

**Curso**  
TÉCNICO EM INFORMÁTICA  
HABILITAÇÃO EM SISTEMAS DE INFORMAÇÃO

**Disciplina**  
Programação Delphi

# **Desenvolvimento de Sistemas**

***Criação: Prof. Alexandre Bendlin***

**Versão: Abril/2008**

# Índice

1. Fazendo Download do FireBird.....	3
2. Instalando o FireBird.....	5
3. Instalando o MDO (Mercury Database Objects).....	10
4. Palheta Mercury.....	10
4.1. Componentes da palheta Mercury.....	11
5. Conhecendo os componentes de controle de dados.....	20
5.1. Conhecendo alguns Componentes da palheta Jv Data Controls.....	22
6. Criando o Banco da Apostila.....	25
7. Usando os componentes MDOQuery e MDOUpdateSQL.....	28
8. Mestre / Detalhe com MDOQuery e MDOUpdateSQL.....	32
9. Criando Gráficos.....	35
10. Construindo Relatórios.....	40
11. Projeto Piloto.....	52
11.2. Aplicação do Projeto Piloto.....	57
11.2.1. Criando a Janela Principal.....	57
11.2.2. Criando a Janela de Login.....	63
11.2.3. Criando a Janela de Abertura.....	66
11.2.4. Criando a Janela Sobre.....	68
11.2.8. Criando a Janela Relatório (Qreport).....	79
11.2.9. Criando a Janela de Cadastro Padrão.....	85
11.2.10. Criando a Janela Cadastro de Grupos.....	104
11.2.11. Criando a Janela Cadastro de Materiais.....	112
12. Tratamento e Tradução dos Erros Globais.....	125

# 1. Fazendo Download do FireBird

- 1) Acesse o site: <http://www.firebirdsql.org/>;
- 2) Na barra de menu deste site clique na opção **Download** e novo menu apresentado clique em **Firebird Relational Database**;



- 3) Na nova página apresentada clique em **Download Firebird 2.0.3**;



4) Na página de **Firebird V.2.0.x Downloads**, clique no link **Windows 32-bit Superserver & Classic** para fazer o download;

**Firebird**  
... going where you're going

→ [HOME](#) [Download](#) [Documentation](#) [Resources](#) [Development](#) [Foundation](#) [License](#)

**Firebird V.2.0.x Downloads**

Latest release: v.2.0.3

**Download**

Our main commodity is the Firebird Relational Database Engine, but Firebird project is much more than the engine itself.

- ◆ [Firebird Database Engine](#)
- ◆ [Firebird ODBC Driver](#)
- ◆ [Firebird JCA-JDBC Driver](#)
- ◆ [Firebird .NET Data Provider](#)
- ◆ [InterClient](#)
- ◆ Test system
- ◆ [Documentation](#)
- ◆ [Firebird Quick Start Guide](#)

You can also get all our sources directly from our [CVS](#).

All our releases are also listed [here](#).

**Platforms**

<a href="#">Windows 32-bit Superserver &amp; Classic</a>	Windows 64-bit (not available)
Linux x86 Classic	Linux x64 Classic
Linux x86 Superserver (non-NPTL)	Linux x64 Superserver
Linux x86 Superserver NPTL	MacOSX Classic & Superserver for x86 & PPC
Solaris 10 Intel & Sparc Classic	Solaris 10 Intel Superserver
HP-UX 11 Classic Only	

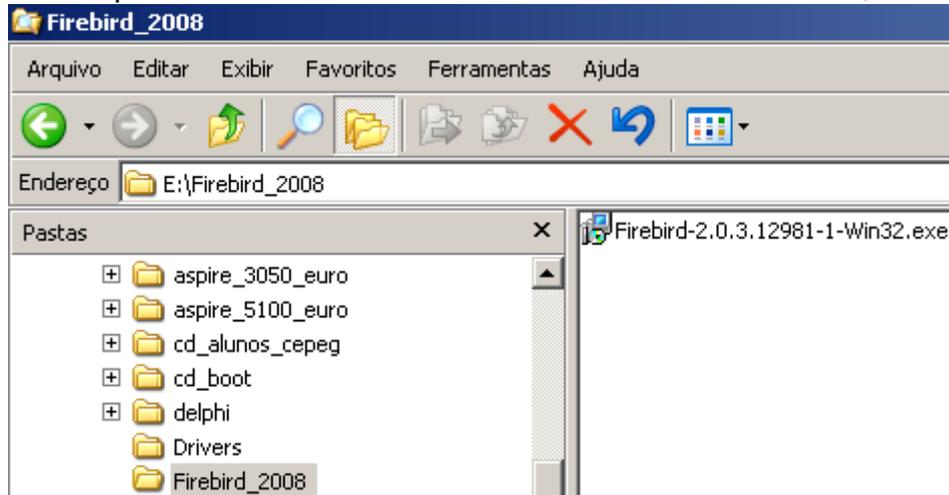
5) Agora clique no link **Firebird-2.0.3.12981-1-Win32.exe** para fazer o download;

**Win32 Superserver and Classic**

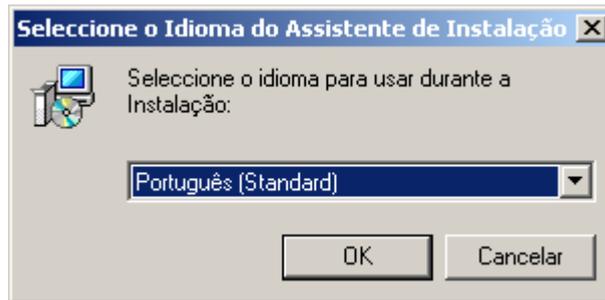
Release Date	File Name	Size	Description
2007-09-24	<a href="#">Firebird-2.0.3.12981-1-Win32.exe</a>	4.1 Mb	Official Windows Setup and Installer For Classic and SuperServer
2007-09-24	<a href="#">Firebird-2.0.3.12981-1_win32.zip</a>	5.6 Mb	SuperServer and Classic for Windows, zipped archive with folders structure
2007-09-24	<a href="#">Firebird-2.0.3.12981-1_embed_win32.zip</a>	3.1 Mb	Embedded Server for Windows
<i>Debug Builds</i>			
2007-09-24	<a href="#">Firebird-2.0.3.12981-1-Win32_pdb.exe</a>	7.0 Mb	Windows Setup and Installer For Classic and SuperServer Debug Build
2007-09-24	<a href="#">Firebird-2.0.3.12981-1_win32_pdb.zip</a>	10.4 Mb	Windows SS and Classic Debug Build
2007-09-24	<a href="#">Firebird-2.0.3.12981-1_embed_win32_pdb.zip</a>	5.3 Mb	Embedded Server Debug Build for Windows

## 2. Instalando o FireBird

- 1) Execute o arquivo do FireBird: **Firebird-2.0.3.12981-1-Win32.exe**;



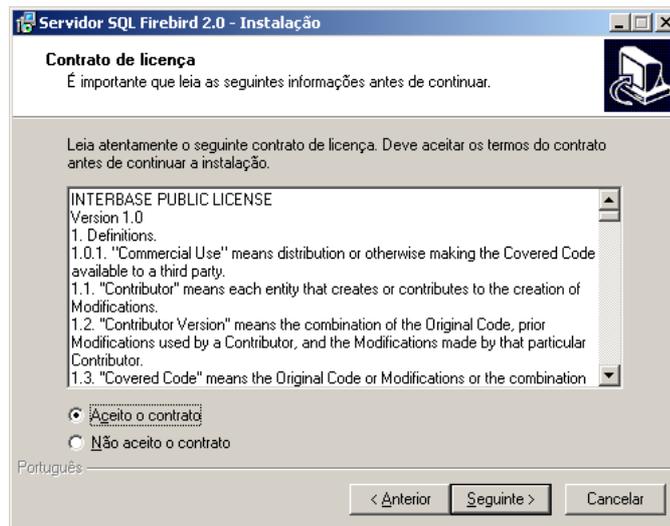
- 2) Na janela **Selecione o Idioma do Assistente de instalação**, deixe selecionada a opção **Português (Standard)** e dê um clique no botão **OK**;



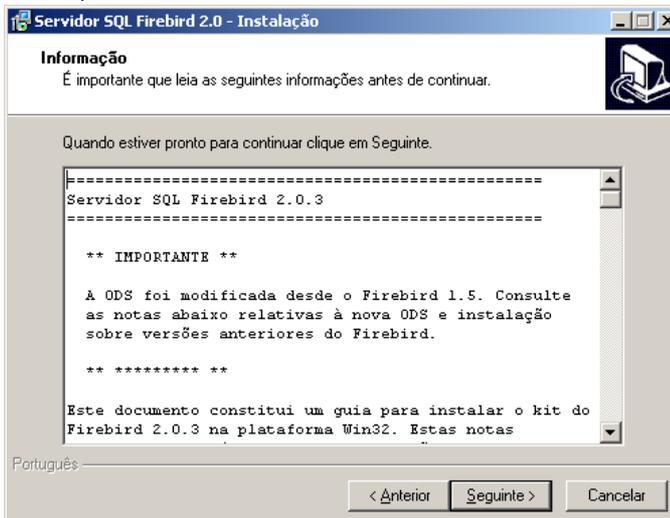
- 3) Na janela **Servidor SQL Firebird 2.0 – Instalação**, dê um clique no botão **Seguinte >**;



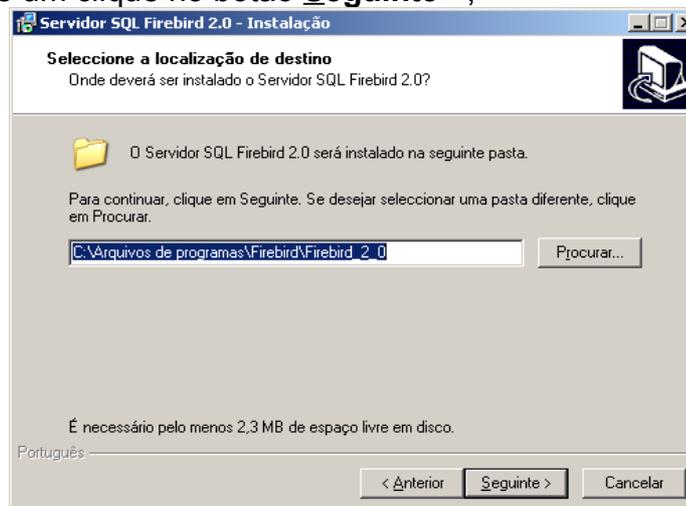
- 4) Na janela **Servidor SQL Firebird 2.0 – Instalação [Contrato de licença]**, selecione a opção **Aceito o contrato** e dê um clique no botão **Seguinte >**;



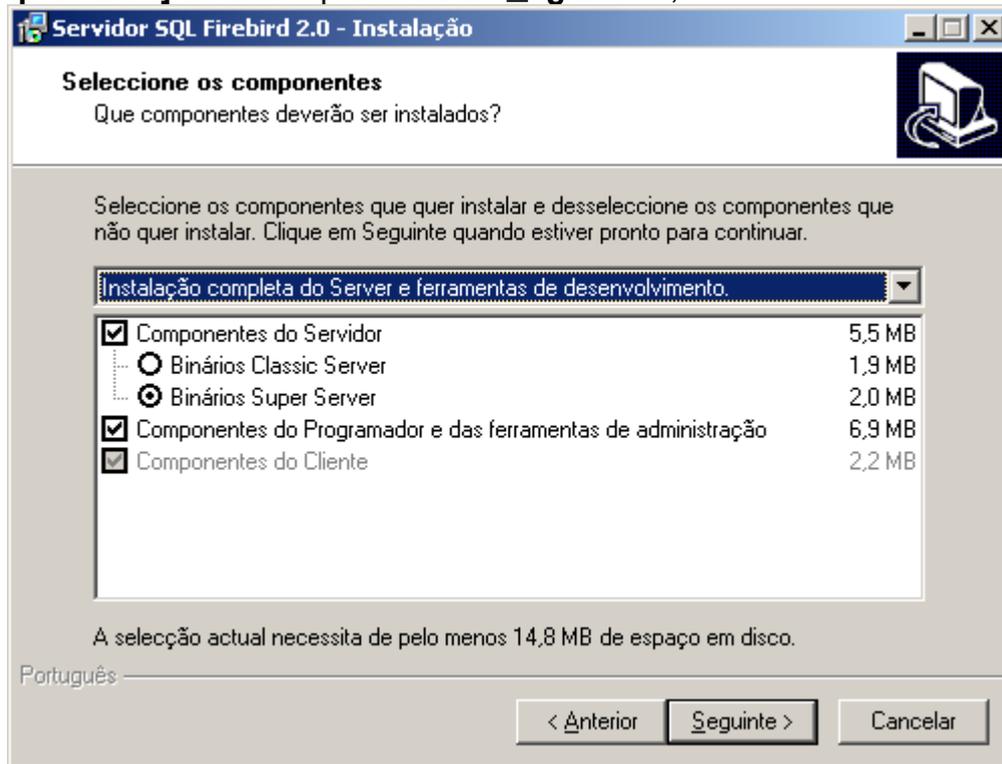
- 5) Na janela **Servidor SQL Firebird 2.0 – Instalação [Informação]**, dê um clique no botão **Seguinte >**;



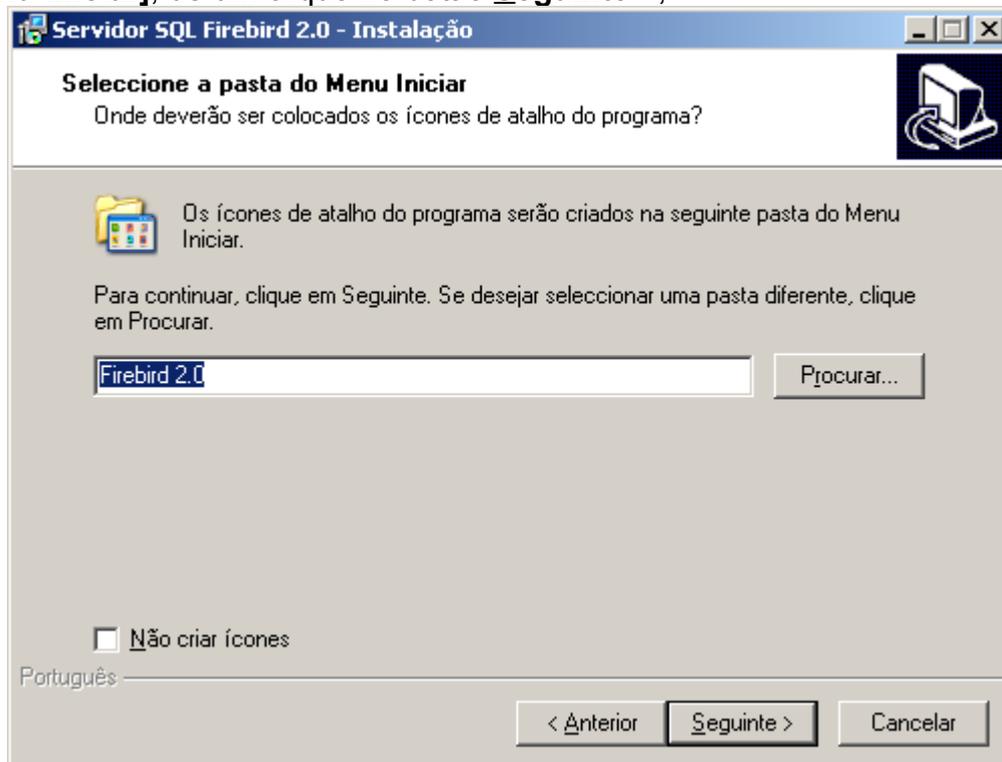
- 6) Na janela **Servidor SQL Firebird 2.0 – Instalação [Selecione a localização de destino]** dê um clique no botão **Seguinte >**;



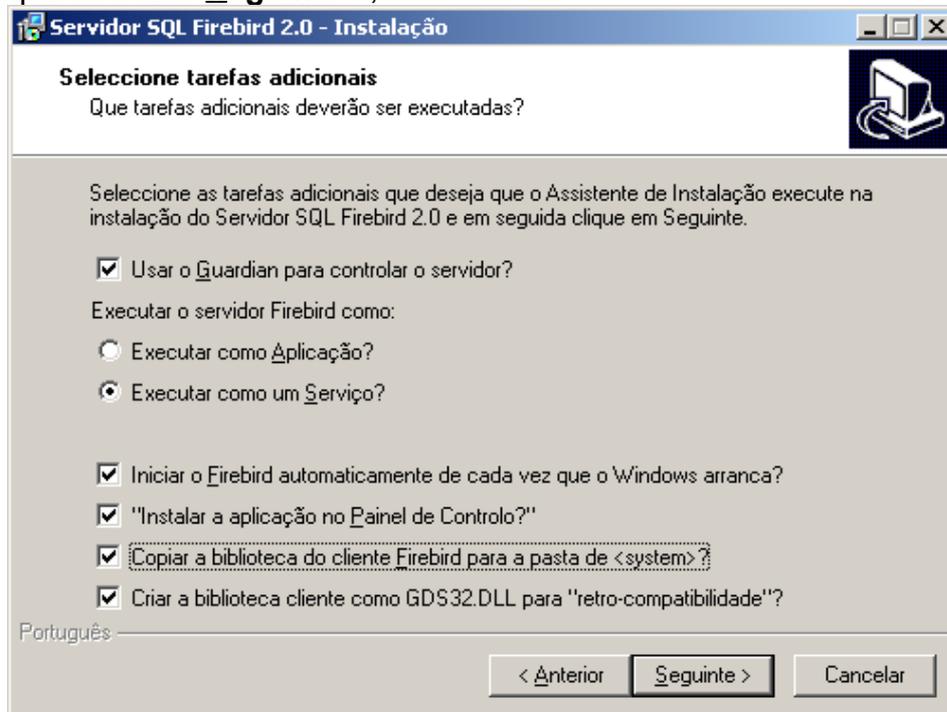
- 7) Na janela **Servidor SQL Firebird 2.0 – Instalação [Selecione os componentes]** dê um clique no botão **Seguinte >**;



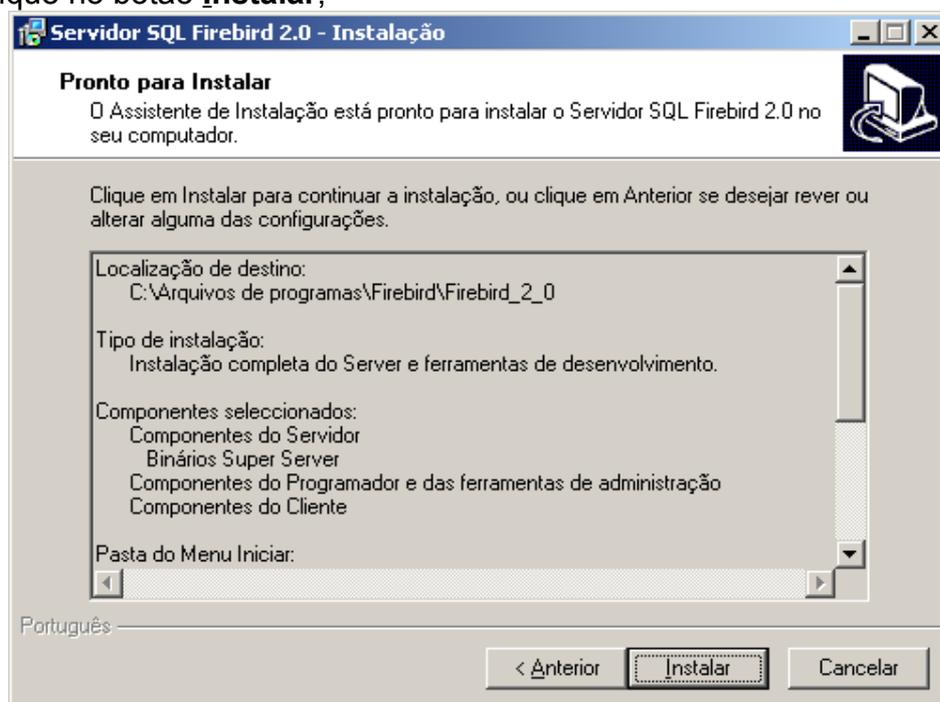
- 8) Na janela **Servidor SQL Firebird 2.0 – Instalação [Selecione a pasta do Menu Iniciar]**, dê um clique no botão **Seguinte >**;



- 9) Na janela **Servidor SQL Firebird 2.0 – Instalação [Selecione tarefas adicionais]**, marque as opções: **Usar o Guardian para controlar o servidor?**, **Executar como um Serviço?**, **Iniciar o Firebird automaticamente de cada vez que o Windows arranca?**, **Copiar a biblioteca do cliente Firebird para a pasta de <system>?**, **Criar a biblioteca cliente como GDS32.DLL para "retro-compatibilidade"?** e **Instalar a aplicação no Painel de Controlo?** e, dê um clique no botão **Seguinte >**;

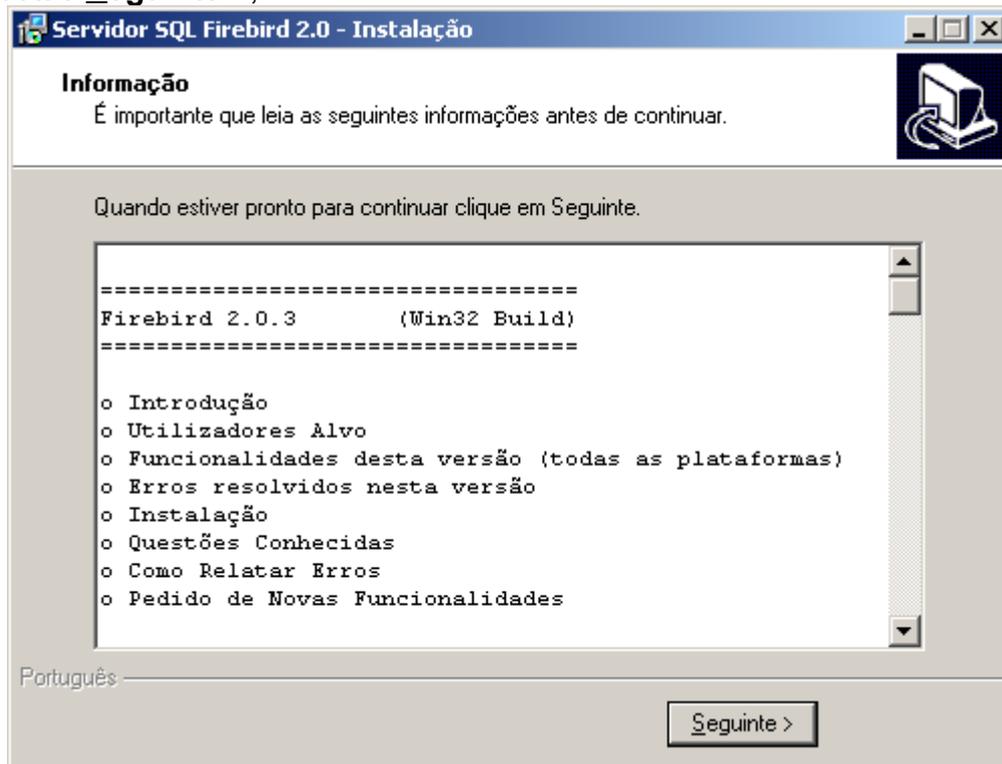


- 10) Na janela **Servidor SQL Firebird 2.0 – Instalação [Pronto para Instalar]**, dê um clique no botão **Instalar**;



- 11) Aguarde o término da instalação;

12) Na janela **Servidor SQL Firebird 2.0 – Instalação [Informação]**, dê um clique no botão **Seguinte >**;



13) Na janela **Servidor SQL Firebird 2.0 – Instalação**, dê um clique no botão **Concluir**;

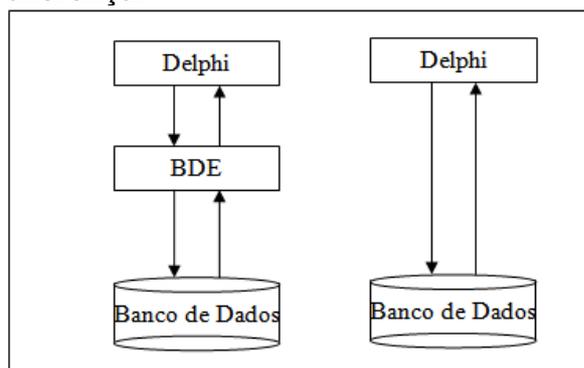


### 3. Instalando o MDO (Mercury Database Objects)

1. Baixe a última versão do MDO no site <http://sourceforge.net/projects/mdo>;
2. Desinstale qualquer versão anterior do MDO, caso exista;
3. Apague todos os arquivos .BPL relacionados ao MDO já existente, que geralmente se encontram na pasta (diretório): C:\Arquivos de programas\Borland\Delphi7\Projects\Bpl;
4. Descompacte-o na pasta (diretório):  
Delphi 7: C:\Arquivos de programas\Borland\Delphi7\mdo-rc2
5. Abra o arquivo **MDO\_D7.BPG** para instalação no **Delphi 7**;
6. Na **IDE do Delphi** selecione: **Project / Build All Projects**;
7. Para **Delphi 7** abra **DCLMDO70?.DPK**, então clique em **Install**;
8. Adicione o caminho onde o código fonte dos arquivos de runtime está armazenado, como por exemplo: "C:\Arquivos de programas\Borland\Delphi7\mdo-rc2\Source\runtime". Para isto, selecione: **Tools / Environment Options**, clique na aba **Library** então adicione em **Library Path**.

### 4. Palheta Mercury

A palheta Mercury fornece o melhor método para criar uma aplicação Delphi acessando base de dados FireBird. Uma aplicação compilada em Delphi 7, utilizando os componentes da palheta Mercury apresentará uma performance superior do que as aplicações compiladas em versões anteriores do Delphi. Isto porque os componentes Mercury dispensam a camada BDE – ou qualquer outra camada. Uma aplicação construída com estes componentes terá o código de acesso ao banco “built-in”, ou seja, dentro do próprio executável. Repare na figura abaixo a exemplificação desta diferença:



Além de maior desempenho, o desenvolvedor passa a ganhar outros recursos. Com esses objetos é possível extrair várias informações do servidor, como memória disponível, espaço em disco, banco de dados utilizado, percentual de recursos disponíveis e outras. Um maior controle do SQL e das transações realizadas também foi adicionado aos componentes. Apesar de todas estas vantagens, os componentes Mercury oferecem um óbvio ponto negativo: Uma aplicação construída com estes componentes deverá ser totalmente reconstruída caso o cliente resolva mudar de banco de dados.

## 4.1. Componentes da palheta Mercury

A seguir encontra-se uma breve descrição dos principais componentes da palheta. Eles estão listados por ordem de importância dentro do projeto:



### **MDODatabase**

Representa a instância do banco de dados da aplicação, e realiza a conexão ao banco através da propriedade Connected.

#### Principais **Propriedades**:

Connected	Quando <b>True</b> inicia a conexão com o servidor.
DatabaseName	Nome do arquivo de banco de dados.
DefaultTransaction	Indica um objeto <b>MDOTransaction</b> como objeto de transação <i>Default</i> .
IdleTimer	Especifica quanto tempo o cliente irá esperar caso o servidor não envie nenhuma resposta. Se o tempo for ultrapassado e o servidor não responder, a conexão será desfeita.
LoginPrompt	Se <b>True</b> , o objeto pedirá nome e senha ao usuário.
SQLDialect	Indica o código de dialeto SQL utilizado pelo cliente.
TraceFlags	Indica quais serão as ações monitoras pelo objeto MDOSQLMonitor.

#### Principais **Eventos**:

AfterConnect	Ocorre após a conexão ser realizada.
AfterDisconnect	Ocorre após a conexão ser terminada.
BeforeConnect	Ocorre ao pedido de conexão.
BeforeDisconnect	Ocorre ao pedido de término da conexão.
OnIdleTimer	Ocorre enquanto o cliente espera por uma resposta do servidor.
OnLogin	Este evento pode ser usado para substituir o pedido de <i>username</i> e <i>password</i> do objeto. Ao definir alguma rotina neste evento, automaticamente a janela de login não será exibida e o nome e a senha deverão ser passadas via código.

### **MDOTransaction**

O componente MDOTransaction representa a instância da transação atual e simplifica muito o controle de transações. Com este objeto, é possível controlar transações concorrentes originárias da mesma conexão com o banco de dados, ou em Threads independentes.

### Principais **Propriedades**:

Active	Quando <b>True</b> , executa o método <i>StartTransaction</i> no servidor.
DefaultAction	Indica qual será o método executado quando a aplicação exceder o tempo de idle time out, ou seja, quando nenhum comando for mais enviado para o servidor. Seus valores são: <b>taRollback</b> : Executa a instrução de <i>RollBack</i> . <b>taCommit</b> : Executa a instrução de <i>Commit</i> . <b>taRollbackRetaining</b> : Executa a instrução <i>RollBack</i> e mantém o <i>handle</i> da transação aberto (os dados continuam na tela). <b>taCommitRetaining</b> : Executa a instrução <i>Commit</i> e mantém o <i>handle</i> da transação aberto (os dados continuam na tela).
DefaultDatabase	Indica a instância do banco onde a transação acontecerá.
IdleTimer	Especifica quanto tempo o objeto irá esperar para executar o método especificado na propriedade <i>DefaultAction</i> . O tempo começa a ser contado a partir do momento que nenhuma instrução for enviada para o servidor.

### Principais **Eventos**:

OnIdleTimer	Ocorre enquanto a aplicação não envia nenhuma instrução SQL para o servidor.
-------------	--

### Principais **Métodos**:

Commit	Confirma a transação no servidor.
CommitRetaining	Confirma a transação e mantém o <i>handle</i> aberto. Neste caso, os dados permanecem na tela.
Rollback	Cancela a transação atual.
RollbackRetaining	Cancela a transação e mantém o <i>handle</i> aberto. Neste caso, os dados permanecem na tela.
StartTransaction	Inicia a transação.
Call	Retorna as mensagens de erro baseado no ' <i>Error Code</i> ' passado como parâmetro.

### **MDOTable**

Representa uma conexão com alguma tabela do banco de dados. O uso do objeto MDOTable apresenta desvantagens em relação ao uso do objeto MDOQuery, visto que este componente envia uma quantidade de instruções muito maior ao servidor. É recomendável o uso do objeto MDOQuery.

### Principais **Propriedades**:

Active	Abre a conexão com a tabela.
Database	Objeto MDODatabase.
TableName	Nome da tabela dentro do banco de dados.
Transaction	Objeto MDOTransaction, que representa a transação ativa para as ações executadas através deste objeto.
ReadOnly	Quando <b>True</b> , permite acesso aos dados do DataSet para leitura.
BOF	Identifica se o ponteiro da tabela está antes do primeiro registro.
EOF	Identifica se o ponteiro da tabela está depois do último registro.

**Principais Eventos:**

OnAfterInsert	Ocorre após a inserção de um registro.
OnBeforeInsert	Ocorre antes da inserção de um registro.
OnAfterEdit	Ocorre após a edição de um registro.
OnBeforeEdit	Ocorre antes da edição de um registro.
OnAfterCancel	Ocorre após o cancelamento da operação corrente.
OnBeforeCancel	Ocorre antes do cancelamento da operação ativa.
OnAfterClose	Ocorre após a tabela ser fechada.
OnBeforeClose	Ocorre antes de a tabela ser fechada.
OnAfterOpen	Ocorre após a tabela ser aberta.
OnBeforeOpen	Ocorre antes de a tabela ser aberta.
OnAfterDelete	Ocorre após uma exclusão de registro.
OnBeforeDelete	Ocorre antes de uma exclusão de registro.
OnAfterPost	Ocorre após a efetivação da última operação de inserção ou edição.
OnBeforePost	Ocorre antes da concretização da última operação de inserção ou edição.
OnAfterScroll	Ocorre após ter havido um movimento no ponteiro da tabela.
OnBeforeScroll	Ocorre antes de haver um movimento no ponteiro da tabela.
OnCalcFields	Ocorre quando um campo calculado passa pela operação que o calcula.
OnNewRecord	Ocorre quando um novo registro é incluído por qualquer dos métodos de inserção de registros.
OnDeleteError	Ocorre quando um erro no processo de deleção de um registro é detectado.
OnEditError	Ocorre quando um erro no processo de edição é detectado.
OnPostError	Ocorre quando um erro no processo de concretização de uma operação de inserção ou edição de um registro é detectado.

**Principais Métodos:**

Open	Abre a tabela.
Close	Fecha a tabela.
First	Desloca o ponteiro da tabela para o primeiro registro.
Last	Desloca o ponteiro da tabela para o último registro.
Next	Desloca o ponteiro da tabela para o próximo registro.
Prior	Desloca o ponteiro da tabela para o registro anterior.
MoveBy	Movimenta o cursor em N posições. Se N for positivo haverá um avanço do ponteiro e se N for negativo haverá um recuo.
Locate	Permite localizar um registro na tabela com base em uma ou mais chaves especificadas. Nesse método não há necessidade de a tabela estar indexada. O método possibilita a não distinção entre maiúsculas e minúsculas e <b>pesquisa</b> por chave <b>exata</b> ou <b>aproximada</b> .
Insert	Inclui um novo registro na tabela.
FieldByName	Sintaxe: <table>.FieldByName(<campo>).<tipo>; Exemplo: tblClientes.FieldByName('CLI_NOME').AsString; Este método é usado para acessar o conteúdo de uma coluna de tabela ou armazenar um conteúdo nela.

Delete	Deleta o registro corrente da tabela.
Edit	Coloca o registro corrente da tabela em estado de edição.
Post	Concretiza as operações de inserção (insert) e alteração (edit).
Cancel	Cancela uma operação de inserção (insert) ou alteração (edit).
EmptyTable	Deleta todos os registros da tabela.

### **MDOQuery**

Representa uma conexão SQL com o banco de dados. Praticamente toda a linguagem SQL(DDL/DML) suportada pelo banco de dados pode ser utilizada através deste objeto. Para utilizar o objeto MDOQuery como um objeto de edição, e recomendável o seu uso juntamente com o objeto MDOUpdateSQL.

#### **Principais Propriedades:**

Database	Objeto MDODatabase, que representa a instância do banco de dados.
SQL	Código SQL a ser enviado para o servidor.
Transaction	Objeto MDOTransaction, que representará a transação ativa para as ações executadas através deste objeto.
Active	Habilita geração do resultado do comando SQL.
RecordCount	Propriedade de run-time, que mantém o número de linhas retornadas após o último comando SQL.
RowsAffected	Propriedade de run-time, que mantém o número de linhas afetadas após a última operação de atualização (Deleção ou alteração).

#### **Principais Eventos:**

OnAfterInsert	Ocorre após a inserção de um registro.
OnBeforeInsert	Ocorre antes da inserção de um registro.
OnAfterEdit	Ocorre após a edição de um registro.
OnBeforeEdit	Ocorre antes da edição de um registro.
OnAfterCancel	Ocorre após o cancelamento da operação que estava sendo realizada.
OnBeforeCancel	Ocorre antes do cancelamento da operação que estava sendo realizada.
OnAfterClose	Ocorre após a tabela ser fechada.
OnBeforeClose	Ocorre antes de a tabela ser fechada.
OnAfterOpen	Ocorre após a tabela ser aberta.
OnBeforeOpen	Ocorre antes de a tabela ser aberta.
OnAfterDelete	Ocorre após uma deleção de registro.
OnBeforeDelete	Ocorre antes de uma deleção de registro.
OnAfterPost	Ocorre após a concretização da última operação de inserção ou edição.
OnBeforePost	Ocorre antes da concretização da última operação de inserção ou edição.
OnAfterScroll	Ocorre após ter havido um movimento no ponteiro da tabela.
OnBeforeScroll	Ocorre antes de haver um movimento no ponteiro da tabela.
OnCalcFields	Ocorre quando um campo calculado passa pela operação que o calcula.

OnNewRecord	Ocorre quando um novo registro é incluído por qualquer dos métodos de inserção de registros.
OnDeleteError	Ocorre quando um erro no processo de deleção de um registro é detectado.
OnEditError	Ocorre quando um erro no processo de edição de um registro é detectado.
OnPostError	Ocorre quando um erro no processo de concretização de uma operação de inserção ou edição de um registro é detectado.

#### Principais **Métodos**:

ExecSQL	Executa um comando <b>SQL</b> . Para caso de uso dos comandos de atualização ( <b>INSERT</b> , <b>UPDATE</b> e <b>DELETE</b> ), este método deve ser invocado. Os comandos <b>SQL</b> de atualização não retornam linhas. Portanto, o uso de <b>Open</b> para operações de atualização não é permitido, visto que é um método existente para abrir uma consulta.
Clear	Limpa o conteúdo da propriedade SQL.
Add	Realiza a montagem do comando <b>SQL</b> por meio de uma seqüência de linhas inseridas usando-se o método <b>Add</b> .
<b>Exemplo:</b> <pre>qryMateriais.Sql.Close; qryMateriais.Sql.Clear; qryMateriais.Sql.Add('Select * From MATERIAIS'); qryMateriais.Sql.Add('Where MAT_CODIGO &gt; 100'); qryMateriais.Sql.Open;</pre>	
SaveToFile	Salva o commando SQL atualmente em uso em um arquivo texto. <b>Exemplo:</b> <pre>qryMateriais.Sql.SaveToFile('c:\materiais_sql.txt');</pre>
LoadFromFile	Carrega um comando SQL armazenado em um arquivo texto. <b>Exemplo:</b> <pre>qryMateriais.Sql.LoadFromFile('c:\materiais_sql.txt'); qryMateriais.Open;</pre>

**Nota:** Os métodos **Insert**, **Delete**, **IndexOf** e **Move** aplicáveis a itens de uma lista, também são aplicáveis à propriedade **SQL**.

#### **MDOStoredProc**

Permite executar uma procedure armazenada no servidor.

#### Principais **Propriedades**:

DataBase	Objeto MDODatabase, que representa a instância do banco.
StoreProcName	Nome da procedure armazenada.

#### **MDOUpdateSQL**

Permite configurar o objeto MDOQuery ReadOnly para trabalhar com Live Result Sets. Com este objeto, é possível definir instruções SQL para cada método Append, Edit ou Delete. O objeto MDOUpdateSQL garante maior performance para editar uma tabela, pois a query trabalha em modo ReadOnly.

### Principais **Propriedades**:

InsertSQL	Define a instrução SQL a ser executada quando o método Append for chamado.
ModifySQL	Define a instrução SQL a ser executada quando a tabela for editada.
DeleteSQL	Define a instrução SQL a ser executada quando algum registro for excluído.
RefreshSQL	Define a instrução SQL a ser executada quando o método Refresh for executado.

### **MDODataset**

Este componente é o objeto ancestral para os componentes de acesso a dados. Sua utilização não é essencial, pois o objeto TQuery possui todos os seus recursos.

### Principais **Propriedades**:

Active	Indica se a query será aberta.
BufferChunks	Número de registros no buffer.
Database	Objeto MDODatabase, que representa o banco de dados atual.
DeleteSQL	Instrução SQL que será executada quando algum registro excluído.
InsertSQL	Instrução SQL que será executada quando algum registro for inserido.
ModifySQL	Instrução SQL que será executada quando algum registro for alterado.
RefreshSQL	Instrução SQL que será executada quando o método Refresh for chamado.
SelectSQL	Instrução SQL que será executada quando a query for aberta.

### Principais **Eventos**:

OnAfterDatabaseDisconnect	Ocorre após o término da conexão com o banco.
OnAfterTransactionEnd	Ocorre após a execução de uma transação (com COMMIT ou ROLLBACK). Este evento captura apenas as transações "hard", ou seja, terminadas com o método Commit ou RollBack. Os métodos CommitRetaining e RollBackRetaining não são capturados por este evento.
BeforeDatabaseDisconnect	Ocorre antes do término da conexão com o banco.
OnBeforeTransactionEnd	Ocorre antes da execução de um comando Commit ou RollBack.
DatabaseFree	Ocorre após a liberação do handle's alocados pelo objeto MDODatabase.
TransactionFree	Ocorre após a liberação dos handle's alocados pelo objeto MDOTransaction.

## MDOSQL

Componente para executar instruções SQL. Sua diferença para com o objeto MDOQuery, é que este apresenta a maior performance de execução, e aloca o mínimo de recursos possíveis. Em contrapartida, este objeto não oferece interface para conexão com objetos DataControl – ele apenas executa as instruções. É recomendável o seu uso em operações que exijam um rápido desempenho, como por exemplo, rotinas de backup. Este componente também pode ser utilizado para criação de aplicações DSQL (Dynamic SQL), ou seja, aplicações em que o próprio usuário cria as instruções SQL.

### Principais **Propriedades**:

Database	Objeto MDODatabase que representa o banco de dados atual.
GoToFirstRecordOnExecute	Se <b>True</b> vai para o primeiro registro após a execução de uma instrução <b>SQL Live Result Set</b> .
ParamCheck	Esta propriedade deve ser True quando o objeto MDOSQL possuir uma instrução SQL DDL com parâmetros. Se <b>True</b> , o objeto irá prevenir de não misturar os parâmetros definidos na propriedade SQL com os parâmetros contidos dentro da instrução SQL. Por exemplo, se a instrução criar uma Stored Procedure com um parâmetro, setar esta propriedade para True irá prevenir o objeto de tentar substituir o parâmetro da instrução <b>CREATE PROCEDURE</b> por um valor da propriedade <i>Params</i> .
SQL	Instrução que será executada.
Transaction	Objeto MDOTransaction que irá representar a transação atual.

### Principais **Eventos**:

OnSQLchanging	Ocorre quando a instrução SQL é modificada.
---------------	---

### Principais **Métodos**:

Call	Retorna a mensagem correspondente ao código do erro.
CheckClosed	Gera uma exceção quando a query está aberta.
CheckOpen	Gera uma exceção quando a query está fechada.
CheckValidStatement	Gera uma exceção se a query não possuir uma cláusula SQL válida.
Close	Elimina os handles gerados pela query.
Current	Retorna o tipo TMDOXSQLDA atualmente alocado pela query. Este <i>descriptor</i> representa uma área de transferência de dados entre o banco de dados e a aplicação cliente. Toda execução de uma instrução SQL cria uma instância deste <i>Record</i> .
ExecQuery	Executa a query SQL.
FieldByName	Utilizado para acessar um campo da tabela pelo nome.
Next	Move o ponteiro para o próximo registro.
Prepare	Utilizado antes de passar parâmetros para o objeto. Permite preparar uma Query para execução.

## **MDODatabaseInfo**

Retorna varias informações a respeito do banco de dados especificado. Basta inserir um componente MDODatabaseInfo, configurar sua propriedade Database para o objeto MDODatabase correspondente, e capturar os valores de suas propriedades.

### Principais **Propriedades**:

Allocation	Retorna o número de páginas alocadas.
BackoutCount	Indica o número de vezes que um registro do banco de dados foi removido.
BaseLevel	Retorna o número de versão do banco de dados.
CurrentMemory	Retorna a quantidade de memória alocada pelo servidor.
DataBase	Objeto MDODatabase, que indicará o banco de dados provedor das informações.
DBFilename	Retorna o nome do arquivo de banco de dados.
DbImplementationClass	Retorna o número de classe do banco de dados.
DbImplementationNo	Retorna o número de implementação do banco.
DbSiteName	Retorna o nome do site do banco de dados.
DbSQLDialect	Retorna o número <b>SQL Dialect</b> utilizado pelo servidor.
DeleteCount	Retorna o número de <i>Deletes</i> executados desde a última conexão com o servidor.
ExpungeCount	Retorna o número de registros removidos.
Fetches	Retorna o número de <i>Fetchs</i> realizados.
ForcedWrites	Indica como a gravação de dados está sendo realizada. 0 para assíncrono e 1 para síncrono.
InsertCount	Retorna o número de <i>Inserts</i> executados desde a última conexão com o servidor.
Marks	Retorna o número de gravações efetuadas no buffer.
MaxMemory	Retorna, em bytes, a quantidade de memória utilizada desde a primeira conexão realizada.
NoReserve	Indica se foi reservado espaço para backup de registros modificados ou não.
NumBuffers	Indica o número de buffers alocados.
ODSMajorVersion	Retorna a parte superior da versão do ODS.
ODSMinorVersion	Retorna a parte inferior da versão do ODS.
PageSize	Retorna o número de bytes alocados por página.
PurgeCount	Retorna o número de <i>Purgings</i> realizados.
ReadIdxCount	Retorna o número de leituras feitas através de índices.
ReadOnly	Indica quando o banco de dados é ou não read-only.
Reads	Retorna o número de páginas lidas no banco.
ReadSeqCount	Retorna o número de leituras seqüenciais realizadas nas tabelas do banco.
SweepInterval	Retorna o número de transações confirmadas entre "sweeps".
UpdateCount	Retorna o número de <i>Updates</i> executados desde a última conexão com o servidor.
UserNames	Retorna em uma lista de String, os usuários conectados.
Version	Retorna a versão do FireBird.
Writes	Retorna o número de páginas de dados no banco.

## **MDOSQLMonitor**

O uso deste objeto é muito simples. Não existem propriedades a serem configuradas, basta inserir um componente MDOSQLMonitor na aplicação (ou em outro aplicativo). Após, o usuário deve configurar a propriedade TraceFlags do objeto Database da aplicação cliente, indicando quais serão os comandos filtrados pelo objeto MDOSQLmonitor.

As instruções SQL podem ser capturadas através do evento OnSQL do objeto. Este evento passa um parâmetro EventText, do tipo string, indicando o comando enviando para o servidor. Por exemplo, a aplicação poderia ter um ListBox para receber as instruções:

```
procedure TForm1.MDOSQLMonitor1SQL(EventText: String;
    EventTime: TDateTime);
begin
    ListBox1.Items.Add(EventText);
end;
```

Com este código, todos os comando enviados ao servidor (definidos na propriedade TraceFlags do objeto Database) seriam exibidos no ListBox.

## **MDOEvents**

Componente para utilizar os “eventos” do FireBird. O conceito de eventos no FireBird é muito interessante: o desenvolvedor pode programar eventos no banco de dados, que serão executados em instruções como INSERT, UPDATE ou DELETE. Estes eventos podem ser programados em Triggers ou Stored Procedures, e as aplicações que registrarem os eventos através do objeto MDOEvents, receberão uma notificação toda vez que algum registro for inserido ou deletado, por exemplo. Este objeto é muito útil para criar telas de refresh automático. Por exemplo, um evento poderia ser criado toda vez que um registro for inserido na tabela. Todas as máquinas clientes iriam receber este evento, e o software poderia ser programado para dar um Refresh em suas Query’s abertas, criando o efeito de Refresh automático sem causar Overhead de CPU. Este objeto roda em um Thread separado, e não degrada o desempenho do sistema.

### **Principais Propriedades:**

Database	Objeto MDODatabase, que representa o banco de dados atual.
Events	Nome dos eventos que serão recebidos pelo objeto.
Registered	Se True indica que o objeto estará ativo.

### **Principais Eventos:**

OnEventAlert	Ocorre quando algum evento é recebido pelo objeto. Configurar o parâmetro <b>CancelAlerts</b> para <b>True</b> , indica que o evento recebido deverá ser ignorado. Para retornar e receber determinado evento, basta chamar o método <b>QueueEvents</b> . Nem todas as operações envolvendo objetos da VCL poderão ser executadas neste evento, visto que ele roda em um Thread separado.
--------------	---

### Principais **Métodos**:

CancelEvents	Cancela o recebimento dos eventos pendentes.
QueueEvents	Indica ao objeto para iniciar o recebimento dos eventos.
RegisterEvents	Registra os eventos listados na propriedade <b>Events</b> . Este método já executa o método <b>QueueEvents</b> .
UnRegisterEvents	Cancela o recebimento dos eventos listados na propriedade <b>Events</b> .

### **MDOScript**

Componente para a execução de Scripts, este componente pode ser utilizado para criação de bancos bem como para criar um sistema de atualização de Metadados.

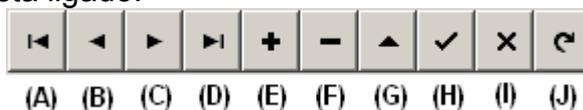
## 5. Conhecendo os componentes de controle de dados.



Os componentes relativos a dados exibem as informações do Banco de Dados (conectados através de componentes **Mercury** e **DataAccess**) no formulário. Estes componentes são muito parecidos com os componentes do *Windows*. Por exemplo: o componente **DBEdit** funciona da mesma forma que o **Edit**, com a diferença de exibir e permitir que se edite um dado de uma tabela.

### **DBNavigator**

O DBNavigator permite que o usuário realize operações padrão de controle de dados. Cada um dos botões do DBNavigator chama um método do componente MDOQuery ao qual está ligado.



#### Legenda:

- (A) – Primeiro (First);
- (B) – Anterior (Prior);
- (C) – Posterior (Next);
- (D) – Último (Last);
- (E) – Incluir (Insert);
- (F) – Excluir (Delete);
- (G) – Editar (Edit);
- (H) – Confirmar (Post);
- (I) – Cancelar (Cancel);
- (J) – Atualizar (Refresh).

Propriedades	Descrição
ConfirmDelete	Define se uma solicitação de exclusão de registro será exibida.
DataSource	DataSource que contém os valores a serem manipulados.
Hints	Hints exibidos pelos botões.
VisibleButtons	Define botões que serão visíveis.
Eventos	Descrição
BeforeAction	Quando um botão é pressionado, antes da ação ser executada.
OnClick	Quando um botão é pressionado, depois da ação ser executada.

### DBGrid

Mostra os registros de uma tabela em forma de grade, cada coluna é um campo e cada registro, uma linha.

Propriedades	Descrição
Columns	Lista do tipo TDBGridColumns, com as colunas da Grid, cada item da lista é do tipo TColumn.
DataSource	DataSource que contém os valores a serem manipulados.
Fields	Lista de objetos TField mostrado nas colunas.
FixedColor	Cor fixa, usada nas colunas e indicadores.
Options	Configurar as opções da Grid, como ConfirmDelete, MultiSelect, ColumnResize.
SelectedField	Campo da coluna selecionada.
SelectedIndex	Índice da coluna selecionada.
SelectedRows	Lista do tipo TBookmarkList, com os registros selecionados em uma Grid com MultiSelect.
TitleFont	Fonte do título das colunas.

Eventos	Descrição
OnCellClick	Ao clicar em uma célula da Grid.
OnColEnter	Quando uma célula de alguma coluna da Grid recebe o foco.
OnColExit	Quando uma célula de alguma coluna perde o foco.
OnColumnMoved	Quando do usuário mover uma coluna.
OnDrawDataCell	Evento usado para personalizar a forma de desenhar os dados que são apresentados na Grid.
OnEditButtonClick	Ao clicar no botão de edição de uma célula, mostrado pela propriedade ButtonStyle da coluna.
OnTitleClick	Ao clicar no título das colunas.

### DBText A, DBEdit , DBListBox , DBComboBox , DBRichEdit

Controles genéricos ligados a um campo de uma tabela.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado.
DataSource	DataSource que contém os valores a serem manipulados.

### DBImage

É usado para mostrar uma figura armazenada em um campo BLOB.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado.
DataSource	DataSource que contém os valores a serem manipulados.

### **DBMemo**

É usado para permitir que o usuário veja e modifique um grande campo de texto, eventualmente armazenado em um campo Memo ou BLOB (Grande Objeto Binário).

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado.
DataSource	DataSource que contém os valores a serem manipulados.

### **DBRadioGroup**

Mostra algumas opções para o preenchimento de um campo.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado.
DataSource	DataSource que contém os valores a serem manipulados.
Values	Valor a ser armazenado para cada botão de rádio.

### **DBLookupListBox** , **DBLookupComboBox**

Preenche um campo com dados contidos em outra tabela. Se o campo mostrado nesses componentes for um campo Lookup, você só precisa especificar as propriedades: DataSource e DataField.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado.
DataSource	DataSource que contém os valores a serem manipulados.
ListSource	DataSource que contém os valores a serem exibidos na lista.
ListField	Campo de ListSource que será exibido.
KeyField	Campo de ListSource usado no relacionamento.
KeyValue	Valor do campo de ListSource usado no relacionamento.

### **DBChart**

É um componente gráfico onde os dados são oriundos de um DataSet.

## **5.1. Conhecendo alguns Componentes da palheta Jv Data Controls**



A seguir encontra-se uma breve descrição dos principais componentes da palheta.

## JvDBSpinEdit

Utilizado normalmente para realizar incremento em dados numéricos.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado
DataSource	Especifica o componente da origem dos dados
MinValue	Valor mínimo
MaxValue	Valor máximo
Increment	Incremento dado ao valor do componente
ArrowKeys	Indica que o componente recebe os incrementos das SETAS
Thousands	Indica se irá aparecer o separador de milhar
Value	Indica o valor o valor do componente
ValueType	Especifica o tipo do valor do controle

## JvDBCalcEdit

Controle de edição com uma tecla para exibir uma calculadora.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado
DataSource	Especifica o componente da origem dos dados de onde se extrairá valores para os campos
ButtonHint	Texto explicativo que será exibido para o usuário quando ele pousar o mouse sobre o botão
MinValue	Valor mínimo
MaxValue	Valor máximo
DecimalPlaceRound	Especifica se o valor real será arredondado ou não
DecimalPlaces	Especifica o número de casas decimais
Glyph	Especifica a figura bitmap que aparece no botão
ImageKind	Especifica o tipo da imagem do botão

## JvDBDateTimePicker

Componente que permite a seleção visual de uma data ou hora.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado
DataSource	Especifica o componente da origem dos dados de onde se extrairá valores para os campos
Time	Hora selecionada
Date	Data Selecionada
DateMode	Forma de seleção da data [dmComboBox, dmUpDown]
DateFormat	Formato da Data [dfShort, dfLong]
Kind	Seleciona o componente para Data ou Hora [dtkDate, dtkTime]
ShowCheckbox	Exibe um CheckBox
Check	Indica se o CheckBox esta selecionado

## JvDBDateEdit

Controle de edição com um botão para exibir um calendário.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado
DataSource	Especifica o componente da origem dos dados de onde se extrairá valores para os campos
ButtonHint	Texto explicativo que será exibido para o usuário quando ele pousar o mouse sobre o botão
CalendarHints	Exibe um texto explicativo para os botões: Ano anterior, Mês anterior, Mês posterior e Ano posterior
Glyph	Especifica a figura bitmap que aparece no botão
ImageKind	Especifica o tipo da imagem do botão
MinDate	Data mínima
MaxDate	Data máxima
StartOfWeek	Especifica o primeiro dia da semana
WeekendColor	Especifica a cor do texto do(s) dia(s) do fim de semana.
Weekends	Especifica os dias do fim de semana
YearDigits	Especifica o número dos dígitos usados para o ano

## JvDBStatusLabel

Sua função é mostrar detalhes dos registros como: Estado do DataSet, posição do registro a atual, quantidade total de registros e tamanho do registro.

Propriedades	Descrição
DataSource	Especifica o componente da origem dos dados de onde se extrairá valores para os campos
Captions	Exibe um texto explicativo para cada um dos seguintes estados (state) do dataset: Inativo (dsInactive), Navegação (dsBrowse), Alteração (dsEdit) e Inclusão (dsInsert)
DataSetName	Nome legível do dataset para exibição.
GlyphAlign	Posição da imagem que representa o estado do dataset
ShowOptions	Especifica o conteúdo a ser exibido. Conteúdo que pode ser: texto (DataSetName), imagem ou ambos.
Style	Especifica o estilo. Estilo que pode ser: estado do dataset (state), número do registro corrente ou o tamanho do registro corrente

## JvDBFindEdit

Tem como função realizar o filtro em uma série de dados.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado.
DataSource	Especifica o componente da origem dos dados.
FindMode	Especifica o modo de procura. Modo que pode ser: Primeira Posição ou Qualquer Posição
FindStyle	Especifica o estilo da procura, que pode ser: Filtro ou Navegação

## JvDBSearchComboBox

Caixa combo para navegação em uma série de dados.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado
DataSource	Especifica o componente da origem dos dados de onde se extrairá valores para os campos
Style	Determina como os itens serão exibidos

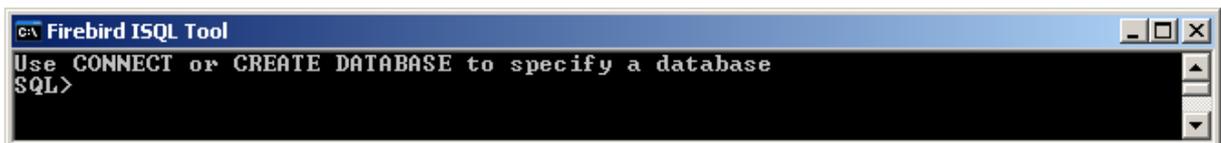
## JvDBSearchEdit

Realiza a navegação nos dados, à medida que o dado for sendo digitado.

Propriedades	Descrição
DataField	Campo ao qual o controle está ligado
DataSource	Especifica o componente da origem dos dados de onde se extrairá valores para os campos
SearchOptions	Especifica as opções de pesquisa, que podem ser: diferenciar maiúsculas de minúsculas e parciais

## 6. Criando o Banco da Apostila

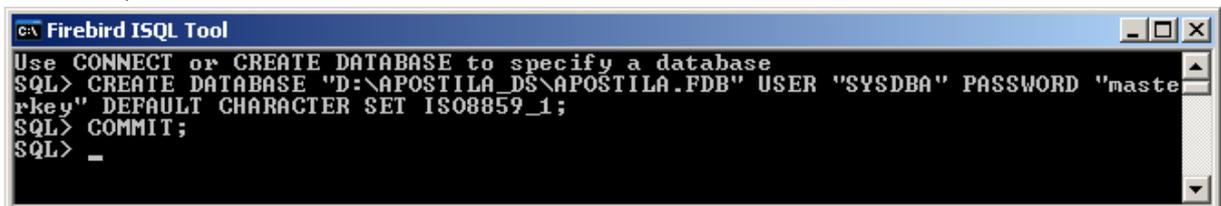
Iniciar / Todos os programas / Firebird 2.0 / Firebird ISQL Tool.



```
c:\ Firebird ISQL Tool
Use CONNECT or CREATE DATABASE to specify a database
SQL>
```

Criando o arquivo do banco de dados:

```
CREATE DATABASE "D:\APOSTILA_DS\APOSTILA.FDB" USER "SYSDBA"
PASSWORD "masterkey" DEFAULT CHARACTER SET ISO8859_1;
COMMIT;
```



```
c:\ Firebird ISQL Tool
Use CONNECT or CREATE DATABASE to specify a database
SQL> CREATE DATABASE "D:\APOSTILA_DS\APOSTILA.FDB" USER "SYSDBA" PASSWORD "maste
rkey" DEFAULT CHARACTER SET ISO8859_1;
SQL> COMMIT;
SQL> _
```

**Nota:** Se o banco de dados já existir, será necessário estabelecer uma conexão com o mesmo. Estabelece-se uma conexão com o banco de dados através do seguinte comando:

```
CONNECT "D:\APOSTILA_DS\APOSTILA.FDB" USER "SYSDBA" PASSWORD
"masterkey";
```

```

c:\ Firebird ISQL Tool
Use CONNECT or CREATE DATABASE to specify a database
SQL> CREATE DATABASE "D:\APOSTILA_DS\APOSTILA.FDB" USER "SYSDBA" PASSWORD "masterkey"
DEFAULT CHARACTER SET ISO8859_1;
SQL> COMMIT;
SQL> CONNECT "D:\APOSTILA_DS\APOSTILA.FDB" USER "SYSDBA" PASSWORD "masterkey";
Database: "D:\APOSTILA_DS\APOSTILA.FDB", User: SYSDBA
SQL>

```

### Criando a tabela de funcionários:

```

CREATE TABLE FUNCIONARIO (
    FUN_CODIGO INTEGER NOT NULL,
    FUN_NOME VARCHAR(40) NOT NULL COLLATE PT_BR,
    FUN_IDADE INTEGER NOT NULL,
    FUN_FOTO BLOB SUB_TYPE 0,
    CONSTRAINT PK_FUNCIONARIO PRIMARY KEY (FUN_CODIGO),
    CONSTRAINT IX_FUN_NOME UNIQUE (FUN_NOME)
);
COMMIT;

```

```

c:\ Firebird ISQL Tool
Use CONNECT or CREATE DATABASE to specify a database
SQL> CONNECT "D:\APOSTILA_DS\APOSTILA.FDB" USER "SYSDBA" PASSWORD "masterkey";
Database: "D:\APOSTILA_DS\APOSTILA.FDB", User: SYSDBA
SQL> CREATE TABLE FUNCIONARIO (FUN_CODIGO INTEGER NOT NULL, FUN_NOME VARCHAR(40)
NOT NULL COLLATE PT_BR, FUN_IDADE INTEGER NOT NULL, FUN_FOTO BLOB SUB_TYPE 0, C
ONSTRAINT PK_FUNCIONARIO PRIMARY KEY (FUN_CODIGO), CONSTRAINT IX_FUN_NOME UNIQUE
(FUN_NOME));
SQL> COMMIT;
SQL>

```

### Mostrando a estrutura da tabela de funcionários:

```
SHOW TABLE FUNCIONARIO;
```

```

c:\ Firebird ISQL Tool
Use CONNECT or CREATE DATABASE to specify a database
SQL> CONNECT "D:\APOSTILA_DS\APOSTILA.FDB" USER "SYSDBA" PASSWORD "masterkey";
Database: "D:\APOSTILA_DS\APOSTILA.FDB", User: SYSDBA
SQL> SHOW TABLE FUNCIONARIO;
FUN_CODIGO          INTEGER Not Null
FUN_NOME            VARCHAR(40) CHARACTER SET ISO8859_1 Not Null
                   COLLATE PT_BR
FUN_IDADE           INTEGER Not Null
FUN_FOTO            BLOB segment 80, subtype BINARY CHARACTER SET NO
NE Nullable
CONSTRAINT PK_FUNCIONARIO:
  Primary key (FUN_CODIGO)
CONSTRAINT IX_FUN_NOME:
  Unique key (FUN_NOME)
SQL> _

```

### Criando a tabela de dependentes:

```

CREATE TABLE DEPENDENTE (
    DEP_CODIGO INTEGER NOT NULL,
    DEP_NOME VARCHAR(40) NOT NULL COLLATE PT_BR,
    DEP_IDADE SMALLINT NOT NULL,
    FUN_CODIGO INTEGER NOT NULL,
    CONSTRAINT PK_DEPENDENTE PRIMARY KEY (DEP_CODIGO, FUN_CODIGO),
    CONSTRAINT IX_DEP_NOME UNIQUE (DEP_NOME),
    CONSTRAINT FK_DEPENDENTE_FUNCIONARIO
        FOREIGN KEY (FUN_CODIGO) REFERENCES FUNCIONARIO
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
COMMIT;

```

```

Firebird ISQL Tool
FUN_NOME          VARCHAR(40) CHARACTER SET ISO8859_1 Not Null
                  COLLATE PT_BR
FUN_IDADE         INTEGER Not Null
FUN_FOTO         BLOB segment 80, subtype BINARY CHARACTER SET NO
NE Nullable
CONSTRAINT PK_FUNCIONARIO:
  Primary key (FUN_CODIGO)
CONSTRAINT IX_FUN_NOME:
  Unique key (FUN_NOME)
SQL> CREATE TABLE DEPENDENTE (DEP_CODIGO INTEGER NOT NULL, DEP_NOME VARCHAR(40)
NOT NULL COLLATE PT_BR, DEP_IDADE SMALLINT NOT NULL, FUN_CODIGO INTEGER NOT NULL
, CONSTRAINT PK_DEPENDENTE PRIMARY KEY (DEP_CODIGO, FUN_CODIGO), CONSTRAINT IX_D
EP_NOME UNIQUE (DEP_NOME), CONSTRAINT FK_DEPENDENTE_FUNCIONARIO FOREIGN KEY (FUN
_CODIGO) REFERENCES FUNCIONARIO ON DELETE CASCADE ON UPDATE CASCADE);
SQL> COMMIT;
SQL>

```

Mostrando a estrutura da tabela de dependentes:

SHOW TABLE DEPENDENTE;

```

Firebird ISQL Tool
_CODIGO) REFERENCES FUNCIONARIO ON DELETE CASCADE ON UPDATE CASCADE);
SQL> COMMIT;
SQL> SHOW TABLE DEPENDENTE;
DEP_CODIGO          INTEGER Not Null
DEP_NOME            VARCHAR(40) CHARACTER SET ISO8859_1 Not Null
                  COLLATE PT_BR
DEP_IDADE           SMALLINT Not Null
FUN_CODIGO          INTEGER Not Null
CONSTRAINT FK_DEPENDENTE_FUNCIONARIO:
  Foreign key (FUN_CODIGO) References FUNCIONARIO (FUN_CODIGO) On Update Casc
ade On Delete Cascade
CONSTRAINT PK_DEPENDENTE:
  Primary key (DEP_CODIGO, FUN_CODIGO)
CONSTRAINT IX_DEP_NOME:
  Unique key (DEP_NOME)
SQL>

```

Listando todas as tabelas do banco apostila:

SHOW TABLES;

```

Firebird ISQL Tool
DEP_CODIGO          INTEGER Not Null
DEP_NOME            VARCHAR(40) CHARACTER SET ISO8859_1 Not Null
                  COLLATE PT_BR
DEP_IDADE           SMALLINT Not Null
FUN_CODIGO          INTEGER Not Null
CONSTRAINT FK_DEPENDENTE_FUNCIONARIO:
  Foreign key (FUN_CODIGO) References FUNCIONARIO (FUN_CODIGO) On Update Casc
ade On Delete Cascade
CONSTRAINT PK_DEPENDENTE:
  Primary key (DEP_CODIGO, FUN_CODIGO)
CONSTRAINT IX_DEP_NOME:
  Unique key (DEP_NOME)
SQL> SHOW TABLES;
DEPENDENTE          FUNCIONARIO
SQL>

```

Saindo da ferramenta Firebird ISQL Tool:

EXIT;

```

Firebird ISQL Tool
DEP_CODIGO          INTEGER Not Null
DEP_NOME            VARCHAR(40) CHARACTER SET ISO8859_1 Not Null
                  COLLATE PT_BR
DEP_IDADE           SMALLINT Not Null
FUN_CODIGO          INTEGER Not Null
CONSTRAINT FK_DEPENDENTE_FUNCIONARIO:
  Foreign key (FUN_CODIGO) References FUNCIONARIO (FUN_CODIGO) On Update Casc
ade On Delete Cascade
CONSTRAINT PK_DEPENDENTE:
  Primary key (DEP_CODIGO, FUN_CODIGO)
CONSTRAINT IX_DEP_NOME:
  Unique key (DEP_NOME)
SQL> SHOW TABLES;
DEPENDENTE          FUNCIONARIO
SQL> EXIT;

```

## 7. Usando os componentes MDOQuery e MDOUpdateSQL

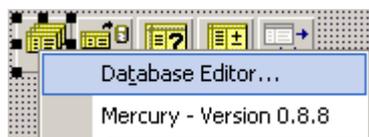
Altere as seguintes propriedades do Form:

**Caption:** Cadastro de Funcionários

**Name:** frmCadFuncionario

Primeiramente vamos instanciar no Form um MDODatabase, um MDOTransaction, um MDOQuery, um MDOUpdateSQL (todos da paleta Mercury) e um componente DataSource (paleta Data Access).

Agora vamos configurar o componente **MDODatabase**. Clique com o botão direito do mouse sobre o componente e selecione a opção **Database Editor** como descrito na figura abaixo.



Na opção **Database Editor** vamos configurar os seguintes tópicos:

**Connection:** Local

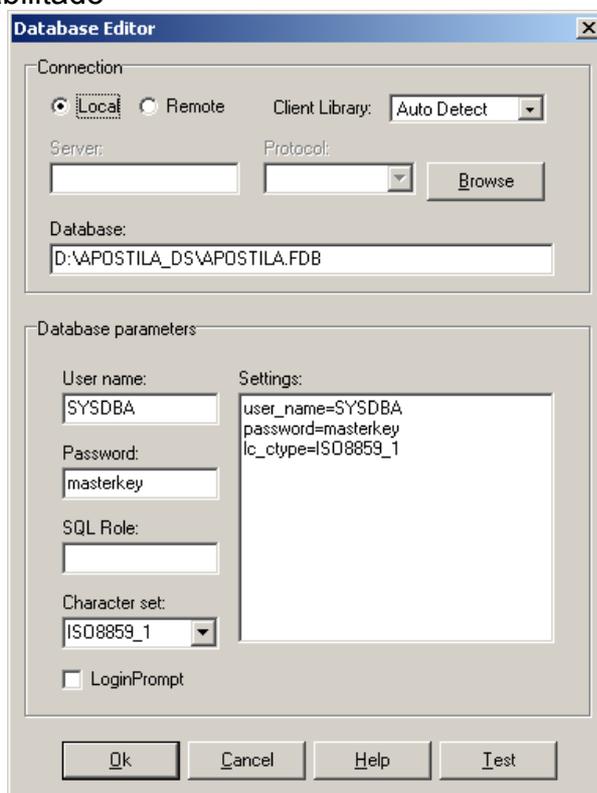
**Database:** APOSTILA.FDB. Criado anteriormente no diretório D:\APOSTILA\_DS\APOSTILA.FDB

**User Name:** SYSDBA

**Password:** masterkey

**Character set:** ISO8859\_1

**Login Prompt:** Desabilitado



Já podemos dar um **OK** e testar se a conexão está correta. Para isso, mude o a propriedade **Connected** do componente **MDODatabase** para **True**.

Connected  True

Agora vamos ligar o componente **MDOTransaction** ao **MDODatabase**. Vá à propriedade **DefaultDatabase** do **MDOTransaction** e coloque o **MDODatabase1**.

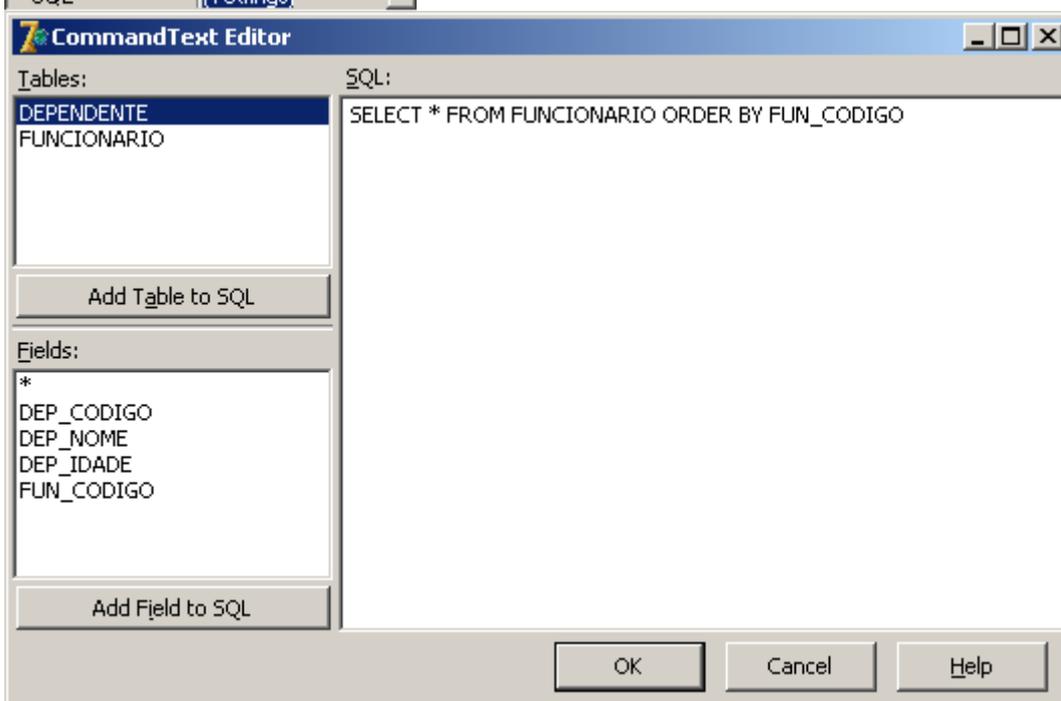
DefaultDatabase  MDODatabase1

Você também deve para **True** as propriedades: **Active** e **AutoCommit**.

Active  True  
AutoCommit  True

Para ligar o componente **MDOQuery1** ao banco, você deve colocar o **MDODatabase1** na propriedade **Database**. Note que a propriedade **Transaction** automaticamente traz o **MDOTransaction1** devido à ligação que nós fizemos agora a pouco. Na propriedade **SQL**, insira `SELECT * FROM FUNCIONARIO ORDER BY FUN_CODIGO`.

Database  MDODatabase1  
SQL



O componente **DataSource1** deve ser ligado ao **MDOQuery1**. Para isto, altere a propriedade **DataSet** de forma que fique **MDOQuery1**.

DataSet  MDOQuery1

Vamos instanciar agora um **DBGrid** e um **DBNavigator** (paleta **Data Controls**). A propriedade **DataSource** do **DBGrid** e do **DBNavigator** devem apontar para o **DataSource1**.

DataSource  DataSource1

Instancie agora um componente **OpenPictureDialog** (paleta **Dialogs**) e configure as seguintes propriedades:

Filter: Todas as fotos (\*.jpg;\*.jpeg;\*.bmp)|\*.jpg;\*.jpeg;\*.bmp|Foto JPEG (\*.jpg;\*.jpeg)|\*.jpg;\*.jpeg|Foto Bitmap (\*.bmp)|\*.bmp

Filter

InitialDir: . (ponto)

InitialDir

Você deve agora instanciar o componente **DBImage** (paleta **Data Controls**) e configure as seguintes propriedades:

DataSource: DataSource1

DataSource DataSource1

DataField: FUN\_FOTO

DataField FUN\_FOTO

Stretch: True

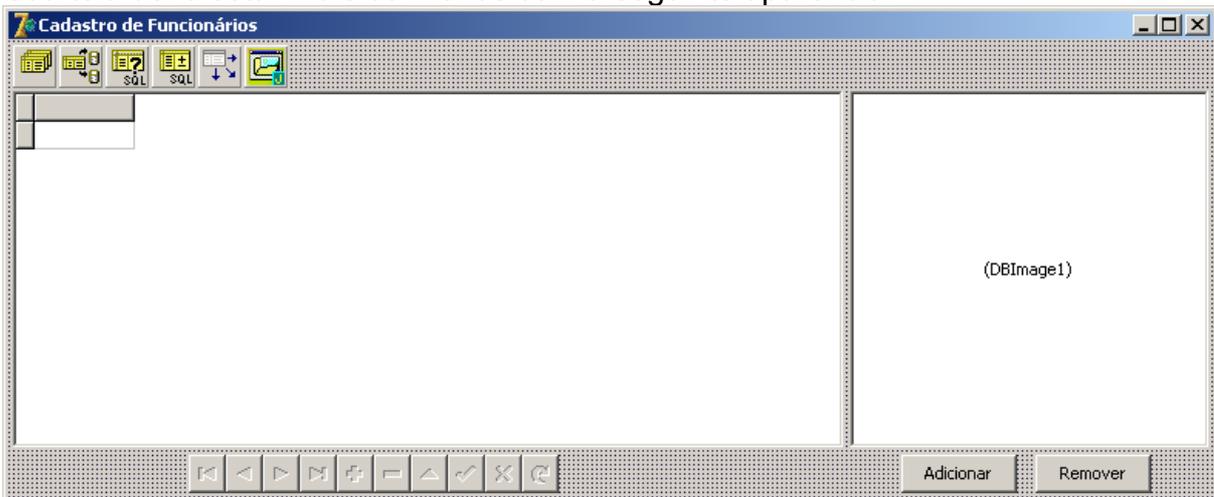
Stretch  True

Instancie dois componentes **Button** (paleta **Standard**) e configure a propriedade **Caption** do primeiro para **Adicionar** e do segundo para **Remover**.

Caption Adicionar

Caption Remover

Sua tela deve estar mais ou menos com a seguinte aparência:



Dê um duplo clique sobre o botão chamado **Adicionar** e digite as linhas em negrito do código abaixo:

```
procedure TfrmCadFuncionarios.Button1Click(Sender: TObject);
```

```
var
```

```
    strArquivo: String;
```

```
    imagem: TPicture;
```

```
begin
```

```
    //Não esquecer de declarar na seção uses: Clipbrd, Jpeg
```

```
    if (OpenPictureDialog1.Execute) then
```

```
    begin
```

```
        strArquivo := OpenPictureDialog1.FileName;
```

```
        //Cria a variável para armazenar a imagem
```

```
        imagem := TPicture.Create();
```

```
        //Busca a imagem no arquivo e joga na variável
```

```
        imagem.LoadFromFile(strArquivo);
```

```
        //Passa imagem para o clipboard
```

```
        clipboard.Assign(imagem);
```

```
        //Copia a imagem para o DBImage
```

```
        DBImage1.PasteFromClipboard;
```

```
        //Libera a variável
```

```
        imagem.Free;
```

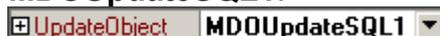
```
    end;
```

```
end;
```

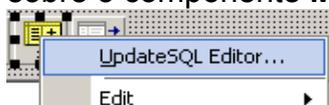
Você precisa fazer a mesma coisa para o botão chamado **Remove**:

```
procedure TfrmCadFuncionarios.Button2Click(Sender: TObject);  
begin  
    DBImage1.Picture.Bitmap.Assign(nil);  
end;
```

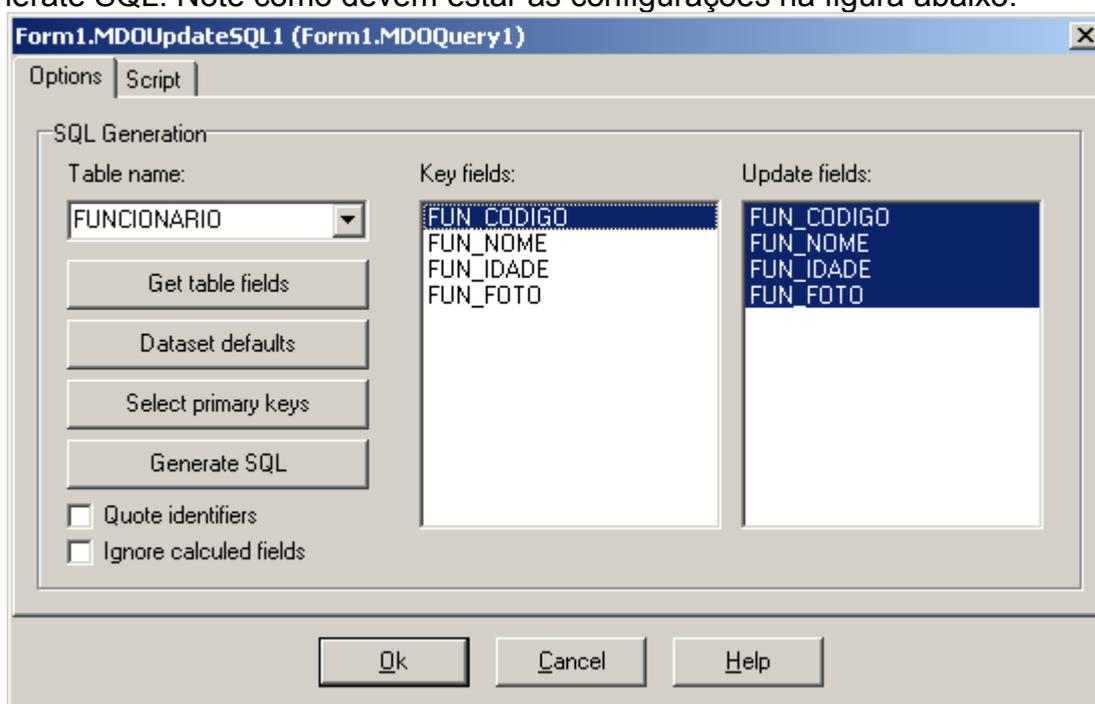
Agora vamos fazer a conexão do componente **MDOUpdateSQL**. Vá até a propriedade **UpdateObject** do componente **MDOQuery1** e aponte para **MDOUpdateSQL1**.



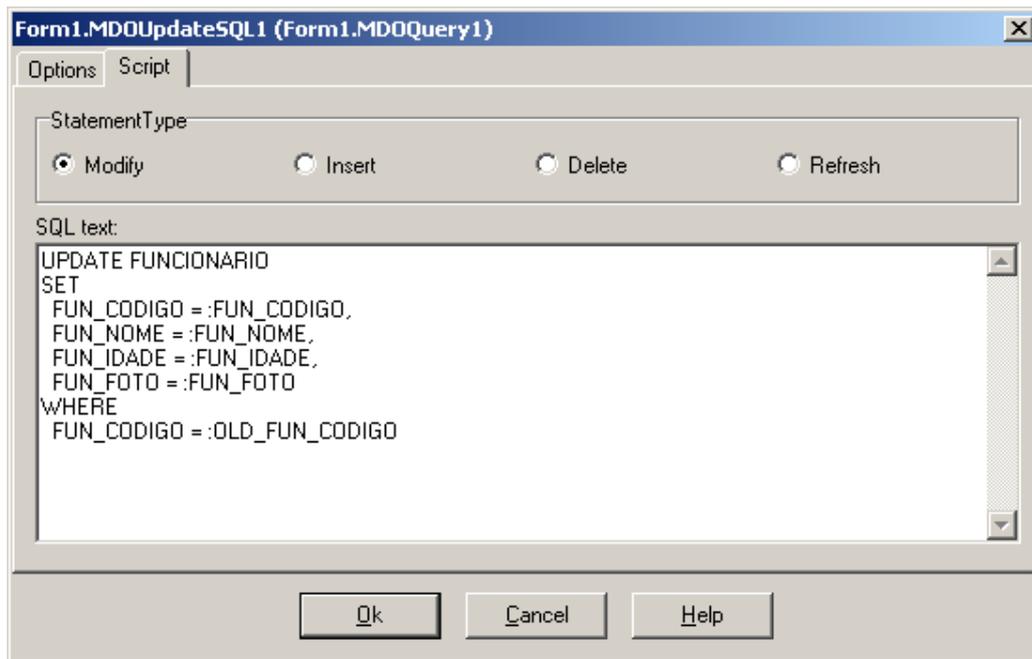
Vamos às configurações do **MDOUpdateSQL**. Clique com o botão direito do mouse sobre o componente **MDOUpdateSQL** e selecione a opção **UpdateSQL Editor**.



Na coluna **Key Fields** você deve deixar marcado somente o(s) **campo(s) chave da tabela** (no nosso exemplo a chave primária da tabela é o campo **FUN\_CODIGO**). No **Update Fields** você deve deixar **todos marcados**. Agora é só você clicar na opção **Generate SQL**. Note como devem estar às configurações na figura abaixo.



Todas as queries de inclusão, exclusão, alteração e atualização da tabela já foram geradas. Clicando nas opções **Modify**, **Insert**, **Update** e **Refresh** você vai notar que todas as queries foram preenchidas automaticamente.



Agora é só você configurar a propriedade **Active** do **MDOQuery1** para **True** e rodar o aplicativo. A partir de agora você pode incluir, excluir ou alterar dados da tabela.

Salve esta aplicação seguindo as seguintes orientações:

- Pasta / Diretório: **D:\APOSTILA\_DS**;
- **untCadFuncionario.pas** – para a **Unit**;
- **Funcionario.dpr** – para o **Project**.

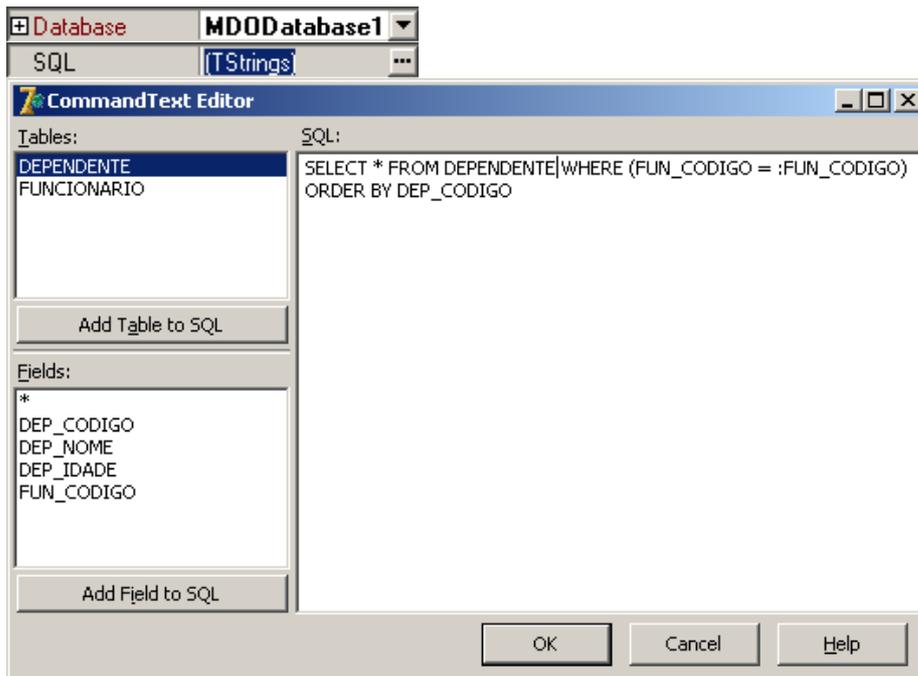
## 8. Mestre / Detalhe com MDOQuery e MDOUpdateSQL

Nos relacionamento de 1 para N, uma tabela pode estar ligada a outra em uma relação Mestre / Detalhe, nesse tipo de relação os registros da tabela de ordem N são filtrados pelo campo de relacionamento com a tabela de ordem 1. Por exemplo, se o relacionamento de Funcionários com Dependentes for mestre / detalhe, só serão acessados em dependentes, os registros cujo campo FUN\_CODIGO seja igual ao FUN\_CODIGO da tabela Funcionários.

Abra o projeto Funcionario.dpr, salvo no capítulo anterior, e proceda as mudanças solicitadas.

Vamos instanciar no Form um **MDOQuery**, um **MDOUpdateSQL** (todos da paleta **Mercury**) e um componente **DataSource** (paleta **Data Access**).

Para ligar o componente **MDOQuery2** ao banco, você deve colocar o **MDODatabase1** na propriedade **Database**. Note que a propriedade **Transaction** automaticamente traz o **MDOTransaction1**. Na propriedade **SQL**, insira *SELECT \* FROM DEPENDENTE WHERE (FUN\_CODIGO = :FUN\_CODIGO) ORDER BY DEP\_CODIGO*.



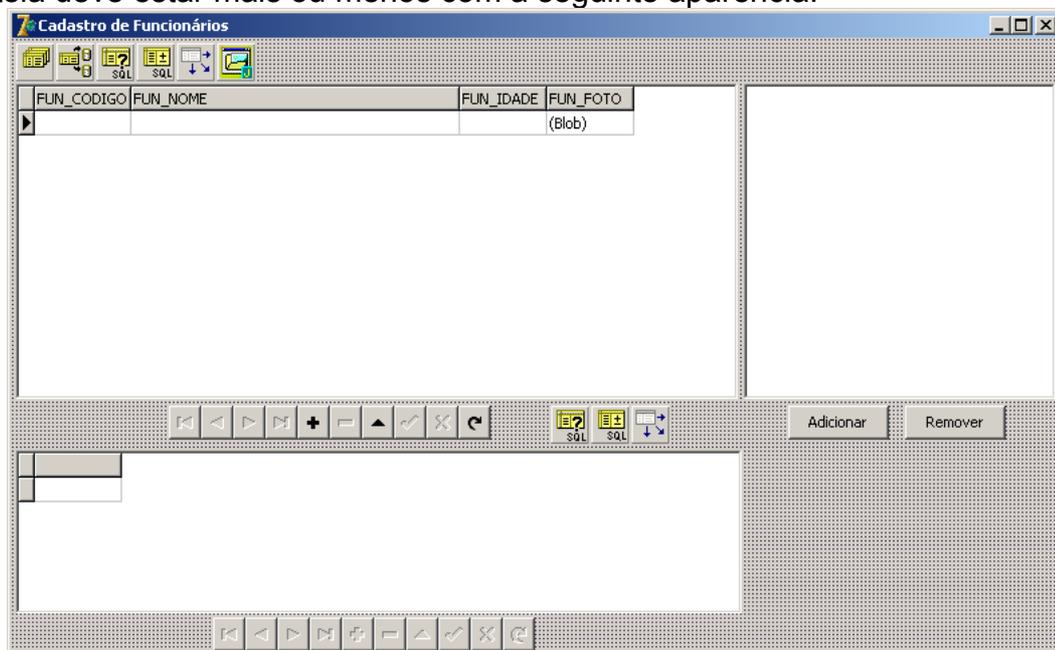
O componente **DataSource2** deve ser ligado ao **MDOQuery2**. Para isto, altere a propriedade **DataSet** de forma que fique **MDOQuery2**.

**DataSet** **MDOQuery2**

Vamos instanciar agora um **DBGrid** e um **DBNavigator** (paleta **Data Controls**). A propriedade **DataSource** do **DBGrid** e do **DBNavigator** devem apontar para o **DataSource2**.

**DataSource** **DataSource2**

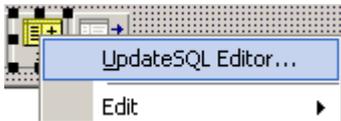
Sua tela deve estar mais ou menos com a seguinte aparência:



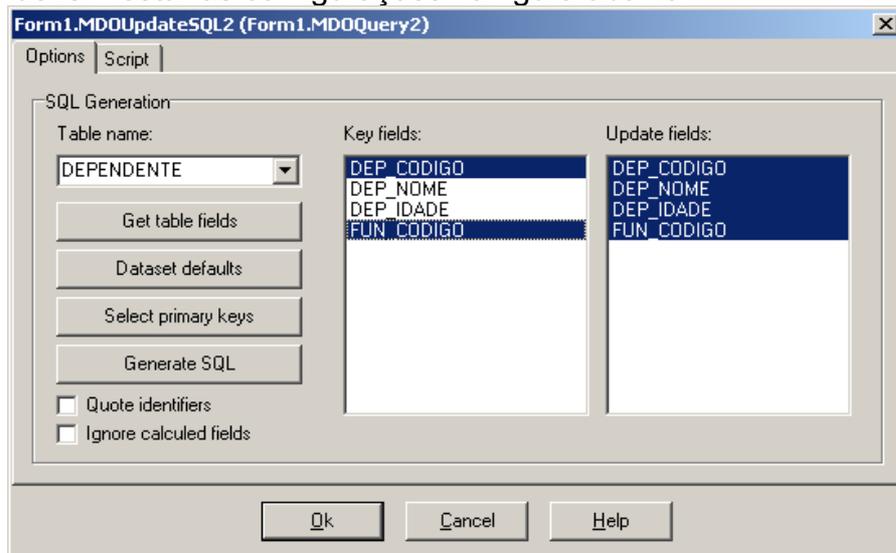
Agora vamos fazer a conexão do componente **MDOUpdateSQL2**. Vá até a propriedade **UpdateObject** do componente **MDOQuery2** e aponte para **MDOUpdateSQL2**.

**UpdateObject** **MDOUpdateSQL2**

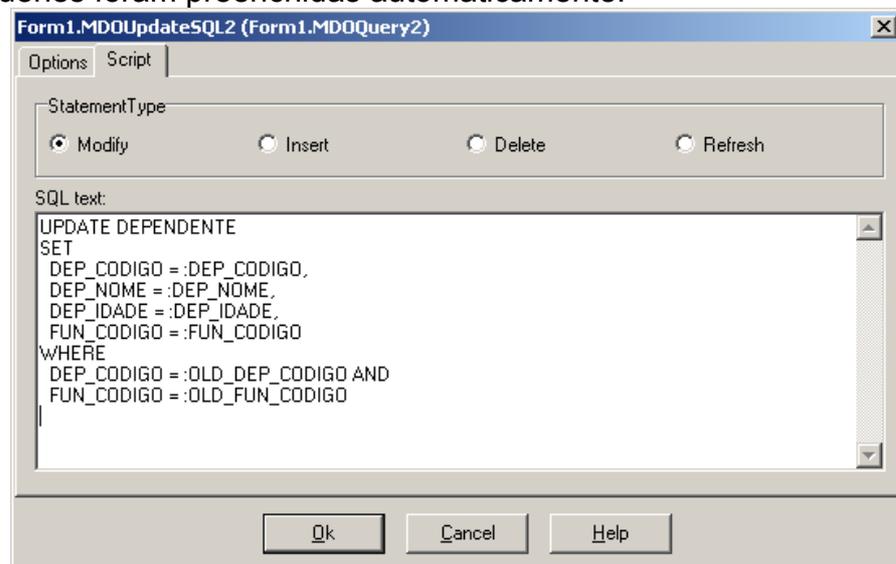
Agora vamos às configurações do **MDOUpdateSQL2**. Clique com o botão direito do mouse sobre o componente **MDOUpdateSQL2** e selecione a opção **UpdateSQL Editor**.



Na coluna **Key Fields** você deve deixar marcado somente o(s) **campo(s) chave da tabela** (no nosso exemplo a chave primária da tabela são os campos DE\_CODIGO e FU\_CODIGO). No **Update Fields** você deve deixar **todos marcados**. Para isso clique no botão **Dataset defaults**. Agora é só você clicar no botão **Generate SQL**. Note como devem estar às configurações na figura abaixo.



Todas as queries de inclusão, exclusão, alteração e atualização da tabela já foram geradas. Clicando nas opções Modify, Insert, Update e Refresh você vai notar que todas as queries foram preenchidas automaticamente.



Para efetivar o relacionamento mestre / detalhe entre as tabelas: Funcionários e Dependentes, configure a propriedade **DataSource** do **MDOQuery2** para **DataSource1**.



No **Form** selecione o componente **MDOQuery2**, no **Object Inspector** selecione a aba **Events**, agora selecione evento **OnNewRecord**, dê um duplo clique no retângulo vazio a direita deste evento e digite as linhas em negrito do código abaixo:

```
procedure TfrmCadFuncionarios.MDOQuery2NewRecord(DataSet: TDataSet);
begin
    MDOQuery2.FieldByName('FUN_CODIGO').AsInteger :=
    MDOQuery1.FieldByName('FUN_CODIGO').AsInteger;
end;
```

Este código é responsável por passar o valor do campo FUN\_CODIGO da tabela de Funcionários para a tabela de Dependentes, mantendo desta forma o relacionamento.

Agora é só você configurar a propriedade **Active** do **MDOQuery2** para **True** e rodar o aplicativo. A partir de agora você pode incluir, excluir ou alterar dados da tabela.

**Nota:** Não se esqueça de salvar a aplicação, para isso basta clicar em Save All .

## 9. Criando Gráficos

Hoje os sistemas de informação são baseados em bancos de dados e softwares, tornando a representação gráfica das informações de suma importância. Pois a representação gráfica fornece subsídios para tomada de decisões, projeção financeira, metas, etc.

Abra o projeto Funcionario.dpr e proceda as mudanças solicitadas.

Instancie um componente **MDOQuery** (paleta **Mercury**) e configure as seguintes propriedades:

**Database:** MDODatabase1;

**SQL:** SELECT DEP\_IDADE AS Idade, COUNT(DEP\_IDADE) AS Total FROM DEPENDENTE GROUP BY DEP\_IDADE ORDER BY DEP\_IDADE;

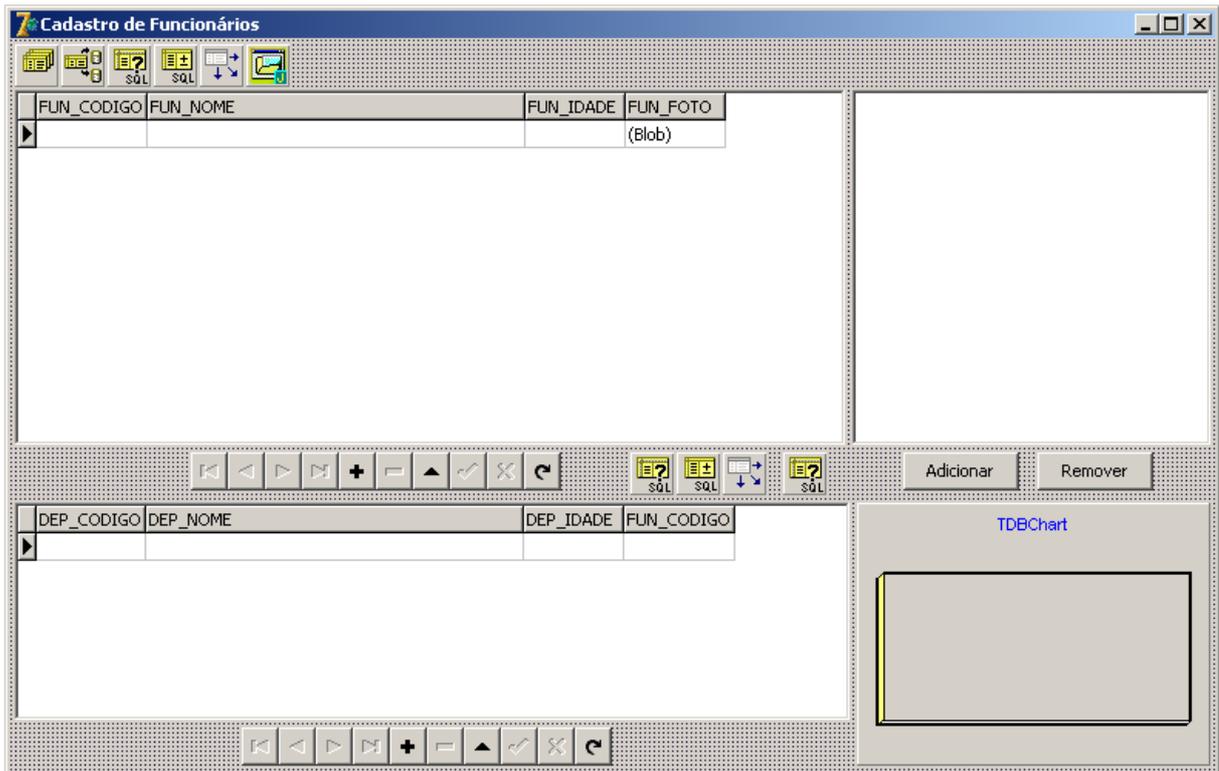
**Active:** True.

Instancie um componente **DBChart** (paleta **Data Controls**) e configure as seguintes propriedades:

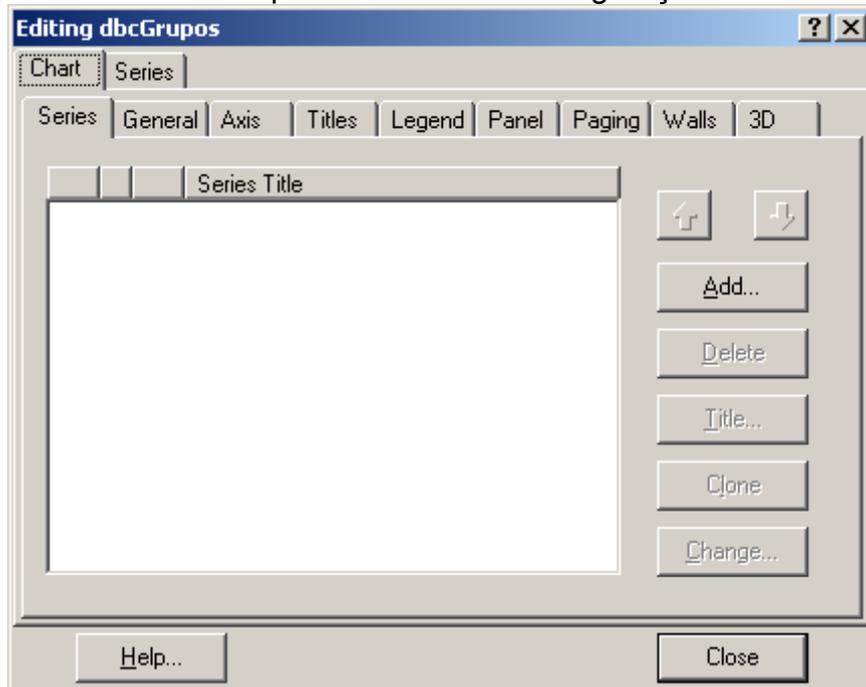
**Height:** 205;

**Width:** 226.

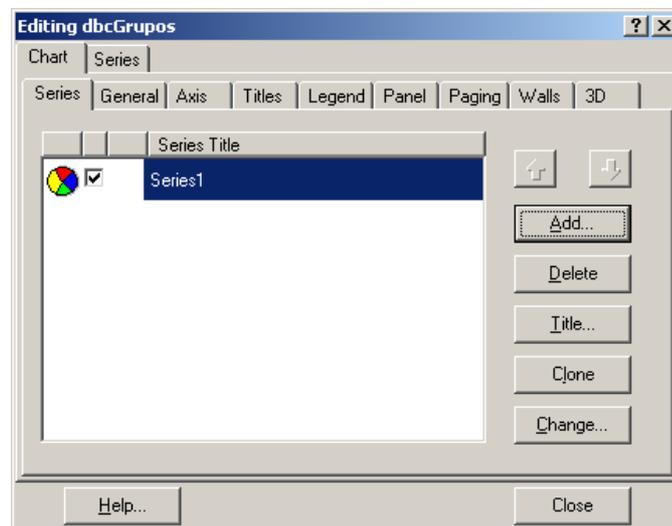
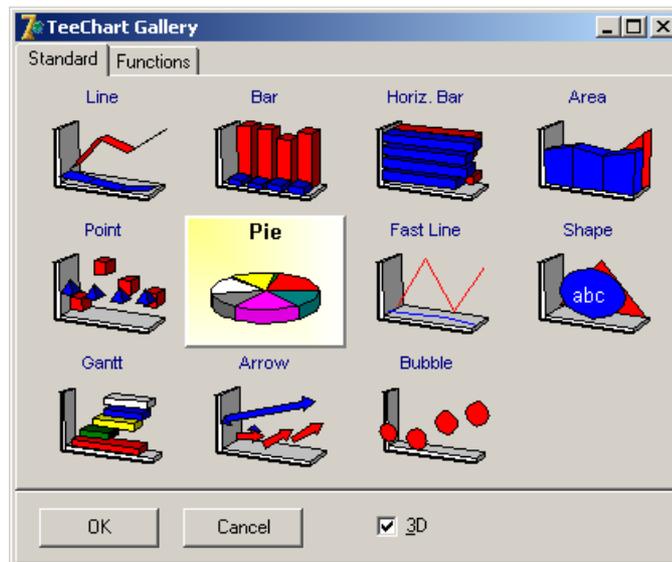
O Form deverá estar parecido com este:



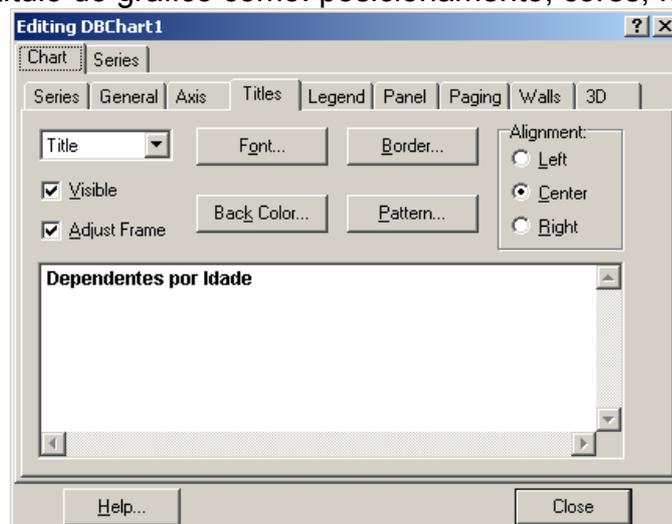
Vamos continuar a configuração do componente **DBChart**. Dê um duplo clique sobre o mesmo, para abrir o wizard que irá auxiliar na configuração do DBChart.



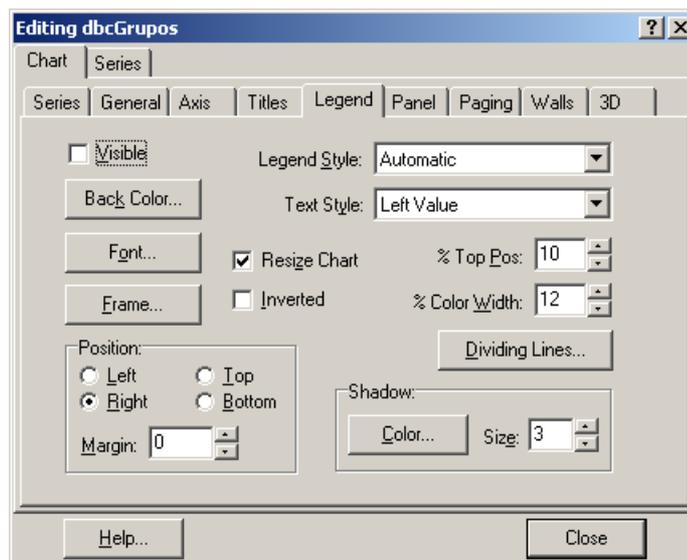
Na aba **Series**, definiremos o tipo de gráfico a ser gerado, clique no botão **Add...** e selecione o **tipo de gráfico Pie**.



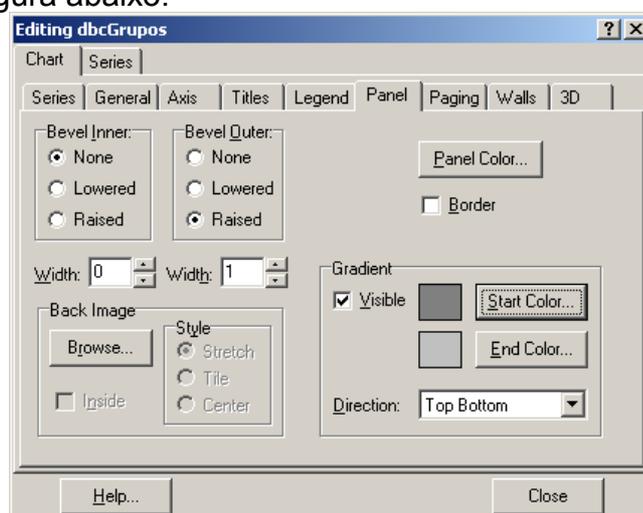
Abra a aba **Titles** e defina o título do gráfico. Nesta etapa podemos definir as configurações do título do gráfico como: posicionamento, cores, fontes e ambiente.



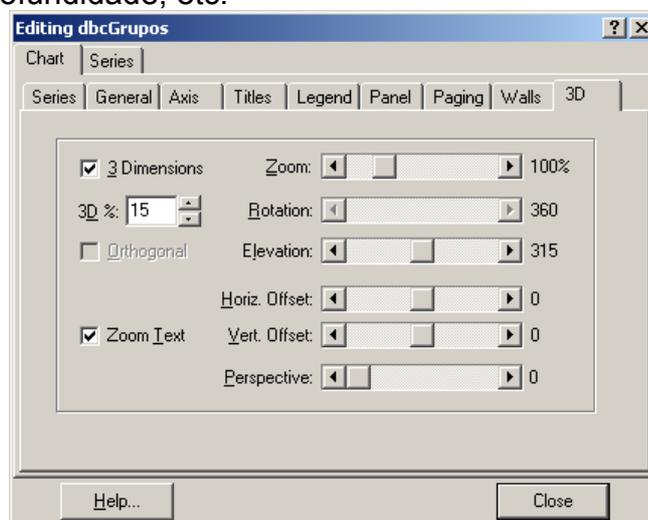
Na aba **Legend** estão todas as configurações da legenda, **desative** a opção **Visible**.



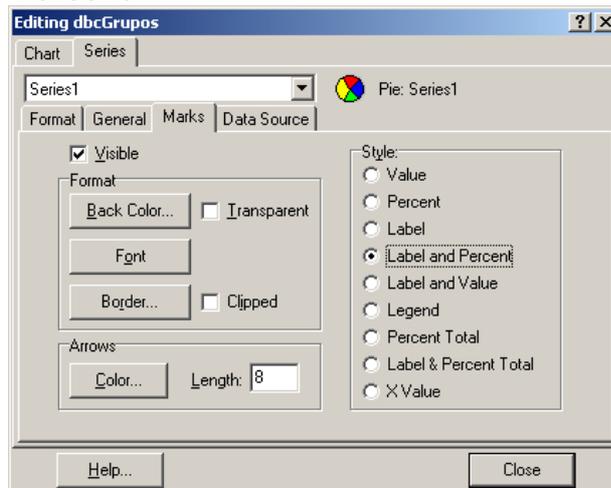
Na aba **Panel** estão às configurações de fundo do gráfico como o preenchimento gradiente. Veja a figura abaixo:



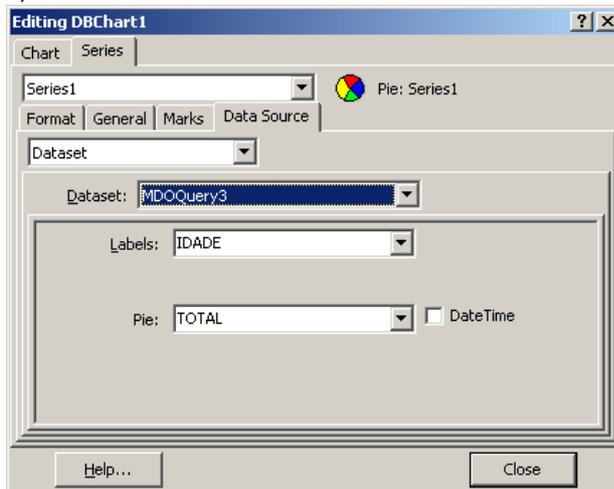
Na aba **3D** estão às configurações relacionadas ao gráfico como distância, posicionamento, profundidade, etc.



Abra a aba **Series** (superior) e em seguida a sub-aba **Marks** e no grupo **Style** selecione **Label and Percent**.



Abra a aba **Data Source**, nesta aba vamos configurar a fonte de dados para a geração do gráfico. Defina no combobox a fonte de dados **DataSet**, em **DataSet** defina o **MDOQuery3**, em **Label** defina **Idade** e em **Pie** defina **Total**.



Agora feche o **Editing DBChart**, clicando no botão **Close**;  
No **Object Inspector**, selecione o formulário **MDOQuery2**;



Agora selecione a aba **Events** e em seguida o evento **AfterPost**;

Dê um duplo-clique no retângulo em branco a direita do evento AfterPost e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TForm1.MDOQuery2AfterPost(DataSet: TDataSet);
begin
    MDOQuery3.Active := False;
    MDOQuery3.Active := True;
end;
```

Agora, no **Object Inspector**, selecione o evento **AfterDelete** e no retângulo em branco a direita selecione **MDOQuery2AfterPost**, desta forma estamos fazendo com que o evento **AfterDelete** utilize o código escrito para o evento **AfterPost**;

**Nota:** Não se esqueça de salvar a aplicação, para isso basta clicar em **Save All** .

## 10. Construindo Relatórios

O Quick Report permite que você projete visualmente os Relatórios. Você constrói relatórios com bandas, adicionando títulos, cabeçalhos e rodapés de página, detalhes, sumários, cabeçalhos e rodapés de grupo, etc. O Quick Report possui um conjunto de componentes nativos que permite ao desenvolvedor desenhar o relatório dentro de um Form do Delphi. Estes componentes executam toda a tarefa de visualização e impressão do relatório. Os relatórios podem ser tão complexos e sofisticados quanto for necessário.



O Delphi possui uma paleta de componentes chamada **QReport**. Nesta paleta está uma série de componentes relativos à criação de relatórios baseados em tabelas de um banco de dados.

Para usar os componentes do **QReport** você terá que adicionar um formulário em branco à sua aplicação, em seguida inserir neste formulário o componente **QuickReport**. Desta forma, o formulário se tornará um Formulário de Relatório; ligar o componente **QuickReport** à origem de dados através da sua propriedade **DataSet**, inserir bandas no formulário, e dentro delas colocar os componentes relativos a dados. Finalmente, para imprimir o relatório, executar os métodos **Preview** ou **Print**.

A grande vantagem do Quick Report é a forma de criar um relatório, ela é muito similar à criação das janelas de uma aplicação. Ou seja, você cria um relatório com o QuickReport da mesma maneira que uma janela no Delphi, visualmente.

Principais Componentes do Quick Report		
COMPONENTE	VISUAL	DESCRIÇÃO
QuickReport		É o principal componente na execução da impressão. Transforma um formulário em um relatório.
QRSubDetail		Usado para criar qualquer tipo de relatório <i>Master/Detail (um-para-vários)</i>
QRBand		Um relatório é construído em <i>Bandas (regiões)</i> . Estas regiões podem ser: cabeçalho, detalhes, rodapé, etc.
QRGroup		Permite que se agrupem os registros em blocos, com pelo menos uma característica em comum.
QRLabel		Mostra um texto no relatório. Utilizado para representar títulos, tais como <i>Cabeçalhos de Colunas</i> .
QRDBText		Mostra o conteúdo de um determinado campo do <i>DataSet (Tabela ou Query)</i> .
QRExpr		Avalia e imprime qualquer expressão Quick Report. Permite usar campos do banco de dados em expressões.
QRSysData		Exibe informações do sistema, tais como: <i>data, hora, número de página, título do relatório</i> , etc.

Principais Componentes do Quick Report		
COMPONENTE	VISUAL	DESCRIÇÃO
QRMemo		Usado para imprimir um texto com múltiplas linhas. Usado para áreas longas de texto estático.
QRRichText		Exibe um componente <i>Memo</i> do tipo <i>RichText</i> . Pode ser ligado a um componente <i>RichText</i> no formulário e imprimir o conteúdo.
QRDBRichText		Tal qual <i>QRRichText</i> , só que para campos de banco de dados ( <i>Data-Aware</i> )
QRShape		Adiciona objetos de desenho (retângulo, circunferência e linhas) ao formulário.
QRImage		Usado para mostrar imagem estática no relatório.
QRDBImage		Usado para imprimir imagens armazenadas em <i>Campos Binários (BLOB)</i> .
QRPreview		Possibilita que você crie seu próprio formulário de <i>Preview (Visualização)</i> de impressão.

## Bandas de Dados

O Quick Report gera relatórios usando bandas. Você constrói seus relatórios adicionando o componente *TQRBand* (um tipo especial de *TPanel*) ao *TQuickReport*. Cada banda do seu relatório possui uma finalidade diferente no resultado final. Um relatório pode ter várias bandas de tipos diferentes e/ou iguais.

**Title:** Representa o título do relatório. É impressa apenas uma vez durante todo o relatório: no início da primeira página. Você pode ter várias bandas deste tipo: elas serão impressas uma após a outra, no início da primeira página.

**PageHeader:** Essa banda é impressa uma vez a cada início de página. Para cada página do relatório existe uma impressão de uma banda deste tipo. Na primeira página esta banda também é impressa. Você pode ter várias bandas deste tipo no relatório, que elas serão impressas uma após a outra automaticamente.

**ColumnHeader:** Esta banda se assemelha bastante com a *PageHeader*, com a única diferença que ela é impressa no topo de cada coluna, num relatório de várias colunas na mesma página.

**Detail:** Esta banda é repetida para cada registro de dados do seu relatório. Usualmente existe somente uma banda *Detail* em cada relatório, embora seja possível colocar mais de uma.

**PageFooter:** A banda *PageFooter* é impressa no final de cada página do relatório. Funciona como a *PageHeader*, com a diferença que é impressa no rodapé da página.

**Summary:** É impressa na última página depois de todas as bandas de detalhe.

**SubDetail:** É um tipo de banda especial usada pelo componente *TQRSubDetail*. Serve para efetuar relatórios Master / Detail.

**GroupHeader:** Este tipo de banda é usualmente usada para agrupar valores (em relatórios Master / Detail) e você pode efetuar um somatório em pequenos grupos de valores, dependendo de uma certa condição.

**GroupFooter:** É um tipo que serve como finalizador para a banda *GroupHeader*.

**Child:** É um tipo especial de banda, usada pelo componente *TQRChildBand*.

Ordem de impressão de Bandas:

*Não importa em que ordem às bandas aparecem em tempo de desenvolvimento no seu relatório. Elas sempre aparecerão na ordem correta na impressão.*

Vamos aqui mostrar qual é a ordem seguida pelo Quick Report:

**PageHeader** (para todas as páginas) → **Title** (primeira página) → **ColumnHeader** (para todas as colunas) → **Detail** (repete para todos os registros do relatório) → **SubDetail** (caso haja um) → **Summary** (última página) → **PageFooter** (para todas as páginas).

### **Roteiro para Criação de Relatório:**

1. Criar um novo formulário para a montagem do relatório (File - New Form);
2. Trazer para o formulário um componente QuickRep (Página QReport);
  - a) Para configurar o tamanho do papel, dê um duplo clique na margem;
  - b) Preencher a propriedade DataSet com a tabela que será impressa;
3. Adicionar no QuickRep as "bandas" necessárias para a impressão do seu relatório (Componente QRBand).

Obs:

Utilize a propriedade Frame para colocar bordas numa banda.

Bandas de cabeçalho de grupo (GroupHeader) e de sub-detilhe (SubDetail) NÃO PODEM ser construídas com componente QRBand. Utilize QRGroup e QRSubDetail.

O Quick Report não efetua filtragens de dados ou relacionamento de tabelas. Você deve filtrar e relacionar as tabelas a partir dos componentes de acesso a dados.

### **Relatório com Quebra de Grupo**

#### **O Cabeçalho do Grupo:**

- Traga para o QuickRep um componente QRGroup;
- Verifique se a propriedade Master aponta para o QuickRep;
- Preencha a propriedade Expression com o campo que será a chave de quebra;
- A tabela deverá estar ordenada por este campo no momento da impressão;
- Coloque sobre a banda o título do grupo;
- Você pode tornar True a propriedade ForceNewPage para saltar de página a cada grupo.

### **O Rodapé do Grupo:**

- Coloque uma nova banda sobre o QuickRep (QRBand);
- Altere BandType para rbGroupFooter (ela saltará para o final);
- Altere a propriedade Name com um nome explicativo;
- Selecione a banda de cabeçalho de grupo (GroupHeader) e preencha a propriedade FooterBand com o nome da banda de rodapé (ela saltará para depois da banda de detalhe);
- Os totais (QRExpr) que você colocar na banda de rodapé de grupo serão acumulados de um grupo para outro, a não ser que para cada QRExpr você altere para True a propriedade ResetAfterPrint.

### **Relatório Mestre/Detalhe (um para vários)**

- Componente QuickRep deve ter sua propriedade DataSet apontando para a tabela Mestre;
- Traga para o QuickRep um QRBand e altere sua propriedade BandType para rbDetail. Nela coloque os campos da sua tabela Mestre;
- Traga para o QuickRep um QRSubDetail;
- Sua propriedade Master deve apontar para o QuickRep;
- Sua propriedade DataSet deve apontar para a tabela de detalhe;
- Nesta banda coloque os campos da tabela de detalhe;
- As tabelas já devem estar relacionadas a partir dos componentes de acesso a dados;
- Para totalizar os registros do sub-detalhe devemos trazer uma banda (QRBand) e configurar sua propriedade BandType para rbGroupFooter:
  - A propriedade FooterBand do sub-detalhe deve apontar para a banda de GroupFooter;
  - A propriedade Master dos componentes QRExpr que efetuam os cálculos de totais devem apontar para a banda de sub-detalhe;
  - Os totais (QRExpr) que você colocar na banda de rodapé do grupo serão acumulados de um grupo para outro, a não ser que para cada QRExpr você altere para True a propriedade ResetAfterPrint.

### **Instalando Quick Report no Delphi 7**

No Delphi 7, o Quick Report vem incluso, ele apenas não instala os componentes automaticamente.

Para instalar, abra o Delphi, use o menu **Componet / Install Packages**, clique em **ADD** e localize o arquivo *c:\arquivos de programas\borland\delphi7\bin\dc1qtr70.bpl* e agora clique em **ADD** novamente e localize o arquivo *c:\arquivos de programas\borland\delphi7\bin\dc1tqr70.bpl*.

Após isso a paleta do **QReport** estará instalada.

## Criando um relatório

Abra o projeto Funcionario.dpr e proceda as mudanças solicitadas.

Para criarmos um relatório, precisaremos de um novo form. Então:

- Clique em **File / New / Form**;
- Clique em **File / Use Unit** (Para usarmos os componentes de acessos a dados da Unit untCadFuncionario);
- Selecione a Unit **untCadFuncionario** e clique no botão **OK**;
- Altere a propriedade **Caption** para **Relatório de Funcionários**;
- Altere a propriedade **Name** para **frmRelFuncionario**;
- Clique **File / Save As** e salve a nova Unit como **untRelatorio**;
- Agora vá até a aba **QReport** da paleta de componentes e procure pelo componente **TQuickRep**  (o primeiro);
- Inclua um no Form;
- Procure sua propriedade **Bands** (a primeira). Vamos ver um pouco sobre as Bands:

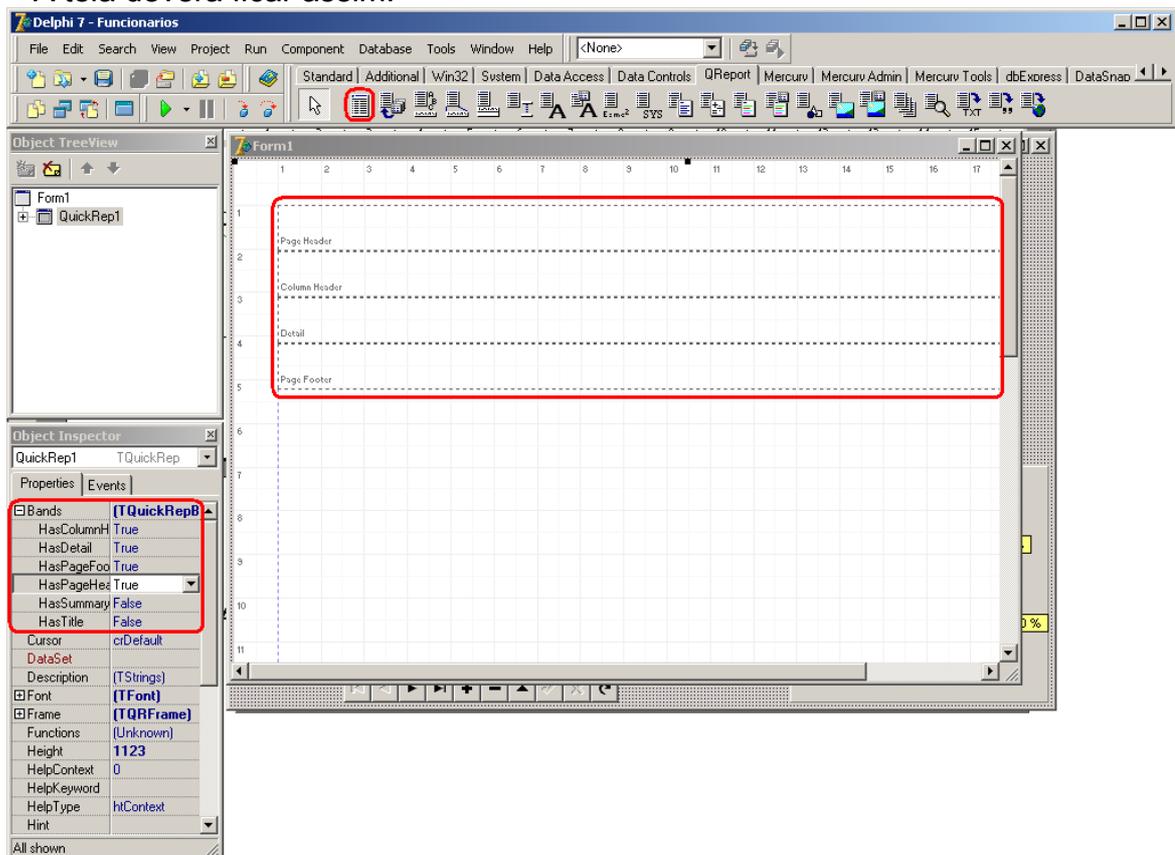
### Bands

As Bands do QuickRep são responsáveis pela divisão do relatório em faixas, de acordo com a função desejada. Existe uma faixa para o cabeçalho, outra para o rodapé, etc.

- Clique sobre o sinal de **+** ao lado da propriedade **Bands** (ou de um duplo clique sobre ela), e observe a lista que se abre:

Band	Função
HasColumnHeader	Põe uma faixa para edição do cabeçalho de coluna. Essa faixa aparece entre o cabeçalho e a linha de detalhe. ➤ Coloque em <b>True</b> .
HasDetail	Põe uma faixa para edição de linha de detalhe. A linha de detalhe aparece uma vez para cada registro da tabela. ➤ Coloque em <b>True</b> .
HasPageFooter	Põe uma faixa para edição do rodapé. Essa faixa aparece no fim de cada página. ➤ Coloque em <b>True</b> .
HasPageHeader	Põe uma faixa para edição do cabeçalho. O cabeçalho aparece no topo de cada página do relatório. ➤ Coloque em <b>True</b> .
HasSummary	Põe uma faixa para edição do sumário. Essa faixa aparece no fim da última página do relatório, antes do rodapé. ➤ Deixe em <b>False</b> .
HasTitle	Põe uma faixa (band) para edição do título do relatório. Título é aquela parte do relatório que só aparece no topo da primeira página, depois do cabeçalho. ➤ Deixe em <b>False</b> .

➤ A tela deverá ficar assim:



Veremos agora, outras propriedades importantes do componente QuickRep:

Propriedades	Funções
DataSet	Esta é a propriedade que aponta para a fonte dos dados que serão impressos. ➤ Selecione <b>frmCadFuncionario.MDOQuery1</b> nesta propriedade.
Frame	Coloca linhas (margens) ao redor do relatório. <b>Não alteraremos esta parte</b> , porém altere-a e veja os resultados. (para visualizar o arquivo a ser impresso, basta clicar sobre o QuickRep com o botão direito do mouse e selecionar Preview).
Name	É o nome dado ao componente. <b>Neste caso deixe como esta.</b>
Options	Estas opções permitem impedir que o cabeçalho apareça na primeira página ou que o rodapé apareça na última. Isso é útil se você quiser que o título ou a faixa de fim de relatório apareçam isolados, sem cabeçalho ou rodapé. <b>Neste caso deixe como esta.</b>
Page	Define o formato da página. Altura, largura, margens, orientação (retrato ou paisagem), etc. <b>Por enquanto deixe como esta</b> , após o término do projeto, altere estas propriedades e veja o resultado.
ReportTitle	É o nome que vai aparecer na fila de impressão do Windows ou na rede. <b>Altere para Relatório de Funcionários.</b>

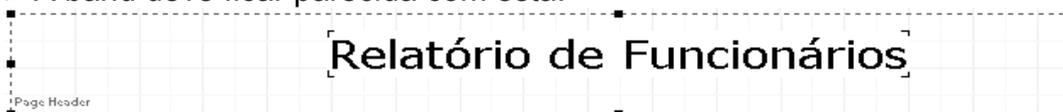
Vamos agora iniciar o desenho de cada Band do nosso relatório:

### Cabeçalho:

- Selecione a faixa PageHeader (a primeira);
- Clique e arraste a âncora inferior da faixa para aumentar o tamanho. Âncoras são aqueles pontos pretos que aparecem em volta do componente selecionado;



- Coloque um componente **QRLabel**  na faixa **PageHeader**. Note que os componentes não podem ser movidos de uma faixa para outra;
- Centralize o componente **QRLabel** mudando a propriedade **Alignment** para **taCenter** e a propriedade **AlignToBand** para **True**;
- Mude a propriedade **Caption** para **Relatório de Funcionários**;
- Mude a opção **Size** da propriedade **Font** para **20** e **Name** para **Verdana**;
- A band deve ficar parecida com esta:



### Cabeçalho de Coluna:

- Coloque 3 componentes **QRLabel**  na faixa **ColumnHeader** (um para cada coluna que queremos imprimir no relatório);
- Altere as propriedades **Caption** para **Código**, **Nome** e **Idade** do Funcionário;
- Altere o **Style** da propriedade **Font** para **Underline**;



### Linha de detalhe:

- Coloque 3 componentes **QRDBText**  na faixa **Detail** (um para cada coluna do relatório). Faça com que eles fiquem grudados na parte superior da faixa;
- Reduza a altura da faixa até não sobrar espaço entre os componentes e o fim da faixa. Se sobrar algum espaço, as linhas de detalhe vão aparecer afastadas;
- Mude a propriedade **DataSet** de cada componente **QRDBText** para **frmCadFuncionario.MDOQuery1**. Isso vai indicar a fonte dos dados;
- Mude as propriedades **DataField** para os respectivos campos de cada coluna do relatório: **FUN\_CODIGO**, **FUN\_NOME** e **FUN\_IDADE**;
- O QuickRep deverá ficar assim:



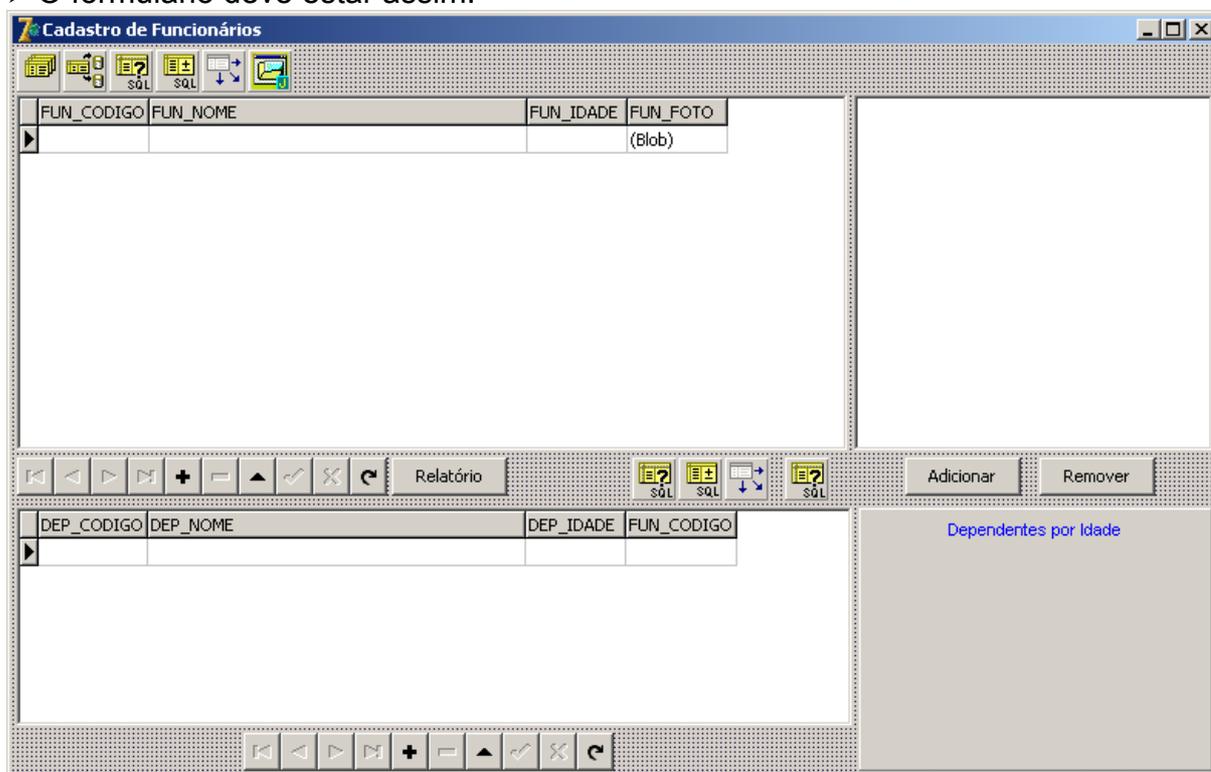
## Rodapé:

- Coloque um componente **QRSysData**  na faixa **PageFooter** à esquerda. Esse componente é usado para imprimir informações do sistema, tais como data, hora e número da página;
- Mude a propriedade **Data** para **qrsDateTime** (data e hora atual);
- Coloque outro componente **QRSysData**  na faixa **PageFooter** à direita;
- Mude a propriedade **Data** para **qrsPageNumber** (número da página), e a propriedade **text** para **Pág.:**;
- O QuickRep deverá ficar assim:

Relatório de Funcionários			
Page Header			
Código <sub>der</sub>	Nome	Idade	
FUN_CODIGO	FUN_NOME	FUN_IDADE	
{Date/Time}			Pág.: (Page#)

## Chamando a Tela de Visualização:

- Clique em **View / Forms / frmCadFuncionario / OK**;
- Agora clique em **File / Use Unit / untRelatorio / OK**. Ou seja, para chamarmos a tela de visualização do relatório, precisamos fazer com que o frmFuncionario "enxergue" o frmRelFuncionarios;
- Coloque um componente **Button** (paleta **Standard**);
- Mude a propriedade **Caption** para **Relatório**;
- O formulário deve estar assim:



FUN_CODIGO	FUN_NOME	FUN_IDADE	FUN_FOTO
			(Blob)

DEP_CODIGO	DEP_NOME	DEP_IDADE	FUN_CODIGO

Adicionar    Remover

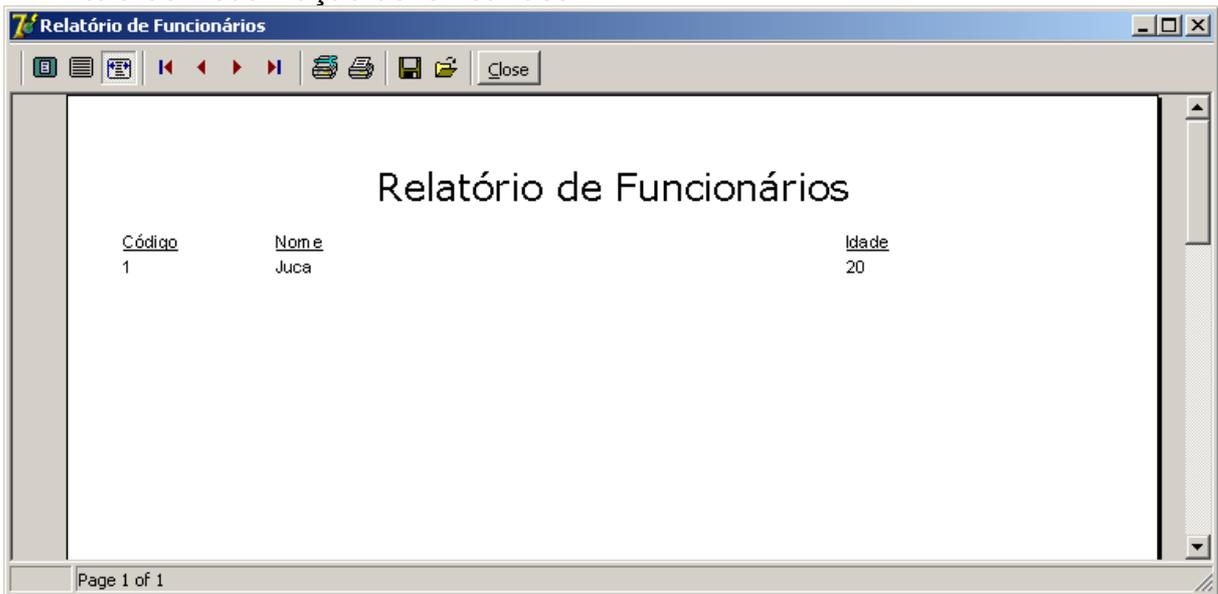
Relatório    Dependentes por Idade

- Agora dê um duplo-clique sobre o botão Relatório e digite as linhas em negrito do código abaixo:

```
procedure TfrmCadFuncionario.Button3Click(Sender: TObject);  
begin  
    frmRelFuncionario.QuickRep1.Preview;  
end;
```

**Preview:** É uma propriedade do **QuickRep**, que chama a tela para visualização do relatório, caso queira imprimi-lo diretamente basta trocar a procedure **Preview** pela **Print**.

- Clique em **File / Save All, Compile (F9)** e veja o resultado;
- A tela de visualização deve ficar assim:



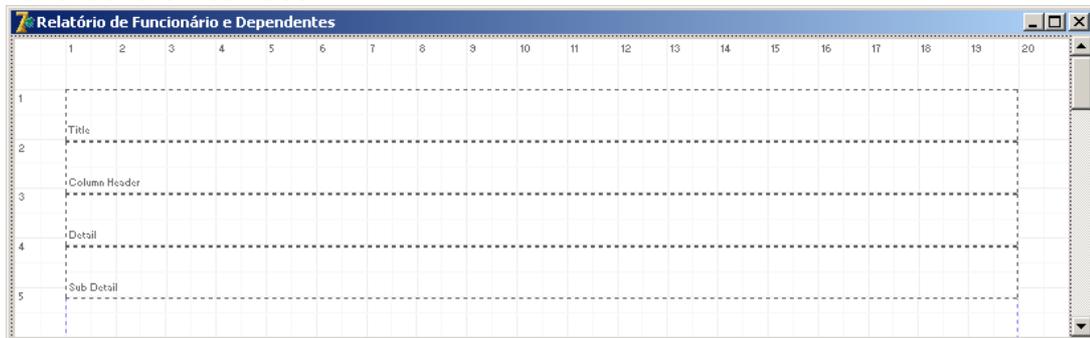
## Criando um Relatório Mestre-Detalhe

Um relatório composto de mais de uma tabela onde há um relacionamento explícito através de uma chave primária e uma estrangeira é denominado Mestre-Detalhe. A tabela que fornece a chave primária é a tabela Mestre e a tabela que trabalha com a chave estrangeira é a tabela Detalhe.

Por exemplo, vamos criar um relatório entre Funcionário e Dependente, para cada funcionário (chave primária) pode haver: zero, um ou vários dependentes (a tabela dependente tem o campo FUN\_CODIGO como chave estrangeira). Será exibido da tabela Mestre: o código e o nome do funcionário, da tabela Detalhe: o nome do dependente.

- Crie um novo formulário;
- Clique em **File / Use Unit** (Para usarmos os componentes de acessos a dados da Unit *untCadFuncionario*);
- Selecione a *Unit untCadFuncionario* e clique no botão **OK**;
- Altere a propriedade **Caption** para *Relatório de Funcionário e Dependentes*;
- Altere a propriedade **Name** para *frmRelFuncDep*;

- Clique **File / Save As** e salve a nova Unit como *untRelFunDep*;
- Insira um **QuickRep** com as seguintes bandas: *Title*, *Column Header*, *Detail* e *Sub Detail*;
- A tela deverá ficar assim:

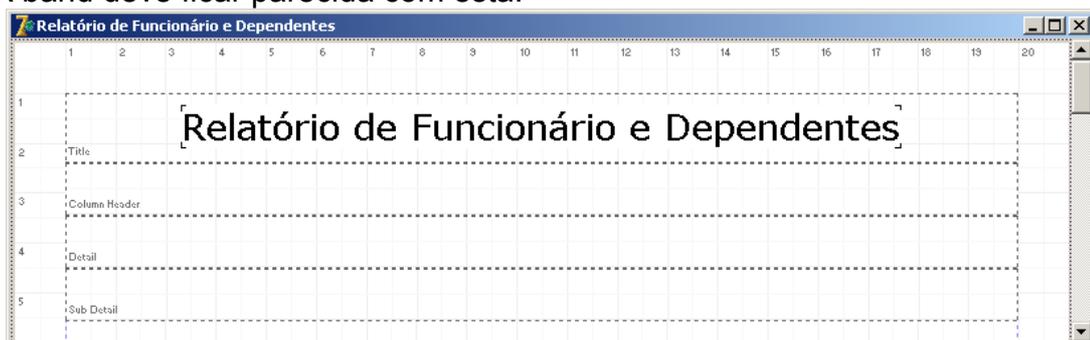


Vamos agora, alterar as seguintes propriedades do **QuickRep**:

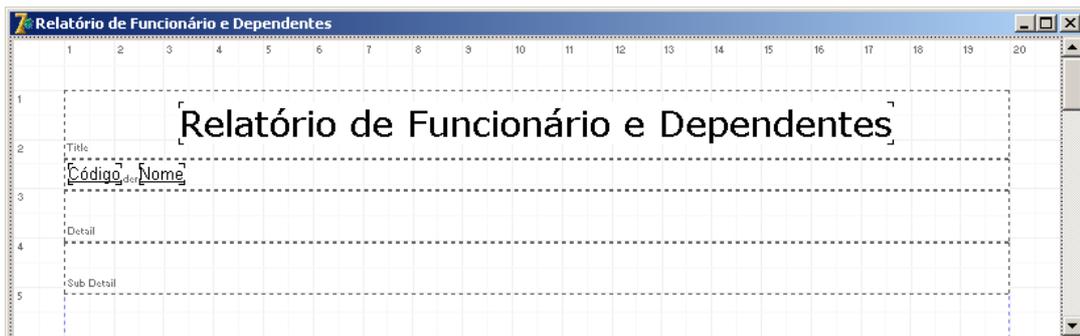
- **DataSet**: `frmCadFuncionario.MDOQuery1`
- **ReportTitle**: *Relatório de Funcionário e Dependentes*
- Selecione a faixa *Title* (a primeira);
- Clique e arraste a âncora inferior da faixa para aumentar o tamanho. Âncoras são aqueles pontos pretos que aparecem em volta do componente selecionado;



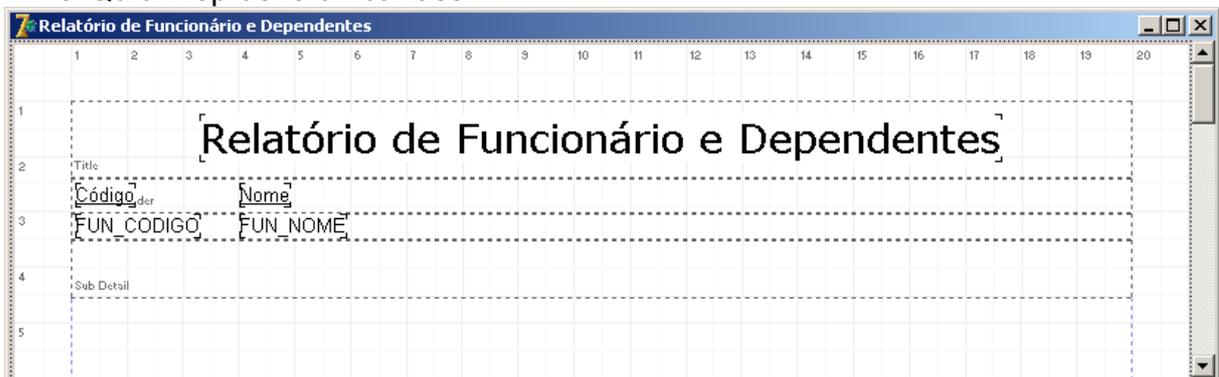
- Coloque um componente **QRLabel** na faixa *Title*;
- Centralize o componente **QRLabel** mudando a propriedade **Alignment** para *taCenter* e a propriedade **AlignToBand** para *True*;
- Mude a propriedade **Caption** para *Relatório de Funcionário e Dependentes*;
- Mude a opção **Size** da propriedade **Font** para *20* e **Name** para *Verdana*;
- A band deve ficar parecida com esta:



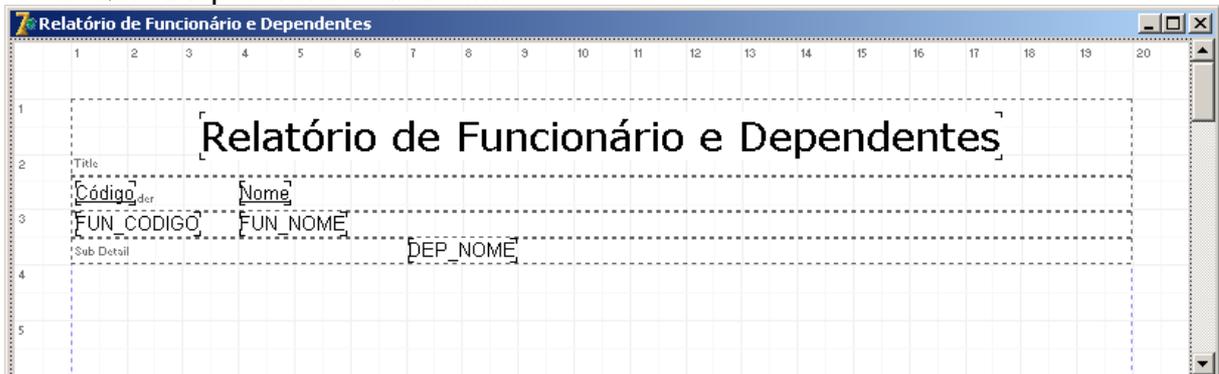
- Coloque 2 componentes **QRLabel** na faixa **ColumnHeader**;
- Altere as propriedades **Caption** para *Código* e *Nome*;
- Altere o **Style** da propriedade **Font** para *Underline*;



- Coloque 2 componentes **QRDBText** na faixa **Detail** (um para cada coluna do relatório). Faça com que eles fiquem grudados na parte superior da faixa;
- Reduza a altura da faixa até não sobrar espaço entre os componentes e o fim da faixa. Se sobrar algum espaço, as linhas de detalhe vão aparecer afastadas;
- Mude a propriedade **DataSet** de cada componente **QRDBText** para *frmCadFuncionario.MDOQuery1*. Isso vai indicar a fonte dos dados;
- Mude as propriedades **DataField** para os respectivos campos de cada coluna do relatório: *FUN\_CODIGO* e *FUN\_NOME*;
- O QuickRep deverá ficar assim:

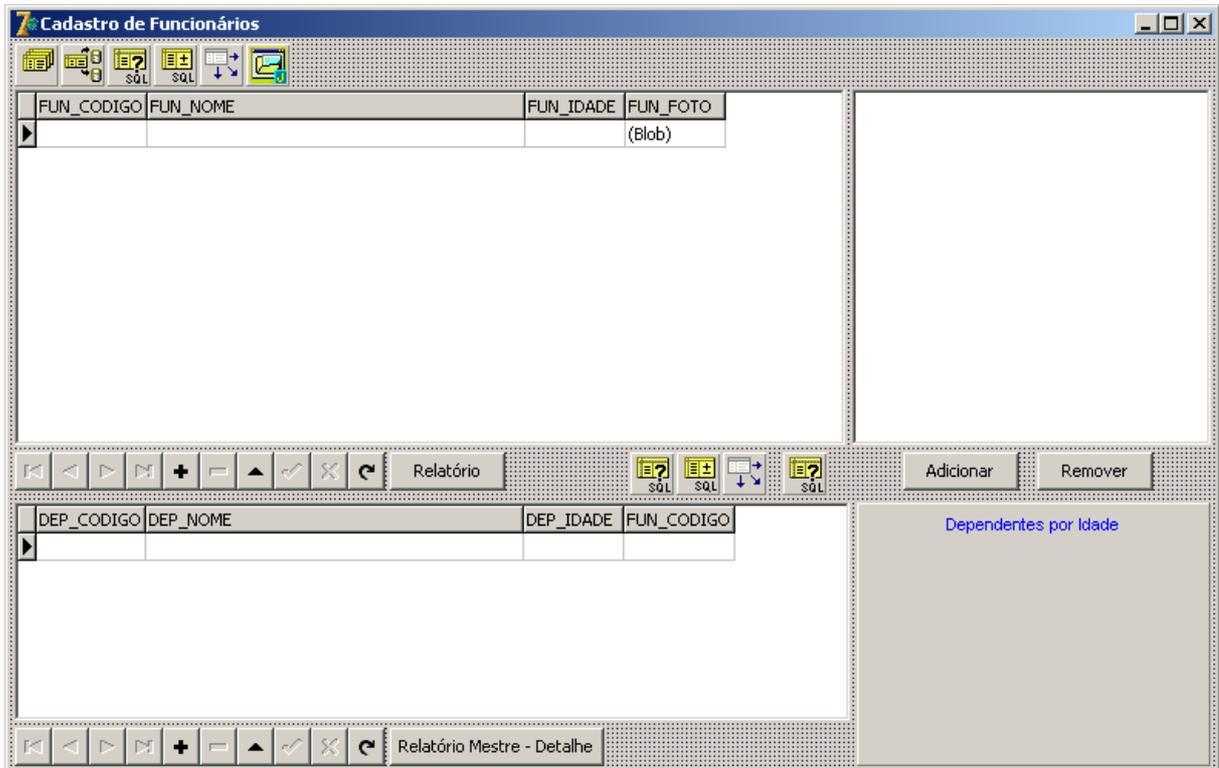


- Mude a propriedade **DataSet** da faixa **Sub Detail** para *frmCadFuncionario.MDOQuery2*. Isso vai indicar a fonte de dados detalhe;
- Coloque 1 componentes **QRDBText** na faixa **Detail** (um para cada coluna do relatório). Faça com que eles fiquem grudados na parte superior da faixa;
- Reduza a altura da faixa até não sobrar espaço entre os componentes e o fim da faixa. Se sobrar algum espaço, as linhas de detalhe vão aparecer afastadas;
- Mude a propriedade **DataSet** de cada componente **QRDBText** para *frmCadFuncionario.MDOQuery2*. Isso vai indicar a fonte dos dados;
- Mude a propriedades **DataField** para o respectivo campo de cada coluna do relatório: *FUN\_NOME*;
- O QuickRep deverá ficar assim:



- Clique em **View / Forms / frmCadFuncionario / OK**;

- Agora clique em **File / Use Unit / untRelFuncDep / OK**. Ou seja, para chamarmos a tela de visualização do relatório, precisamos fazer com que o frmFuncionario "enxergue" o frmRelFuncDep;
- Coloque um componente **Button** (paleta **Standard**);
- Mude a propriedade **Caption** para *Relatório Mestre - Detalhe*;
- O formulário deve estar assim:

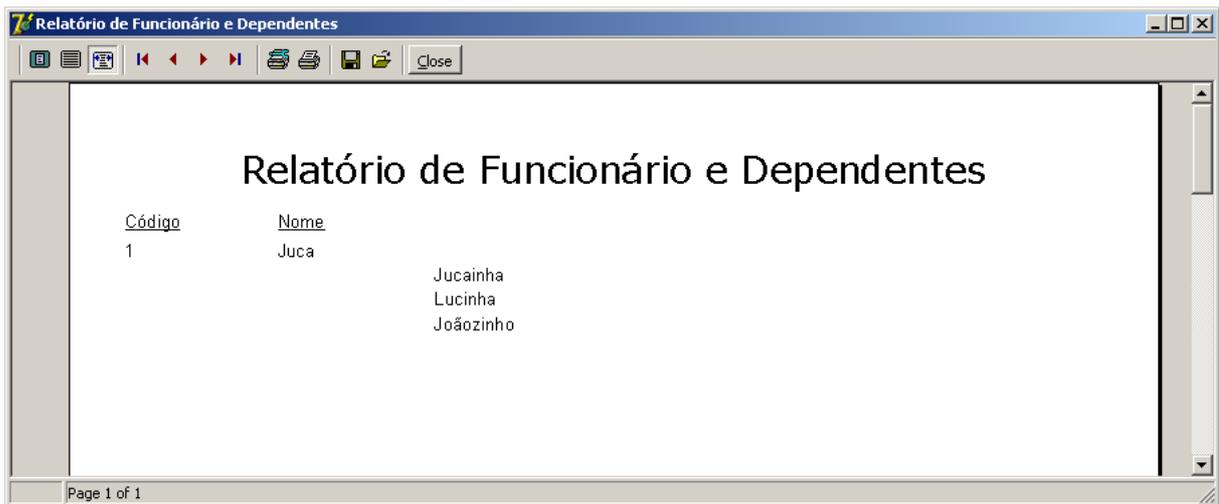


- Agora dê um duplo-clique sobre o botão Relatório e digite as linhas em negrito do código abaixo:
 

```

      procedure TfrmCadFuncionario.Button4Click(Sender: TObject);
      begin
        frmRelFuncDep.QuickRep1.Preview;
      end;
      
```

**Preview:** É uma propriedade do **QuickRep**, que chama a tela para visualização do relatório, caso queira imprimir diretamente basta trocar a procedure **Preview** pela **Print**.
- Clique em **File / Save All, Compile (F9)** e veja o resultado;
- A tela de visualização deve ficar assim:



## 11. Projeto Piloto

É óbvio que fica mais simples o aprendizado de uma nova ferramenta quando se faz algum tipo de aplicativo, principalmente um que seja útil, então ao longo deste estudo, você irá desenvolver um aplicativo destinado ao Controle de Materiais.

Para darmos partida ao nosso primeiro aplicativo (Projeto Piloto), definiremos inicialmente as nossas necessidades:

1. Permitir o cadastro completo e a consulta aos materiais;
2. Se possível separar os materiais em grupos, facilitando deste modo à busca e o armazenamento;
3. Quanto aos grupos é preciso cadastrar: código, descrição e prazo de entrega;
4. Quanto aos materiais deve ser permitido cadastrar: código, descrição, grupo, quantidade e valor unitário;
5. Permitir a inclusão da foto do material;
6. Comportar relatórios de conferência;
7. Permitir acesso ao sistema através do uso de login e senha, através de um cadastro de usuários pré-definidos.

Com base no que foi sugerido acima, vamos definir as tabelas:

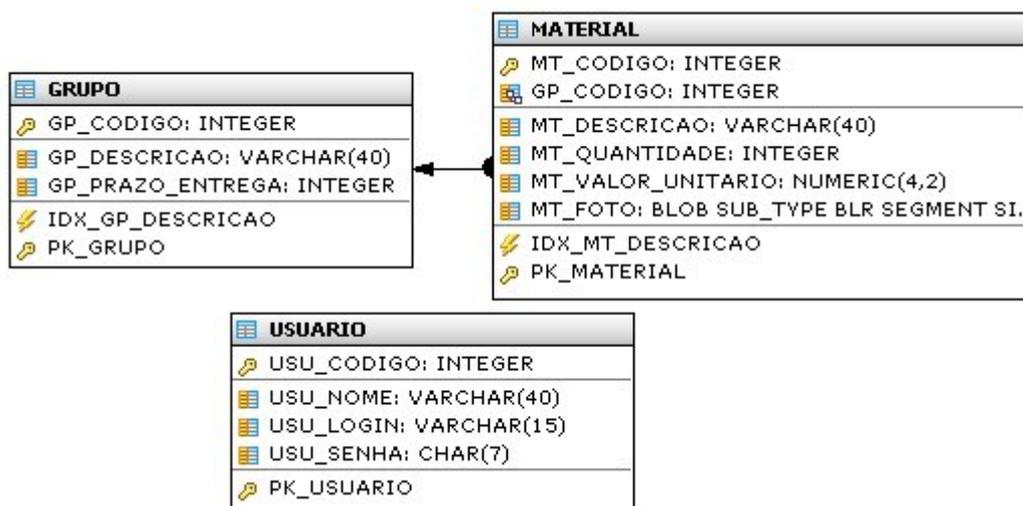
Tabela:	GRUPO
Objetivo:	Dados dos grupos de materiais.
Campos:	CÓDIGO DO GRUPO – Código seqüência do grupo.
	DESCRIÇÃO DO GRUPO – Descrição do grupo.
	PRAZO DE ENTREGA – Prazo de entrega do material do grupo em questão.

Tabela:	MATERIAL
Objetivo:	Dados dos materiais.
Campos:	CÓDIGO DO MATERIAL – Código seqüencial do material.
	DESCRIÇÃO DO MATERIAL – Nome do material.
	QUANTIDADE DO MATERIAL – Quantidade do material no estoque.
	VALOR UNITÁRIO – Valor individual de cada peça do material.
	FOTO DO MATERIAL – Foto ilustrativa do material.
	CÓDIGO DO GRUPO – Grupo do material.

Tabela:	USUARIO
Objetivo:	Dados dos usuários.
Campos:	CÓDIGO DO USUÁRIO – Código seqüencial do usuário.
	NOME DO USUÁRIO – Nome do usuário.
	LOGIN – Nome usado para acessar o sistema.
	SENHA – Conjunto de caracteres que permite acesso ao sistema.

### **Modelo Relacional**

Vamos agora visualizar um modelo que demonstrará como as tabelas deverão se relacionar no sistema, este modelo também facilitará as relações e a integração dos formulários do sistema.



No modelo de entidade e relacionamento acima se lê: 1 registro da entidade MATERIAL se relaciona com 1 registro da entidade GRUPO, enquanto que 1 registro da entidade GRUPO se relaciona com n registros da entidade MATERIAL.

Com o MER nas mãos já se tem a idéia de como deve ficar as estruturas das tabelas.

Tabela	GRUPO		
Campo	Tipo	Tamanho	Chave
GP_CODIGO	Integer		Primary

GP_DESCRICAO	Varchar	40	
GP_PRAZO_ENTREGA	Integer		

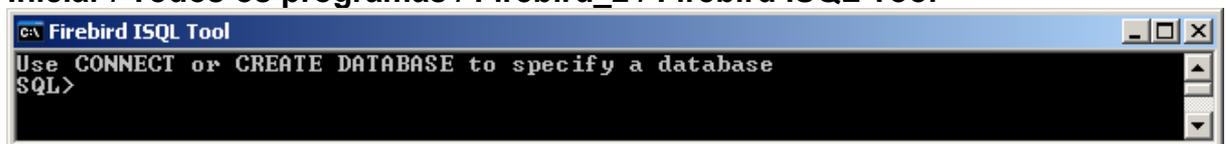
Tabela			
MATERIAL			
Campo	Tipo	Tamanho	Chave
MT_CODIGO	Integer		Primary
MT_DESCRICAO	Varchar	40	
MT_QUANTIDADE	Integer		
MT_VALOR_UNITARIO	Numeric	4, 2	
MT_FOTO	Blob (2)	1	
GP_CODIGO	Integer		Foreign

Tabela			
USUARIO			
Campo	Tipo	Tamanho	Chave
USU_CODIGO	Integer		Primary
USU_NOME	Varchar	40	
USU_LOGIN	Varchar	15	
USU_SENHA	Caractere	7	

## 11.1. Criando o Banco do Projeto Piloto

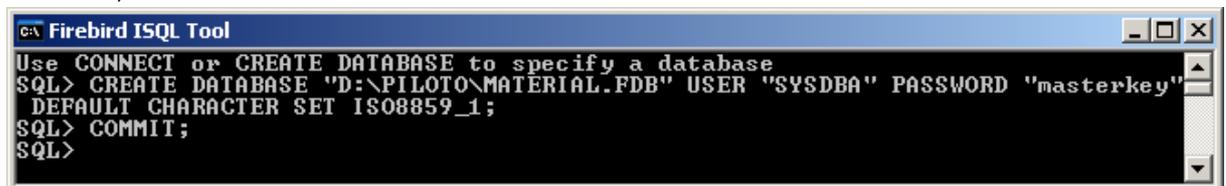
Crie com o Gerenciador de Arquivos a pasta (diretório) c:\piloto, onde ficará armazenado o sistema.

### Iniciar / Todos os programas / Firebird\_2 / Firebird ISQL Tool



Criando o arquivo do banco de dados:

```
CREATE DATABASE "D:\PILOTO\MATERIAL.FDB" USER "SYSDBA" PASSWORD
"masterkey" DEFAULT CHARACTER SET ISO8859_1;
COMMIT;
```



**Nota:** Se o banco de dados já existir, será necessário estabelecer uma conexão com o mesmo. Estabelece-se uma conexão com o banco de dados através do seguinte comando:

```
CONNECT "D:\PILOTO\MATERIAL.FDB" USER "SYSDBA" PASSWORD "masterkey";
```



### Criando a tabela de grupo:

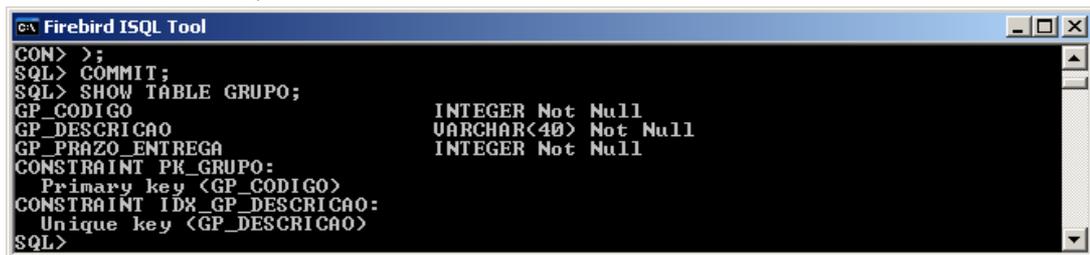
```
CREATE TABLE GRUPO (  
    GP_CODIGO INTEGER NOT NULL,  
    GP_DESCRICAO VARCHAR(40) NOT NULL,  
    GP_PRAZO_ENTREGA INTEGER NOT NULL,  
    CONSTRAINT PK_GRUPO PRIMARY KEY (GP_CODIGO),  
    CONSTRAINT IDX_GP_DESCRICAO UNIQUE (GP_DESCRICAO)  
);  
COMMIT;
```



```
ca Firebird ISQL Tool  
SQL> CONNECT "c:\piloto\material.fdb" user "SYSDBA" password "masterkey";  
Database: "c:\piloto\material.fdb", User: SYSDBA  
SQL> CREATE TABLE GRUPO <  
CON> GP_CODIGO INTEGER NOT NULL,  
CON> GP_DESCRICAO VARCHAR(40) NOT NULL,  
CON> GP_PRAZO_ENTREGA INTEGER NOT NULL,  
CON> CONSTRAINT PK_GRUPO PRIMARY KEY (GP_CODIGO),  
CON> CONSTRAINT IDX_GP_DESCRICAO UNIQUE (GP_DESCRICAO)  
CON> >;  
SQL> COMMIT;  
SQL>
```

### Mostrando a estrutura da tabela de grupo:

```
SHOW TABLE GRUPO;
```



```
ca Firebird ISQL Tool  
CON> >;  
SQL> COMMIT;  
SQL> SHOW TABLE GRUPO;  
GP_CODIGO                INTEGER Not Null  
GP_DESCRICAO              VARCHAR(40) Not Null  
GP_PRAZO_ENTREGA         INTEGER Not Null  
CONSTRAINT PK_GRUPO:  
  Primary key (GP_CODIGO)  
CONSTRAINT IDX_GP_DESCRICAO:  
  Unique key (GP_DESCRICAO)  
SQL>
```

### Criando a tabela de material:

```
CREATE TABLE MATERIAL (  
    MT_CODIGO INTEGER NOT NULL,  
    MT_DESCRICAO VARCHAR(40) NOT NULL,  
    MT_QUANTIDADE INTEGER NOT NULL,  
    MT_VALOR_UNITARIO NUMERIC(4, 2) NOT NULL,  
    MT_FOTO BLOB SUB_TYPE 2 SEGMENT SIZE 1,  
    GP_CODIGO INTEGER NOT NULL,  
    CONSTRAINT PK_MATERIAL PRIMARY KEY (MT_CODIGO),  
    CONSTRAINT IDX_MT_DESCRICAO UNIQUE (MT_DESCRICAO),  
    CONSTRAINT FK_MATERIAL_GRUPO  
        FOREIGN KEY (GP_CODIGO) REFERENCES GRUPO  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);  
COMMIT;
```

```
ca Firebird ISQL Tool
SQL> CREATE TABLE MATERIAL (
CON>   MT_CODIGO INTEGER NOT NULL,
CON>   MT_DESCRICAO VARCHAR(40) NOT NULL,
CON>   MT_QUANTIDADE INTEGER NOT NULL,
CON>   MT_VALOR_UNITARIO NUMERIC(4, 2) NOT NULL,
CON>   MT_FOTO BLOB SUB_TYPE 2 SEGMENT SIZE 1,
CON>   GP_CODIGO INTEGER NOT NULL,
CON>   CONSTRAINT PK_MATERIAL PRIMARY KEY (MT_CODIGO),
CON>   CONSTRAINT IDX_MT_DESCRICAO UNIQUE (MT_DESCRICAO),
CON>   CONSTRAINT FK_MATERIAL_GRUPO
CON>     FOREIGN KEY (GP_CODIGO) REFERENCES GRUPO
CON>     ON DELETE CASCADE
CON>     ON UPDATE CASCADE
CON> );
SQL> COMMIT;
SQL>
```

Mostrando a estrutura da tabela de material:

SHOW TABLE MATERIAL;

```
ca Firebird ISQL Tool
SQL> COMMIT;
SQL> SHOW TABLE MATERIAL;
MT_CODIGO                INTEGER Not Null
MT_DESCRICAO              VARCHAR(40) Not Null
MT_QUANTIDADE             INTEGER Not Null
MT_VALOR_UNITARIO        NUMERIC(4, 2) Not Null
MT_FOTO                   BLOB segment 1, subtype BLR Nullable
GP_CODIGO                 INTEGER Not Null
CONSTRAINT FK_MATERIAL_GRUPO:
  Foreign key (GP_CODIGO) References GRUPO (GP_CODIGO) On Update Cascade On D
elete Cascade
CONSTRAINT PK_MATERIAL:
  Primary key (MT_CODIGO)
CONSTRAINT IDX_MT_DESCRICAO:
  Unique key (MT_DESCRICAO)
SQL>
```

Criando a tabela de usuários:

```
CREATE TABLE USUARIO (
  USU_CODIGO INTEGER NOT NULL,
  USU_NOME VARCHAR(40) NOT NULL,
  USU_LOGIN VARCHAR(15) NOT NULL,
  USU_SENHA CHAR(7) NOT NULL,
  CONSTRAINT PK_USUARIO PRIMARY KEY (USU_CODIGO)
);
COMMIT;
```

```
ca Firebird ISQL Tool
SQL> CREATE TABLE USUARIO (
CON>   USU_CODIGO INTEGER NOT NULL,
CON>   USU_NOME VARCHAR(40) NOT NULL,
CON>   USU_LOGIN VARCHAR(15) NOT NULL,
CON>   USU_SENHA CHAR(7) NOT NULL,
CON>   CONSTRAINT PK_USUARIO PRIMARY KEY (USU_CODIGO)
CON> );
SQL> COMMIT;
SQL>
```

Mostrando a estrutura da tabela de usuários:

SHOW TABLE USUARIO;

```

c:\ Firebird ISQL Tool
SQL> SHOW TABLE USUARIO;
USU_CODIGO          INTEGER Not Null
USU_NOME            UARCHAR(40) Not Null
USU_LOGIN           UARCHAR(15) Not Null
USU_SENHA           CHAR(7) Not Null
CONSTRAINT PK_USUARIO:
  Primary key (USU_CODIGO)
SQL>

```

Listando todas as tabelas do banco:

SHOW TABLES;

```

c:\ Firebird ISQL Tool
SQL> SHOW TABLES;
GRUPO              MATERIAL
USUARIO
SQL>

```

Saindo da ferramenta Firebird ISQL Tool:

EXIT;

## 11.2. Aplicação do Projeto Piloto

### 11.2.1. Criando a Janela Principal

Sempre que uma nova seção do Delphi é iniciada, sua área de trabalho está pronta para receber um novo projeto. A *Form1* que está sendo exibida será janela principal da aplicação piloto. Assim sendo, vamos configurar suas propriedades para que atenda a essas necessidades.

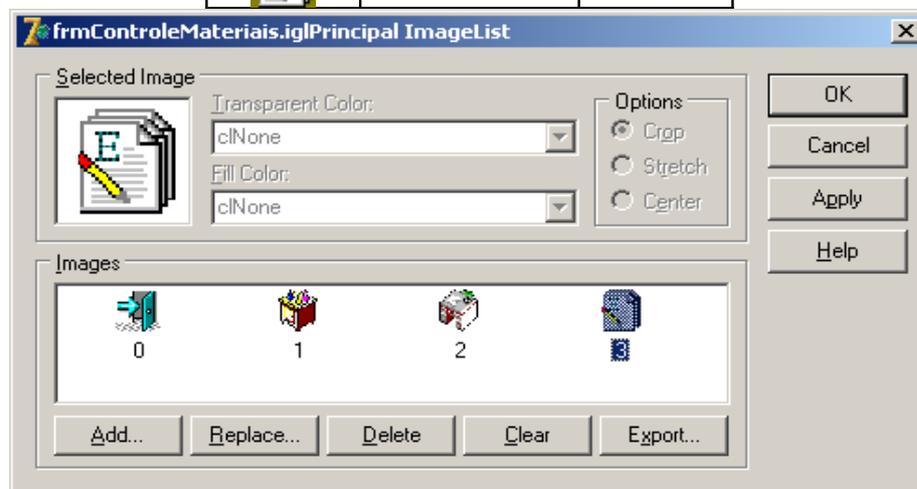
- Clique na janela **Form1** para selecioná-la;
- No **Object Inspector**, altere suas propriedades conforme ilustra a tabela seguinte:

Propriedade	Valor
BorderIcons	
biMaximize	False
BorderStyle	bsSizeable
Caption	Controle de Materiais
Name	frmPrincipal
Position	poScreenCenter
WindowState	wsMaximized

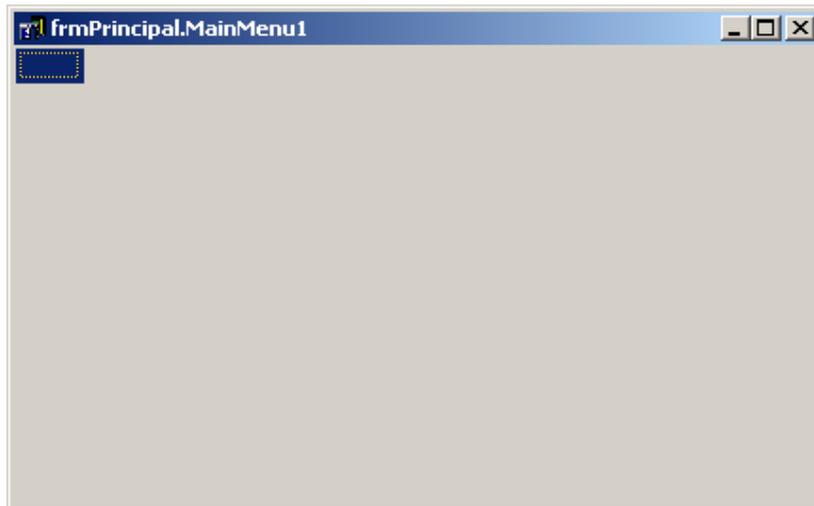
- Salve o trabalho clicando em **File / Save Project As...** ou clicando sobre o botão  (Save All) que se encontra na barra de ferramentas. Na caixa de diálogo "**Save Unit1 As**", selecione a pasta (diretório) `c:\piloto`. Na caixa de texto *Nome do Arquivo*, digite **untMaterial** e clique sobre o botão **Salvar**;
- Uma nova caixa de diálogo será exibida, solicitando que seja salvo o projeto. Na caixa de texto **Nome do Arquivo**, digite **Material** e clique na opção **Salvar**;
- Execute a aplicação (**F9**) e observe o resultado;

- Selecione a janela **frmPrincipal**;
- Clique na paleta **Win32** da barra de ferramentas do Delphi;
- Clique sobre o componente **ImageList** para selecioná-lo, e em seguida clique em qualquer área interna da janela *frmPrincipal* para inseri-lo nessa janela;
- Altere o valor da propriedade **Height** para **32**;
- Altere o valor da propriedade **Name** para **imgPrincipal**;
- Altere o valor da propriedade **Width** para **32**;
- Dê um duplo-clique sobre o componente *ImageList* que foi inserido na janela principal do aplicativo e insira as imagens na mesma ordem descrita na tabela abaixo. As imagens estão na pasta (diretório) *c:\piloto\imagens*.

Imagem	Nome	Índice
	sair_1.bmp	0
	grupo.bmp	1
	material.bmp	2
	sobre_1.bmp	3



- Clique no botão OK e salve o trabalho, clicando sobre o botão  (**Save All**);
- Selecione o janela **frmPrincipal**;
- Clique na paleta **Standard** e insira na janela *frmPrincipal* o componente MainMenu ;
- Altere a propriedade **Images** para **imgPrincipal**, para ligar as imagens usadas pelos itens do menu com as imagens contidas no ImageList;
- Altere a propriedade **Name** para **mmuPrincipal**;
- Dê um duplo-clique sobre o componente MainMenu que foi inserido no *frmPrincipal*. Uma janela denominada "frmPrincipal.MainMenu1" será aberta em sua área de trabalho;



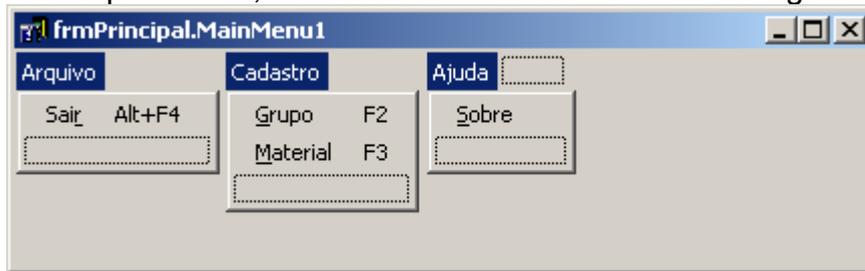
- Observe uma área escurecida, contornada por um traço pontilhado, localizada na parte superior dessa janela. É aí que se inicia a inserção dos itens que comporão o menu. Assim sendo, digite na propriedade **Caption**, no *Object Inspector*, a palavra **&Arquivo** e aperte a tecla **[Enter]** e na propriedade **Name** digite **mmuArquivo** e aperte a tecla **[Enter]**;
- Ao executar o item anterior, a palavra *Arquivo* será inserida na barra de menus e um novo retângulo pontilhado será posicionado à sua direita. Clique sobre o referido retângulo para selecioná-lo, e na propriedade **Caption** do *Object Inspector*, digite a palavra **&Cadastro** e aperte a tecla **[Enter]**. Altere a propriedade **Name** para **mmuCadastro** e aperte a tecla **[Enter]**;
- Repita o procedimento anterior e insira o item **A&juda**. Ao final deste processo, o construtor de menus deve ter a seguinte aparência:



- Na janela do *frmPrincipal.MainMenu1* clique sobre o menu *Arquivo*. Observe que um retângulo contornado com uma linha pontilhada posicionado logo abaixo da palavra *Arquivo*. Selecione esse retângulo, altere a propriedade **Caption** para **Sai&r**, altere a propriedade **Hint** para **Sair|Encerra a execução do Controle de Materiais**, altere a propriedade **ImageIndex** para **0**, altere a propriedade **Name** para **mmuArqSair**, altere a propriedade **Shortcut** para **Alt+F4** (digitar);
- Na janela do *frmPrincipal.MainMenu1* clique sobre o menu *Cadastro*. Selecione o retângulo contornado com uma linha pontilhada é posicionado logo abaixo da palavra *Cadastro*, altere a propriedade **Caption** para **&Grupo**, altere a propriedade **Hint** para **Cadastro de Grupos|Chama o Cadastro de Grupos**, altere a propriedade **ImageIndex** para **1**, altere a propriedade **Name** para **mmuCadGrupo**, altere a propriedade **Shortcut** para **F2**;
- Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Grupo*), altere a propriedade **Caption** para **&Material**, altere a propriedade **Hint** para **Cadastro de Materiais|Chama o Cadastro de Materiais**, altere a propriedade **ImageIndex** para **2**, altere a propriedade **Name** para **mmuCadMaterial**, altere a propriedade **Shortcut** para **F3**;
- Na janela do *frmPrincipal.MainMenu1* clique sobre o menu *Ajuda*. Selecione o retângulo contornado com uma linha pontilhada posicionado logo abaixo da palavra *Ajuda*, altere a propriedade **Caption** para **&Sobre**, altere a propriedade

**Hint** para **Sobre|Chama o Sobre do Aplicativo**, altere a propriedade **ImageIndex** para **3**, altere a propriedade **Name** para **mmuAjuSobre**;

- Ao final destes processos, o construtor de menus deve ter a seguinte aparência:

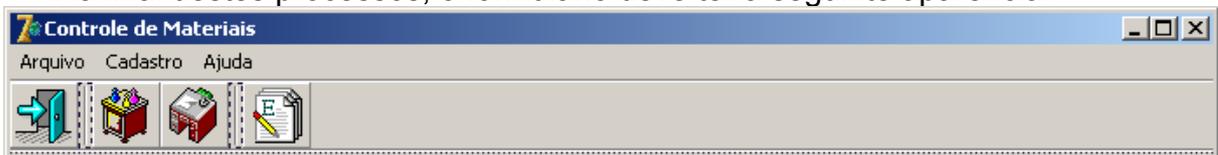


- Feche a janela *frmPrincipal.MainMenu1*, clique sobre o botão  (**Save All**) e execute (**F9**) o programa para ver o resultado;
- Selecione a janela **frmPrincipal**;
- Clique na paleta **Win32** e insira na janela *frmPrincipal* o componente **ToolBar** ;
- Ajuste suas propriedades de acordo com a tabela seguinte:

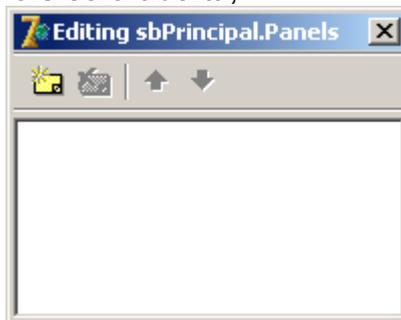
Propriedade	Valor	Descrição
EdgeBorders	ebLeft = Fase ebTop = True ebRight = False ebButton = True	Define se a borda superior, inferior, esquerda e direita serão exibidas.
Flat	True	Define se o botão ficará somente realçado ao passarmos o mouse sobre ele.
Height	42	Determina a altura da barra de ferramentas.
Images	imgPrincipal	Informa o nome do componente que possui as imagens a serem exibidas pelos botões.
Name	tbPrincipal	Define o nome que o componente terá e que será usado no código quando for necessário se referir ao mesmo.
ShowHint	True	Ativa a ajuda contextual quando o ponteiro do mouse apontar para um de seus botões.

- Salve o trabalho, clicando sobre o botão  (**Save All**) e execute (**F9**) o programa para ver o resultado;
- Selecione a janela **frmPrincipal**;
- Clique sobre a barra de ferramentas para selecioná-la;
- Clique com o botão direito do mouse sobre a barra de ferramentas. Ao abrir o menu suspenso, clique sobre a opção **New Button**. Um botão contendo a imagem *sair\_1.bmp* será inserido na barra de ferramentas;
- Clique com o botão direito do mouse sobre a barra de ferramentas. Ao abrir o menu suspenso, clique sobre a opção **New Separator**. Um separador de botões será inserido na barra de ferramentas;
- Clique com o botão direito do mouse sobre a barra de ferramentas. Ao abrir o menu suspenso, clique sobre a opção **New Button**. Um botão contendo a imagem *grupo.bmp* será inserido na barra de ferramentas;
- Clique com o botão direito do mouse sobre a barra de ferramentas. Ao abrir o menu suspenso, clique sobre a opção **New Button**. Um botão contendo a imagem *material.bmp* será inserido na barra de ferramentas;

- Clique com o botão direito do mouse sobre a barra de ferramentas. Ao abrir o menu suspenso, clique sobre a opção **New Separator**. Um separador de botões será inserido na barra de ferramentas;
- Clique com o botão direito do mouse sobre a barra de ferramentas. Ao abrir o menu suspenso, clique sobre a opção **New Button**. Um botão contendo a imagem *sobre\_1.bmp* será inserido na barra de ferramentas;
- Selecione o *primeiro botão* da barra de ferramentas, altere a propriedade **Menultem** para **mmuArqSair** e a propriedade **Name** para **tbtSair**;
- Selecione o *segundo botão* da barra de ferramentas, altere a propriedade **Menultem** para **mmuCadGrupo** e a propriedade **Name** para **tbtCadGrupo**;
- Selecione o *terceiro botão* da barra de ferramentas, altere a propriedade **Menultem** para **mmuCadMaterial** e a propriedade **Name** para **tbtCadMaterial**;
- Selecione o *quarto botão* da barra de ferramentas, altere a propriedade **Menultem** para **mmuAjuSobre** e a propriedade **Name** para **tbtSobre**;
- Ao final destes processos, o formulário deve ter a seguinte aparência:



- Salve o trabalho, clicando sobre o botão  (**Save All**) e execute (F9) o programa para ver o resultado;
- Selecione a janela **frmPrincipal**;
- Clique na paleta **Win32** e insira na janela *frmPrincipal* o componente **StatusBar** ;
- Altere a propriedade **AutoHint** para **True**;
- Altere a propriedade **Name** para **sbPrincipal**;
- Dê um duplo-clique na barra de status inserida. Uma janela definida como **Editing sbPrincipal.Panels** será aberta;



- Clique sobre o botão **Add New**  para inserir uma divisão no *panel* da barra de status. Observe que será inserido na janela de edição um *panel* de nome **0 – TstatusPanel**. Mantenha-o selecionado e, na propriedade **Width** do *Object Inspector*, digite **600** para alterar sua largura;
- Clique novamente sobre o botão **Add New**  para inserir uma divisão no *panel* da barra de status. Observe que será inserido na janela de edição um *panel* de nome **1 – TstatusPanel**;
- Selecione a janela **frmPrincipal**;
- Clique na guia **Events** do **Object Inspector**;
- Dê um duplo-clique sobre a caixa de texto do evento **OnActivate** para abrir sua procedure;
- Insira as linhas que estão em negrito na listagem seguinte:  

```
procedure TfrmPrincipal.FormActivate(Sender: TObject);
```

```
begin
  sbPrincipal.Panels[1].Text :=
    FormatDateTime(' dddd", " dd " de " mmmm " de " yyyy', Now);
end;
```

- Na seção **var** do código acrescente as palavras em negrito do código abaixo:

```
var
  frmPrincipal: TfrmPrincipal;
  bolLogin: Boolean = false;
```

- Selecione o janela **frmPrincipal**;
- Insira o componente **MDODatabase** , localizado na paleta **Mercury**;
- Altere as propriedades de **MDODatabase**, conforme a tabela seguinte:

Propriedade	Valor
Databasename	.\MATERIAL.FDB
LoginPrompt	False
Name	dbMaterial
Params	user_name=SYSDBA password=masterkey
Connected	True

- Insira o componente **MDOTransaction** , localizado na paleta **Mercury**;
- Altere as propriedades de **MDOTransaction**, conforme a tabela seguinte:

Propriedade	Valor
DefaultDatabase	dbMaterial
Name	tstMaterial
Active	True
AutoCommit	True

- Insira o componente **MDOQuery** , localizado na paleta **Mercury**;
- Altere as propriedades de **MDOQuery**, conforme a tabela seguinte:

Propriedade	Valor
Database	dbMaterial
Name	qryLogin
SQL	select * from usuario where usu_login = :usu_login and usu_senha = :usu_senha

- Selecione a janela **frmPrincipal**;
- Clique na guia **Events** do **Object Inspector**;
- Dê um duplo-clique sobre a caixa de texto do evento **OnCreate** para abrir sua procedure;
- Insira as linhas que estão em negrito na listagem seguinte:

```
procedure TfrmPrincipal.FormCreate(Sender: TObject);
var
  BarraTarefas: HWND; {Barra Iniciar}
  intAltBarraTarefas: Integer;
  tmRectBarraTarefas: TRect;
begin
  //localiza o Handle da janela iniciar
  BarraTarefas := FindWindow('Shell_TrayWnd', nil);
  //Pega o "retângulo" que envolve a barra e sua altura
  GetWindowRect(BarraTarefas, tmRectBarraTarefas);
  intAltBarraTarefas := tmRectBarraTarefas.Bottom - tmRectBarraTarefas.Top;
```

```
//Definindo posicionamento, altura e largura do formulário principal
frmPrincipal.Left := 0;
frmPrincipal.Top := 0;
frmPrincipal.Height := Screen.Height - intAltBarraTarefas + 2;
frmPrincipal.Width := Screen.Width;
end;
```

- Salve o trabalho, clicando sobre o botão  (**Save All**) e execute (**F9**) o programa para ver o resultado;

## 11.2.2. Criando a Janela de Login

- Agora vamos criar a janela de login, clique em **File / New / Form**;
- Na janela do *Object Inspector*, altere suas propriedades conforme tabela abaixo:

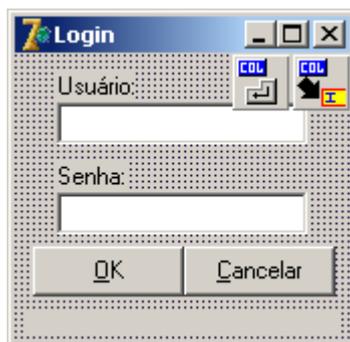
Propriedade	Valor	Descrição
BorderIcons	False (todas as opções)	Define os ícones a serem exibidos na barra de título da janela.
BorderStyle	bsSingle	Tipo da borda.
Caption	Login	Título da janela.
Height	178	Altura da janela.
Keypreview	True	Habilita a captura geral de teclas.
Name	frmLogin	Nome da janela.
Position	poScreenCenter	Posição da janela na tela.
Width	174	Largura da janela.

- Salve o trabalho clicando **File / Save As** ou clicando sobre o botão  (**Save**) que se encontra na barra de ferramentas. Na caixa de diálogo "Save Unit 1 As", selecione a pasta (diretório) **c:\piloto**. Na caixa de texto *Nome do Arquivo*, digite **untLogin** e clique sobre o botão **Salvar**;
- Selecione a janela **frmLogin**;
- Use a tecla **F12** para chamar o **Editor de Código**;
- Na seção **var** do código acrescente a linha em negrito do código abaixo:

```
var
    frmLogin: TfrmLogin;
    bolFechar: Boolean = false;
```

```
implementation
```

- Use a tecla **F12** para chamar o **Formulário**;
- Insira três componentes **Label A**, dois componentes **Edit**  e dois componentes **Button** , localizados na paleta **Standard**;
- Posicione os componentes **Label**, os componentes **Edit** e os componentes **Button**, conforme a figura seguinte;

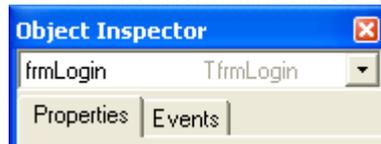


- Altere as propriedades dos componentes **Label** e dos componentes **Edit**, conforme a tabela seguinte:

Componente	Propriedade	Valor
Label1	Caption	Usuário:
	Name	lblUsuario
	Left	21
	Top	9
Edit1	CharCase	ecUpperCase
	Hint	Digite seu nome de login
	Name	edtUsuario
	Text	(vazio)
	Left	21
Label2	Caption	Senha:
	Name	lblSenha
	Left	21
Edit2	CharCase	ecLowerCase
	Hint	Digite a sua senha (máximo de 7 dígitos)
	MaxLength	7
	Name	edtSenha
	PassWordChar	*
Button1	Caption	&OK
	Name	btnOK
	Left	9
Button2	Caption	&Cancelar
	Name	btnCancelar
	Left	84
Label3	Alignment	taCenter
	AutoSize	False
	Caption	(vazio)
	Left	6
	Name	lblMsg
	Top	128

Componente	Propriedade	Valor
	Width	154

- Insira um componente **ColEnter** , e um componente **ColFocus** , localizados na paleta **Col**;
- Selecione o componente **ColEnter** e no **Object Inspector** altere o valor da propriedade **Ativo** para **True**;
- Selecione o componente **ColFocus** e no **Object Inspector** altere o valor da propriedade **AtivaColor** para **True**;
- No **Object Inspector**, selecione o formulário **frmLogin**;



- Agora selecione a aba **Events** e em seguida o evento **OnCloseQuery**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnCloseQuery e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmLogin.FormCloseQuery(Sender: TObject; var
  CanClose: Boolean);
begin
  CanClose := bolFechar;
end;
```

- Selecione **File / Use Unit... / untMaterial / OK**;
- Dê um duplo-clique sobre o botão **OK** e insira as linhas em negrito da listagem seguinte:

```
procedure TfrmLogin.btnOKClick(Sender: TObject);
begin
  frmPrincipal.qryLogin.Active := false;
  frmPrincipal.qryLogin.ParamByName('usu_login').AsString :=
    UpperCase(Trim(edtUsuario.Text));
  frmPrincipal.qryLogin.ParamByName('usu_senha').AsString :=
    LowerCase(Trim(edtSenha.Text));
  frmPrincipal.qryLogin.Active := true;
  if (frmPrincipal.qryLogin.RecordCount = 1) then
  begin
    bolFechar := true;
    frmLogin.Close;
  end
  else
  begin
    lblMsg.Caption := 'Dados inválidos!';
    edtUsuario.Clear;
    edtSenha.Clear;
    edtUsuario.SetFocus;
  end;
end;
```

- Dê um duplo-clique sobre o botão **Cancelar** e insira a linha em negrito da listagem seguinte:

```
procedure TfrmLogin.btnCancelClick(Sender: TObject);
begin
  Application.Terminate;
end;
```

- Selecione **View / Forms / frmPrincipal / OK**;

- Selecione a janela **frmPrincipal**;
- Clique na guia **Events** do **Object Inspector**;
- Dê um duplo-clique sobre a caixa de texto do evento **OnActivate** para abrir sua procedure;
- Insira as linhas que estão em negrito na listagem seguinte:

```

procedure TfrmPrincipal.FormActivate(Sender: TObject);
begin
  sbPrincipal.Panels[1].Text :=
    FormatDateTime(' dddd", " dd " de " mmmm " de " yyyy', Now);
  if ((bolLogin = false) and (frmLogin = nil)) then
    begin
      //Criando o formulário de Login
      Application.CreateForm(TfrmLogin, frmLogin);
      bolLogin := true;
      frmLogin.showmodal; //Exibindo o formulário de Login
    end;
  //Colocando a aplicação no topo das aplicações
  SetForegroundWindow(Handle);
end;

```

- Selecione **File / Use Unit... / untLogin / OK**;
  - Selecione **Project / Options / Aba Forms / frmLogin (em Auto-create forms) / clique no botão seta para direita > / OK**;
- Neste passo estamos desabilitando a criação automática do formulário de Login, a criação automática de formulário ocorre quando executamos a aplicação. Todos os formulários criados automaticamente ficam residentes em memória, o que causa um consumo desnecessário da mesma.
- Salve o trabalho, clicando sobre o botão  (**Save All**) e o execute (**F9**);

### 11.2.3. Criando a Janela de Abertura

- Selecione o menu **File / New / Form** para que um novo formulário seja criado;
- Altere suas propriedades de acordo com as informações da tabela a seguir:

Propriedade	Valor
BorderIcons	Configure todas as suas subpropriedades para False.
BorderStyle	bsNone
Caption	(Não há necessidade de alterá-lo, já que a barra de título dessa janela não será exibida na tela).
FormStyle	fsStayOnTop (Coloca essa janela fique na frente das demais).
Height	326
Name	frmAbertura
Position	poScreenCenter
Width	408

- Salve o trabalho clicando **File / Save As** ou clicando sobre o botão  (**Save**) que se encontra na barra de ferramentas. Na caixa de diálogo "Save Unit 1 As", selecione a pasta (diretório) **d:\piloto**. Na caixa de texto *Nome do Arquivo*, digite **untAbertura** e clique sobre o botão **Salvar**;
- Insira na janela *frmAbertura* um componente **Image** que se encontra na paleta **Additional**;

- Altere a propriedade **Align** para **alClient**, para fazer com que a figura ocupe toda a tela;
- Altere a propriedade **Name** para **imgAbertura**;
- Selecione o objeto *imgAbertura* e insira em sua propriedade **Picture** a figura **logo\_abertura.jpg** que se encontra na pasta *d:\piloto\imagens*;
- Altere a propriedade **Stretch** para **True**;
- Selecione **Project / Options / Aba Forms / frmAbertura (em Auto-create forms)** / clique no botão seta para direita  / OK;
- Uma vez configurada a janela de abertura do sistema, vamos fazer uma pequena alteração na fonte do projeto que contém o **Controle de Materiais** que se encontra no arquivo **material.dpr**. Para abri-lo, clique no menu **Project / View Source**. A fonte contendo o projeto Controle de Materiais será exibida na tela, conforme ilustra a listagem seguinte:

```
program Material;

uses
  Forms,
  untMaterial in 'untMaterial.pas' {frmPrincipal},
  untLogin in 'untLogin.pas' {frmLogin},
  untAbertura in 'untAbertura.pas' {frmAbertura};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TfrmPrincipal, frmPrincipal);
  Application.Run;
end.
```

- Uma vez que o formulário de abertura não mais é criado automaticamente pelo Delphi, devemos criá-lo manualmente, o que será feito antes da criação do formulário principal (frmPrincipal), já que o formulário de abertura deve ser exibido assim que o programa inicie seu processo de carga. Assim sendo, insira logo abaixo do comando *Application.Initialize* as linhas que estão em negrito na listagem seguinte:

```
program Material;

uses
  Forms,
  SysUtils,
  untMaterial in 'untMaterial.pas' {frmPrincipal},
  untLogin in 'untLogin.pas' {frmLogin},
  untAbertura in 'untAbertura.pas' {frmAbertura};

{$R *.res}

begin
  Application.Initialize;
  Application.Title := 'Controle de Materiais';
  //Formulário de Abertura
```

```

frmAbertura := TfrmAbertura.Create(Application);
frmAbertura.Show;
frmAbertura.Refresh;
//Formulário Principal
Application.CreateForm(TfrmPrincipal, frmPrincipal);
//Tempo de exibição do formulário de abertura
sleep(1500); //declarar na seção uses: SysUtils
//Fechando e liberando da memória o formulário de abertura
frmAbertura.Free;
Application.Run;

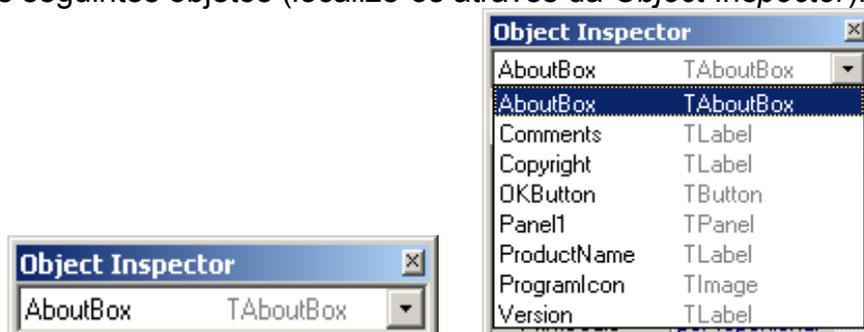
```

end.

- Salve o trabalho, clicando sobre o botão  (**Save All**) e execute (**F9**) o programa para ver o resultado;

#### 11.2.4. Criando a Janela Sobre

- Selecione o menu **File / New / Other**, aparecerá a janela da *New Items*, clique na página *Forms*, clique no objeto entitulado *About box* e clique no botão **OK**;
- Salve o trabalho clicando **File / Save As** ou clicando sobre o botão  (**Save**) que se encontra na barra de ferramentas. Na caixa de diálogo "Save Unit 1 As", selecione a pasta (diretório) **d:\piloto**. Na caixa de texto *Nome do Arquivo*, digite **untSobre** e clique sobre o botão **Salvar**;
- Altere os seguintes objetos (localize-os através da *Object Inspector*):



##### AboutBox (TAboutBox):

Propriedade	Valor	Descrição
Caption	Sobre o sistema	Label escrito na tarja superior da janela
Name	frmSobre	Nome interno do objeto

##### ProgramIcon (TImage):

Propriedade	Valor	Descrição
Name	imgSobre	Nome interno do objeto
Picture	insira a figura <b>material.bmp</b> que se encontra na pasta <i>d:\piloto\imagens</i>	Objeto imagem

##### ProductName (TLabel):

Propriedade	Valor	Descrição
-------------	-------	-----------

Caption	Controle de Materiais	Label do objeto
Font	Ms Sans Serif, Negrito e Itálico, 12, Azul-marinho	Tipo de letra a ser mostrada no objeto
Name	lblNomeProduto	Nome interno do objeto

#### Version (TLabel):

Propriedade	Valor	Descrição
Caption	Versão Educacional	Label do objeto
Font	Ms Sans Serif, Negrito e Itálico, 8, Verde	Tipo de letra a ser mostrada no objeto
Name	lblVersao	Nome interno do objeto

#### Copyright (TLabel):

Propriedade	Valor	Descrição
Caption	Copyright © 2006 – Prof. Alexandre Bendlin	Label do objeto
Font	Ms Sans Serif, Normal, 8, Azul-marinho	Tipo de letra a ser mostrada no objeto
Name	lblCopyright	Nome interno do objeto

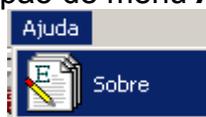
**Nota:** Para conseguir um © pressione ALT+184 e um ® pressione ALT+169.

#### Comments (TLabel):

Propriedade	Valor	Descrição
Caption	Este aplicativo foi criado para fins educacionais.	Label do objeto
Height	13	Altura do componente
Font	Ms Sans Serif, Normal, 8, Preto	Tipo de letra a ser mostrada no objeto
Name	lblComentarios	Nome interno do objeto
Width	227	Largura do componente

- Selecione a janela **frmSobre**;
- Clique na guia **Events** do *Object Inspector* e dê um duplo-clique na caixa de texto da propriedade **OnClose**;
- Insira em sua procudere as linhas que estão em negrito da listagem seguinte:

```
procedure TfrmSobre.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action := caFree;
  frmSobre := nil;
end;
```
- Selecione **Project / Options / Aba Forms / frmSobre (em Auto-create forms) / clique no botão seta para direita**  / **OK**;
- Selecione **View / Forms / frmPrincipal / OK**;
- Selecione em **frmPrincipal** a opção de menu **Ajuda** e depois clique em **Sobre**;



- Insira em sua procudere as linhas que estão em negrito da listagem seguinte:

```
procedure TfrmPrincipal.mmuAjuSobreClick(Sender: TObject);
```

```

begin
  if (frmSobre = nil) then
    Application.CreateForm(TfrmSobre, frmSobre);
    frmSobre.showmodal;
end;

```

- Selecione **File / Use Unit... / untSobre / OK**;
- Selecione em **frmPrincipal** a opção de menu **Arquivo** e depois em **Sair**;



- Insira em sua procedure as linhas que estão em negrito da listagem seguinte:
- ```

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

```

- Selecione a janela **frmPrincipal**;
- Clique na guia **Events** do *Object Inspector* e dê um duplo-clique na caixa de texto da propriedade **OnCloseQuery**;

- Insira em sua procedure as linhas que estão em negrito da listagem seguinte:

```

procedure TfrmPrincipal.FormCloseQuery(Sender: TObject;
  var CanClose: Boolean);
var
  intResposta: integer;
begin
  intResposta := Application.MessageBox('Deseja sair da aplicação?',
    'Confirmação', mb_YesNo + mb_DefButton2 + mb_IconQuestion);
  if (intResposta = IDNO) then CanClose := False;
end;

```

- Salve o trabalho, clicando sobre o botão  (**Save All**) e execute (**F9**) o programa para ver o resultado;

## 11.2.8. Criando a Janela Relatório Dinâmico

- Crie uma nova janela clicando em **File / New / Form**;
- Altere suas propriedades conforme ilustra a tabela seguinte:

| Propriedade | Valor                                                                                                  |
|-------------|--------------------------------------------------------------------------------------------------------|
| BorderIcons | Marque com <b>True</b> somente a subpropriedade <i>biSystemMenu</i> e todas as outras com <b>False</b> |
| BorderStyle | bsDialog                                                                                               |
| Caption     | Gerador de Relatórios Dinâmicos                                                                        |
| Height      | 370                                                                                                    |
| Name        | frmRelatorioDinamico                                                                                   |
| Position    | poScreenCenter                                                                                         |
| Width       | 406                                                                                                    |

- Salve o trabalho clicando **File / Save As** ou clicando sobre o botão  (**Save**) que se encontra na barra de ferramentas. Na caixa de diálogo "Save Unit 1 As", selecione a pasta (diretório) **d:\piloto**. Na caixa de texto *Nome do Arquivo*, digite **untRelatorioDinamico** e clique sobre o botão **Salvar**;

- Selecione o janelo **frmRelatorioDinamico**;
- Use a tecla **F12** para chamar o **Editor de Código**;
- Na seção **uses** do código acrescente as palavras em negrito do código abaixo:

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Buttons, ComCtrls,
  CheckLst, ExtCtrls, MDOQuery, DB, QRCtrls;
```

O que você acabou de realizar foi a adição das bibliotecas: MDOQuery e DB.

- Na seção **type** acrescente as linhas em negrito do código abaixo:

```
type
  TfrmRelatorioDinamico = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
    qryTabela: TMDOQuery;
    strBarraTitulo: string;
    Pagina: Integer;
    QrLabel: array[1..8] of TQrLabel;
    QrDbImage: array[1..8] of TqrDbImage;
    QrDbText: array[1..8] of TQrDbText;
    Desc,Campo: array[1..8] of String;
    Largura: array[1..8] of Integer;
    function PosCampo(strCampo: String): Integer;
  end;
```

```
var
  frmRelatorioDinamico: TfrmRelatorioDinamico;
```

```
implementation
```

```
{ $R *.dfm }
```

```
function TfrmRelatorioDinamico.PosCampo(strCampo: String): Integer;
var
  intCampo: integer;
begin
  Result := 0;
  for intCampo := 1 to 8 do
  begin
    if (strCampo = Desc[intCampo]) then
    begin
      Result := intCampo;
      Break;
    end;
  end;
end;
```

- Use a tecla **F12** para chamar o **formulário frmRelatorioDinamico**;
- Clique na paleta **Standard** e insira na janela *frmRelatorioDinamico* o componente **Panel** ;
- No **Object Inspector** selecione a propriedade **Align** e altere seu valor para **alTop**;
- Ainda no **Object Inspector**, selecione a propriedade **BevelInner** e altere seu valor para **bvRaised**;
- Continuando no **Object Inspector**, selecione a propriedade **BevelOuter** e altere seu valor para **bvLowered**;

- Ainda no **Object Inspector**, selecione a propriedade **Caption** e apague o texto que ela contém;
- Continuando no **Object Inspector**, selecione a propriedade **Color** e altere seu valor para **clWhite**;
- Ainda no **Object Inspector**, selecione a propriedade **Height** e altere seu valor para **50**;
- Continuando no **Object Inspector**, selecione a propriedade **Name** e a altere para **pnITitulo**;
- Clique na paleta **Standard** e coloque em **pnITitulo** o componente **Label A**;
- No **Object Inspector** selecione a propriedade **Caption** e altere seu valor para **Relatório Dinâmico**;
- Continuando no **Object Inspector**, selecione a propriedade **Font** e altere os seguintes itens:
  - ✓ Fonte: Tahoma;
  - ✓ Estilo da Fonte: Negrito;
  - ✓ Tamanho: 18;
  - ✓ Cor: Azul-marinho.
- No **Object Inspector**, selecione a propriedade **Left** e altere seu valor para **8**;
- Continuando no **Object Inspector**, selecione **Name** e altere seu valor para **lblTitulo**;
- Ainda no **Object Inspector**, selecione a propriedade **Top** e altere seu valor para **10**;
- Siga as orientações da tabela abaixo, depositando todos os componentes indicados pela mesma sobre o formulário *frmRelatorioDinamico*:

| <b>Componente</b> | <b>Propriedade</b> | <b>Valor</b>             |
|-------------------|--------------------|--------------------------|
| Label             | Caption            | Titulo:                  |
|                   | Left               | 3                        |
|                   | Name               | lblTituloRelatorio       |
|                   | Top                | 57                       |
| Edit              | Left               | 3                        |
|                   | Name               | edtTitulo                |
|                   | Text               | (vazio)                  |
|                   | Top                | 72                       |
| Label             | Width              | 392                      |
|                   | Caption            | Campo(s):                |
|                   | Left               | 3                        |
|                   | Name               | lblCampo                 |
| Label             | Top                | 98                       |
|                   | Caption            | Campo(s) Selecionado(s): |
|                   | Left               | 213                      |
|                   | Name               | lblCampoSelecionado      |
| ListBox           | Top                | 98                       |
|                   | Height             | 184                      |
|                   | Left               | 3                        |
|                   | Name               | lbtCampo                 |
|                   | Top                | 114                      |

| <b>Componente</b> | <b>Propriedade</b> | <b>Valor</b>                                                                        |
|-------------------|--------------------|-------------------------------------------------------------------------------------|
|                   | Width              | 181                                                                                 |
| SpeedButton       | Caption            | (vazio)                                                                             |
|                   | Glyph              | C:\Arquivos de programas\Arquivos comuns\ Borland Shared\Images\Buttons\ARROW1R.BMP |
|                   | Height             | 22                                                                                  |
|                   | Left               | 188                                                                                 |
|                   | Name               | btnIncluirCampo                                                                     |
|                   | Top                | 177                                                                                 |
|                   | Width              | 23                                                                                  |
| SpeedButton       | Caption            | (vazio)                                                                             |
|                   | Glyph              | C:\Arquivos de programas\Arquivos comuns\ Borland Shared\Images\Buttons\ARROW1L.BMP |
|                   | Height             | 22                                                                                  |
|                   | Left               | 188                                                                                 |
|                   | Name               | btnRemoverCampo                                                                     |
|                   | Top                | 207                                                                                 |
|                   | Width              | 23                                                                                  |
| ListBox           | Height             | 158                                                                                 |
|                   | Left               | 213                                                                                 |
|                   | Name               | ltbCampoSelecioneado                                                                |
|                   | Top                | 114                                                                                 |
|                   | Width              | 181                                                                                 |
| Label             | Caption            | Ordem de Impressão:                                                                 |
|                   | Left               | 225                                                                                 |
|                   | Name               | lblOrdemImpressao                                                                   |
|                   | Top                | 280                                                                                 |
| SpeedButton       | Caption            | (vazio)                                                                             |
|                   | Glyph              | C:\Arquivos de programas\Arquivos comuns\ Borland Shared\Images\Buttons\ARROW1U.BMP |
|                   | Height             | 22                                                                                  |
|                   | Left               | 328                                                                                 |
|                   | Name               | btnCampoAcima                                                                       |
|                   | Top                | 276                                                                                 |
|                   | Width              | 23                                                                                  |
| SpeedButton       | Caption            | (vazio)                                                                             |
|                   | Glyph              | C:\Arquivos de programas\Arquivos comuns\ Borland Shared\Images\Buttons\ARROW1D.BMP |
|                   | Height             | 22                                                                                  |
|                   | Left               | 355                                                                                 |

| <b>Componente</b> | <b>Propriedade</b> | <b>Valor</b>   |
|-------------------|--------------------|----------------|
|                   | Name               | btnCampoAbaixo |
|                   | Top                | 276            |
|                   | Width              | 23             |

- Clique na paleta **Standard** e insira na janela *frmRelatorioDinamico* o componente **Panel** ;
- No **Object Inspector** selecione a propriedade **Align** e altere seu valor para **alBottom**;
- Ainda no **Object Inspector**, selecione a propriedade **BevelInner** e altere seu valor para **bvRaised**;
- Continuando no **Object Inspector**, selecione a propriedade **BevelOuter** e altere seu valor para **bvLowered**;
- Ainda no **Object Inspector**, selecione a propriedade **Caption** e apague o texto que ela contém;
- Ainda no **Object Inspector**, selecione a propriedade **Height** e altere seu valor para **41**;
- Continuando no **Object Inspector**, selecione a propriedade **Name** e a altere para **pnlBotoes**;
- Clique na paleta **Additional** e coloque em *pnlBotoes* o componente **BitBtn** ;
- No **Object Inspector**, selecione a propriedade **Kind** e altere seu valor para **bkOK**;
- Continuando no **Object Inspector**, selecione a propriedade **Caption** e altere seu valor para **Gerar**;
- Ainda no **Object Inspector**, selecione a propriedade **Left** e altere seu valor para **207**;
- Ainda no **Object Inspector**, selecione a propriedade **ModalResult** e altere seu valor para **mrNone**;
- Ainda no **Object Inspector**, selecione a propriedade **Name** e altere seu valor para **bbtGerar**;
- Continuando no **Object Inspector**, selecione a propriedade **Top** e altere seu valor para **9**;
- Ainda no **Object Inspector**, selecione a propriedade **Width** e altere seu valor para **90**;
- Clique na paleta **Additional** e coloque em *pnlBotoes* o componente **BitBtn** ;
- No **Object Inspector**, selecione a propriedade **Kind** e altere seu valor para **bkClose**;
- Continuando no **Object Inspector**, selecione a propriedade **Caption** e altere seu valor para **Fechar**;
- Ainda no **Object Inspector**, selecione a propriedade **Left** e altere seu valor para **300**;
- Ainda no **Object Inspector**, selecione a propriedade **Name** e altere seu valor para **bbtFechar**;
- Continuando no **Object Inspector**, selecione a propriedade **Top** e altere seu valor para **9**;
  
- Ainda no **Object Inspector**, selecione a propriedade **Width** e altere seu valor para **90**;
  
- No **Object Inspector**, selecione o formulário **frmRelatorioDinamico**;



- Agora selecione a aba **Events** e em seguida o evento **OnShow**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnShow e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmRelatorioDinamico.FormShow(Sender: TObject);
var
  intCampo, intCampoTabela: integer;
begin
  Self.Caption := strBarraTitulo;
  Pagina := 0;
  //Matriz com Nome de Campo na Tabela
  intCampo := 0;
  for intCampoTabela := 0 To (qryTabela.FieldCount - 1) do
    if (qryTabela.Fields[intCampoTabela].Visible) then
      begin
        Campo[intCampo + 1] :=
          qryTabela.Fields[intCampoTabela].FieldName;
        intCampo := intCampo + 1;
      end;

    //Matriz Com Descrição das Colunas do Relatório
    intCampo := 0;
    for intCampoTabela := 0 To (qryTabela.FieldCount - 1) do
      if (qryTabela.Fields[intCampoTabela].Visible) then
        begin
          Desc[intCampo + 1] :=
            qryTabela.Fields[intCampoTabela].DisplayName;
          intCampo := intCampo + 1;
        end;

        //Matriz Com a Largura das Colunas
        intCampo := 0;
        for intCampoTabela := 0 To (qryTabela.FieldCount - 1) do
          if (qryTabela.Fields[intCampoTabela].Visible) then
            begin
              if (Length(Desc[intCampo + 1]) >
                qryTabela.Fields[intCampoTabela].DisplayWidth) then
                Largura[intCampo + 1] := Length(Desc[intCampo + 1])
              else
                Largura[intCampo + 1] :=
                  qryTabela.Fields[intCampoTabela].DisplayWidth;
              intCampo := intCampo + 1;
            end;

            //Colocamos as Descrições dos Campos para Seleção
            edtTitulo.Clear;
            ltbCampo.Clear;
            ltbCampoSelecionado.Clear;
            for intCampo := 0 to (qryTabela.FieldCount - 1) do
              ltbCampo.Items.Add(Desc[intCampo + 1]);
            end;

```

- No **Object Inspector**, selecione o botão **btnIncluirCampo**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;

- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmRelatorioDinamico.btnIncluirCampoClick(Sender: TObject);
begin
  if (ltbCampo.ItemIndex >= 0) then
  begin
    //Evitando que um campo Indisponível seja selecionado
    if (ltbCampo.Items.Strings[ltbCampo.ItemIndex] = 'Indisponível') then
      exit;
    if (Pagina + Largura[PosCampo(ltbCampo.Items.Strings[ltbCampo.ItemIndex])] >
      125) then
      Application.MessageBox('Campo Supera o Tamanho da Pagina', 'Aviso',
        MB_ICONINFORMATION + MB_OK)
    else
      begin
        Pagina := Pagina +
          Largura[PosCampo(ltbCampo.Items.Strings[ltbCampo.ItemIndex])];
        ltbCampoSelecioneado.Items.Add(
          ltbCampo.Items.Strings[ltbCampo.ItemIndex]);
        ltbCampo.Items.Delete(ltbCampo.ItemIndex);
      end;
  end;
end;

```

- No **Object Inspector**, selecione o botão **btnRemoverCampo**;



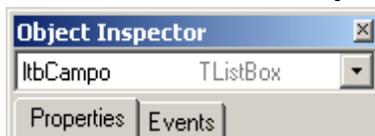
- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmRelatorioDinamico.btnRemoverCampoClick(Sender: TObject);
begin
  if (ltbCampoSelecioneado.Items.Count > 0) then
  begin
    Pagina := Pagina - Largura[PosCampo(
      ltbCampoSelecioneado.Items.Strings[ltbCampoSelecioneado.ItemIndex])];
    ltbCampo.Items.Add(
      ltbCampoSelecioneado.Items.Strings[ltbCampoSelecioneado.ItemIndex]);
    ltbCampoSelecioneado.Items.Delete(ltbCampoSelecioneado.ItemIndex);
  end;
end;

```

- No **Object Inspector**, selecione o listbox **ltbCampo**;



- Agora selecione a aba **Events** e em seguida o evento **OnDbClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento **OnDbClick** e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmRelatorioDinamico.ltbCampoDbClick(Sender: TObject);
begin
  btnIncluirCampo.Click;
end;

```

- No **Object Inspector**, selecione o listbox **ltbCampoSelecioneado**;



- Agora selecione a aba **Events** e em seguida o evento **OnDbClick**;

- Dê um duplo-clique no retângulo em branco a direita do evento OnDbClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmRelatorioDinamico.ltbCampoSelecioneadoDbClick(
  Sender: TObject);
begin
  btnRemoverCampo.Click;
end;
```

- No **Object Inspector**, selecione o speedbutton **btnCampoAcima**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmRelatorioDinamico.btnCampoAcimaClick(Sender: TObject);
var
  intCampo: integer;
begin
  if (ltbCampoSelecioneado.ItemIndex > 0) then begin
    intCampo := ltbCampoSelecioneado.ItemIndex;
    ltbCampoSelecioneado.Items.Move(intCampo, intCampo-1);
    ltbCampoSelecioneado.ItemIndex := intCampo-1;
  end;
end;
```

- No **Object Inspector**, selecione o speedbutton **btnCampoAbaixo**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmRelatorioDinamico.btnCampoAbaixoClick(Sender: TObject);
var
  intCampo: integer;
begin
  if (ltbCampoSelecioneado.ItemIndex <
    (ltbCampoSelecioneado.Items.Count - 1)) then
    begin
      intCampo := ltbCampoSelecioneado.ItemIndex;
      ltbCampoSelecioneado.Items.Move(intCampo, intCampo+1);
      ltbCampoSelecioneado.ItemIndex := intCampo+1;
    end;
end;
```

- No **Object Inspector**, selecione o bitbtn **bbtGerar**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmRelatorioDinamico.bbtGerarClick(Sender: TObject);
var
  intCampo, intColuna, intTamCampo: integer;
  strNomeCampo: String;
```

```

begin
  if (Trim(edtTitulo.Text) = '') then
  begin
    Application.MessageBox('Favor digitar um título para o relatório!',
      'Aviso', MB_ICONINFORMATION + MB_OK);
    edtTitulo.SetFocus;
    exit;
  end;

  if (lbtCampoSelecioneado.Items.Count > 0) then
  begin
    intColuna := 5;

    //Criando o formulário que contém o relatório
    if (frmRelatorio = nil) then
      Application.CreateForm(TfrmRelatorio, frmRelatorio);

    //Configuramos a fonte de dados do relatório
    FrmRelatorio.QuickRep1.DataSet := qryTabela;

    // Informamos o Título do Relatório
    FrmRelatorio.QrLabel2.Caption := edtTitulo.Text;

    for intCampo := 0 to (lbtCampoSelecioneado.Items.Count - 1) do
    begin
      //Obtendo o Tamanho e Nome do Campo
      intTamCampo :=
        Largura[PosCampo(lbtCampoSelecioneado.Items.Strings[intCampo])] + 2;
      strNomeCampo :=
        Campo[PosCampo(lbtCampoSelecioneado.Items.Strings[intCampo])];

      //Criando Componentes Para os Titulos de Colunas
      QrLabel[intCampo+1] :=
        TQrLabel.Create(FrmRelatorio.QuickRep1.Bands.ColumnHeaderBand);
      QrLabel[intCampo+1].Parent :=
        FrmRelatorio.QuickRep1.Bands.ColumnHeaderBand;
      QrLabel[intCampo+1].Left := intColuna;
      QrLabel[intCampo+1].Top := 4;
      QrLabel[intCampo+1].Caption :=
        lbtCampoSelecioneado.Items.Strings[intCampo];
      QrLabel[intCampo+1].Font.Style := [fsunderline,fsbold];

      //Criando Componentes de Exibição Dados das Colunas
      if (qryTabela.FieldName(strNomeCampo).DataType in [ftBlob]) then
      begin
        QrDbImage[intCampo+1] :=
          TQrDbImage.Create(frmRelatorio.QuickRep1.Bands.DetailBand);
        QrDbImage[intCampo+1].Parent :=
          frmRelatorio.QuickRep1.Bands.DetailBand;
        QrDbImage[intCampo+1].Left := intColuna;
        QrDbImage[intCampo+1].Top := 8;
        QrDbImage[intCampo+1].DataSet := qryTabela;
        QrDbImage[intCampo+1].DataField := strNomeCampo;
        QrDbImage[intCampo+1].Stretch := True;
        FrmRelatorio.DetailBand1.Height := QrDbImage[intCampo+1].Height;
        intColuna := intColuna + (5 * intTamCampo);
      end
      else
      begin
        QrDbText[intCampo+1] :=
          TQrDbText.Create(FrmRelatorio.QuickRep1.Bands.DetailBand);
        QrDbText[intCampo+1].Parent :=
          frmRelatorio.QuickRep1.Bands.DetailBand;
        if (qryTabela.FieldName(strNomeCampo).DataType in
          [ftSmallint,ftInteger,ftFloat,ftCurrency,ftLargeint]) then
        begin
          QrDbText[intCampo+1].AutoSize := False;
          QrDbText[intCampo+1].Alignment := taRightJustify;

```

```

        QrDbText[intCampo+1].Width := 4 * intTamCampo;
    end;
    QrDbText[intCampo+1].Left := intColuna;
    QrDbText[intCampo+1].Top := 8;
    QrDbText[intCampo+1].DataSet := qryTabela;
    QrDbText[intCampo+1].DataField := strNomeCampo;
    intColuna := intColuna + (5 * intTamCampo);
end;
end;

//Chamamos o Relatório
frmRelatorio.QuickRep1.Preview;

//Liberamos os Componentes utilizados
for intCampo := 0 to ltbCampoSelecioneado.Items.Count - 1 do begin
    QrLabel[intCampo + 1].free;
    QrDbText[intCampo + 1].free;
end;

//Retirando o formulário do relatório da memória
frmRelatorio.Destroy;
frmRelatorio := nil;
end
else
begin
    Application.MessageBox('Favor selecionar pelo menos um campo!', 'Aviso',
        MB_ICONINFORMATION + MB_OK);

    ltbCampo.SetFocus;
end;
end;
end;

```

- Selecione **Project / Options / Aba Forms / frmRelatorioDinamico (em Auto-create forms) / clique no botão seta para direita** > / OK;
- Salve o trabalho, clicando sobre o botão  (**Save All**).



### 11.2.8. Criando a Janela Relatório (Qreport)

- Crie uma nova janela clicando em **File / New / Form**;
- Altere suas propriedades conforme ilustra a tabela seguinte:

| Propriedade | Valor |
|-------------|-------|
|             |       |

|         |              |
|---------|--------------|
| Caption | Relatórios   |
| Height  | 522          |
| Name    | frmRelatorio |
| Width   | 687          |

- Salve o trabalho clicando **File / Save As** ou clicando sobre o botão  (**Save**) que se encontra na barra de ferramentas. Na caixa de diálogo “Save Unit 1 As”, selecione a pasta (diretório) **c:\piloto**. Na caixa de texto *Nome do Arquivo*, digite **untRelatorio** e clique sobre o botão **Salvar**;
- Selecione o janela **frmRelatorio**;
- Na seção **uses** do código acrescente as palavras em negrito do código abaixo:

```
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms, Dialogs, QuickRpt, ExtCtrls, QRCtrls, DB,
    Printers;
```

O que você acabou de realizar foi a adição das bibliotecas do MDOQuery e do DB.
- Clique na paleta **QReport** e insira na janela *frmRelatorio* o componente **QuickRep** ;
- No **Object Inspector** selecione a propriedade **Left** e altere seu valor para **3**;
- No **Object Inspector** selecione a propriedade **Zoom** e altere seu valor para **80**;
- Clique com o botão direito do mouse sobre o componente *QuickRep1* no formulário e no menu apresentado clique na opção **Report Settings**;



- Na caixa de diálogo **Report Settings**, modifique os seguintes abaixo:
  - ✓ Paper Size: **A4 210 x 297 mm**
  - ✓ Bands – Marque **Page header** e defina seu tamanho em **10,91**
  - ✓ Bands – Marque **Column header** e defina seu tamanho em **6,61**
  - ✓ Bands – Marque **Detail Band** e defina seu tamanho em **6,61**

**Report Settings**

Paper size  
 A4 210 x 297 mm Width 210,0 Length 297,0 Portrait

Margins  
 Top 10,00 Left 10,00 Column space 0,00  
 Bottom 10,00 Right 10,00 Number of columns 1

Other  
 Font Arial Size 10 Units MM

Page frame  
 Top  Left Color ■ Change Frame width 1  
 Bottom  Right

Bands

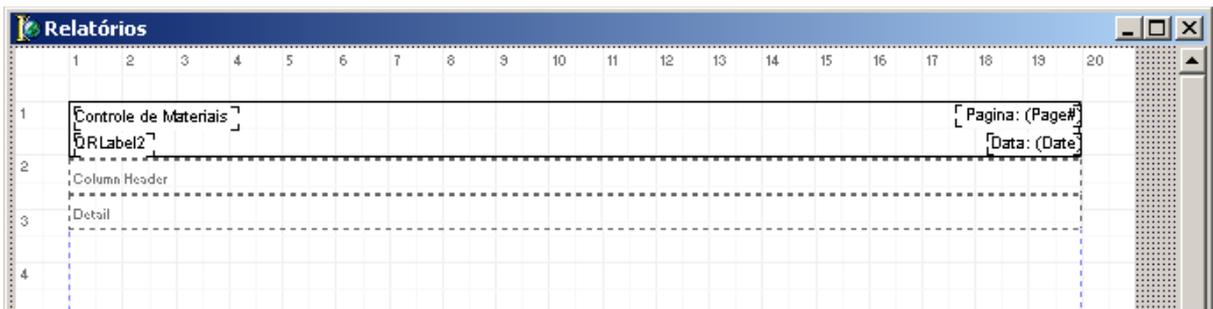
| Band                                              | Length |
|---------------------------------------------------|--------|
| <input checked="" type="checkbox"/> Page header   | 10,91  |
| <input type="checkbox"/> Title                    |        |
| <input checked="" type="checkbox"/> Column header | 6,61   |
| <input checked="" type="checkbox"/> Detail band   | 6,61   |

Page footer Length  
 Summary Length  
 Print first page header  
 Print last page footer

About QuickReport Preview Apply OK Cancel

- Selecione no formulário **frmRelatorio** o componente QuickRep1;
- Clique na paleta **QReport** e insira na banda *PageHeaderBand* do componente *QuickRep1* o componente **QLabel** **A**;
- No **Object Inspector** selecione a propriedade **Caption** e altere seu valor para **Controle de Materiais**;
- No **Object Inspector** selecione a propriedade **Left** e altere seu valor para **3**;
- No **Object Inspector** selecione a propriedade **Top** e altere seu valor para **3**;
- Clique na paleta **QReport** e insira na banda *PageHeaderBand* do componente *QuickRep1* o componente **QLabel** **A**;
- No **Object Inspector** selecione a propriedade **Left** e altere seu valor para **3**;
- No **Object Inspector** selecione a propriedade **Top** e altere seu valor para **18**;
- Clique na paleta **QReport** e insira na banda *PageHeaderBand* do componente *QuickRep1* o componente **QRSysData** **sys**;
- No **Object Inspector** selecione a propriedade **Alignment** e altere seu valor para **taRightJustify**;
- No **Object Inspector** selecione a propriedade **Data** e altere seu valor para **qrsPageNumber**;
- No **Object Inspector** selecione a propriedade **Left** e altere seu valor para **503**;
- No **Object Inspector** selecione a propriedade **Text** e altere seu valor para **Pagina:**;
- No **Object Inspector** selecione a propriedade **Top** e altere seu valor para **3**;
- Clique na paleta **QReport** e insira na banda *PageHeaderBand* do componente *QuickRep1* o componente **QRSysData** **sys**;
- No **Object Inspector** selecione a propriedade **Alignment** e altere seu valor para **taRightJustify**;

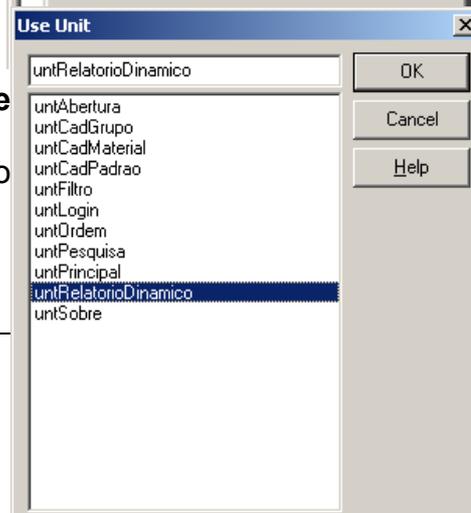
- No **Object Inspector** selecione a propriedade **Data** e altere seu valor para **qrsDate**;
- No **Object Inspector** selecione a propriedade **Left** e altere seu valor para **521**;
- No **Object Inspector** selecione a propriedade **Text** e altere seu valor para **Data**;
- No **Object Inspector** selecione a propriedade **Top** e altere seu valor para **18**;



- Agora vamos ligar este relatório com janela Relatório Dinâmico, da seguinte forma:
  - ✓ File / Use Unit;



diálogo Use  
clique no botão



✓ Na caixa de  
seleção  
Unit,  
*untRelatorioDinamico* e  
OK;

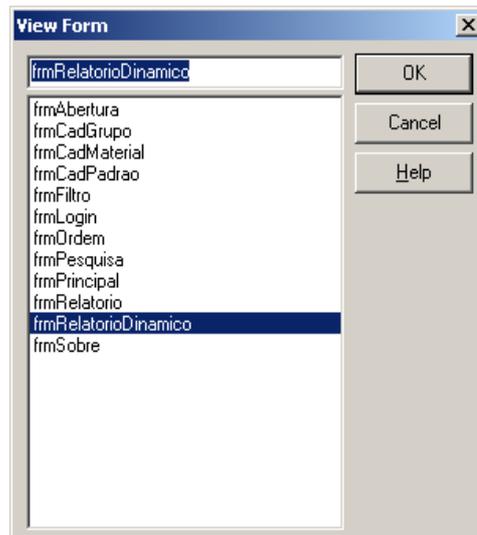
- No **Object Inspector**, selecione o formulário **QuickRep1**;



- No **Object Inspector** selecione a aba **Events** e depois o evento **BeforePrint**;
- Agora dê um duplo-clique no retângulo branco a direita do evento BeforePrint e digite as linhas em negrito do código abaixo:

```
procedure TfrmRelatorio.QuickRep1BeforePrint(Sender: TCustomQuickRep;  
    var PrintReport: Boolean);  
begin  
    if (frmRelatorioDinamico.Pagina <= 94) then  
        QuickRep1.Page.Orientation := poPortrait  
    else  
        QuickRep1.Page.Orientation := poLandscape;  
end;
```

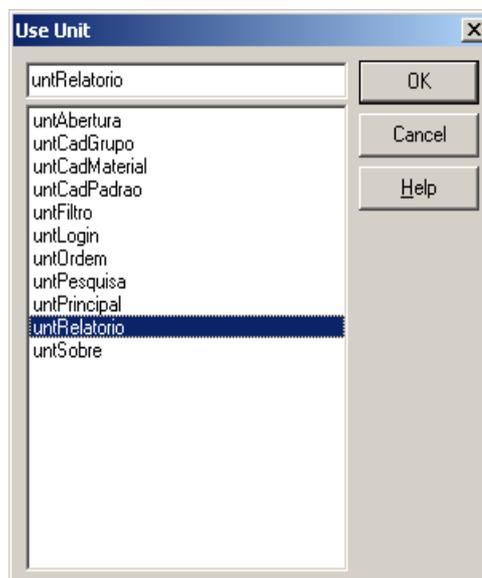
- Agora pressione **Shift + F12**, selecione o formulário **frmRelatorioDinamico** e dê um clique no botão **OK**;



- Agora vamos ligar a janela Relatório Dinâmico com Relatório, da seguinte forma:
  - ✓ File / Use Unit;



- ✓ Na caixa de diálogo **Use Unit**, selecione **untRelatorio** e clique no botão **OK**;



- Selecione **Project / Options / Aba Forms / frmRelatorio** (em **Auto-create forms**) / clique no botão seta para direita > / **OK**;
- Salve o trabalho, clicando sobre o botão  (**Save All**).

### 11.2.9. Criando a Janela de Cadastro Padrão

- Crie uma nova janela clicando em **File / New / Form**;
- Altere suas propriedades conforme ilustra a tabela seguinte:

| Propriedade   | Valor           |
|---------------|-----------------|
| + BorderIcons |                 |
| biSystemMenu  | True            |
| biMinimize    | False           |
| biMaximize    | False           |
| biHelp        | False           |
| BorderStyle   | bsSingle        |
| Caption       | Cadastro Padrão |
| Height        | 370             |
| KeyPreview    | True            |
| Name          | frmCadPadrao    |
| Position      | poScreenCenter  |
| Width         | 537             |

- Salve o trabalho clicando **File / Save As** ou clicando sobre o botão  (**Save**) que se encontra na barra de ferramentas. Na caixa de diálogo "Save Unit 1 As", selecione a pasta (diretório) **d:\piloto**. Na caixa de texto *Nome do Arquivo*, digite **untCadPadrao** e clique sobre o botão **Salvar**;
- Selecione o janela **frmCadPadrao**;
- Na seção **uses** do código acrescente as palavras em negrito do código abaixo:  
uses  
    Windows, Messages, SysUtils, Variants, Classes, Graphics,  
    Controls, Forms, Dialogs, Menus, ExtCtrls, DB, DBCtrls,  
    Buttons, ComCtrls, ImgList, Grids, DBGrids, **MDOQuery**;
- O que você acabou de realizar foi a adição da biblioteca do MDOQuery.
- Selecione o janela **frmCadPadrao**;
- Clique na paleta **Standard** e insira na janela *frmCadPadrao* o componente **MainMenu** ;
- No **Object Inspector**, selecione e propriedade **Name** e altere seu valor para **mmuCadastro**;
- Dê um duplo-clique sobre o componente **MainMenu** que foi inserido no **frmCadPadrao**. Uma janela denominada "frmCadPadrao.mmuCadastro" será aberta em sua área de trabalho;



- Observe uma área escurecida, contornada por um traço pontilhado, localizada na parte superior dessa janela. É aí que se inicia a inserção dos itens que comporão o menu. Assim sendo, digite na propriedade **Caption**, no *Object Inspector*, a palavra **&Arquivo** e aperte a tecla **[Enter]** e na propriedade **Name** digite **mmuArquivo** e aperte a tecla **[Enter]**;
- Ao executar o item anterior, a palavra *Arquivo* será inserida na barra de menus e um novo retângulo pontilhado será posicionado à sua direita. Clique sobre o referido retângulo para selecioná-lo, e na propriedade **Caption** do *Object Inspector*, digite a palavra **&Registro** e aperte a tecla **[Enter]**. Altere a propriedade **Name** para **mmuRegistro** e aperte a tecla **[Enter]**;
- Ao final deste processo, o construtor de menus deve ter a seguinte aparência:

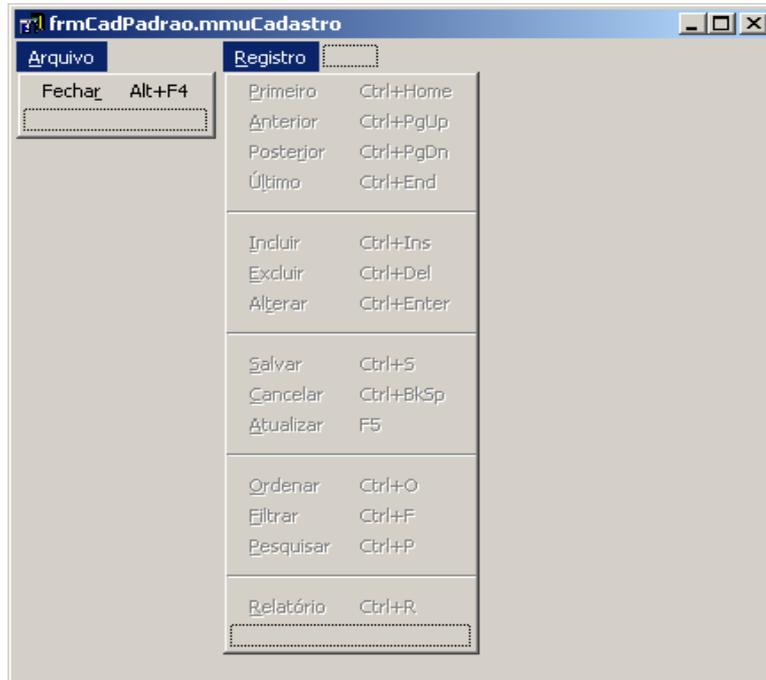


- Na janela do *frmCadPadrao.mmuCadastro* clique sobre o menu *Arquivo*. Observe que um retângulo contornado com uma linha pontilhada posicionado logo abaixo da palavra *Arquivo*. Selecione esse retângulo, altere a propriedade **Caption** para **Sai&r**, altere a propriedade **Hint** para **Sair|Encerra a execução do Cadastro**, altere a propriedade **Name** para **mmuArqSair**, altere a propriedade **Shortcut** para **Alt+F4** (digitar);
- Na janela do *frmCadPadrao.mmuCadastro* clique sobre o menu *Registro*. Selecione o retângulo contornado com uma linha pontilhada logo abaixo da palavra *Registro*, altere a propriedade **Caption** para **&Primeiro**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Busca os dados do primeiro registro**, altere a propriedade **Name** para **mmuRegPrimeiro** e altere a propriedade **Shortcut** para **Ctrl+Home**;
- Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Primeiro*), altere a propriedade **Caption** para **&Anterior**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Busca os dados do registro anterior**, altere a propriedade **Name** para **mmuRegAnterior** e altere a propriedade **Shortcut** para **Ctrl+PgUp**;
- Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Anterior*), altere a propriedade **Caption** para **Poste&rior**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Busca os dados do registro posterior**, altere a propriedade **Name** para **mmuRegPosterior** e altere a propriedade **Shortcut** para **Ctrl+PgDn**;
- Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Posterior*), altere a propriedade **Caption** para **Ú&ltimo**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Busca os dados do último registro**, altere a propriedade **Name** para **mmuRegUltimo** e altere a propriedade **Shortcut** para **Ctrl+End**;
- Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Último*), altere a propriedade **Caption** para **-**, altere a propriedade **Name** para **mmuRegSeparador1**;
- Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item **-**), altere a propriedade **Caption** para **&Incluir**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Permite incluir um novo**

- registro**, altere a propriedade **Name** para **mmuRegIncluir** e altere a propriedade **ShortCut** para **Ctrl+Ins**;
- Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Incluir*), altere a propriedade **Caption** para **&Excluir**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Permite excluir um registro**, altere a propriedade **Name** para **mmuRegExcluir** e altere a propriedade **ShortCut** para **Ctrl+Del**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Excluir*), altere a propriedade **Caption** para **Al&terar**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Permite modificar os dados do registro**, altere a propriedade **Name** para **mmuRegAlterar** e altere a propriedade **ShortCut** para **Ctrl+Enter**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Último*), altere a propriedade **Caption** para **-**, altere a propriedade **Name** para **mmuRegSeparador2**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item **-**), altere a propriedade **Caption** para **&Salvar**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Confirma as modificações em um registro**, altere a propriedade **Name** para **mmuRegSalvar** e altere a propriedade **ShortCut** para **Ctrl+S**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Salvar*), altere a propriedade **Caption** para **&Cancelar**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Cancela as modificações em um registro**, altere a propriedade **Name** para **mmuRegCancelar** e altere a propriedade **ShortCut** para **Ctrl+BkSp**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Cancelar*), altere a propriedade **Caption** para **&Atualizar**, altere a propriedade **Enabled** para **False**, altere a propriedade **Hint** para **Atualiza os dados do cache**, altere a propriedade **Name** para **mmuRegAtualizar** e altere a propriedade **ShortCut** para **F5**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Atualizar*), altere a propriedade **Caption** para **-**, altere a propriedade **Name** para **mmuRegSeparador3**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item **-**), altere a propriedade **Caption** para **&Ordenar**, altere a propriedade **Enabled** para **False**, altere a propriedade **Name** para **mmuRegOrdenar** e altere a propriedade **ShortCut** para **Ctrl+O**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Ordenar*), altere a propriedade **Caption** para **&Filtrar**, altere a propriedade **Enabled** para **False**, altere a propriedade **Name** para **mmuRegFiltrar** e altere a propriedade **ShortCut** para **Ctrl+F**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Filtrar*), altere a propriedade **Caption** para **&Pesquisar**, altere a propriedade **Enabled** para **False**, altere a propriedade **Name** para **mmuRegPesquisar** e altere a propriedade **ShortCut** para **Ctrl+P**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item *Pesquisar*), altere a propriedade **Caption** para **-**, altere a propriedade **Name** para **mmuRegSeparador4**;
  - Selecione o próximo retângulo contornado com uma linha pontilhada (abaixo do item **-**), altere a propriedade **Caption** para **&Relatório**, altere a propriedade

**Enabled** para **False**, altere a propriedade **Hint** para **Gera um relatório com todos os registros**, altere a propriedade **Name** para **mmuRegRelatorio** e altere a propriedade **ShortCut** para **Ctrl+R**;

- Ao final destes processos, o construtor de menus deve ter a seguinte aparência:



- Feche a janela *frmCadPadrao.mmuCadastro*, clique sobre o botão (**Save All**);
- Clique na paleta **Standard** e insira na janela *frmCadPadrao* o componente **Panel** ;
- No **Object Inspector**, altere as seguintes propriedades:

| <b>Propriedade</b> | <b>Valor</b> |
|--------------------|--------------|
| Align              | alTop        |
| BevelInner         | bvNone       |
| BevelOuter         | bvNone       |
| Caption            | (vazio)      |
| Height             | 39           |
| Name               | pnlBotoes    |

- Clique na paleta **Data Access** e insira na janela *frmCadPadrao* o componente **DataSource** ;
- No **Object Inspector**, selecione a propriedade **Name** e altere seu valor para **dsRegistro**;
- Clique na paleta **Data Controls** e insira no componente *pnlBotoes* o componente **DBNavigator** ;
- No **Object Inspector**, altere as seguintes propriedades:

| <b>Propriedade</b> | <b>Valor</b> |
|--------------------|--------------|
| ConfirmDelete      | False        |
| DataSource         | dsRegistro   |
| Height             | 35           |

| <b>Propriedade</b> | <b>Valor</b>                                                                                                                                                                                                                |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hints              | Primeiro [Ctrl+Home]<br>Anterior [Ctrl+PgUp]<br>Posterior [Ctrl+PgDn]<br>Último [Ctrl+End]<br>Incluir [Ctrl+Ins]<br>Excluir [Ctrl+Del]<br>Alterar [Ctrl+Enter]<br>Salvar [Ctrl+S]<br>Cancelar [Ctrl+BkSp]<br>Atualizar [F5] |
| Left               | 0                                                                                                                                                                                                                           |
| Name               | dbnRegistro                                                                                                                                                                                                                 |
| ShowHint           | True                                                                                                                                                                                                                        |
| Top                | 0                                                                                                                                                                                                                           |
| Width              | 350                                                                                                                                                                                                                         |

- Clique na paleta **Additional** e insira no componente *pnlBotoes* o componente **SpeedButton** .

- No **Object Inspector**, altere as seguintes propriedades:

| <b>Propriedade</b> | <b>Valor</b>                                                                      |
|--------------------|-----------------------------------------------------------------------------------|
| Caption            | (vazio)                                                                           |
| Enabled            | False                                                                             |
| Glyph              | C:\Arquivos de programas\Arquivos comuns\Borland Shared\Images\Buttons\REPORT.BMP |
| Height             | 35                                                                                |
| Hint               | Relatório [Ctrl+R]                                                                |
| Left               | 455                                                                               |
| Name               | sbtRelatorio                                                                      |
| ShowHint           | True                                                                              |
| Top                | 0                                                                                 |
| Width              | 35                                                                                |

- Clique na paleta **Additional** e insira no componente *pnlBotoes* o componente **SpeedButton** .

- No **Object Inspector**, altere as seguintes propriedades:

| <b>Propriedade</b> | <b>Valor</b>                                                                        |
|--------------------|-------------------------------------------------------------------------------------|
| Caption            | (vazio)                                                                             |
| Enabled            | True                                                                                |
| Glyph              | C:\Arquivos de programas\Arquivos comuns\Borland Shared\Images\Buttons\DOOROPEN.BMP |
| Height             | 35                                                                                  |

| <b>Propriedade</b> | <b>Valor</b>  |
|--------------------|---------------|
| Hint               | Sair [Alt+F4] |
| Left               | 490           |
| Name               | sbtSair       |
| ShowHint           | True          |
| Top                | 0             |
| Width              | 35            |

- Clique na paleta **Standard** e insira na janela *frmCadPadrao* o componente **Panel** ;
- No **Object Inspector**, altere as seguintes propriedades:

| <b>Propriedade</b> | <b>Valor</b>           |
|--------------------|------------------------|
| Align              | alTop                  |
| BevelInner         | bvNone                 |
| BevelOuter         | bvNone                 |
| Caption            | (vazio)                |
| Height             | 67                     |
| Name               | pnlOrdemPesquisaFiltro |

- Clique na paleta **Standard** e insira no painel *pnlOrdemPesquisaFiltro* o componente **RadioGroup** ;
- No **Object Inspector**, altere as seguintes propriedades:

| <b>Propriedade</b> | <b>Valor</b>        |
|--------------------|---------------------|
| Caption            | Ordem:              |
| Columns            | 1                   |
| Height             | 64                  |
| Items              | Código<br>Descrição |
| ItemIndex          | 0                   |
| Left               | 0                   |
| Name               | rgOrdem             |
| Top                | 0                   |
| Width              | 106                 |

- Clique na paleta **Standard** e insira no painel *pnlOrdemPesquisaFiltro* o componente **GroupBox** ;
- No **Object Inspector**, altere as seguintes propriedades:

| <b>Propriedade</b> | <b>Valor</b> |
|--------------------|--------------|
| Caption            | Pesquisa:    |
| Height             | 64           |
| Left               | 108          |

| <i>Propriedade</i> | <i>Valor</i> |
|--------------------|--------------|
| Name               | grpPesquisa  |
| Top                | 0            |
| Width              | 153          |

- Clique na paleta **Standard** e insira no groupbox **grpPesquisa** o componente **RadioButton** ;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i> |
|--------------------|--------------|
| Caption            | Código       |
| Checked            | True         |
| Left               | 6            |
| Name               | rbCodigoPesq |
| Top                | 18           |
| Width              | 58           |

- Clique na paleta **Standard** e insira no groupbox **grpPesquisa** o componente **RadioButton** ;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i>    |
|--------------------|-----------------|
| Caption            | Descrição       |
| Left               | 72              |
| Name               | rbDescricaoPesq |
| Top                | 18              |
| Width              | 73              |

- Clique na paleta **Standard** e insira no groupbox **grpPesquisa** o componente **Edit** ;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i> |
|--------------------|--------------|
| Left               | 6            |
| Name               | edtPesquisa  |
| Text               | (vazio)      |
| Top                | 36           |
| Width              | 140          |

- Clique na paleta **Standard** e insira no panel **pnIOrdemPesquisaFiltro** o componente **GroupBox** ;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i> |
|--------------------|--------------|
| Caption            | Filtro:      |
| Height             | 64           |
| Left               | 264          |

| <i>Propriedade</i> | <i>Valor</i> |
|--------------------|--------------|
| Name               | grpFiltro    |
| Top                | 0            |
| Width              | 265          |

- Clique na paleta **Standard** e insira no groupbox **grpFiltro** o componente **RadioButton** ;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i>   |
|--------------------|----------------|
| Caption            | Código         |
| Left               | 6              |
| Name               | rbCodigoFiltro |
| Top                | 18             |
| Width              | 58             |

- Clique na paleta **Standard** e insira no groupbox **grpFiltro** o componente **RadioButton** ;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i>      |
|--------------------|-------------------|
| Caption            | Descrição         |
| Left               | 72                |
| Name               | rbDescricaoFiltro |
| Top                | 18                |
| Width              | 73                |

- Clique na paleta **Standard** e insira no groupbox **grpFiltro** o componente **Label** **A**;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i> |
|--------------------|--------------|
| Caption            | De:          |
| Left               | 6            |
| Name               | lblDe        |
| Top                | 39           |
| Width              | 17           |

- Clique na paleta **Standard** e insira no groupbox **grpFiltro** o componente **Edit** ;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i> |
|--------------------|--------------|
| Enabled            | False        |
| Left               | 25           |
| Name               | edtDe        |
| Text               | (vazio)      |
| Top                | 36           |
| Width              | 102          |

- Clique na paleta **Standard** e insira no groupbox **grpFiltro** o componente **Label A**;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i> |
|--------------------|--------------|
| Caption            | Até:         |
| Left               | 135          |
| Name               | lblAte       |
| Top                | 39           |
| Width              | 19           |

- Clique na paleta **Standard** e insira no groupbox **grpFiltro** o componente **Edit lab1**;

- No **Object Inspector**, altere as seguintes propriedades:

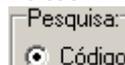
| <i>Propriedade</i> | <i>Valor</i> |
|--------------------|--------------|
| Enabled            | False        |
| Left               | 156          |
| Name               | edtAte       |
| Text               | (vazio)      |
| Top                | 36           |
| Width              | 102          |

- Clique na paleta **Additional** e insira no groupbox **grpFiltro** o componente **SpeedButton** ;

- No **Object Inspector**, altere as seguintes propriedades:

| <i>Propriedade</i> | <i>Valor</i>                                                                         |
|--------------------|--------------------------------------------------------------------------------------|
| Caption            | Habilitar                                                                            |
| Enabled            | False                                                                                |
| Glyph              | C:\Arquivos de programas\Arquivos comuns\Borland Shared\Images\Buttons\ HOURGLAS.BMP |
| Height             | 22                                                                                   |
| Left               | 164                                                                                  |
| Name               | btnFiltro                                                                            |
| Top                | 12                                                                                   |
| Width              | 94                                                                                   |

- Dê um duplo clique sobre o **RadioButton Código** do **GroupBox Pesquisa**;



- Agora no Editor de Código digite as linhas em negrito a seguir:

```
procedure TfrmCadPadrao.rbCodigoPesqClick(Sender: TObject);
begin
    edtPesquisa.Clear;
    edtPesquisa.SetFocus;
end;
```

- Dê um duplo clique sobre o **RadioButton Descrição** do **GroupBox Pesquisa**;



- Agora no Editor de Código digite as linhas em negrito a seguir:

```
procedure TfrmCadPadrao.rbDescricaoPesqClick(Sender: TObject);
begin
    edtPesquisa.Clear;
    edtPesquisa.SetFocus;
end;
```

- Selecione o componente **Edit** do **GroupBox Pesquisa**;

- Agora selecione a aba **Events** e em seguida o evento **OnExit**;

- Dê um duplo-clique no retângulo em branco a direita do evento OnExit e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.edtPesquisaExit(Sender: TObject);
begin
    edtPesquisa.Clear;
end;
```

- Agora selecione o evento **OnKeyPress**;

- Dê um duplo-clique no retângulo em branco a direita do evento OnKeyPress e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.edtPesquisaKeyPress(Sender: TObject;
    var Key: Char);
begin
    if ((rbCodigoPesq.Checked) and (not(Key in ['0'..'9', #8]))) then
        Key := #0;
end;
```

- Pressione a tecla **F12** para retornar ao formulário;

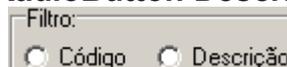
- Dê um duplo clique sobre o **RadioButton Código** do **GroupBox Filtro**;



- Agora no Editor de Código digite as linhas em negrito a seguir:

```
procedure TfrmCadPadrao.rbCodigoFiltroClick(Sender: TObject);
begin
    edtDe.Enabled := True;
    edtDe.Clear;
    edtAte.Clear;
    edtDe.SetFocus;
end;
```

- Dê um duplo clique sobre o **RadioButton Descrição** do **GroupBox Filtro**;



- Agora no Editor de Código digite as linhas em negrito a seguir:

```
procedure TfrmCadPadrao.rbDescricaoFiltroClick(Sender: TObject);
begin
    edtDe.Enabled := True;
    edtDe.Clear;
    edtAte.Clear;
    edtDe.SetFocus;
end;
```

- Selecione o componente **Edit edtDe** do **GroupBox Filtro**;
- Agora selecione a aba **Events** e em seguida o evento **OnKeyPress**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnKeyPress e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.edtDeKeyPress(Sender: TObject; var Key: Char);
begin
    if ((rbCodigoFiltro.Checked) and (not(Key in ['0'..'9', #8]))) then
        Key := #0;
end;
```

- Agora selecione o evento **OnChange**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnChange e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.edtDeChange(Sender: TObject);
begin
    if (Trim(edtDe.Text) = '') then
        begin
            edtDe.Text := '';
            btnFiltro.Enabled := False;
            edtAte.Enabled := False;
        end
    else
        begin
            btnFiltro.Enabled := True;
            edtAte.Enabled := True;
        end;
end;
```

- Selecione o componente **Edit edtAte** do **GroupBox Filtro**;
- Agora selecione a aba **Events** e em seguida o evento **OnKeyPress**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnKeyPress e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.edtAteKeyPress(Sender: TObject; var Key: Char);
begin
    if ((rbCodigoFiltro.Checked) and (not(Key in ['0'..'9', #8]))) then
        Key := #0;
end;
```

- Agora selecione o evento **OnChange**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnChange e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.edtAteChange(Sender: TObject);
begin
    if (Trim(edtAte.Text) = '') then
        edtAte.Text := '';
end;
```

- Dê um duplo clique sobre o **SpeedButton Habilitar** do **GroupBox Filtro**;



- Agora no Editor de Código digite as linhas em negrito a seguir:

```

procedure TfrmCadPadrao.btnFiltroClick(Sender: TObject);
begin
    //Se for preenchido somente o inicial, o final recebe o inicial
    if ((Trim(edtDe.Text) <> '') AND (Trim(edtAte.Text) = '')) Then
        edtAte.Text := edtDe.Text;

    if (btnFiltro.Caption = 'Habilitar') then
    begin
        btnFiltro.Caption := 'Desabilitar';
        rbCodigoFiltro.Enabled := False;
        rbDescricaoFiltro.Enabled := False;
        edtDe.Enabled := False;
        edtAte.Enabled := False;
    end
    else
    begin
        btnFiltro.Caption := 'Habilitar';
        rbCodigoFiltro.Enabled := True;
        rbCodigoFiltro.Checked := False;
        rbDescricaoFiltro.Enabled := True;
        rbDescricaoFiltro.Checked := False;
        edtDe.Clear;
        edtDe.Enabled := False;
        edtAte.Clear;
        edtAte.Enabled := False;
    end;
end;

```

- Insira um componente **ColEnter** , e um componente **ColFocus** , localizados na paleta **Col**;
- Selecione o componente **ColEnter** e no **Object Inspector** altere o valor da propriedade **Ativo** para **True**;
- Selecione o componente **ColFocus** e no **Object Inspector** altere o valor da propriedade **AtivaColor** para **True**;
- Clique na paleta **Win32** da barra de ferramentas do Delphi;
- Clique sobre o componente **ImageList**  para selecioná-lo, e em seguida clique em qualquer área interna da janela *frmCadPadrao* para inseri-lo nessa janela;
- Altere o valor da propriedade **Name** para **imgIAbas**;
- Dê um duplo-clique sobre o componente *ImageList* que foi inserido na janela Cadastro Padrão e insira as imagens na mesma ordem descrita na tabela abaixo. As imagens estão na pasta (diretório) *C:\Arquivos de programas\Arquivos comuns\Borland Shared\Images\Buttons*.

| Imagem                                                                              | Nome      | Índice |
|-------------------------------------------------------------------------------------|-----------|--------|
|  | FORM.BMP  | 0      |
|  | TABLE.BMP | 1      |

- Clique na paleta **Win32** da barra de ferramentas do Delphi;
- Clique sobre o componente **PageControl**  para selecioná-lo, e em seguida clique em qualquer área interna da janela *frmCadPadrao* para inseri-lo nessa janela;

- Altere suas propriedades conforme ilustra a tabela seguinte:

| Propriedade | Valor    |
|-------------|----------|
| Align       | alClient |
| Images      | imglAbas |
| Name        | pgcAbas  |

- Clique com o botão direito do mouse sobre o componente *PageControl* na janela *frmCadPadrao*;
- No menu suspenso apresentado. Clique sobre a opção **New Page**, para criar uma página dentro do componente *PageControl*;
- Repita o passo anterior por mais duas vezes;
- No componente *PageControl*, clique sobre a aba chamada **TabSheet1** e por fim clique no interior desta aba para selecioná-la;
- No *Object Inspector* altere as seguintes propriedades desta aba:

| Propriedade | Valor         |
|-------------|---------------|
| Caption     | Formulário    |
| ImageIndex  | 0             |
| Name        | tbsFormulario |

- No componente *PageControl*, clique sobre a aba chamada **TabSheet2** e por fim clique no interior desta aba para selecioná-la;
- No *Object Inspector* altere as seguintes propriedades desta aba:

| Propriedade | Valor     |
|-------------|-----------|
| Caption     | Tabela    |
| ImageIndex  | 1         |
| Name        | tbsTabela |

- No componente *PageControl*, clique sobre a aba chamada **Tabela** e por fim clique no interior desta aba para selecioná-la;
- Clique na paleta **Data Controls** na paleta de componentes do Delphi;
- Clique sobre o componente **DBGrid**  para selecioná-lo, e em seguida clique na área interna da aba chamada *Tabela* para inseri-lo;
- Altere suas propriedades conforme ilustra a tabela seguinte:

| Propriedade | Valor      |
|-------------|------------|
| Align       | alClient   |
| DataSource  | dsRegistro |
| Name        | dbgDados   |

- Salve o trabalho clicando sobre o botão  (**Save All**);
- Agora construiremos o código que personalizará o componente *DBNavigator*, para tanto iremos criar uma classe que nos facilitará o trabalho de personalização. Digite a linha em negrito do código a seguir:

```

private
  { Private declarations }
public
  { Public declarations }
end;

BarraNavReg = class(TDBNavigator);

var
  frmCadPadrao: TfrmCadPadrao;
```

- No **Object Inspector**, selecione o formulário *frmCadPadrao*;



- Agora selecione a aba **Events** e em seguida o evento **OnCreate**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnCreate e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.FormCreate(Sender: TObject);
begin
    BarraNavReg(dbnRegistro).buttons[nbFirst].Hint := 'Primeiro [Ctrl+Home]';
    BarraNavReg(dbnRegistro).buttons[nbPrior].Hint := 'Anterior [Ctrl+PgUp]';
    BarraNavReg(dbnRegistro).buttons[nbNext].Hint := 'Posterior [Ctrl+PgDn]';
    BarraNavReg(dbnRegistro).buttons[nbLast].Hint := 'Último [Ctrl+End]';
    BarraNavReg(dbnRegistro).buttons[nbInsert].Hint := 'Incluir [Ctrl+Ins]';
    BarraNavReg(dbnRegistro).buttons[nbDelete].Hint := 'Excluir [Ctrl+Del]';
    BarraNavReg(dbnRegistro).buttons[nbEdit].Hint := 'Alterar [Ctrl+Enter]';
    BarraNavReg(dbnRegistro).buttons[nbPost].Hint := 'Salvar [Ctrl+S]';
    BarraNavReg(dbnRegistro).buttons[nbCancel].Hint := 'Cancelar [Ctrl+BkSp]';
    BarraNavReg(dbnRegistro).buttons[nbRefresh].Hint := 'Atualizar [F5]';
end;
```

- No passo anterior encerramos a personalização do DBNavigator;
- No **Object Inspector**, selecione o formulário *frmCadPadrao*;



- Agora selecione a aba **Events** e em seguida o evento **OnClose**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClose e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.FormClose(Sender: TObject;
    var Action: TCloseAction);
begin
    frmRelatorioDinamico.Destroy;
    frmRelatorioDinamico := nil;
end;
```

- No **Object Inspector**, selecione o formulário *frmCadPadrao*;



- Agora selecione a aba **Events** e em seguida o evento **OnCloseQuery**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnCloseQuery e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.FormCloseQuery(Sender: TObject;
    var CanClose: Boolean);
var
    intResposta: integer;
begin
    if (TMDOQuery(dsRegistro.DataSet).State in [dsEdit, dsInsert]) then
        begin
            intResposta :=
                Application.MessageBox('Deseja salvar as alterações?',
                    'Confirmação', mb_YesNo + mb_DefButton2 + mb_IconQuestion);
            if (intResposta = IDYES) then
                TMDOQuery(dsRegistro.DataSet).Post
            else
                TMDOQuery(dsRegistro.DataSet).Cancel;
        end;
end;
```

- No **Object Inspector**, selecione o formulário **frmCadPadrao**;



- Agora selecione a aba **Events** e em seguida o evento **OnShow**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnShow e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.FormShow(Sender: TObject);
begin
    if (frmRelatorioDinamico = nil) then
        Application.CreateForm(TfrmRelatorioDinamico, frmRelatorioDinamico);
    frmRelatorioDinamico.qryTabela := TMDOQuery(dsRegistro.DataSet);
    frmRelatorioDinamico.strBarraTitulo := Self.Caption;
    dsRegistro.DataSet.Last;
    dsRegistro.DataSet.First;
end;
```

- No **Object Inspector**, selecione o datasource **dsRegistro**;



- Agora selecione a aba **Events** e em seguida o evento **OnChange**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnChange e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.dsRegistroDataChange(Sender: TObject;
    Field: TField);
var
    bolPodeModificar, bolEdicao: boolean;
    bolAnterior, bolPosterior: boolean;
    bolVazio: boolean;
    intComponente: integer;
begin
    bolPodeModificar := dsRegistro.DataSet.Active and
        dsRegistro.DataSet.CanModify;
    bolEdicao := dsRegistro.DataSet.State in [dsEdit, dsInsert];
    bolAnterior := dsRegistro.DataSet.Active and
        (not dsRegistro.DataSet.BOF);
    bolPosterior := dsRegistro.DataSet.Active and
        (not dsRegistro.DataSet.EOF);
    bolVazio := dsRegistro.DataSet.Bof and dsRegistro.DataSet.Eof;

    mmuRegPrimeiro.Enabled := bolAnterior;
    mmuRegAnterior.Enabled := bolAnterior;
    mmuRegPosterior.Enabled := bolPosterior;
    mmuRegUltimo.Enabled := bolPosterior;
    mmuRegIncluir.Enabled := bolPodeModificar;
    mmuRegExcluir.Enabled := bolPodeModificar and (not bolVazio);
    mmuRegAlterar.Enabled := bolPodeModificar and (not bolEdicao);
    mmuRegSalvar.Enabled := bolPodeModificar and bolEdicao;
    mmuRegCancelar.Enabled := bolPodeModificar and bolEdicao;
    mmuRegAtualizar.Enabled := bolPodeModificar;
    mmuRegRelatorio.Enabled := not bolVazio;
    sbtRelatorio.Enabled := not bolVazio;
```

```
//Habilita ou desabilita a alteração dos dados nos campos
for intComponente := 0 to ComponentCount - 1 do
begin
  //Declarar na seção uses: DBCtrls
  if (Components[intComponente] is TDBEdit) then
    (Components[intComponente] as TDBEdit).ReadOnly := not(bolEdicao);
end;

if (dsRegistro.DataSet.State in [dsInsert, dsEdit]) then
  pgcAbas.ActivePageIndex := 0;
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegPrimeiro**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegPrimeiroClick(Sender: TObject);
begin
  dbnRegistro.BtnClick(nbFirst);
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegAnterior**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegAnteriorClick(Sender: TObject);
begin
  dbnRegistro.BtnClick(nbPrior);
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegPosterior**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegPosteriorClick(Sender: TObject);
begin
  dbnRegistro.BtnClick(nbNext);
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegUltimo**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;

- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegUltimoClick(Sender: TObject);
begin
    dbnRegistro.BtnClick(nbLast);
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegIncluir**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegIncluirClick(Sender: TObject);
begin
    dbnRegistro.BtnClick(nbInsert);
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegExcluir**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegExcluirClick(Sender: TObject);
begin
    dbnRegistro.BtnClick(nbDelete);
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegAlterar**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegAlterarClick(Sender: TObject);
begin
    dbnRegistro.BtnClick(nbEdit);
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegSalvar**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegSalvarClick(Sender: TObject);
begin
    dbnRegistro.BtnClick(nbPost);
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegCancelar**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegCancelarClick(Sender: TObject);  
begin  
    dbnRegistro.BtnClick(nbCancel) ;  
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegAtualizar**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegAtualizarClick(Sender: TObject);  
begin  
    dbnRegistro.BtnClick(nbRefresh) ;  
end;
```

- No **Object Inspector**, selecione o formulário **mmuRegRelatorio**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuRegRelatorioClick(Sender: TObject);  
begin  
    sbtRelatorio.Click;  
end;
```

- No **Object Inspector**, selecione o formulário **mmuArqSair**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.mmuArqSairClick(Sender: TObject);  
begin  
    Close;  
end;
```

- No **Object Inspector**, selecione o formulário **sbtRelatório**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;

- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.sbtRelatorioClick(Sender: TObject);
begin
    frmRelatorioDinamico.showmodal;
end;
```

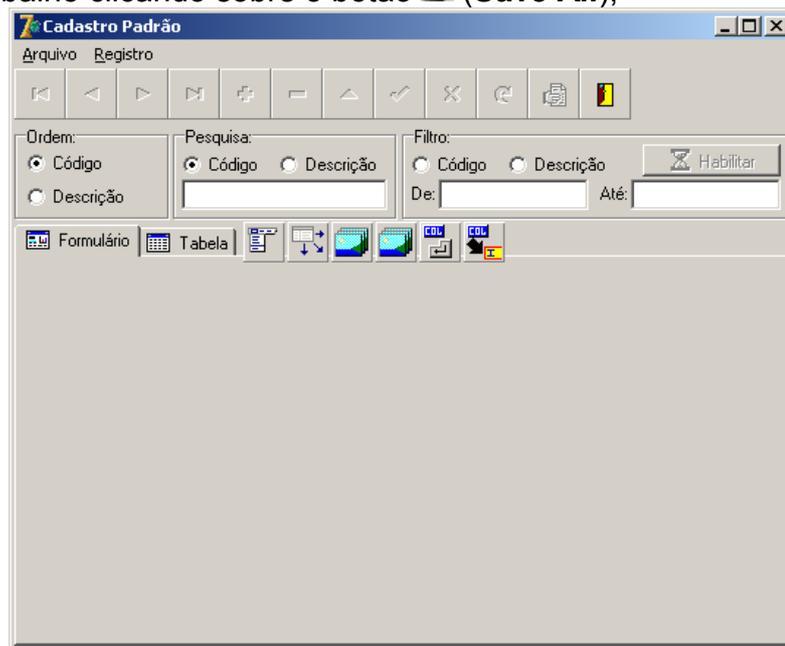
- No **Object Inspector**, selecione o formulário **sbtFechar**;

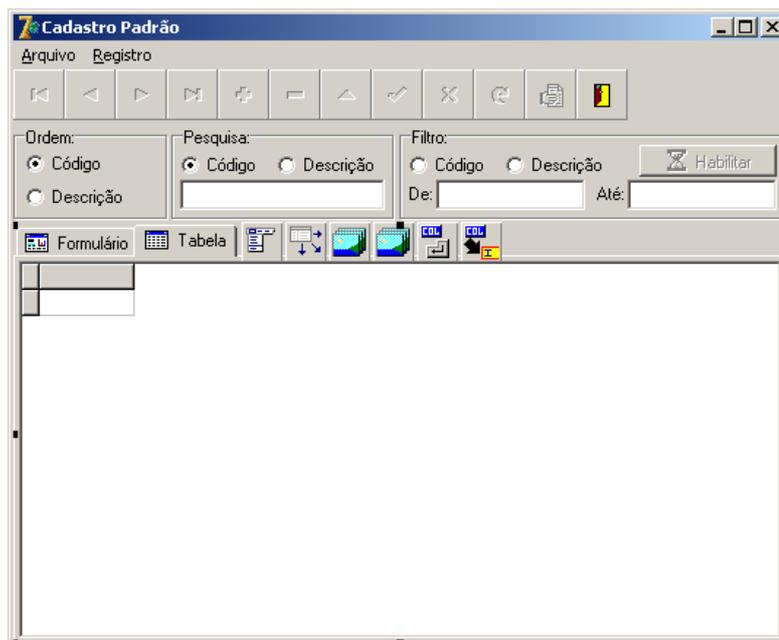


- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadPadrao.sbtSairClick(Sender: TObject);
begin
    mmuArqSair.Click;
end;
```

- Selecione **File / Use Unit... / untRelatorioDinamico / OK**;
- Selecione **Project / Options / Aba Forms / frmCadPadrao (em Auto-create forms) / clique no botão seta para direita > / OK**;
- Salve o trabalho clicando sobre o botão  (**Save All**);





### 11.2.10. Criando a Janela Cadastro de Grupos

- Crie uma nova janela clicando em **File / New / Other / Material (aba) / frmCadPadrao / OK**;
- Salve o trabalho clicando **File / Save As** ou clicando sobre o botão  (**Save**) que se encontra na barra de ferramentas. Na caixa de diálogo "Save Unit 1 As", selecione a pasta (diretório) **d:\piloto**. Na caixa de texto *Nome do Arquivo*, digite **untCadGrupo** e clique sobre o botão **Salvar**;
- No **Object Inspector**, selecione o formulário **frmCadPadrao1**;



- No **Object Inspector**, altere o valor da propriedade **Caption** para **Cadastro de Grupo**;
- No **Object Inspector**, altere o valor da propriedade **Name** para **frmCadGrupo**;
- Selecione **File / Use Unit... / untMaterial / OK**;
- Clique na paleta **Mercury** e insira no formulário **frmCadgrupo** o componente **MDOSQL** ;
- No **Object Inspector**, altere as seguintes propriedades:

| <b>Propriedade</b> | <b>Valor</b>             |
|--------------------|--------------------------|
| Database           | frmPrincipal.dbMaterial  |
| Name               | sqlAI                    |
| Transaction        | frmPrincipal.tstMaterial |

- Clique na paleta **Mercury** e insira no formulário **frmCadgrupo** o componente **MDOUpdateSQL**  .

- No **Object Inspector**, altere as seguintes propriedades:

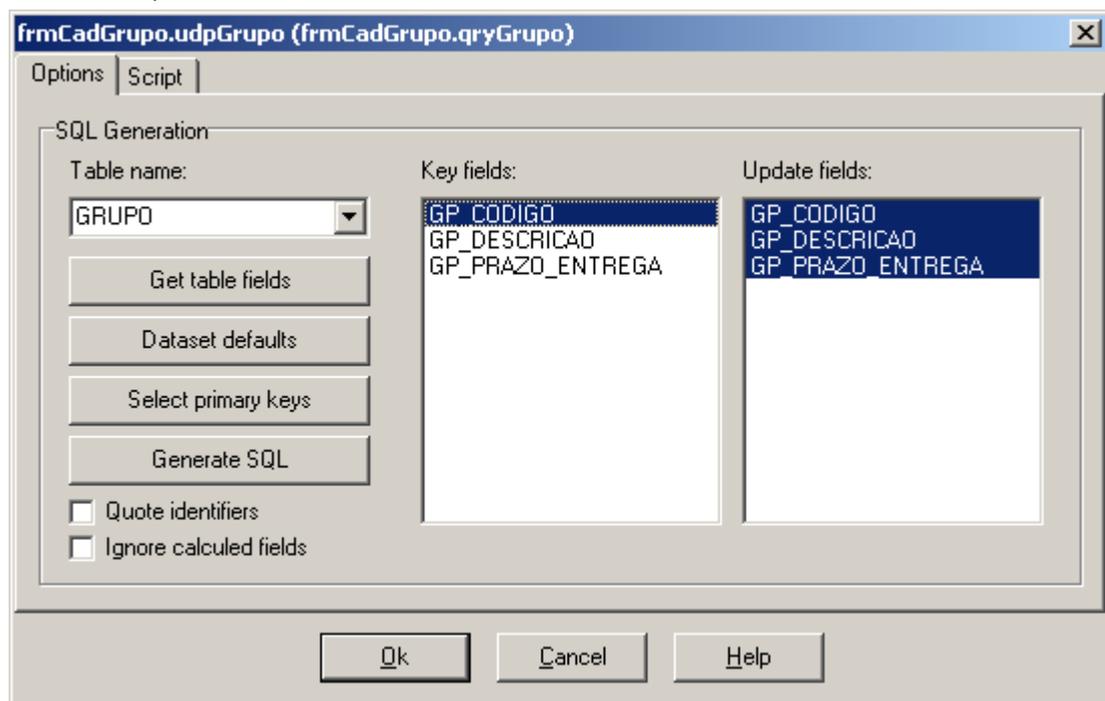
| <b>Propriedade</b> | <b>Valor</b> |
|--------------------|--------------|
| Name               | udpGrupo     |

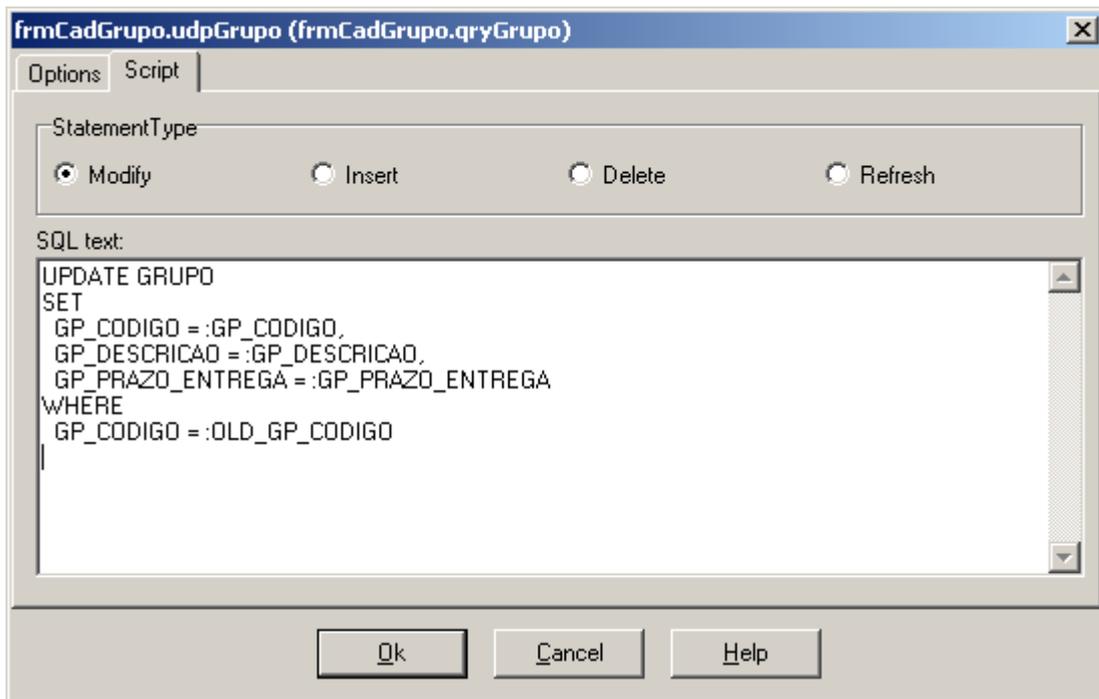
- Clique na paleta **Mercury** e insira no formulário *frmCadgrupo* o componente **MDOQuery** ;

- No **Object Inspector**, altere as seguintes propriedades:

| <b>Propriedade</b> | <b>Valor</b>                           |
|--------------------|----------------------------------------|
| Database           | frmPrincipal.dbMaterial                |
| Name               | qryGrupo                               |
| SQL                | select * from GRUPO order by GP_CODIGO |
| Transaction        | frmPrincipal.tstMaterial               |
| UpdateObject       | udpGrupo                               |
| Active             | True                                   |

- Dê um duplo-clique sobre o componente **udpGrupo**, e na caixa de diálogo apresentada – **aba Options** – clique nos botões: **Dataset defaults**, **Generate SQL** e **OK**;

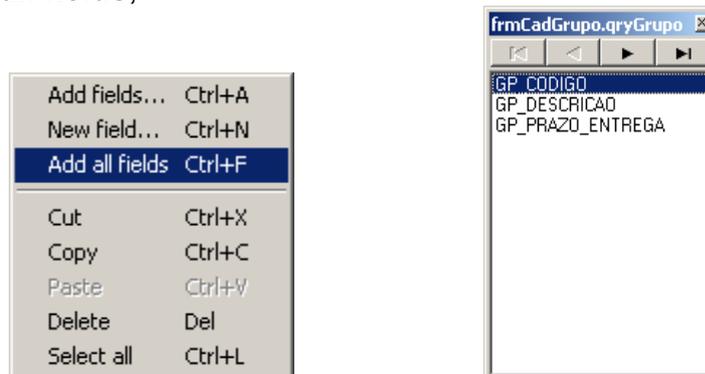




- Dê um duplo clique sobre o componente **qryGrupo**  para acessar a janela do *Editor Fields*;



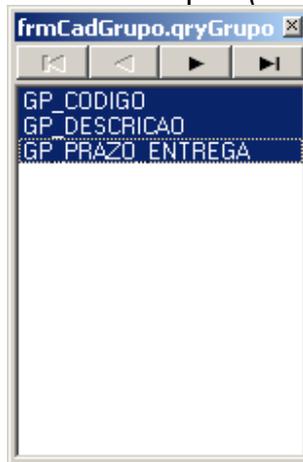
- Para inserir todos os campos da tabela no *Editor Fields*, dê um clique com o botão direito do mouse no interior do *Editor Fields* e no menu suspenso selecione a opção **Add all fields**;



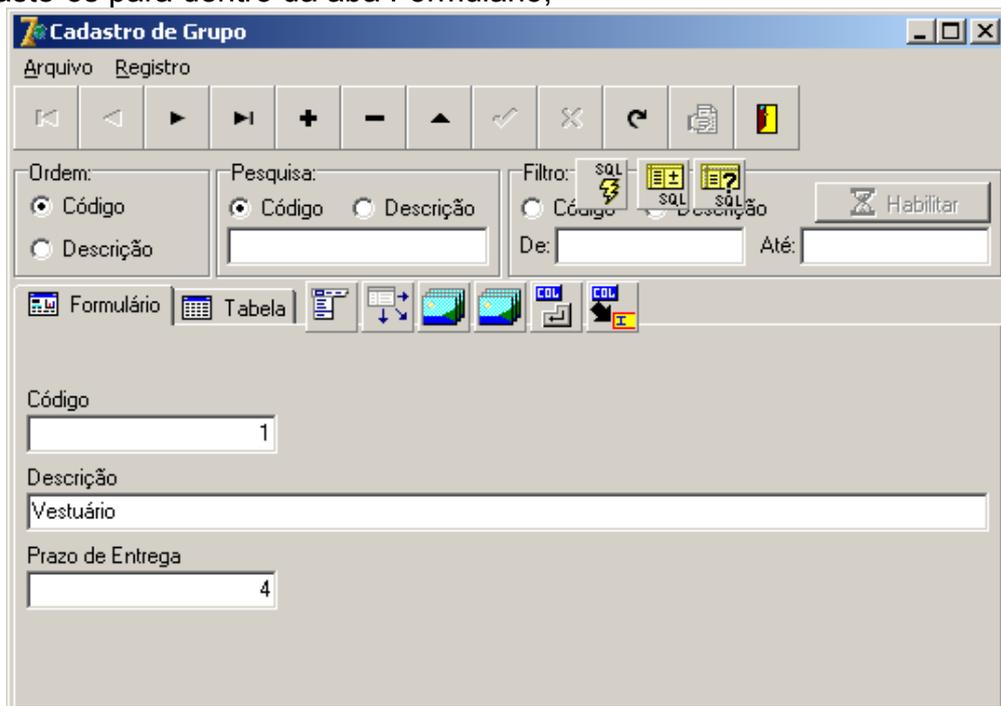
- Agora selecione o campo **GP\_CODIGO** e no **Object Inspector** mude o valor da propriedade **DisplayLabel** para **Código**;
- Agora selecione o campo **GP\_DESCRICAO** e no **Object Inspector** mude o valor da propriedade **DisplayLabel** para **Descrição**;
- Agora selecione o campo **GP\_PRAZO\_ENTREGA** e no **Object Inspector** mude o valor da propriedade **DisplayLabel** para **Prazo de Entrega**;
- No **Object Inspector**, selecione o formulário **dsRegistro**;



- No **Object Inspector**, altere o valor da propriedade **DataSet** para **qryGrupo**;
- No componente **PageControl**, clique sobre a aba chamada **Formulário** e por fim clique no interior desta aba para selecioná-la;
- Dê um duplo clique sobre o componente **qryGrupo**  para acessar a janela do **Editor Fields**;
- No **Editor Fields**, selecione todos os campos (**Select All**);



- Arraste-os para dentro da aba **Formulário**;



- No **Object Inspector**, selecione o formulário **qryGrupo**;



- Agora selecione a aba **Events** e em seguida o evento **AfterDelete**;
- Dê um duplo-clique no retângulo em branco a direita do evento AfterDelete e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadGrupo.qryGrupoAfterDelete(DataSet: TDataSet);  
begin  
    try  
        frmPrincipal.tstMaterial.CommitRetaining;  
    except  
        frmPrincipal.tstMaterial.Rollback;  
        raise;  
    end;  
end;
```

- No **Object Inspector**, selecione o formulário **qryGrupo**;



- Agora selecione a aba **Events** e em seguida o evento **AfterPost**;
- Dê um duplo-clique no retângulo em branco a direita do evento AfterPost e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadGrupo.qryGrupoAfterPost(DataSet: TDataSet);  
var  
    intCodigo: integer;  
begin  
    try  
        frmPrincipal.tstMaterial.CommitRetaining;  
        intCodigo := qryGrupo.FieldByName('GP_CODIGO').AsInteger;  
        qryGrupo.Active := False;  
        qryGrupo.Active := True;  
        qryGrupo.Locate('GP_CODIGO', intCodigo, []);  
    except  
        frmPrincipal.tstMaterial.Rollback;  
        raise;  
    end;  
end;
```

- No **Object Inspector**, selecione o formulário **qryGrupo**;



- Agora selecione a aba **Events** e em seguida o evento **BeforeDelete**;

- Dê um duplo-clique no retângulo em branco a direita do evento BeforeDelete e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadGrupo.qryGrupoBeforeDelete(DataSet: TDataSet);
var
    intResposta: integer;
begin
    intResposta :=
        Application.MessageBox('Deseja excluir este registro?',
            'Confirmação', mb_YesNo + mb_DefButton2 + mb_IconQuestion);
    if (intResposta = IDNO) then
        SysUtils.Abort;
end;
```

- No **Object Inspector**, selecione o formulário **qryGrupo**;



- Agora selecione a aba **Events** e em seguida o evento **OnNewRecord**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnNewRecord e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadGrupo.qryGrupoNewRecord(DataSet: TDataSet);
begin
    if (qryGrupo.FieldName('GP_CODIGO').IsNull) then
        begin
            sqlAI.SQL.Text := 'select max(GP_CODIGO) as CODIGO from GRUPO';
            sqlAI.ExecQuery;
            try
                qryGrupo.FieldName('GP_CODIGO').AsInteger :=
                    sqlAI.FieldName('CODIGO').AsInteger + 1;
            finally
                sqlAI.Close;
            end;
            qryGrupo.FieldName('GP_DESCRICAO').FocusControl;
        end;
end;
```

- No **Object Inspector**, selecione o formulário **qryGrupo**;



- Agora selecione a aba **Events** e em seguida o evento **OnPostError**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnPostError e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadGrupo.qryGrupoPostError(DataSet: TDataSet;
    E: EDatabaseError; var Action: TDataAction);
var
    strMsg, strTitulo: String;
begin
    strMsg := E.Message;
    strTitulo := 'Erro de Gravação';

    //Verifica se a chave primária foi violada.
    if Pos(UpperCase('violation of PRIMARY or UNIQUE KEY constraint "PK_'),
        UpperCase(E.Message)) <> 0 then
        begin
            strMsg := 'Já existe um grupo cadastrado com este código!';
            strTitulo := 'Violação';
        end;
end;
```

```

//Verifica se todos os campos foram preenchidos.
if Pos(UpperCase('*** null ***'), UpperCase(E.Message)) <> 0 then
begin
  if Pos(UpperCase('GP_CODIGO'), UpperCase(E.Message)) <> 0 then
  begin
    strMsg := 'Favor preencher o campo Código!';
    strTitulo := 'Campo Vazio';
    qryGrupo.FieldByName('GP_CODIGO').FocusControl;
  end
  else if Pos(UpperCase('GP_DESCRICAO'), UpperCase(E.Message)) <> 0 then
  begin
    strMsg := 'Favor preencher o campo Descrição!';
    strTitulo := 'Campo Vazio';
    qryGrupo.FieldByName('GP_DESCRICAO').FocusControl;
  end
  else if Pos(UpperCase('GP_PRAZO_ENTREGA'), UpperCase(E.Message)) <> 0 then
  begin
    strMsg := 'Favor preencher o campo Prazo de Entrega!';
    strTitulo := 'Campo Vazio';
    qryGrupo.FieldByName('GP_PRAZO_ENTREGA').FocusControl;
  end;
end;

//Mostra a mensagem do erro ocorrido.
Application.MessageBox(PChar(strMsg), PChar(strTitulo), MB_ICONEXCLAMATION +
  MB_OK);

Action := daAbort;
end;

```

- No **Object Inspector**, selecione o formulário **frmCadGrupo**;



- Agora selecione a aba **Events** e em seguida o evento **OnClose**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClose e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmCadGrupo.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  inherited;
  frmCadGrupo := nil;
end;

```

- Para facilitar a ordenação e o filtro dos dados, criaremos uma procedure para fazer isso. Declare a seguinte assinatura na sessão private:

```

type
  TfrmCadGrupo = class(TfrmCadPadrao)
    ...;
  private
    { Private declarations }
    procedure OrdemFiltro;
  public
    { Public declarations }
  end;

```

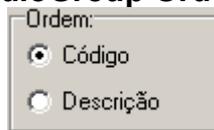
- Pressione as teclas CTRL + SHIFT + C para gerar a implementação dessa procedure e digite o código abaixo:

```

procedure TfrmCadGrupo.OrdemFiltro;
var
  strSQL: string;
begin
  qryGrupo.Close;
  if (btnFiltro.Caption = 'Desabilitar') then
  begin
    if (rbCodigoFiltro.Checked) Then
      strSQL := 'select * from GRUPO where GP_CODIGO >= ' +
        QuotedStr(edtDe.Text) + ' and GP_CODIGO <= ' +
        QuotedStr(edtAte.Text);
    if (rbDescricaoFiltro.Checked) Then
      strSQL := 'select * from GRUPO where upper(GP_DESCRICAO) >= ' +
        QuotedStr(UpperCase(edtDe.Text)) +
        ' and upper(GP_DESCRICAO) <= ' +
        QuotedStr(UpperCase(edtAte.Text));
    end
  else
    strSQL := 'select * from GRUPO';
    if (rgOrdem.ItemIndex = 0) then
      strSQL := strSQL + ' order by GP_CODIGO'
    else
      strSQL := strSQL + ' order by GP_DESCRICAO';
    qryGrupo.SQL.Text := strSQL;
    qryGrupo.Open;
  end;
end;

```

- Dê um duplo clique sobre o **RadioGroup Ordem**;



- Agora no Editor de Código digite as linhas em negrito a seguir:
- ```

procedure TfrmCadGrupo.rgOrdemClick(Sender: TObject);
begin
  OrdemFiltro;
end;

```

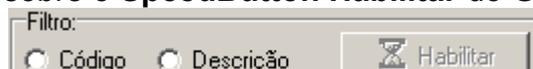
- Selecione o componente **Edit** do **GroupBox Pesquisa**;
- Agora selecione a aba **Events** e em seguida o evento **OnChange**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnChange e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmCadGrupo.edtPesquisaChange(Sender: TObject);
begin
  if (Trim(edtPesquisa.Text) = '') then
    Exit;
  if (rbCodigoPesq.Checked) then
    qryGrupo.Locate('GP_CODIGO', edtPesquisa.Text, [])
  else
    qryGrupo.Locate(
      'GP_DESCRICAO', edtPesquisa.Text,
      [loCaseInsensitive, loPartialKey]);
end;

```

- Dê um duplo clique sobre o **SpeedButton Habilitar** do **GroupBox Filtro**;



- Agora no Editor de Código digite as linhas em negrito a seguir:
 

```
procedure TfrmCadGrupo.btnFiltroClick(Sender: TObject);
begin
    inherited;
    OrdemFiltro;
end;
```
- Selecione **Project / Options / Aba Forms / frmCadGrupo (em Auto-create forms) / clique no botão seta para direita**  / **OK**;
- Clique em **View / Forms / frmPrincipal / OK**;
- Agora clique em **File / Use Unit / untCadGrupo / OK**. Ou seja, para chamarmos a tela de cadastro de grupos, precisamos fazer com que o *frmPrincipal* "enxergue" o *frmCadGrupo*;
- Selecione na janela **frmPrincipal** a opção de menu **Cadastro** e depois dê um clique sobre a opção **Grupo F2**;

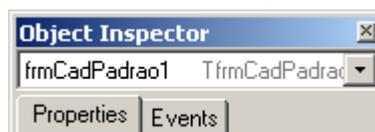


- Agora no editor de códigos digite as linhas em negrito do código a seguir:
 

```
procedure TfrmPrincipal.mmuCadGrupoClick(Sender: TObject);
begin
    if (frmCadGrupo = nil) then
        Application.CreateForm(TfrmCadGrupo, frmCadGrupo);
        frmCadGrupo.showmodal;
end;
```
- Salve o trabalho clicando sobre o botão  (**Save All**);

### 11.2.11. Criando a Janela Cadastro de Materiais

- Crie uma nova janela clicando em **File / New / Other / Material (aba) / frmCadPadrao / OK**;
- Salve o trabalho clicando **File / Save As** ou clicando sobre o botão  (**Save**) que se encontra na barra de ferramentas. Na caixa de diálogo "Save Unit 1 As", selecione a pasta (diretório) **d:\piloto**. Na caixa de texto *Nome do Arquivo*, digite **untCadMaterial** e clique sobre o botão **Salvar**;
- No **Object Inspector**, selecione o formulário **frmCadPadrao1**;



- No **Object Inspector**, altere o valor da propriedade **Caption** para **Cadastro de Material**;
- No **Object Inspector**, altere o valor da propriedade **Name** para **frmCadMaterial**;
- Selecione **File / Use Unit... / untMaterial / OK**;

- Clique na paleta **Mercury** e insira no formulário *frmCadMaterial* o componente **MDOSQL** ;

- No **Object Inspector**, altere as seguintes propriedades:

<b>Propriedade</b>	<b>Valor</b>
Database	frmPrincipal.dbMaterial
Name	sqlAI
Transaction	frmPrincipal.tstMaterial

- Clique na paleta **Mercury** e insira no formulário *frmCadMaterial* o componente **MDOUpdateSQL** ;

- No **Object Inspector**, altere as seguintes propriedades:

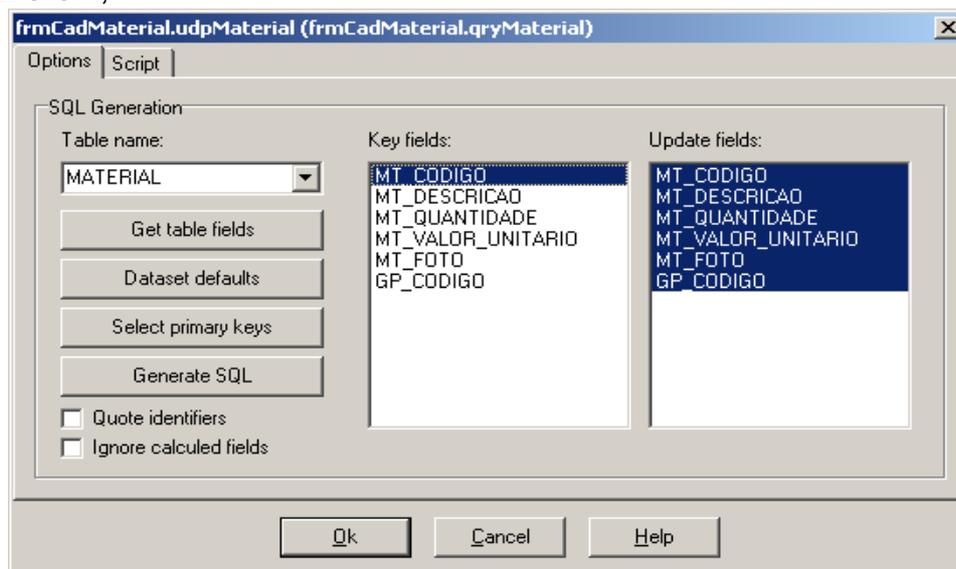
<b>Propriedade</b>	<b>Valor</b>
Name	udpMaterial

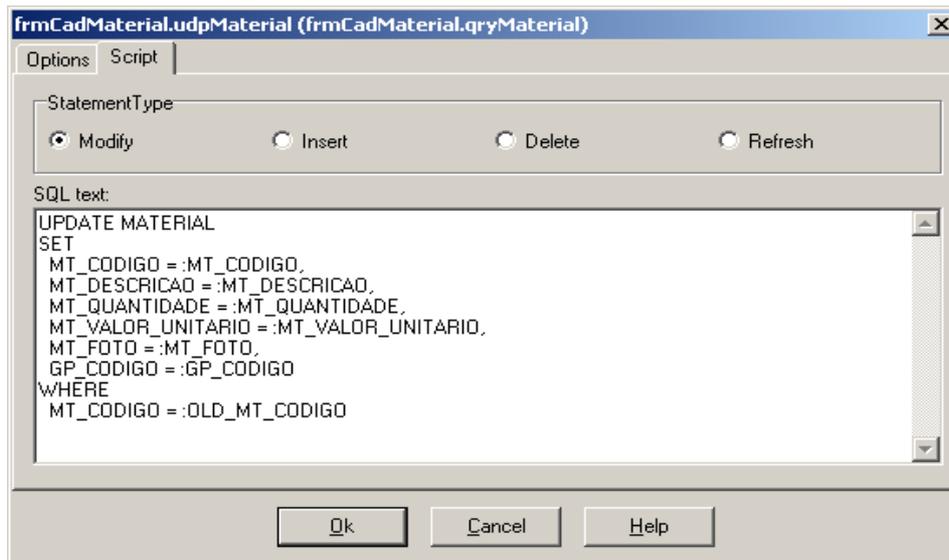
- Clique na paleta **Mercury** e insira no formulário *frmCadMaterial* o componente **MDOQuery** ;

- No **Object Inspector**, altere as seguintes propriedades:

<b>Propriedade</b>	<b>Valor</b>
Database	frmPrincipal.dbMaterial
Name	qryMaterial
SQL	select * from MATERIAL order by MT_CODIGO
Transaction	frmPrincipal.tstMaterial
UpdateObject	UdpMaterial

- Dê um duplo-clique sobre o componente **udpMaterial**, e na caixa de diálogo apresentada – **aba Options** – clique nos botões: **Dataset defaults**, **Generate SQL** e **OK**;

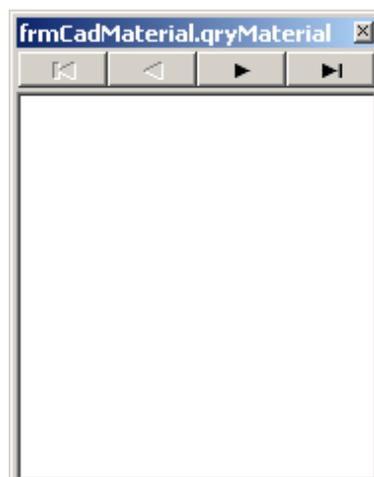




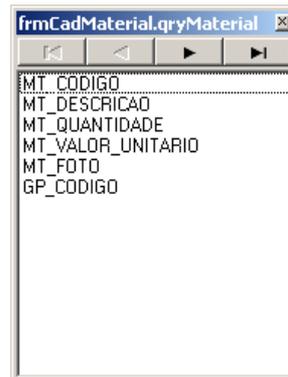
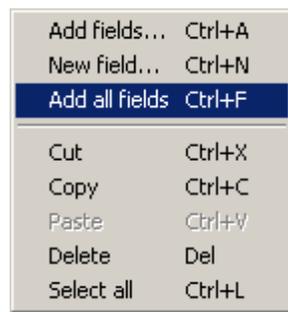
- Clique na paleta **Mercury** da barra de ferramentas do Delphi;
- Clique sobre o componente **MDOQuery**  para selecioná-lo, e em seguida clique em qualquer área interna da janela *frmCadMaterial* para inseri-lo;
- Altere suas propriedades conforme ilustra a tabela seguinte:

Propriedade	Valor
Database	frmPrincipal.dbMaterial
Name	qryGrupo
SQL	select * from GRUPO order by GP_DESCRICAO
Transaction	frmPrincipal.tstMaterial
Active	True

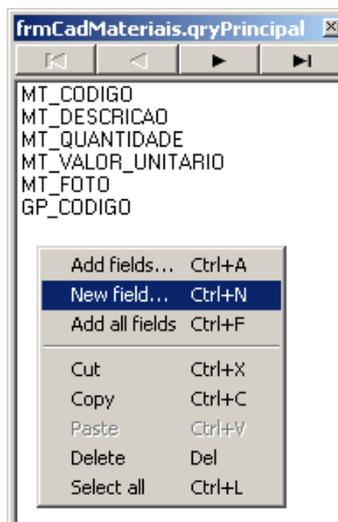
- Dê um duplo clique sobre o componente **qryMaterial**  para acessar a janela do *Editor Fields*;



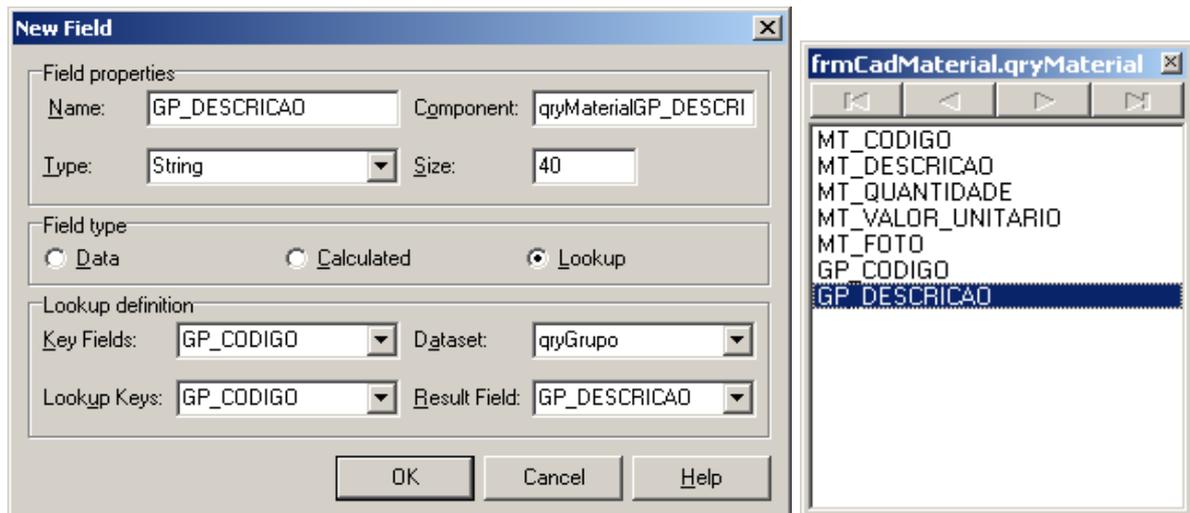
- Para inserir todos os campos da tabela no *Editor Fields*, dê um clique com o botão direito do mouse no interior do *Editor Fields* e no menu suspenso selecione a opção **Add all fields**;



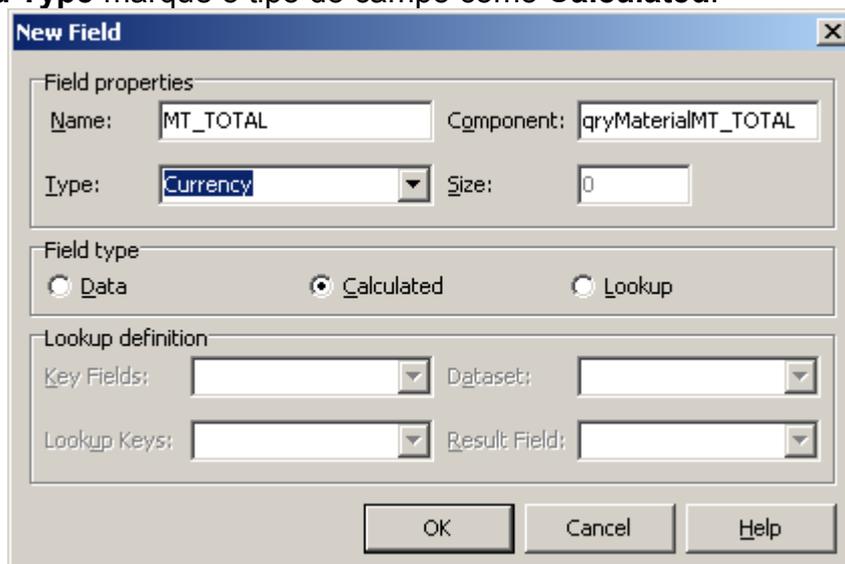
- Agora selecione o campo **MT\_CODIGO** e no **Object Inspector** mude o valor da propriedade **DisplayLabel** para **Código**;
- Agora selecione o campo **MT\_DESCRICAO** e no **Object Inspector** mude o valor da propriedade **DisplayLabel** para **Descrição**;
- Agora selecione o campo **MT\_QUANTIDADE** e no **Object Inspector** mude o valor da propriedade **DisplayLabel** para **Quantidade**;
- Agora selecione o campo **MT\_VALOR\_UNITARIO** e no **Object Inspector** mude o valor da propriedade **DisplayLabel** para **Valor Unitário**;
- Agora selecione o campo **MT\_FOTO** e no **Object Inspector** mude o valor da propriedade **DisplayLabel** para **Foto**;
- Agora selecione o campo **GP\_CODIGO** e no **Object Inspector** mude o valor da propriedade **Visible** para **False**;
- Quando precisamos mostrar campos que se encontra em outra tabela, temos que criar campos lookup. Para criar um campo lookup, na janela do *Editor Fields*, dê um clique com o botão direito do mouse em seu interior e escolha a opção **New Field**;



- Em seguida, a janela *New Field* será apresentada. Digite no campo **Name**: **GP\_DESCRICAO**, altere o campo **Type** para **String**, digite no campo **Size**: **40**, em **Field type** marque a opção **Lookup**, altere o campo **Key Fields** para **GP\_CODIGO**, altere o campo **Dataset** para **qryGrupo**, altere o campo **Lookup Keys** para **GP\_CODIGO** e o campo **Result Field** para **GP\_DESCRICAO** e agora clique no botão **OK**;



- Agora selecione o campo **GP\_DESCRICA0** e no **Object Inspector** mude o valor da propriedade **DisplayLabel** para **Grupo**;
- Dê duplo-clique sobre o objeto **qryMaterial** ;
- Tecele **[Ctrl] + [N]** para inserir um novo campo;
- Em **Fields Properties** digite o nome do campo na caixa de texto **Name** e escolha o tipo do mesmo em **Type**;  
**Name:** *MT\_TOTAL*      **Type:** *Currency*
- Em **Field Type** marque o tipo do campo como **Calculated**.



- Clique sobre o botão **OK**;
- Para que o campo calculado funcione inclua o cálculo no evento **OnCalcFields** do componente **qryMaterial**:

```

procedure TfrmCadMaterial.qryMaterialCalcFields(DataSet: TDataSet);
begin
    qryMaterial.FieldByName('MT_TOTAL').AsCurrency :=
        qryMaterial.FieldByName('MT_QUANTIDADE').AsCurrency *
        qryMaterial.FieldByName('MT_VALOR_UNITARIO').AsCurrency;
end;

```

- No **Object Inspector**, selecione o formulário **dsRegistro**;

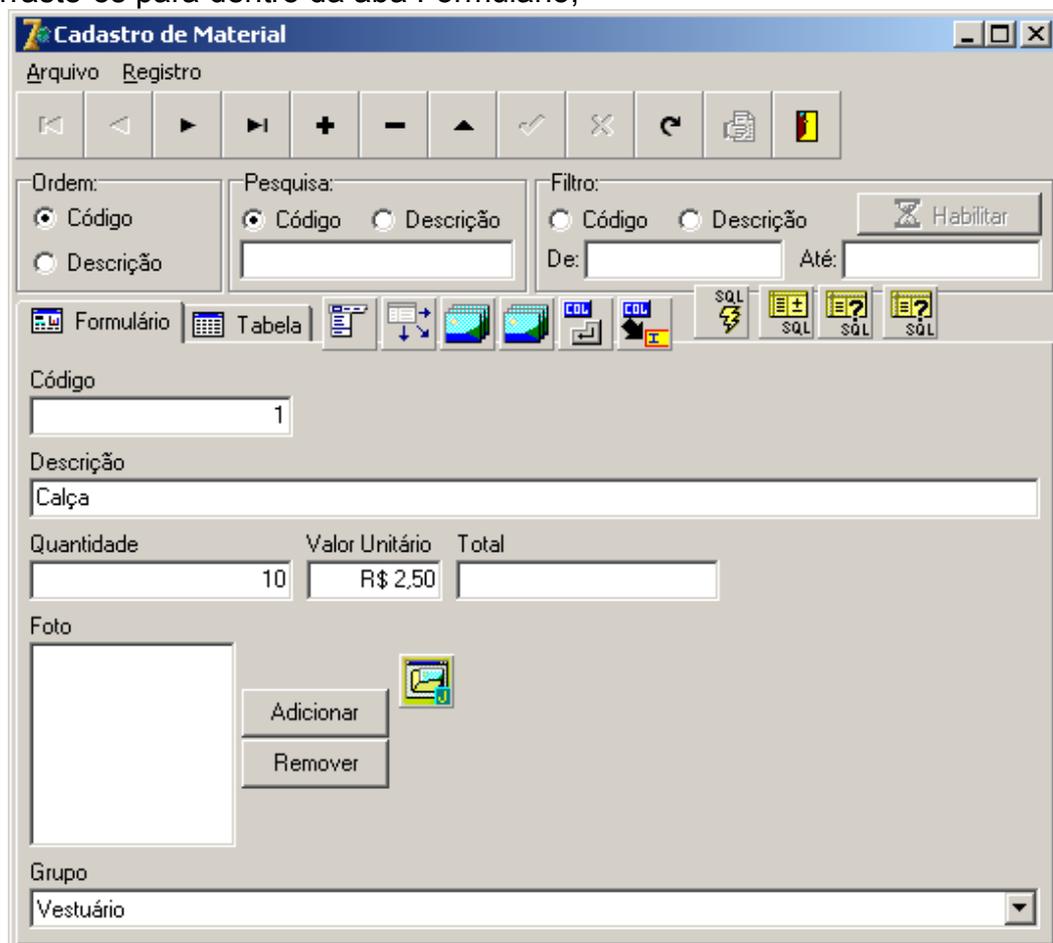


- No **Object Inspector**, altere o valor da propriedade **DataSet** para **qryMaterial**;
- No componente **PageControl**, clique sobre a aba chamada **Formulário** e por fim clique no interior desta aba para selecioná-la;
- Dê um duplo clique sobre o componente **qryMaterial**  para acessar a janela do **Editor Fields**;
- No **Editor Fields**, selecione todos os campos (**Select All**);



**Obs:** Pressione a tecla Ctrl e dê um clique no campo GP\_CODIGO para desmarcá-lo.

- Arraste-os para dentro da aba Formulário;



- Clique na paleta **Standard** na paleta de componentes do Delphi;
- Clique sobre o componente **Button**  para selecioná-lo, e em seguida clique na área interna da aba chamada *Formulário* para inseri-lo;
- Altere suas propriedades conforme ilustra a tabela seguinte:

Propriedade	Valor
Caption	Adicionar
Enabled	False
Left	111
Name	btnAdicionar
Top	216
Width	75

- Clique na paleta **Standard** na paleta de componentes do Delphi;
- Clique sobre o componente **Button**  para selecioná-lo, e em seguida clique na área interna da aba chamada *Formulário* para inseri-lo;
- Altere suas propriedades conforme ilustra a tabela seguinte:

Propriedade	Valor
Caption	Remover
Enabled	False
Left	111
Name	btnRemover
Top	242
Width	75

- Clique na paleta **Dialogs** da barra de ferramentas do Delphi;
- Clique sobre o componente **OpenPictureDialog**  para selecioná-lo, e em seguida deposite este componente ao lado do componente DBImage da foto;
- Altere suas propriedades conforme ilustra a tabela seguinte:

Propriedade	Valor
Filter	Todas as fotos (*.jpg;*.jpeg;*.bmp) *.jpg;*.jpeg;*.bmp Foto JPEG (*.jpg;*.jpeg) *.jpg;*.jpeg Foto Bitmap (*.bmp) *.bmp
InitialDir	. (ponto)
Name	opdFoto

- No **Object Inspector**, selecione o formulário *qryMaterial*;



- Agora selecione a aba **Events** e em seguida o evento **AfterDelete**;
- Dê um duplo-clique no retângulo em branco a direita do evento AfterDelete e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmCadMaterial.qryMaterialAfterDelete(DataSet: TDataSet);
begin
    try
        frmPrincipal.tstMaterial.CommitRetaining;
    except
        frmPrincipal.tstMaterial.Rollback;
        raise;
    end;
end;

```

- No **Object Inspector**, selecione o formulário *qryMaterial*;
- Agora selecione a aba **Events** e em seguida o evento **AfterPost**;

- Dê um duplo-clique no retângulo em branco a direita do evento AfterPost e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmCadMaterial.qryMaterialAfterPost(DataSet: TDataSet);
var
  intCodigo: integer;
begin
  try
    frmPrincipal.tstMaterial.CommitRetaining;
    intCodigo := qryMaterial.FieldName('MT_CODIGO').AsInteger;
    qryMaterial.Active := False;
    qryMaterial.Active := True;
    qryMaterial.Locate('MT_CODIGO', intCodigo, []);
  except
    frmPrincipal.tstMaterial.Rollback;
    raise;
  end;
end;

```

- No **Object Inspector**, selecione o formulário **qryMaterial**;



- Agora selecione a aba **Events** e em seguida o evento **BeforeDelete**;

- Dê um duplo-clique no retângulo em branco a direita do evento BeforeDelete e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmCadMaterial.qryMaterialBeforeDelete(DataSet: TDataSet);
var
  intResposta: integer;
begin
  intResposta :=
    Application.MessageBox('Deseja excluir este registro?',
      'Confirmação', mb_YesNo + mb_DefButton2 + mb_IconQuestion);
  if (intResposta = IDNO) then SysUtils.Abort;
end;

```

- No **Object Inspector**, selecione o formulário **qryMaterial**;



- Agora selecione a aba **Events** e em seguida o evento **OnNewRecord**;

- Dê um duplo-clique no retângulo em branco a direita do evento OnNewRecord e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmCadMaterial.qryMaterialNewRecord(DataSet: TDataSet);
begin
  if (qryMaterial.FieldName('MT_CODIGO').IsNull) then
  begin
    sqlAI.SQL.Text := 'select max(MT_CODIGO) as CODIGO from MATERIAL';
    sqlAI.ExecQuery;
    try
      qryMaterial.FieldName('MT_CODIGO').AsInteger :=
        sqlAI.FieldName('CODIGO').AsInteger + 1;
    finally
      sqlAI.Close;
    end;
    qryMaterial.FieldName('MT_DESCRICAÇÃO').FocusControl;
  end;
end;

```

- No **Object Inspector**, selecione o formulário **qryMaterial**;



- Agora selecione a aba **Events** e em seguida o evento **OnPostError**;
- Dê um duplo-clique no retângulo em branco a direita do evento **OnPostError** e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmCadMaterial.qryMaterialPostError(DataSet: TDataSet;
  E: EDatabaseError; var Action: TDataAction);
var
  strMsg, strTitulo: String;
begin
  strMsg := E.Message;
  strTitulo := 'Erro de Gravação';

  //Verifica se a chave primário foi violada.
  if Pos(UpperCase('violation of PRIMARY or UNIQUE KEY constraint "PK_'),
    UpperCase(E.Message)) <> 0 then
    begin
      strMsg := 'Já existe um material cadastrado com este código!';
      strTitulo := 'Violação';
    end;

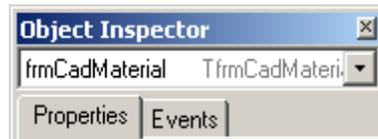
  //Verifica se todos os campos foram preenchidos.
  if Pos(UpperCase('*** null ***'), UpperCase(E.Message)) <> 0 then
    begin
      if Pos(UpperCase('MT_CODIGO'), UpperCase(E.Message)) <> 0 then
        begin
          strMsg := 'Favor preencher o campo Código!';
          strTitulo := 'Campo Vazio';
          qryMaterial.FieldByName('MT_CODIGO').FocusControl;
        end
      else if Pos(UpperCase('MT_DESCRICAO'), UpperCase(E.Message)) <> 0 then
        begin
          strMsg := 'Favor preencher o campo Descrição!';
          strTitulo := 'Campo Vazio';
          qryMaterial.FieldByName('MT_DESCRICAO').FocusControl;
        end
      else if Pos(UpperCase('MT_QUANTIDADE'), UpperCase(E.Message)) <> 0 then
        begin
          strMsg := 'Favor preencher o campo Quantidade!';
          strTitulo := 'Campo Vazio';
          qryMaterial.FieldByName('MT_QUANTIDADE').FocusControl;
        end
      else if Pos(UpperCase('MT_VALOR_UNITARIO'), UpperCase(E.Message)) <> 0 then
        begin
          strMsg := 'Favor preencher o campo Valor Unitário!';
          strTitulo := 'Campo Vazio';
          qryMaterial.FieldByName('MT_VALOR_UNITARIO').FocusControl;
        end
      else if Pos(UpperCase('GP_CODIGO'), UpperCase(E.Message)) <> 0 then
        begin
          strMsg := 'Favor selecionar um grupo!';
          strTitulo := 'Campo Vazio';
          qryMaterial.FieldByName('GP_CODIGO').FocusControl;
        end
    end;

  //Mostra a mensagem do erro ocorrido.
  Application.MessageBox(PChar(strMsg), PChar(strTitulo), MB_ICONEXCLAMATION +
    MB_OK);

  Action := daAbort;
end;

```

- No **Object Inspector**, selecione o formulário **frmCadMaterial**;



- Agora selecione a aba **Events** e em seguida o evento **OnClose**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClose e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmCadMaterial.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  inherited;
  frmCadMaterial := nil;
end;

```

- No **Object Inspector**, selecione o formulário **btnAdicionar**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnClick e no editor de código digite as linhas em negrito do código a seguir:

```

procedure TfrmCadMaterial.btnAdicionarClick(Sender: TObject);
var
  strArquivo: String;
  imagem: TPicture;
begin
  //Não esquecer de declarar na seção uses: Clipbrd, Jpeg
  if (opdFoto.Execute) then
  begin
    strArquivo := opdFoto.FileName;
    //Cria a variável para armazenar a imagem
    imagem := TPicture.Create();
    //Busca a imagem no arquivo e joga na variável
    imagem.LoadFromFile(strArquivo);
    //Passa imagem para o clipboard
    clipboard.Assign(imagem);
    //Copia a imagem para o DBImage
    DBImage1.PasteFromClipboard;
    //Libera a variável
    imagem.Free;
    btnRemover.Enabled := True;
  end;
end;

```

- No **Object Inspector**, selecione o formulário **btnRemover**;



- Agora selecione a aba **Events** e em seguida o evento **OnClick**;

- Dê um duplo-clique no retângulo em branco a direita do evento **OnClick** e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadMaterial.btnRemoverClick(Sender: TObject);
begin
    DBImage1.Picture.Bitmap.Assign(nil);
end;
```

- No **Object Inspector**, selecione o formulário **dsRegistro**;



- Agora selecione a aba **Events** e em seguida o evento **OnStateChange**;

- Dê um duplo-clique no retângulo em branco a direita do evento **OnStateChange** e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadMaterial.dsRegistroStateChange(Sender: TObject);
begin
    inherited;
    if (dsRegistro.DataSet.State = dsInsert) then
        btnAdicionar.Enabled := True
    else if (dsRegistro.DataSet.State = dsEdit) then
        begin
            btnAdicionar.Enabled := True;
            btnRemover.Enabled := True;
        end
    else
        begin
            btnAdicionar.Enabled := False;
            btnRemover.Enabled := False;
        end;
end;
```

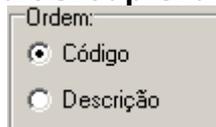
- Para facilitar a ordenação e o filtro dos dados, criaremos uma procedure para fazer isso. Declare a seguinte assinatura na sessão **private**:

```
type
    TfrmCadMaterial= class(TfrmCadPadrao)
        ...;
    private
        { Private declarations }
        procedure OrdemFiltro;
    public
        { Public declarations }
    end;
```

- Pressione as teclas CTRL + SHIFT + C para gerar a implementação dessa procedure e digite o código abaixo:

```
procedure TfrmCadMaterial.OrdemFiltro;
var
  strSQL: string;
begin
  qryMaterial.Close;
  if (btnFiltro.Caption = 'Desabilitar') then
    begin
      if (rbCodigoFiltro.Checked) Then
        strSQL := 'select * from MATERIAL where MT_CODIGO >= ' +
          QuotedStr(edtDe.Text) + ' and MT_CODIGO <= ' + QuotedStr(edtAte.Text);
      if (rbDescricaoFiltro.Checked) Then
        strSQL := 'select * from MATERIAL where upper(MT_DESCRICAO) >= ' +
          QuotedStr(UpperCase(edtDe.Text)) + ' and upper(MT_DESCRICAO) <= ' +
          QuotedStr(UpperCase(edtAte.Text));
      end
    else
      strSQL := 'select * from MATERIAL';
      if (rgOrdem.ItemIndex = 0) then
        strSQL := strSQL + ' order by MT_CODIGO'
      else
        strSQL := strSQL + ' order by MT_DESCRICAO';
      qryMaterial.SQL.Text := strSQL;
      qryMaterial.Open;
    end;
end;
```

- Dê um duplo clique sobre o **RadioGroup Ordem**;



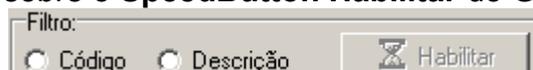
- Agora no Editor de Código digite as linhas em negrito a seguir:

```
procedure TfrmCadMaterial.rgOrdemClick(Sender: TObject);
begin
  OrdemFiltro;
end;
```

- Selecione o componente **Edit** do **GroupBox Pesquisa**;
- Agora selecione a aba **Events** e em seguida o evento **OnChange**;
- Dê um duplo-clique no retângulo em branco a direita do evento OnChange e no editor de código digite as linhas em negrito do código a seguir:

```
procedure TfrmCadMaterial.edtPesquisaChange(Sender: TObject);
begin
  if (Trim(edtPesquisa.Text) = '') then
    Exit;
  if (rbCodigoPesq.Checked) then
    qryMaterial.Locate('MT_CODIGO', edtPesquisa.Text, [])
  else
    qryMaterial.Locate(
      'MT_DESCRICAO', edtPesquisa.Text,
      [loCaseInsensitive, loPartialKey]);
end;
```

- Dê um duplo clique sobre o **SpeedButton Habilitar** do **GroupBox Filtro**;



- Agora no Editor de Código digite as linhas em negrito a seguir:

```
procedure TfrmCadMaterial.btnFiltroClick(Sender: TObject);
begin
  inherited;
  OrdemFiltro;
end;
```

- No **Object Inspector**, selecione o formulário **qryMaterial**;



- No **Object Inspector**, altere o valor da propriedade **Active** para **True**;
- Selecione **Project / Options / Aba Forms / frmCadMaterial (em Auto-create forms)** / clique no botão seta para direita  / **OK**;
- Clique em **View / Forms / frmPrincipal / OK**;
- Agora clique em **File / Use Unit / untCadMaterial / OK**. Ou seja, para chamarmos a tela de cadastro de grupos, precisamos fazer com que o *frmPrincipal* "enxergue" o *frmCadMaterial*;
- Selecione na janela **frmPrincipal** a opção de menu **Cadastro** e depois dê um clique sobre a opção **Material F3**;



- Agora no editor de códigos digite as linhas em negrito do código a seguir:  

```

procedure TfrmPrincipal.mmuCadMaterialClick(Sender: TObject);
begin
  if (frmCadMaterial = nil) then
    Application.CreateForm(TfrmCadMaterial, frmCadMaterial);
    frmCadMaterial.showmodal;
end;
```
- Salve o trabalho clicando sobre o botão  (**Save All**);

## 12. Tratamento e Tradução dos Erros Globais

- Clique em **View / Forms / frmPrincipal / OK**;
- Clique na paleta **Additional** da barra de ferramentas do Delphi;
- Clique sobre o componente **ApplicationEvents**  para selecioná-lo, e em seguida deposite este componente no formulário *frmPrincipal*;
- Altere suas propriedades conforme ilustra a tabela seguinte:

Propriedade	Valor
Name	aevPrincipal

- No **Object Inspector**, selecione a aba **Events**, depois selecione o evento **OnException** e agora dê um duplo-clique no quadro branco a direita;

- No *Code Editor* (Editor de Código), digite as linhas em negrito do código abaixo:

```
procedure TfrmPrincipal.aevPrincipalException(Sender: TObject;
  E: Exception);
var
  strTitulo, strMsg: String;
begin
  strTitulo := 'Erro Global';
  strMsg := E.Message;
  //Tradução das mensagens de erro
  if (E is EConvertError) then
    begin
      strTitulo := 'Erro de Conversão';
      if (Pos('is not a valid integer value', E.Message) > 0) then
        strMsg := 'Não é um valor inteiro válido'
      else if (Pos('is not a valid floating point value',
        E.Message) > 0) then
        strMsg := 'Não é um valor real válido'
      else if (Pos('is not a valid date', E.Message) > 0) then
        strMsg := 'Data inválida'
      else if (Pos('is not a valid time', E.Message) > 0) then
        strMsg := 'Hora inválida';
    end
  else if (E is EDivByZero) then //Divisão (DIV)
    begin
      strTitulo := 'Erro de Divisão';
      if (Pos('Division by zero', E.Message) > 0) then
        strMsg := 'Divisão por zero';
    end
  else if (E is EZeroDivide) then //Divisão (/)
    begin
      strTitulo := 'Erro de Divisão';
      if (Pos('Floating point division by zero', E.Message) > 0) then
        strMsg := 'Divisão de um valor real por zero';
    end;
  //Mostrando a mensagem de erro
  Application.MessageBox(PChar(strMsg), PChar(strTitulo),
    MB_ICONEXCLAMATION + MB_OK);
  //Colocando o foco no componente onde o erro ocorreu
  if (Sender is TEdit) then //uses StdCtrls
    (Sender as TEdit).SetFocus;
end;
```

- Salve o trabalho clicando sobre o botão  (**Save All**);