

## Gráficos, tabelas dinâmicas e sub-relatórios

### Transcrição

Para esta aula queremos visualizar graficamente os gastos agrupando por titular. Na paleta de componentes existe o elemento `chart`, que arrastaremos para a seção `summary`.

O iReport já vem com vários charts disponíveis nesta janela. O chart que queremos é o *Stack Chart* (uma pilha de cores). Queremos visualizar os gastos agrupados por titular, portanto, cada pilha representará um titular de uma conta, e cada cor fará referência à soma de um tipo de gasto.

Na próxima página podemos selecionar qual fonte ou *dataSet* o chart utilizará. O iReport oferece o *dataSet* do relatório principal, o *main report dataset*. O problema é que a query do *dataSet* não agrupa os gastos por titular, sendo necessário alterá-la.

Observando a query, percebemos que qualquer alteração impactará no relatório existente. Não queremos alterar os dados, apenas gerar um gráfico adicional no fim do relatório.

### Um novo dataSet para o gráfico

Para resolver este problema, adicionaremos um segundo *dataSet* ao relatório com o nome *chartDataSet*. Nesse *dataSet* utilizaremos novamente a conexão `financas`. Definiremos a query através do *query designer*, clicando no botão *Design Query*. No designer, selecionamos as tabelas `categorias`, `contas` e `movimentacoes`. O iReport automaticamente realiza um join entre as tabelas. Os únicos campos que queremos na query são: titular da conta, nome da categoria e o valor da movimentação. Para isso, vamos desmarcar os outros. No caso do valor, somaremos os valores através de uma função.

Depois de ajustar as janelas, renomearemos os campos para simplificar o trabalho. Os campos ficam como `titular`, `categoria` e `soma`. No final agruparemos a query pelo titular da conta e categoria, mas como queremos apenas os gastos, adicionaremos uma condição na cláusula *where* para receber apenas as movimentações com o tipo `SAIDA`.

Após confirmarmos, podemos seguir para o próximo passo e adicionar todos os campos no *dataSet*. Não haverá nenhum agrupamento. Finalizando a definição do *dataSet*, aparece o novo elemento no *report inspector*.

### Criação do gráfico

Com o *dataSet* definido, voltaremos para criação do gráfico. Para simplificar, ajustaremos primeiro o tamanho da seção. Novamente arrastaremos o elemento `chart` para a seção *summary*, e na próxima tela escolheremos o *Stack Chart*. O *dataSet* será o *chartDataSet*.

O primeiro passo é definir a série, o valor de cada cor dentro de uma pilha. Cada cor é uma categoria de gastos. Consequentemente escolheremos o campo `categoria`. Na tela seguinte, a legenda estará definida abaixo do eixo X. Cada pilha é relacionada com um titular, e os valores das categorias são representados pela soma dos gastos. Podemos confirmar o gráfico e ajustá-lo na seção *summary*, ocupando todo o espaço.

Podemos visualizar o relatório clicando no botão *preview*. No fim do relatório aparecerá o gráfico que agrupa a soma dos gastos por categoria e titular. O gráfico visualiza bem os gastos no sumário do relatório, mas é difícil tirar um valor exato dessa forma.

### Resumindo e agrupando valores através de crosstable

Claro que podemos encontrar todos os valores dentro do relatórios, mas as informações estão espalhadas pelas páginas. Pode ser muito útil apresentar um sumário no final do relatório, depois do gráfico, que mostra os valores mais importantes dentro de uma tabela. Para isto existe o componente *crosstable*(ou *pivotable*). O objetivo é mostrar os gastos por mês de cada titular. Arrastaremos este componente para a seção sumário, aumentando um pouco o seu espaçamento.

Nesse caso, usaremos o *dataSet* do relatório principal. Na próxima tela definiremos o que queremos mostrar em cada linha. Neste caso, o nome do titular. Depois, definiremos o que usar no cabeçalho da coluna, que será o campo *data* (apenas o mês). O próximo passo é decidir qual valor aparecerá em cada célula.

Queremos a soma dos gastos por mês deste titular, então, escolheremos o campo valor com a função *sum*.

Na última tela podemos mudar o layout padrão da tabela. Usaremos a cor bege. Depois de confirmado, o iReport criará automaticamente uma nova aba para mostrar o design da tabela. Nesta tabela podemos ver os dados dos cabeçalhos e células. Além disso, também foram criados campos para somar cada linha e coluna.

No relatório principal, precisamos ajustar o layout que está em conflito com o gráfico, diminuindo o espaço da tabela. Depois disso podemos visualizar o relatório. Indo para a última página do relatório, aparece a tabela, com os titulares em cada linha e a soma do mês na coluna. Porém, a tabela foi criada baseada em apenas um mês porque nossa query principal a utilizou num período muito pequeno.

A tarefa agora é aumentar este período, mas antes vamos inserir quebras de páginas entre o gráfico e a tabela para melhorar o design. Já existe um componente pronto na paleta. Para aumentar o período é preciso alterar o parâmetro *DATA\_INI* . Podemos acessá-lo pelo *report inspector*, selecionado a opção para editar a expressão, mostrando os meses junho, julho e agosto. Novamente, usando o *preview* e indo para a última página, percebemos que a tabela foi criada com os três meses.

## Usando sub-relatórios

O gráfico e a *crosstable* são opções para visualizar as informações, diferente do relatório principal. Existem casos aos quais queremos integrar dados complexos, nesse caso, é preciso um novo relatório em anexo ao relatório principal. Com o iReport, podemos criar e integrar um novo relatório através do elemento *subreport*, que se encontra na paleta. Arrastando este elemento na seção *title*, um wizard de criação de relatórios aparecerá automaticamente.

Na primeira tela, definiremos se vamos criar um novo relatório ou integrar um já existente. Neste caso criaremos um novo, mas precisamos também escolher um layout para ele. Usaremos o template *Blank Letter* com a mesma conexão *financas* .

A query será bem simples para ilustrar a funcionalidade de um sub-relatório. Pesquisaremos todas as contas executando `select * from contas` . Na próxima tela, selecionaremos todos os campos da query. Não haverá agrupamento e manteremos o nome padrão fornecido pelo iReport ao salvar o *subreport*. Na última tela, confirmamos o uso da conexão do relatório mestre.

O iReport abre uma nova aba para o design do relatório. O sub-relatório é um relatório comum, então existem as mesmas seções e variáveis disponíveis. Também encontraremos parâmetros, campos e variáveis no *report inspector*.

Para definir os dados do relatório, colocaremos todos os campos na seção e também apagaremos as seções sem uso, como por exemplo: *column footer*, *page footer*, *summary*, *page header* e *title*. Por último, diminuiremos o tamanho das seções *column header* e *detail* para otimizar o espaço. Falta visualizar e testar o relatório, clicando no botão *preview*.

## Integração do sub-relatório e relatório mestre

Vamos voltar ao relatório mestre. O nosso sub-relatório deve ficar no início, na seção *title*. É preciso ajustá-lo para o sub-relatório ter espaço suficiente.

Como queremos na primeira página apenas os dados das contas e titulares, adicionaremos uma quebra de pagina após do sub-relatório. Visualizando o relatório, aparecem os dados da conta e titulares na primeira página. Nas próximas páginas, os dados principais do relatório mestre e no final, o gráfico seguido pelo *crosstable*.