

 01

Instalação e o primeiro relatório

Transcrição

Você pode baixar a **versão utilizada do iReport** nesse [link](http://s3.amazonaws.com/caelum-online-public/FJ-24/iReport-4.7.0.zip) (<http://s3.amazonaws.com/caelum-online-public/FJ-24/iReport-4.7.0.zip>). Também vamos usar um banco de dados no treinamento. Para facilitar a importação dos dados preparamos um arquivo SQL disponível nesse link: [financas.sql](http://s3.amazonaws.com/caelum-online-public/FJ-24/financas.sql) (<http://s3.amazonaws.com/caelum-online-public/FJ-24/financas.sql>).

Case queira usar a versão mais atual do iReport segue o link da [Jaspersoft](http://community.jaspersoft.com/download) (<http://community.jaspersoft.com/download>).

Relatórios em HTML e seus problemas

Qualquer aplicação de negócio tem que se preocupar com a apresentação de dados. O usuário final quer gerar relatórios com dados agrupados e bem formatados e tirar conclusões sobre o que é apresentado.

A forma mais comum é o desenvolvedor gerar uma página HTML, por exemplo, com uma tabela, formatando-a através de CSS. O problema é que isso nem sempre tem a qualidade e precisão desejada após a impressão. Como, por exemplo, paginar este relatório?

Outra questão: o que fazer se a aplicação não é uma aplicação web? E se ainda desejarmos numa aplicação web outras saídas, não só HTML com CSS, como também PDF? Muitas vezes misturamos dados do relatório com sua visualização, sendo necessário rescrever parte do código para outras saídas.

Isso também significa que estamos dependentes do desenvolvedor para criar um relatório. Queremos uma abordagem mais simples, uma abordagem que permita criar relatórios sofisticados sem termos que conhecer detalhes técnicos a fundo.

Report engine: JasperReports e iReport

Para resolver essas questões foram criadas ferramentas específicas para a geração de relatórios, que também são chamadas de *report engines*: são eles que assumem a responsabilidade pela formatação e geração.

Uma ferramenta famosa do mundo Java é o iReport. Ele nos fornece uma interface gráfica bem elaborada para definirmos o relatório e para exportá-lo em vários formatos como PDF, PPT, OpenDocument, etc.

O iReport é apenas o *report designer*, a ferramenta que ajuda na construção de relatórios. A definição é salva dentro de um XML com terminação `.jrxml`. Este XML é independente da saída concreta.

O responsável pela leitura do XML é o JasperReports - o *report engine*, que preenche o relatório com os dados exportando-o para a saída desejada.

Instalação do iReport

Vamos começar a utilizar o *report designer* iReport com JasperReports, acessando a página community.jaspersoft.com (<https://community.jaspersoft.com/project/ireport-designer>).

Existem outros projetos relacionados ao JasperReports, mas o que nos interessa é o report designer. Podemos baixar o programa clicando no link. Há um download para cada sistema operacional específico (Windows, Mac e Linux) e também um ZIP genérico para qualquer plataforma, inclusive há um plugin para o Netbeans.

Utilizaremos o ZIP genérico. Ao baixar o ZIP, será pedido um login e senha, basta clicar no link logo abaixo que pulará esta etapa. Assim, chegamos na página de download.

Agora com o ZIP baixado, vamos extraí-lo. Temos a distribuição genérica do iReport com os arquivos, plugin para Netbeans e uma pasta `bin` com os arquivos de inicialização do iReport. Cada um dos arquivos destina-se a uma plataforma diferente (Windows, Mac e Linux).

Definição da conexão e template

Temos a página inicial da ferramenta na qual podemos ver um assistente para criarmos nosso primeiro relatório. Primeiro definiremos de onde virão nossos dados. A nossa aplicação com iReport utilizará o MySQL. Nele teremos um banco já configurado chamado `finanças`. Neste banco teremos uma tabela chamada `movimentações` e selecionaremos todos os seus dados.

Dentro do iReport definiremos uma conexão com o nosso banco de dados (`jdbc connection`) que se chamará `finanças`. O driver é mysql e a nossa URL será `finanças`. O login será root, sem senha. Após configurar, salvaremos e testaremos a conexão.

O próximo passo é selecionar através do assistente o layout, o template do nosso relatório. Existem templates pré-definidos. Utilizaremos o template *simple blue*. Podemos fazer um zoom para vermos mais detalhes sobre este template. Após inicializar, chamaremos o relatório de `finanças`. No menu `file`, na parte `admin`, criaremos um arquivo chamado `financas.jrxml`. Utilizaremos nosso `datasource` recém criado. Nossa query será um `select * from movimentacoes`. Ele já fez o `select`, analisou nossa query e mostrou todos esses fields. Adicionaremos o `id`, a `data`, a descrição e o tipo da movimentação, como também o valor. O `id` da categoria e da conta não importam neste relatório. Na próxima tela não agruparemos dados.

Organização do iReport

Estamos vendo a tela do nosso relatório e no centro o *Designer*.

O iReport criou colunas para nossos campos e logo abaixo de cada um deles uma espécie de *expression language*, para indicar que teremos os valores.

No lado esquerdo temos o *Report inspector*, que mostra a estrutura do nosso relatório e também as variáveis mais importantes como os parâmetros, os fields e as variáveis. Em fields aparecem os campos que selecionamos no momento de criação do nosso relatório.

No lado direito temos as propriedades que mudam sempre quando selecionamos um elemento dentro do relatório.

Visualizando o primeiro relatório

O primeiro passo é alterar o relatório, mas já podemos pré-visualizá-lo clicando em *Preview*. O que o iReport faz é usar nosso relatório, acessar o banco de dados, gerando um PDF com essas duas informações.

Várias páginas foram geradas com os dados das movimentações, mas como podemos perceber, a data mostra um horário que não queremos apresentar. Voltando ao *Designer* e selecionando a data, podemos, no lado direito, adicionar um pattern (`dd/MM/yyyy`) manualmente ou utilizar um assistente de data. Vamos formatar a data mostrando-a sem horário.

Adicionaremos o título, movimentações, mudaremos o tamanho, apagaremos a descrição e também colocaremos em maiúsculo o cabeçalho. Fazendo o preview novamente, podemos ver o nosso atalho e também a data bem formatada.

