

Formatação e ajustes de visualização do relatório

Transcrição

Olá! Vamos continuar com o nosso relatório *Gastos por conta*. Ele é funcional, mas ao visualizá-lo, é possível perceber que está mal formatado. Vamos melhorar o seu design e aumentar o zoom nos ajudará. Atacaremos inicialmente a primeira seção, o *title*. Podemos selecionar o elemento diretamente no *Designer* ou navegar através do *Report inspector*.

A primeira coisa que faremos é selecionar um retângulo que servirá como background e escolher uma cor para ele nas propriedades do componente. Ele não terá um traço e será adaptado levando em consideração o tamanho do seu *parent*, a seção no qual ele está incluído. Para tal, usaremos os *Formatting Tools - adapt to parent*. Agora só falta colocar o componente para trás com *send to back* e centralizar o elemento dentro da seção *title*.

Nosso próximo passo é deixar todos os cabeçalhos com letra maiúscula. Depois disso vamos alinhar os nomes do cabeçalho, bastando selecionar todos ao mesmo tempo e aplicar alinhamento pelo centro. Já podemos visualizar o resultado pelo iReport.

Formatar e concatenar campos de texto

Ainda falta trabalhar a parte do titular do relatório, pois queremos ele também em letra maiúscula. O problema é que o titular é um campo que veio do banco de dados, fora do padrão desejado. O campo dentro do relatório é uma expressão apresentada através do `${}`. Podemos manipular essa expressão através do *Expression Editor*. Repare que o tipo do campo é `java.lang.String`, sendo assim, utilizaremos os próprios recursos desta classe para colocarmos a informação em letra maiúscula. No editor da expressão, na lista de métodos, aparece o método `toUpperCase()`. Além disso, queremos concatenar a informação *CONTA DE* com o titular usando o operador `+` da linguagem Java.

O tamanho do campo não é suficiente, sendo necessário adaptá-lo ao seu *parent*, como fizemos com nosso background. Na visualização, vemos a nova informação e tudo centralizado, mas ainda falta o background na cor azul. Voltando ao *Designer*, ao selecionar o componente de texto com o titular, podemos alterar a cor de fundo nas propriedades dos componentes.

Formatando a data e os valores numéricos

A próxima etapa é formatar os dados das colunas `valor` e `data`. Com o campo `valor` selecionado, na opção *field pattern*, escolhemos a opção *currency*, para apresentar o valor monetário. A mesma coisa será feita com o campo `data`, usando a opção *field pattern*, mas escolheremos a opção *date*, mostrando apenas a data(sem a hora).

Resultados condicionais

As modificações são aplicadas, mas ao visualizarmos mais uma vez o relatório, vemos que em alguns casos o campo `descricao` não se posiciona conforme o esperado quando o mesmo não existe. É apresentado o valor `null`, que não faz sentido para o leitor do relatório.

Para resolvemos este problema, basta selecionar o campo `descricao` e marcar a opção *Blank When Null*. Com a opção marcada e visualizando, vemos que o resultado ainda não é o esperado. Mais uma vez sobre o campo, marcamos a opção *Stretch with overflow*. Esta opção organizará a informação mesmo quando não há espaço disponível, sempre mostrando todas as informações do campo.

Diminuir o campo de texto

Há situações onde é necessário diminuir o tamanho da coluna, por exemplo, mostrar apenas a letra *S*, para indicar que a movimentação é de *SAÍDA*. Com o campo selecionado e editando a expressão, podemos recorrer ao método `substring()` e selecionar apenas a primeira letra. Por fim, centralizaremos alguns campos. Agora é só visualizar. Aparece apenas a letra *S* na coluna *Tipo*.

Mais formatações

Falta ainda lidar com os campos ou variáveis que agrupam o valor total do mês e da conta. Primeiro, alinharemos o resultado e cada grupo à direita e definiremos o *pattern currency* para eles. Depois do *pattern* definido, criaremos outro retângulo na cor azul que ocupará todo o espaço de seu *parent*, produzindo um efeito que já fizemos.

No lugar de criarmos mais um retângulo, o copiaremos para o próximo grupo. Por fim, jogaremos os retângulos para trás com *send to back*. Visualizando outra vez, vemos o resultado esperado. Os agrupamentos aparecem na cor azul para se destacarem no relatório.

Mas ainda há um problema: o texto do número da página está em inglês. Editando o elemento e selecionando a expressão, podemos modificar o texto original *page* para *página*. Novamente visualizando para verificar o resultado. Em cada página aparece o rodapé em português. Por último, trocaremos a cor de todos os retângulos de azul para cinza, basta selecionar todos os retângulos com *control + shift* e mudar o *background color* para cinza. Agora, em nosso relatório, as seções serão apresentadas em cinza.

Passando parâmetros para o relatório

Temos um total de 33 páginas, que pode ser demais para quem deseja retirar alguma informação do relatório. Para o usuário final seria mais fácil se ele pudesse filtrar os gastos. Por exemplo, pela data, selecionando um período. A ideia é renderizar apenas os gastos entre duas datas definidas.

Para esta situação, o iReport oferece parâmetros. Os parâmetros são valores passados da aplicação para o JasperReport, o responsável pela renderização do relatório. Já existem parâmetros pré-definidos, por exemplo, de localização visível no *Report Inspector*. Podemos adicionar nossos parâmetros através do menu, e uma vez feito isso, podemos alterar as suas propriedades pelo editor. O nome do nosso parâmetro será `DATA_INI` e seu tipo `java.util.Date`. Criaremos outro parâmetro que representa a data final, novamente no *Report Inspector*. O nome dele será `DATA_FIM`, também do tipo `Date`.

Filtrando o relatório pelos parâmetros

Como resultado, temos os dois parâmetros disponíveis para usar dentro do *Report Inspector*. Podemos agora utilizá-los arrastando para o nosso relatório, mas não faremos isto porque modificaremos nossa query com base nestes parâmetros.

Ao abrir o editor da query, alteraremos a cláusula `where` utilizando a data e filtrando-a pelos dois parâmetros que podem ser arrastados para a query. A data da movimentação deve estar entre `DATA_INI` e `DATA_FIM`. Vamos confirmar o diálogo e pré-visualizar o relatório. Quando o relatório for renderizado, o iReport mostrará automaticamente um prompt solicitando um valor, pois as datas são necessárias para executar a query. Vamos filtrar as movimentações pelo mês de agosto. O relatório mostrará apenas as movimentações desse mês.

Para não termos que definir toda vez a data, podemos definir um valor padrão. Nos parâmetros criados, desmarcamos a opção *Use as a prompt* e editamos *Default Value Expression*. Utilizaremos a classe `SimpleDateFormat`, que receberá um pattern com o formato da data desejada e que realizará um parse da `01/08/2012`.

Veja o código:

```
new java.text.SimpleDateFormat("dd/MM/yyyy").parse("01/08/2012");
```

Faremos a mesma coisa para `DATA_FIM`, desabilitando o prompt e alterando o valor que sofrerá o parse, neste caso, final de agosto. Renderizando novamente, percebemos que o prompt não aparece mais e mesmo assim tivemos nosso relatório filtrado.