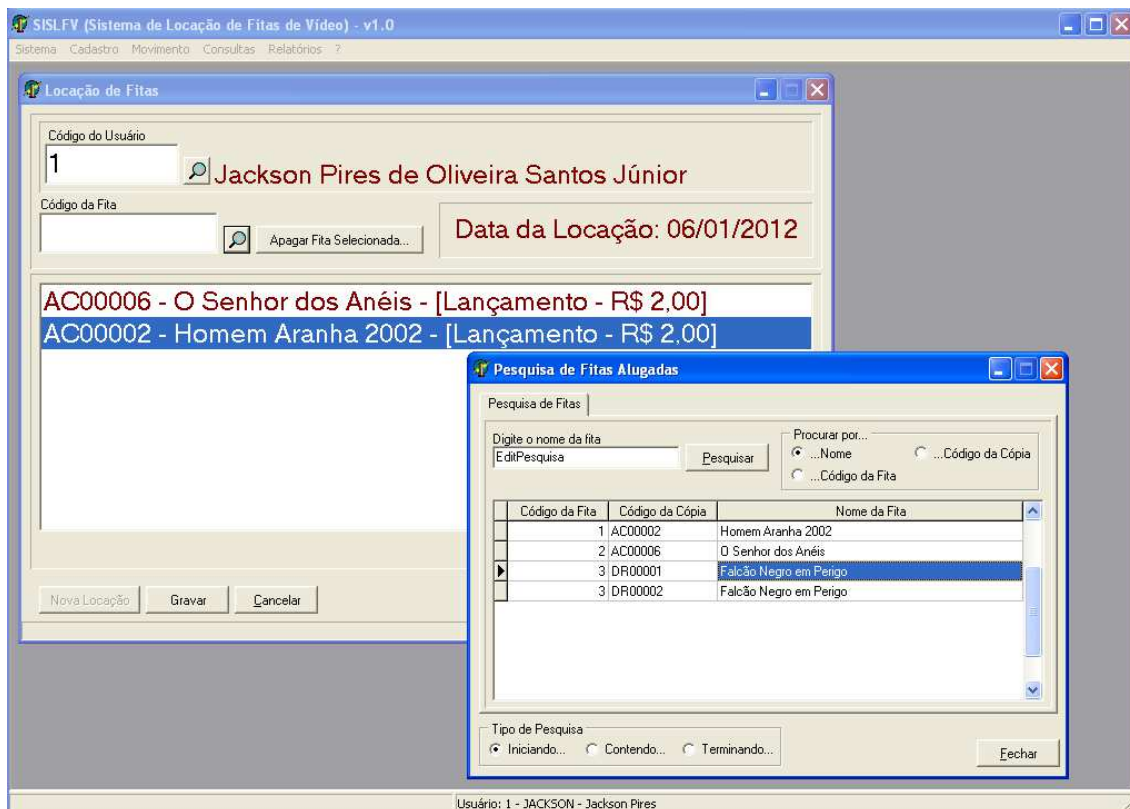


Curso de Delphi

Projeto de Sistema

SISLFV

Sistema de Locação de Fitas de Vídeo



Jackson Pires de Oliveira Santos Júnior
2002

- 1º Passo (Iniciando o Projeto) -

1) Nome do Projeto: **SISLFV** (Sistema de Locação de Fitas de Vídeo)

2) Crie os diretórios necessários para iniciar o projeto:



3) Crie um ALIAS no BDE (Borland Database Engine) tipo STANDARD: **SISLFV**

4) Crie um arquivo de nome “ALIAS.TXT” contendo o texto:

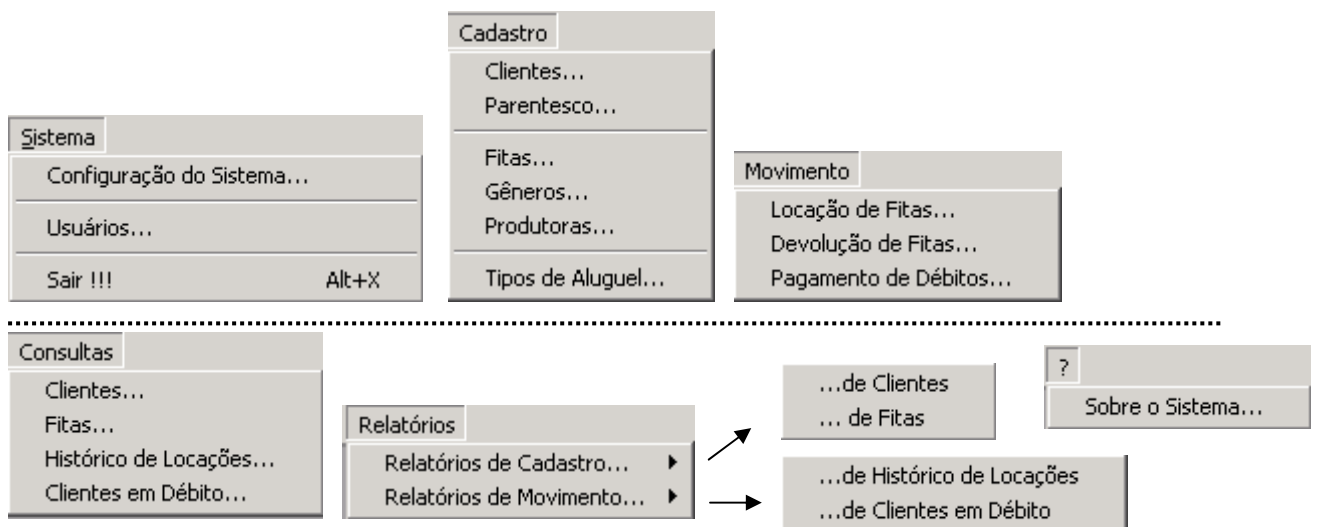
- “ **Projeto SISLFV (Sistema de Locação de Fitas de Vídeo)** ”
- “ **ALIAS (STANDARD) = SISLFV** ”
- “ **Projeto iniciado em: 19/10/2002** ”

Salve-o dentro da pasta “SISLFV/Documentos”.

5) Crie as tabelas (lembre-se de gravá-las na pasta SISLFV/Tabelas) a seguir: (Folha em Anexo)

6) Inicie o Delphi e siga os procedimentos abaixo:

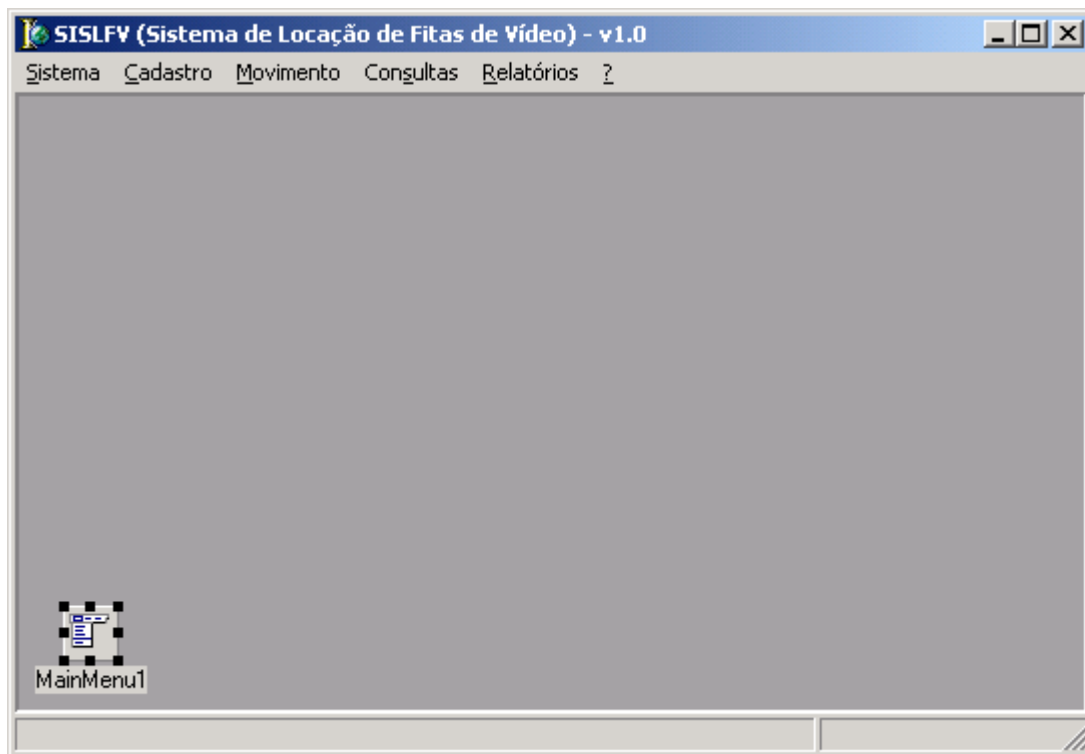
- Salve a Unit como “**Principal.pas**”;
- Salve o Projeto como “**SISLFV.dpr**”;
- Altere a propriedade **NAME** do **Form1** para “**FormPrincipal**”;
- Altere a propriedade **CAPTION** do **FormPrincipal** para “**SISLFV (Sistema de Locação de Fitas de Vídeo) - v1.0**”;
- Insira um (01) componente não visual **MainMenu** localizado na palheta Standard e crie os seguintes menus:



7) Após Criar os Menus siga os seguintes passos:

- Adicione um (01) componente visual **StatusBar** localizado na palheta **Win32**;
- Altere a propriedade **Panel** do componente **StatusBar1** inserindo dois (02) novos Panels;
- Selecione o Panel **0** (Zero) e altere a propriedade **Width** para **400**;
- Selecione o **FormPrincipal** e Altere a propriedade **WindowState** para **wsMaximized**;
- Insira um (01) componente visual **Panel** localizado na palheta **Standard**;
- Altere a propriedade **Caption** apagando o nome **Panel1**;
- Altere a propriedade **BevelWidth** para **2**;
- Altere a propriedade **BevelOuter** para **bvLowered**;
- Altere a propriedade **Align** para **alClient**; Altere a propriedade **Color** para **clMedGray**

Após alterar essas opções seu FormPrincipal deve estar parecido com a figura abaixo.



8) Para terminar esse 1º Passo insira o comando abaixo no evento **OnClick** do menu **Sistema/Sair**:

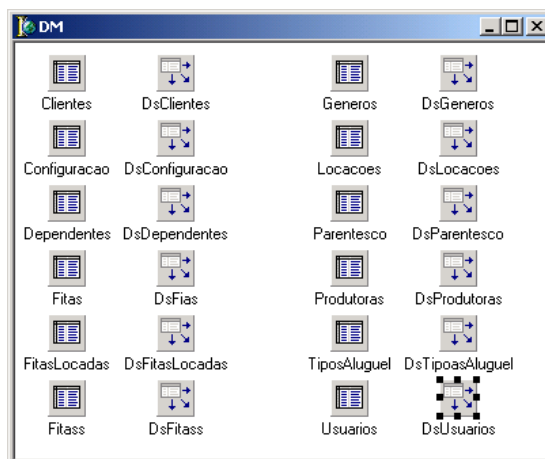
```
procedure TFormPrincipal.Sair1Click(Sender: TObject);  
begin  
  Application.Terminate;  
end;
```

- 2º Passo (Criando e Configurando o Data Module) -

- 1) Com o seu sistema aberto no Delphi, clique no menu **File/New/Data Module** do Delphi;
- 2) Salve o DataModule1 com o nome **“Tabelas.pas”** na pasta **SISLFV/Units**;
- 3) Altere a propriedade **Name** do DataModule1 para **“DM”**; Pressione **F12**, vá até a cláusula **Uses** e adicione a Unit **DBIProcs**;
- 4) Adicione doze (12) componentes não visuais **Table** que estão localizados na palheta **BDE**;
- 5) Selecione todos os doze (12) componentes e altere a propriedade **DataBaseName** para **“SISLFV”**;
- 6) Agora selecione apenas o Table1 e altere as seguintes propriedades:
 - Na propriedade **TableName** selecione a Tabela **Clientes.DB**
 - Altere a propriedade **Name** para **Clientes**
- 7) Faça um clique duplo em cima da Tabela **Clientes** (aparecerá o Fields Editor), clique com o botão direito do mouse e selecione a opção **Add All Fields**.
- 8) Ainda selecionando a tabela **Clientes** vá até o evento **AfterPost** e digite a seguinte linha de comando:
DBISaveChanges(Clientes.Handle);
- 9) Repita as operações 6, 7 e 8 para todas as tabelas, mudando é claro os suas respectivas propriedades de acordo com de cada uma das tabelas criadas;
- 10) Adicione doze (12) componentes não visuais **DataSource** que estão localizados na palheta **Data Access**;
- 11) Selecione o **DataSource1** e altere a propriedade **DataSet** selecionando a tabela **Clientes**;
- 12) Altere a Propriedade **Name** do **DataSource1** para **DsClientes**;

13) Repita a operação 11 e 12 para todos os DataSources alterando é claro suas propriedades conforme cada tabela;

Após terminar todo este processo o seu DM deverá ficar com a aparência da figura abaixo



Veja como deve ficar a UNIT Tabelas:

unit Tabelas;

interface

uses

SysUtils, Classes, DB, DBTables, DBIProcs;

type

TDM = class(TDataModule)

Clientes: TTable;

Configuracao: TTable;

Dependentes: TTable;

Fitas: TTable;

FitasLocadas: TTable;

Fitass: TTable;

Generos: TTable;

Locacoes: TTable;

Parentesco: TTable;

Produtoras: TTable;

TiposAluguel: TTable;

Usuarios: TTable;

DsClientes: TDataSource;

DsConfiguracao: TDataSource;

DsDependentes: TDataSource;

DsFias: TDataSource;

DsFitasLocadas: TDataSource;

DsFitass: TDataSource;

DsGeneros: TDataSource;

DsLocacoes: TDataSource;

DsParentesco: TDataSource;

DsProdutoras: TDataSource;

DsTiposAluguel: TDataSource;

DsUsuarios: TDataSource;

FitasFit_Codigo: TAutoIncField;

FitasFit_Nome: TStringField;

FitasFit_Categoria: TIntegerField;

FitasFit_Duracao: TTimeField;

FitasFit_SistGravacao: TStringField;

FitasFit_Produtora: TIntegerField;

FitasFit_DubLeg: TStringField;

FitasFit_TipoAluguel: TIntegerField;

FitasFit_Sinopse: TMemoField;

FitasFit_OBS: TMemoField;

DependentesDep_Codigo: TAutoIncField;

DependentesDep_Cliente: TIntegerField;

DependentesDep_Nome: TStringField;

DependentesDep_Parentesco: TIntegerField;

DependentesDep_Apelido: TStringField;

DependentesDep_DataNascimento: TDateField;

ConfiguracaoCon_Codigo: TAutoIncField;

ConfiguracaoCon_NomeLocadora: TStringField;

ConfiguracaoCon_Endereco: TStringField;

ConfiguracaoCon_Cidade: TStringField;

ConfiguracaoCon_Estado: TStringField;

ConfiguracaoCon_CEP: TStringField;

ConfiguracaoCon_Fone1: TStringField;

ConfiguracaoCon_Fone2: TStringField;

ClientesCli_Codigo: TAutoIncField;

ClientesCli_Nome: TStringField;

ClientesCli_Endereco: TStringField;

ClientesCli_Bairro: TStringField;

ClientesCli_Cidade: TStringField;

ClientesCli_Estado: TStringField;

ClientesCli_CEP: TStringField;

ClientesCli_CPF: TStringField;

ClientesCli_Identidade: TStringField;

ClientesCli_OrgEmissor: TStringField;

ClientesCli_CompResidencia: TStringField;

ClientesCli_DataCompResid: TDateField;

ClientesCli_Fone1: TStringField;

ClientesCli_Fone2: TStringField;

ClientesCli_Sexo: TStringField;

ClientesCli_DataNascimento: TDateField;

ClientesCli_Apelido: TStringField;

ClientesCli_OBS: TMemoField;

ClientesCli_Foto: TBlobField;

FitasLocadasFitl_Codigo: TAutoIncField;

FitasLocadasFitl_Fitass: TStringField;

FitasLocadasFitl_ValorAluguel: TCurrencyField;

FitasLocadasFitl_Pago: TStringField;

FitasLocadasFitl_Devolvido: TStringField;

FitasLocadasFitl_HoraDevolvido: TTimeField;

FitasLocadasFitl_DataDevolvido: TDateField;

FitassFits_Codigo: TStringField;

FitassFits_Fita: TIntegerField;

FitassFits_NumCopia: TStringField;

FitassFits_Alugada: TStringField;

GenerosGen_Codigo: TAutoIncField;

```
GenerosGen_Nome: TStringField;
GenerosGen_Abreviacao: TStringField;
GenerosGen_UltimoCodigo: TStringField;
LocacoesLoc_Codigo: TAutoIncField;
LocacoesLoc_Cliente: TIntegerField;
LocacoesLoc_DataAluguel: TDateField;
LocacoesLoc_HoraAluguel: TTimeField;
LocacoesLoc_FitasLocadas: TIntegerField;
LocacoesLoc_Usuario: TIntegerField;
ParentescoPar_Codigo: TAutoIncField;
ParentescoPar_Nome: TStringField;
TiposAluguelTip_Codigo: TAutoIncField;
TiposAluguelTip_Nome: TStringField;
TiposAluguelTip_Valor: TCurrencyField;
UsuariosUsu_Codigo: TAutoIncField;
UsuariosUsu_Nome: TStringField;
UsuariosUsu_NomeUsuario: TStringField;
UsuariosUsu_Senha: TStringField;
UsuariosUsu_Endereco: TStringField;
UsuariosUsu_Cidade: TStringField;
UsuariosUsu_Estado: TStringField;
UsuariosUsu_Fone1: TStringField;
ClientesCli_NumeroCasa: TStringField;
ClientesCli_Complemento: TStringField;
ClientesCli_DataEmissao: TDateField;
procedure ClientesAfterPost(DataSet: TDataSet);
procedure ConfiguracaoAfterPost(DataSet: TDataSet);
procedure DependentesAfterPost(DataSet: TDataSet);
procedure FitasAfterPost(DataSet: TDataSet);
procedure FitasLocadasAfterPost(DataSet: TDataSet);
procedure FitassAfterPost(DataSet: TDataSet);
procedure GenerosAfterPost(DataSet: TDataSet);
procedure LocacoesAfterPost(DataSet: TDataSet);
procedure ParentescoAfterPost(DataSet: TDataSet);
procedure ProdutorasAfterPost(DataSet: TDataSet);
procedure TiposAluguelAfterPost(DataSet: TDataSet);
procedure UsuariosAfterPost(DataSet: TDataSet);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  DM: TDM;

implementation

{$R *.dfm}

procedure TDM.ClientesAfterPost(DataSet: TDataSet);
begin
  DBISaveChanges(Clientes.Handle);
end;

procedure TDM.ConfiguracaoAfterPost(DataSet:
  TDataSet);
begin
  DBISaveChanges(Configuracao.Handle);
end;

procedure TDM.DependentesAfterPost(DataSet:
  TDataSet);
begin
  DBISaveChanges(Dependentes.Handle);
end;

procedure TDM.FitasAfterPost(DataSet: TDataSet);
begin
  DBISaveChanges(Fitas.Handle);
end;

procedure TDM.FitasLocadasAfterPost(DataSet:
  TDataSet);
begin
  DBISaveChanges(FitasLocadas.Handle);
end;

procedure TDM.FitassAfterPost(DataSet: TDataSet);
begin
  DBISaveChanges(Fitass.Handle);
end;

procedure TDM.GenerosAfterPost(DataSet: TDataSet);
begin
  DBISaveChanges(Generos.Handle);
end;

procedure TDM.LocacoesAfterPost(DataSet: TDataSet);
begin
  DBISaveChanges(Locacoes.Handle);
end;

procedure TDM.ParentescoAfterPost(DataSet: TDataSet);
begin
  DBISaveChanges(Parentesco.Handle);
end;

procedure TDM.ProdutorasAfterPost(DataSet: TDataSet);
begin
  DBISaveChanges(Produtoras.Handle);
end;

procedure TDM.TiposAluguelAfterPost(DataSet:
  TDataSet);
begin
  DBISaveChanges(TiposAluguel.Handle);
end;

procedure TDM.UsuariosAfterPost(DataSet: TDataSet);
begin
  DBISaveChanges(Usuarios.Handle);
end;

end.
```

- 3º Passo (Criando e Configurando Forms para Cadastro) -

Criando o Form para o Cadastro de Clientes

- 1) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;
- 2) Salve o novo Form com o Nome de **“CadastroClientes.pas”**;
- 3) Altere a propriedade **Name** do Form para **FormCadastroClientes**;
- 4) Usando os componentes da palheta **DataControls** crie seu form de cadastro de clientes, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado nas figuras abaixo):

The figure displays five screenshots of the 'Cadastro de Clientes' form, illustrating the layout for different data sections:

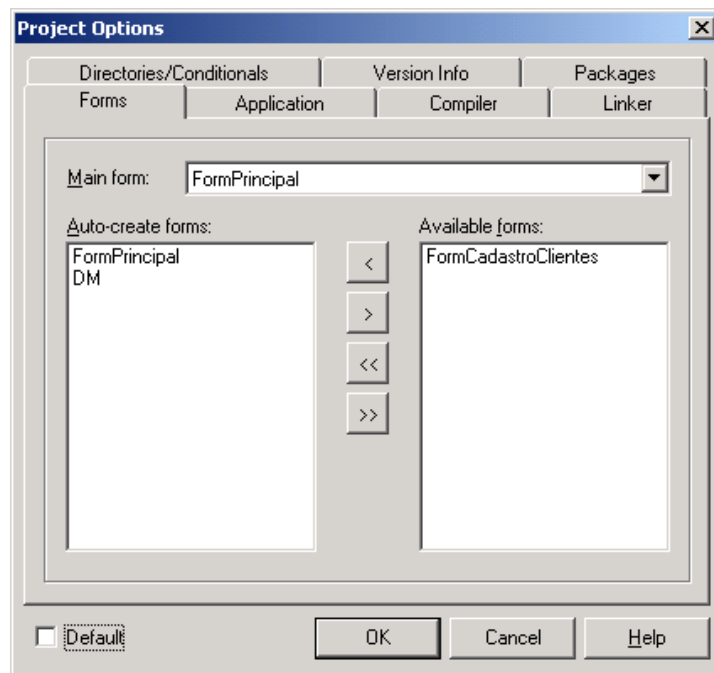
- Cliente Tab:** Includes fields for 'Código do Cliente' (DBE diCodigo), 'Nome do Cliente' (DBE diNome), 'Apelido' (DBE diApelido), 'Data de Nascimento' (DBE diDNasc), 'Telefone Residência' (DBE diTResidencia), 'Telefone - 02' (DBE diTel02), 'Sexo' (radio buttons for M - Masculino and F - Feminino), 'Data da Entrega' (DBE diDEntrega), and a checkbox for 'Comprovante de Residência Entregue?'. It also features a 'Foto do Cliente' area with an 'OpenPictureDialog1' button and a '(DBImageFoto)' label.
- Endereço Tab:** Includes fields for 'Endereço' (DBE diEndereco), 'Número' (DBE diNumero), 'Complemento' (DBE diComplemento), 'Bairro' (DBE diBairro), 'Cidade' (DBE diCidade), 'Estado' (DB dropdown), and 'CEP' (DBE diCEP).
- Documentos Tab:** Includes fields for 'Identidade' (DBE diIdentidade), 'CPF' (DBE diCPF), 'Órgão Emissor' (DBE diOE), and 'Data da Emissão' (DBE diEmissa).
- Observações Tab:** Features a large 'DBMemoOBS' text area.
- Dependentes Tab:** Includes a 'Novo Dependente' section with a grid and a search section labeled 'Pesquisar Por...' with radio buttons for '...Nome' and '...Apelido', and an 'EditPesquisa' text field.

Each screenshot shows a set of control buttons on the right: 'Inserir', 'Gravar', 'Cancelar', 'Excluir', and 'Pesquisar', along with a 'Fechar' button at the bottom right.

- 5) No exemplo acima altere as seguintes propriedades:

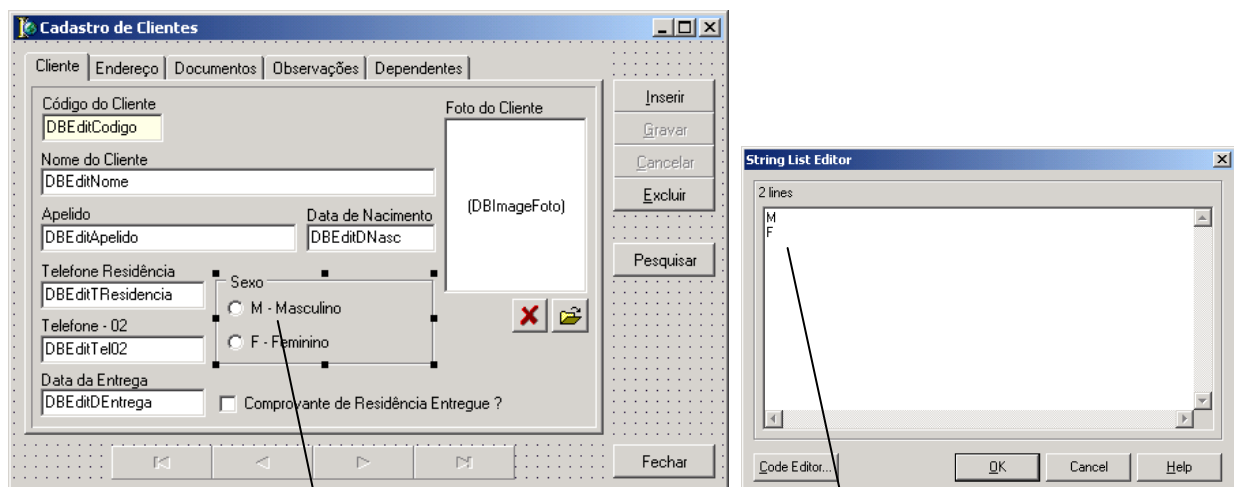
- Selecione o FormCadastroClientes e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;
- Selecione o DBNavigator e Altere a propriedade **Flat** para **True**;
- Na aba **Dependentes** Selecione o **DBGridDependentes** e altere a propriedade **ReadOnly** para **True**;
- Selecione o **DBImage** na aba **Cliente** e altere a propriedade **TabStop** para **False**;
- Selecione o **DBEditCodigo** na aba **Cliente** e altere a propriedade **ReadOnly** para **True**;

- 6) Clique em **Project/Options...** passe o **FormCadastroClientes** para o Lado **Available Forms**; Veja figura abaixo:



Lembre-se que o item anterior serve para deixar o FormCadastroClientes disponível para ser criado futuramente na memória, ou seja ele só será criando na memória, quando for solicitado pelo usuário

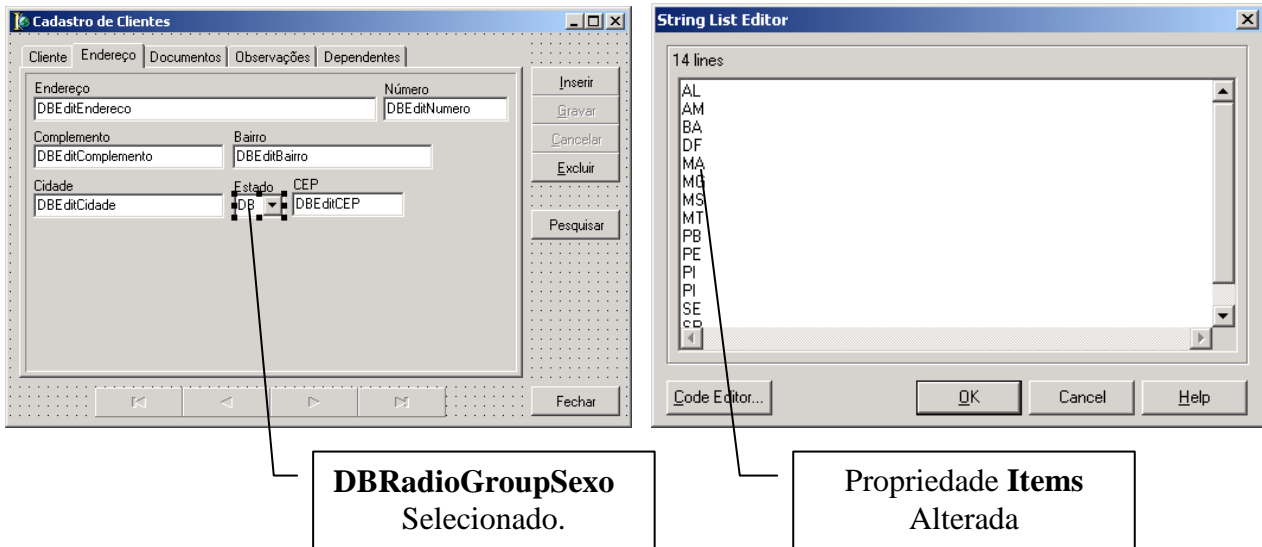
- 7) Clique no Menu **File>Use Unit**, escolha a unit **Tabelas** e clique em **OK**;
- 8) Na Aba Cliente do FormCadastroClientes Selecione o **DBRadioGroupSexo**, Selecione a propriedade **Values**, clique nos três pontinhos e Adicione os valores **M** e **F**, um abaixo do outro como mostram as figuras:



DBRadioGroupSexo
Selecionado.

Propriedade **Values**
Alterada.

- 9) Agora selecione cada um dos componentes e altere suas propriedades **DataSource** e **DataField** para suas respectivas tabelas (DataSource) e Campos (DataField). Ex: O DBEditNome deve ter seu **DataSource** ligado à **DM.DsClientes** e seu **DataField** ligado à **Cli_Nome**, e assim por diante;
- 10) Faça o mesmo (item 9) para as abas Endereço, Documentos e Observações;
- 11) Após ter completado todas as ligações, Selecione o **DBComboboxEstado**, situado na Aba **Endereço** e altere a propriedade **Items** Adicionando todos os estados como mostra a figura:



- 12) Ainda com o **DBComboBoxEstado** selecionado Altera a propriedade **Sorted** para **True**;
- 13) Selecione o **DBNavigator** e altere a propriedade **DataSource** selecionando **"DM.DsClientes"**; Selecione o **DBCheckBox1** e altere a propriedade **ValueChecked** para **S** e a propriedade **ValueUnchecked** para **N**;
- 14) Crie (na unit CadastroClientes) as seguintes procedures:

```

procedure TFormCadastroClientes.INSERIR_ALTERAR;
Begin
    BitInserir.Enabled:=False;
    BitGravar.Enabled:=True;
    BitCancelar.Enabled:=True;
    BitExcluir.Enabled:=False;
    BitPesquisar.Enabled:=False;
    BitFechar.Enabled:=False;
    DBNavigator.Enabled:=False;
    BitDependentes.Enabled:=False;
    RadioGroupPesquisa.Enabled:=False;
    EditPesquisa.Enabled:=False;
end;

procedure TFormCadastroClientes.GRAVAR_CANCELAR;
Begin
    BitInserir.Enabled:=True;
    BitGravar.Enabled:=False;
    BitCancelar.Enabled:=False;
    BitExcluir.Enabled:=True;
    BitPesquisar.Enabled:=True;
    BitFechar.Enabled:=True;
    DBNavigator.Enabled:=True;
    BitDependentes.Enabled:=True;
    RadioGroupPesquisa.Enabled:=True;
    EditPesquisa.Enabled:=True;
    FormCadastroClientes.Caption:='[Cliente] = '+DM.ClientesCli_Nome.Value;
end;

```

- 15) Declare as procedures antes da sessão **PRIVATE** da unit;

```

Procedure INSERIR_ALTERAR;
Procedure GRAVAR_CANCELAR;

```

- 16) No evento **OnClick** do Botão **Inserir** digite as seguintes linhas de comando:

```

Try
    INSERIR_ALTERAR;
    DM.Clientes.Insert;
    PageControl1.ActivePageIndex:=0;
    DBEditNome.SetFocus;
Except

```



```
Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Contate seu  
programador...','Erro...','Mb_IconError+Mb_Ok);  
GRAVAR_CANCELAR;  
BitInserir.SetFocus;  
end;
```

17) No evento **OnClick** do Botão **Gravar** digite as seguintes linhas de comando:

```
Try  
GRAVAR_CANCELAR;  
DM.Clientes.Post;  
BitInserir.SetFocus;  
Except  
Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será  
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok);  
DM.Clientes.Cancel;  
GRAVAR_CANCELAR;  
BitInserir.SetFocus;  
end;
```

18) No evento **OnClick** do Botão **Cancelar** digite as seguintes linhas de comando:

```
Try  
GRAVAR_CANCELAR;  
DM.Clientes.Cancel;  
BitInserir.SetFocus;  
Except  
Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será  
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok);  
DM.Clientes.Cancel;  
BitCancelar.SetFocus;  
end;
```

19) No evento **OnClick** do Botão **Excluir** digite as seguintes linhas de comando:

```
Try  
If DM.Clientes.RecordCount > 0 Then  
Begin  
If Application.MessageBox(Pchar('Deseja realmente apagar este  
cliente?'+#13+'[ '+DM.ClientesCli_Nome.Value+ ' ]'),'Pergunta...','Mb_IconExclamation+Mb_YesNo) = MrYes  
Then  
DM.Clientes.Delete;  
end  
else  
Application.MessageBox('Não há cliente para ser excluído...','Aviso...','Mb_IconInformation+Mb_Ok);  
Except  
Application.MessageBox('Erro inesperado...'+#13+'Este programa será fechado...'+#13+'Contate seu  
programador...','Erro...','Mb_IconError+Mb_Ok);  
Application.Terminate;  
end;
```

20) Vá até o DataModule Selecione a tabela Clientes e no Evento AfterEdit digite a seguinte linha de comando:

```
If FormCadastroClientes <> Nil Then  
FormCadastroClientes.INSERIR_ALTERAR;
```

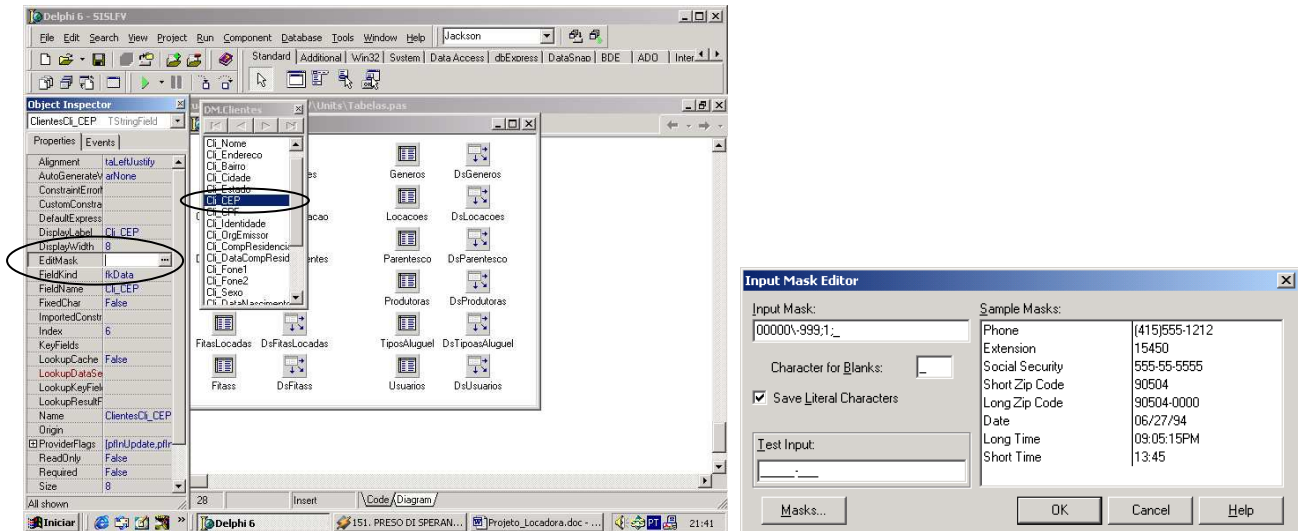
21) Vá até o **FormPrincipal** e no evento **onClick** do Menu **Cadastro>Clientes** digite as seguintes linhas de comando:

```
DM.Clientes.Open;  
FormCadastroClientes:=TFormCadastroClientes.Create(Application);  
FormCadastroClientes.PageControl1.ActivePageIndex:=0;  
FormCadastroClientes.Caption:='[Cliente] = '+DM.ClientesCli_Nome.Value;  
FormCadastroClientes.ShowModal;  
FormCadastroClientes.Free;  
FormCadastroClientes:=Nil;  
DM.Clientes.Close;
```

- 22) Compile e teste o programa, lembre-se que as datas e CEP estão sem máscara e a opção de foto ainda está desativada;

-- Ajustando as Máscaras dos Campos:

- 23) Vá até o **DataModule**, faça um duplo clique na tabela clientes selecione o campo **Cli_CEP** e configure a propriedade **EditMask**, lembrando de **desmarcar** a opção **Save Literal Characters**, veja a figura abaixo;



- 24) Faça o mesmo para todos os campos que são datas, mas desta vez lembre-se de **não desmarcar** a opção **Save Literal Characters**;

-- Fazendo funcionar a inserção da foto do Cliente:

- 25) Vá até a unit CadastroClientes e na Cláusula uses adicione a unit DB;
- 26) No evento **OnClick** do botão **BitAbrir** digite as seguintes linhas de comando:

```
If DM.Clientes.State = DsBrowse Then
Begin
  Try
    DM.Clientes.Edit;
  Except
    Application.MessageBox('Erro inesperado...'+#13'Não É possível inserir uma foto neste
momento'+#13+'Contate seu programador...','Erro...',Mb_IconError+Mb_Ok);
    ABORT;
  end;
end;

If OpenPictureDialog1.Execute Then
  DBImageFoto.Picture.LoadFromFile(OpenPictureDialog1.FileName);
```

- 27) No evento **OnClick** do botão **BitApagar** digite as seguintes linhas de comando:

```
If Application.MessageBox('Tem certeza que deseja apagar a foto deste
cliente?', 'Pergunta...', Mb_IconQuestion+Mb_YesNo) = MrYes Then
Begin
  If DM.Clientes.State = DsBrowse Then
  Begin
    Try
      DM.Clientes.Edit;
    Except
```

```
Application.MessageBox('Erro inesperado...'+#13'Não É possível apagar a foto neste  
momento'+#13+'Contate seu programador...','Erro...',Mb_IconError+Mb_Ok);  
ABORT;  
end;  
end;  
  
DBImageFoto.Picture:=Nil;  
end;
```

28) No evento **OnClick** do botão **BitEsticar** digite as seguintes linhas de comando:

```
DBImageFoto.Stretch := not DBImageFoto.Stretch;
```

29) Compile e Teste o Projeto; (Lembre-se que até aí a aba de dependentes ainda não tem funcionamento nenhum)

Criando o Form para o Cadastro de Dependentes

- 01) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;
- 02) Salve o novo Form com o Nome de **“CadastroDependentes.pas”**;
- 03) Altere a propriedade **Name** do Form para **FormCadastroDependentes**;
- 04) Usando os componentes da palheta **DataControls** crie seu form de cadastro de clientes, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado nas figuras abaixo):

05) No exemplo acima altere as seguintes propriedades:

- Selecione o FormCadastroDependentes e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;
- Selecione o DBNavigator e Altere a propriedade **Flat** para **True**;
- Selecione o **DBEditCodigo** na aba **Dependente** e altere a propriedade **ReadOnly** para **True**;

06) Clique em **Project/Options...** passe o **FormCadastroDependentes** para o Lado **Available Forms**; Veja figura abaixo:

07) Lembre-se que o item anterior serve para deixar o FormCadastroDependentes disponível para ser criado futuramente na memória, ou seja ele só será criado na memória, quando for solicitado pelo usuário;

08) Agora selecione cada um dos componentes e altere suas propriedades **DataSource** e **DataField** para suas respectivas tabelas (DataSource) e Campos (DataField).
Ex: O DBEditNome deve ter seu **DataSource** ligado à **DM.DsDependentes** e seu **DataField** ligado à **Dep_Nome**, e assim por diante;

09) Selecione o DBNavigator e altere a propriedade **DataSource** selecionando “**DM.DsDependentes**”;

10) Selecione o **DbLookUpComboBoxParentesco** e altere as seguintes propriedades:

ListSource = DM.DsParentesco
ListField = Par_Nome
KeyField = Par_Codigo

11) Crie (na unit CadastroDependentes) as seguintes procedures:

```
procedure TFormCadastroDependentes.INSERIR_ALTERAR;
Begin
  BitInserir.Enabled:=False;
  BitGravar.Enabled:=True;
  BitCancelar.Enabled:=True;
  BitExcluir.Enabled:=False;
  BitFechar.Enabled:=False;
  DBNavigator.Enabled:=False;
end;

procedure TFormCadastroDependentes.GRAVAR_CANCELAR;
Begin
  BitInserir.Enabled:=True;
  BitGravar.Enabled:=False;
  BitCancelar.Enabled:=False;
  BitExcluir.Enabled:=True;
  BitFechar.Enabled:=True;
  DBNavigator.Enabled:=True;
end;
```

12) Declare as procedures antes da sessão PRIVATE da unit;

Procedure INSEIRIR_ALTERAR;
Procedure GRAVAR_CANCELAR;

13) No evento **OnClick** do Botão **Inserir** digite as seguintes linhas de comando:

```
Try
  INSEIRIR_ALTERAR;
  DM.Dependentes.Insert;
  DBEditNome.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Contate seu
programador...','Erro...','Mb_IconError+Mb_Ok');
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

14) No evento **OnClick** do Botão **Gravar** digite as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Dependentes.Post;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok');
  DM.Dependentes.Cancel;
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

15) No evento **OnClick** do Botão **Cancelar** digite as seguintes linhas de comando:

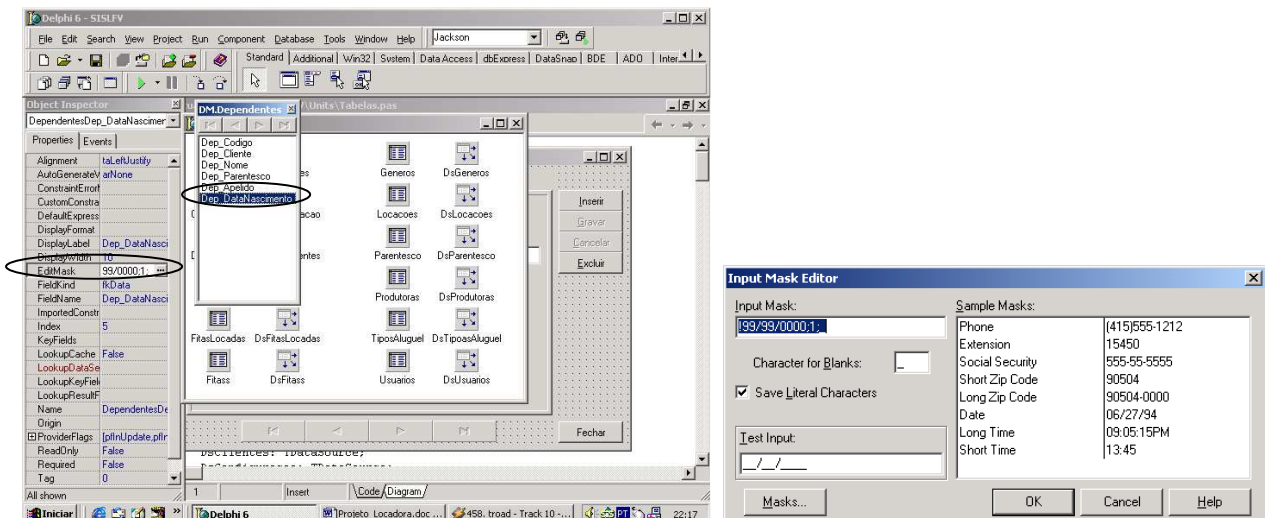
```
Try
  GRAVAR_CANCELAR;
  DM.Dependentes.Cancel;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...',Mb_IconError+Mb_Ok);
  DM.Dependentes.Cancel;
  BitCancelar.SetFocus;
end;
```

16) No evento **OnClick** do Botão **Excluir** digite as seguintes linhas de comando:

```
Try
  If DM.Dependentes.RecordCount > 0 Then
    Begin
      If Application.MessageBox(Pchar('Deseja realmente apagar este
cliente?'+#13+'['+DM.DependentesDep_Nome.Value+']'),'Pergunta...',Mb_IconExclamation+Mb_YesNo) =
MrYes Then
        DM.Dependentes.Delete;
      end
    else
      Application.MessageBox('Não há dependentes para ser excluído...'
,'Aviso...',Mb_IconInformation+Mb_Ok);
    Except
      Application.MessageBox('Erro inesperado...'+#13+'Este programa será fechado...'+#13+'Contate seu
programador...','Erro...',Mb_IconError+Mb_Ok);
      Application.Terminate;
    end;
```

-- Ajustando as Máscaras dos Campos:

17) Vá até o **DataModule**, faça um duplo clique na tabela **Dependentes** selecione o campo **Dep_DataNascimento** e configure a propriedade **EditMask**, lembrando de **marcar** a opção **Save Literal Characters**, veja a figura abaixo;



18) Vá até o **DataModule** selecione a tabela **Dependentes** e no Evento **AfterEdit** digite a seguinte linha de comando:

```
If FormCadastroDependentes <> Nil Then
  FormCadastroDependentes.INSERIR_ALTERAR;
```

- Vá até o evento **AfterScroll** da Tabela **Clientes** e digite as seguintes linha de comando:

```
If FormCadastroClientes <> Nil Then
  FormCadastroClientes.Caption:='[Cliente] '+DM.ClientesCli_Nome.Value;
```

19) Vá até o **FormPrincipal** e no evento **onClick** do Menu **Cadastro>Clientes** digite as seguintes linhas de comando:

```
DM.Clientes.Open;  
DM.Dependentes.Open;  
FormCadastroClientes:=TFormCadastroClientes.Create(Application);  
FormCadastroClientes.PageControl1.ActivePageIndex:=0;  
FormCadastroClientes.Caption:='[Cliente] = '+DM.ClientesCli_Nome.Value;  
FormCadastroClientes.ShowModal;  
FormCadastroClientes.Free;  
FormCadastroClientes:=Nil;  
DM.Dependentes.Close;  
DM.Clientes.Close;
```

20) Vá até o **FormCadastroClientes** na aba **Dependentes** e no **Evento OnClick** do botão **Novo Dependente** e digite as seguintes linhas de comando:

```
FormCadastroDependentes:=TFormCadastroDependentes.Create(Application);  
FormCadastroDependentes.ShowModal;  
FormCadastroDependentes.Free;  
FormCadastroDependentes:=Nil;
```

- Criando um Campo Virtual (Tipo LookUp)

21) Vá até o **DataModule**, faça um duplo clique na tabela **Dependentes** (o editor de campos abrirá), clique com o botão direito do mouse no editor de campos e escolha a opção **New Field...**;

22) Preencha a tela como mostrado na figura abaixo:

Nome do Campo que está sendo criado

Tamanho do campo que está sendo criado

Tipo do Campo que está sendo Criado

Tipo da conexão do Campo

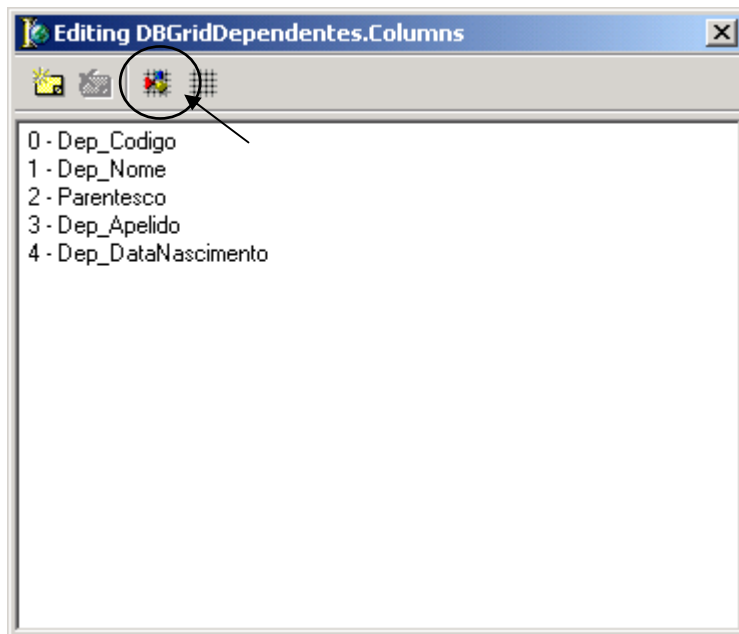
Campo da tabela atual que faz referência com a outra

Tabela de onde virão as informações para o novo campo que está sendo criado

Chave primária da tabela de onde virão as informações para o novo campo

Campo que aparecerá como resultado do LookUp

23) Volte para o **FormCadastroClientes** na aba **Dependentes**, altere a propriedade **DataSource** selecionando **DM.DsDependentes**, faça um duplo clique no **DBGrid** (aparecerá o editor de colunas), clique no **botão colorido** (como mostra a figura abaixo), note que aparecerão todos os capôs da tabela Dependentes, apague os campos que não forem necessários serem visualizados pelo usuário do programa;



- 24) Selecione o **RadioGroupPesquisa** (certifique-se que a propriedade **ItemIndex = 0**), vá até o evento **OnClick** e digite as seguintes linhas de comando:

```
Case RadioGroupPesquisa.ItemIndex of  
  0: DM.Dependentes.IndexName:='idxNome';  
  1: DM.Dependentes.IndexName:='idxApelido';  
end;
```

```
EditPesquisa.Color:=clWindow;  
EditPesquisa.Enabled:=True;  
EditPesquisa.SetFocus;
```

- 25) Selecione o EditPesquisa e altere as seguintes propriedades:

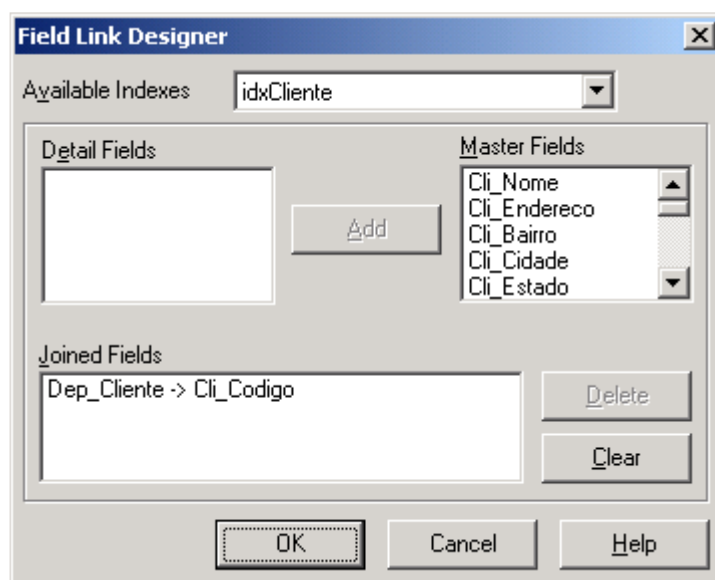
Color = clBtnFace
Enabled = False

- 26) Vá até o evento onlick do EditPesquisa e digite as seguintes linhas de comando:

```
DM.Dependentes.FindNearest([EditPesquisa.Text]);
```

- Criando um MasterSource (Relação Cliente/Dependentes)

- 30) Vá até o **DataModule** e Selecione a tabela **Dependentes**, Altere a opção **MasterSource** para **DsClientes**; Selecione a propriedade **MasterFields** e clique nos 03 pontinhos; Veja afigura abaixo:



31) Para deixar a janela do mesmo jeito da apresentada acima, clique no combobox **Available Indexes** e Selecione **IdxCliente**, logo após selecione **Dep_Cliente** na caixa **Detail Fields**, e depois selecione **Cli_Código** na caixa **Master Fields**, pronto, agora é só clicar no botão **Add** e em seguida, no botão **OK**;

→ Pronto, Agora compile e rode seu programa, a parte de Cadastro de Cliente/Dependentes está completa....

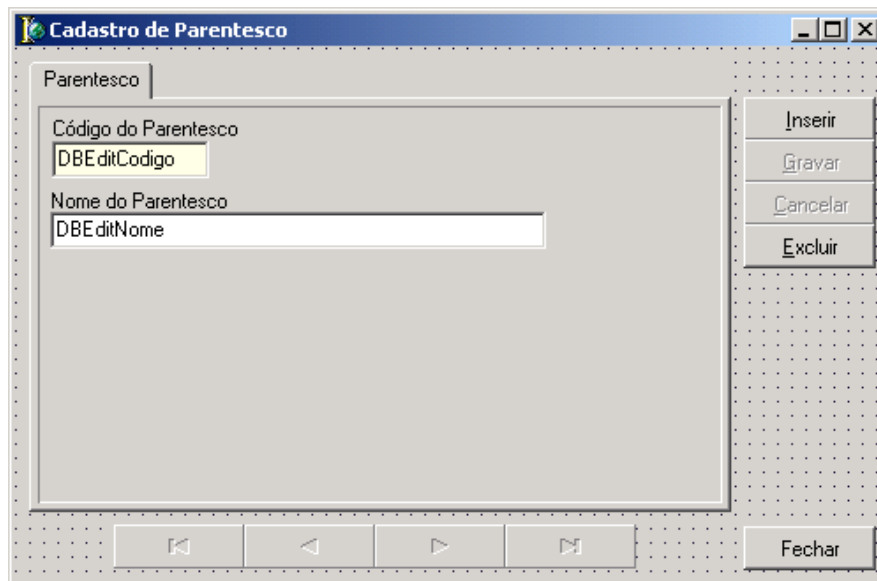
Criando o Form para o Cadastro de Parentesco

01) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;

02) Salve o novo Form com o Nome de **“CadastroParentesco.pas”**;

03) Altere a propriedade **Name** do Form para **FormCadastroParentesco**;

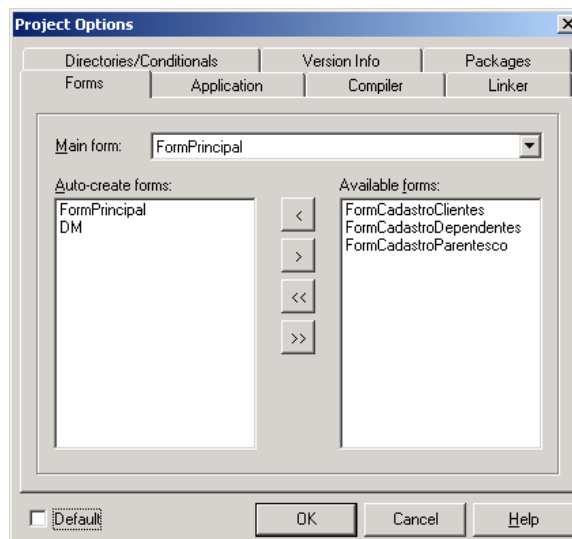
04) Usando os componentes da palheta **DataControls** crie seu form de cadastro de clientes, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado nas figuras abaixo):



05) No exemplo acima altere as seguintes propriedades:

- Selecione o **FormCadastroParentesco** e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;
- Selecione o **DBNavigator** e Altere a propriedade **Flat** para **True**;
- Selecione o **DBEditCodigo** na aba **Parentesco** e altere a propriedade **ReadOnly** para **True**;

06) Clique em **Project/Options...** passe o **FormCadastroParentesco** para o Lado **Available Forms**; Veja figura abaixo:



07) Lembre-se que o item anterior serve para deixar o **FormCadastroParentesco** disponível para ser criado futuramente na memória, ou seja ele só será criando na memória, quando for solicitado pelo usuário;

- 08) Agora selecione cada um dos componentes e altere suas propriedades **DataSource** e **DataField** para suas respectivas tabelas (DataSource) e Campos (DataField). Ex: O DBEditNome deve ter seu **DataSource** ligado à **DM.DsParentesco** e seu **DataField** ligado à **Par_Nome**, e assim por diante;
- 09) Selecione o DBNavigator e altere a propriedade **DataSource** selecionando “**DM.DsParentesco**”;
- 10) Crie (na unit CadastroParentesco) as seguintes procedures:

```
procedure TFormCadastroParentesco.INSERIR_ALTERAR;
Begin
  BitInserir.Enabled:=False;
  BitGravar.Enabled:=True;
  BitCancelar.Enabled:=True;
  BitExcluir.Enabled:=False;
  BitFechar.Enabled:=False;
  DBNavigator.Enabled:=False;
end;

procedure TFormCadastroParentesco.GRAVAR_CANCELAR;
Begin
  BitInserir.Enabled:=True;
  BitGravar.Enabled:=False;
  BitCancelar.Enabled:=False;
  BitExcluir.Enabled:=True;
  BitFechar.Enabled:=True;
  DBNavigator.Enabled:=True;
end;
```

- 11) Declare as procedures antes da sessão PRIVATE da unit;

```
Procedure INSEIR_ALTERAR;
Procedure GRAVAR_CANCELAR;
```

- 12) No evento **OnClick** do Botão **Inserir** digite as seguintes linhas de comando:

```
Try
  INSEIR_ALTERAR;
  DM.Dependentes.Insert;
  DBEditNome.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Contate seu
programador...','Erro...',Mb_IconError+Mb_Ok);
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- 13) No evento **OnClick** do Botão **Gravar** digite as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Parentesco.Post;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...',Mb_IconError+Mb_Ok);
  DM.Parentesco.Cancel;
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

14) No evento **OnClick** do Botão **Cancelar** digite as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Parentesco.Cancel;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok);
  DM.Parentesco.Cancel;
  BitCancelar.SetFocus;
end;
```

15) No evento **OnClick** do Botão **Excluir** digite as seguintes linhas de comando:

```
Try
  If DM.Parentesco.RecordCount > 0 Then
    Begin
      If Application.MessageBox(Pchar('Deseja realmente apagar este tipo de
parentesco?'+#13+'['+DM.ParentescoPar_Nome.Value+']'),'Pergunta...','Mb_IconExclamation+Mb_YesNo) =
MrYes Then
        DM.Parentesco.Delete;
      end
    else
      Application.MessageBox('Não há parentesco para ser excluído...','Aviso...','Mb_IconInformation+Mb_Ok);
    Except
      Application.MessageBox('Erro inesperado...'+#13'Este programa será fechado...'+#13+'Contate seu
programador...','Erro...','Mb_IconError+Mb_Ok);
      Application.Terminate;
    end;
```

16) No evento **OnClick** do Botão **Fechar** digite as seguintes linhas de comando:

```
Close;
```

17) Vá até o **DataModule** e selecione a tabela **Parentesco**, selecione o evento **AfterEdit** e digite as seguintes linhas de comando:

```
If FormCadastroParentesco <> Nil Then
  FormCadastroParentesco.INSERIR_ALTERAR;
```

18) Vá até o evento **OnClick** do Menu **Cadastro>Parentesco** e digite as seguintes linhas de comando:

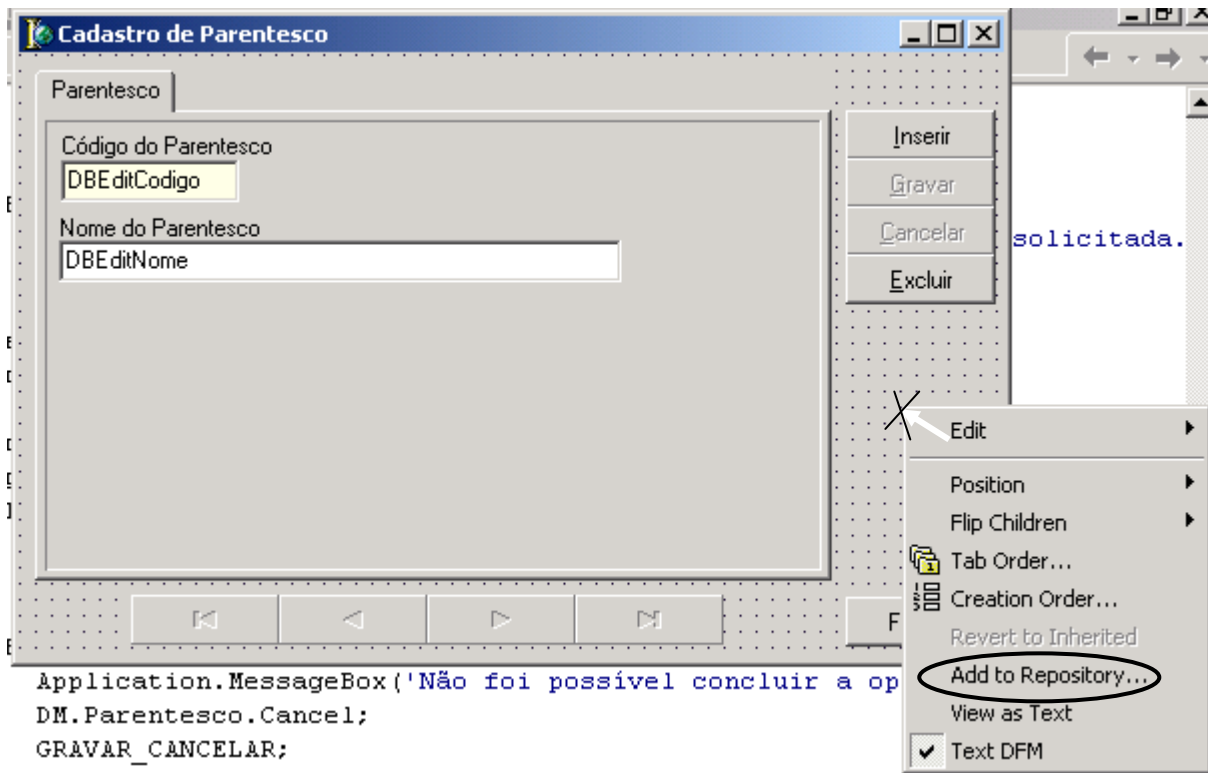
```
DM.Parentesco.Open;
FormCadastroParentesco:=TFormCadastroParentesco.Create(Application);
FormCadastroParentesco.ShowModal;
FormCadastroParentesco.Free;
FormCadastroParentesco:=Nil;
DM.Parentesco.Close;
```

→ Pronto, agora vc pode testar seu projeto cadastrando os Parentescos, os clientes e os dependentes....

Criando o Form para o Cadastro de Tipos de Aluguel

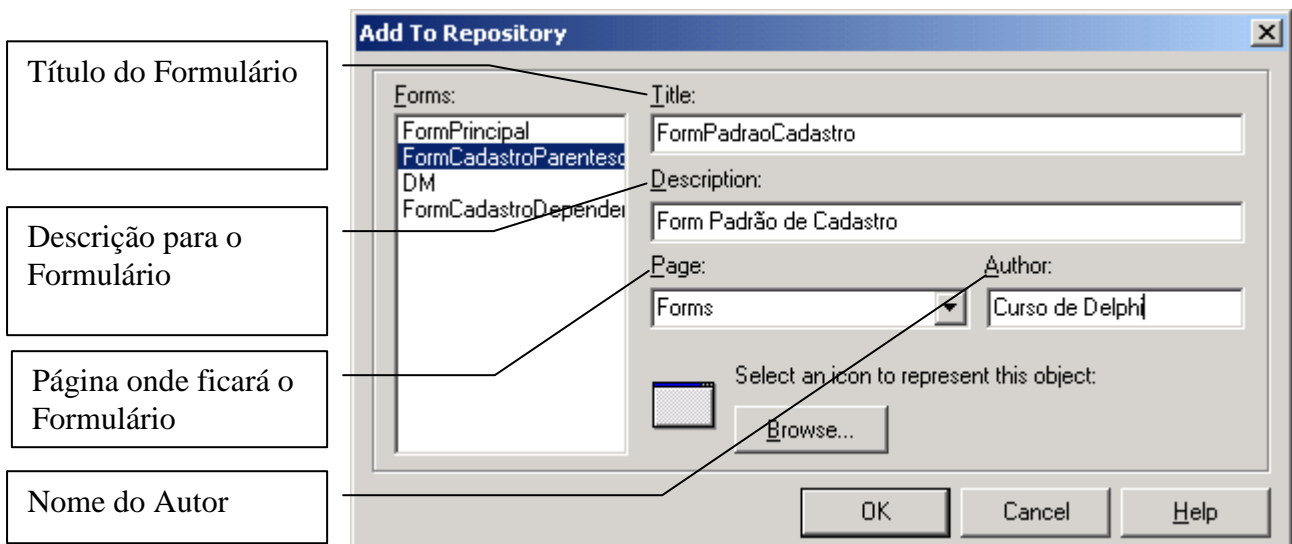
→ Desta vez vamos criar o Form de Cadastro de Tipos de Aluguel de uma forma diferente, vamos conhecer como usar o Repositório do Delphi, uma ferramenta muito importante para os seus projetos, pois com ele vc economizará linhas e linhas de digitação.

01) Selecione o **FormCadastroParentesco** e clique com o botão direito do mouse sobre qualquer parte do Form (Lembre-se que tem que ser um clique no Form, e não nos componentes do Form), como mostra a figura abaixo:



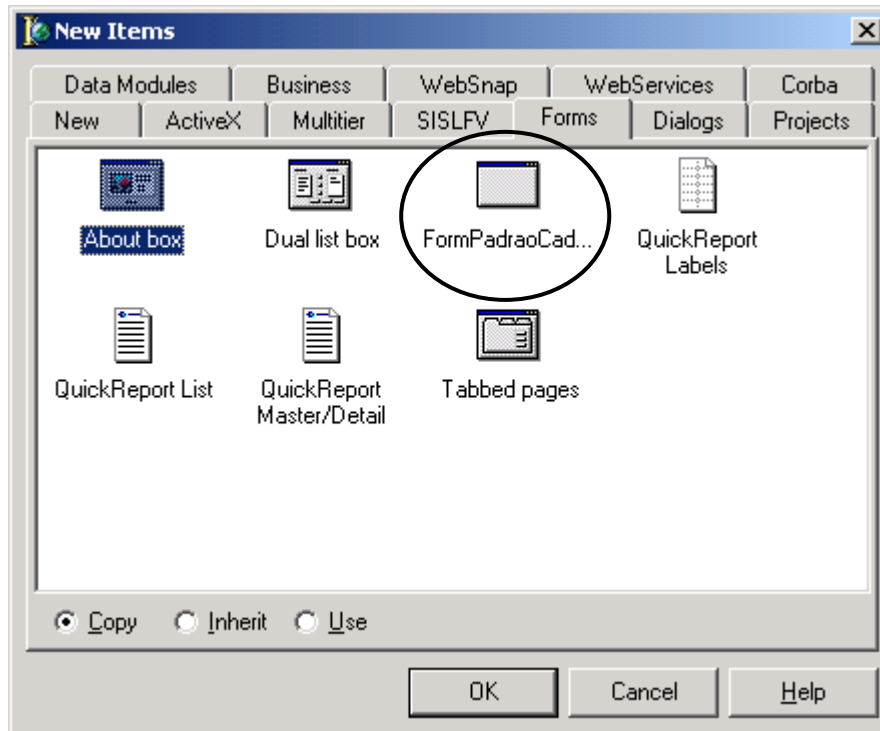
02) Logo após clicar com o botão direito do mouse aparecerá um Menu PopUp como mostra a figura acima, daí é só clicar em **Add to Repository**;

03) Logo aparecerá uma janela, daí é só preencher como é mostrado na figura Abaixo e clicar em OK:

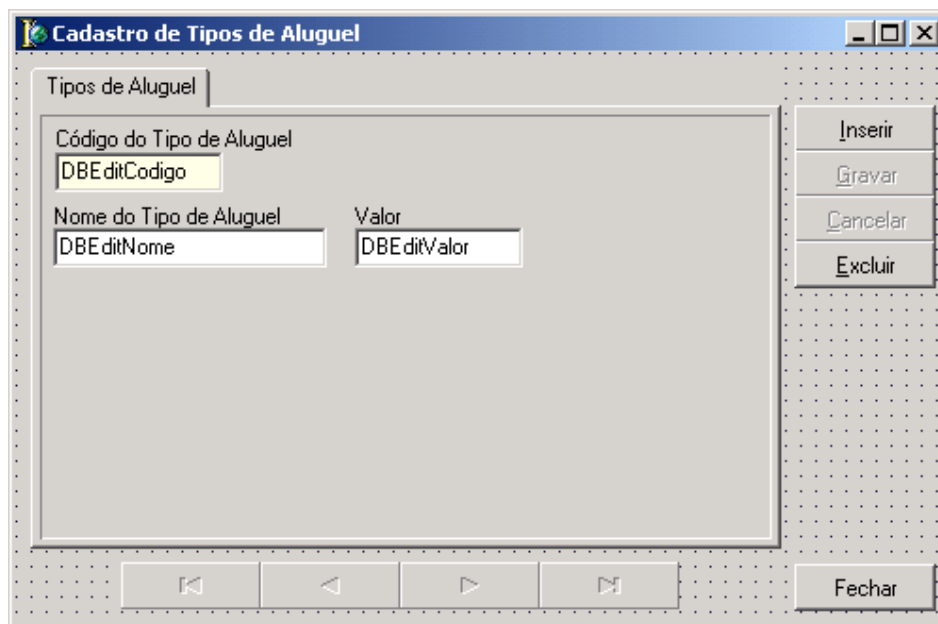


04) Logo após clicar em OK na janela mostrada na figura anterior, clique no menu **File>Close All**, para fechar todo o seu projeto; (estou fazendo isso apenas para que vc não se confunda com o que iremos fazer)

05) Abra o seu projeto novamente, e logo após abri-lo clique no menu **File>New>Other...**, fazendo isso aparecerá a janela mostrada na figura abaixo:



- 06) A janela mostrada na figura acima é o repositório onde vc depositou uma cópia do **FormCadastroParentesco**, daí então é só selecionar a aba **Forms** (já que foi ela que vc escolheu para depositar a cópia do formulário) e selecionar o item **FormPadraoCadastro** e clicar no botão OK.
- 07) Logo após que vc clicar no botão OK aparecerá um form igual ao do FormCadastroParentesco, ou seja, uma cópia fiel do form que vc já criou, daí então é só deixá-lo como mostra a figura abaixo adicionando mais um (01) Dbedit:



- 08) Lembre-se que vc precisará alterar o **Caption** do Form para “**Cadastro de Tipos de Aluguel**”, o **Name** do Form para “**FormCadastroTiposAluguel**”;
- 09) Salve-o com o nome “**CadastroTiposAluguel.pas**”;
- 10) Clique em **Project/Options...** passe o **FormCadastroTiposAluguel** para o Lado **Available Forms**;
- 11) Lembre-se que o item anterior serve para deixar o FormCadastroTiposAluguel disponível para ser criado futuramente na memória, ou seja ele só será criando na memória, quando for solicitado pelo usuário;
- 12) Agora selecione cada um dos componentes e altere suas propriedades **DataSource** e **DataField** para suas respectivas tabelas (DataSource) e Campos (DataField). Ex: O DBEditNome deve ter seu **DataSource** ligado à **DM.DsTiposAluguel** e seu **DataField** ligado à **Tip_Nome**, e assim por diante;

13) Selecione o DBNavigator e altere a propriedade **DataSource** selecionando “**DM.DsTiposAluguel**”;

14) Agora basta alterar a função de cada um dos botões INSERIR, GRAVAR, CANCELAR, EXCLUIR, veja a seguir:

- No evento **OnClick** do Botão **Inserir** altere para as seguintes linhas de comando:

```
Try
  INSERIR_ALTERAR;
  DM.TiposAluguel.Insert;
  DBEditNome.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Contate seu
programador...','Erro...','Mb_IconError+Mb_Ok);
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Gravar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.TiposAluguel.Post;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok);
  DM.TiposAluguel.Cancel;
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Cancelar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.TiposAluguel.Cancel;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok);
  DM.TiposAluguel.Cancel;
  BitCancelar.SetFocus;
end;
```

- No evento **OnClick** do Botão **Excluir** altere para as seguintes linhas de comando:

```
Try
  If DM.TiposAluguel.RecordCount > 0 Then
    Begin
      If Application.MessageBox(Pchar('Deseja realmente apagar este tipo de
aluguel?'+#13+'['+DM.TiposAluguelTip_Nome.Value+']'),'Pergunta...','Mb_IconExclamation+Mb_YesNo) =
MrYes Then
        DM.TiposAluguel.Delete;
      end
    else
      Application.MessageBox('Não há tipos de aluguel para ser excluído...'
,'Aviso...','Mb_IconInformation+Mb_Ok);
    Except
      Application.MessageBox('Erro inesperado...'+#13+'Este programa será fechado...'+#13+'Contate seu
programador...','Erro...','Mb_IconError+Mb_Ok);
      Application.Terminate;
    end;
```

- 15) Vá até o **DataModule**, selecione a tabela **TiposAluguel**, vá até o evento **AfterEdit** e digite as seguintes linhas de comando:

```
If FormCadastroTiposAluguel <> Nil Then  
    FormCadastroTiposAluguel.INSERIR_ALTERAR;
```

- 16) Para finalizar o cadastro de tipos de aluguel, basta digitar as seguintes linhas no evento **OnClick** do Menu **Cadastro>Tipos de Aluguel**:

```
DM.TiposAluguel.Open;  
FormCadastroTiposAluguel:=TFormCadastroTiposAluguel.Create(Application);  
FormCadastroTiposAluguel.ShowModal;  
FormCadastroTiposAluguel.Free;  
FormCadastroTiposAluguel:=Nil;  
DM.TiposAluguel.Close;
```

→ Pronto agora vc pode rodar e testar o cadastro de Tipos de Aluguel;

Criando o Form para o Cadastro de Produtoras

Bem, como nós já usamos o repositório no cadastro anterior, isso quer dizer que tb poderemos usar novamente neste cadastro, e melhor, não precisaremos adicionar o form novamente no repositório, pq já adicionamos ele anteriormente, sendo assim ele ficará lá (no repositório) durante todo o nosso projeto mesmo que vc feche o delphi para usá-lo depois.

- 01) Se vc estiver com o Delphi aberto clique no Menu **File>Close All**;
- 02) Abra o seu projeto novamente, e logo após abri-lo clique no menu **File>New>Other...** ,
- 03) Escolha a aba **Forms** e o ítem **FormPadraoCadastro**;
- 04) Após aparecer a cópia do **FormCadastroParentesco**, altere seus componentes e deixe-o com a aparência da figura abaixo (vc precisará adicionar no form mais um DbEdit e um DbMemo):

- 05) Lembre-se que vc precisará alterar o **Caption** do Form para "**Cadastro de Tipos de Produtoras**", o **Name** do Form para "**FormCadastroProdutoras**";
- 06) Salve-o com o nome "**CadastroProdutoras.pas**";
- 07) Clique em **Project/Options...** passe o **FormCadastroProdutoras** para o Lado **Available Forms**;
- 08) Lembre-se que o item anterior serve para deixar o FormCadastroProdutoras disponível para ser criado futuramente na memória, ou seja ele só será criando na memória, quando for solicitado pelo usuário;

09) Agora selecione cada um dos componentes e altere suas propriedades **DataSource** e **DataField** para suas respectivas tabelas (DataSource) e Campos (DataField). Ex: O DBEditNome deve ter seu **DataSource** ligado à **DM.DsProdutoras** e seu **DataField** ligado à **Pro_Nome**, e assim por diante;

10) Selecione o DBNavigator e altere a propriedade **DataSource** selecionando “**DM.DsProdutoras**”;

11) Agora basta alterar a função de cada um dos botões INSERIR, GRAVAR, CANCELAR, EXCLUIR, veja a seguir:

- No evento **OnClick** do Botão **Inserir** altere para as seguintes linhas de comando:

```
Try
  INSERIR_ALTERAR;
  DM.Produtoras.Insert;
  DBEditNome.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Contate seu
programador...','Erro...','Mb_IconError+Mb_Ok);
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Gravar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Produtoras.Post;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok);
  DM.Produtoras.Cancel;
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Cancelar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Produtoras.Cancel;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok);
  DM.Produtoras.Cancel;
  BitCancelar.SetFocus;
end;
```

- No evento **OnClick** do Botão **Excluir** altere para as seguintes linhas de comando:

```
Try
  If DM.Produtoras.RecordCount > 0 Then
    Begin
      If Application.MessageBox(Pchar('Deseja realmente apagar esta
produtora?'+#13+'['+DM.ProdutorasPro_Nome.Value+']'),'Pergunta...','Mb_IconExclamation+Mb_YesNo) =
MrYes Then
        DM.Produtoras.Delete;
      end
    else
      Application.MessageBox('Não há produtora para ser excluída...' , 'Aviso...','Mb_IconInformation+Mb_Ok);
    Except
      Application.MessageBox('Erro inesperado...'+#13+'Este programa será fechado...'+#13+'Contate seu
programador...','Erro...','Mb_IconError+Mb_Ok);
      Application.Terminate;
    end;
```

- 12) Vá até o **DataModule**, selecione a tabela **Produtoras**, vá até o evento **AfterEdit** e digite as seguintes linhas de comando:

```
If FormCadastroProdutoras <> Nil Then  
    FormCadastroProdutoras.INSERIR_ALTERAR;
```

- 13) Para finalizar o cadastro de tipos de aluguel, basta digitar as seguintes linhas no evento **OnClick** do Menu **Cadastro>Produtoras**:

```
DM.Produtoras.Open;  
FormCadastroProdutoras:=TFormCadastroProdutoras.Create(Application);  
FormCadastroProdutoras.ShowModal;  
FormCadastroProdutoras.Free;  
FormCadastroProdutoras:=Nil;  
DM.Produtoras.Close;
```

→ Agora é só testar o Cadastro de Produtoras...

Criando o Form para o Cadastro de Gêneros

Usando o mesmo conceito do cadastro anterior usaremos novamente o repositório para continuar criando os nossos forms de cadastro.

- 01) Se vc estiver com o Delphi aberto clique no Menu **File>Close All**;
- 02) Abra o seu projeto novamente, e logo após abri-lo clique no menu **File>New>Other...** ,
- 03) Escolha a aba **Forms** e o item **FormPadraoCadastro**;
- 04) Após aparecer a cópia do **FormCadastroParentesco**, altere seus componentes e deixe-o com a aparência da figura abaixo (basta adicionar mais um DBEdit):

- 05) Lembre-se que vc precisará alterar o **Caption** do Form para "**Cadastro de Gêneros de Filme**", o **Name** do Form para "**FormCadastroGeneros**";
- 06) Salve-o com o nome "**CadastroGeneros.pas**";
- 07) Clique em **Project/Options...** passe o **FormCadastroGeneros** para o Lado **Available Forms**;
- 08) Lembre-se que o item anterior serve para deixar o FormCadastroGeneros disponível para ser criado futuramente na memória, ou seja ele só será criando na memória, quando for solicitado pelo usuário;

09) Agora selecione cada um dos componentes e altere suas propriedades **DataSource** e **DataField** para suas respectivas tabelas (DataSource) e Campos (DataField). Ex: O DBEditNome deve ter seu **DataSource** ligado à **DM.DsGeneros** e seu **DataField** ligado à **Gen_Nome**, e assim por diante;

10) Selecione o DBNavigator e altere a propriedade **DataSource** selecionando “**DM.DsGeneros**”;

11) Agora basta alterar a função de cada um dos botões INSERIR, GRAVAR, CANCELAR, EXCLUIR, veja a seguir:

- No evento **OnClick** do Botão **Inserir** altere para as seguintes linhas de comando:

```
Try
  INSERIR_ALTERAR;
  DM.Generos.Insert;
  DM.GenerosGen_UltimoCodigo.Value:='00000';
  DBEditNome.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Contate seu
programador...','Erro...',Mb_IconError+Mb_Ok);
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Gravar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Generos.Post;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...',Mb_IconError+Mb_Ok);
  DM.Generos.Cancel;
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Cancelar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Generos.Cancel;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...',Mb_IconError+Mb_Ok);
  DM.Generos.Cancel;
  BitCancelar.SetFocus;
end;
```

- No evento **OnClick** do Botão **Excluir** altere para as seguintes linhas de comando:

```
Try
  If DM.Generos.RecordCount > 0 Then
    Begin
      If Application.MessageBox(Pchar('Deseja realmente apagar este gênero de
fita?'+#13+'['+DM.GenerosGen_Nome.Value+']'),'Pergunta...',Mb_IconExclamation+Mb_YesNo) = MrYes
      Then
        DM.Generos.Delete;
      end
    else
      Application.MessageBox('Não há gênero de fita para ser excluído...'
,'Aviso...',Mb_IconInformation+Mb_Ok);
    Except
      Application.MessageBox('Erro inesperado...'+#13+'Este programa será fechado...'+#13+'Contate seu
programador...','Erro...',Mb_IconError+Mb_Ok);
      Application.Terminate;
    end;
```

- 12) Vá até o **DataModule**, selecione a tabela **Generos**, vá até o evento **AfterEdit** e digite as seguintes linhas de comando:

```
If FormCadastroGeneros <> Nil Then  
    FormCadastroGeneros.INSERIR_ALTERAR;
```

- 13) Para finalizar o cadastro de tipos de aluguel, basta digitar as seguintes linhas no evento **OnClick** do Menu **Cadastro>Gêneros**:

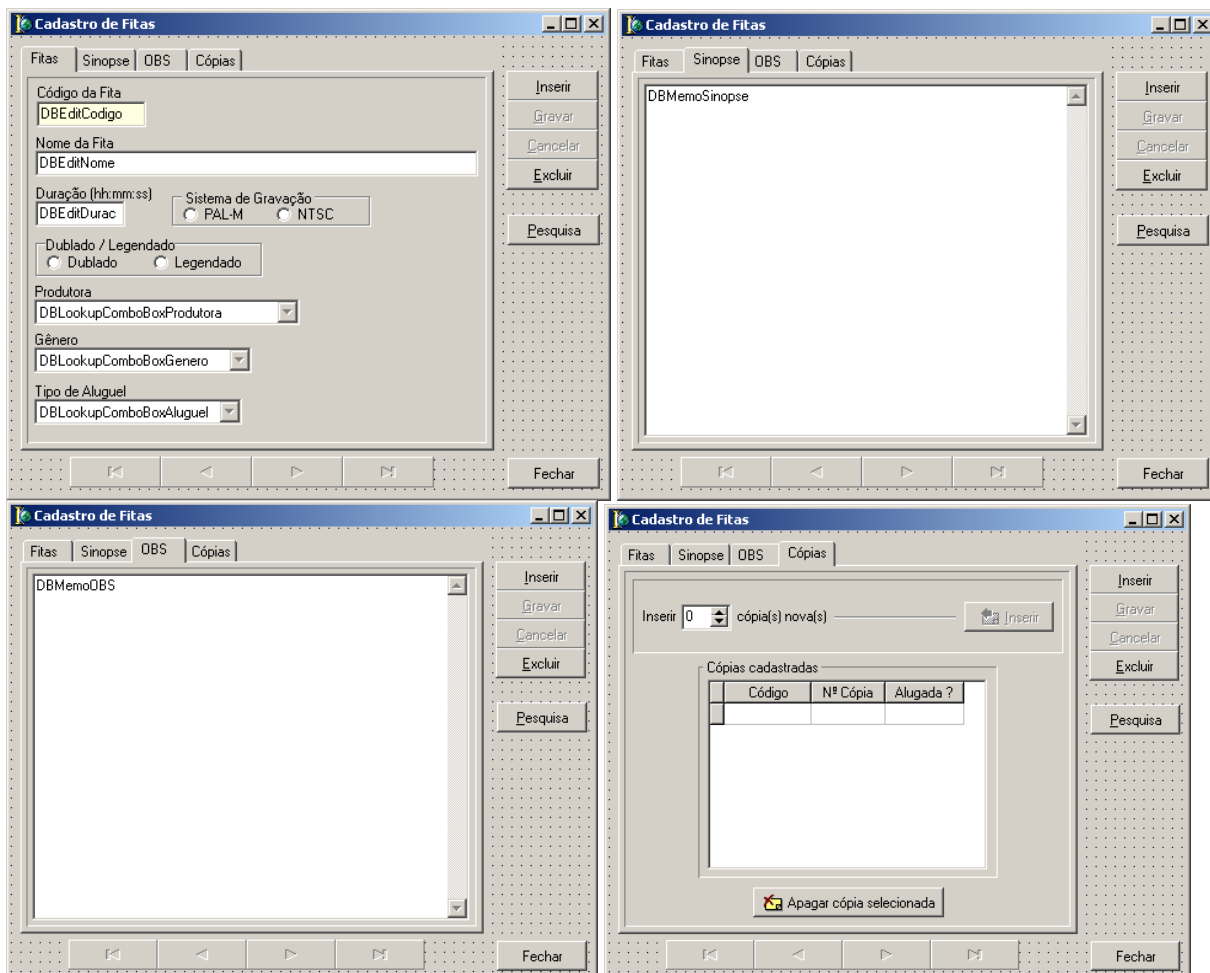
```
DM.Generos.Open;  
FormCadastroGeneros:=TFormCadastroGeneros.Create(Application);  
FormCadastroGeneros.ShowModal;  
FormCadastroGeneros.Free;  
FormCadastroGeneros:=Nil;  
DM.Generos.Close;
```

→ Pronto, agora é só rodar e testar o cadastro de Gêneros de Filme;

Criando o Form para o Cadastro de Fitas

Usando o mesmo conceito do cadastro anterior usaremos novamente o repositório para continuar criando os nossos forms de cadastro.

- 01) Se vc estiver com o Delphi aberto clique no Menu **File>Close All**;
- 02) Abra o seu projeto novamente, e logo após abri-lo clique no menu **File>New>Other...** ,
- 03) Escolha a aba **Forms** e o ítem **FormPadraoCadastro**;
- 04) Após aparecer a cópia do **FormCadastroParentesco**, altere seus componentes e deixe-o com a aparência da figura abaixo:



- 05) Lembre-se que vc precisará alterar o **Caption** do Form para “**Cadastro de Fitas**”, o **Name** do Form para “**FormCadastroFitas**”;
 - 06) Salve-o com o nome “**CadastroFitas.pas**”;
 - 07) Clique em **Project/Options...** passe o **FormCadastroFitas** para o Lado **Available Forms**;
 - 08) Altere a propriedade **DataSource** do **DbGrid** que está na aba Cópias para **DsFitas**;
 - 09) Agora selecione cada um dos componentes e altere suas propriedades **DataSource** e **DataField** para suas respectivas tabelas (DataSource) e Campos (DataField). Ex: O DBEditNome deve ter seu **DataSource** ligado à **DM.DsFitas** e seu **DataField** ligado à **Fit_Nome**, e assim por diante;
 - 10) Selecione o DBNavigator e altere a propriedade **DataSource** selecionando “**DM.DsFitas**”;
 - 11) Agora basta alterar a função de cada um dos botões INSERIR, GRAVAR, CANCELAR, EXCLUIR, veja a seguir:
- No evento **OnClick** do Botão **Inserir** altere para as seguintes linhas de comando:

```
Try
  INSERIR_ALTERAR;
  DM.Fitas.Insert;
  DM.FitasFit_Duracao.Value:=StrToTime('00:00:00');
  DM.FitasFit_SistGravacao.Value:='P';
  DM.FitasFit_DubLeg.Value:='D';
  PageControl1.ActivePageIndex:=0;
  DBEditNome.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Contate seu
programador...','Erro...','Mb_IconError+Mb_Ok);
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Gravar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Fitas.Post;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok);
  DM.Fitas.Cancel;
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Cancelar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Fitas.Cancel;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...','Mb_IconError+Mb_Ok);
  DM.Fitas.Cancel;
  BitCancelar.SetFocus;
end;
```

- No evento **OnClick** do Botão **Excluir** altere para as seguintes linhas de comando:

```
Try
  If DM.Fitas.RecordCount > 0 Then
    Begin
      If Application.MessageBox(Pchar('Deseja realmente apagar esta
Fita?'+#13+'[Fita] = '+DM.FitasFit_Nome.Value+']),'Pergunta...',Mb_IconExclamation+Mb_YesNo) = MrYes Then
        DM.Fitas.Delete;
      end
    else
      Application.MessageBox('Não há fita para ser excluída...' , 'Aviso...',Mb_IconInformation+Mb_Ok);
    Except
      Application.MessageBox('Erro inesperado...'+#13'Este programa será fechado...'+#13+'Contate seu
programador...', 'Erro...',Mb_IconError+Mb_Ok);
      Application.Terminate;
    end;
  end;
```

- 12) Vá até o **DataModule**, selecione a tabela **Fitas**, vá até o evento **AfterEdit** e digite as seguintes linhas de comando:

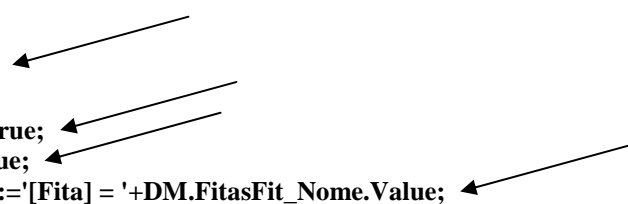
```
If FormCadastroFitas <> Nil Then
  FormCadastroFitas.INSERIR_ALTERAR;
```

- 13) Ainda com a tabela **Fitas** selecionada, vá até o evento **AfterScroll** e digite as seguintes linhas de comando:

```
FormCadastroFitas.Caption:=[Fita] = '+DM.FitasFit_Nome.Value;
```

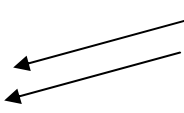
- 14) Volte ao FormCadastroFitas e na procedure **GRAVAR_CANCELAR** acrescente as seguinte linhas de comando:

```
BitInserir.Enabled:=True;
BitGravar.Enabled:=False;
BitCancelar.Enabled:=False;
BitExcluir.Enabled:=True;
BitFechar.Enabled:=True;
BitPesquisa.Enabled:=True;
DBNavigator.Enabled:=True;
BitInserirCopia.Enabled:=True;
BitApagarFita.Enabled:=True;
FormCadastroFitas.Caption:=[Fita] = '+DM.FitasFit_Nome.Value;
```



- 15) Ainda no FormCadastroFitas vá até a procedure **INSERIR_ALTERAR** e acrescente as seguinte linhas de comando:

```
BitInserir.Enabled:=False;
BitGravar.Enabled:=True;
BitCancelar.Enabled:=True;
BitExcluir.Enabled:=False;
BitFechar.Enabled:=False;
BitPesquisa.Enabled:=False;
DBNavigator.Enabled:=False;
BitInserirCopia.Enabled:=False;
BitApagarFita.Enabled:=False;
```



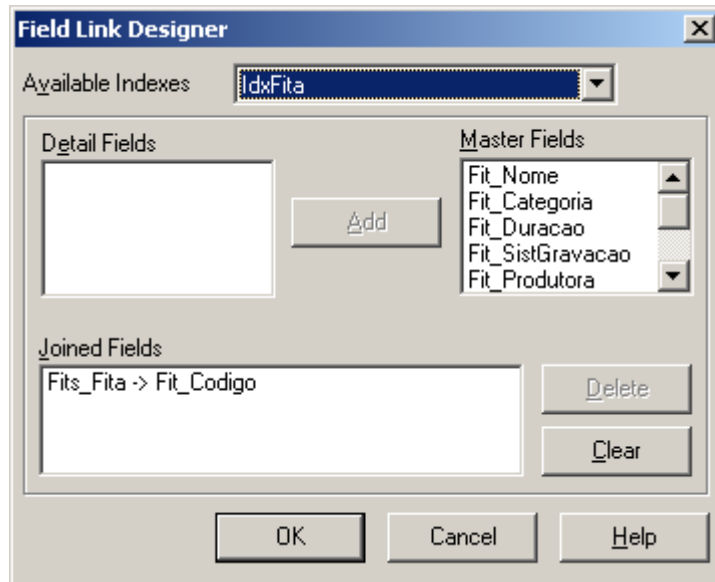
- 16) Para finalizar o cadastro de Fitas, basta digitar as seguintes linhas no evento **OnClick** do Menu **Cadastro>Fitas**:

```
DM.Generos.Open;
DM.Produoras.Open;
DM.TiposAluguel.Open;
DM.Fitass.Open;
DM.Fitas.Open;
FormCadastroFitas:=TFormCadastroFitas.Create(Application);
PageControl1.ActivePageIndex:=0;
FormCadastroFitas.Caption:=[Fita] = '+DM.FitasFit_Nome.Value;
FormCadastroFitas.ShowModal;
FormCadastroFitas.Free;
FormCadastroFitas:=Nil;
DM.Generos.Close;
```

```
DM.Produutoras.Close;  
DM.TiposAluguel.Close;  
DM.Fitass.Close;  
DM.Fitas.Close;
```

- Criando um MasterSource (Relação Fitas/Fitass)

- 17) Vá até o **DataModule** e Selecione a tabela **Fitass**, Altere a opção **MasterSource** para **DsFitas**; Selecione a propriedade **MasterFields** e clique nos 03 pontinhos; Veja afigura abaixo:



- 18) Para deixar a janela do mesmo jeito da apresentada acima, clique no combobox **Available Indexes** e Selecione **IdxFita**, logo após selecione **Fits_Fita** na caixa **Detail Fields**, e depois selecione **Fit_Codigo** na caixa **Master Fields**, pronto, agora é só clicar no botão **Add** e em seguida, no botão **OK**;

-- Fazendo o Cadastro de Cópias

- 19) Selecione o **FormCadastroFitas**, selecione a aba **Cópias**, vá até o evento **onClick** do Botão Inserir (da aba Cópias) e digite as seguintes linhas de comando que estão entre chaves:

```
procedure TFormCadastroFitas.BitInserirCopiaClick(Sender: TObject);  
{  
  Var  
  I:Integer;  
  begin  
    For I:=1 to SpinEdit1.Value do  
      Begin  
        DM.Fitass.Insert;  
  
        DM.FitassFits_Codigo.Value:=DM.GenerosGen_Abreviacao.Value+FormatFloat('00000',StrToInt(DM.GenerosGen_UltimoCodigo.Value)+1);  
  
        DM.FitassFits_NumCopia.Value:=FormatFloat('00000',StrToInt(DM.GenerosGen_UltimoCodigo.Value)+1);  
        DM.FitassFits_Alugada.Value:='N';  
        DM.Fitass.Post;  
  
        DM.Generos.Edit;  
  
        DM.GenerosGen_UltimoCodigo.Value:=IntToStr(StrToInt(DM.GenerosGen_UltimoCodigo.Value)+1);  
        DM.Generos.Post;  
      end;  
    end;  
  end;
```

- 20) Agora na mesma aba vá até o evento **OnClick** do Botão “**Apagar Cópia Selecionada**” e digite as seguintes linhas:

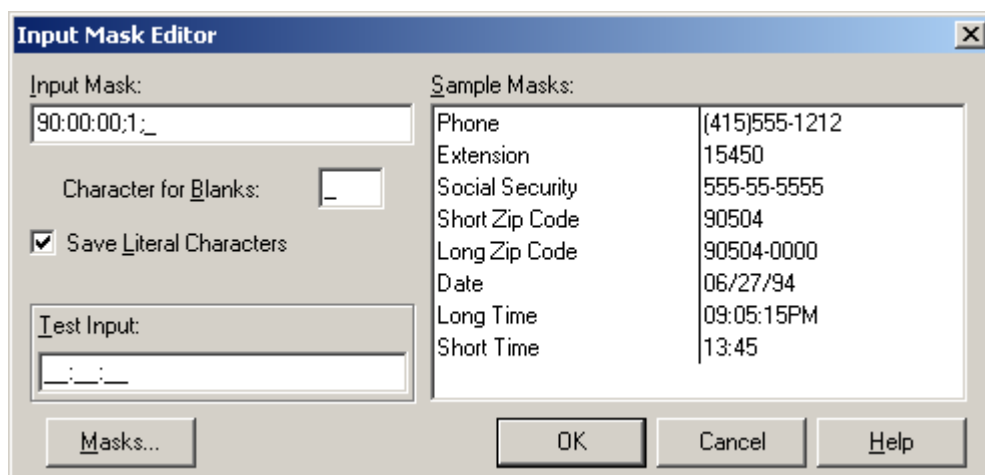
```
Try
  If DM.Fitass.RecordCount > 0 Then
    Begin
      If Application.MessageBox(Pchar('Deseja realmente apagar esta Cópia de Fita?'+#13+'[ N° - '+DM.Fitass.Fits_Codigo.Value+' ]'),'Pergunta...','Mb_IconExclamation+Mb_YesNo) = MrYes Then
        DM.Fitass.Delete;
      end
    else
      Application.MessageBox('Não há cópia de fita para ser excluída...' , 'Aviso...', Mb_IconInformation+Mb_Ok);
    Except
      Application.MessageBox('Erro inesperado...'+#13+'Este programa será fechado...'+#13+'Contate seu programador...' , 'Erro...', Mb_IconError+Mb_Ok);
    Application.Terminate;
  end;
```

- 21) Para finalizar vá até o **SpinEdit** da mesma aba e digite no Evento **OnChange** a seguinte linha de Comando:

```
BitInserirCopia.Enabled:=SpinEdit1.Value > 0;
```

-- Ajustando a Máscara da Duração

- 22) Vá até o **DataModule** e Selecione a tabela **Fitas**, faça um duplo clique na mesma e selecione o campo **Fit_Duracao**, depois configure a propriedade **EditMask** como na figura a seguir:



- 23) Clique em Ok e pronto, está configurado a duração da Tabela Fitas.

Criando o Form para o Cadastro de Usuários

Usando o mesmo conceito do cadastro anterior usaremos novamente o repositório para continuar criando os nossos forms de cadastro.

- 01) Se vc estiver com o Delphi aberto clique no Menu **File>Close All**;
- 02) Abra o seu projeto novamente, e logo após abri-lo clique no menu **File>New>Other...** ,
- 03) Escolha a aba **Forms** e o item **FormPadraoCadastro**;
- 04) Após aparecer a cópia do **FormCadastroParentesco**, altere seus componentes e deixe-o com a aparência da figura abaixo:

- 05) Lembre-se que vc precisará alterar o **Caption** do Form para “**Cadastro de Usuários de Sistema**”, o **Name** do Form para “**FormCadastroUsuarios**”;
- 06) Salve-o com o nome “**CadastroUsuariosSistema.pas**”;
- 07) Clique em **Project/Options...** passe o **FormCadastroUsuarios** para o Lado **Available Forms**;
- 08) Agora selecione cada um dos componentes e altere suas propriedades **DataSource** e **DataField** para suas respectivas tabelas (DataSource) e Campos (DataField). Ex: O DBEditNome deve ter seu **DataSource** ligado à **DM.DsUsuarios** e seu **DataField** ligado à **Usu_Nome**, e assim por diante;
- 09) Selecione o DBNavigator e altere a propriedade **DataSource** selecionando “**DM.DsUsuarios**”;
- 10) Agora basta alterar a função de cada um dos botões INSERIR, GRAVAR, CANCELAR, EXCLUIR, veja a seguir:
 - No evento **OnClick** do Botão **Inserir** altere para as seguintes linhas de comando:

```
Try
  INSERIR_ALTERAR;
  DM.Usuarios.Insert;
  DBEditNome.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Contate seu
programador...','Erro...',Mb_IconError+Mb_Ok);
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Gravar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Usuarios.Post;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13+'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...',Mb_IconError+Mb_Ok);
  DM.Usuarios.Cancel;
  GRAVAR_CANCELAR;
  BitInserir.SetFocus;
end;
```

- No evento **OnClick** do Botão **Cancelar** altere para as seguintes linhas de comando:

```
Try
  GRAVAR_CANCELAR;
  DM.Usuarios.Cancel;
  BitInserir.SetFocus;
Except
  Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13'Este processo será
cancelado...'+#13+'Contate seu programador...','Erro...',Mb_IconError+Mb_Ok);
  DM.Usuarios.Cancel;
  BitCancelar.SetFocus;
end;
```

- No evento **OnClick** do Botão **Excluir** altere para as seguintes linhas de comando:

```
Try
  If DM.Usuarios.RecordCount > 0 Then
    Begin
      If Application.MessageBox(Pchar('Deseja realmente apagar esse
usuário?'+#13+'[ '+DM.UsuariosUsu_Nome.Value+' ]'),'Pergunta...',Mb_IconExclamation+Mb_YesNo) = MrYes
    Then
      DM.Usuarios.Delete;
    end
    else
      Application.MessageBox('Não há usuário para ser excluído...','Aviso...',Mb_IconInformation+Mb_Ok);
    Except
      Application.MessageBox('Erro inesperado...'+#13'Este programa será fechado...'+#13+'Contate seu
programador...','Erro...',Mb_IconError+Mb_Ok);
      Application.Terminate;
    end;
```

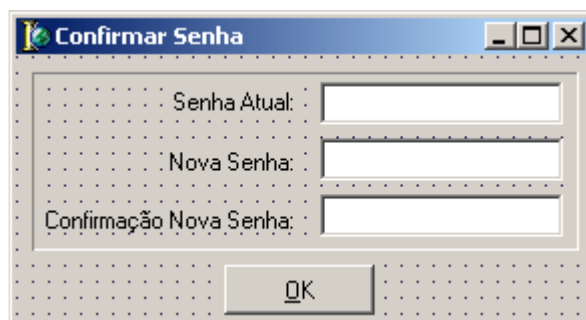
11) Note que ainda não fizemos nada para que fosse armazenada a senha do usuário no banco de dados; Agora criaremos tudo necessário para armazenar e alterar a senha dos usuários;

12) Crie um novo form clicando em **File>New>Form**;

13) Salve o novo Form com o Nome de “**ConfirmarSenha.pas**”;

14) Altere a propriedade **Name** do Form para **FormConfirmarSenha**;

15) Deixe o form com a aparência da figura abaixo:



16) No form mostrado acima altere as seguintes propriedades:

- Selecione o FormConfirmarSenha e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;
 - **ActiveControl** para **BitInserir**;

17) Selecione os três (03) edits e altere a propriedade **PasswordChar** para * ;

18) Clique em **Project/Options...** passe o **FormConfirmarSenha** para o Lado **Available Forms**;

19) Selecione o Botão OK e no evento OnClick do mesmo digite o seguinte comando:

```
Close;
```

20) Vá até a unit do FormConfirmarSenha e depois da cláusula **Public** declare a variável como mostrado abaixo:

```
private
{ Private declarations }
public
vTIPO:Integer;
{ Public declarations }
end;
```

21) Selecione o **FormConfirmarSenha** e no evento **OnClose** digite as seguintes linhas:

```
If vTipo = 1 Then
Begin
  If (Trim(EditNovaSenha.Text) = Trim(EditConfirmaSenha.Text)) and
  (Length(Trim(EditConfirmaSenha.Text)) > 0) Then
    DM.UsuariosUsu_Senha.Value:=EditNovaSenha.Text
  else
  Begin
    ShowMessage('Senha não confere, ou senha inválida !!!'+#13+'Tente novamente...');
    EditNovaSenha.Clear;
    EditConfirmaSenha.Clear;
    EditNovasenha.SetFocus;
    ABORT;
  end;
end;

//=====

If vTipo = 2 Then
Begin
  If (Trim(EditNovaSenha.Text) = Trim(EditConfirmaSenha.Text)) and
  (Length(Trim(EditConfirmaSenha.Text)) > 0) and (DM.UsuariosUsu_Senha.Value = EditSenhaAtual.Text)
  Then
  Begin
    DM.Usuarios.Edit;
    DM.UsuariosUsu_Senha.Value:=EditNovaSenha.Text;
    DM.Usuarios.Post;
    FormCadastroUsuarios.GRAVAR_CANCELAR;
    FormCadastroUsuarios.BitInserir.SetFocus;
  end
  else
  Begin
    If Application.MessageBox('Senha não confere, ou senha inválida !!!'+#13+'Deseja tentar novamente...', 'Pergunta...', Mb_IconExclamation+Mb_YesNo) = MrYes Then
      Begin
        EditSenhaAtual.Clear;
        EditNovaSenha.Clear;
        EditConfirmaSenha.Clear;
        EditSenhaAtual.SetFocus;
        ABORT;
      end;
    end;
  end;
end;
```

22) Volte ao FormCadastroUsuarios e declare na sessão **Uses** (logo no início) da unit a biblioteca **DB**;

23) Selecione o Botão “**Alterar Senha**” e digite as seguintes linhas de comando:

```
If Application.MessageBox(Pchar('Deseja realmente alterar a senha desse
usuário?'+#13+'['+DM.UsuariosUsu_Nome.Value+']'), 'Pergunta...', Mb_IconExclamation+Mb_YesNo) = MrYes
Then
Begin
```

```
FormConfirmaSenha:=TFormConfirmaSenha.Create(Application);
FormConfirmaSenha.vTIPO:=2;
FormConfirmaSenha.Caption:='Digite a sua nova senha... (Alteração)';
FormConfirmaSenha.ShowModal;
FormConfirmaSenha.Free;
FormConfirmaSenha:=Nil;
end;
```

24) Para finalizar, selecione o botão “**Gravar**” e no evento **OnClick** adicione as seguintes linhas:

```
Try
{
If DM.Usuarios.State = DsInsert Then
Begin
    FormConfirmaSenha:=TFormConfirmaSenha.Create(Application);
    FormConfirmaSenha.vTIPO:=1;
    FormConfirmaSenha.Caption:='Digite a sua senha... (Nova Senha)';
    FormConfirmaSenha.EditSenhaAtual.Enabled:=False;
    FormConfirmaSenha.EditSenhaAtual.Color:=clBtnFace;
    FormConfirmaSenha.ShowModal;
    FormConfirmaSenha.Free;
    FormConfirmaSenha:=Nil;
end;
    GRAVAR_CANCELAR;
    DM.Usuarios.Post;
    BitInserir.SetFocus;
Except
    Application.MessageBox('Não foi possível concluir a operação solicitada...'+#13'Este processo será cancelado...'+#13'Contate seu programador...','Erro...',Mb_IconError+Mb_Ok);
    DM.Usuarios.Cancel;
    GRAVAR_CANCELAR;
    BitInserir.SetFocus;
end;
```

- 4º Passo (Configurando um Data Module para SQL) -

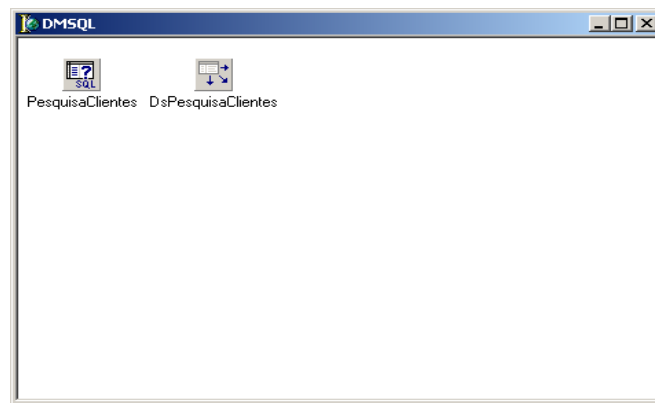
- 1) Com o seu sistema aberto no Delphi, clique no menu **File/New/Data Module** do Delphi;
- 2) Salve o DataModule com o nome “**TabelasSQL.pas**” na pasta **SISLFV/Units**;
- 3) Altere a propriedade **Name** do DataModule1 para “**DMSQL**”;
- 4) Adicione um(01) componente não visuais **Query** que está localizado na palheta **BDE**;
- 5) Selecione a **Query** e altere a propriedade **DataBaseName** para “**SISLFV**”;
- 6) Agora selecione a Query e altere a propriedade **Name** para **PesquisaClientes**
- 7) Vá até a propriedade SQL da **Query** e clique nos três pontinhos; Aparecerá um editor, então digite a seguinte linha de comando:

```
Select Cli_Codigo, Cli_Nome, Cli_Apelido from Clientes
```

Clique em OK;

- 8) Faça um clique duplo em cima da Query **PesquisaClientes** (aparecerá o Fields Editor), clique com o botão direito do mouse e selecione a opção **Add All Fields**.
- 10) Adicione um(01) componente não visual **DataSource** que está localizado na palheta **Data Access**;
- 11) Selecione o **DataSource1** e altere a propriedade **DataSet** selecionando a tabela **PesquisaClientes**;
- 12) Altere a Propriedade **Name** do **DataSource1** para **DsPesquisaClientes**;

Após terminar todo este processo o seu DM deverá ficar com a aparência da figura abaixo



- 5º Passo (Criando e Configurando Forms para Pesquisa) -

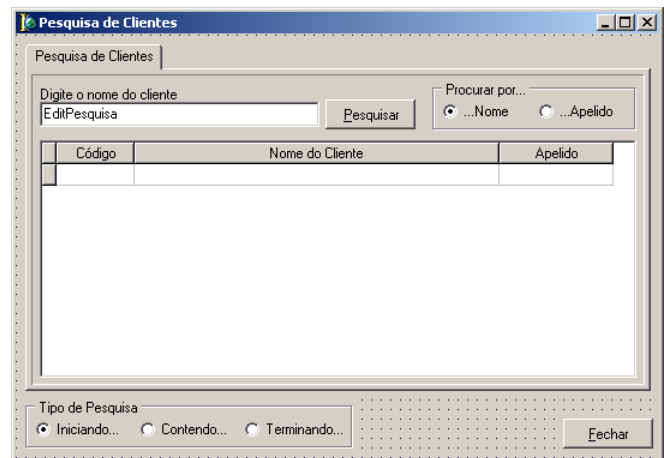
Criando o Form para o Pesquisa de Clientes

1) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;

2) Salve o novo Form com o Nome de **“PesquisaClientes.pas”**;

3) Altere a propriedade **Name** do Form para **FormPesquisaClientes**;

4) Usando os componentes da palheta **DataControls** crie seu form de Pesquisa de clientes, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado na figura abaixo):



5) No exemplo acima altere as seguintes propriedades:

- Selecione o FormPesquisaClientes e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;
 - **ActiveControl** para **EditPesquisa**;

6) Clique em **Project/Options...** passe o **FormPesquisaClientes** para o Lado **Available Forms**;

7) Selecione o **DBGrid** e altere a sua propriedade **DataSource** para **DMSQL.DsPesquisaClientes**;

8) Selecione o botão **Pesquisar** e no evento **OnClick** do mesmo digite as seguintes linhas de comando:

```
DMSQL.PesquisaClientes.Close;
DMSQL.PesquisaClientes.SQL.Clear;
DMSQL.PesquisaClientes.SQL.Add('Select Cli_Codigo, Cli_Nome, Cli_Apelido from Clientes Where');
```

```
Case RadioProcurar.ItemIndex of
0: DMSQL.PesquisaClientes.SQL.Add('Cli_Nome Like :Nome ');
1: DMSQL.PesquisaClientes.SQL.Add('Cli_Apelido Like :Apelido ');
end;
```

```
Case RadioTipoPesquisa.ItemIndex of
0: DMSQL.PesquisaClientes.Params[0].AsString:= EditPesquisa.Text+'%';
1: DMSQL.PesquisaClientes.Params[0].AsString:='%'+ EditPesquisa.Text+'%';
2: DMSQL.PesquisaClientes.Params[0].AsString:='%'+ EditPesquisa.Text;
end;
```

```
DMSQL.PesquisaClientes.Open;
```

9) Selecione o **EditPesquisa** e no evento **OnKeyDown** digite as seguintes linhas:

```
If Key = VK_Return Then
    BitPesquisar.Click;
```

10) Vá até o **FormCadastroClientes** e no evento **OnClick** do botão **Pesquisar** digite as seguintes linhas:

```
FormPesquisaClientes:=TFormPesquisaClientes.Create(Application);  
  
DMSQL.PesquisaClientes.Open;  
  
FormPesquisaClientes.ShowModal;  
FormPesquisaClientes.Free;  
FormPesquisaClientes:=Nil;  
  
DM.Clientes.FindKey([DMSQL.PesquisaClientesCli_Codigo.Value]);  
DMSQL.PesquisaClientes.Close;
```

11) Para finalizar esta parte, vá até o evento **OnClick** do Menu **Pesquisa>Clientes** do seu sistema e digite as seguintes linhas de comando:

```
DMSQL.PesquisaClientes.Close;  
DMSQL.PesquisaClientes.SQL.Clear;  
DMSQL.PesquisaClientes.SQL.Add('Select Cli_Codigo, Cli_Nome, Cli_Apelido from Clientes');  
DMSQL.PesquisaClientes.Open;  
  
FormPesquisaClientes:=TFormPesquisaClientes.Create(Application);  
FormPesquisaClientes.ShowModal;  
  
DMSQL.PesquisaClientes.Close;  
  
FormPesquisaClientes.Free;  
FormPesquisaClientes:=Nil;
```

Criando o Form para o Pesquisa de Fitas

1) Inicialmente vá até o DMSQL e adicione mais um **Query** e um **DataSource** localizados na palheta **BDE** e **DataAccess** respectivamente;

2) Selecione a **Query** e altere as seguintes propriedades:

- **DatabaseName** = **SISLFV**
- **Name** = **PesquisaFitas**
- Selecione a propriedade SQL e digite a seguinte linha de comando:

Select Fit_Codigo, Fits_Codigo, Fit_Nome from Fitas, Fitass Where Fit_Codigo = Fits_Fita

3) Faça um clique duplo em cima da Query **PesquisaFitas** (aparecerá o Fields Editor), clique com o botão direito do mouse e selecione a opção **Add All Fields**.

4) Selecione o **DataSource** e altere as seguintes propriedades:

- **DataSet** = **PesquisaFitas**
- **Name** = **DsPesquisaFitas**

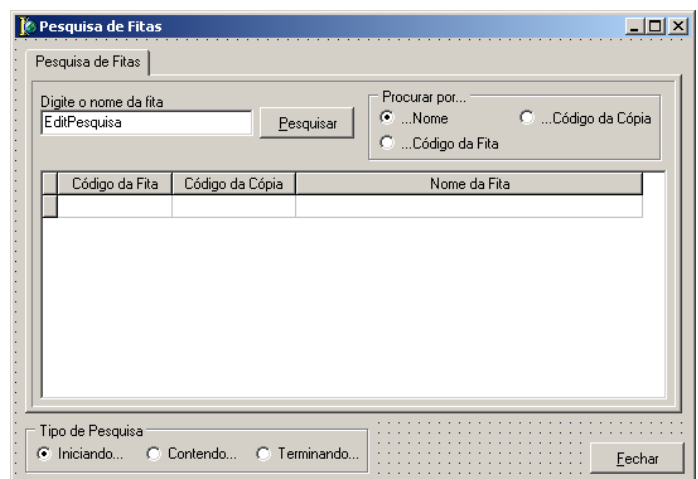
5) Use o **Repositório (como aprendido na página 23)** para criar uma cópia do **FormPesquisaClientes**; após criar a cópia deixe-o com a aparência abaixo:

06) Lembre-se que vc precisará alterar o **Caption** do Form para “**Pesquisa de Fitas**”, o **Name** do Form para “**FormPesquisaFitas**”;

07) Salve-o com o nome “**PesquisaFitas.pas**”;

08) Clique em **Project/Options...** passe o **FormPesquisaFitas** para o Lado **Available Forms**;

09) Agora selecione o DBGrid e altere a propriedade **DataSource** para **DMSQL.DsPesquisaFitas**;



10) Selecione o botão **Pesquisar** e no evento **OnClick** digite as seguintes linhas de comando:

```
procedure TFormPesquisaFitas.BitPesquisarClick(Sender: TObject);  
  Var  
  {  
  vPesquisa:String;  
  begin  
    Case RadioTipoPesquisa.ItemIndex of  
      0: vPesquisa:=EditPesquisa.Text+'%';  
      1: vPesquisa:='%'+EditPesquisa.Text+'%';  
      2: vPesquisa:='%'+EditPesquisa.Text;  
    end;  
  
    DMSQL.PesquisaFitas.Close;  
    DMSQL.PesquisaFitas.SQL.Clear;  
    DMSQL.PesquisaFitas.SQL.Add('Select Fit_Codigo, Fits_Codigo,Fit_Nome from Fitas, Fitass Where  
    Fit_Codigo = Fits_Fita');  
  
    Case RadioProcurar.ItemIndex of  
      0: Begin  
        DMSQL.PesquisaFitas.SQL.Add('and Fit_Nome Like :Nome ');  
        DMSQL.PesquisaFitas.Params[0].AsString:=vPesquisa;  
      end;  
  
      1: Begin  
        DMSQL.PesquisaFitas.SQL.Add('and Fit_Codigo =:Codigo ');  
        DMSQL.PesquisaFitas.Params[0].AsInteger:=StrToInt(EditPesquisa.Text);  
      end;  
  
      2: Begin  
        DMSQL.PesquisaFitas.SQL.Add('and Fits_Codigo Like :Codigo ');  
        DMSQL.PesquisaFitas.Params[0].AsString:=vPesquisa;  
      end;  
    end;  
  
    DMSQL.PesquisaFitas.Open;  
  end;
```

11) Vá até o **FormCadatroFitas** e no evento **OnClick** do botão Pesquisar digite as seguintes linhas de comando:

```
FormPesquisaFitas:=TFormPesquisaFitas.Create(Application);  
  
DMSQL.PesquisaFitas.Open;  
  
FormPesquisaFitas.ShowModal;  
FormPesquisaFitas.Free;  
FormPesquisaFitas:=Nil;  
  
DM.Fitas.FindKey([DMSQL.PesquisaFitasFit_Codigo.Value]);  
DMSQL.PesquisaFitas.Close;
```

12) Para finalizar esta parte, vá até o evento **OnClick** do Menu **Pesquisa>Fitas** do seu sistema e digite as seguintes linhas de comando:

```
DMSQL.PesquisaFitas.Close;  
DMSQL.PesquisaFitas.SQL.Clear;  
DMSQL.PesquisaFitas.SQL.Add('Select Fit_Codigo, Fits_Codigo,Fit_Nome from Fitas, Fitass Where  
Fit_Codigo = Fits_Fita');  
DMSQL.PesquisaFitas.Open;  
  
FormPesquisaFitas:=TFormPesquisaFitas.Create(Application);  
FormPesquisaFitas.ShowModal;  
  
DMSQL.PesquisaFitas.Close;  
  
FormPesquisaFitas.Free;  
FormPesquisaFitas:=Nil;
```

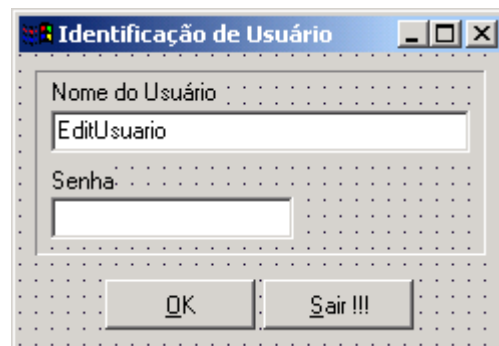
- 6º Passo (Criando e Configurando o Form para a Identificação do Usuário no Sistema) -

Criando o Form para a Identificação do Usuário no Sistema

OBS: Antes de iniciar a fazer esta parte do exercício, crie um usuário MESTRE, com senha 123 no seu programa !!!

- 1) Inicialmente vá até o **DMSQL** e adicione mais um **Query** e um **DataSource** localizados na palheta **BDE** e **DataAccess** respectivamente;
- 2) Selecione a **Query** e altere as seguintes propriedades:
 - **DatabaseName** = **SISLFV**
 - **Name** = **PesquisaUsuarios**
 - Selecione a propriedade **SQL** e digite a seguinte linha de comando:

Select * from Usuarios
- 3) Faça um clique duplo em cima da Query **PesquisaUsuarios** (aparecerá o Fields Editor), clique com o botão direito do mouse e selecione a opção **Add All Fields**.
- 4) Selecione o **DataSource** e altere as seguintes propriedades:
 - **DataSet** = **PesquisaFitas**
 - **Name** = **DsPesquisaFitas**
- 5) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;
- 6) Salve o novo Form com o Nome de **“IdentificacaoUsuario.pas”**;
- 7) Altere a propriedade **Name** do Form para **FormIdentificacaoUsuario**;
- 8) Usando os componentes crie o seu form de identificação de usuário, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado na figura abaixo):



- 9) No exemplo acima altere as seguintes propriedades:
 - Selecione o **FormIdentificacaoUsuarios** e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsDialog**;
 - **ActiveControl** para **EditUsuario**;
- 10) Clique em **Project/Options...** passe o **FormIdentificacaoUsuario** e o **DMSQL** para o Lado **Available Forms**;
- 11) Selecione o botão **“Sair !!!”** e no evento **OnClick** do mesmo digite as seguintes linhas de comando:


```
If Application.MessageBox('Tem certeza que deseja sair do sistema ?', 'Pergunta...', Mb_IconExclamation+Mb_YesNo) = MrYes Then  
    Application.Terminate  
else  
    Abort;
```
- 12) Selecione o botão **“OK”** e no evento **OnClick** do mesmo digite a seguinte linha de comando:


```
Close;
```

- 13) Selecione o botão **FormIdentificacaoUsuario** e no evento **OnClose** do mesmo digite as seguintes linhas de comando:

```
DMSQL.PesquisaUsuarios.Close;
DMSQL.PesquisaUsuarios.SQL.Clear;
DMSQL.PesquisaUsuarios.SQL.Add('Select * from Usuarios Where Usu_NomeUsuario=:Usuario and
Usu_Senha=:Senha');
DMSQL.PesquisaUsuarios.Params[0].AsString:=EditUsuario.Text;
DMSQL.PesquisaUsuarios.Params[1].AsString:=EditSenha.Text;
DMSQL.PesquisaUsuarios.Open;

If DMSQL.PesquisaUsuarios.RecordCount = 1 Then
  Close
else
begin
  Application.MessageBox('O usuário ou senha estão incorretos...'+#13+'Verifique o CAPSLOCK e tente
novamente...','Erro...','Mb_IconError+Mb_OK');
  EditSenha.Clear;
  EditUsuario.SetFocus;
  ABORT;
end;
```

- 15) Agora vá até o evento **OnCreate** do **FormPrincipal** e digite as seguintes linhas de comando:

```
FormIdentificacaoUsuario:=TFormIdentificacaoUsuario.Create(Application);
DMSQL:=TDMSQL.Create(Application);
FormIdentificacaoUsuario.ShowModal;
FormIdentificacaoUsuario.Free;
StatusBar1.Panels[1].Text:='Usuário: ' + IntToStr(DMSQL.PesquisaUsuariosUsu_Codigo.Value) + ' - ' +
DMSQL.PesquisaUsuariosUsu_NomeUsuario.Value + ' - ' + DMSQL.PesquisaUsuariosUsu_Nome.Value;
```

- 7º Passo (Criando e Configurando o Form para a Pesquisa de Fitas Alugadas) -

Criando o Form para a Pesquisa de Fitas Alugadas

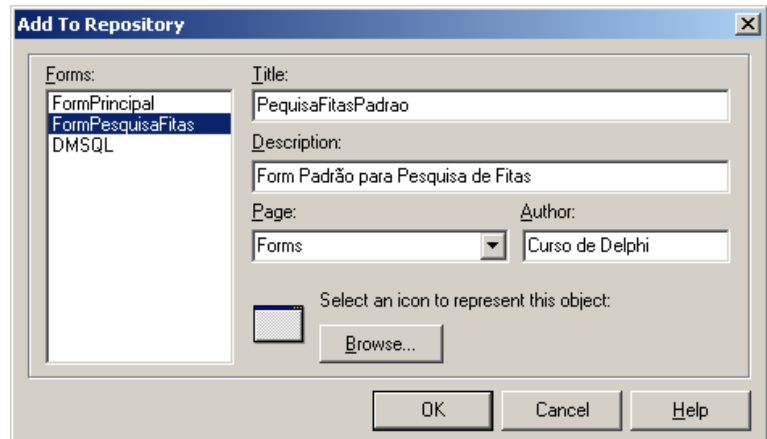
- 1) Inicialmente vá até o **DMSQL** e adicione mais um **Query** e um **DataSource** localizados na palheta **BDE** e **DataAccess** respectivamente;
- 2) Selecione a **Query** e altere as seguintes propriedades:
 - **DatabaseName** = SISLFV
 - **Name** = PesquisaFitasAlugadas
 - Selecione a propriedade SQL e digite a seguinte linha de comando:
**Select Fit_Codigo, Fits_Codigo, Fits_Alugada, Fit_Nome, Fit_TipoAluguel, Tip_Nome, Tip_Valor
From Fitas, Fitass, TiposAluguel Where Fit_Codigo = Fits_Fita and Fit_TipoAluguel = Tip_Codigo and
Fits_Alugada = 'N'**
- 3) Faça um clique duplo em cima da Query **PesquisaFitasAlugadas** (aparecerá o Fields Editor), clique com o botão direito do mouse e selecione a opção **Add All Fields**.
- 4) Selecione o **DataSource** e altere as seguintes propriedades:
 - **DataSet** = PesquisaFitasAlugadas
 - **Name** = DsPesquisaFitasAlugadas
- 5) Ainda no **DMSQL** adicione mais um **Query** e um **DataSource** localizados na palheta **BDE** e **DataAccess** respectivamente;
- 6) Selecione a **Query** e altere as seguintes propriedades:
 - **DatabaseName** = SISLFV
 - **Name** = SQL

7) Selecione o **DataSource** e altere as seguintes propriedades:

- **DataSet = SQL**
- **Name = DsSQL**

8) Para podermos continuar precisaremos de um form para procurar fitas alugadas, então como já temos um form de procura de fitas, basta seguir os seguintes procedimentos:

- Selecione o FormPesquisaFitas;
- Clique com o botão direito do mouse em qualquer lugar do form e clique na opção **"Add to Repository..."**.
- Preceenha a janela que irá aparecer como mostrado na figura abaixo e clique em OK:



9) Após executar os procedimentos citados anteriormente, feche o seu projeto clicando em **File>Close All**.

10) Agora abra seu projeto novamente e após tê-lo aberto clique em **File>New>Other**, escolha a aba Forms e selecione o ícone **PesquisaFitasPadrao**;

11) Lembre-se que vc precisará alterar o **Caption** do Form para **"Pesquisa de Fitas Alugadas"**, o **Name** do Form para **"FormPesquisaFitasAlugadas"**;

12) Salve-o com o nome **"PesquisaFitasAlugadas.pas"**;

13) Clique em **Project/Options...** passe o **FormPesquisaFitasAlugadas** para o Lado **Available Forms**;

14) Selecione o DBGrid do **FormPesquisaFitasAlugadas** e altere a propriedade DataSource para **DMSQL.DsPesquisaFitasAlugadas**;

15) Agora vá até o evento onClick do Botão pesquisar e altere as linhas de comando substituindo as linhas mostradas abaixo:

```
procedure TFormPesquisaFitasAlugadas.BitPesquisarClick(Sender: TObject);
Var
vPesquisa:String;
begin
  Case RadioTipoPesquisa.ItemIndex of
    0: vPesquisa:=EditPesquisa.Text+'%';
    1: vPesquisa:='%'+EditPesquisa.Text+'%';
    2: vPesquisa:='%'+EditPesquisa.Text;
  end;

  DMSQL.PesquisaFitasAlugadas.Close;
  DMSQL.PesquisaFitasAlugadas.SQL.Clear;
  DMSQL.PesquisaFitasAlugadas.SQL.Add('Select Fit_Codigo,
  Fits_Codigo,Fit_Nome,Fits_Alugada,Fit_TipoAluguel, Tip_Valor, Tip_Nome from Fitas, Fitass,
  TiposAluguel Where Fit_Codigo = Fits_Fita and Fit_TipoAluguel = Tip_Codigo and Fits_Alugada =
  "N"');

  Case RadioProcurar.ItemIndex of
    0: Begin
      DMSQL.PesquisaFitasAlugadas.SQL.Add('and Fit_Nome Like :Nome ');
      DMSQL.PesquisaFitasAlugadas.Params[0].AsString:=vPesquisa;
    end;

    1: Begin
      DMSQL.PesquisaFitasAlugadas.SQL.Add('and Fit_Codigo =:Codigo ');
      DMSQL.PesquisaFitasAlugadas.Params[0].AsInteger:=StrToInt(EditPesquisa.Text);
    end;
  end;
```



```
2: Begin
  DMSQL.PesquisaFitasAlugadas.SQL.Add('and Fits_Codigo Like :Codigo ');
  DMSQL.PesquisaFitasAlugadas.Params[0].AsString:=vPesquisa;
end;
end;

DMSQL.PesquisaFitasAlugadas.Open;
end;
```

- 8º Passo (Criando e Configurando o Form para a Locação de Fitas) -

Criando o Form para a Locação de Fitas

- 1) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;
- 2) Salve o novo Form com o Nome de **“LocacaoFitas.pas”**;
- 3) Altere a propriedade **Name** do Form para **FormLocacaoFitas**;
- 4) Usando os componentes (Label, Bevel, Edit, BitBtn e StatusBar) crie seu form de Locação de Fitas, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado na figura abaixo):

- 5) No exemplo acima altere as seguintes propriedades:

- Selecione o **FormLocacaoFitas** e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;
 - **ActiveControl** para **BitNovaLocacao**;

- 6) Clique em **Project/Options...** passe o **FormLocacaoFitas** para o Lado **Available Forms**;

- 7) Clique no Menu **File>Use Unit**, escolha a unit **Tabelas** e clique em **OK**;

- 8) Declare as procedures antes da sessão **PRIVATE** da unit **LocacaoFitas**;

```
Procedure INSERIR_LOCACAO;
Procedure GRAVAR_CANCELAR_LOCACAO;
```

9) Crie (na unit LocacaoFitas) as seguintes procedures:

```
Procedure TFormLocacaoFitas.GRAVAR_CANCELAR_LOCACAO;
begin
  BitBtnLocacao.Enabled:=True;
  BitBtnGravar.Enabled:=False;
  BitBtnCancelar.Enabled:=False;
  BitBtnFechar.Enabled:=True;
  ListBoxFitas.Enabled:=False;
  ListBoxFitas.Color:=clBtnface;
  EditCodFita.Enabled:=False;
  EditCodFita.Color:=clBtnFace;
  EditCodFita.Font.Color:=clGray;
  BitBtnProcurarFita.Enabled:=False;
  BitBtnApagarFita.Enabled:=False;
  EditCodUsuario.Enabled:=False;
  EditCodUsuario.Color:=clBtnFace;
  EditCodUsuario.Font.Color:=clGray;
  BitBtnProcurarUsuario.Enabled:=False;
  LabelUsuario.Font.Color:=clGray;
  EditCodFita.Clear;
  EditCodUsuario.Clear;
  LabelUsuario.Caption:='Nenhum Usuário...';
  ListBoxFitas.Clear;
  LabelTotal.Caption:='Total:R$ 0,00';
  LabelTotal.Font.Color:=clGray;
  LabelData.Font.Color:=clGray;
  LabelData.Caption:='Data da Locação: 00/00/0000';
end;
```

```
Procedure TFormLocacaoFitas.INSERIR_LOCACAO;
begin
  BitBtnLocacao.Enabled:=False;
  BitBtnGravar.Enabled:=True;
  BitBtnCancelar.Enabled:=True;
  BitBtnFechar.Enabled:=False;
  ListBoxFitas.Enabled:=True;
  ListBoxFitas.Color:=clWindow;
  EditCodFita.Enabled:=True;
  EditCodFita.Color:=clWindow;
  EditCodFita.Font.Color:=clBlack;
  BitBtnProcurarFita.Enabled:=True;
  BitBtnApagarFita.Enabled:=True;
  EditCodUsuario.Enabled:=True;
  EditCodUsuario.Color:=clWindow;
  EditCodUsuario.Font.Color:=clBlack;
  BitBtnProcurarUsuario.Enabled:=True;
  LabelUsuario.Font.Color:=clMaroon;
  LabelTotal.Font.Color:=clNavy;
  LabelData.Font.Color:=clMaroon;
  LabelData.Caption:='Data da Locação: '+DateToStr(Date);
end;
```

10) Vá até o evento **OnClick** do Botão “**Nova Locação**” e digite as seguintes linhas de comando:

```
vTotal:=0;
INSERIR_LOCACAO;
EditCodUsuario.SetFocus;
```

11) Vá até o evento **OnClick** do Botão “**Gravar**” e digite as seguintes linhas de comando:

```
procedure TFormLocacaoFitas.BitBtnGravarClick(Sender: TObject);
var
  I:Integer;
  vPagamento:String;
begin
  I:=0;
  DM.Locacoes.Open;
  DM.FitasLocadas.Open;
  DM.Fitass.Open;
  If (Length(Trim(EditCodUsuario.Text)) > 0) and (ListBoxFitas.Count > 0) Then
  Begin

    If Application.MessageBox('Pagamento a vista?','Pergunta...',Mb_IconExclamation+Mb_YesNo)= MrYes
    Then
      vPagamento:='S'
    else
      vPagamento:='N';

    DM.Locacoes.Insert;
    DM.LocacoesLoc_Cliente.Value:=StrToInt(EditCodUsuario.Text);
    DM.LocacoesLoc_DataAluguel.Value:=Date;
    DM.LocacoesLoc_HoraAluguel.Value:=Time;
    DM.LocacoesLoc_Usuario.Value:=DMSQL.PesquisaUsuariosUsu_Codigo.Value;
    DM.Locacoes.Post;

    For I:=0 to ListBoxFitas.Count-1 do
    Begin
      DMSQL.PesquisaFitasAlugadas.Close;
      DMSQL.PesquisaFitasAlugadas.SQL.Clear;
      DMSQL.PesquisaFitasAlugadas.SQL.Add('Select Fit_Codigo,
Fits_Codigo,Fit_Nome,Fits_Alugada,Fit_TipoAluguel, Tip_Valor, Tip_Nome from Fitas, Fitass, TiposAluguel
Where Fit_Codigo = Fits_Fita and Fit_TipoAluguel = Tip_Codigo and Fits_Alugada = "N"');
      DMSQL.PesquisaFitasAlugadas.SQL.Add('and Fits_Codigo =:Codigo ');
      DMSQL.PesquisaFitasAlugadas.Params[0].AsString:=Copy(ListBoxFitas.Items.Strings[I],1,7);
      DMSQL.PesquisaFitasAlugadas.Open;

      DM.FitasLocadas.Insert;
      DM.FitasLocadasFitl_Locacao.Value:=DM.LocacoesLoc_Codigo.Value;
      DM.FitasLocadasFitl_Fitass.Value;
      DM.FitasLocadasFitl_Fitass.Value:=DMSQL.PesquisaFitasAlugadasFits_Codigo.Value;
      DM.FitasLocadasFitl_ValorAluguel.Value:=DMSQL.PesquisaFitasAlugadasTip_Valor.Value;
      DM.FitasLocadasFitl_Pago.Value:=vPagamento;
      DM.FitasLocadasFitl_Devolvido.Value:='N';
      DM.FitasLocadas.Post;

      DMSQL.SQL.Close;
      DMSQL.SQL.SQL.Clear;
      DMSQL.SQL.SQL.Add('Update Fitass set Fits_Alugada="S" Where Fits_Codigo=:Codigo');
      DMSQL.SQL.Params[0].AsString:=DMSQL.PesquisaFitasAlugadasFits_Codigo.Value;
      DMSQL.SQL.ExecSQL;
    end;

    GRAVAR_CANCELAR_LOCACAO;

    Application.MessageBox('Locação efetuada com sucesso...','Informação...',Mb_IconInformation+Mb_Ok);
  end
  else
    Application.MessageBox('Você precisa preencher os dados corretamente para poder
continuar...','Pergunta...',Mb_IconError+Mb_Ok);
  DM.Locacoes.Close;
  DM.FitasLocadas.Close;
  DM.Fitass.Close;
```

11) Vá até o evento **OnClick** do Botão “**Cancelar**” e digite a seguinte linha de comando:
GRAVAR_CANCELAR_LOCACAO;

12) Agora vá até o evento **OnKeyPress** do Edit “**EditCodUsuario**” e digite as seguintes linhas de comando:

```
If not( key in['0'..'9','#8] ) then
begin
  beep;
  key:=#0;
end;
```

13) Vá até o evento **OnExit** do Edit “**EditCodUsuario**” e digite as seguintes linhas de comando:

```
If (IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value) <> EditCodUsuario.Text) and
(Length(Trim(EditCodUsuario.Text)) > 0) Then
Begin
  DMSQL.PesquisaClientes.Close;
  DMSQL.PesquisaClientes.SQL.Clear;
  DMSQL.PesquisaClientes.SQL.Add('Select Cli_Codigo,Cli_Nome,Cli_Apelido from Clientes Where
Cli_Codigo =:Codigo ');
  DMSQL.PesquisaClientes.Params[0].AsInteger:= StrToInt(EditCodUsuario.Text);
  DMSQL.PesquisaClientes.Open;

  If DMSQL.PesquisaClientes.RecordCount > 0 Then
  Begin
    EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);
    LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;
  end
  else
  Begin
    EditCodUsuario.SetFocus;
    LabelUsuario.Caption:= 'Nenhum Usuário...';
  end;
end;

If Length(Trim(EditCodUsuario.Text)) = 0 Then
  LabelUsuario.Caption:= 'Nenhum Usuário...';

DMSQL.PesquisaClientes.Close;
```

14) Vá até o evento **OnClick** do BotãoLupa “**BitBtnProcurarUsuário**” e digite as seguintes linha des comando:

```
FormPesquisaClientes:=TFormPesquisaClientes.Create(Application);

DMSQL.PesquisaClientes.Open;

FormPesquisaClientes.ShowModal;
FormPesquisaClientes.Free;
FormPesquisaClientes:=Nil;

If DMSQL.PesquisaClientes.RecordCount > 0 Then
Begin
  EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);
  LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;
end;

DMSQL.PesquisaClientes.Close;
```

15) Vá até o evento **OnExit** do Edit “**EditCodFita**” e digite as seguintes linhas de comando:

```
If (Length(Trim(EditCodFita.Text)) > 0) Then //(DM.PesquisaTituloscodexemplar.Value <>
EditCodLivro.Text)
Begin
  DMSQL.PesquisaFitasAlugadas.Close;
  DMSQL.PesquisaFitasAlugadas.SQL.Clear;
  DMSQL.PesquisaFitasAlugadas.SQL.Add('Select Fit_Codigo,
Fits_Codigo,Fit_Nome,Fits_Alugada,Fit_TipoAluguel, Tip_Valor, Tip_Nome from Fitas, Fitass, TiposAluguel
Where Fit_Codigo = Fits_Fita and Fit_TipoAluguel = Tip_Codigo and Fits_Alugada = "N");
  DMSQL.PesquisaFitasAlugadas.SQL.Add('and Fits_Codigo =:Codigo');
```

```
DMSQL.PesquisaFitasAlugadas.Params[0].AsString:= EditCodFita.Text;
DMSQL.PesquisaFitasAlugadas.Open;
```

```
If DMSQL.PesquisaFitasAlugadas.RecordCount > 0 Then
Begin
  //EditCodFita.Clear;
  vTotal:=vTotal+DMSQL.PesquisaFitasAlugadasTip_Valor.Value;
  ListBoxFitas.Items.Add(DMSQL.PesquisaFitasAlugadasFits_Codigo.Value+' -
'+DMSQL.PesquisaFitasAlugadasFit_Nome.Value +' - ['+DMSQL.PesquisaFitasAlugadasTip_Nome.Value+' -
'+FormatFloat('R$ #,##0.00',DMSQL.PesquisaFitasAlugadasTip_Valor.Value)+']');
  LabelTotal.Caption:=FormatFloat('Total: R$ #,##0.00',vTotal);
end
else
  ShowMessage('Este livro não existe ou está emprestado!!!');

  EditCodFita.SetFocus;
end;

EditCodFita.Clear;
```

- 16) Vá até o evento **OnClick** do BotãoLupa “**BitBtnProcurarFita**” e digite as seguintes linhas de comando:

```
FormPesquisaFitasAlugadas:=TFormPesquisaFitasAlugadas.Create(Application);

DMSQL.PesquisaFitasAlugadas.Open;

FormPesquisaFitasAlugadas.ShowModal;
FormPesquisaFitasAlugadas.Free;
FormPesquisaFitasAlugadas:=Nil;

If DMSQL.PesquisaFitasAlugadas.RecordCount > 0 Then
Begin
  EditCodFita.Clear;
  vTotal:=vTotal+DMSQL.PesquisaFitasAlugadasTip_Valor.Value;
  ListBoxFitas.Items.Add(DMSQL.PesquisaFitasAlugadasFits_Codigo.Value+' -
'+DMSQL.PesquisaFitasAlugadasFit_Nome.Value +' - ['+DMSQL.PesquisaFitasAlugadasTip_Nome.Value+' -
'+FormatFloat('R$ #,##0.00',DMSQL.PesquisaFitasAlugadasTip_Valor.Value)+']');
  LabelTotal.Caption:=FormatFloat('Total: R$ #,##0.00',vTotal);
end;

DMSQL.PesquisaFitasAlugadas.Close;
```

- 17) Vá até o evento **OnClick** do Botão “**Apagar Fita Selecionada**” e digite as seguintes linhas de comando:

```
If ListBoxFitas.ItemIndex >= 0 Then
Begin
  If Application.MessageBox(Pchar('Deseja realmente apagar esta
Fita?'+#13+(''+ListBoxFitas.Items[ListBoxFitas.ItemIndex]+'')), 'Pergunta...', Mb_IconExclamation+Mb_YesNo)
= MrYes Then
  Begin
    DMSQL.PesquisaFitas.Close;
    DMSQL.PesquisaFitas.SQL.Clear;
    DMSQL.PesquisaFitas.SQL.Add('Select Fit_Codigo, Fits_Codigo, Fit_Nome, Fit_TipoAluguel, Tip_Valor,
Tip_Nome from Fitas, Fitass, TiposAluguel Where Fit_Codigo = Fits_Fita and Fit_TipoAluguel =
Tip_Codigo');
    DMSQL.PesquisaFitas.SQL.Add('and Fits_Codigo =:Codigo ');
    DMSQL.PesquisaFitas.Params[0].AsString:=Copy(ListBoxFitas.Items[ListBoxFitas.ItemIndex],1,7);
    DMSQL.PesquisaFitas.Open;

    ListBoxFitas.DeleteSelected;
    vTotal:= vTotal - DMSQL.PesquisaFitasTip_Valor.Value;
    LabelTotal.Caption:=FormatFloat('Total: R$ #,##0.00',vTotal);
  end;
end
else
  ShowMessage('Você deve primeiro selecionar uma fita!!!');
```

18) Vá até o evento **OnClick** do Botão “**Fechar**” e digite a seguintes linha de comando:

Close;

- 9º Passo (Criando e Configurando o Form para a Devolução de Fitas) -

Criando o Form para a Devolução de Fitas

1) Inicialmente vá até o **DMSQL** e adicione mais um **Query** e um **DataSource** localizados na palheta **BDE** e **DataAccess** respectivamente;

2) Selecione a **Query** e altere as seguintes propriedades:

- **DatabaseName** = SISLFV
- **Name** = PesquisaDevolucao
- Selecione a propriedade SQL e digite a seguinte linha de comando:

```
Select * from Clientes, Fitas, Fitass, Locacoes, FitasLocadas, TiposAluguel Where Clientes.Cli_Codigo =  
Locacoes.Loc_Cliente and Fitas.Fit_Codigo = Fitass.Fits_Fita and Fitass.Fits_Codigo =  
FitasLocadas.Fitl_Fitass and Locacoes.Loc_Codigo = FitasLocadas.Fitl_Locacao and  
Fitas.Fit_TipoAluguel = TiposAluguel.Tip_Codigo  
and Fitl_Devolvido = 'N' Order by Clientes.Cli_Nome
```

3) Faça um clique duplo em cima da Query **PesquisaDevolucao** (aparecerá o Fields Editor), clique com o botão direito do mouse e selecione a opção **Add All Fields**.

4) Selecione o **DataSource** e altere as seguintes propriedades:

- **DataSet** = PesquisaDevolucao
- **Name** = DsPesquisaDevolucao

5) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;

6) Salve o novo Form com o Nome de “**DevolucaoFitas.pas**”;

7) Altere a propriedade **Name** do Form para **FormDevolucaoFitas**;

8) Usando os componentes (Label, Bevel, Edit, BitBtn, DbGrid e StatusBar) crie seu form de Devolução de Fitas, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado na figura abaixo):

9) No exemplo acima altere as seguintes propriedades:

- Selecione o **FormDevolucaoFitas** e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **ActiveControl** para **EditCodUsuario**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;

10) Clique em **Project/Options...** passe o **FormDevolucaoFitas** para o Lado **Available Forms**;

11) Clique no Menu **File>Use Unit**, escolha a unit **TabelasSQL** e clique em **OK**;

12) Declare a procedure antes da sessão PRIVATE da unit LocacaoFitas;

Procedure PESQUISA_FITAS;

13) Crie (na unit DevolucaoFitas) a seguinte procedures:

```
procedure TFormDevolucaoFitas.PESQUISA_FITAS;
begin
  DMSQL.PesquisaDevolucao.Close;
  DMSQL.PesquisaDevolucao.SQL.Clear;
  DMSQL.PesquisaDevolucao.SQL.Add('Select * from Clientes, Fitas, Fitass, Locacoes, FitasLocadas,
  TiposAluguel Where Clientes.Cli_Codigo = Locacoes.Loc_Cliente and ');
  DMSQL.PesquisaDevolucao.SQL.Add('Fitas.Fit_Codigo = Fitass.Fits_Fita and Fitass.Fits_Codigo =
  FitasLocadas.Fitl_Fitass and Locacoes.Loc_Codigo = FitasLocadas.Fitl_Locacao and');
  DMSQL.PesquisaDevolucao.SQL.Add('Fitas.Fit_TipoAluguel = TiposAluguel.Tip_Codigo and
  Fitl_Devolvido = "N"');
  DMSQL.PesquisaDevolucao.SQL.Add('and Clientes.Cli_Codigo=:Codigo Order by Clientes.Cli_Nome');
  DMSQL.PesquisaDevolucao.Params[0].AsInteger:=DMSQL.PesquisaClientesCli_Codigo.Value;
  DMSQL.PesquisaDevolucao.Open;
end;
```

14) Agora vá até o evento **OnKeyPress** do Edit **“EditCodUsuario”** e digite as seguintes linhas de comando:

```
If not( key in['0'..'9',#8] ) then
begin
  beep;
  key:=#0;
end;
```

15) Vá até o evento **OnExit** do Edit **“EditCodUsuario”** e digite as seguintes linhas de comando:

```
If (IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value) <> EditCodUsuario.Text) and
(Length(Trim(EditCodUsuario.Text)) > 0) Then
Begin
  DMSQL.PesquisaClientes.Close;
  DMSQL.PesquisaClientes.SQL.Clear;
  DMSQL.PesquisaClientes.SQL.Add('Select Cli_Codigo,Cli_Nome,Cli_Apelido from Clientes Where
  Cli_Codigo =:Codigo ');
  DMSQL.PesquisaClientes.Params[0].AsInteger:= StrToInt(EditCodUsuario.Text);
  DMSQL.PesquisaClientes.Open;

  If DMSQL.PesquisaClientes.RecordCount > 0 Then
  Begin
    EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);
    LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;
    PESQUISA_FITAS;
  end
  else
  Begin
    EditCodUsuario.SetFocus;
    DMSQL.PesquisaDevolucao.Close;
    LabelUsuario.Caption:= 'Nenhum Usuário...';
  end;
end;

If Length(Trim(EditCodUsuario.Text)) = 0 Then
Begin
  LabelUsuario.Caption:= 'Nenhum Usuário...';
  DMSQL.PesquisaDevolucao.Close;
end;

DMSQL.PesquisaClientes.Close;
```

16) Vá até o evento **OnClick** do BotãoLupa “**BitBtnProcurarUsuário**” e digite as seguintes linha des comando:

```
FormPesquisaClientes:=TFormPesquisaClientes.Create(Application);

DMSQL.PesquisaClientes.Open;

FormPesquisaClientes.ShowModal;
FormPesquisaClientes.Free;
FormPesquisaClientes:=Nil;

If DMSQL.PesquisaClientes.RecordCount > 0 Then
Begin
  EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);
  LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;
end;

DMSQL.PesquisaClientes.Close;

PESQUISA_FITAS;
```

17) Vá até o evento **OnClick** do Botão “**BitBtnDevolverFitas**” e digite as seguintes linha de comando:

```
procedure TFormDevolucaoFitas.BitBtnDevolverFitasClick(Sender: TObject);
Var
{
  I:Integer;
  vTotal:Real;
  vPago:String;
begin
  I:=0;

  If (Length(Trim(EditCodUsuario.Text)) > 0) and (DMSQL.PesquisaDevolucao.RecordCount > 0) Then
  Begin
    If (DBGridFitas.SelectedRows.Count = 0) Then
    Begin
      Application.MessageBox('Você precisa selecionar pelo menos uma fita para ser
devolvida...','Pergunta...',Mb_IconError+Mb_Ok);
      ABORT;
    end;

    DM.FitasLocadas.Open;

    For I:=0 to DBGridFitas.SelectedRows.Count-1 do
    Begin
      DBGridFitas.DataSource.DataSet.Bookmark:=DBGridFitas.SelectedRows[I];

      If DMSQL.PesquisaDevolucaoFitl_Pago.Value = 'N' Then
        vTotal:=vTotal+DMSQL.PesquisaDevolucaoFitl_ValorAluguel.Value;
      end;

      If vTotal > 0 Then
      Begin
        Case Application.MessageBox(PChar('Vai pagar agora?'+#13+'Total --> '+FormatFloat('R$
#,#0.00',vTotal)), 'Pergunta...',Mb_IconExclamation+Mb_YesNoCancel)of
          IdYes: vPago:='S';
          IdNo: vPago:='N';
          IdCancel: ABORT;
        end;
      end;

      For I:=0 to DBGridFitas.SelectedRows.Count-1 do
      Begin
        DBGridFitas.DataSource.DataSet.Bookmark:=DBGridFitas.SelectedRows[I];

        If DM.FitasLocadas.FindKey([DMSQL.PesquisaDevolucaoFitl_Codigo.Value]) Then
        Begin
          DM.FitasLocadas.Edit;
```



```
DM.FitasLocadasFitl_Devolvido.Value:='S';
```

```
If DMSQL.PesquisaDevolucaoFitl_Pago.Value = 'N' Then
```

```
DM.FitasLocadasFitl_Pago.Value:=vPago;
```

```
DM.FitasLocadas.Post;
```

```
end;
```

```
DMSQL.SQL.Close;
```

```
DMSQL.SQL.SQL.Clear;
```

```
DMSQL.SQL.SQL.Add('Update Fitass set Fits_Alugada="N" Where Fits_Codigo=:Codigo');
```

```
DMSQL.SQL.Params[0].AsString:=DMSQL.PesquisaDevolucaoFits_Codigo.Value;
```

```
DMSQL.SQL.ExecSQL;
```

```
end;
```

```
DM.FitasLocadas.Close;
```

```
DMSQL.PesquisaClientes.Close;
```

```
DMSQL.PesquisaClientes.SQL.Clear;
```

```
DMSQL.PesquisaClientes.SQL.Add('Select Cli_Codigo,Cli_Nome,Cli_Apelido from Clientes Where  
Cli_Codigo =:Codigo ');
```

```
DMSQL.PesquisaClientes.Params[0].AsInteger:= StrToInt(EditCodUsuario.Text);
```

```
DMSQL.PesquisaClientes.Open;
```

```
PESQUISA_FITAS;
```

```
DMSQL.PesquisaClientes.Close;
```

```
Application.MessageBox('Devolução efetuada com sucesso...','Informação...',Mb_IconInformation+Mb_Ok);
```

```
end
```

```
else
```

```
Application.MessageBox('Você precisa preencher os dados corretamente para poder  
continuar...','Pergunta...',Mb_IconError+Mb_Ok);
```

```
end;
```

18) Selecione o **DBGridFitas** e altere a propriedade **DataSource** para **DMSQL.DsPesquisaDevolucao**;

19) Ainda com o **DBGridFitas** selecionado Expanda a opção **Options** e altere as seguintes propriedades:

```
dgRowSelect = True;
```

```
dgMultiSelect = True;
```

20) Vá até o evento OnClose do FormDevolucaoFitas e digite a seguinte linha de comando:

```
DMSQL.PesquisaDevolucao.Close;
```

21) Para finalizar vá até o **FormPrincipal** e no evento **OnClick** do menu **Movimento>Devolução de Fitas...** digite as seguintes linhas de comando:

```
FormDevolucaoFitas:=TFormDevolucaoFitas.Create(Application);
```

```
FormDevolucaoFitas.ShowModal;
```

```
FormDevolucaoFitas.Free;
```

- 10º Passo (Criando e Configurando o Form para Pagamento de Débitos) -

Criando o Form para Pagamento de Débitos

- 1) Inicialmente vá até o **DMSQL** e adicione mais um **Query** e um **DataSource** localizados na palheta **BDE** e **DataAccess** respectivamente;
- 2) Selecione a **Query** e altere as seguintes propriedades:
 - **DatabaseName** = SISLFV
 - **Name** = PesquisaDebitos
 - Selecione a propriedade SQL e digite a seguinte linha de comando:
Select * from Clientes, Fitas, Fitass, Locacoes, FitasLocadas, TiposAluguel Where Clientes.Cli_Codigo = Locacoes.Loc_Cliente and Fitas.Fit_Codigo = Fitass.Fits_Fita and Fitass.Fits_Codigo = FitasLocadas.Fitl_Fitass and Locacoes.Loc_Codigo = FitasLocadas.Fitl_Locacao and Fitas.Fit_TipoAluguel = TiposAluguel.Tip_Codigo and Fitl_Pago = 'N' Order by Clientes.Cli_Nome
- 3) Faça um clique duplo em cima da Query **PesquisaDebitos** (aparecerá o Fields Editor), clique com o botão direito do mouse e selecione a opção **Add All Fields**.
- 4) Selecione o **DataSource** e altere as seguintes propriedades:
 - **DataSet** = PesquisaDebitos
 - **Name** = DsPesquisaDebitos
- 5) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;
- 6) Salve o novo Form com o Nome de **“PagamentoDebitos.pas”**;
- 7) Altere a propriedade **Name** do Form para **FormPagamentoDebitos**;
- 8) Usando os componentes (Label, Bevel, Edit, BitBtn, DbGrid e StatusBar) crie seu form de Pagamento de Débitos, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado na figura abaixo):

- 9) No exemplo acima altere as seguintes propriedades:
 - Selecione o **FormPagamentoDebitos** e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **ActiveControl** para **EditCodUsuario**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;
- 10) Clique em **Project/Options...** passe o **FormPagamentoDebitos** para o Lado **Available Forms**;
- 11) Clique no Menu **File>Use Unit**, escolha a unit **TabelasSQL** e clique em **OK**;

12) Declare a procedure antes da sessão PRIVATE da unit PagamentoDebitos;

Procedure PESQUISA_FITAS;

13) Crie (na unit DevolucaoFitas) a seguinte procedure:

```
procedure TFormPagamentoDebitos.PESQUISA_FITAS;
begin
  DMSQL.PesquisaDebitos.Close;
  DMSQL.PesquisaDebitos.SQL.Clear;
  DMSQL.PesquisaDebitos.SQL.Add('Select * from Clientes, Fitas, Fitass, Locacoes, FitasLocadas,
TiposAluguel Where Clientes.Cli_Codigo = Locacoes.Loc_Cliente and ');
  DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_Codigo = Fitass.Fits_Fita and Fitass.Fits_Codigo =
FitasLocadas.Fitl_Fitass and Locacoes.Loc_Codigo = FitasLocadas.Fitl_Locacao and');
  DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_TipoAluguel = TiposAluguel.Tip_Codigo and Fitl_Pago =
"N");
  DMSQL.PesquisaDebitos.SQL.Add('and Clientes.Cli_Codigo=:Codigo Order by Clientes.Cli_Nome');
  DMSQL.PesquisaDebitos.Params[0].AsInteger:=DMSQL.PesquisaClientesCli_Codigo.Value;
  DMSQL.PesquisaDebitos.Open;
end;
```

14) Agora vá até o evento **OnKeyPress** do Edit “**EditCodUsuario**” e digite as seguintes linhas de comando:

```
If not( key in['0'..'9',#8] ) then
begin
  beep;
  key:=#0;
end;
```

15) Vá até o evento **OnExit** do Edit “**EditCodUsuario**” e digite as seguintes linhas de comando:

```
If (IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value) <> EditCodUsuario.Text) and
(Length(Trim(EditCodUsuario.Text)) > 0) Then
Begin
  DMSQL.PesquisaClientes.Close;
  DMSQL.PesquisaClientes.SQL.Clear;
  DMSQL.PesquisaClientes.SQL.Add('Select Cli_Codigo,Cli_Nome,Cli_Apelido from Clientes Where
Cli_Codigo =:Codigo ');
  DMSQL.PesquisaClientes.Params[0].AsInteger:= StrToInt(EditCodUsuario.Text);
  DMSQL.PesquisaClientes.Open;

  If DMSQL.PesquisaClientes.RecordCount > 0 Then
  Begin
    EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);
    LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;
    PESQUISA_FITAS;
  end
  else
  Begin
    EditCodUsuario.SetFocus;
    DMSQL.PesquisaDebitos.Close;
    LabelUsuario.Caption:= 'Nenhum Usuário...';
  end;
end;

If Length(Trim(EditCodUsuario.Text)) = 0 Then
Begin
  LabelUsuario.Caption:= 'Nenhum Usuário...';
  DMSQL.PesquisaDebitos.Close;
end;

DMSQL.PesquisaClientes.Close;
```

16) Vá até o evento **OnClick** do BotãoLupa “**BitBtnProcurarUsuário**” e digite as seguintes linha des comando:

```
FormPesquisaClientes:=TFormPesquisaClientes.Create(Application);

DMSQL.PesquisaClientes.Open;

FormPesquisaClientes.ShowModal;
FormPesquisaClientes.Free;
FormPesquisaClientes:=Nil;

If DMSQL.PesquisaClientes.RecordCount > 0 Then
Begin
  EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);
  LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;
end;

DMSQL.PesquisaClientes.Close;

PESQUISA_FITAS;
```

17) Vá até o evento **OnClick** do Botão “**BitBtnPagarFitas**” e digite as seguintes linha de comando:

```
procedure TFormPagamentoDebitos.BitBtnPagarFitasClick(Sender: TObject);
{
  Var
  I:Integer;
  vTotal:Real;
  vPago:String;
  begin
  I:=0;

  If (Length(Trim(EditCodUsuario.Text)) > 0) and (DMSQL.PesquisaDebitos.RecordCount > 0) Then
  Begin
    If (DBGridFitas.SelectedRows.Count = 0) Then
    Begin
      Application.MessageBox('Você precisa selecionar pelo menos uma fita para ser
paga...','Pergunta...',Mb_IconError+Mb_Ok);
      ABORT;
    end;

    DM.FitasLocadas.Open;

    For I:=0 to DBGridFitas.SelectedRows.Count-1 do
    Begin
      DBGridFitas.DataSource.DataSet.Bookmark:=DBGridFitas.SelectedRows[I];

      If DMSQL.PesquisaDebitosFitl_Pago.Value = 'N' Then
        vTotal:=vTotal+DMSQL.PesquisaDebitosFitl_ValorAluguel.Value;
      end;

      If vTotal > 0 Then
      Begin
        If Application.MessageBox(PChar('Deseja Continuar?'+#13+'Total a pagar: --> '+FormatFloat('R$
#,#0.00',vTotal)),'Pergunta...',Mb_IconExclamation+Mb_YesNo) = MrNo Then
          ABORT;
        end;

        For I:=0 to DBGridFitas.SelectedRows.Count-1 do
        Begin
          DBGridFitas.DataSource.DataSet.Bookmark:=DBGridFitas.SelectedRows[I];

          If DM.FitasLocadas.FindKey([DMSQL.PesquisaDebitosFitl_Codigo.Value]) Then
          Begin
            DM.FitasLocadas.Edit;

            If DMSQL.PesquisaDebitosFitl_Pago.Value = 'N' Then
              DM.FitasLocadasFitl_Pago.Value:='S';
```

```
DM.FitasLocadas.Post;
end;

end;

DM.FitasLocadas.Close;

DMSQL.PesquisaClientes.Close;
DMSQL.PesquisaClientes.SQL.Clear;
DMSQL.PesquisaClientes.SQL.Add('Select Cli_Codigo,Cli_Nome,Cli_Apelido from Clientes Where
Cli_Codigo =:Codigo ');
DMSQL.PesquisaClientes.Params[0].AsInteger:= StrToInt(EditCodUsuario.Text);
DMSQL.PesquisaClientes.Open;

PESQUISA_FITAS;

{
DMSQL.PesquisaClientes.Close;

Application.MessageBox('Pagamento de débito efetuado com
sucesso...','Informação...',Mb_IconInformation+Mb_Ok);
end
else
Application.MessageBox('Você precisa preencher os dados corretamente para poder
continuar...','Pergunta...',Mb_IconError+Mb_Ok);
end;
}
```

18) Selecione o **DBGridFitas** e altere a propriedade **DataSource** para **DMSQL.DsPesquisaDebitos**;

19) Ainda com o **DBGridFitas** selecionado Expanda a opção **Options** e altere as seguintes propriedades:

```
dgRowSelect = True;
dgMultiSelect = True;
```

20) Vá até o evento **OnClose** do **FormDevolucaoFitas** e digite a seguinte linha de comando:

```
DMSQL.PesquisaDebitos.Close;
```

21) Para finalizar vá até o **FormPrincipal** e no evento **OnClick** do menu **Movimento>Pagamento de Débitos...** digite as seguintes linhas de comando:

```
FormPagamentoDebitos:=TFormPagamentoDebitos.Create(Application);
FormPagamentoDebitos.ShowModal;
FormPagamentoDebitos.Free;
```

- 11º Passo (Criando e Configurando o Form para a Configuração do Sistema) -

Criando o Form para a Configuração do Sistema

1) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;

2) Salve o novo Form com o Nome de **“ConfiguracaoSistema.pas”**;

3) Altere a propriedade **Name** do Form para **FormConfiguracaoSistema**;

4) Usando os componentes (DbEdit, DBCombobox, Bevel e PageControl) crie seu form de Configuração de Sistema, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado na figura abaixo):

The image shows a Windows-style dialog box titled 'Configuração do Sistema'. Inside, there are several text input fields and a dropdown menu. The fields are labeled: 'Nome da Locadora:', 'Endereço:', 'Cidade:', 'CEP:', 'Fone:', and 'Fone 2:'. Each label is followed by a text box containing a placeholder text like 'DBEditNomeLocadora'. The 'Estado:' label is followed by a dropdown menu with 'DBCo' selected. At the bottom right, there is a button labeled 'Fechar'.

5) No exemplo acima altere as seguintes propriedades:

- Selecione o **FormConfiguracaoSistema** e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **ActiveControl** para **DBEditNomeLocadora**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;

6) Clique em **Project/Options...** passe o **FormConfiguracaoSistema** para o Lado **Available Forms**;

7) Clique no Menu **File>Use Unit**, escolha a unit **Tabelas** e clique em **OK**;

8) Agora selecione cada um dos componentes e altere suas propriedades **DataSource** e **DataField** para suas respectivas tabelas (DataSource) e Campos (DataField). Ex: O DBEditNomeLocadora deve ter seu **DataSource** ligado à **DM.DsConfiguracao** e seu **DataField** ligado à **Con_NomeLocadora**, e assim por diante; (Não esqueça de preencher a propriedade **ITEMS** do **ComboboxEstado** com todos os estados)

9) No Evento OnClose do FormConfiguracaoSistema digite a seguinte linha de comando:

```
DM.Configuracao.Post;
```

10) Para finalizar esta parte vá até o FormPrincipal e no menu Sistema>Configuração do Sistema digite as seguintes linhas de comando:

```
FormConfiguracaoSistema:=TFormConfiguracaoSistema.Create(Application);
```

```
DM.Configuracao.Open;
```

```
If DM.Configuracao.RecordCount > 0 Then
```

```
  DM.Configuracao.Edit
```

```
else
```

```
  DM.Configuracao.Insert;
```

```
FormConfiguracaoSistema.ShowModal;
```

```
FormConfiguracaoSistema.Free;
```

```
DM.Configuracao.Close;
```

- 12º Passo (Criando e Configurando o Form para o Histórico de Locações) -

Criando o Form para o Histórico de Locações

- 1) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;
- 2) Salve o novo Form com o Nome de “**HistoricoLocacoes.pas**”;
- 3) Altere a propriedade **Name** do Form para **FormHistoricoLocacoes**;
- 4) Usando os componentes (Edit, BitBtn, Bevel, PageControl, DBGrid e Label) crie seu form de Histórico de Locações, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado na figura abaixo):

Código	Nome do Cliente	Data do Aluguel	Hora do Aluguel	Nome do Usuário

Código	Nome da Fita	Valor do Aluguel	Pago ?	Devolvido ?	Hora da Devolução	Data da Devolução

- 5) No exemplo acima altere as seguintes propriedades:
 - Selecione o **FormHistoricoLocacoes** e altere as propriedades
 - **Position** para **poDesktopCenter**;
 - **ActiveControl** para **EditCodUsuario**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;
- 6) Clique em **Project/Options...** passe o **FormHistoricoLocacoes** para o Lado **Available Forms**;
- 7) Clique no Menu **File>Use Unit**, escolha a unit **Tabelas** e clique em **OK**;
- 8) Selecione o **DbGridLocacoes** e altere a propriedade **DataSource** para **DM.DsLocacoes**;
- 9) Selecione o **DbGridFitasLocadas** e altere a propriedade **DataSource** para **DM.DsFitasLocadas**;
- 10) Agora vá até o evento **OnKeyPress** do Edit “**EditCodUsuario**” e digite as seguintes linhas de comando:

```
If not( key in['0'..'9','#8'] ) then  
begin  
  beep;  
  key:=#0;  
end;
```

11) Vá até o evento **OnExit** do Edit “**EditCodUsuario**” e digite as seguintes linhas de comando:

```
If (IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value) <> EditCodUsuario.Text) and
(Length(Trim(EditCodUsuario.Text)) > 0) Then
Begin
  DMSQL.PesquisaClientes.Close;
  DMSQL.PesquisaClientes.SQL.Clear;
  DMSQL.PesquisaClientes.SQL.Add('Select Cli_Codigo,Cli_Nome,Cli_Apelido from Clientes Where
Cli_Codigo =:Codigo ');
  DMSQL.PesquisaClientes.Params[0].AsInteger:= StrToInt(EditCodUsuario.Text);
  DMSQL.PesquisaClientes.Open;

  If DMSQL.PesquisaClientes.RecordCount > 0 Then
  Begin
    EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);
    LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;

    DM.Locacoes.Filtered:=False;
    DM.Locacoes.Filter:='Loc_Cliente =' +EditCodUsuario.Text;
    DM.Locacoes.Filtered:=True;
  end
  else
  Begin
    DM.Locacoes.Filtered:=False;
    EditCodUsuario.SetFocus;
    LabelUsuario.Caption:= 'Nenhum Usuário...';
  end;
end;

If Length(Trim(EditCodUsuario.Text)) = 0 Then
Begin
  DM.Locacoes.Filtered:=False;
  LabelUsuario.Caption:= 'Nenhum Usuário...';
end;

DMSQL.PesquisaClientes.Close;
```

12) Vá até o evento **OnClick** do BotãoLupa “**BitBtnProcurarUsuário**” e digite as seguintes linha des comando:

```
FormPesquisaClientes:=TFormPesquisaClientes.Create(Application);

DMSQL.PesquisaClientes.Open;

FormPesquisaClientes.ShowModal;
FormPesquisaClientes.Free;
FormPesquisaClientes:=Nil;

If DMSQL.PesquisaClientes.RecordCount > 0 Then
Begin
  EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);
  LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;

  DM.Locacoes.Filtered:=False;
  DM.Locacoes.Filter:='Loc_Cliente =' +EditCodUsuario.Text;
  DM.Locacoes.Filtered:=True;
end;

DMSQL.PesquisaClientes.Close;
```

No DBgridLocacoes vamos precisar que apareça o nome do cliente que fez a locação, o nome do usuário que realizou a locação, o nome da fita locada e o código da fita no acervo ou seja precisaremos criar alguns campos virtuais, para isso siga as instruções a seguir:

- Criando um Campo Virtual (Tipo LookUp)

- 13) Vá até o **DataModule**, faça um duplo clique na tabela **Locacoes** (o editor de campos abrirá), clique com o botão direito do mouse no editor de campos e escolha a opção **New Field...**;
- 14) Preencha a tela como mostrado na figura abaixo:

The 'New Field' dialog box is shown with the following settings:

- Field properties:**
 - Name: NomeCliente
 - Component: LocacoesNomeCliente
 - Type: String
 - Size: 60
- Field type:**
 - ☐ Data
 - ☐ Calculated
 - ☒ Lookup
- Lookup definition:**
 - Key Fields: Loc_Cliente
 - Dataset: Clientes
 - Lookup Keys: Cli_Codigo
 - Result Field: Cli_Nome

Buttons: OK, Cancel, Help

- 15) Clique em Ok. (Você voltará para o editor de campos)
- 16) Clique novamente com o botão direito do mouse no editor de campos e escolha a opção **New Field...**;
- 17) Preencha novamente a janela como mostrado na figura abaixo:

The 'New Field' dialog box is shown with the following settings:

- Field properties:**
 - Name: NomeUsuario
 - Component: LocacoesNomeUsuario
 - Type: string
 - Size: 15
- Field type:**
 - ☐ Data
 - ☐ Calculated
 - ☒ Lookup
- Lookup definition:**
 - Key Fields: Loc_Usuario
 - Dataset: Usuarios
 - Lookup Keys: Usu_Codigo
 - Result Field: Usu_NomeUsuario

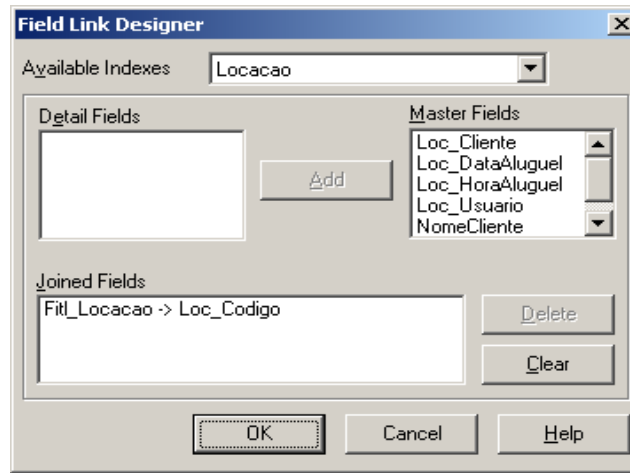
Buttons: OK, Cancel, Help

- 18) Clique em Ok.

Agora iremos criar um MasterSource para que quando nós selecionemos uma locação, automaticamente apareça as fitas locadas.

- Criando um MasterSource (Relação Locacao/FitasLocadas)

- 19) Vá até o **DataModule** e Selecione a tabela **FitasLocadas**, Altere a opção **MasterSource** para **DsLocacoes**; Selecione a propriedade **MasterFields** e clique nos 03 pontinhos; Veja a figura abaixo:



20) Para deixar a janela do mesmo jeito da apresentada acima, clique no combobox **Available Indexes** e Selecione **IdxFitass**, logo após selecione **Fitl_Fitass** na caixa **Detail Fields**, e depois selecione **Loc_Codigo** na caixa **Master Fields**, pronto, agora é só clicar no botão **Add** e em seguida, no botão **OK**;

21) Para finalizar vá até o **FormPrincipal** no menu **Consultas>Histórico de Locações** e digite as seguintes linhas de comando:

```
FormHistoricoLocacoes:=TFormHistoricoLocacoes.Create(application);
DM.Locacoes.Open;
DM.FitasLocadas.Open;
FormHistoricoLocacoes.ShowModal;
FormHistoricoLocacoes.Free;
DM.Locacoes.Filtered:=False;
DM.Locacoes.Close;
DM.FitasLocadas.Close;
```

- 13º Passo (Criando e Configurando o Form para consulta de Clientes em Débito) -

Criando o Form para a consulta de Clientes em Débito

- 1) Estando com o seu projeto aberto no Delphi, Crie um novo Form clicando em **File>New>Form**;
- 2) Salve o novo Form com o Nome de **“ClientesDebito.pas”**;
- 3) Altere a propriedade **Name** do Form para **FormClientesDebito**;
- 4) Usando os componentes (Edit, BitBtn, Bevel, PageControl, DBGrid e Label) crie seu form de Consulta de Clientes em Débito, se preferir use o exemplo abaixo (Lembre-se de alterar o nome de cada componente como mostrado na figura abaixo):

5) No exemplo acima altere as seguintes propriedades:

- Selecione o **FormClientesDebito** e altere as propriedades
 - **Position** para **poDesktopCenter**; - **ActiveControl** para **EditCodUsuario**;
 - **BorderIcons/biMaximize** para **False**;
 - **BorderStyle** para **bsSingle**;

6) Clique em **Project/Options...** passe o **FormClientesDebito** para o Lado **Avaliable Forms**;

7) Clique no Menu **File>Use Unit**, escolha a unit **TabelasSQL** e clique em **OK**;

8) Selecione o **DbGridDebitos** e altere a propriedade DataSource para **DMSQL.DsPesquisaDebitos**;

9) Declare a procedure antes da sessão PRIVATE da unit PagamentoDebitos;

```
procedure PESQUISA_DEBITOS;  
procedure TODOS_OS_DEBITOS;
```

10) Crie (na unitClientesDebito) as seguintes procedures:

```
procedure TFormClientesDebito.TODOS_OS_DEBITOS;  
Begin  
  DMSQL.PesquisaDebitos.Close;  
  DMSQL.PesquisaDebitos.SQL.Clear;  
  DMSQL.PesquisaDebitos.SQL.Add('Select * from Clientes, Fitas, Fitass, Locacoes, FitasLocadas,  
TiposAluguel Where Clientes.Cli_Codigo = Locacoes.Loc_Cliente and ');  
  DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_Codigo = Fitass.Fits_Fita and Fitass.Fits_Codigo =  
FitasLocadas.Fitl_Fitass and Locacoes.Loc_Codigo = FitasLocadas.Fitl_Locacao and');  
  DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_TipoAluguel = TiposAluguel.Tip_Codigo and Fitl_Pago =  
"N"');  
  DMSQL.PesquisaDebitos.SQL.Add('Order by Clientes.Cli_Nome');  
  DMSQL.PesquisaDebitos.Open;  
  
  DMSQL.SQL.Close;  
  DMSQL.SQL.SQL.Clear;  
  DMSQL.SQL.SQL.Add('Select Sum(Fitl_ValorAluguel) as TOTAL from FitasLocadas Where Fitl_Pago =  
"N" ');  
  DMSQL.SQL.Open;  
end;  
  
procedure TFormClientesDebito.PESQUISA_DEBITOS;  
Begin  
  DMSQL.PesquisaDebitos.Close;  
  DMSQL.PesquisaDebitos.SQL.Clear;  
  DMSQL.PesquisaDebitos.SQL.Add('Select * from Clientes, Fitas, Fitass, Locacoes, FitasLocadas,  
TiposAluguel Where Clientes.Cli_Codigo = Locacoes.Loc_Cliente and ');  
  DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_Codigo = Fitass.Fits_Fita and Fitass.Fits_Codigo =  
FitasLocadas.Fitl_Fitass and Locacoes.Loc_Codigo = FitasLocadas.Fitl_Locacao and');  
  DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_TipoAluguel = TiposAluguel.Tip_Codigo and Fitl_Pago =  
"N"');  
  DMSQL.PesquisaDebitos.SQL.Add('and Clientes.Cli_Codigo=:Codigo Order by Clientes.Cli_Nome');  
  DMSQL.PesquisaDebitos.Params[0].AsInteger:=DMSQL.PesquisaClientesCli_Codigo.Value;  
  DMSQL.PesquisaDebitos.Open;  
  
  DMSQL.SQL.Close;  
  DMSQL.SQL.SQL.Clear;  
  DMSQL.SQL.SQL.Add('Select Sum(Fitl_ValorAluguel) as TOTAL from FitasLocadas, Locacoes, Clientes  
Where Loc_Codigo = Fitl_Locacao and Loc_Cliente = Cli_Codigo and Fitl_Pago = "N" and  
Cli_Codigo=:Codigo');  
  DMSQL.SQL.Params[0].AsInteger:=StrToInt(EditCodUsuario.Text);  
  DMSQL.SQL.Open;  
  
  FormClientesDebito.LabelTotal.caption:=FormatFloat('Total a receber: R$ #,##0.00',  
DMSQL.SQL.FieldByName('TOTAL').Value);  
end;
```

11) Agora vá até o evento **OnKeyPress** do Edit “**EditCodUsuario**” e digite as seguintes linhas de comando:

```
If not( key in['0'..'9',#8] ) then  
begin  
  beep;  
  key:=#0;  
end;
```

12) Vá até o evento **OnExit** do Edit “**EditCodUsuario**” e digite as seguintes linhas de comando:

```
If (IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value) <> EditCodUsuario.Text) and  
(Length(Trim(EditCodUsuario.Text)) > 0) Then  
Begin  
  DMSQL.PesquisaClientes.Close;  
  DMSQL.PesquisaClientes.SQL.Clear;  
  DMSQL.PesquisaClientes.SQL.Add('Select Cli_Codigo,Cli_Nome,Cli_Apelido from Clientes Where  
Cli_Codigo =:Codigo ');  
  DMSQL.PesquisaClientes.Params[0].AsInteger:= StrToInt(EditCodUsuario.Text);  
  DMSQL.PesquisaClientes.Open;  
  
  If DMSQL.PesquisaClientes.RecordCount > 0 Then  
  Begin  
    EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);  
    LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;  
    PESQUISA_DEBITOS;  
  end  
else  
  Begin  
    TODOS_OS_DEBITOS;  
    EditCodUsuario.SetFocus;  
    LabelUsuario.Caption:= 'Nenhum Usuário...';  
  end;  
end;  
  
If Length(Trim(EditCodUsuario.Text)) = 0 Then  
Begin  
  TODOS_OS_DEBITOS;  
  LabelUsuario.Caption:= 'Nenhum Usuário...';  
end;  
  
DMSQL.PesquisaClientes.Close;
```

13) Vá até o evento **OnClick** do BotãoLupa “**BitBtnProcurarUsuário**” e digite as seguintes linha

```
FormPesquisaClientes:=TFormPesquisaClientes.Create(Application);  
  
DMSQL.PesquisaClientes.Open;  
  
FormPesquisaClientes.ShowModal;  
FormPesquisaClientes.Free;  
FormPesquisaClientes:=Nil;  
  
If DMSQL.PesquisaClientes.RecordCount > 0 Then  
Begin  
  EditCodUsuario.Text:=IntToStr(DMSQL.PesquisaClientesCli_Codigo.Value);  
  LabelUsuario.Caption:=DMSQL.PesquisaClientesCli_Nome.Value;  
  PESQUISA_DEBITOS;  
end;  
  
DMSQL.PesquisaClientes.Close;
```

14) Para finalizar vá até o **FormPrincipal** no menu **Consultas>Clientes em Débito...** e digite as seguintes linhas de comando:

```
FormClientesDebito:=TFormClientesDebito.Create(application);

DMSQL.PesquisaDebitos.Close;
DMSQL.PesquisaDebitos.SQL.Clear;
DMSQL.PesquisaDebitos.SQL.Add('Select * from Clientes, Fitas, Fitass, Locacoes, FitasLocadas,
TiposAluguel Where Clientes.Cli_Codigo = Locacoes.Loc_Cliente and ');
DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_Codigo = Fitass.Fits_Fita and Fitass.Fits_Codigo =
FitasLocadas.Fitl_Fitass and Locacoes.Loc_Codigo = FitasLocadas.Fitl_Locacao and');
DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_TipoAluguel = TiposAluguel.Tip_Codigo and Fitl_Pago =
"N"');
DMSQL.PesquisaDebitos.SQL.Add('Order by Clientes.Cli_Nome');
DMSQL.PesquisaDebitos.Open;

DMSQL.SQL.Close;
DMSQL.SQL.SQL.Clear;
DMSQL.SQL.SQL.Add('Select Sum(Fitl_ValorAluguel) as TOTAL from FitasLocadas Where Fitl_Pago =
"N" ');
DMSQL.SQL.Open;

FormClientesDebito.LabelTotal.caption:=FormatFloat('Total a receber: R$ #,##0.00',
DMSQL.SQL.FieldByName('TOTAL').Value);

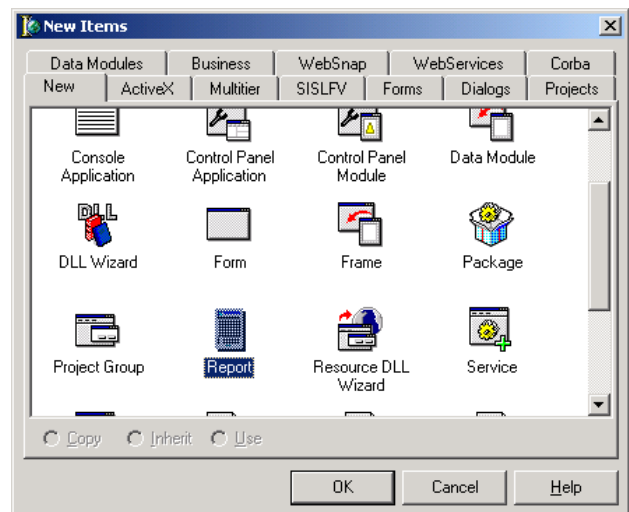
FormClientesDebito.ShowModal;
```

- 14º Passo (Criando Relatórios) -

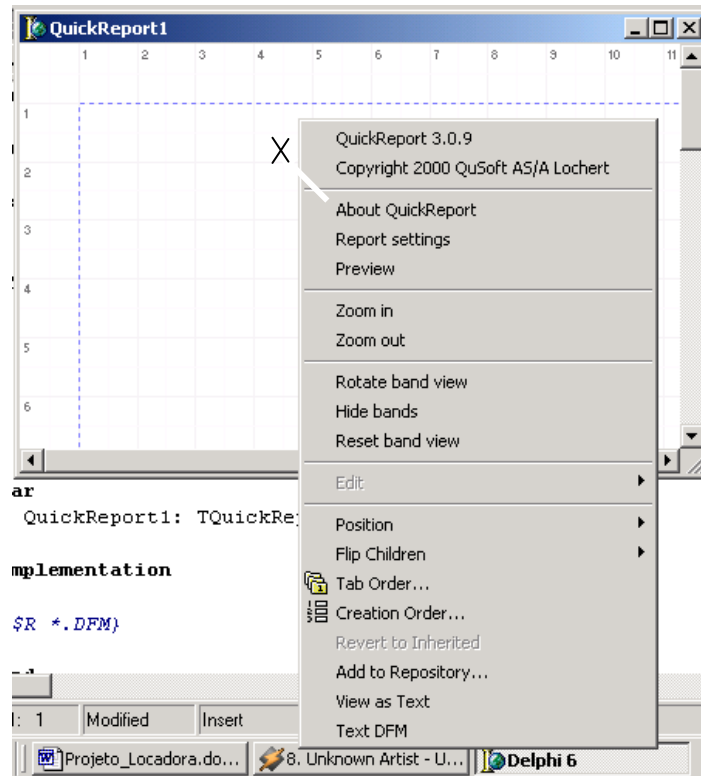
Criando o Relatório de Cadastro de Clientes

Nesta parte da apostila aprenderemos a criar relatórios usando o QuickReport que é um programa que já acompanha o delphi a bastante tempo, e que com certeza acompanhará nas próximas versões.

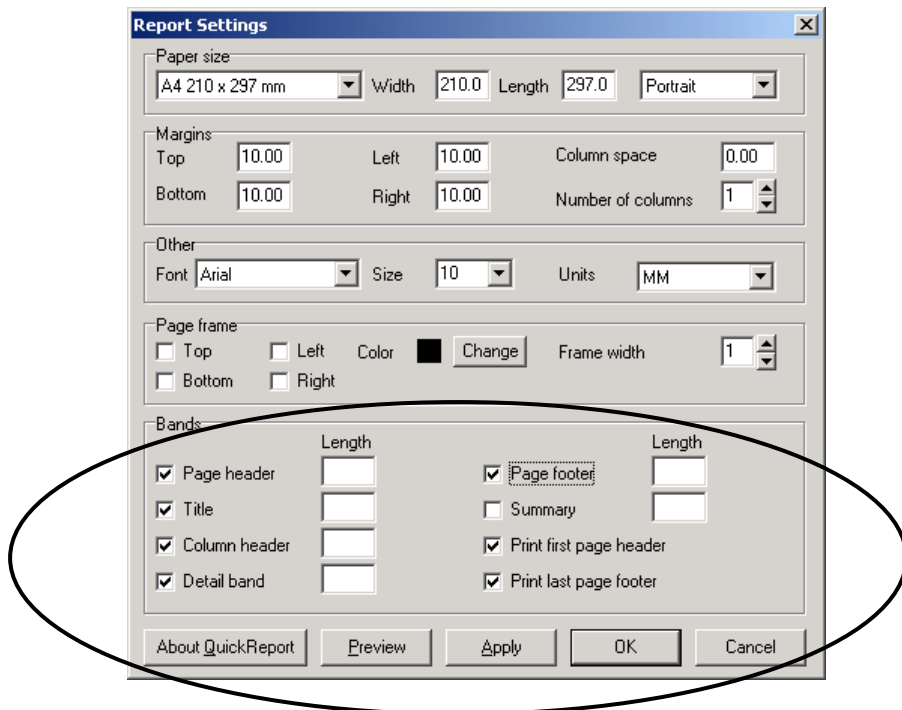
- 1) Para iniciar a criação do relatório de cadastro de clientes, clique no menu **File>New>Other...**, como de costume aparecerá a janela **New Items**.
- 2) Selecione o Ítem **Report** como mostra a figura abaixo, e logo após clique em **OK**.
- 3) Após clicar em OK aparecerá um form como mostrado na figura abaixo:



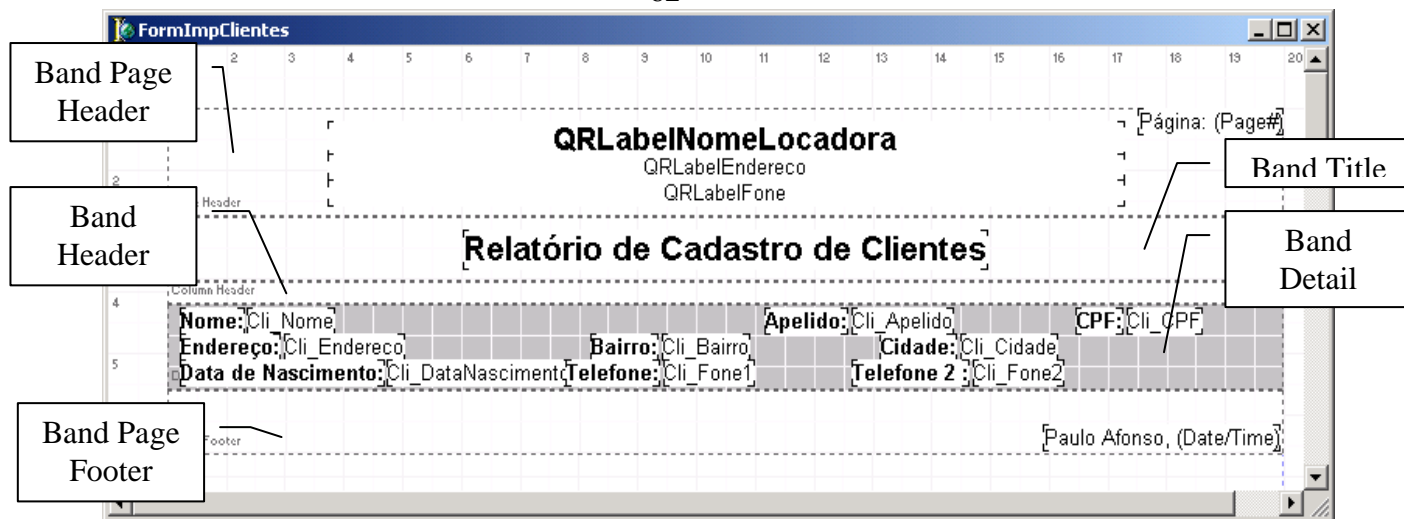
- 4) Salve o novo Form com o Nome de “**ImpClientes.pas**”;
- 5) Altere a propriedade **Name** do Form para **FormImpClientes**;
- 6) Clique em **Project/Options...** passe o **FormImpClientes** para o Lado **Available Forms**;
- 7) Agora clique com o botão direito do mouse em alguma parte do form, então aparecerá um menu PopUp como mostrado na figura abaixo:



- 7) Selecione a opção **Report settins** e então aparecerá a janela Report settins, aí é só selecionar as **Bands** como a figura abaixo:



- 8) Após selecionar as bands como na figura anterior adicione os componentes (palheta QReport) necessários para deixar seu Report igual ao da figura abaixo:



9) Note que a **Band Detail** está com a cor **clSilver** e que existe um **qrLabel** e um **QrDbText** para cada situação, como por exemplo o **QrLabel “Nome:”** e o **QrDbText “Cli_Nome”**; Note também que na **band Page Header** existe um componente **QRSysData** configurado com a propriedade **Text = Página** e a propriedade **data** igual a **qrsPageNumber**; Note ainda que existe um mesmo componente **QrSysdata** na **band Page Footer** com a propriedade **Text = Paulo Afonso**, e a propriedade **data** para **qrsDateTime**;

10) Clique no Menu **File>Use Unit**, escolha a unit **Tabelas** e clique em **OK**;

11) Após adicionar os componentes necessários nas devidas Bands selecione o **FormImpClientes** e alere a propriedade **DataSet** para **DM.Clientes**; Selecione todos os componentes da **Band Detail** e altere a propriedade **Transparent** para **True**;

12) Agora, selecione cada um dos **QrDBText** e altere a propriedade **DataSet** para **DM.Clientes** e a propriedade **DataField** para o devido Campo;

13) Selecione a **Band Page Header** e no evento **BeforePrint** digite as seguintes linhas de comando:

```
QrLabelNomeLocadora.Caption:=DM.ConfiguracaoCon_NomeLocadora.Value;
QrLabelEndereco.Caption:=DM.ConfiguracaoCon_Endereco.Value;
QrLabelFone.Caption:=DM.ConfiguracaoCon_Fone1.Value;
```

14) Selecione a **Band Detail** e no evento **BeforePrint** digite as seguintes linhas de comando:

```
If Odd(DM.Clientes.RecNo) Then
    DetailBand1.Color:=ClSilver
else
    DetailBand1.Color:=ClWhite;
```

13) Selecione o menu **Relatórios>Relatórios de Cadastro...>...de Clientes** no **FormPrincipal** e digite as seguintes linhas:

```
FormImpClientes:=TFormImpClientes.Create(Application);
DM.Clientes.Open;
DM.Configuracao.Open;
FormImpClientes.Preview;
FormImpClientes.Free;
DM.Clientes.Close;
DM.Configuracao.Close;
```

Criando o Relatório de Cadastro de Fitas

1) Para iniciar a criação do relatório de cadastro de fitas, clique no menu **File>New>Other...** . Como de costume aparecerá a janela **New Items**.

2) Selecione o Ítem **Report**, e logo após clique em **OK**.

3) Salve o novo Form com o Nome de **“ImpFitas.pas”**;

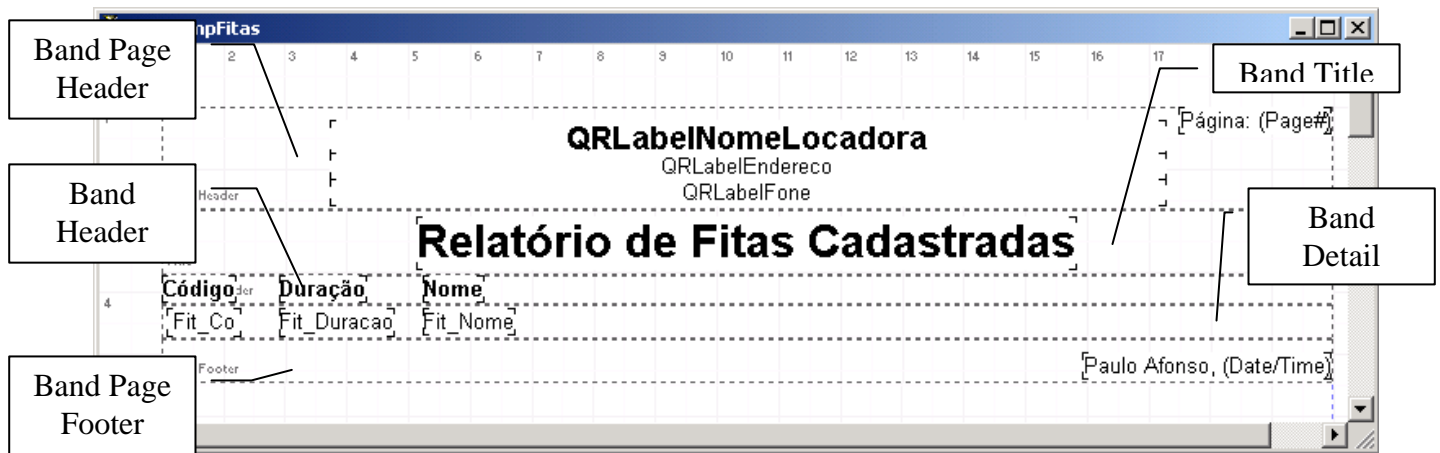
4) Altere a propriedade **Name** do Form para **FormImpFitas**;

5) Clique em **Project/Options...** passe o **FormImpFitas** para o Lado **Available Forms**;

6) Agora clique com o botão direito do mouse em alguma parte do form, e selecione a opção **Report settings**.

7) Na janela **Report Settings** selecione as Bands (**Page Header, Title, Header, Detail e Page Footer**) e clique em OK;

8) Adicione os componentes necessários para deixar o form como mostrado na figura abaixo:



9) Note que na band **Page Header** existe um componente **QRSysData** configurado com a propriedade **Text = Página** e a propriedade **data** igual a **qrsPageNumber**; Note ainda que existe um mesmo componente **QrSysdata** na band **Page Footer** com a propriedade **Text = Paulo Afonso**, e a propriedade **data** para **qrsDateTime**;

9.1) Clique no Menu **File>Use Unit**, escolha a unit **Tabelas** e clique em **OK**;

10) Após adicionar os componentes necessários nas devidas Bands selecione o **FormImpFitas** e alere a propriedade **DataSet** para **DM.Fitas**; Selecione todos os componentes da Band **Detail** e altere a propriedade **Transparent** para **True**;

11) Agora, selecione cada um dos **QrDBText** e altere a propriedade **DataSet** para **DM.Fitas** e a propriedade **DataField** para o devido Campo;

12) Selecione a Band **Page Header** e no evento **BeforePrint** digite as seguintes linhas de comando:

```
QrLabelNomeLocadora.Caption:=DM.ConfiguracaoCon_NomeLocadora.Value;  
QrLabelEndereco.Caption:=DM.ConfiguracaoCon_Endereco.Value;  
QrLabelFone.Caption:=DM.ConfiguracaoCon_Fone1.Value;
```

13) Selecione a Band **Detail** e no evento **BeforePrint** digite as seguintes linhas de comando:

```
If Odd(DM.Fitas.RecNo) Then  
    DetailBand1.Color:=ClSilver  
else  
    DetailBand1.Color:=ClWhite;
```

14) Selecione o menu **Relatórios>Relatórios de Cadastro...>...de Fitas** no **FormPrincipal** e digite as seguintes linhas:

```
FormImpFitas:=TFormImpFitas.Create(Application);  
DM.Fitas.Open;  
DM.Configuracao.Open;  
FormImpFitas.Preview;  
FormImpFitas.Free;  
DM.Fitas.Close;  
DM.Configuracao.Close;
```


Criando o Relatório de Movimento de Histórico de Locações

1) Inicialmente vá até o **DMSQL** e adicione mais um **Query** e um **DataSource** localizados na palheta **BDE** e **DataAccess** respectivamente;

2) Selecione a **Query** e altere as seguintes propriedades:

- **DatabaseName** = SISLFV
- **Name** = PesquisaLocacoes
- Selecione a propriedade SQL e digite a seguinte linha de comando:

```
Select * from Locacoes, FitasLocadas, Fitass, Fitas, Clientes, Usuarios
Where Locacoes.Loc_Codigo = FitasLocadas.Fitl_Locacao and
      FitasLocadas.Fitl_Fitass = Fitass.Fits_Codigo and
      Fitass.Fits_Fita = Fitas.Fit_codigo and
      Locacoes.Loc_Cliente = Clientes.Cli_Codigo and
      Locacoes.Loc_Usuario = Usuarios.Usu_Codigo
```

3) Faça um clique duplo em cima da Query **PesquisaLocacoes** (aparecerá o Fields Editor), clique com o botão direito do mouse e selecione a opção **Add All Fields**.

4) Selecione o **DataSource** e altere as seguintes propriedades:

- **DataSet** = PesquisaLocacoes
- **Name** = DsPesquisaLocacoes

5) Inicie a criação do relatório de Movimento de histórico de locações, clicando no menu **File>New>Other...** . Como de costume aparecerá a janela **New Items**.

6) Selecione o Ítem **Report**, e logo após clique em **OK**.

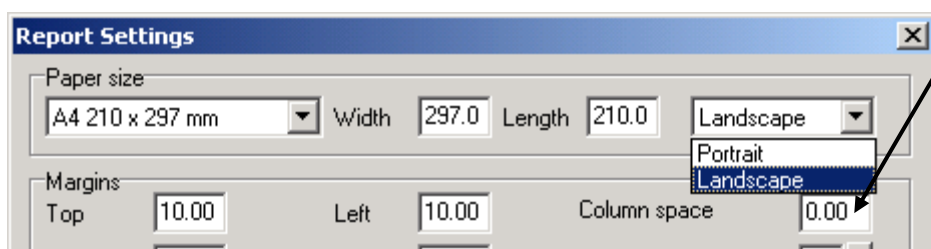
7) Salve o novo Form com o Nome de “**ImpHistoricoLocacoes.pas**”;

8) Altere a propriedade **Name** do Form para **FormImpHistoricoLocacoes**;

8.1) Clique em **Project/Options...** passe o **FormImpHistoricoLocacoes** para o Lado **Available Forms**;

8.2) Agora clique com o botão direito do mouse em alguma parte do form, e selecione a opção **Report settings**.

8.3) Na janela **Report Settings** selecione as Bands (**Page Header**, **Title**, **Header**, **Detail** e **Page Footer**); Altere a propriedade **PaperSize** para **Landscape** como mostra a figura abaixo; Clique em OK:



8.4) Adicione os componentes necessários para deixar o form como mostrado na figura abaixo:

9) Note que na band **Page Header** existe um componente **QRSysData** configurado com a propriedade **Text = Página** e a propriedade **data** igual a **qrsPageNumber**; Note ainda que existe um mesmo componente **QrSysdata** na band **Page Footer** com a propriedade **Text = Paulo Afonso**, e a propriedade **data** para **qrsDateTime**;

10) Clique no Menu **File>Use Unit**, escolha a unit **TabelasSQL** e clique em **OK**;

11) Após adicionar os componentes necessários nas devidas Bands selecione o **FormImpHistoricoLocacoes** e altere a propriedade **DataSet** para **DMSQL.PesquisaLocacoes**; Selecione todos os componentes da Band **Detail** e altere a propriedade **Transparent** para **True**;

12) Agora, selecione cada um dos **QrDBText** e altere a propriedade **DataSet** para **DMSQL.PesquisaLocacoes** e a propriedade **DataField** para o devido Campo;

13) Selecione a Band **Page Header** e no evento **BeforePrint** digite as seguintes linhas de comando:

```
QrLabelNomeLocadora.Caption:=DM.ConfiguracaoCon_NomeLocadora.Value;
QrLabelEndereco.Caption:=DM.ConfiguracaoCon_Endereco.Value;
QrLabelFone.Caption:=DM.ConfiguracaoCon_Fone1.Value;
```

14) Selecione a Band **Detail** e no evento **BeforePrint** digite as seguintes linhas de comando:

```
If Odd(DMSQL.PesquisaLocacoes.RecNo) Then
    DetailBand1.Color:=CISilver
else
    DetailBand1.Color:=CIWhite;
```

15) Selecione o menu **Relatórios>Relatórios de Movimento...>...de Histórico de Locações** no **FormPrincipal** e digite as seguintes linhas:

```
FormImpHistoricoLocacoes:=TFormImpHistoricoLocacoes.Create(Application);
DM.Configuracao.Open;
DMSQL.PesquisaLocacoes.Open;
FormImpHistoricoLocacoes.Preview;
FormImpHistoricoLocacoes.Free;
DMSQL.PesquisaLocacoes.Close;
DM.Configuracao.Close;
```

Criando o Relatório de Movimento de Clientes em Débito

1) Inicie a criação do relatório de Movimento de Clientes em Débito, clicando no menu **File>New>Other...** . Como de costume aparecerá a janela **New Items**.

2) Selecione o Ítem **Report**, e logo após clique em **OK**.

3) Salve o novo Form com o Nome de **"ImpClientesDebito.pas"**;

4) Altere a propriedade **Name** do Form para **FormImpClientesDebito**;

5) Clique em **Project/Options...** passe o **FormImpClientesDebito** para o Lado **Available Forms**;

6) Agora clique com o botão direito do mouse em alguma parte do form, e selecione a opção **Report settings**.

7) Na janela **Report Settings** selecione as Bands (**Page Header, Title, Header, Detail e Page Footer**); Altere a propriedade **PaperSize** para **Landscape**; Clique em OK:

8) Adicione os componentes necessários para deixar o form como mostrado na figura abaixo:

9) Note que na band **Page Header** existe um componente **QRSysData** configurado com a propriedade **Text = Página** e a propriedade **data** igual a **qrsPageNumber**; Note ainda que existe um mesmo componente **QrSysdata** na band **Page Footer** com a propriedade **Text = Paulo Afonso**, e a propriedade **data** para **qrsDateTime**;

10) Clique no Menu **File>Use Unit**, escolha a unit **TabelasSQL** e clique em **OK**;

11) Após adicionar os componentes necessários nas devidas Bands selecione o **FormImpDebitosCliente** e altere a propriedade **DataSet** para **DMSQL.PesquisaDebitos**; Selecione todos os componentes da Band **Detail** e altere a propriedade **Transparent** para **True**;

12) Agora, selecione cada um dos QrDBText e altere a propriedade **DataSet** para **DMSQL.PesquisaDebitos** e a propriedade **DataField** para o devido Campo;

13) Selecione a Band **Page Header** e no evento **BeforePrint** digite as seguintes linhas de comando:

```
QrLabelNomeLocadora.Caption:=DM.ConfiguracaoCon_NomeLocadora.Value;
QrLabelEndereco.Caption:=DM.ConfiguracaoCon_Endereco.Value;
QrLabelFone.Caption:=DM.ConfiguracaoCon_Fone1.Value;
```

14) Selecione a Band **Detail** e no evento **BeforePrint** digite as seguintes linhas de comando:

```
If Odd(DMSQL.PesquisaDebitos.RecNo) Then
    DetailBand1.Color:=CISilver
else
    DetailBand1.Color:=CIWhite;
```

15) Selecione o menu **Relatórios>Relatórios de Movimento...>...de Clientes em Débito** no **FormPrincipal** e digite as seguintes linhas:

```
FormImpClientesDebito:=TFormImpClientesDebito.Create(Application);

DM.Configuracao.Open;

DMSQL.PesquisaDebitos.Close;
DMSQL.PesquisaDebitos.SQL.Clear;
DMSQL.PesquisaDebitos.SQL.Add('Select * from Clientes, Fitas, Fitass, Locacoes, FitasLocadas,
TiposAluguel Where Clientes.Cli_Codigo = Locacoes.Loc_Cliente and ');
DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_Codigo = Fitass.Fits_Fita and Fitass.Fits_Codigo =
FitasLocadas.Fitl_Fitass and Locacoes.Loc_Codigo = FitasLocadas.Fitl_Locacao and');
DMSQL.PesquisaDebitos.SQL.Add('Fitas.Fit_TipoAluguel = TiposAluguel.Tip_Codigo and Fitl_Pago = "N"
Order by Locacoes.Loc_DataAluguel');
DMSQL.PesquisaDebitos.Open;

FormImpClientesDebito.Preview;
FormImpClientesDebito.Free;

DMSQL.PesquisaDebitos.Close;
DM.Configuracao.Close;
```

Finalizando o Programa

Bem, agora basta criar o **form sobre o sistema** do jeito que vc quiser e pronto, seu programa SISLFV já pode ser usado...

OBS: Sabemos que o programa que desenvolvemos contém muitos erros, mas, ele serve apenas para demonstrar a funcionabilidade de alguns componentes e de alguns comandos no Delphi...

Espero que tenham gostado e que tenham principalmente aprendido aquilo que tentei transmitir durante esses meses.

Lembre-se que todos podem contar comigo sempre que precisarem, é só telefonar...

Espero vc's na turma de delphi Módulo II... ;-)

Há.. desculpem os erros gramaticais da apostila, é que o corretor ortográfico do Word está desativado.

[]'s

Jackson Pires de O. S. Júnior
Téc. em Informática
Paulo Afonso - BA

Cientes.DB

	Field Name	Type	Size	Key
1	Cli_Codigo	+		*
2	Cli_Nome	A	60	
3	Cli_Endereco	A	60	
4	Cli_NumeroCasa	A	10	
5	Cli_Complemento	A	20	
6	Cli_Bairro	A	30	
7	Cli_Cidade	A	30	
8	Cli_Estado	A	2	
9	Cli_CEP	A	8	
10	Cli_CPF	A	11	
11	Cli_Identidade	A	15	
12	Cli_OrgEmissor	A	10	
13	Cli_DataEmissao	D		
14	Cli_CompResidencia	A	1	
15	Cli_DataCompResid	D		
16	Cli_Fone1	A	15	
17	Cli_Fone2	A	15	
18	Cli_Sexo	A	1	
19	Cli_DataNascimento	D		
20	Cli_Apelido	A	15	
21	Cli_OBS	M	240	
22	Cli_Foto	B	240	

Configuracao.DB

	Field Name	Type	Size	Key
1	Con_Codigo	+		*
2	Con_NomeLocadora	A	45	
3	Con_Endereco	A	45	
4	Con_Cidade	A	30	
5	Con_Estado	A	2	
6	Con_CEP	A	8	
7	Con_Fone1	A	15	
8	Con_Fone2	A	15	

Dependentes.DB

	Field Name	Type	Size	Key
1	Dep_Codigo	+		*
2	Dep_Cliente	I		
3	Dep_Nome	A	60	
4	Dep_Parentesco	I		
5	Dep_Apelido	A	15	
6	Dep_DataNascimento	D		

Fitas.DB

	Field Name	Type	Size	Key
1	Fit_Codigo	+		*
2	Fit_Nome	A	45	
3	Fit_Categoria	I		
4	Fit_Duracao	T		
5	Fit_SistGravacao	A	1	
6	Fit_Produtora	I		
7	Fit_DubLeg	A	1	
8	Fit_TipoAluguel	I		
9	Fit_Sinopse	M	240	
10	Fit_OBS	M	240	

FitasLocadas.DB

	Field Name	Type	Size	Key
1	Fitl_Codigo	+		*
2	Fitl_Fitass	A	7	
3	Fitl_ValorAluguel	\$		
4	Fitl_Pago	A	1	
5	Fitl_Devolvido	A	1	
6	Fitl_HoraDevolvido	T		
7	Fitl_DataDevolvido	D		
8	Fitl_Locacao	I		

FitasS.DB

	Field Name	Type	Size	Key
1	Fits_Codigo	A	7	*
2	Fits_Fita	I		
3	Fits_NumCopia	A	5	
4	Fits_Alugada	A	1	

Generos.DB

	Field Name	Type	Size	Key
1	Gen_Codigo	+		*
2	Gen_Nome	A	15	
3	Gen_Abreviacao	A	2	
4	Gen_UltimoCodigo	A	5	

Locacoes.DB

	Field Name	Type	Size	Key
1	Loc_Codigo	+		*
2	Loc_Cliente	I		
3	Loc_DataAluguel	D		
4	Loc_HoraAluguel	T		
5	Loc_Usuario	I		

Parentesco.DB

	Field Name	Type	Size	Key
1	Par_Codigo	+		*
2	Par_Nome	A	15	

Produtoras.DB

	Field Name	Type	Size	Key
1	Pro_Codigo	+		*
2	Pro_Nome	A	30	
3	Pro_URL	A	100	
4	Pro_OBS	M	240	

TiposAluguel.DB

	Field Name	Type	Size	Key
1	Tip_Codigo	+		*
2	Tip_Nome	A	15	
3	Tip_Valor	\$		

Usuarios.DB

	Field Name	Type	Size	Key
1	Usu_Codigo	+		*
2	Usu_Nome	A	60	
3	Usu_NomeUsuario	A	15	
4	Usu_Senha	A	15	
5	Usu_Endereco	A	60	
6	Usu_Cidade	A	30	
7	Usu_Estado	A	2	
8	Usu_Fone1	A	15	