

Sistema para Controle Financeiro com o Framework

PROF. ESP. EZEQUIEL JULIANO MÜLLER

HORUS FACULDADES

CURSO DE SISTEMAS DE INFORMAÇÃO

PINHALZINHO - SC

Problematização

O software deverá ser desenvolvido para atender a seguinte situação:

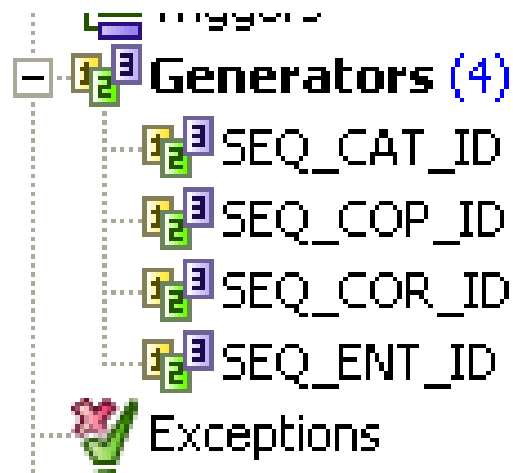
Deseja-se um software para controle de finanças pessoais, onde seja possível se estabelecer um orçamento baseado nas receitas e despesas que estão por vir como salários, aluguéis, contas de água, luz, escola, mercado, transporte, impostos. Estas receitas e despesas devem ter um valor, uma data de vencimento, uma descrição, uma categoria, além das informações de pagamentos (data, valor, juro, desconto). A partir destas informações deseja-se saber quanto dinheiro se tem e quanto vai se receber e gastar no futuro.

Problematização

Analisando em itens o sistema deve ser capaz de gerenciar:

- Cadastrar Entidades (Clientes e Fornecedores);
- Cadastrar Categorias;
- Cadastrar Contas a Pagar;
- Cadastrar Contas a Receber;

Modelo ER



ENTIDADE		
<u>ENT_ID</u>	INTEGER	<pk>
ENT_CLIENTE	SMALLINT	
ENT_FORNECEDOR	SMALLINT	
ENT_TIPO_PESSOA	CHAR(1)	
ENT_NOME	VARCHAR(60)	
ENT_CPFCNPJ	VARCHAR(18)	
ENT_ENDERECO	VARCHAR(100)	
ENT_CEP	VARCHAR(10)	
ENT_CIDADE	VARCHAR(60)	
ENT_UF	CHAR(2)	

CONTA_PAGAR		
<u>COP_ID</u>	INTEGER	<pk>
ENT_ID	INTEGER	<fk1>
CAT_ID	INTEGER	<fk2>
COP_VALOR	DECIMAL(15,2)	
COP_VENCIMENTO	DATE	
COP_DESCRICAO	VARCHAR(60)	
COP_PGTO_DATA	DATE	
COP_PGTO_VALOR	DECIMAL(15,2)	
COP_PGTO_JURO	DECIMAL(15,2)	
COP_PGTO_DESC	DECIMAL(15,2)	
COP_TOTAL	DECIMAL(15,2)	

CATEGORIA		
<u>CAT_ID</u>	INTEGER	<pk>
CAT_DESCRICAO	VARCHAR(60)	
CAT_TIPO	CHAR(1)	

CONTA_RECEBER		
<u>COR_ID</u>	INTEGER	<pk>
ENT_ID	INTEGER	<fk1>
CAT_ID	INTEGER	<fk2>
COR_VALOR	DECIMAL(15,2)	
COR_VENCIMENTO	DATE	
COR_DESCRICAO	VARCHAR(60)	
COR_TOTAL	DECIMAL(15,2)	
COR_PGTO_DATA	DATE	
COR_PGTO_VALOR	DECIMAL(15,2)	
COR_PGTO_JURO	DECIMAL(15,2)	

Entidade

O sistema deve possuir uma tela para cadastrar entidades.

O campo **ENT_ID** deve ser gerado automaticamente pelo sistema (**Generator SEQ_ENT_ID**), deve ser mostrado em tela e não pode sofrer alterações pelo usuário.

O campo **ENT_CLIENTE** deve receber os valores: 0 – Se não for cliente, 1 – Se for cliente. Use um DBCheckBox.

O campo **ENT_FORNECEDOR** deve receber os valores: 0 – Se não for fornecedor, 1 – Se for fornecedor. Use um DBCheckBox.

O campo **ENT_TIPO_PESSOA** deve receber **F** para pessoa física e **J** para pessoa jurídica. Use um DBComboBox.

Categoria

O sistema deve possuir uma tela para cadastrar categorias.

O campo **CAT_ID** deve ser gerado automaticamente pelo sistema (**Generator SEQ_CAT_ID**), deve ser mostrado em tela e não pode sofrer alterações pelo usuário.

O campo **CAT_TIPO** deve receber os valores: **R – Receita** e **D – Despesa**. Use um DBComboBox.

Contas a Pagar

O sistema deve possuir uma tela para cadastrar contas a pagar.

O campo **COP_ID** deve ser gerado automaticamente pelo sistema (**Generator SEQ_COP_ID**), deve ser mostrado em tela e não pode sofrer alterações pelo usuário.

A consulta de Entidade e Categoria deve ser feito usando a tela padrão de pesquisa (ver últimos slides que tem um exemplo).

O campo COP_TOTAL deve receber o valor do seguinte cálculo:

Valor Total := Valor Pago + Valor de Juro – Valor de Desconto.

Contas a Receber

O sistema deve possuir uma tela para cadastrar contas a receber.

O campo **COR_ID** deve ser gerado automaticamente pelo sistema (**Generator SEQ_COR_ID**), deve ser mostrado em tela e não pode sofrer alterações pelo usuário.

A consulta de Entidade e Categoria deve ser feito usando a tela padrão de pesquisa (ver últimos slides que tem um exemplo).

O campo COR_TOTAL deve receber o valor do seguinte cálculo:

Valor Total := Valor Pago + Valor de Juro – Valor de Desconto.

Estrutura de Pastas

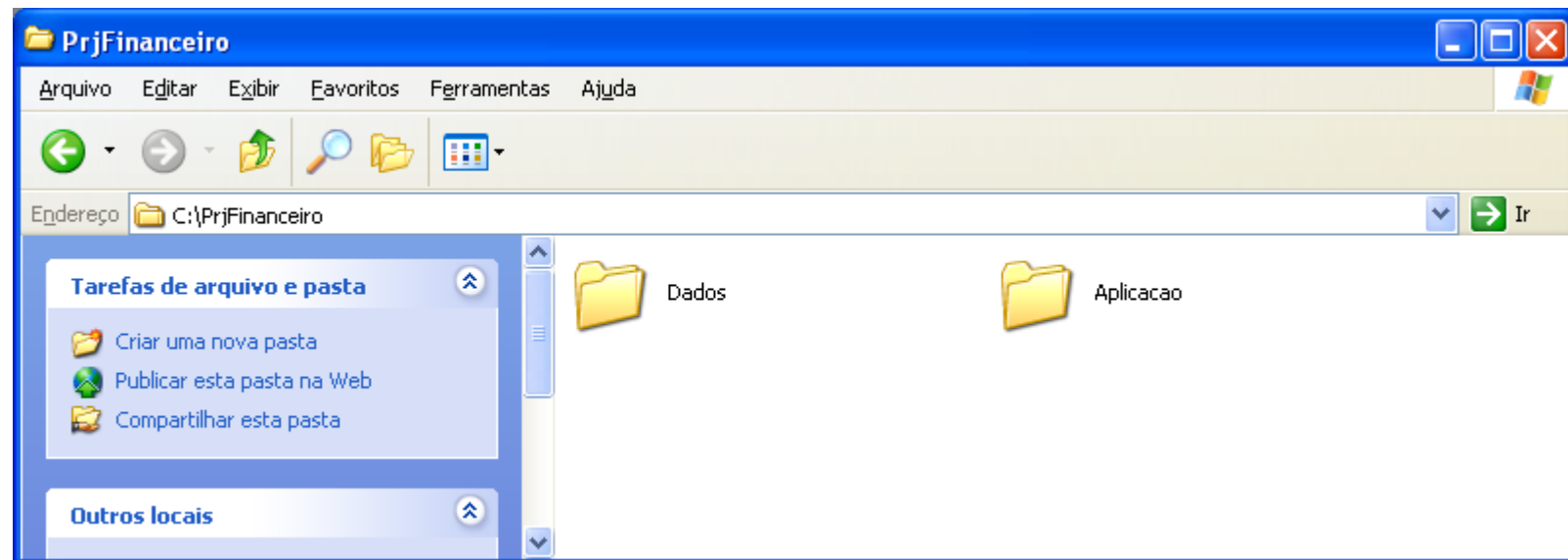
Na sua unidade **C:** crie a seguinte estrutura de pastas que vai servir para o desenvolvimento do projeto:

C:\PrjFinanceiro

C:\PrjFinanceiro\Dados

C:\PrjFinanceiro\Aplicacao

Dentro da pasta “**Dados**” você deve colocar o arquivo de banco de dados do Firebird denominado **FINANCEIRO.FDB** que se encontra no portal de ensino. E na pasta “**Aplicacao**” você deve salvar os arquivos fontes e de projeto da software.



Criando a Aplicação

No Delphi crie uma nova aplicação **File > New > VCL Forms Application – Delphi**.

Agora vamos salvar o projeto clique em **Save All**.

Salve da seguinte forma:

- **PrjFinanceiro.dproj** (salvar em C:\PrjFinanceiro\Aplicacao).
- Form Principal – Unit: **Finan.Frm.Principal.pas** | Form: **FrmPrincipal** (salvar em C:\PrjFinanceiro\Aplicacao).

Configurando o Projeto

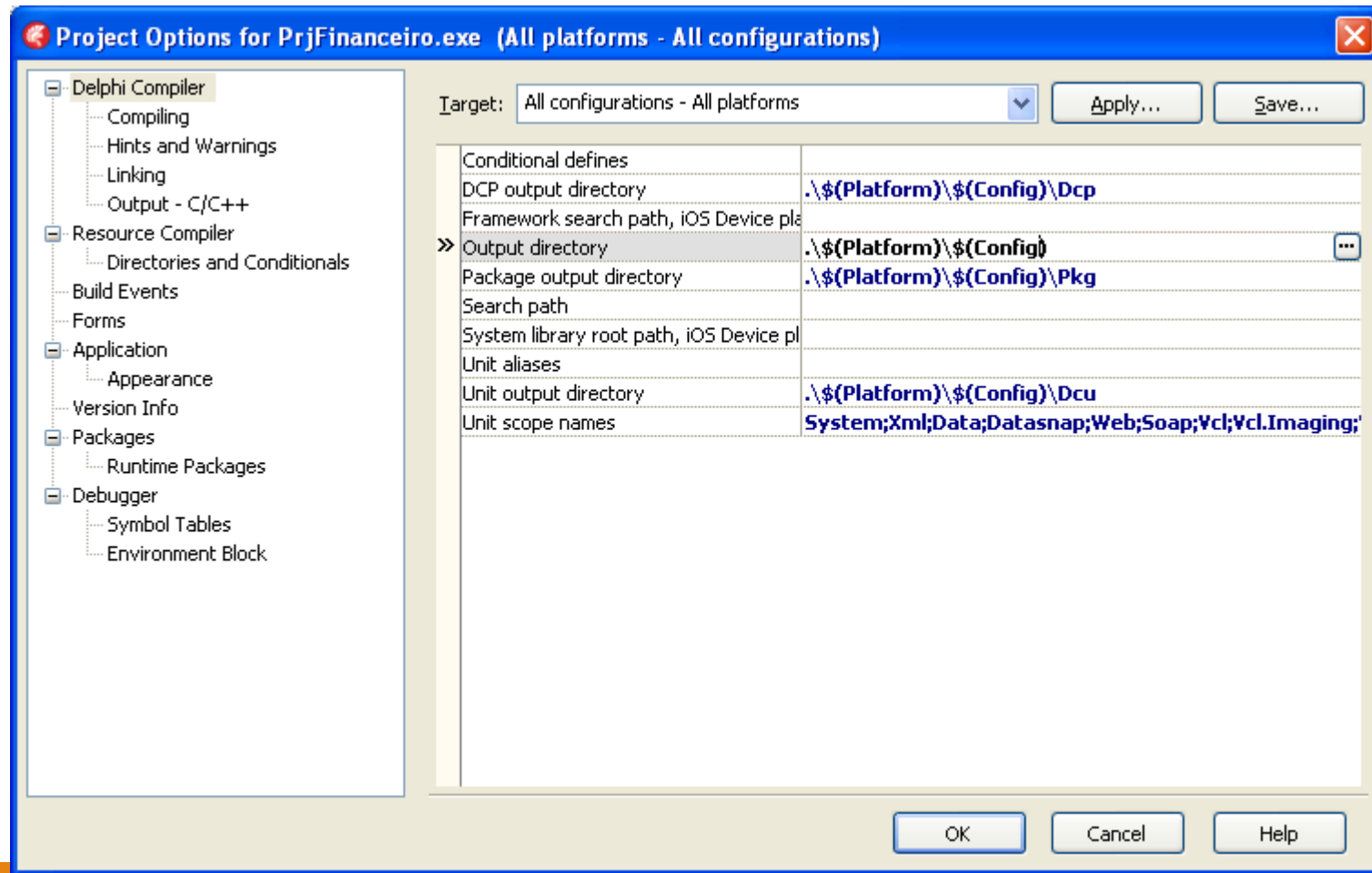
Agora no menu do Delphi vá em **Project > Options**. Clique em **Delphi Compiler** e no **Target** selecione **All Configurations**.

DCP output directory:
.\\$(Platform)\\$(Config)\Dcp

Output directory:
.\\$(Platform)\\$(Config)

Package output directory:
.\\$(Platform)\\$(Config)\Pkg

Unit output directory:
.\\$(Platform)\\$(Config)\Dcu



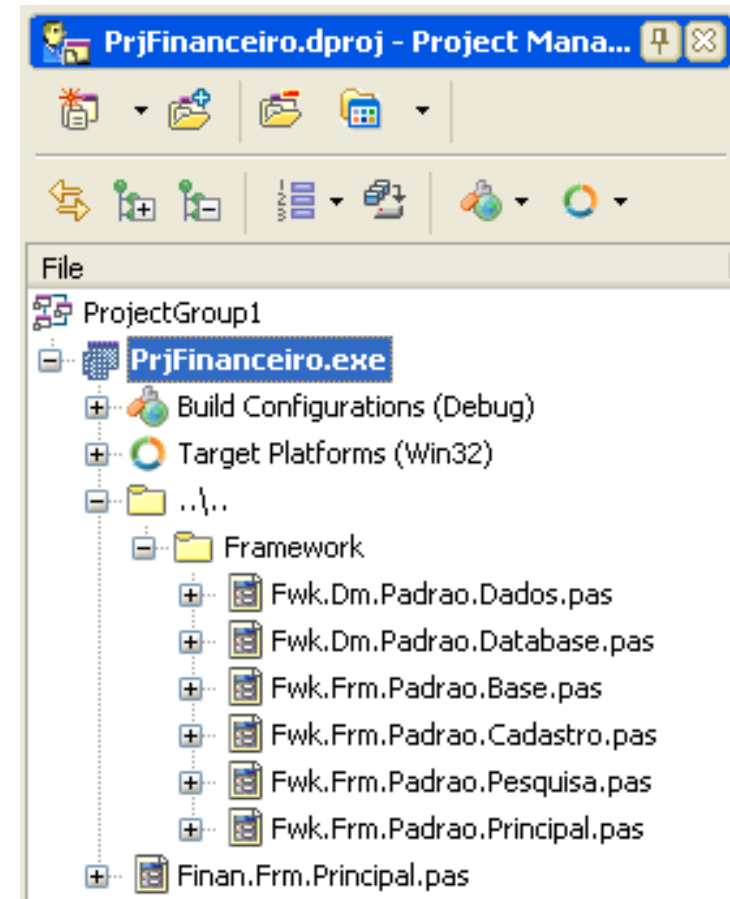
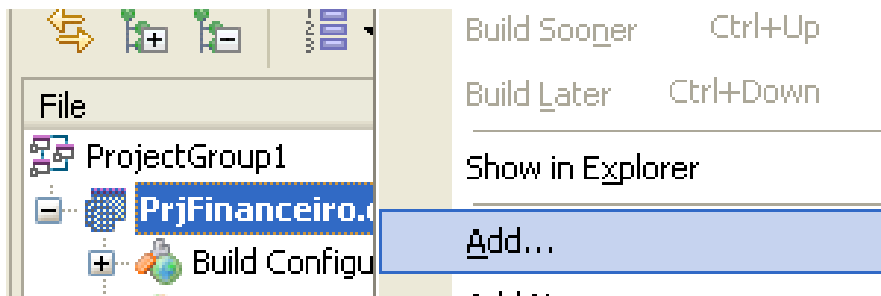
Adicionando Framework

Agora é necessário adicionar as **Units** do Framework em nosso projeto.

Para isso clique com o botão direito sobre o projeto e selecione a opção **Add...**

Navegue até a pasta **C:\Framework**

Selecione todos os arquivos do Framework e de um OK.



Configurando o FrmPrincipal

Agora vamos configurar nosso Formulário Principal.

Adicionar novo Form com os seguintes componentes:

TlbPrincipal: TToolBar;

StbPrincipal: TStatusBar;

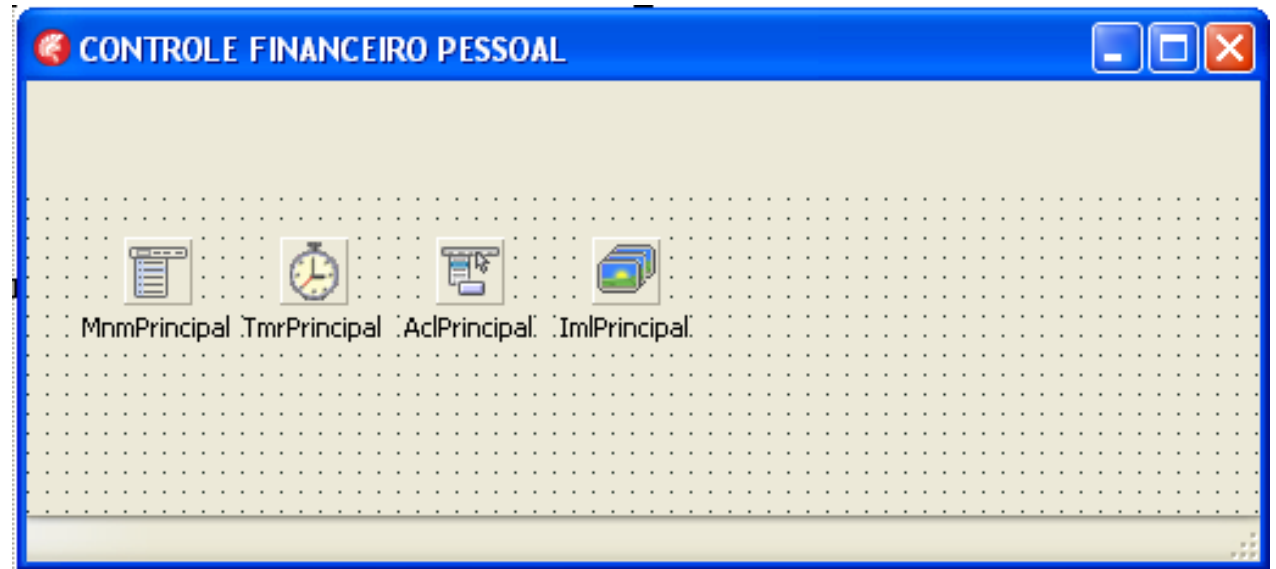
MnmPrincipal: TMainMenu;

TmrPrincipal: TTimer;

ActPrincipal: TActionList;

ImlPrincipal: TImageList;

Propriedades: Position = poDesktopCenter; ShowHint = True; WindowsState = wsMaximized;
Menu = MnmPrincipal;



Configurando o FrmPrincipal

Adicione um novo painel ao StatusBar (**StbPrincipal**).

Depois no evento **OnTimer** do **TTimer (TmrPrincipal)** adicione o seguinte código que mostra a data e hora atual na barra de status de nosso sistema.

```
procedure TFrmPrincipal.TmrPrincipalTimer(Sender: TObject);  
begin  
    StbPrincipal.Panels[0].Text := 'Data e Hora Atual: ' +  
        FormatDateTime('C', Now);  
end;
```

Criando Conexão com o Banco de Dados

Agora vamos herdar o **FwkDmPadraoDatabase** do que foi criado no Framework.

Para isso no Delphi vá em **File > New > Other...**

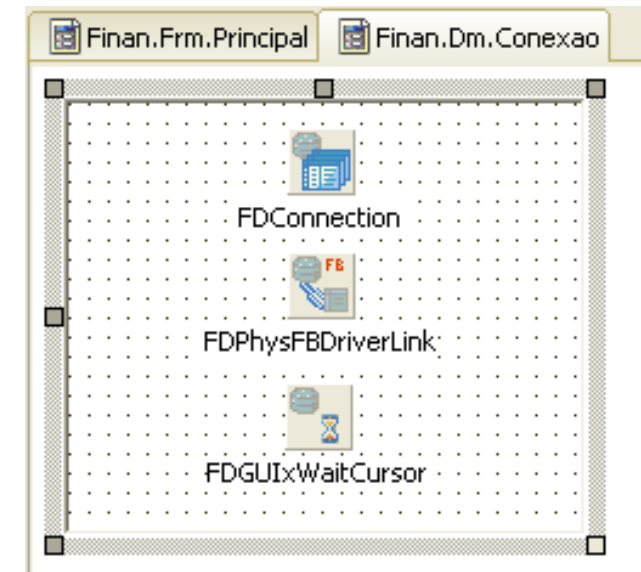
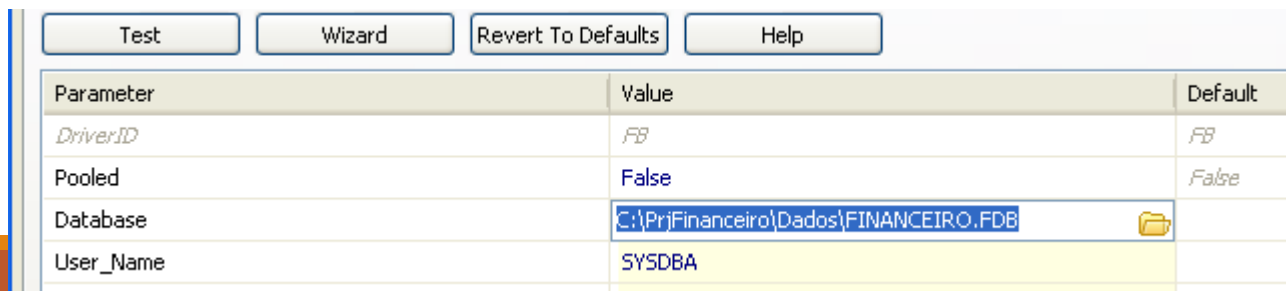
Selecione a opção **Inheritable Items** e selecione o **FwkDmPadraoDatabase**.

Salve o novo Data Module da seguinte forma:

Name = DmConexao

Unit = Finan.Dm.Conexao.pas em C:\PrjFinanceiro\Aplicacao.

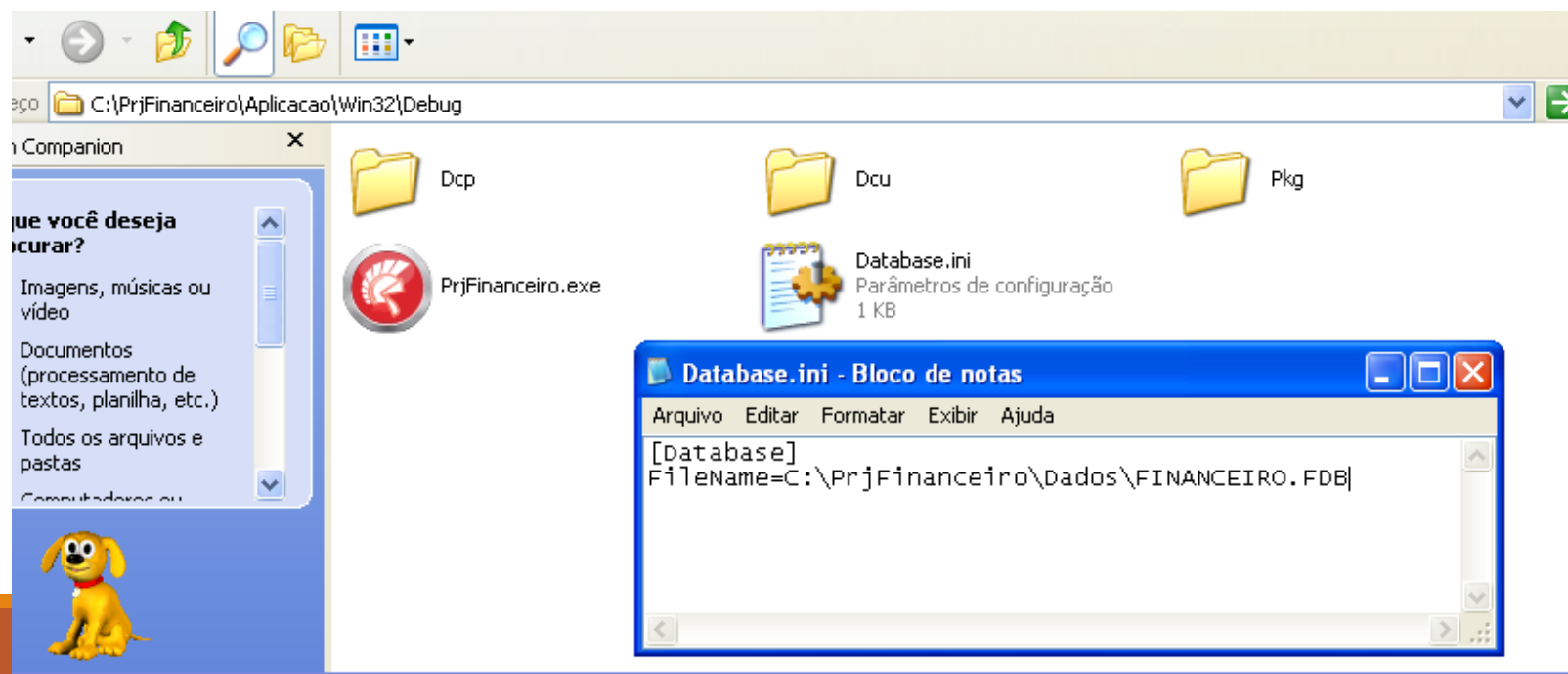
De um duplo clique sobre o componente de conexão e configure o arquivo do banco de dados do Firebird. Somente isso é necessário.



Criando Arquivo de Configuração da Conexão do Banco de Dados

Dentro da pasta onde é gerado o executável, conforme a configuração deve ter ficado em **C:\PrjFinanceiro\Aplicacao\Win32\Debug** você deve criar o arquivo **.ini** que vai armazenar a configuração de conexão com o banco de dados.

Por isso crie um novo arquivo denominado **Database.ini** e dentro dele especifique as informações conforme a imagem.



Criando Data Module para Cadastros

Agora vamos herdar o **FwkDmPadraoDados** do que foi criado no Framework.

Para isso no Delphi vá em **File > New > Other...**

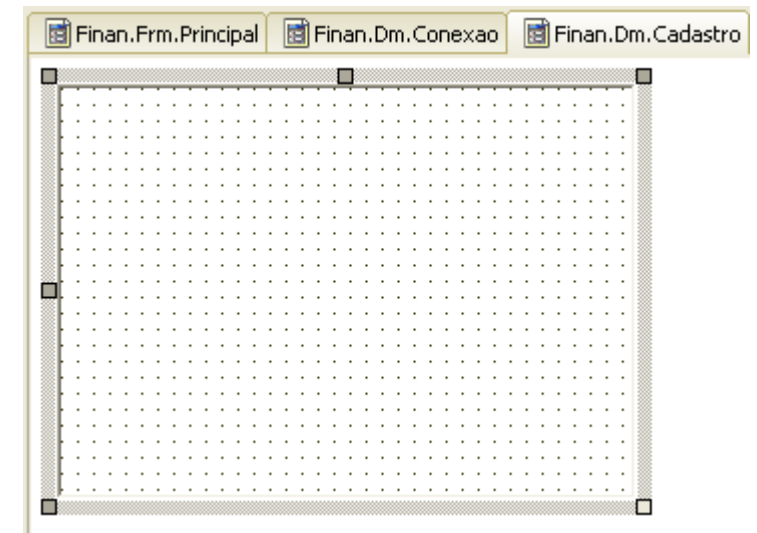
Selecione a opção **Inheritable Items** e selecione o **FwkDmPadraoDados**.

De Uses do **DmConexao**.

Salve o novo Data Module da seguinte forma:

Name = DmCadastro

Unit = Finan.Dm.Cadastro.pas em C:\PrjFinanceiro\Aplicacao.



Configurando Auto-Create Forms

No Delphi vá em **Project > Options**.

Selecione a opção **Forms**.

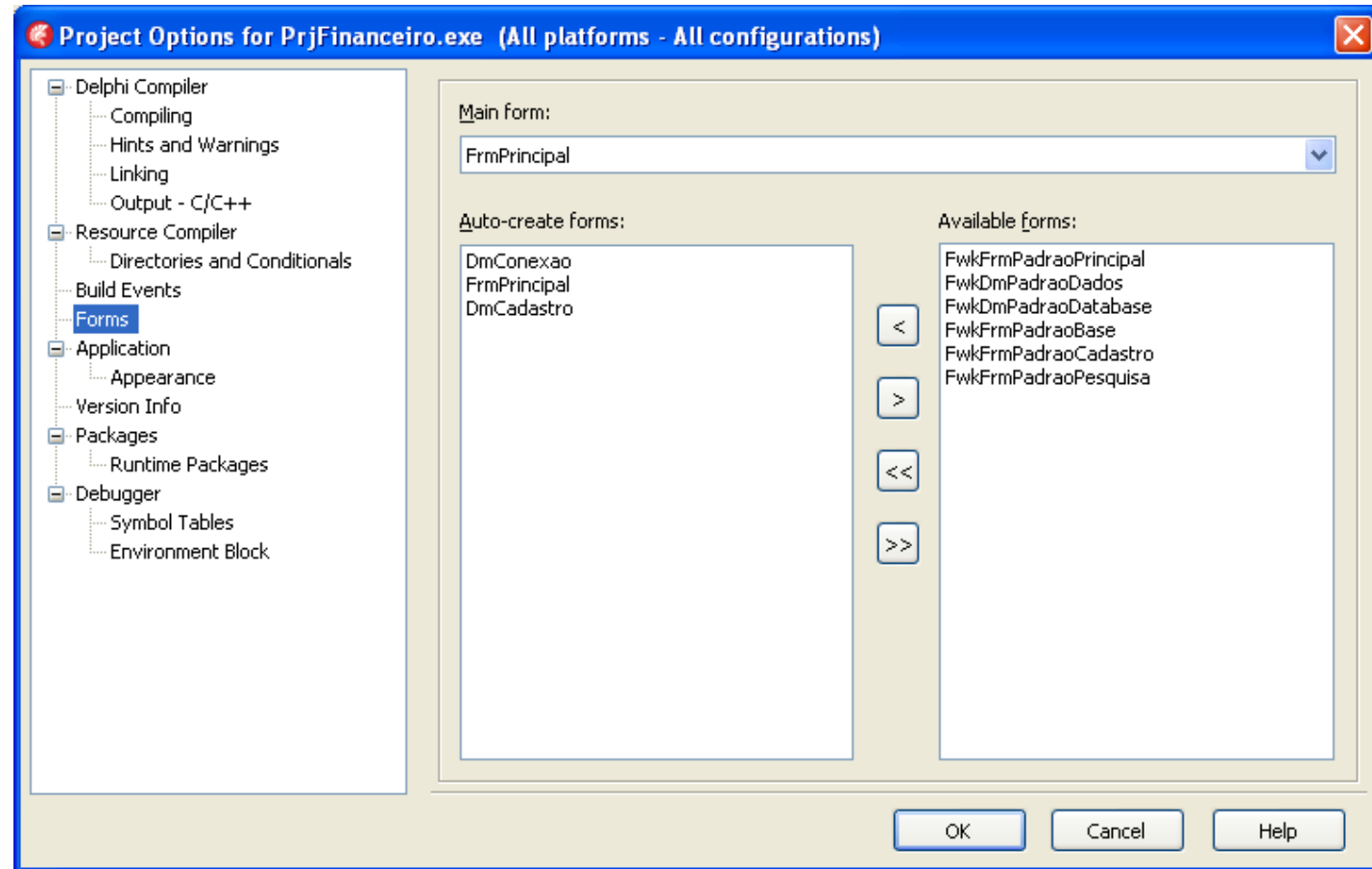
Deixei somente o **FrmPrincipal** e o **DmConexao** no **Auto-create forms**. Depois clique em OK.

Deixe na seguinte ordem:

DmConexao

FrmPrincipal

DmCadastro



Criando o Cadastro de Entidades

No Data Module denominado **DmCadastro** adicione um novo componente **TFDQuery**.

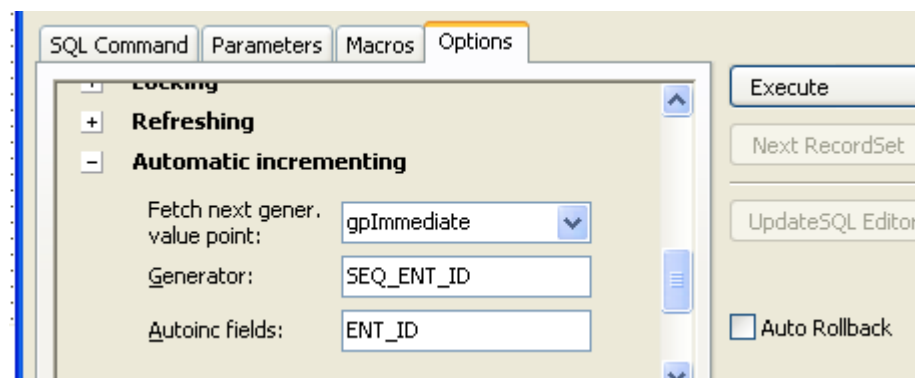
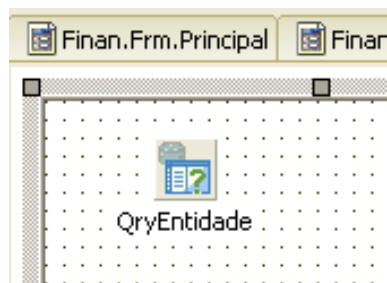
Coloque o nome deste componente de **QryEntidade**.

Na sua propriedade **Connection** selecione o **DmConexao.FDConnection**.

De um duplo clique sobre a **QryEntidade** e informe o comando **SQL**.

Depois adicione e configure os **Fields** do **DataSet** (**DisplayLabel**, **ProviderFlags**).

Configure o auto incremento automático (**gpImmediate**) do Field **ENT_ID** através das propriedades **Options** do componente **QryEntidade**.



Criando o Cadastro de Entidades

Agora é necessário fazer uma herança do formulário criado no Framework **FwkFrmPadraoCadastro**.

Para isso no Delphi vá em **File > New > Other...**

Selecione a opção **Inheritable Items** e selecione o **FwkFrmPadraoCadastro**.

Salve o Formulário da seguinte forma: **Name = FrmEntidade**, **Unit = Finan.Frm.Entidade.pas** em **C:\PrjFinanceiro\Aplicacao**.

No **Caption** do Formulário coloque **CADASTRO DE ENTIDADES**.

De Uses do **DmCadastro**.

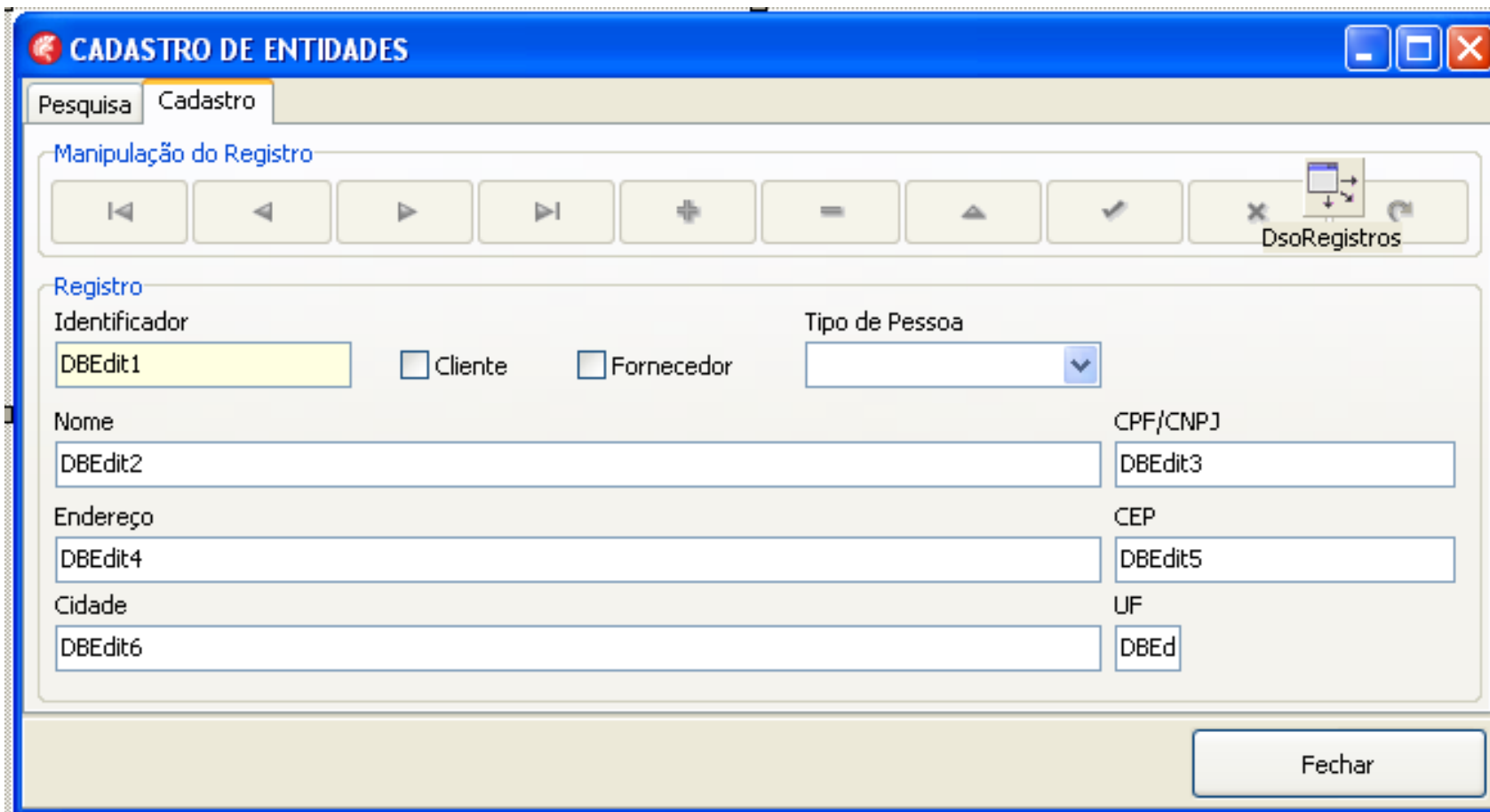
Ligue o **DsoRegistros** ao componente **DmCadastro.QryEntidade**.

Adicione os componentes **Data Controls** ao Formulário e ligue eles ao **DsoRegistros** e seus respectivos **Fields**.

Deixei o DBEdit do ENT_ID com Enabled = False e configure o color com clInfoBk.

Retire o **FrmEntidade** do **Auto-create forms**.

Criando o Cadastro de Entidades



The screenshot shows a software window titled "CADASTRO DE ENTIDADES" with a blue title bar and standard Windows window controls. Inside, there are two tabs: "Pesquisa" and "Cadastro", with "Cadastro" being the active tab. Below the tabs is a section titled "Manipulação do Registro" containing a row of navigation buttons: first and previous, previous and next, next and last, first, last, insert, delete, and a button labeled "DsoRegistros" with a refresh icon. The main area is titled "Registro" and contains several input fields and checkboxes. The "Identificador" field (DBEdit1) is highlighted in yellow. To its right are checkboxes for "Cliente" and "Fornecedor", and a "Tipo de Pessoa" dropdown menu. Below these are fields for "Nome" (DBEdit2), "CPF/CNPJ" (DBEdit3), "Endereço" (DBEdit4), "CEP" (DBEdit5), "Cidade" (DBEdit6), and "UF" (DBEdit). A "Fechar" button is located at the bottom right of the window.

CADASTRO DE ENTIDADES

Pesquisa Cadastro

Manipulação do Registro

DBEdit1 ☐ Cliente ☐ Fornecedor Tipo de Pessoa

Nome DBEdit2 CPF/CNPJ DBEdit3

Endereço DBEdit4 CEP DBEdit5

Cidade DBEdit6 UF DBEdit

Fechar

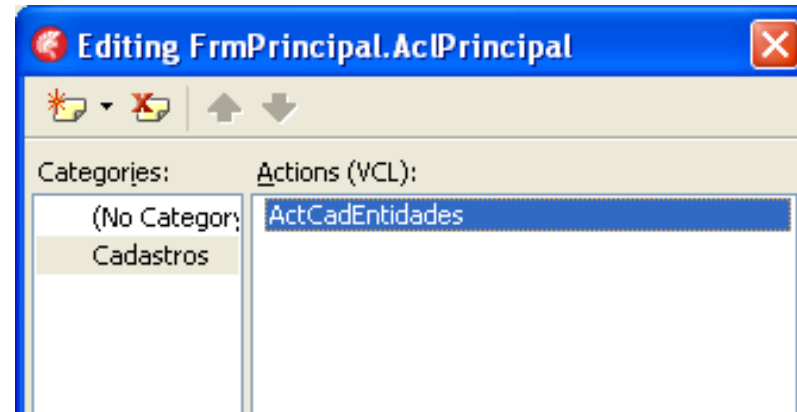
Criando o Cadastro de Entidades

No **FrmPrincipal** de um duplo clique sobre o componente **AclPrincipal**.

Adicione uma nova ação **ActCadEntidades**.

Na **Action** em **Category** informe **Cadastros**.

E no **Caption** informe **Entidades**.



De um duplo clique sobre a **Action** e informe o código que chama o **Formulário** de Entidades.

Lembre-se de dar **Uses** do **FrmEntidade**.

Criando o Cadastro de Entidades

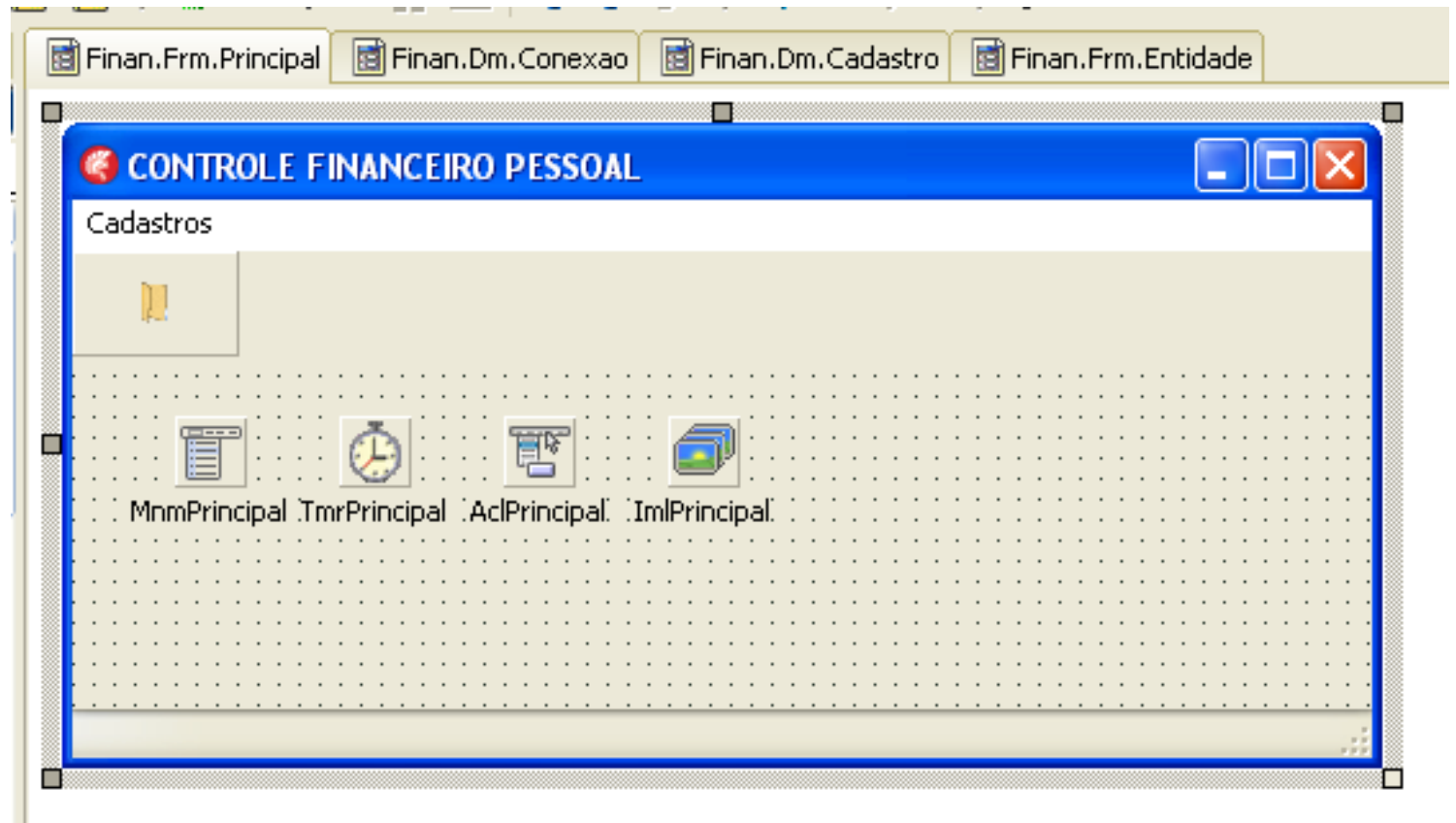
```
procedure TFrmPrincipal.ActCadEntidadesExecute(Sender: TObject);  
var  
    vFrmEntidade: TFrmEntidade;  
begin  
    vFrmEntidade := TFrmEntidade.Create(nil);  
    try  
        vFrmEntidade.ShowModal;  
    finally  
        FreeAndNil(vFrmEntidade);  
    end;  
end;
```

Criando o Cadastro de Entidades

Agora no **Toolbar** do **FrmPrincipal** adicione um novo botão e vincule ele a **Action ActCadEntidades**.

No **MainMenu** também adicione uma novo sub-menu chamado **Cadastros**.

Depois crie um novo item no menu e vincule ele também a **Action ActCadEntidades**.



Exemplo de Uso da Tela de Pesquisa

De uses da tela padrão de pesquisa do Framework **FwkFrmPadraoPesquisa**.

No método que vai chamar a tela de pesquisa utilize o seguinte código para buscar e depois jogar os valores pesquisados ao seu cadastro.

```
with Pesquisar (DmCadastro.QryEntidade) do
begin
    if Active then
    begin
        if not (DmCadastro.QryEntidade.State in [dsInsert, dsEdit]) then
            DmCadastro.QryEntidade.Edit;

            DmCadastro.QryEntidadeENT_NOME.AsString := FieldByName('ENT_NOME').AsString;
            DmCadastro.QryEntidadeENT_CPFCNPJ.AsString := FieldByName('ENT_CPFCNPJ').AsString;

            Close;
        end;
    end;
```

Exercitando

Agora criei os próximos cadastros baseado no que foi feito no cadastro de Entidades.

1. Categoria;
2. Contas a Pagar;
3. Contas a Receber;

Dúvidas?

E-mail

ezequieljuliano@gmail.com

