

Lazarus - Exemplo MDI

Nesta aula criaremos uma aplicação simples de cadastro para avaliar o nível de dificuldade/facilidade que é desenvolver *softwares* com a tecnologia Object Pascal.

Na IDE Lazarus, crie um novo projeto e dê o nome de "ExemploMDI". Escolha um local adequado para salvar o seu projeto.

Na construção deste sistema simplificaremos ao máximo o nível de detalhamento das telas, pois, temos três fatores primordiais a serem levados em consideração:

- não é objetivo desta aula ensinar como se deve construir de sistemas de informação,
- o tempo é curto e
- quanto mais simples for o sistema, mais fácil será para avaliarmos o grau de dificuldade de utilização desta tecnologia.

Sua tela do Lazarus deve estar semelhante à apresentada na Figura 1.

Ao abrir o Lazarus, provavelmente haverá um novo formulário já criado no ambiente. Se assim for, basta você acessar o *menu File | Save All* que o Lazarus lhe pedirá um nome para o projeto, que terá a extensão (.lpr), e um nome para a nova *unit* criada (o formulário), que terá a extensão (.pas). Faça o seguinte:

- ao projeto, dê o nome de "ExemploMDI" e
- a *unit* dê o nome de "Principal", pois, este será nosso formulário principal.

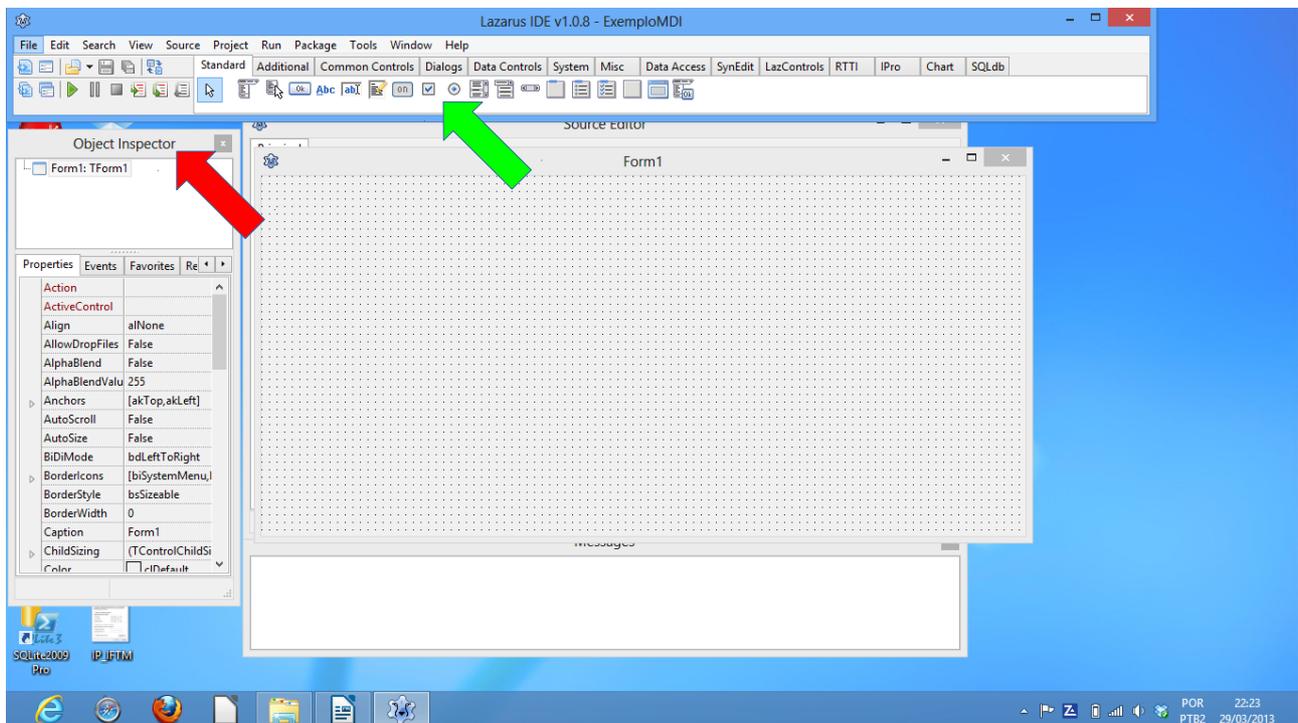


Figura 1: Criação do Projeto "ExemploMDI"

Antes de darmos início ao nosso projeto, vale ressaltar dois pontos importantes sobre esta IDE. São eles:

- na seta verde (Figura 1) podemos ver a aba de componentes do Lazarus de nome *Standard* e
- na seta vermelha (Figura 1) podemos ver o chamado *Object Inspector* que é o lugar onde são mostrados todos os eventos e todas as propriedades dos objetos. Basta clicar num objeto qualquer e o *Object Inspector* automaticamente mostrará suas propriedades e seus eventos.

Tela “Principal”

A tela Principal será constituída apenas de um *menu* para acesso aos demais formulários que comporão nosso sistema.

Adicionando Componentes ao Formulário

Vamos moldar a tela principal, faça o seguinte:

- primeiramente, procure no *Object Inspector*, deste ponto em diante referenciado apenas por OI, a propriedade *Name* e coloque *frmPrincipal*;
- procure a propriedade *Caption* e escreva *Principal*,
- ainda no OI, procure a propriedade *FormStyle* e escolha, dentre as opções disponíveis, a opção *fsMDIForm*,
- na propriedade *Position* escolha a opção *poScreenCenter* e
- na propriedade *WindowState* escolha *wsMaximized*.

Agora vamos componentizar nosso formulário, que também é um componente:

- na aba *Standard* clique no componente *TMainMenu*, arraste-o e cole-o no *frmPrincipal*. Dê um clique duplo no componente *TMainMenu* ou então, clique com o botão direito do *mouse* encima do componente e escolha a opção *Menu Editor*;
- no editor que foi apresentado (Figura 2) pode-se observar que foi criado um *New Item1* (seta vermelha na Figura 2). Este já é o primeiro item de *menu* do nosso formulário principal;
- clique com o botão direito do *mouse* sobre o *New Item1* e escolha a opção *Insert New Item (after)*, então aparecerá um *New Item2*. Faça isto mais uma vez e então teremos os três itens de *menu* que precisamos;
- agora dê um clique simples no *New Item1* e no OI procure a propriedade *Name* e altere-a para *mnuArquivo*. Procure a propriedade *Caption* e digite a palavra *Arquivo*;
- dê um clique simples no *New Item2* e mude as respectivas propriedades para *mnuCadastro* e *Cadastro* e para o *New Item3* mude para *mnuAjuda* e *Ajuda*.

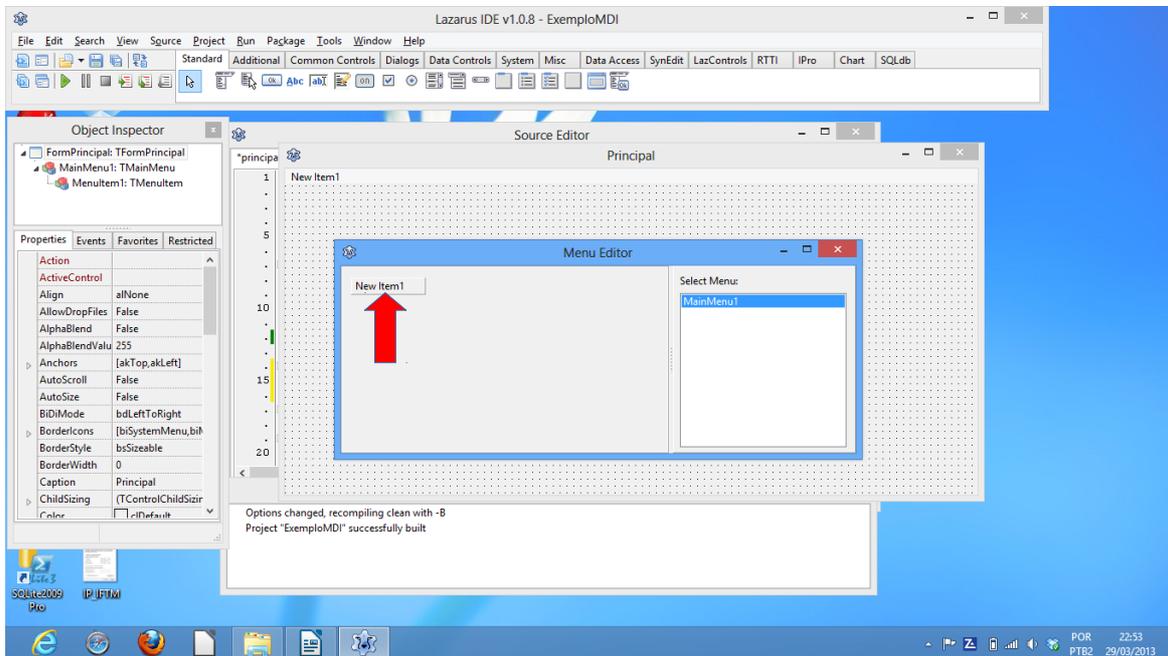


Figura 2: Inclusão de itens de menu no formulário Principal.

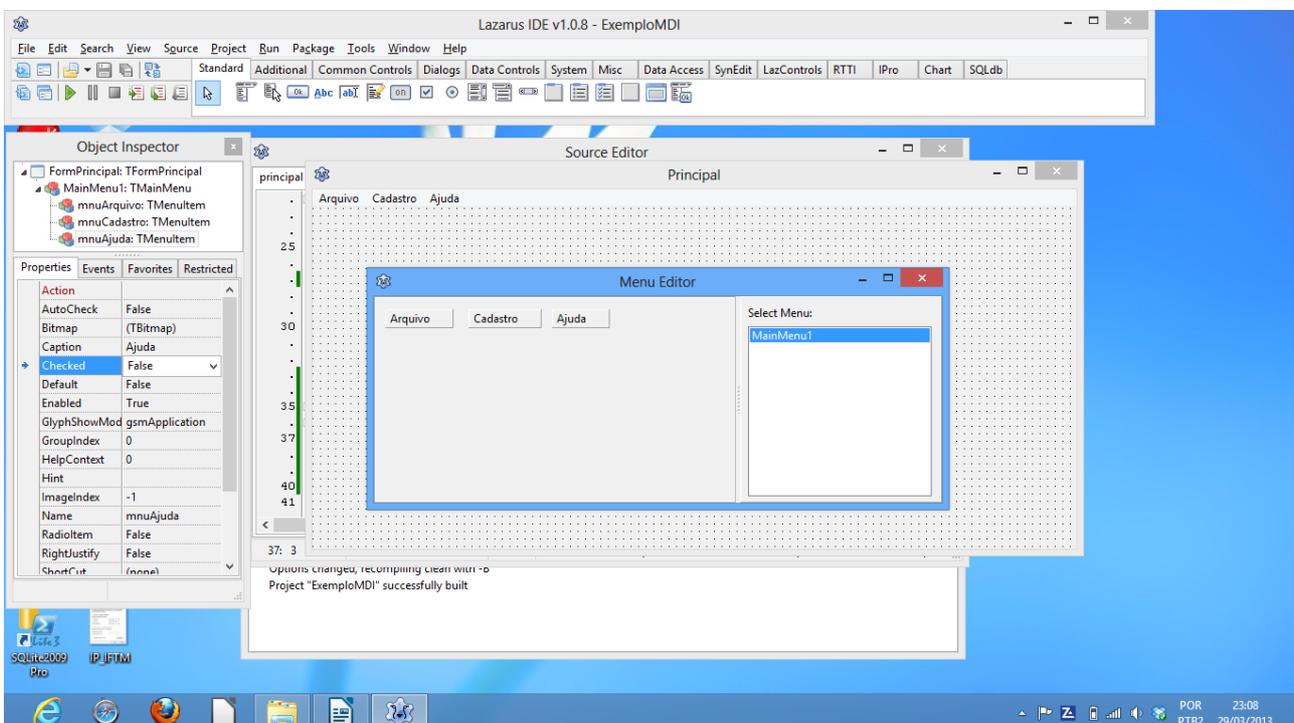


Figura 3: Dando nome aos itens de menu.

Neste momento sua tela deverá estar semelhante à mostrada na Figura 3.

Agora vamos criar os *submenus*. Então façamos o seguinte:

- dê um clique com o botão direito do *mouse* sobre o *item de menu* *Arquivo* e no *menu* que aparecer escolha a opção *Create Submenu*. Aparecerá um novo *item de menu*, porém, este subordinado ao *item de menu* *Arquivo*. Configure-o de uma vez, ou seja, no *OI*, na propriedade *Name* coloque *mnuSair* e na propriedade *Caption* coloque *Sair*;
- crie os seguintes *submenus* para *Cadastro*:
 - *Cliente* e *Estado* e configure as respectivas propriedades de ambos com o padrão que estamos utilizando e
- para o *item de menu* *Ajuda*, crie o último *submenu*:
 - *Sobre* e configure as respectivas propriedades que estamos utilizando.

Pronto! Feito isto, nosso formulário *Principal* já estará pronto para chamar os demais formulários que comporão nosso *software*.

Antes de criarmos os demais formulários, façamos uma última mudança no formulário *Principal*.

Dê um clique simples no *item de menu* *Sair*, como mostrado na Figura 4 (seta vermelha).

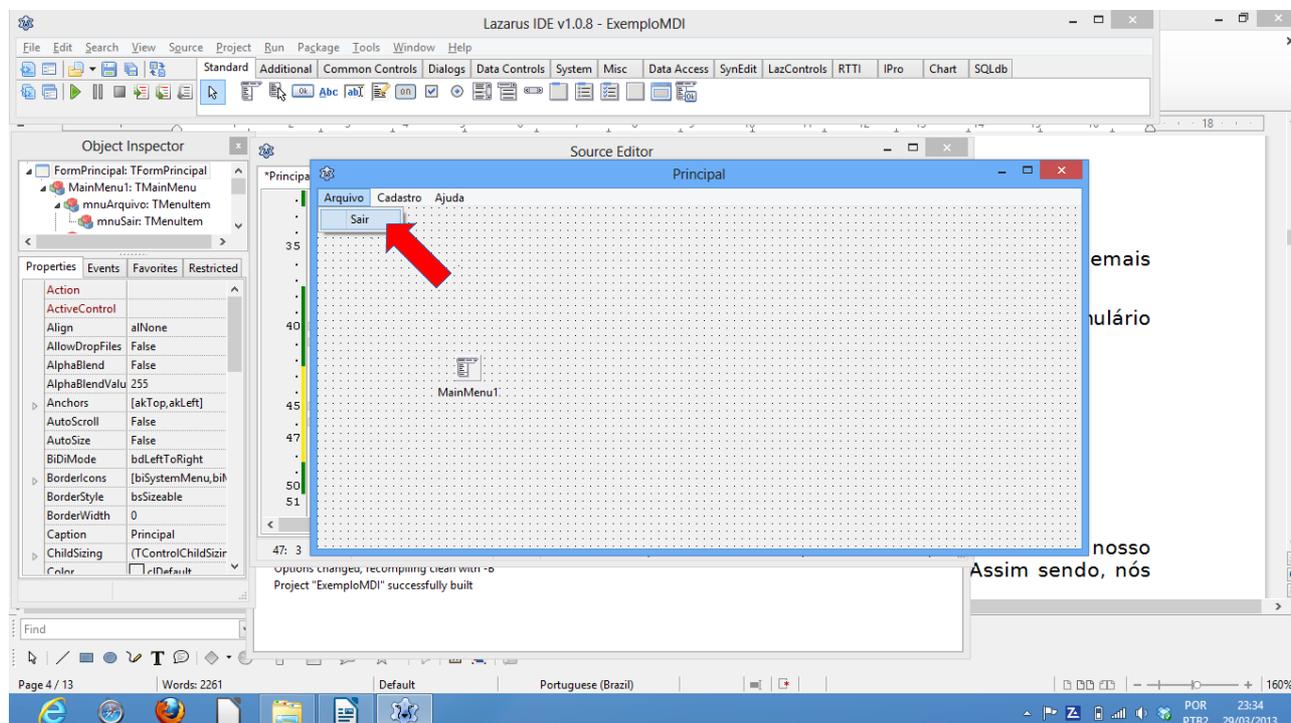


Figura 4: Clique no menu *Sair*.

Aparecerá uma tela mostrando que foi criado o evento para o *item de menu* *Sair* e o cursor do *mouse* já estará posicionado entre os comandos: *begin* e *end* do evento criado. Então digite o comando *Close*; como mostrado a seguir:

```
procedure TfrmPrincipal.mnuSairClick(Sender: TObject);  
begin  
    Close;  
end;
```

Agora clique em qualquer parte vazia do formulário *Principal* e no OI acesse a aba *Events*. Nela procure o evento *OnClose* e dê um clique duplo na área em frente ao nome do evento. Então, o evento será criado e o editor será aberto para que você possa digitar o seguinte código:

```
procedure TfrmPrincipal.FormClose(Sender: TObject;  
    var CloseAction: TCloseAction);  
begin  
    { Libera a memória ocupada pelo formulário. }  
    CloseAction := caFree;  
    frmPrincipal := nil;  
end;
```

Nota: Na linguagem Pascal os comentários são feitos utilizando-se “{ }” (chaves) ou “(* *)” (parêntesis asterisco e vice-versa).

Criando os Outros Formulários do Software

Vamos criar os outros formulários que devem estar presentes no sistema. No nosso caso, nosso banco de dados possui apenas duas tabelas: *Cliente* e *Estado*. Assim sendo, nós criaremos mais dois formulários. São eles: *Clientes* e *Estados*.

Sendo que os formulários de *Cliente* e *Estados* deverão ser dois para cada qual, pois, será um para visualização e outro para manipulação dos dados.

Então vamos lá:

- clique no item de *menu File | New...* e na tela que abrir (Figura 5) escolha a opção *Form*. Então será criado um novo formulário no projeto. Vamos ao OI e vamos modificar a propriedade *Name* para *frmClientesVisao*, a propriedade *Caption* para *Clientes*, a propriedade *FormStyle* para *fsMDIChild* e a propriedade *WindowState* para *wsMaximized*. Vá até o *menu File | Save*, na caixa de diálogo que aparecer digite o nome *FormClientesVisao* e em seguida pressione o botão *Salvar*;
- crie um outro formulário, vá ao OI e mude suas propriedades: *Name* para *frmClienteEdicao* e *Caption* para *Cliente*. Vá até o *menu File | Save*, na caixa de diálogo

que aparecer digite o nome *FormClienteEdicao* e em seguida pressione o botão *Salvar*;

- siga os mesmos passos para a criação dos formulários *frmEstadosVisao* e *frmEstadoEdicao*. Observe que os passos para criar o formulário de Visão são diferentes dos passos para criar o formulário de Edição;
- por último crie o formulário com a propriedade *Name* de *frmSobre*, a propriedade *Position* com *poScreenCenter* e a propriedade *Caption* com *Sobre*. Não se esqueça de salvá-lo.

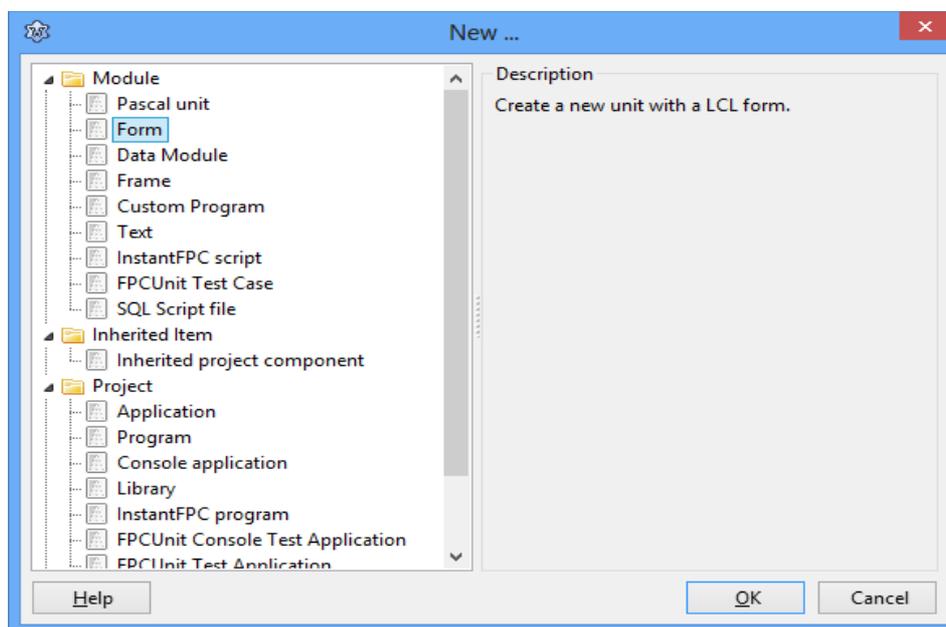


Figura 5: Janela para criação de novo tipo de formulário ou projeto.

Acabamos de criar os formulários para visualização e edição de dados.

Agora vá até o menu *File | Close All Editor Files*. Após clicar neste item de menu, todos os formulários que estavam abertos serão fechados. Muito bem, vamos aprender como abrir novamente o formulário ou a *unit* (código-fonte do formulário) desejado:

- para abrir o formulário vá até a opção de menu *Project | Forms ...*, na janela que aparecer é só você escolher o formulário desejado;
- para abrir a *unit* desejada vá até a opção *Project | Units ...* ou ainda,
- pode-se abrir a janela *Project Inspector* para se ter uma visão de toda a composição do seu projeto. Ela se encontra no item de menu *Project | Project Inspector* (Figura 6).

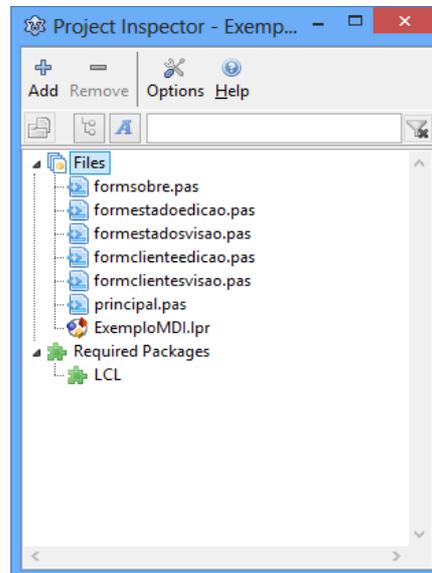


Figura 6: Tela Project Inspector.

Agora vamos criar os devidos componentes para visualização e edição nos respectivos formulários:

- abra o formulário *frmClientesVisao*. Se ao abrir o formulário aparecer apenas a *unit*, aperte a tecla *F12* que o formulário daquela *unit* aparecerá;
- após aberto o formulário, vá até a aba de componentes *Standard* e selecione o componente *TPanel*. Cole-o no *frmClientesVisao* e em seguida vá ao OI, na propriedade *Align* escolha *alTop*. Agora pegue outro *TPanel*, coloque-o abaixo do *TPanel* já existente no *frmClientesVisao* e na propriedade *Align* deste *TPanel* escolha a opção *alClient*. Seu formulário deverá estar semelhante ao apresentado na Figura 7. Mude a propriedade *Name* do primeiro *TPanel* para *pnlBotoesClientes* e a do segundo para *pnlGridClientes*. Apague a propriedade *Caption* de ambos.
- agora vá até a aba de componentes *Standard* e selecione o componente *TButton*. Cole quatro *TButtons* no *pnlBotoesClientes* do formulário. Agora dê um clique simples no primeiro botão, apenas para selecioná-lo. Vamos ao OI e na propriedade:
 - *Name* coloque *btnNovo*, na propriedade *Caption* coloque *Novo*;
 - clique no segundo botão e na propriedade *Name* coloque *btnAlterar*, na propriedade *Caption* coloque *Alterar*;
 - clique no terceiro botão e na propriedade *Name* coloque *btnExcluir*, na propriedade *Caption* coloque *Excluir* e por último
 - clique no quarto botão e na propriedade *Name* coloque *btnAtualizar*, na propriedade *Caption* coloque *Atualizar*.
- agora clique na aba de componentes *Data Controls*, escolha o componente *TDBGrid* e cole-o no *pnlGridClientes* do formulário. No OI, altere a propriedade *Align* para *alClient*, para que o componente *TDBGrid* preencha todo o componente *TPanel* do formulário (Figura 8), a propriedade *Name* do *TDBGrid* para *dgrClientes*, a propriedade *Options > dgEditing* para *false* e a propriedade *Options > dgRowSelect* para *true*;
- agora siga os mesmos passos para colocar os componentes no formulário

frmEstadosVisao.

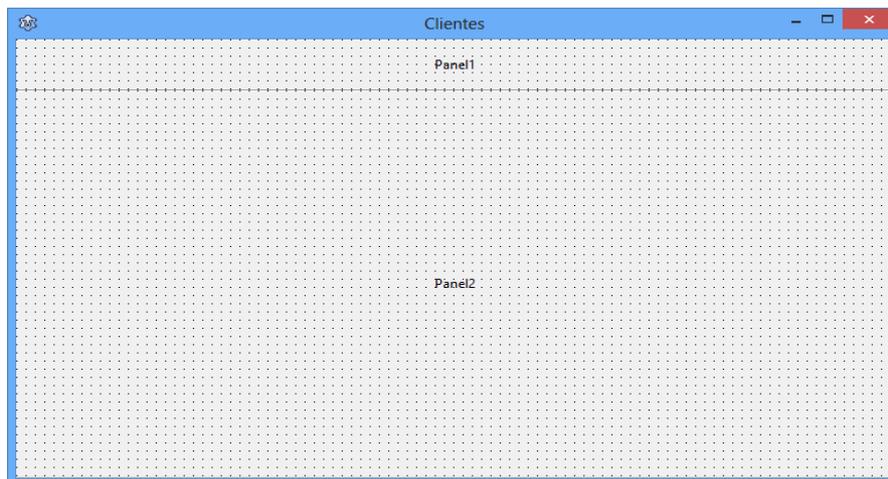


Figura 7: Formulário com TPanel's.

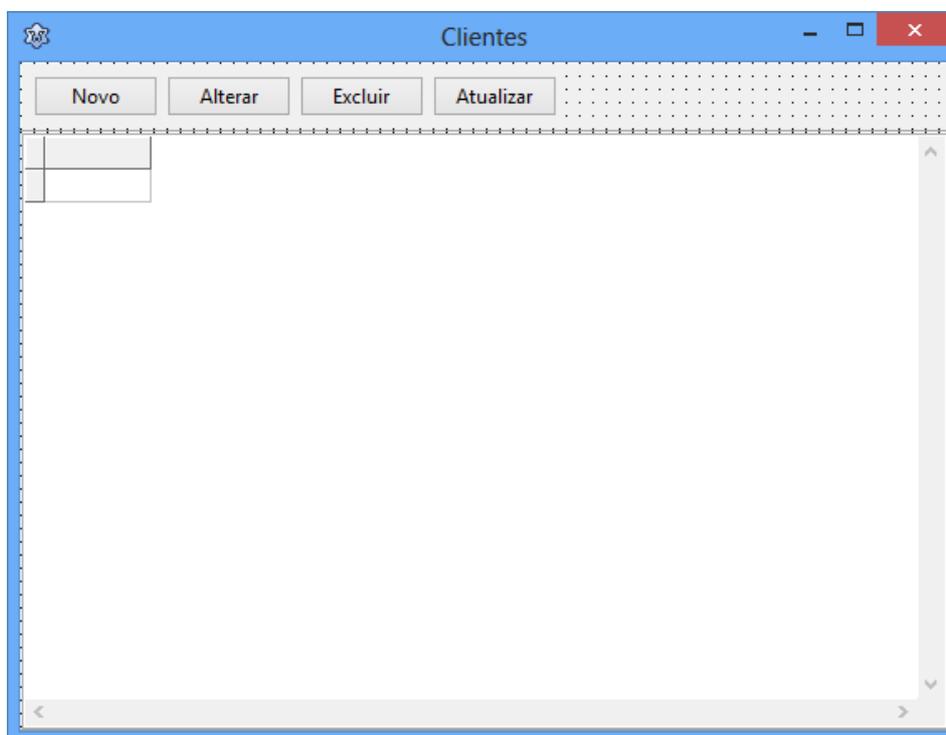


Figura 8: Visualização do formulário frmClientesVisao com os seus respectivos componentes.

Agora, vamos colocar os componentes no formulário *frmClienteEdicao*. Abra o formulário e:

- na aba *Data Controls* escolha o componente *TDBEdit*, cole sete componentes deste tipo

no formulário. Cole também um *TDBLookupComboBox* e, pulando para a aba *Standard*, escolha o componente *TLabel* e ponha um para cada um dos itens que colamos anteriormente. Coloque também dois componentes do tipo *TButton*;

- certifique-se de que seu formulário esteja semelhante ao apresentado na Figura 9 (repare que nesta figura os componentes já estão todos em seus devidos lugares e com suas respectivas propriedades todas preenchidas, vamos então preencher as propriedades dos nossos componentes);
- clique no primeiro *TLabel* e na propriedade *Name* digite *IblCodigo* e na propriedade *Caption* digite *Código*;
- clique no segundo *TLabel* e na propriedade *Name* digite *IblNome* e na propriedade *Caption* digite *Nome*;
- clique no terceiro *TLabel* e na propriedade *Name* digite *IblLogradouro* e na propriedade *Caption* digite *Logradouro*;
- clique no quarto *TLabel* e na propriedade *Name* digite *IblNumero* e na propriedade *Caption* digite *Número*;
- clique no quinto *TLabel* e na propriedade *Name* digite *IblBairro* e na propriedade *Caption* digite *Bairro*;
- clique no sexto *TLabel* e na propriedade *Name* digite *IblMunicipio* e na propriedade *Caption* digite *Município*;
- clique no sétimo *TLabel* e na propriedade *Name* digite *IblCEP* e na propriedade *Caption* digite *C.E.P.*;
- clique no oitavo *TLabel* e na propriedade *Name* digite *IblEstado* e na propriedade *Caption* digite *Estado*;

agora vamos nomear os componentes do tipo *TBEdit* que colocamos no formulário. Então faça o seguinte:

- clique no primeiro componente *TBEdit* (abaixo do rótulo *Código*) e na propriedade *Name* digite *dedCodigo*;
- clique no segundo componente *TBEdit* (abaixo do rótulo *Nome*) e na propriedade *Name* digite *dedNome*;
- clique no terceiro componente *TBEdit* (abaixo do rótulo *Logradouro*) e na propriedade *Name* digite *dedLogradouro*;
- clique no quarto componente *TBEdit* (abaixo do rótulo *Número*) e na propriedade *Name* digite *dedNumero*;
- clique no quinto componente *TBEdit* (abaixo do rótulo *Bairro*) e na propriedade *Name* digite *dedBairro*;
- clique no sexto componente *TBEdit* (abaixo do rótulo *Município*) e na propriedade *Name* digite *dedMunicipio*;
- clique no sétimo componente *TBEdit* (abaixo do rótulo *C.E.P.*) e na propriedade *Name* digite *dedCEP*;
- clique no único *TDBLookupComboBox* e na propriedade *Name* digite *dlcEstado*.

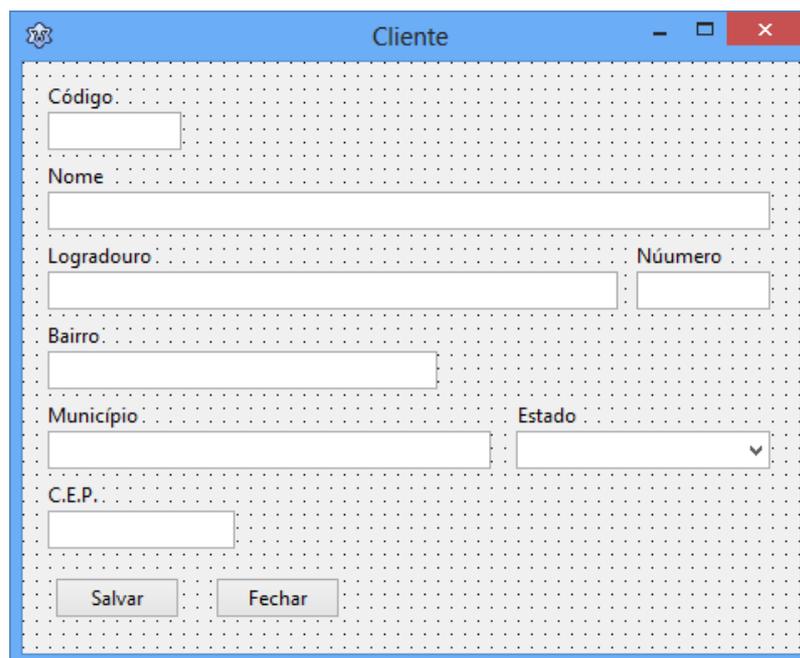
Já foram dados nomes a todos os componentes *TBEdit* e *TLabel*'s existentes no formulário. Agora só faltam dois componentes *TButton*. Então vamos lá:

- clique no primeiro *TButton*. Na propriedade *Name* digite *btnSalvar* e na propriedade *Caption* digite *Salvar* e então
- clique no segundo *TButton* e na propriedade *Name* digite *btnFechar* e na propriedade *Caption* digite *Fechar*.

Configuraremos uma última propriedade neste formulário, portanto, dê um clique numa área vazia do formulário e vá ao *OI*. Na propriedade *ActiveControl* escolha o componente *dedCodigo*. Esta propriedade serve para dizer ao *software* que o componente *dedCodigo* deve receber o foco assim que o formulário *frmClienteEdicao* for carregado na tela.

E agora seu formulário *frmClienteEdicao* deve estar bem semelhante ao apresentado na Figura 9.

Agora use este mesmo processo para colocar os componentes no formulário *frmEstadoEdicao*. Lembre-se que para este formulário nós só temos três campos, que são: o código, o nome e a sigla do estado. Não se esqueça de colocar os botões de *Salvar* e *Fechar*, o rótulo de cada campo e os nomes de todos os componentes seguindo o padrão que estamos utilizando.



The image shows a screenshot of a Windows-style application window titled "Cliente". The window contains a form with the following fields and controls:

- Código:** A text input field.
- Nome:** A wide text input field.
- Logradouro:** A text input field.
- Número:** A text input field.
- Bairro:** A text input field.
- Município:** A text input field.
- Estado:** A dropdown menu.
- C.E.P.:** A text input field.
- Buttons:** Two buttons labeled "Salvar" and "Fechar" at the bottom.

Figura 9: Formulário *frmClienteEdicao*.

Criando o Contêiner de Componentes de Acesso a Dados

A tecnologia empregada no Lazarus é a idêntica à tecnologia empregada no antigo *software* conhecido como Delphi da empresa Borland / Embarcadero.

O Delphi em sua época de ouro, introduziu o conceito de um formulário exclusivo para abrigar os componentes que faziam a interface entre o *software* que está sendo criado e a base de dados que este deve acessar. Este formulário é um componente do tipo *DataModule* e faremos uso desta tecnologia para separar a parte de acesso a banco de dados, da interface e da regra de negócio do *software*.

Portanto, acesse o *menu File | New...* e na tela que se abrirá escolha a opção *Data Module*. Será criado um contêiner como mostrado na Figura 10.

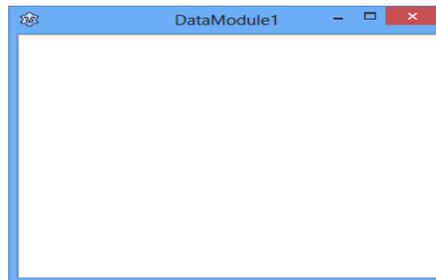


Figura 10: Contêiner "Data Module".

Se você observar bem, quando se está trabalhando com um *Data Module*, são poucos os componentes que aparecem nas barras de componentes e que podem ser utilizados.

Bem com o *Data Module* em foco, vá até o OI e mude a propriedade *Name* para *dmlPrincipal*. As outras duas propriedades existentes devem ficar do jeito que estão.

Por fim, salve o *Data Module* com o nome de *DmdPrincipal*.

Criando a Conexão com o Banco de Dados

Com o *Data Module* em foco, vá até a aba de componentes *SQLdb* e escolha, arraste e cole no *Data Module* o componente do tipo *TPQConnection* (usado para conexão com o SGBD¹ PostgreSQL).

Vamos configurá-lo imediatamente, então com este componente em foco, vá até o OI e:

- na propriedade *Name* digite *pqcExemploMDI*;
- na propriedade *DatabaseName* digite *ExemploMDI*;
- na propriedade *HostName* digite *localhost*;
- na propriedade *Password* digite a senha do SGBD, no meu caso, *123456*;
- na propriedade *UserName* digite o usuário de acesso ao SGBD, no meu caso, *postgres*.

Agora, após preenchidas estas propriedades, podemos testar a conexão. Para isto, basta mudarmos a propriedade *Connected* que está com o valor *false* para o valor *true*. Para basta dar um clique duplo sobre o valor *false*.

Caso a conexão seja feita, então a propriedade permanecerá com o valor *true*. Caso aconteça alguma falha de conexão, então a propriedade voltará ao valor *false* e uma mensagem explicando o erro será apresentada na tela.

1 Sistema Gerenciador de Banco de Dados.

Criando os DataSet's e a Conexão

Ainda na aba de componentes *SQLdb* escolha, arraste e cole um componente *TSQLQuery* no *Data Module*. Vá até a aba de componentes *Data Access* escolha, arraste e cole um componente do tipo *TDataSource* no *Data Module*. O *TDataSource* é o objeto que permite que os componentes de um formulário tenham acesso aos dados de um *DataSet*.

Escolha também um componente do tipo *TSQLTransaction*, cole-o também no *Data Module*. Altere a propriedade *Name* para *trsGeral*. Este é o objeto que controla as transações que são efetuadas no banco de dados. Portanto, dê um clique no componente *pqcExemploMDI*, e na propriedade *Transaction* escolha a opção que você acabou de criar ou seja, *trsGeral*.

Dê um clique simples no componente *TDataSource*, vá ao OI e mude a propriedade *Name* para *dtsEstado*. Na propriedade *DataSet* escolha a única opção disponível, ou seja, o nome do componente *TSQLQuery* que você já deve ter colocado no *Data Module*.

Dê um clique simples no componente *TSQLQuery* e vá até o OI, mude a propriedade *Name* para *qryEstado*, na propriedade *Database* escolha a opção *pqcExemploMDI* (ou seja, o componente *pqcExemploMDI* que acabamos de configurar para acessar o PostgreSQL), na propriedade *Transaction* escolha o *trsGeral* e na propriedade:

- *SQL* dê um clique no botão com reticências que aparece ao se clicar na propriedade. O editor será aberto. Então digite no editor a seguinte sentença SQL:
 - ```
SELECT est_cod, est_nom, est_sgl
 FROM "Estado"
 ORDER BY est_nom
```
- *DeleteSQL* dê um clique no botão com reticências, abra o editor e escreva a sentença SQL:
  - ```
DELETE
      FROM "Estado"
      WHERE est_cod = :est_cod
```
- *InsertSQL* dê um clique no botão com reticências, abra o editor e escreva a sentença SQL:
 - ```
INSERT INTO "Estado"
 VALUES (:est_cod, :est_nom, :est_sgl)
```
- *UpdateSQL* dê um clique no botão com reticências, abra o editor e escreva a sentença SQL:
  - ```
UPDATE "Estado"
      SET est_nom = :est_nom, est_sgl = :est_sgl
      WHERE est_cod = :est_cod
```
- seu *Data Module* deve estar semelhante ao apresentado na Figura 7. Agora vamos, com os mesmos passos acima, criar um *TDataSource* e um *TSQLQuery* para a tabela *Cliente*.

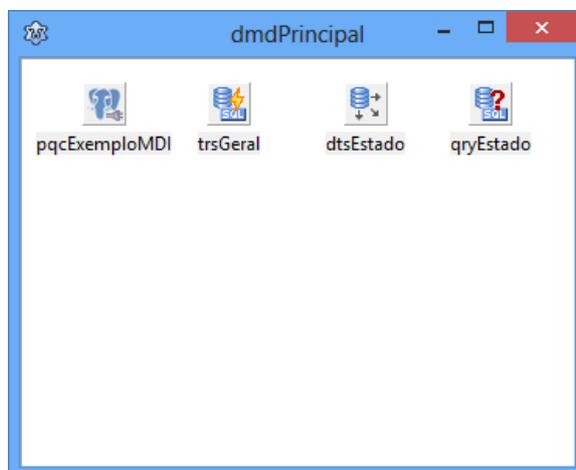


Figura 11: Criação do componentes para acesso ao PostgreSQL e suas tabelas.

As sentenças SQL para o componente *TSQLQuery* da tabela *Cliente* são:

- propriedade *SQL*:
 - ```
SELECT cli_cod, cli_nom, cli_lgd, cli_num,
 cli_bai, cli_mun, cli_cep, est_cod
FROM "Cliente"
ORDER BY cli_nom
```
- propriedade *DeleteSQL*:
  - ```
DELETE  
FROM "Cliente"  
WHERE cli_cod = :cli_cod
```
- propriedade *InsertSQL*:
 - ```
INSERT INTO "Cliente"
VALUES (:cli_cod, :cli_nom, :cli_lgd, :cli_num,
 :cli_bai, :cli_mun, :cli_cep, :est_cod)
```
- propriedade *UpdateSQL*:
  - ```
UPDATE "Cliente"  
SET cli_nom = :cli_nom,  
    cli_lgd = :cli_lgd,  
    cli_num = :cli_num,  
    cli_bai = :cli_bai,  
    cli_mun = :cli_mun,  
    cli_cep = :cli_cep,  
    est_cod = :est_cod
```

```
WHERE cli_cod = :cli_cod
```

- por fim, seu *Data Module* deve estar semelhante ao apresentado na Figura 12.

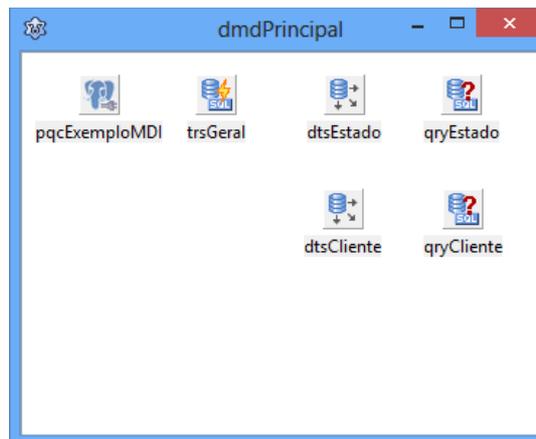


Figura 12: *Data Module Principal* com seus componentes.

Ligando os Componentes aos Respectivos DataSources

Façamos agora a ligação dos componentes dos formulários aos seus respectivos *Data Sources*.

Antes de conectarmos os componentes aos seus respectivos *DataSources* temos que primeiramente ir ao *dmlPrincipal* e ativar a conexão com o banco de dados. Para isto faça o seguinte:

- clique no componente *pgcExemploMDI* e dê um clique duplo na propriedade *Connected* passando-a de *false* para *true*;
- agora, vá ao componente *qryEstado* e na propriedade *Active* dê um clique duplo, passando-a de *false* para *true* e por último
- vá ao componente *qryCliente* e faça o mesmo, ou seja, dê um clique duplo na propriedade *Active*, passando-a de *false* para *true*.

Depois desta etapa completada, sabemos que nossas conexões estão corretas e já podemos ligar os componentes dos formulários aos seus respectivos campos nas tabelas do banco de dados.

Abra o formulário *frmClientesVisao* e dê um clique simples no componente *grid dgrClientes*, vá ao OI e na propriedade *DataSource* escolha *dmlPrincipal.dtsCliente*.

Agora faça a mesma coisa com o formulário *frmEstadosVisao*, porém escolha *dmlPrincipal.dtsEstado*.

Abra o formulário *frmClienteEdicao* e dê um clique simples no primeiro componente *TDBEdit*, vá ao OI e na propriedade *DataSource* escolha *dmlPrincipal.dtsCliente*, em seguida vá a propriedade *DataField* e escolha o campo que este *TDBEdit* representa na tabela, ou seja, neste caso o campo *cli_cod*. Faça isto com todos os componentes *TDBEdit* da tela.

Quando chegar a vez do componente *TDBLookupComboBox*, então faremos o seguinte:

- na propriedade *DataSource* escolheremos *dmlPrincipal.dtsCliente*;
- na propriedade *DataField* escolheremos *est_cod*;
- na propriedade *ListSource* escolheremos *dmlPrincipal.dtsEstado*;
- na propriedade *ListField* escolheremos *est_nom* e por último
- na propriedade *KeyField* escolheremos *est_cod*.

Agora façamos o mesmo com o formulário *frmEstadoEdicao*.

Observação: só será possível a escolha dos campos na propriedade *DataField* se as *queries* estiverem ativadas (propriedade *Active = true*).

Preenchendo os Eventos dos Formulários e Componentes

Voltando ao formulário *Principal*, dê um clique simples sobre o *menu Cadastro*. Aparecerá o *submenu Clientes e Estados*. Clique em *Clientes*, será aberto o evento *OnClick* deste componente. Podemos até colocar a chamada para o formulário *frmClientesVisao* (Figura 13), porém esta chamada gerará um erro de compilação, pois, o formulário *frmPrincipal* não conhece o *frmClientesVisao*.

Para que este erro não aconteça devemos adicionar a *unit* do formulário *frmClientesVisao* na *unit* do *frmPrincipal*. E a maneira mais simples de se fazer isto é pressionando conjuntamente as teclas: "Alt + F11", onde abrir-se-á uma janela (Figura 14) para a escolha do formulário que será adicionado e este será incluso na cláusula *uses* no início da *unit*.

Aproveite para adicionar também, ao *frmPrincipal*, as *units* do *frmSobre*, do *frmEstadosVisao* e do *dmlPrincipal*.

Feito isto, nos eventos que foram criados para *Clientes* e *Estados*, copie os respectivos códigos a seguir:

```
procedure TfrmPrincipal.mnuClientesClick(Sender: TObject);
begin
    if (frmClientesVisao = nil) then
        frmClientesVisao := TfrmClientesVisao.create(Self);
    frmClientesVisao.Show;
end;
```

```
procedure TfrmPrincipal.mnuEstadosClick(Sender: TObject);
begin
```

```
if(frmEstadosVisao = nil)then  
    frmEstadosVisao := TfrmEstadosVisao.create(Self);  
frmEstadosVisao.Show;  
end;
```

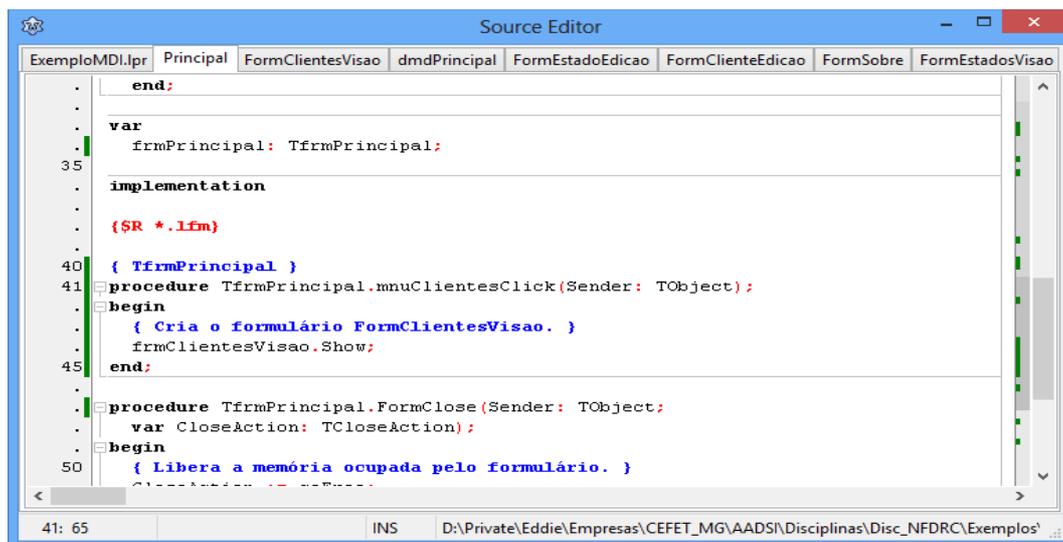


Figura 13: Criação do evento *mnuClientesClick*.

Agora faça o mesmo para o formulário *frmEstadosVisao*.

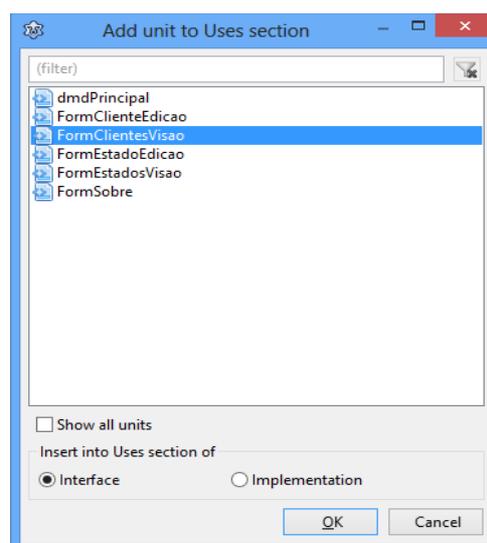


Figura 14: Janela de adição de units.

Para o formulário *frmSobre* não chamaremos o método *Show* e sim o método *ShowModal*, sendo que o evento ficará assim:

```
procedure TfrmPrincipal.mnuSobreClick(Sender: TObject);
begin
    if (frmSobre = nil) then
        frmSobre := TfrmSobre.create(Self);
    frmSobre.ShowModal;
end;
```

Dê um clique no *menu* Cliente

Agora dê um clique numa área vazia do formulário *frmPrincipal*. Vá ao OI na aba *Events* e procure o evento *OnShow* e dê um clique duplo na área a frente do nome do evento. O editor da IDE se abrirá já com o cursor piscando no evento *OnShow* do formulário, então escreva as linhas de código mostradas abaixo. Estas linhas abrirão a conexão com o banco de dados e abrirão também as tabelas *Estado* e *Cliente*.

```
procedure TfrmPrincipal.FormShow(Sender: TObject);
begin
    { Abre a conexão com o banco de dados. }
    dmlPrincipal.pqcExemploMDI.Open;
    { Abrem as tabelas para consulta e alteração. }
    dmlPrincipal.qryEstado.Open;
    dmlPrincipal.qryCliente.Open;
end;
```

Agora complemente o evento *OnClose* do formulário *frmPrincipal* de modo que o mesmo apresente as seguintes linhas de código:

```
procedure TfrmPrincipal.FormClose(Sender: TObject;
    var CloseAction: TCloseAction);
begin
    { Fecham as tabelas para consulta e alteração. }
    dmlPrincipal.qryCliente.Close;
    dmlPrincipal.qryEstado.Close;
    { Fecha a conexão com o banco de dados. }
    dmlPrincipal.pqcExemploMDI.Close;
```

```
{ Libera a memória ocupada pelo formulário. }  
CloseAction := caFree;  
frmPrincipal := nil;  
end;
```

Passemos agora para o formulário *frmClientesVisao*. Onde primeiramente preencheremos o evento *OnClose* da seguinte forma:

```
procedure TfrmClientesVisao.FormClose(Sender: TObject;  
  var CloseAction: TCloseAction);  
begin  
  { Libera a memória ocupada pelo formulário. }  
  CloseAction := caFree;  
  frmClientesVisao := nil;  
end;
```

Vamos preencher o evento *OnClose* de todos os outros formulários da mesma forma, com exceção dos formulários: *frmClienteEdicao* e *frmEstadoEdicao* que não devem ter este evento preenchido.

Em seguida abra o formulário *frmClientesVisao* e dê um clique duplo no botão *btnNovo*. O evento do botão se abrirá. Antes de preencher o evento você deve adicionar os formulários que serão referenciados na cláusula *uses*; para tanto, aperte a combinação de teclas *Alt + F11* e escolha o formulário *dmdPrincipal* e *frmClienteEdicao*. Feito isto, então volte ao evento criado *btnNovoClick* e coloque o seguinte código nele:

```
procedure TfrmClientesVisao.btnNovoClick(Sender: TObject);  
begin  
  dmlPrincipal.qryCliente.Append;  
  frmClienteEdicao.ShowModal;  
end;
```

Agora, neste mesmo formulário dê um clique duplo no botão *Alterar*. E então digite o seguinte código no evento que será aberto:

```
procedure TfrmClientesVisao.btnAlterarClick(Sender: TObject);  
begin  
  frmClienteEdicao.ShowModal;
```

end;

Agora dê um clique duplo no botão *Excluir* e digite o seguinte código:

```
procedure TfrmClientesVisao.btnExcluirClick(Sender: TObject);
begin
  if(dgrClientes.SelectedIndex > 0) then
  begin
    if(MessageDlg('Questão', 'Confirma exclusão do registro?', mtConfirmation,
      [mbYes, mbNo, mbIgnore], 0) = mrYes) then
    begin
      dmlPrincipal.qryCliente.Delete;
    end
    else
      ShowMessage('Registro não excluído!');
  end
  else
    ShowMessage('Antes de tentar excluir selecione um registro.');
```

end;

Por último dê um clique duplo no botão *Atualizar* e coloque o código a seguir no evento que se abrirá:

```
procedure TfrmClientesVisao.btnAtualizarClick(Sender: TObject);
begin
  dmlPrincipal.qryCliente.Close;
  dmlPrincipal.qryCliente.Open;
end;
```

Finalizado os eventos do *frmClientesVisao*, passaremos agora para o *frmEstadosVisao*. Já na *unit* do formulário *frmEstadosVisao*, pressione *Alt + F11* e adicione os formulários: *dmdPrincipal* e *frmEstadoEdicao*. Feito isto, então dê um clique duplo no botão *btnNovo* e escreva o seguinte código no evento deste botão:

```
procedure TfrmEstadosVisao.btnNovoClick(Sender: TObject);
begin
  dmlPrincipal.qryEstado.Append;
```



```
frmEstadoEdicao.ShowModal;  
end;
```

Agora dê um clique duplo no botão *Alterar* deste formulário e preencha o evento com:

```
procedure TfrmEstadosVisao.btnAlterarClick(Sender: TObject);  
begin  
    dmlPrincipal.qryEstado.Edit;  
    frmEstadoEdicao.ShowModal;  
end;
```

Clique duplo no botão *Excluir* e preencha o evento com:

```
procedure TfrmEstadosVisao.btnExcluirClick(Sender: TObject);  
begin  
    if (dgrEstados.SelectedIndex >= 0) then  
        begin  
            if (MessageDlg('Questão', 'Confirma exclusão do registro?', mtConfirmation,  
                [mbYes, mbNo], 0) = mrYes) then  
                begin  
                    dmlPrincipal.qryEstado.Delete;  
                end  
            else  
                ShowMessage('Registro não excluído!');  
            end  
        else  
            ShowMessage('Antes de tentar excluir selecione um registro.');
```

```
end;
```

E por último, clique duplo no botão *Atualizar* e o seguinte código no evento:

```
procedure TfrmEstadosVisao.btnAtualizarClick(Sender: TObject);  
begin  
    dmlPrincipal.qryEstado.Close;  
    dmlPrincipal.qryEstado.Open;
```



end;

Agora abra o formulário *frmClienteEdicao*. Adicione o formulário (*Alt + F11*): *dmdPrincipal*. Dê um clique duplo no botão *Salvar* e digite o seguinte código:

```
procedure TfrmClienteEdicao.btnSalvarClick(Sender: TObject);  
begin  
    dmlPrincipal.qryCliente.Post;  
    ShowMessage('Dados salvos com sucesso!');  
    Close;  
end;
```

Agora dê um clique duplo no botão *Fechar* e digite o código a seguir:

```
procedure TfrmClienteEdicao.btnFecharClick(Sender: TObject);  
begin  
    dmlPrincipal.qryCliente.Cancel;  
    Close;  
end;
```

Agora, abra o *frmEstadoEdicao*. Adicione o formulário (*Alt + F11*): *dmdPrincipal*. Dê um duplo clique no botão *Salvar*. Em seguida, digite o código abaixo:

```
procedure TfrmEstadoEdicao.btnSalvarClick(Sender: TObject);  
begin  
    dmlPrincipal.qryEstado.Post;  
    ShowMessage('Dados salvos com sucesso!');  
    Close;  
end;
```

Agora, dê um clique duplo no botão *Fechar* e digite o código a seguir:

```
procedure TfrmEstadoEdicao.btnFecharClick(Sender: TObject);  
begin  
    dmlPrincipal.qryEstado.Cancel;  
    Close;
```

end;

Abra o *Data Module dmlPrincipal* e clique no componente *qryEstado*. Vá ao OI, na aba *Events* e dê um clique duplo na área a frente do evento *AfterCancel*, *AfterDelete* e *AfterPost* (um de cada vez). Coloque o seguinte código nos três eventos distintos:

```
procedure TdmlPrincipal.qryEstadoAfterCancel(DataSet: TDataSet);  
begin  
    TSQLQuery(DataSet).CancelUpdates;  
end;
```

```
procedure TdmlPrincipal.qryEstadoAfterDelete(DataSet: TDataSet);  
begin  
    TSQLQuery(DataSet).ApplyUpdates;  
    trsGeral.CommitRetaining;  
end;
```

```
procedure TdmlPrincipal.qryEstadoAfterPost(DataSet: TDataSet);  
begin  
    TSQLQuery(DataSet).ApplyUpdates;  
    trsGeral.CommitRetaining;  
end;
```

Agora faça a mesma coisa com o componente *qryCliente*. Caso você se esqueça de preencher estes eventos com estes comandos, nada que for feito no *software* será concretizado no banco de dados.

Moldando a Apresentação de Dados nos DataGrid's

Atualmente, do jeito que está, quando abrimos os formulários de visão (tanto de Estados quanto de Clientes), o título de cada coluna mostrada no *DataGrid* é exatamente o nome da coluna no banco de dados. Isto não é legal, pois, os nomes das colunas nos bancos de dados são abreviados ou são siglas ou ainda podem ser mnemônicos (como é o nosso caso).

Assim sendo, vamos agora mudar a realidade de nosso *software*.

Abra o *Data Module dmlPrincipal* e ative o componente *pgcExemploMDI* e ambas as *queries*.

Dê um clique duplo encima do componente *qryEstado* e a seguinte tela aparecerá

(Figura 15). Nesta tela clique no botão com o sinal de “+”, então a tela mostrada na Figura 16 aparecerá. Nela escolha os campos desejados, para este exemplo foram escolhidos todos os campos, pois, desejamos formatar todos. Então, clique no botão.

Após o clique no botão *Create*, os campos aparecerão na tela mostrada na Figura 15. Então, é só clicar num deles e fazer toda formatação desejada no OI. As principais propriedades a serem formatadas são:

- *DisplayLabel* (aqui é o título da coluna) e
- *DisplayWidth* (aqui é a largura da coluna).

Exemplos:

- para o campo *est_cod* do componente *qryEstado*, no OI, você preencherá a propriedade *DisplayLabel* com o texto “Código” (sem as aspas), pois, o campo *est_cod* representa o código do estado. A propriedade *DisplayWidth* para este campo em especial, você poderá deixar com o valor padrão;
- para o campo *est_nom* do componente *qryEstado*, no OI, você preencherá a propriedade *DisplayLabel* com o texto “Nome” (sem as aspas), pois, o campo *est_nom* representa o nome do estado. A propriedade *DisplayWidth* para este campo, você poderá colocar o tamanho 30 e
- para o campo *est_sgl* do componente *qryEstado*, no OI, você preencherá a propriedade *DisplayLabel* com o texto “Sigla” (sem as aspas), pois, o campo *est_sgl* representa a sigla do estado. A propriedade *DisplayWidth* para este campo, você poderá colocar o tamanho 5, pois, apesar das siglas estaduais terem apenas duas letras, o tamanho 5 é para que a coluna não fique muito pequena.

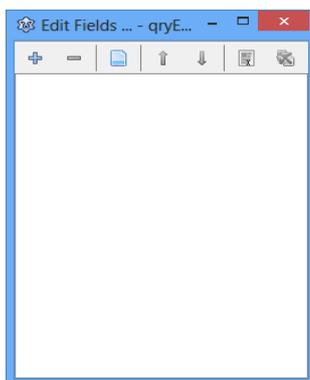


Figura 15: Tela de edição de campos



Figura 16: Tela para escolha dos campos.

Defina as propriedades *DisplayLabel* e *DisplayWidth* de acordo com o que você julgar melhor. Infelizmente não há como ver as modificações automaticamente, ou seja, para vê-las você terá que abrir as *queries* ou executar o programa. Faça as modificações para o componente *qryCliente*.



E pronto! Se você fez tudinho certo até aqui, seu sistema já deve estar totalmente pronto e operante. Teste-o para confirmar esta hipótese.