

Nesta seção você encontra artigos intermediários sobre Delphi Win32 e Delphi .NET



Others

## Dicas de RAVE Reports

Veja como criar relatórios em RAVE reports, configurá-los e exportá-los para PDF

Utilizado para exibir e reportar resultados de buscas, cálculos, gráficos, entre inúmeras outras funcionalidades, os relatórios são indispensáveis aos nossos sistemas. Geralmente para cada tela de cadastro, temos no mínimo um relatório para exibir os registros, e que podem ser exibidos de forma contínua e total, ou em relatórios customizados com informações resultantes a uma busca na base de dados.

Partindo desta metodologia, veremos no decorrer deste artigo a utilização de um dos geradores de relatórios mais utilizados: RAVE Reports. Para que possamos perceber de forma concreta algumas das dicas, utilizaremos o Delphi 7 para criar uma aplicação com acesso a banco de dados e após isso veremos formas de customização e apresentação do *layout* de registros em *runtime*, linhas “zebradas” e também como criar parâmetros para destacar registros no corpo do relatório.

Acessando a base de dados *Employee.fdb*,

que acompanha a instalação do *Firebird*, criaremos também um relatório que será salvo diretamente em arquivo via programação. Iremos salvar o mesmo em uma extensão *.pdf*, que poderá ser enviado diretamente neste formato ao seu cliente.

### Criando a aplicação

Neste artigo utilizarei o Delphi 7 para criação da aplicação e demonstração dos exemplos e dicas de relatórios, mas fique à vontade para utilizar outra versão do Delphi para criação do projeto na plataforma Win32. Crie uma nova aplicação através do menu *File|New>Application*. Altere a propriedade *Name* do formulário inicial para “*frmPrincipal*” e a propriedade *Caption* para “*Dicas avançadas de RAVE Reports*”. Salve a *Unit* principal do projeto como “*uPrincipal.pas*” e o projeto salve como “*prjRelatorios.dpr*”.

Na primeira parte deste artigo veremos a criação de um relatório no RAVE Re-



**Maikel Marcelo Scheid**

([maikelscheid@gmail.com](mailto:maikelscheid@gmail.com))

é técnico em Informática com ênfase em Análise e Programação de Sistemas. Atua na área de Desenvolvimento de Softwares em Delphi para plataforma Win32 e .NET com banco de dados Firebird e MSSQL. É membro da Equipe Editorial ClubeDelphi.



ports, onde através de uma pesquisa SQL ao banco de dados *Employee.fdb* que será configurado a seguir, selecionaremos os registros referentes à tabela *Employee*. Serão então exibidos e configurados no relatório, de forma que a primeira formatação será a alternância entre as cores de cada bloco de registros (zebrar linhas). Faremos um tratamento através de uma condição *if* criada em “*Rave Language*” onde definiremos um parâmetro de valor do salário do funcionário. Esse parâmetro fará com que todos os valores excedentes tenham sua cor alterada.

Para criar a conexão ao banco de dados e estruturar os componentes de pesquisas, utilizaremos um *Data Module*, que deverá ser criado através do menu *File|New>Data Module*. Altere seu nome para “DM” e salve a *Unit* como “uDM.pas”. Adicione um componente *SQLConnection* da paleta *dbExpress* e altere seu nome para “sqlConexao”.

Para configurar os parâmetros de acesso ao banco de dados, dê um duplo clique sobre o componente *sqlConexao* e crie uma nova conexão utilizando o ícone de *Add Connection*, definindo o *Driver Name* para “Interbase” e *Connection Name* para “Employee” (**Figura 1**). Confirme e prossiga para a configuração dos demais parâmetros, começando pela propriedade *Database* onde deverá ser informado o caminho da base de dados *Employee.fdb* (<caminho>\employee.fdb). Defina as credenciais para acesso ao Firebird, usuário “SYSDBA” e senha “masterkey”.

Na propriedade *ServerCharSet* utilizaremos o padrão *Win1252* e em *SQLDialect* utilize o padrão 3 para que possamos utilizar comandos de SQL mais complexos (**Figura 2**).

Ainda no componente *sqlConexao*, de-

finia sua propriedade *LoginPrompt* para *False* e altere a propriedade *Connected* para *True* testando assim o sucesso da conexão.

Posicionado sobre a paleta *dbExpress*, adicione agora um componente *SQLDataSet* (“*sqldsEmployee*”) e ligue a propriedade *SQLConnection* ao componente de conexão *sqlConexao*. Adicione o seguinte comando SQL na propriedade *CommandText* do componente:

```
SELECT
  EMPLOYEE.FULL_NAME, EMPLOYEE.PHONE_EXT,
  DEPARTMENT.DEPARTMENT, EMPLOYEE.SALARY
FROM EMPLOYEE
INNER JOIN DEPARTMENT
ON (EMPLOYEE.DEPT_NO = DEPARTMENT.DEPT_NO)
```

Posicionando-se agora sobre a paleta *Data Access* adicione um componente *DataSetProvider* (“*dspEmployee*”) e relacione a propriedade *DataSet* ao *sqldsEmployee*. Ainda da paleta *Data Access* adicione um componente *ClientDataSet* (“*cdsEmployee*”), ligando-o ao *dspEmployee* através da propriedade *ProviderName*. Altere ainda sua propriedade *Active* para *True* e com um duplo clique sobre o componente abra o *Fields Editor* e adicione todos os campos gerados pelo código SQL através da opção *Add All Fields* disponível ao clique do botão direito do mouse sobre o editor.

**Nota:** Uma boa prática de programação e economia de tempo é selecionar todos os campos do *Fields Editor* individualmente e alterar a propriedade *DisplayLabel* de acordo com o nome do campo. Dessa forma sempre que usarmos tais campos os mesmos já terão um nome amigável.

Volte agora ao formulário principal do sistema onde criaremos uma estrutura simples para a exibição dos registros e a chamada para o relatório. Adicione ao formulário um componente *DataSource* (“*dsEmployee*”). Pressione simultaneamente as teclas *Alt + F11* e na tela que se abre escolha o *Data Module* da aplicação para que o Delphi inclua a *Unit* dele ao *Uses* do formulário.

Ligue a propriedade *DataSet* do componente ao *cdsEmployee*. Arraste também para o formulário um componente *DBGrid* da paleta *Data Controls* e ligue-o ao *dsEmployee* pela propriedade *DataSource*. Arraste ainda um componente *BitBtn* alterando sua propriedade *Caption* para “Imprimir em Rave Reports” e na sua propriedade *Glyph* localize um ícone para o botão, deixando a aparência do seu formulário semelhante à **Figura 3**.

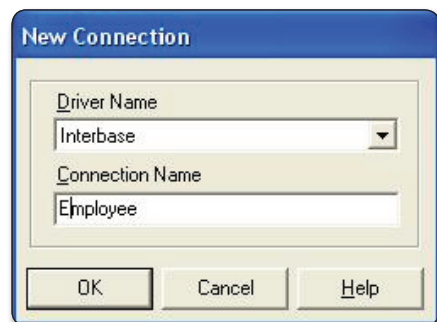


Figura 1. Criando uma nova conexão

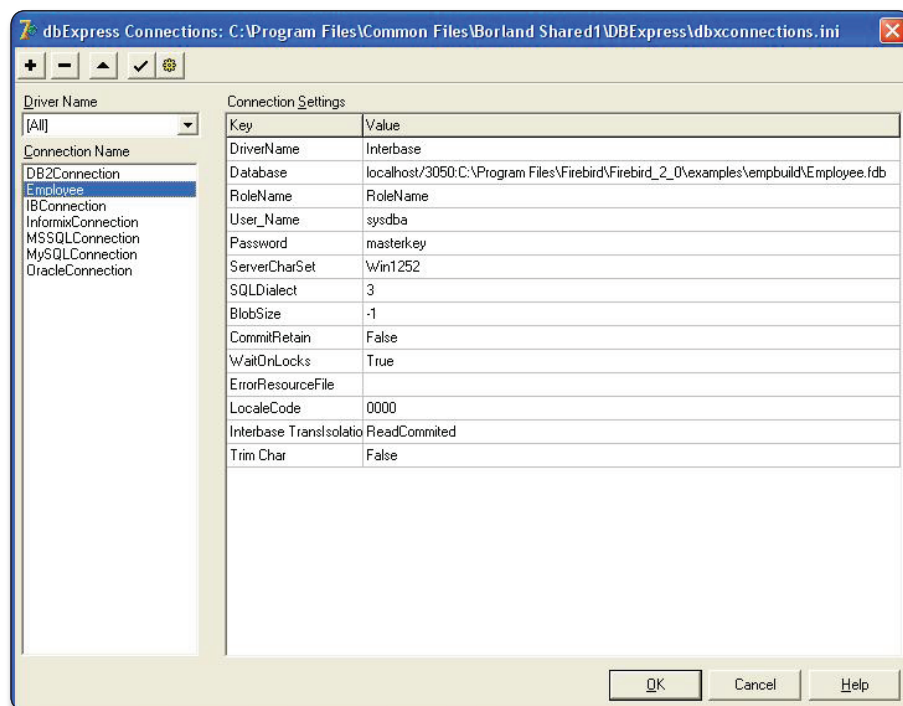


Figura 2. Configurando a conexão



No evento *OnShow* do formulário codificaremos de acordo com o código a seguir ativando então a busca dos registros a serem impressos:

```
DBGrid1.DataSource.DataSet.Open;
```

Compile e execute sua aplicação, de modo que poderá visualizar todos os registros atingidos pela busca. Volte à aplicação e adicione ao *Data Module* o componente *RvSystem*("rvSistema") da paleta *Rave* com a *SystemSetups>ssAllowSetup* configurada como *False* e *SystemPreview>FormState* igual a *wsMaximized*, que são responsáveis por ocultar o diálogo de *preview* antes do relatório e por abrir o relatório maximizado, respectivamente. Adicione ainda da paleta *Rave* um componente *RvProject*("rvProjeto") ligue a propriedade *Engine* ao *rvSistema*.

Em seguida adicione um *RvDataSetConnection*("rvdscEmployee") que fará a comunicação entre os dados contidos no *cdsEmployee* da aplicação e o projeto que criaremos no *Rave Reports*. Ligue sua propriedade *DataSet* ao *cdsEmployee*. Com um duplo clique sobre o componente *rvProjeto* abra o editor visual do *Rave Reports*, onde criaremos nossa estrutura de relatório.

## Criando o relatório no Rave Reports

Na estrutura fornecida pelo *Rave Reports*, visualize na paleta ao lado direito do programa uma árvore onde o primeiro item, *Rave Project*, dá origem a vários outros, e dentre eles a *Report Library*.

O item *Rave Library* contém todos nossos relatórios e por padrão o *Rave* já nos oferece e exibe uma página em branco, apresentada como *Report1*. Selecionando então a página de relatório, acesse, através do *Object Inspector*, sua propriedade *Name* e altere-a para "rptEmployee", nome este que será utilizado pelo Delphi para referenciar o relatório a ser exibido. Digite o valor "Relatório de Empregados" para a propriedade *FullName*. Acesse a propriedade *Parameters* e no *Memo* de configuração digite "Solicitante", que será um parâmetro que informaremos no Delphi e que será exibido no RAVE.

Ainda nesta mesma estrutura (**Figura 4**) ao abrir os nós da *árvore* que compõem o *rptEmployee*, selecione a página do relatório (*Page1*) e altere no *Object Inspector* a propriedade *PaperSize* para "A4 Sheet, 210- by 297-millimeters", ou qualquer outro formato de papel que desejar usar na impressão do relatório. Salve o projeto do relatório no mesmo diretório onde estão armazenados os fontes da aplicação Delphi, utilize o nome "prjRave.rav".

Após adaptar o formato do papel que o relatório será impresso, iremos definir o tamanho da área imprimível e para isso iremos adicionar da paleta *Report* do *Rave* o componente *Region* e definir seu tamanho de modo que ocupe todo o espaço da área em branco, respeitando somente um espaçamento de cerca de 1 cm em todas as margens para que em uma impressão não seja cortada parte do texto.

Antes de começarmos a definir a estrutura dos textos do relatório, iremos adicionar ao projeto a ligação com os componentes da aplicação do Delphi.

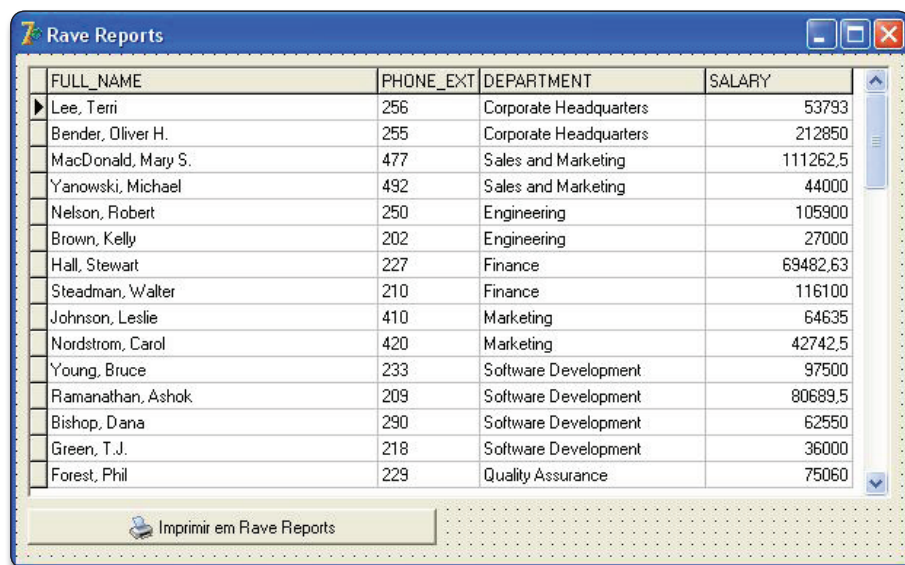
Acessando através do menu *File|New Data Object*, selecione a opção *Direct Data View*, utilizando o botão *Next* avance para a próxima etapa onde serão listados todos os componentes de ligação disponíveis na nossa aplicação Delphi.

Selecione o *rvdscEmployee* e finalize a operação. Temos agora dentro do grupo *Data View Dictionary* na *árvore* do projeto de relatório um novo componente (*DataView1*) que possui dentro dele todos os campos que adicionamos ao *ClientDataSet* no projeto.

Com o componente *DataView1* selecionado, acesse sua propriedade *Name* e altere-a para "dvEmployee". Agora sim, voltando a acessar a paleta *Report* adicione dentro do *Region* um *Band Component*, que ficará posicionado ao topo. Selecionando a *Band1* aumente sua altura e adicione da paleta *Standard* um *Text Component* e altere seu *Text* para "Relatório de Empregados", ajuste também o tamanho da fonte para que este seja o título do nosso relatório.

Agora da paleta *Report*, logo abaixo do título adicione um *Data Text component*, acesse a propriedade *DataField* e abra o *wizard* de configuração (**Figura 5**) e digite o seguinte texto no *Memo* do grupo *Data text*:

```
'Impresso em '+Report.DateShort+' as  
'+Report.TimeShort+' por '+Param.Solicitante
```



FULL_NAME	PHONE_EXT	DEPARTMENT	SALARY
Lee, Terri	256	Corporate Headquarters	53793
Bender, Oliver H.	255	Corporate Headquarters	212850
MacDonald, Mary S.	477	Sales and Marketing	111262,5
Yanowski, Michael	492	Sales and Marketing	44000
Nelson, Robert	250	Engineering	105900
Brown, Kelly	202	Engineering	27000
Hall, Stewart	227	Finance	69482,63
Steadman, Walter	210	Finance	116100
Johnson, Leslie	410	Marketing	64635
Nordstrom, Carol	420	Marketing	42742,5
Young, Bruce	233	Software Development	97500
Ramanathan, Ashok	209	Software Development	80689,5
Bishop, Dana	290	Software Development	62550
Green, T.J.	218	Software Development	36000
Forest, Phil	229	Quality Assurance	75060

Figura 3. Estrutura do formulário principal do sistema

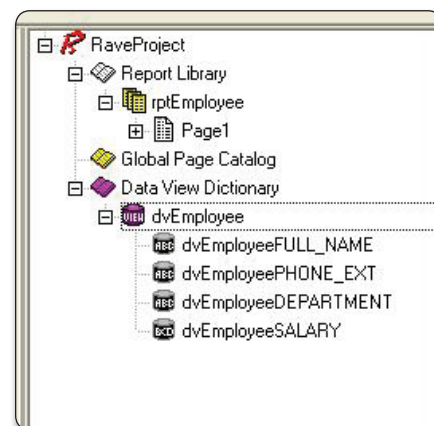


Figura 4. Estrutura do projeto no Rave Reports



Veja que no código adicionado estamos concatenando textos às funcionalidades já pré-definidas do projeto. Para incorporar os textos basta utilizar aspas simples e informar os valores entre as mesmas. Na concatenação das *strings* usa-se o sinal de adição "+". Neste caso estamos adicionando ao *DataField* a data e a hora que são capturadas do sistema operacional, além da utilização do parâmetro "Solicitante" que criamos a pouco e que será informado na codificação do Delphi com o nome da suposta pessoa "logada" no sistema.

Montando agora a estrutura onde iremos configurar os campos para a exibição dos registros do relatório.

Adicione da paleta *Reports* uma *DataBand component*, que será posicionada automaticamente também ao topo do *Region* e imediatamente abaixo do componente *Band1*. Para agilizar os serviços de configuração do relatório, iremos simplesmente arrastar os componentes que desejamos exibir para dentro da *DataBand*. Como já deixamos previamente configurados a descrição dos campos do *ClientDataSet* por meio da propriedade *DisplayLabel* de cada campo adicionado, basta que através do *dvEmployee* o campo seja selecionado e com a tecla "Alt" do teclado pressionada o mesmo seja arrastado para dentro da área da *DataBand*, repetindo assim o processo para todos os campos existentes.

Ao contrário da utilização da tecla "Alt" que adiciona somente os *Labels* de cada *Field*, ao selecionarmos novamente o campo do *dvEmployee* e manter pressionada desta vez a tecla "Ctrl" e arrastarmos o campo para a área da *DataBand*, será adicionado o *DataField* propriamente dito, onde em tempo de execução o mesmo irá assumir os valores a ele correspondentes.

Selecionando ainda a *DataBand component* configure sua propriedade *DataSource* relacionando-a ao *dvEmployee* que referencia o local de origem das informações que nela serão exibidas.

## Zebrando o relatório

Para deixar um relatório do Rave zebrado uma das formas mais simples é a utilização de um *Rectangle component* da paleta *Drawing*. Para utilizar o recurso, iremos adicionar ao *DataBand* um retângulo

de modo que cubra toda a área utilizada pelos campos onde o relatório será exibido, não deixando visíveis neste momento os campos anteriormente arrastados.

Ainda com o retângulo selecionado, acesse sua propriedade *BorderStyle* alterando-a para "psClear" deixando dessa forma o componente sem margens.

Com o botão direito do mouse, envie o componente para trás dos demais através da opção *Order>Send to back*, retornando assim os campos à visibilidade. Ainda com o retângulo selecionado perceba na parte superior do espaço do relatório a existência de duas guias *Page Designer* e *Event Editor*. Selecione a guia *Event Editor* e uma nova área de propriedades e edições do relatório será exibida.

De acordo com o componente selecionado na área de desenho do relatório, diferentes eventos estarão disponíveis na opção *Defined Events*. Iremos selecionar para o nosso exemplo a opção *OnBeforePrint*, que refere-se a um evento que é chamado antes da impressão do relatório. Neste espaço do relatório podemos entrar

com códigos específicos através da *Rave Language*, que são compilados e assumem diversas funções no relatório. Neste caso, iremos codificar conforme a seguir:

```
if Self.FillColor = clWhite then
  Self.FillColor := clSilver;
else
  Self.FillColor := clWhite;
end if;
```

**Nota:** Ao adicionar códigos na *Rave Language* tome um certo cuidado em relação aos códigos utilizados, pois em alguns casos o compilador do Rave diferencia letras maiúsculas e minúsculas podendo assim causar erros durante a exibição do relatório.

Finalizadas todas estas etapas de configuração do relatório, faça a compilação do mesmo (*File>Execute Report* ou tecla de atalho *F9*) para certificar que nenhum erro será detectado.

## Destacando valores no relatório

Uma ferramenta bastante útil e muito utilizada em relatórios com informações

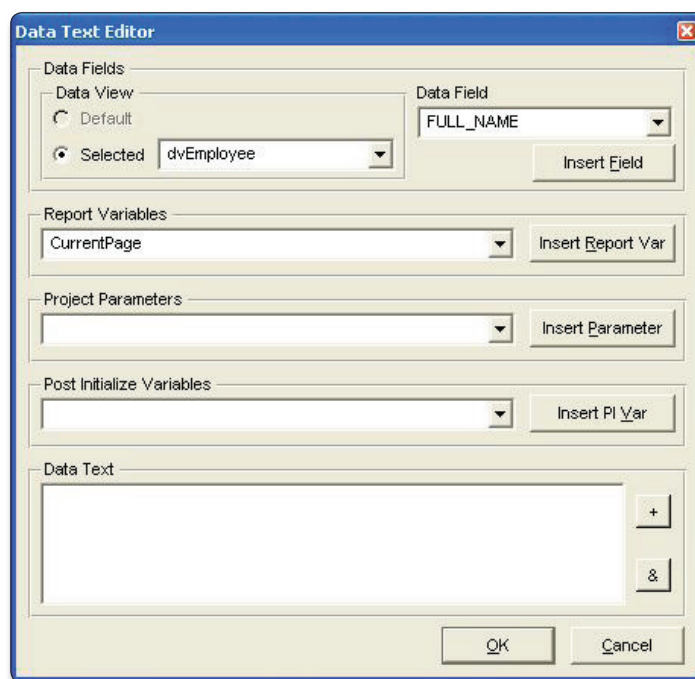
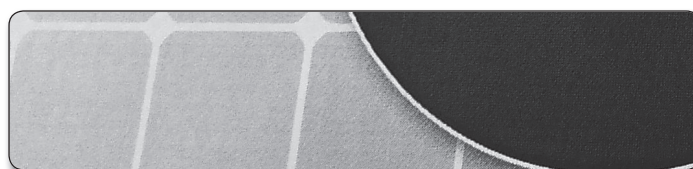


Figura 5. Wizard de configuração de um *DataField* no Rave Reports





de custos e valores é o destaque por cores ou fontes diferentes dos valores que atendem às condições pré-estabelecidas pelo programador ou vindas através de parâmetros da aplicação.

Iremos utilizar um bloco de código em *Rave Language* no evento *OnGetText* do *Data Text Field* responsável por exibir os registros do salário do empregado.

Selecione o *DataText* relacionado ao *DataField* *SALARY*, acesse o *Event Editor* do projeto e em *Available Events* selecione o evento *OnGetText*. Digite o código a seguir. Utilize o botão *Compile* para verificar a coerência do código. Salve seu projeto e execute novamente utilizando a tecla *F9* para verificar todos os valores de salário acima de \$ 100.000,00 serão apresentados em vermelho:

```
if dvEmployeeSALARY.AsFloat > 100000 then
  Self.Color:=clRed;
else
  Self.Color:=clBlack;
end if;
```

## Somando valores do relatório

Utilizado em praticamente todos os relatórios que trabalham com informações de valor numérico, a soma dos campos ao final do relatório é uma informação muito útil ao usuário em análise das informações. Para adicionar o campo de soma no relatório, adicione novamente da paleta *Report* um *Band component*, que será posicionado imediatamente abaixo ao *DataBand* onde os registros são exibidos. Adicione ao *Band2* um *Text component* da paleta *Standard*, alterando sua propriedade *Text* para "Total de Salários" e adicione também da paleta *Report* um *Calc Text component* que utiliza-

remos para exibir a soma dos salários.

Acessando a propriedade *Calc Type* do componente temos várias opções de operações disponíveis, utilizaremos a opção *ctSum*. Precisamos informar ainda qual o campo que vamos somar. Ligue a propriedade *DataView* do componente ao *dvEmployee* e selecione em *DataField* o campo *SALARY*. Ao executar o relatório novamente, localize na última página do relatório a soma dos valores de todos os salários. Ao final de todas as configurações realizadas no projeto do relatório, a estrutura deverá ser semelhante à representada na **Figura 6**.

## Executando o relatório

Retornando ao nosso projeto no Delphi, iremos adicionar os componentes para

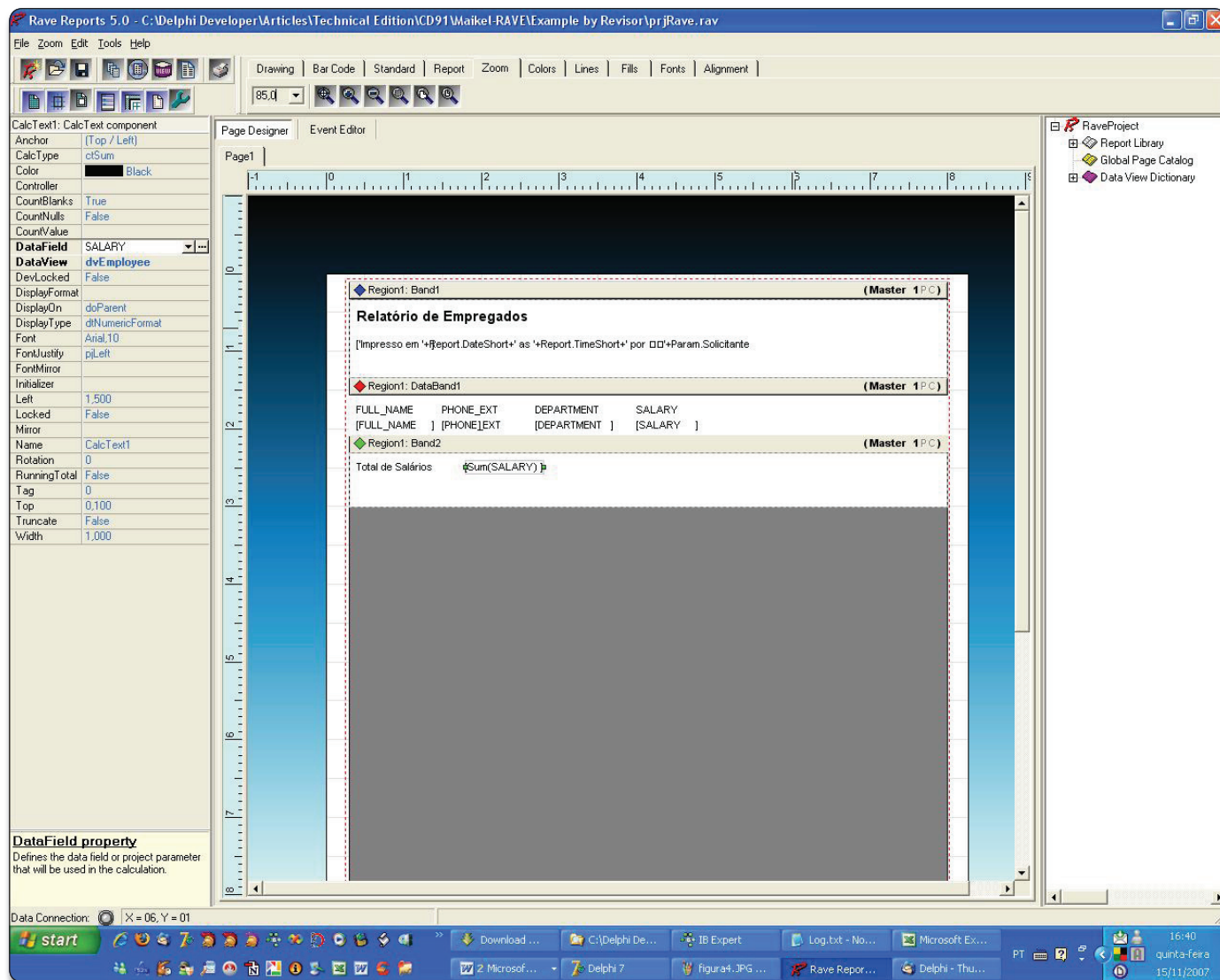


Figura 6. Estrutura do projeto de relatório no Rave



execução do relatório. Posicionando-se no *Data Module*, selecione o *rvProjeto* e na sua propriedade *ProjectFile* localize o arquivo “prjRave.rav” que será executado pela nossa aplicação. Voltando ao formulário principal do sistema, faremos a codificação para que o relatório seja exibido.

Adicione ao evento *OnClick* do *BitBtn1* as instruções do código a seguir que irá selecionar um relatório a ser impresso, passar o valor ao parâmetro que criamos no Rave e executar o relatório:

```
DM.rvProjeto.SelectReport('rptEmployee', True);
DM.rvProjeto.SetParam('Solicitante', 'Maikel');
DM.rvProjeto.Execute;
```

Na primeira linha do código estamos selecionando qual o relatório que queremos exibir na execução do botão. Na condição *SetParam* estamos atribuindo um valor ao parâmetro criado no relatório, informamos primeiro o nome do parâmetro e após a *string* com o valor que o mesmo deverá assumir e por último executamos e visualizamos. (Figura 7).

### Salvando o relatório em arquivo

Uma opção interessante e que atrai muito a atenção dos clientes na hora de demonstrar o sistema é a possibilidade de salvar um relatório no formato .pdf, que poderá ser enviado pra qualquer cliente impossibilitando alterações no formato do arquivo. Para utilizar este recurso, adicione ao *Data Module* um componente *RvRenderPDF* que será o responsável por exportar o relatório para PDF.

Retornando ao formulário principal do sistema, adicione um componente *BitBtn*, configure a propriedade *Caption* para “Salvar PDF”. Adicione também um componente *SaveDialog* da paleta *Dialogs* configurando sua propriedade *Filter* para “\*.pdf”. Adicione agora ao evento *OnClick* do *BitBtn2* o código da **Listagem 1**, que irá abrir um diálogo solicitando ao usuário o nome e a pasta que o arquivo salvou.

Compile sua aplicação e faça a exportação do seu relatório para PDF, e veja que a estrutura do relatório não foi alterada, mudando apenas o formato em que o mesmo foi salvo.

### Conclusão

Vimos neste artigo dicas práticas para criação e customização dos relatórios utilizando o editor de relatórios Rave Re-

ports. Criamos também um item prático da exportação de relatórios no formato PDF que poderá ser facilmente enviado por meio eletrônico, onde além desta possibilidade você poderá criar exportações customizadas dos relatórios ainda nos formatos *HTML*, *RTF* e *TXT*.

Além dos métodos que utilizamos no artigo você poderá criar várias outras formas de relatórios utilizando o Rave Reports, definir seu próprio *Preview* cus-

tomizado e ferramentas a serem exibidas no relatório. Crie processos dinâmicos de *master/detail* utilizando quebras de grupos ou condições de exibição. ●

**Nota:** Você poderá procurar por matérias/dicas no site do fabricante da ferramenta (em inglês) no endereço [www.nevrona.com](http://www.nevrona.com) na seção *Products|Rave Reports>Tips & Tricks*.

#### Listagem 1. Salvando relatório em PDF

```
procedure TfrmPrincipal.Button1Click(Sender: TObject);
begin
  if SaveDialog1.Execute then
  begin
    DM.rvSistema.DoNativeOutput := False;
    DM.rvSistema.DefaultDest := rdFile;
    DM.rvSistema.RenderObject := DM.RvRenderPDF1;
    DM.rvSistema.OutputFileName := SaveDialog1.FileName;
    DM.rvProjeto.SelectReport('rptEmployee', True);
    DM.rvProjeto.SetParam('Solicitante', 'Maikel');
    DM.rvProjeto.Execute;
  end;
end;
```

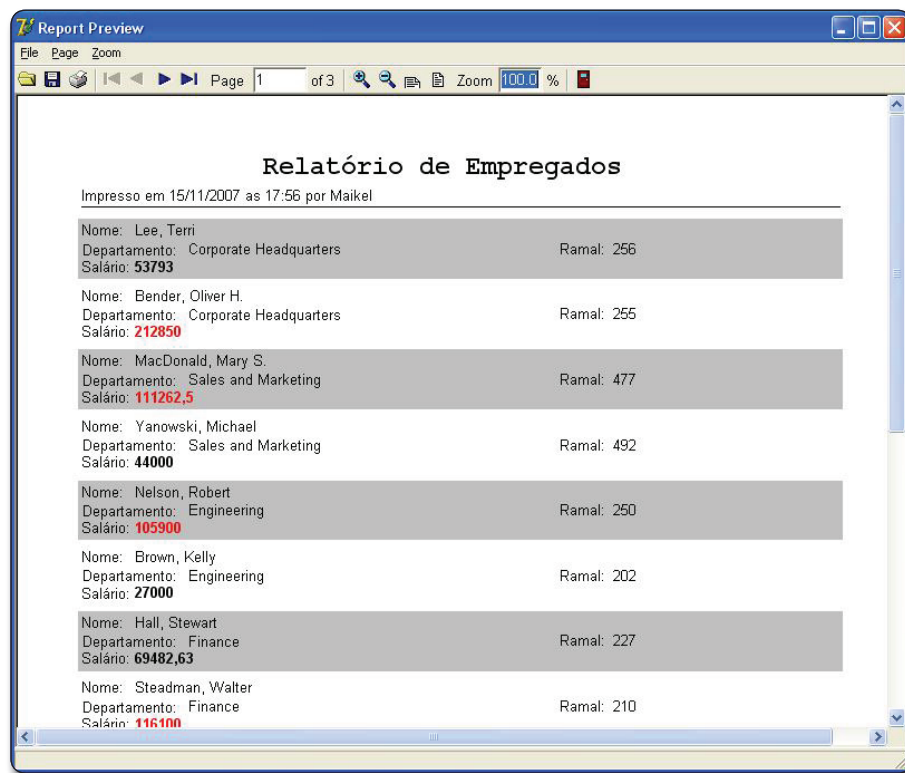


Figura 7. Exibição do relatório na tela

