTroco.Create(Self);

        FCorrigeTroco.ShowModal;
        FCorrigeTroco.Free;
        FCorrigeTroco := nil;

end;

procedure TTelMenus.ConfigurarTelas1Click(Sender: TObject);
begin
    if TelConfiTela = nil then
        TelConfiTela := TTelConfiTela.Create(Self);

        TelConfiTela.ShowModal;
        TelConfiTela.Free;
        TelConfiTela := nil;

end;

procedure TTelMenus.GeraarquivoparaconsultanoscaixasClick(Sender: TObject);
begin
    try
        if MessageDlg('Confirma a geração de arquivo texto para consulta nos caixas?',
            mtConfirmation,[mbYes,mbNo],0) = mrYes then
            begin

                dm.QInserir.Close;
                dm.QInserir.SQL.clear;
                dm.QInserir.SQL.Add('UpDate GrGerTxt set Data = :pData');
                dm.QInserir.ParamByName('pData').AsDateTime := date - 1;
                dm.QInserir.ExecSql;
                dm.QInserir.Close;

                if FEntra = nil then
                    FEntra := TFEntra.Create(Self);

                FEntra.Label4.Caption := 'Gravando arquivo Limite Crédito Cliente';
                FEntra.Refresh;

                gravarArquivoLimiteSaldoCliente;

                gravarArquivoLimiteSaldoTerceiro;
            end
        end
    end
end;

```

```

AssignFile(ArqTexto,vDiretorio + '\Cadastro.txt');
{$I-}
    rewrite(ArqTexto);
{$I+}
    // Gravar arquivo txt;

FEntra.Show;
FEntra.Label4.Visible := true;
FEntra.Refresh;

statusBar;
if fGravarArquivoTxt then
begin
    // Grava data
    dm.QPesquisa.Close;
    dm.QPesquisa.SQL.clear;
    dm.QPesquisa.Sql.Add('UpDate grGerTxt');
    dm.QPesquisa.SQL.Add('Set data = :1');
    dm.QPesquisa.Sql.Add('Where codigo = 1');
    dm.QPesquisa.ParamByName('1').AsDateTime := date;
    dm.QPesquisa.ExecSql;
end;

FEntra.Close;

end;

except
end;
end;

procedure TTelMenus.LimiteClienteClick(Sender: TObject);
begin
    if FLimiteCliente = nil then
        FLimiteCliente := TFLimiteCliente.Create(Self);
    FLimiteCliente.Show;
end;

procedure TTelMenus.GerararquivoparaconsultadeSaldodeClientesClick(
    Sender: TObject);
begin
    try
        if MessageDlg('Confirma a geração de arquivo texto para consulta de Saldo de Clientes nos
caixas?',
            mtConfirmation,[mbYes,mbNo],0) = mrYes then
            begin

                if FEntra = nil then
                    FEntra := TFEntra.Create(Self);

```

```

FEntra.Show;
FEntra.Label4.Visible := true;
FEntra.Refresh;

gravarArquivoLimiteSaldoCliente;
if vUtilizaLimiteCompraTerceiro then
    gravarArquivoLimiteSaldoTerceiro;

    FEntra.Close;
end;
except
end;
end;

procedure TTelMenus.FormKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if vUsuario = 1 then
        begin
            if key = vk_f2 then
                ChequesRecebidos1Click(Sender)
            else
                if key = vk_f12 then
                    Sada1Click(Sender);
                end;
            end;
        end;

end;

procedure TTelMenus.buCadastroClick(Sender: TObject);
begin
    ChequesRecebidos1Click(Sender);
end;

procedure TTelMenus.buFecharClick(Sender: TObject);
begin
    Sada1Click(Sender);
end;

end.

```