

```

unit uClienteControle;

interface

uses uControle, classes, SysUtils, Dialogs ;

type

TClienteControle = class
private
    { private declarations }
    FCodigo      : String;
    FNome        : String;
    FTelefone    : String;
    // classe de persistencia
    FControle    : TControle;
protected
    { protected declarations }
public
    { public declarations }
    aCodigo, aNome, aTelefone: Array of String;
    Auxsql       : String;
    IndControle  : Integer;
    ListaCliente: TStrings;

    constructor Create(pConexaoControle : TControle);
    destructor destroy; override;
//
    property Codigo      : String read FCodigo      write FCodigo;
    property Nome        : String read FNome        write FNome;
    property Telefone    : String read FTelefone    write FTelefone;
//
    function InsereCliente      : Boolean;
    function AlteraCliente      : Boolean;
    function ExcluiCliente      : Boolean;
    function ConsultaCliente    : Boolean;
    function PesquisaCliente    : Boolean;

published
    { published declarations }
end;

implementation

{ TClienteControle }

constructor TClienteControle.Create(pConexaoControle : TControle);
begin
    FControle := pConexaoControle;
end;

destructor TClienteControle.destroy;
begin
    inherited;
end;

function TClienteControle.AlterarCliente: Boolean;
begin
    auxsql := ' update tblcliente ';
    auxsql := auxsql + ' set ';
    auxsql := auxsql + ' nome      = :nome, ';
    auxsql := auxsql + ' telefone = :telefone ';
    auxsql := auxsql + ' where ';
    auxsql := auxsql + ' codigo    = :codigo ';

    FControle.SqqGeral.Close;
    FControle.SqqGeral.CommandText := auxsql;
    FControle.SqqGeral.Params[0].AsString := Self.Nome;
    FControle.SqqGeral.Params[1].AsString := Self.Telefone;
    FControle.SqqGeral.Params[2].AsString := Self.Codigo;

    try

```

```

FControle.SqqGeral.ExecSQL();
Result := True;
except
    Result := False;
end;

end;

function TClienteControle.ExcluiCliente: Boolean;
begin
    auxsql := ' delete from tblcliente where codigo = :codigo ';
    FControle.SqqGeral.Close;
    FControle.SqqGeral.CommandText := auxsql;
    FControle.SqqGeral.Params[0].AsString := Self.Codigo;
    try
        FControle.SqqGeral.ExecSQL();
        Result := True;
    except
        Result := False;
    end;
end;

end;

function TClienteControle.InsereCliente: Boolean;
begin
    if Trim(self.Codigo) = '' then begin
        Result := False;
        exit;
    end;

    if Trim(self.Nome) = '' then begin
        Result := False;
        exit;
    end;

    if Trim(self.Telefone) = '' then begin
        Result := False;
        exit;
    end;

    auxsql := ' select codigo from tblcliente where codigo = :codigo ';
    FControle.SqqGeral.Close;
    FControle.SqqGeral.CommandText := auxsql;
    FControle.SqqGeral.Params[0].AsString := Self.Codigo;

    try
        FControle.SqqGeral.Open;
        if FControle.SqqGeral.Eof = False then begin
            Result := False;
            exit;
        end;
    except
        Result := False;
        exit;
    end;

    auxsql := ' insert into tblcliente ';
    auxsql := auxsql + ' ( codigo, nome, telefone ) ';
    auxsql := auxsql + ' values ';
    auxsql := auxsql + ' ( :codigo, :nome, :telefone ) ';

    FControle.SqqGeral.Close;
    FControle.SqqGeral.CommandText := auxsql;
    FControle.SqqGeral.Params[0].AsString := Self.Codigo;
    FControle.SqqGeral.Params[1].AsString := Self.Nome;
    FControle.SqqGeral.Params[2].AsString := Self.Telefone;

    try
        FControle.SqqGeral.ExecSQL();

```

```

    Result := True;
except
    Result := False;
end;

end;

function TClienteControle.ConsultaCliente: Boolean;
begin
    auxsql := ' select * from tblcliente order by nome ';

    FControle.SqqGeral.Close;
    FControle.SqqGeral.CommandText := auxsql;

    try
        FControle.SqqGeral.Open;

        if FControle.SqqGeral.RecordCount > 0 then begin
            ListaCliente := TStringList.Create;
            end;

        SetLength(aCodigo, FControle.SqqGeral.RecordCount);
        SetLength(aNome, FControle.SqqGeral.RecordCount);
        SetLength(aTelefone, FControle.SqqGeral.RecordCount);

        IndControle := 0;
        while FControle.SqqGeral.Eof = False do begin
            aCodigo[indControle] := FControle.SqqGeral.FieldName('codigo').AsString;
            aNome[indControle] := FControle.SqqGeral.FieldName('nome').AsString;
            aTelefone[indControle] := FControle.SqqGeral.FieldName('telefone').AsString;
            IndControle := IndControle + 1;
            FControle.SqqGeral.Next;
            end;

        Result := True;
    except
        Result := False;
    end;

end;

function TClienteControle.PesquisaCliente : Boolean;
begin
    auxsql := ' select * from tblcliente where codigo = :codigo ';
    FControle.SqqGeral.Close;
    FControle.SqqGeral.CommandText := auxsql;
    FControle.SqqGeral.Params[0].AsString := Self.Codigo;

    try
        FControle.SqqGeral.Open;
        Self.Nome := FControle.SqqGeral.FieldName('nome').AsString;
        Self.Telefone := FControle.SqqGeral.FieldName('telefone').AsString;
        Result := True;
    except
        Result := False;
    end;

end;

end.

```