

Lazarus

08 - Construção de Relatórios com o Lazarus

Marcos Roberto Ribeiro



Instituto Federal Minas Gerais - Campus Bambuí

2018

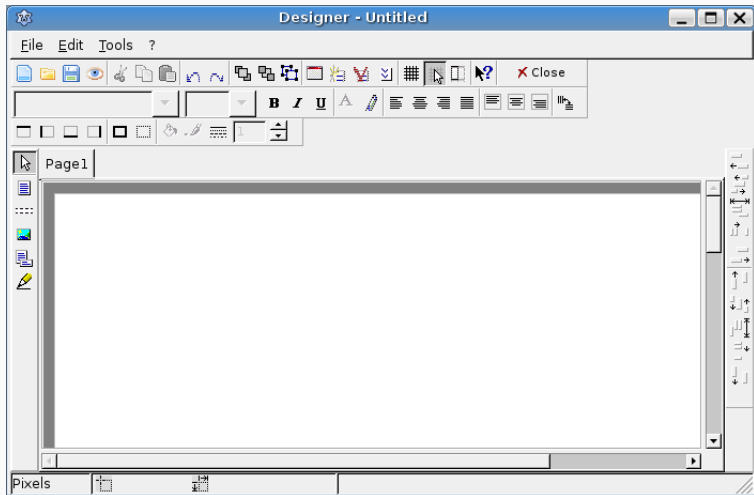
- O conjunto de componentes mais maduros desenvolvidos para o Lazarus é o *LazReport*;
- O *LazReport* é disponibilizado junto com o Lazarus, mas não é instalado por padrão. sua instalação pode ser feita via atalhos globais;
- Após o pacote de componentes ser instalado é exibida a paleta de componentes *LazReport*, na qual podemos destacar os seguintes componentes:
 - frReport** Componente principal para criação e exibição de relatórios;
 - frDBDataSet** Componente para utilizar bancos de dados em relatórios;
- Existem outros componentes que agregam mais funcionalidades aos relatórios, mas não os abordaremos nesta aula.

Os Componente frReport e frDBDataSet

- A propriedade mais importante do componente *frDBDataSet* é *DataSet* onde definimos o componente com o conjunto de dados que serão usados no relatório;
- No caso do componente *frReport* devemos definir como *DataSet* o componente *frDBDataSet* desejado;
- Os principais métodos do *frReport* são:
 - LoadFromFile()** Carrega um relatório armazenado em arquivo;
 - ShowReport()** Exibe o relatório;
- Depois de conectarmos os componentes ao banco de dados podemos entrar no *Design Report* para “desenhar” o relatório;

O Editor de Relatórios

- O Editor de Relatórios possui diversas barras de ferramentas como podemos ver na figura abaixo:



Componentes de um relatórios

- O componente fundamental de relatórios é o *band*. Um *band* é uma região do relatório onde podem ser inseridos elementos de impressão;
- Quando inserimos um *band* no relatório devemos selecionar o seu tipo, os tipos com os quais iremos trabalhar são:

Report title Título do relatório (apenas no início no relatório);

Report summary Sumário do relatório (apenas no final do relatório);

Page header Cabeçalho da página (apenas no início da página);

Page footer Rodapé da página (apenas no fim da página);

Master header Cabeçalho dos dados principais (apenas no início dos dados);

Master data Dados principais (se repete para cada registro existente);

Master footer Rodapé dos dados principais (apenas no fim dos dados);

Group header Cabeçalho de agrupamento (repete no início de cada agrupamento);

Group footer Rodapé de agrupamento (repete no fim de cada agrupamento);

- Dentro de cada *band* podemos inserir componentes como retângulos, linhas e figuras;
- Os retângulos podem ser preenchidos com textos estáticos ou serem ligados a dados de um banco de dados;

Dica

As propriedades de cada componente do relatório podem ser modificados através do *Object Inspector* acessado pela tecla *F11*.

Relatório de Listagem Simples

- Para nosso primeiro relatório vamos utilizar o banco de dados da aula anterior e criar um projeto com o seguinte formulário¹:

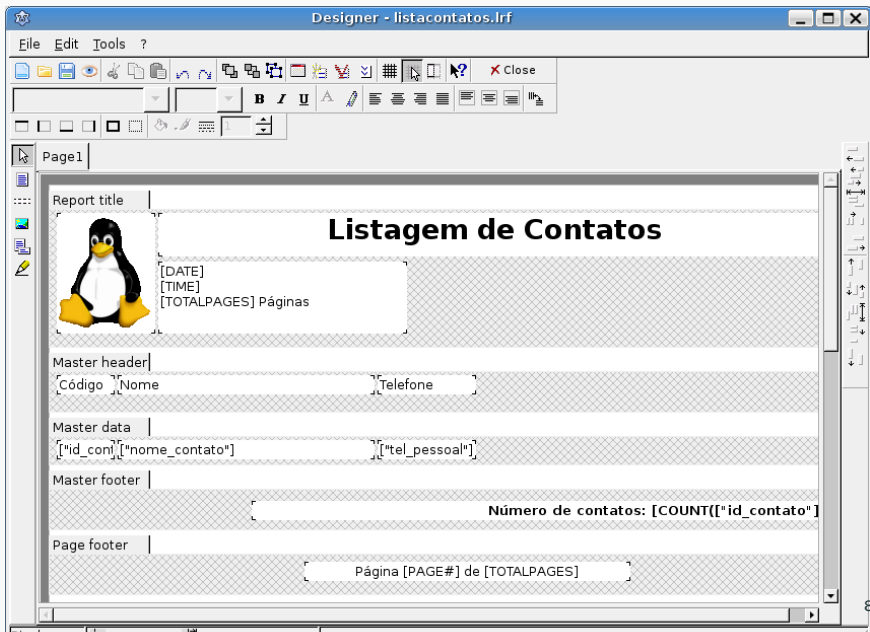


¹Poderíamos incluir este formulário no projeto agenda que já possui os componentes de conexão, porém o desenvolvimento de relatórios em projeto com muito componentes de acesso a dados pode se tornar mais lento. Quando o relatório estiver pronto, podemos adicioná-lo a qualquer outro projeto.

Propriedades do Formulário Principal

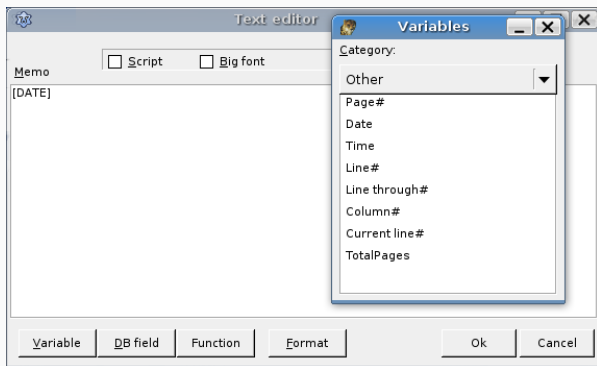
```
object formPrincipal: TformPrincipal
  Caption = 'Exemplos de Relatórios'
  object buttonListagem: TButton
    Caption = '&Listagem de Contatos'
  end
  object buttonContatosCategoria: TButton
    Caption = 'Contatos por Categoria'
  end
  object ZConPrincipal: TZConnection
    Protocol = 'postgresql-8'
    HostName = 'localhost'
    Database = 'agenda'
    User = 'postgres'
    Password = 'postgres'
    Connected = True
  end
  object zqueryConsulta: TZReadOnlyQuery
    Connection = zconPrincipal
    Active = True
    SQL.Strings = ('select * from contato')
  end
  object reportPrincipal: TfrReport
    Dataset = reportDB
  end
  object reportDB: TfrDBDataSet
    DataSet = zqueryConsulta
  end
end
```

Agora entramos no editor de relatórios (através do menu popup *Design Report* do *frReport*) e desenharmos o relatório exibido abaixo:



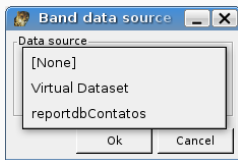
Desenhando o Relatório

- No *band Report title* inserimos uma figura (*picture*), um retângulo com o título e outro retângulo informando a data a hora e o número de páginas;
- Sempre que inserimos um retângulo o editor exibe a janela *Text Editor*, onde podemos inserir campos do banco de dados (*DB Field*), funções *Function* e variáveis (*Variable*) ou ainda formatar (*Format*) o texto do retângulo;
- As informações sobre data, hora e número de páginas são obtidas através das variáveis *[DATE]*, *TIME* e *TOTALPAGES*², respectivamente;

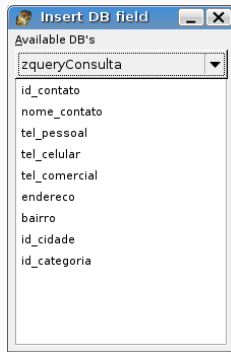


²Para que o número de páginas apareça corretamente devemos marcar a opção *Two-page report* no menu *File > Report Options*

- No *band Master header* adicionamos os cabeçalhos para os dados, para o nosso exemplo vamos exibir apenas código, nome e telefone;
- Quando inserimos o *band Master data* e selecionamos o tipo *Master data* devemos informar também o *DataSet* que será usado para este *band*, no nosso caso selecionamos *reportdbContatos*;



- Depois inseridos um retângulo no *band* para cada coluna a ser exibida e vinculamos cada um com seu respectivo campo;



- Quando selecionamos um campo, o editor automaticamente preenche o texto com *[zqueryConsulta."campo"]*, mas como queremos utilizar o relatório em outro projeto devemos modificar o texto para *["campo"]^a*.

^a Quando já sabemos os nomes dos campos podemos digitar d

- No *band Master footer* vamos inserir o total de contatos, para isto vamos usar a função *COUNT* sobre o campo *id_contato*;

Insert Expression

Expression

Insert

DB fields Variable Function

+ - * / = <> > < >= <= and or not

OK Cancel

LR_FunctEditorForm

Aggregate	AVG
Date and time	COUNT
String	MAX
Other	MIN
Math	SUM

COUNT(<BandName>)

Returns count of data-rows given in the <BandName>.

OK Cancel

LR_FunctEditor1Form

COUNT(<BandName>)

Returns count of data-rows given in the <BandName>.

Arguments

Argument 1 [zqueryConsulta."id_contato"]

Argument 2

Argument 3

OK Cancel

Insert Expression

Expression

1 Total de contatos: COUNT(["id_contato"])

Insert

DB fields Variable Function

+ - * / = <> > < >= <= and or not

OK Cancel

Exibição do Relatório

- No *band Page footer* utilizamos as variáveis *PAGE#* (página atual) e *TOTALPAGES*;
- Ao finalizarmos o desenho do relatório, nós o salvamos na mesma pasta do projeto com o nome de *listacontatos.lrf*
- Para exibir o relatório pelo programa inserimos o seguinte código no clique do botão "listagem de contatos":

```
reportPrincipal.LoadFromFile('listacontatos.lrf');  
reportPrincipal.ShowReport();
```



Automatizando a Exibição

- O próximo relatório que iremos desenvolver também utilizará como conjunto de dados o *zqueryConsulta*, então devemos cuidar para que o relatório atual (Listagem de contatos) seja exibido corretamente;
- Para fazermos isto vamos modificar o clique do botão “Listagem de Contatos” para o seguinte:

```
zqueryConsulta.Close();  
zqueryConsulta.SQL.Clear();  
zqueryConsulta.SQL.Add('SELECT * from contato');  
zqueryConsulta.Open();  
reportPrincipal.LoadFromFile(  
↪ ExtractFilePath(Application.ExeName) +  
↪ 'listacontatos.lrf');  
reportPrincipal.ShowReport;
```

Relatório com Agrupamento

- Para criarmos nosso próximo relatório vamos modificar a propriedade SQL do *zqueryConsulta* para o seguinte:

```
SELECT ca.den_categoria AS categoria,
       co.nome_contato AS nome,
       COUNT(e.email) AS nemails
FROM contato AS co,
     categoria AS ca,
     email AS e
WHERE co.id_categoria = ca.id_categoria
AND co.id_contato = e.id_contato
GROUP BY ca.den_categoria,
         co.nome_contato
ORDER BY ca.den_categoria
```

No editor de relatórios vamos desenhar o seguinte relatório:

Designer - contatocategoria.lrf

File Edit Tools ?

Close

Page1

Report title

Contatos por Categoria

[DATE]

Group header

["categoria"]

Master header

Nome **Nº E-mails**

Master data

["nome"] ["nemails"]

Group footer

Total desta categoria: SUM(["nemail

Report summa

Total de Contatos: SUM(["nema

Page footer

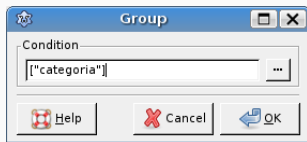
Página [PAGE#] de [TOTALPAGES]

Pixels

15/19

Desenhando o Relatório

- Neste relatório adicionamos um *band Group header* e outro *band Group footer* que serão responsáveis por agrupar os contatos por categoria;
- Quando inserimos o *Group header* devemos selecionar a condição de agrupamento (*GroupCondition*) que no nosso caso será ["categoria"];



- Também inserimos um campo no *Group header* para exibir a categoria;
- No caso do *band Group footer* inserimos a função *SUM* para somar o total de contatos de cada categoria. O primeiro parâmetro da função é o campo a ser somado e o segundo é o *band* que deve ser considerado na soma, ou seja, *[SUM(["nemails"], MasterData1)]*;
- No *band Report summary* utilizamos a mesma função do *Group footer*, mas como agora a função está no *band Report summary*, seu resultado será o total geral;

Automatizando a Exibição do Relatório

- Vamos salvar o relatório como “contatocategoria.lrf” e modificar o clique do botão “Contatos por Categoria” para o seguinte:

```
zqueryConsulta.Close();
zqueryConsulta.SQL.Clear();
zqueryConsulta.SQL.Add('SELECT ca.den_categoria AS categoria,');
zqueryConsulta.SQL.Add('      co.nome_contato AS nome,');
zqueryConsulta.SQL.Add('      count(e.email) AS nemails');
zqueryConsulta.SQL.Add('FROM contato AS co,');
zqueryConsulta.SQL.Add('      categoria AS ca,');
zqueryConsulta.SQL.Add('      email AS e');
zqueryConsulta.SQL.Add('WHERE co.id_categoria =ca.id_categoria');
zqueryConsulta.SQL.Add('  AND co.id_contato = e.id_contato');
zqueryConsulta.SQL.Add('GROUP BY ca.den_categoria,');
zqueryConsulta.SQL.Add('      co.nome_contato');
zqueryConsulta.SQL.Add('ORDER BY ca.den_categoria');
zqueryConsulta.Open();
reportPrincipal.LoadFromFile(ExtractFilePath(Application.ExeName) +
↳ 'contatocategoria.lrf');
reportPrincipal.ShowReport;
```

Exibição do Relatório

Contatos por Categoria	
23-01-2010	
<i>Amigos</i>	
Nome	Nº E-mails
José	2
Total desta categoria:	2
<i>Comercial</i>	
Nome	Nº E-mails
Antônio	1
Joaquim	2
Total desta categoria:	3
<i>Família</i>	
Nome	Nº E-mails
Maria	3
Total desta categoria:	3
Total de Contatos:	8

- FastReport Documentation. Disponível em http://fastreport.free.fr/Frenglish/Fr1_en.html;
- Tutorial de LazReport. Freepascal Wiki. Disponível em http://wiki.freepascal.org/Tutorial_de_LazReport;
- LazReport Documentation. Lazarus Wiki. Disponível em http://wiki.lazarus.freepascal.org/LazReport_Documentation.