

Conteúdo

INTRODUÇÃO	7
OS SOFTWARES REQUERIDOS.....	7
HARDWARE REQUERIDO	7
CAPÍTULO 1	9
CONHECENDO O IREPORT	9
<i>Antes de utilizar o iReport</i>	9
<i>Obtendo e instalando o Java SE</i>	9
<i>Em Windows versão 9x</i>	11
<i>Obtendo e instalando o iReport</i>	12
<i>Algumas características do iReport</i>	12
<i>Instalando o iReport</i>	13
CAPÍTULO 2	19
INICIANDO NO IREPORT	19
ABRINDO O IREPORT PELA PRIMEIRA VEZ	19
<i>Documentos Recentes não são exibidos</i>	20
CRIANDO SEU PRIMEIRO RELATÓRIO COM IREPORT	20
<i>O banco de dados usado</i>	20
<i>Antes de criar o primeiro relatório</i>	21
<i>O primeiro relatório</i>	21
<i>Visualizando o primeiro relatório</i>	28
CONFIGURANDO O IREPORT.....	29
VISUALIZANDO O RELATÓRIO COMPILADO EM OUTRO APLICATIVO	33
CAPÍTULO 3	35
ESTRUTURA DE UM RELATÓRIO E SEU DESENVOLVIMENTO	35
A ESTRUTURA DO RELATÓRIO	35
<i>Title</i>	36
<i>Page Header</i>	37
<i>Column Header</i>	37
<i>Detail</i>	38
<i>Column Footer</i>	38
<i>Page Footer</i>	39
<i>Last Page Footer</i>	39
<i>Summary</i>	40

<i>Background</i>	40
<i>noData</i>	41
COMPREENDENDO AS PROPRIEDADES DE UM RELATÓRIO	41
<i>Iniciando com um relatório em branco</i>	41
CAPÍTULO 4	51
CRIANDO UM RELATÓRIO SEM O USO DE ASSISTENTES	51
<i>Iniciando um relatório manualmente</i>	51
<i>Exibindo o Grid para desenho de relatórios</i>	52
<i>As barras de ferramentas do iReport</i>	52
<i>Adicionando uma conexão a um relatório em branco</i>	53
<i>Criando a query do relatório</i>	56
<i>Os elementos da query registrados no JasperReports</i>	60
<i>Desenhando o relatório</i>	61
<i>Os símbolos para exibir o formato de números</i>	72
<i>Conhecendo melhor o Text Field</i>	73
<i>Exibindo imagens em seu relatório</i>	73
<i>Um pouco mais sobre o elemento image</i>	75
CAPÍTULO 5	78
GERANDO RELATÓRIOS COM GRUPOS, PARÂMETROS E EXECUTANDO POR UMA APLICAÇÃO DESKTOP	78
<i>O relatório que será desenvolvido</i>	78
<i>Gerando o relatório</i>	78
<i>Criando a query do relatório</i>	79
<i>Criando o grupo</i>	81
<i>Adicionando os elementos na band PedidoGroupHeader</i>	84
<i>Criando variáveis com cálculos no iReport</i>	87
<i>Adicionando os elementos na band detail</i>	90
<i>Totalizando por pedido</i>	92
<i>Finalizando a aparência do relatório</i>	93
<i>Adicionando parâmetros ao relatório</i>	93
<i>Chamando o relatório por uma aplicação desktop</i>	97
CAPÍTULO 6	101
DESENVOLVENDO RELATÓRIOS COM NETBEANS IDE	101
DESENVOLVENDO RELATÓRIOS COM NETBEANS	101
OBTENDO E INSTALANDO O IREPORT PARA NETBEANS	101
CRIANDO UM RELATÓRIO COM O PLUGIN	104
A ESTRUTURA E PROPRIEDADES DE UM RELATÓRIO JASPERREPORTS	104

EXIBINDO O GRID	109
ADICIONANDO UMA CONEXÃO A UM RELATÓRIO EM BRANCO	109
CRIANDO A QUERY DO RELATÓRIO	112
DESENHANDO O RELATÓRIO	114
ADICIONANDO OS TEXTOS ESTÁTICOS.....	115
APÊNDICE A.....	119
O MYSQL	119
O QUE É UM BANCO DE DADOS RELACIONAL?	119
INSTALANDO O BANCO DE DADOS	119
INSTALANDO NO WINDOWS	119
INSTALANDO O MYSQL NO LINUX.....	120
ACESSANDO O BANCO DE DADOS MYSQL.....	120
NO WINDOWS.....	120
NO LINUX.....	121
OS COMANDOS CREATE E DROP	121
O COMANDO CREATE.....	121
O COMANDO USE.....	121
O COMANDO DROP.....	122
CRIANDO TABELAS	122
O COMANDO SHOW.....	122
O COMANDO DESCRIBE	123
ALTERANDO TABELAS EXISTENTES.....	123
ALTERANDO O NOME DA COLUNA.....	123
ALTERANDO O TIPO DE COLUNA	123
RENOMEANDO UMA TABELA.....	123
EXCLUINDO / ADICIONANDO COLUNAS E TABELAS.....	123
ELIMINANDO TABELAS E COLUNAS	124
ADICIONANDO COLUNAS	124
ADICIONANDO COLUNAS APÓS UMA OUTRA DETERMINADA.....	124
UTILIZANDO ÍNDICES	124
DECIDINDO QUAIS COLUNAS INCLUIR NO ÍNDICE	124
CRIANDO UM ÍNDICE.....	125
EXCLUINDO ÍNDICES	125
TIPOS DE TABELAS	125
O TIPO MYISAM.....	126
O TIPO INNODB	126

ALTERANDO O TIPO DE UMA TABELA	126
TIPO DE DADOS	127
TIPOS NUMÉRICOS.....	127
MODIFICADORES AUTO_INCREMENT, UNSIGNED E ZEROFILL.....	128
AUTO_INCREMENT	128
UNSIGNED	128
ZEROFILL	128
TIPOS DE CARACTERE OU DE DADOS DE STRING	129
CHAR E VARCHAR	129
TEXT E BLOB	130
TIPOS VARIADOS	130
TIPO ENUM	130
TIPO SET	130
TIPOS DE DATA E HORA (DATE/TIME).....	130
MODIFICADORES ADICIONAIS DE COLUNA	131
SINTAXE BÁSICA DA SQL	131
COMANDO INSERT.....	131
COMANDO SELECT	132
A CLÁUSULA WHERE.....	132
FUNÇÕES QUE TRABALHAM COM A INSTRUÇÃO SELECT.....	133
LIMIT	133
COUNT()	134
ORDER BY	134
ORDER BY ... DESC.....	134
LIKE	134
COMANDO UPDATE.....	135
COMANDO DELETE	135
TRABALHANDO COM JUNÇÕES.....	135
CRIANDO UMA JUNÇÃO COM INNER JOIN	135
CHAVES VARIADAS DO MYSQL	136
O QUE É UMA CHAVE?.....	136
COMO AS CHAVES FUNCIONAM	136
BENEFÍCIOS DE USAR UMA CHAVE	136
SUPORTE DE CHAVES DO MYSQL.....	136
CHAVES PRIMÁRIAS (PRIMARY KEY).....	137
CHAVES ESTRANGEIRAS (FOREIGN KEY)	137

EXCLUINDO UMA CHAVE ESTRANGEIRA.....	137
ADMINISTRANDO O MYSQL.....	137
ENTENDENDO O SISTEMA DE PRIVILÉGIOS DO MYSQL.....	137
CONFIGURANDO USUÁRIOS.....	138
CONFIRMANDO O NOVO USUÁRIO	138
REVOGANDO PRIVILÉGIOS	139
OBTENDO INFORMAÇÕES COM SHOW	139
APÊNDICE B.....	140
CONTEXTO DE DESENVOLVIMENTO WEB EM JAVA	140
<i>No diretório WEB-INF.....</i>	<i>140</i>
<i>Instalando uma aplicação Web em outro local.....</i>	<i>141</i>
APÊNDICE C.....	142
TRABALHANDO COM DATAS.....	142
APÊNDICE D	144
O TOMCAT.....	144
SERVLETS E JSP.....	144
INSTALANDO O TOMCAT.....	144
BAIXANDO E INSTALANDO O APACHE TOMCAT	144
NO WINDOWS.....	145
ADICIONANDO A VARIÁVEL CATALINA_HOME.....	145
NO LINUX.....	145
INICIANDO O TOMCAT.....	145
NO WINDOWS.....	146
NO LINUX.....	146
PARANDO O TOMCAT.....	146
NO WINDOWS.....	146
NO LINUX.....	146
O TOMCAT EM OPERAÇÃO	146
O QUE VOCÊ ENCONTRA NO LIVRO DOMINANDO RELATÓRIOS JASPERREPORTS COM IREPORT.....	147
SUB-RELATÓRIOS.....	147
<i>O relatório mestre-detalle que será desenvolvido.....</i>	<i>147</i>
RELATÓRIOS COM HIBERNATE.....	149
SUB-RELATÓRIO COM HIBERNATE	150
RELATÓRIOS USANDO EJB QL	150
<i>Desenvolvendo o relatório</i>	<i>151</i>

GERANDO RELATÓRIOS COM OUTRAS FONTES DE DADOS	152
SIMPLIFICANDO RELATÓRIOS COM GROOVY	155
CRIANDO GRÁFICOS E SUBDATASET EM SEUS RELATÓRIOS	156
<i>Criando um gráfico simples</i>	156
<i>Subdatasets</i>	157
<i>Utilizando Gráficos com Subdatasets em um relatório com Hibernate</i>	157
INTERNACIONALIZANDO RELATÓRIOS E IMAGENS DE FUNDO	157
CROSSTABS.....	159
<i>Criando um Crosstab com Subdataset</i>	160
OUTRAS CARACTERÍSTICAS DA GERAÇÃO DE RELATÓRIOS	161
<i>Tratando relatórios grandes</i>	161
<i>Exibindo somente um grupo por página</i>	161
<i>Reiniciando o número de páginas</i>	161
<i>Alinhamento de objetos no design do relatório</i>	161
<i>Formatação entre textos no seu relatório</i>	161
APÊNDICE A - EXPORTANDO PARA OUTROS FORMATOS.....	162

Introdução

Desenvolver aplicações com o uso de relatórios é uma rotina comum para um desenvolvedor. Ao longo de anos, diversos softwares de relatórios surgiram. O sucesso de cada um dependia da integração com uma determinada plataforma (leia-se linguagem) que estivesse no auge no momento.

Em Java, a situação é um pouco diferente. Com uma linguagem onde os desenvolvedores se acostumaram com a cultura Open Source (embora Java não fosse até bem pouco tempo), poucas ferramentas de relatórios se destacavam neste cenário. Bem poucas também detinham as características profissionais dos melhores programas para a geração de relatórios. O tempo passou e este cenário mudou. Em 2001, nasceu o JasperReports, criado por Teodor Danciu, quando teve a tarefa de avaliar ferramentas de relatórios para um projeto que estava trabalhando. As soluções existentes eram muito caras para o orçamento do projeto e, em uma situação comum do mundo do desenvolvimento, ele decidiu escrever sua própria ferramenta de relatórios, que ao liberá-la para a comunidade, ficou imensamente popular em pouco tempo.

JasperReports é uma biblioteca escrita em Java, de código fonte open source, projetada para ajudar o desenvolvedor com a tarefa de criar relatórios para aplicações, tanto Desktop como Web, fornecendo uma API que facilita sua geração. Embora seja simples, ainda exige que o desenvolvedor conheça seu formato XML utilizado para criar os relatórios, o que torna dispendioso o tempo de um iniciante.

Mas este cenário começou a mudar em 09 de outubro de 2002, quando Giulio Toffoli lançou, de forma independente, uma ferramenta para gerar relatórios visuais, chamando-a de iReport. Sua característica era de desenvolver relatórios gerando o formato XML no padrão JasperReports.

Em 2005, com a popularidade do iReport, a JasperSoft (mantenedora do JasperReports) tornou esta ferramenta oficial na construção de relatórios para o JasperReports.

Com cinco anos de existência, o iReport juntamente com o JasperReports chega a sua segunda versão, demonstrando não somente amadurecimento, mas reconhecimento e respeito de desenvolvedores que trabalham com a linguagem Java como uma ferramenta profissional para o desenvolvimento de relatórios.

Este e-book é um guia que o ensinará a utilizar o básico do iReport para construir relatórios JasperReports.

Os Softwares requeridos

Os aplicativos criados nesse livro não exigem software proprietário. Portanto ao longo dos capítulos você não só aprenderá a usá-los, como também onde encontrá-los na Internet, caso precise de uma atualização.

Esse livro não está focado especialmente em um sistema operacional, portanto a sua escolha é livre neste sentido.

Hardware requerido

Uma boa configuração de hardware se faz necessário para trabalhar com aplicações escritas em Java. Um computador para rodar bem o iReport deve ter as seguintes configurações para uma confortável utilização, segundo o autor:

Processador: Pentium 3 (similar) ou superior

Memória: 256 MB de RAM mínimo (recomendo 512 GB de RAM)

HD: 20MB de espaço livre no disco

Monitor: 17 polegadas

Os demais programas apresentados ao longo do livro, como o Eclipse e o NetBeans, seguem um padrão estabelecido por seus mantenedores.

Capítulo 1

“Começar já é metade de toda a ação.”

provérbio grego

Conhecendo o iReport

O iReport é um programa Open Source, capaz de criar visualmente os mais complexos relatórios para aplicações Java no formato da biblioteca JasperReports. Como é escrito em 100% Java, significa que também é multiplataforma. Além disso, seus códigos fontes são distribuídos gratuitamente de acordo com a GNU (General Public License).

Através de uma interface gráfica intuitiva, o desenvolvedor é capaz de criar qualquer tipo de relatório de forma simples e rápida. Mesmo sabendo que o iReport desenvolve um formato XML usado pelo JasperReports, o que não é difícil de manipular, há uma vantagem em usar esta ferramenta. Se o desenvolvedor é um usuário iniciante no formato XML do JasperReports, o iReport supre suas necessidades evitando que seja necessário fazer modificações no código fonte. Caso seja experiente neste formato, o iReport minimiza o tempo na criação dos mais complexos relatórios.

Neste capítulo você aprenderá onde obter, os requerimentos necessários para executar, como instalar, compilar e entender sobre algumas das mais importantes características do iReport.

Antes de utilizar o iReport

O iReport, por ser uma ferramenta Java, precisa do Sun Java 2 SDK 1.4 instalado em sua máquina, ou superior. Perceba que não é apenas a JVM (Java Virtual Machine) que deve estar instalada. Isso porque, para compilar os arquivos **jasper**, é necessária a instalação da distribuição do Java SE, sendo impossível somente com a JRE (Java Runtime Environment). Caso queira compilar o iReport, através de seus arquivos fontes, será necessária a instalação do Jakarta Ant na versão 1.6 ou superior.

Obtendo e instalando o Java SE

Atualmente, o JDK (Java TM SE Development Kit) está em sua versão 6 e possui suporte para as seguintes plataformas:

Microsoft Windows

Linux

Sun Solaris

Mac Os

Entre outros

A página Web oficial sobre a linguagem Java da Sun se encontra no endereço <http://java.sun.com>.

A Sun publica periodicamente releases do JDK e costuma numerá-los através do que chamam atualmente de **updates**.

Para obter o Kit de Desenvolvimento Java, que até o momento em que este livro é escrito está na versão 6 update 2 (JDK 6u2), conhecido como Java SE 6.0, acesse o endereço: <http://java.sun.com/javase/downloads/index.jsp>.

Na página de download do Java SE 6.0, existem vários produtos para download. Logo a seguir há uma descrição de alguns dos produtos do qual você encontra para download:

- **JDK 6 Update 2** – Este é o kit de desenvolvimento Java. Este kit consiste em um compilador (javac), uma ferramenta de execução (java), uma ferramenta para geração de documentação (javadoc), um conjunto de bibliotecas prontas para o uso e etc.
- **Java Runtime Environment (JRE) 6 Update 2** – Este é o ambiente runtime do Java, usado apenas para executar os programas feitos em Java.
- **Java SE 6 Documentation** – A documentação do J2SE. Esta documentação contém as informações sobre todas as bibliotecas do Java.

Clique no botão-link **Download**. Para efetuar o download do JDK, você deve aceitar os termos de uso do Java SE, clicando na opção “**Accept**”. Após a página recarregar, haverá a possibilidade de baixar a versão que você precisa para a sua plataforma. Observe que na página de Downloads existem diversos Java SE. Existe Java SE para a plataforma Windows, Linux, Solaris SPARC, Solaris x86 e Linux AMD64. Efetue o download do Java SE para o seu sistema operacional clicando sobre o link do produto que utilizará para instalação.

A instalação no Windows

Tendo em vista que o Windows no qual será descrito está na versão XP ou superior, inicie o assistente com um duplo clique sobre o arquivo executável, que como tantos outros desta plataforma, basta avançar até concluir.

Para configurar qualquer variável de ambiente em um sistema Windows na versão NT, você deve entrar no *Painel de Controle, Sistema* e na guia *Avançado* clique no botão **Variáveis de Ambiente**.

As variáveis de ambiente são seqüências de caracteres que contêm informações como unidade, caminho ou nome de arquivo. As variáveis de ambiente controlam o comportamento de vários programas. Como exemplo de variáveis de ambiente: **PATH, CLASSPATH, USER, TEMP, JAVA_HOME** e etc.

Estas variáveis de ambiente são utilizadas por muitos programas, para definir um espaço em memória principal (RAM), onde alguns processamentos serão executados

Em **Variáveis do sistema**, clique em **Nova**. Adicione o nome da variável como **JAVA_HOME** e no valor o caminho onde você instalou o JDK.

A variável **JRE_HOME** é similar a anterior, mudando apenas para o valor onde se encontra a máquina virtual Java.

Na variável **PATH**, você tem que adicionar a variável **JAVA_HOME**, acrescida do caminho para os binários. Após o último valor, coloque um ponto-e-vírgula (“;”) e adicione **%JAVA_HOME%\bin**.

Ao confirmar as caixas de diálogo, teste se o Java está funcionando na sua máquina em um prompt de comando, como mostrado a seguir:

```
java -version
```

Em Windows versão 9x

Você deve editar o arquivo **autoexec.bat**, encontrado em **C:\autoexec.bat** e adicionar as variáveis como o exemplo a seguir:

```
set JAVA_HOME= C:\jdk1.6.0_02
```

Onde o local indicado no exemplo, é onde você tem instalado o JDK.

A instalação no Linux

No momento em que este livro está sendo escrito, a versão baixada para o sistema operacional Linux é a **jdk-6u2-linux-i586.bin**.

Entre no terminal do seu Linux, após baixar o arquivo, e mude para o usuário Root ou administrativo do sistema e altere a permissão do arquivo para executável, assim você poderá instalá-lo:

```
shell# chmod +x jdk-6u2-linux-i586.bin
```

Execute o arquivo:

```
shell# ./jdk-6u2-linux-i586.bin
```

Assim como ocorre em sistemas operacionais Windows, você terá um assistente para instalação, só que com a diferença de não ser gráfico.

Entre no arquivo **profile** em **/etc/profile** e adicione as linhas em seu final:

```
JAVA_HOME=/jdk1.6.0_02
JRE_HOME=/jdk1.6.0_02/jre
export JAVA_HOME
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/lib
export MANPATH=$MANPATH:$JAVA_HOME/man
```

As variáveis usadas são explicadas rapidamente a seguir, dando a você noção do que cada uma faz:

JAVA_HOME - Diretório onde a JDK está instalada

JRE_HOME - Diretório onde a máquina virtual está instalada

PATH - Diretório dos executáveis, onde inclui o do Java

CLASSPATH - Diretório das libs

MANPATH - Diretório do manual, acessível por **man javac**.

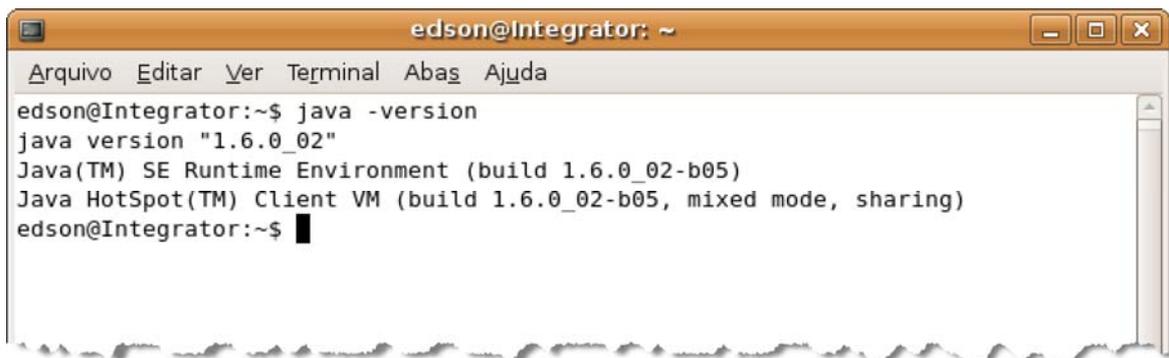
Faça logout no sistema e logon e execute no terminal o seguinte comando:

```
java -version
```

Surgindo as informações mostradas a seguir, a instalação foi feita com sucesso.

```
java version "1.6.0_02"
```

```
Java(TM) SE Runtime Environment (build 1.6.0_02-b06)
```



```
edson@Integrator: ~  
Arquivo Editar Ver Terminal Abas Ajuda  
edson@Integrator:~$ java -version  
java version "1.6.0_02"  
Java(TM) SE Runtime Environment (build 1.6.0_02-b05)  
Java HotSpot(TM) Client VM (build 1.6.0_02-b05, mixed mode, sharing)  
edson@Integrator:~$ █
```

Figura 1 – Verificando a versão de instalação do Java no Linux

Obtendo e instalando o iReport

Para baixar o iReport, entre no endereço oficial <http://jasperforge.org/sf/projects/ireport> ou <http://ireport.sourceforge.net/>. Até o momento em que este livro é escrito a versão atual do iReport está em **2.0.2**. Esta versão tem suporte completo ao **JasperReports 2.0.2**.

Clique no link **Download** para ser redirecionado ao **SourceForge.net**, onde está alocado para baixar. Se você estiver usando o sistema operacional Microsoft Windows, baixe a versão **iReport-2.0.2-windows-installer.exe**, ao qual inclui um instalador, ou o arquivo **iReport-2.0.2.zip**, sem instalador. Para os demais sistemas operacionais, incluindo Linux, baixe a versão **iReport-2.0.2.tar.gz**. O código fonte pode ser encontrado baixando-se o arquivo **iReport-2.0.2-src.zip**.

Atenção: Com algumas mudanças, muitos dos exemplos vistos neste livro poderão ser usados em versões anteriores, após a versão 0.2.0 de iReport.

Algumas características do iReport

O iReport possui diversas características que o tornam uma ferramenta de desenvolvimento de relatórios profissional, no mesmo padrão de outros do mesmo tipo consagrados pelo mundo do desenvolvimento. A seguir você tem algumas de suas principais características que fazem deste programa visual o Oficial para desenvolvimento de relatórios JasperReports:

- Suporte a 100% das tags XML do JasperReports;
- Editor WYSIWYG para criação de relatórios, possuindo ferramentas que incluem desenhos de retângulos, linhas, elipses, caixas de texto, rótulos, gráficos, sub-relatórios, códigos de barras e etc.;
- Um editor para escrever as expressões incluindo destaques (highlights) nas sintaxes;
- Suporte para Unicode e línguas não latinas como: Russo, Chinês, Japonês, Coreano e etc.;
- Integração para compilar e exportar;
- Suporte para todos os bancos de dados acessíveis pela ponte JDBC;
- Suporte virtual para todos os tipos de DataSources;
- Assistentes para criar relatórios rapidamente;
- Suporte para sub-relatórios;
- Backup para o código fonte;
- Suporte para modelos de documentos (templates);
- Suporte para fontes TrueType;
- Extensão através de plug-ins;
- Suporte integrado para scriptlets;
- Suporte a gráficos;
- Gerenciamento de bibliotecas de objetos padrão tais como numeração de páginas;
- Arrastar e Soltar (Drag 'n Drop);
- Ilimitados Desfazer e Refazer;
- Biblioteca de estilos e etc.

Instalando o iReport

Como existem mais de uma forma de instalar o iReport, você verá todas aqui. A primeira forma de instalação que será mostrada é no sistema operacional Windows com o assistente. A segunda forma será sem o uso de assistente, neste caso tanto no sistema operacional Windows como no Linux será mostrada.

No Windows com instalador

O assistente para instalar o iReport no Windows é similar aos demais utilizados em outros programas neste sistema operacional. Dê um duplo clique no arquivo **iReport-2.0.2-windows-installer.exe** para iniciar o processo de instalação pelo assistente.

Na primeira etapa você tem a tela inicial do iReport 2.0.2. Clique no botão **Next** para prosseguir.

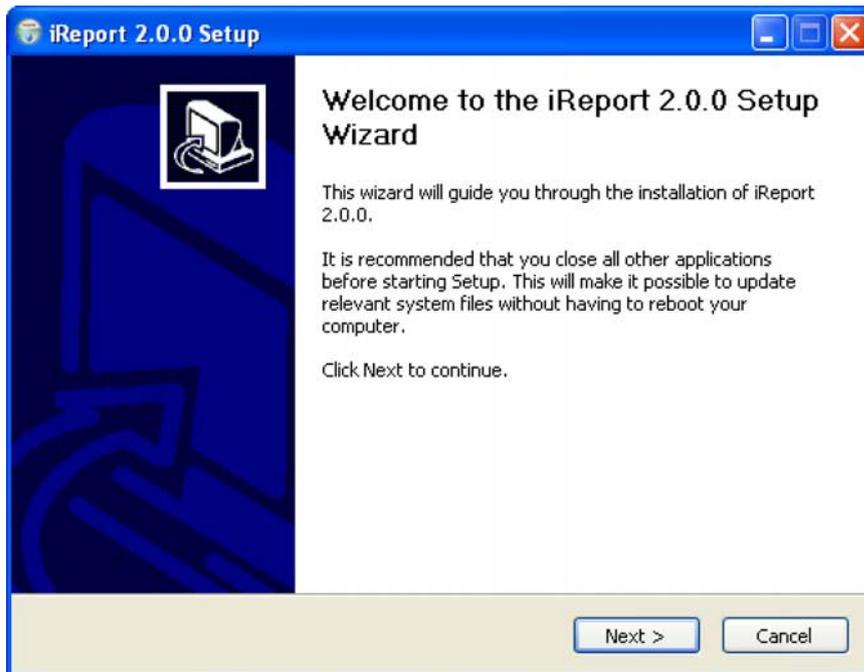


Figura 2 – Primeiro passo: Tela inicial do assistente de instalação do iReport

Na segunda etapa do assistente você deve aceitar a licença de uso do iReport. Caso você não conheça a licença GNU (General Public License), recomendo lê-la antes de prosseguir. Clique no botão **I Agree** para continuar.

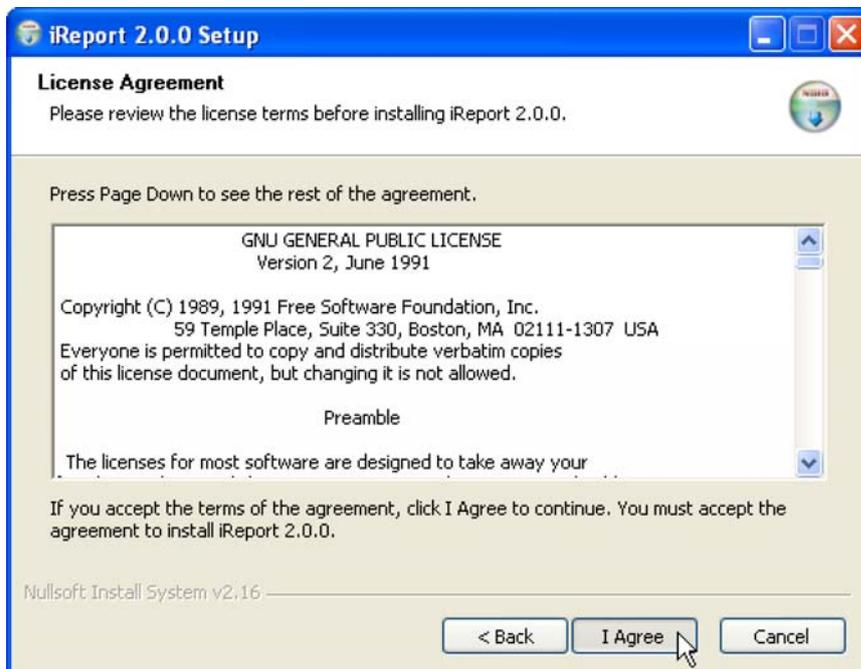


Figura 3 – Segundo passo: Termo de aceite de licença de uso do programa

Na terceira etapa você define os componentes que serão instalados. Caso você queira instalar com os fontes do iReport, deixe marcada a opção **Sources**. Clique no botão **Next** para prosseguir.

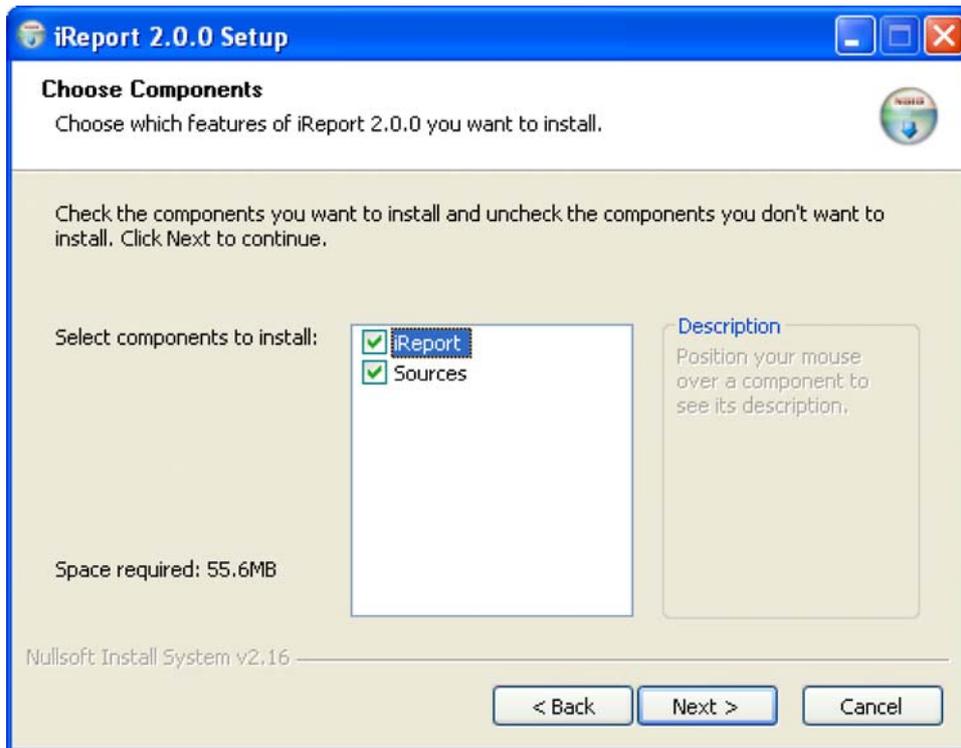


Figura 4– Terceiro passo: Componentes selecionados na instalação do iReport

A quarta etapa é o local onde será instalado o iReport. Por padrão ele é definido dentro do diretório **JasperSoft\iReport-2.0.2** em **Arquivos de programas** do seu sistema operacional Windows. Caso precise mudar, clique no botão **Browse** e selecione o novo local em que deseja instalar. Para prosseguir, clique no botão **Next**.

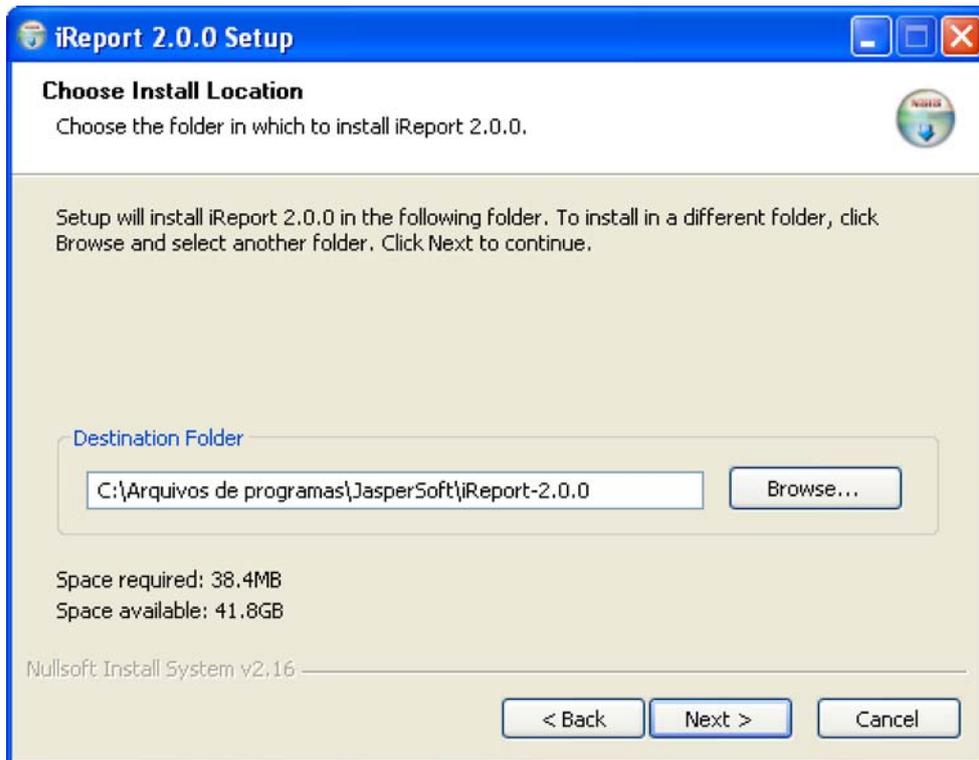


Figura 5 – Quarto passo: Definindo o local de instalação

No quinto passo você define o local de acesso no menu Iniciar do Windows. Caso concorde com a atual configuração, basta clicar no botão **Install** para iniciar o processo de instalação. Caso deseje alterar o acesso, basta digitar no local selecionado, em **JasperSoft\Report-2.0.2**.

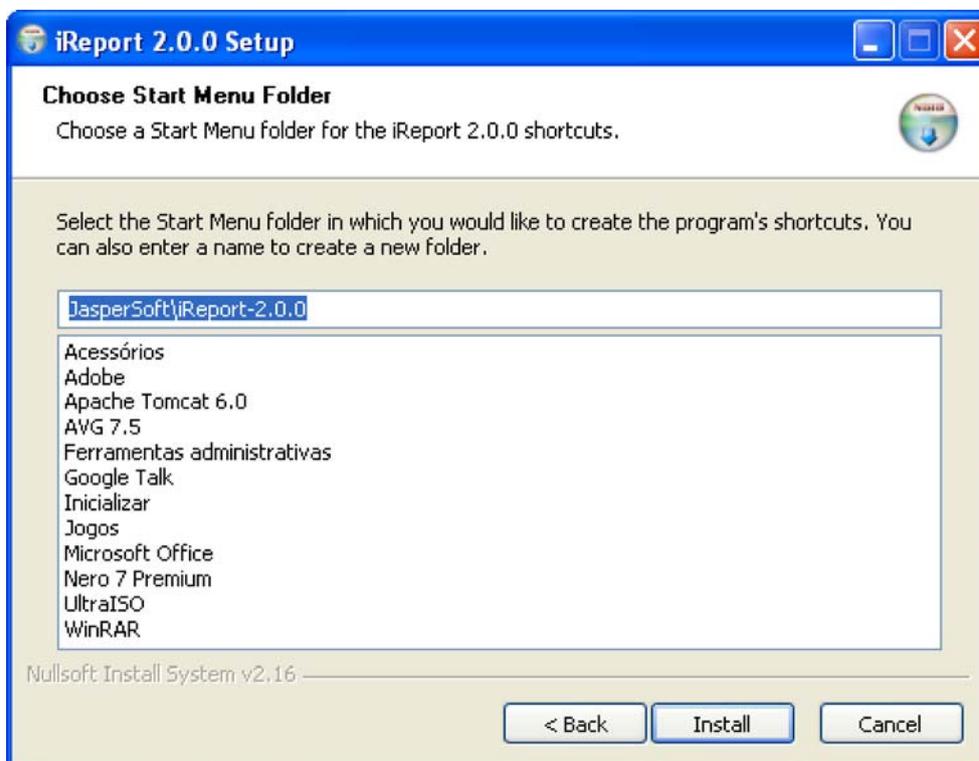


Figura 6 – Quinto passo: Seleção da pasta para acesso no menu Iniciar

Após a instalação, a opção **Run iReport 2.0.2** estará selecionada. Clicando no botão **Finish**, abrirá imediatamente o iReport para trabalhar. O instalador também criará um atalho na área de trabalho para acesso rápido ao programa, através do arquivo **iReport.exe**.

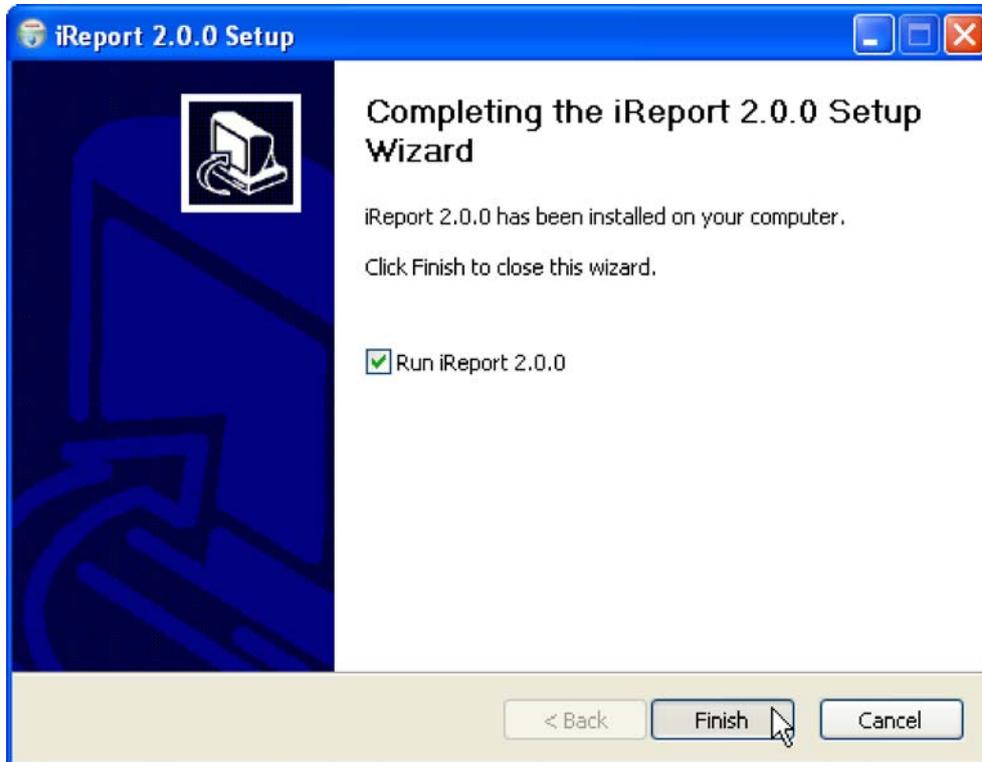


Figura 7 – Finalizando o assistente após a instalação

Atenção: O instalador cria, além do arquivo **iReport.exe**, um arquivo chamado **iReport.bat**, que no caso de não ser possível a inicialização pelo executável, por qualquer motivo que seja, tente através deste, ou diretamente no diretório **bin** através do arquivo **startup.bat**.

No Windows e Linux sem instalador

Caso você prefira a versão sem instalador, existem os dois arquivos compactados **iReport-2.0.2.zip** (recomendável para o sistema operacional Windows) ou **iReport-2.0.2.tar.gz** (para o sistema operacional Linux). Descompacte o de sua preferência em um local desejado.

No Linux, você deve executar antes o comando para descompactação:

```
shell# tar -xpvzf iReport-2.0.2.tar.gz
```

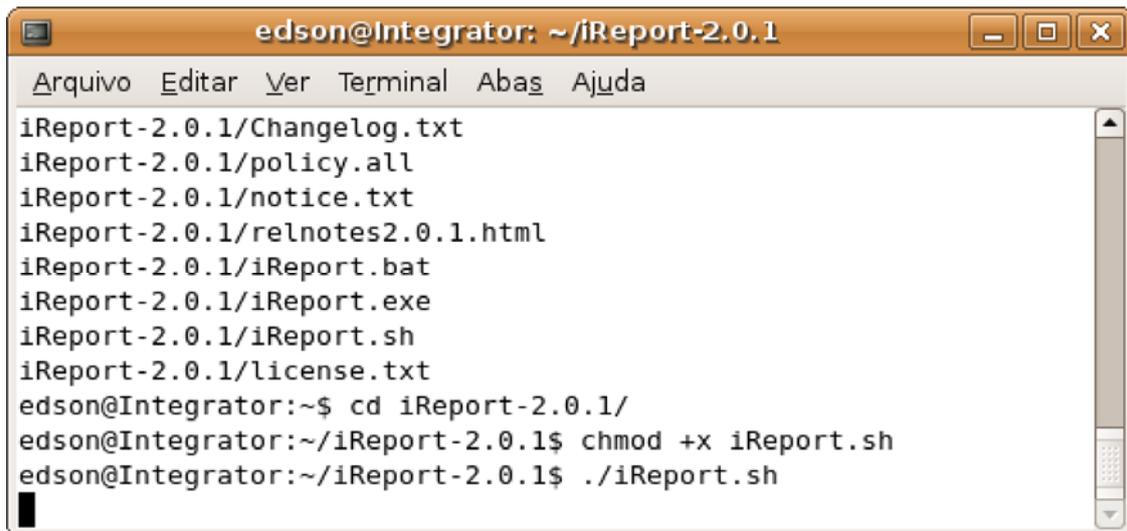
Ao descompactar o iReport, entre no diretório e execute a permissão de execução:

```
shell# cd iReport-2.0.2/
```

```
shell# chmod +x iReport.sh
```

Para o Windows existem os arquivos **iReport.exe** e **iReport.bat** para inicializar, bastando um duplo clique. No sistema operacional Linux, basta entrar no terminal e executar o Shell, como mostrado a seguir:

```
shell# ./iReport.sh
```



```
edson@Integrator: ~/iReport-2.0.1
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
iReport-2.0.1/Changelog.txt
iReport-2.0.1/policy.all
iReport-2.0.1/notice.txt
iReport-2.0.1/relnotes2.0.1.html
iReport-2.0.1/iReport.bat
iReport-2.0.1/iReport.exe
iReport-2.0.1/iReport.sh
iReport-2.0.1/license.txt
edson@Integrator:~$ cd iReport-2.0.1/
edson@Integrator:~/iReport-2.0.1$ chmod +x iReport.sh
edson@Integrator:~/iReport-2.0.1$ ./iReport.sh
```

Figura 8 – Permissão e execução do iReport no Linux

Capítulo 2

“Especialize-se em um setor determinado e procure conhecer mais sobre ele que qualquer outra pessoa”.

Joseph Murphy

Iniciando no iReport

O iReport é uma ferramenta que está preparada para construir visualmente os mais complexos relatórios dentro dos limites de JasperReports.

Embora o primordial de iReport seja de desenhar relatórios, existem também suas características para configurações.

Este capítulo o introduzirá ao desenvolvimento básico de relatórios usando o iReport, partindo de sua execução inicial e suas configurações básicas.

Abrindo o iReport pela primeira vez

Quando o iReport executa pela primeira vez ele carrega informações de configurações regionais de seu sistema operacional e cria um diretório chamado de **.ireport** no diretório do usuário em **home** no Linux ou em **Document and Settings**. Neste diretório você encontra dois arquivos padrão: **config.xml** e **ireport2.0.2.location**.



Figura 1 – Símbolo do iReport

Suas configurações pessoais e as configurações do programa estão gravados neste diretório, em seu arquivo **config.xml**. O **ireport2.0.2.location** apenas armazena o caminho físico para a localização do diretório de instalação do iReport.

Atenção: Se não for criado o diretório **.ireport**, alguns problemas indesejáveis podem ocorrer na execução do programa além da impossibilidade de gravação das configurações personalizadas. Neste caso, poderá ser necessária a criação do diretório manualmente.

Antes de executar o iReport, copie o arquivo **tools.jar** do diretório **lib** do JDK instalado para o diretório de mesmo nome no local de instalação da ferramenta (por exemplo **/home/edson/iReport-2.0.2/lib**). A ausência deste arquivo pode produzir algumas exceções durante a compilação do relatório (ocorridos por usar classes contidas desta biblioteca). No MacOS X utilize o arquivo **classes.jar**.

Documentos Recentes não são exibidos

Em alguns casos, devido à localização de alguns arquivos, quando você fecha o iReport e torna a abri-lo, os **Recent Files** (Documentos Recentes), opção do menu **File** (Arquivo) não os exibe. Isso ocorre porque o caminho pode conter caracteres que são inválidos no formato **UTF-8**. Para arrumar este problema, procure não utilizar acentos em diretórios e arquivos salvos.

Criando seu primeiro relatório com iReport

Antes de continuar nas configurações do iReport, você vai se ambientar na ferramenta, possibilitando assim uma melhor compreensão da geração de relatórios JasperReports.

Para a geração deste relatório, você vai utilizar o assistente de relatórios existente na ferramenta. Isso facilitará na velocidade do desenvolvimento.

O banco de dados usado

O banco de dados usado neste livro será o MySQL 5. Apesar de ser o MySQL, os exemplos mostrados poderão ser feitos em Oracle, SQL Server, PostgreSQL e etc.

No **Apêndice A** você tem todos os passos iniciais deste banco de dados caso não tenha experiência com o MySQL.

O banco de dados que será usado neste exemplo, chamado de **relatorios**, e nos demais ao longo do livro está sendo apresentado na **Figura 2** a seguir, com suas devidas tabelas.

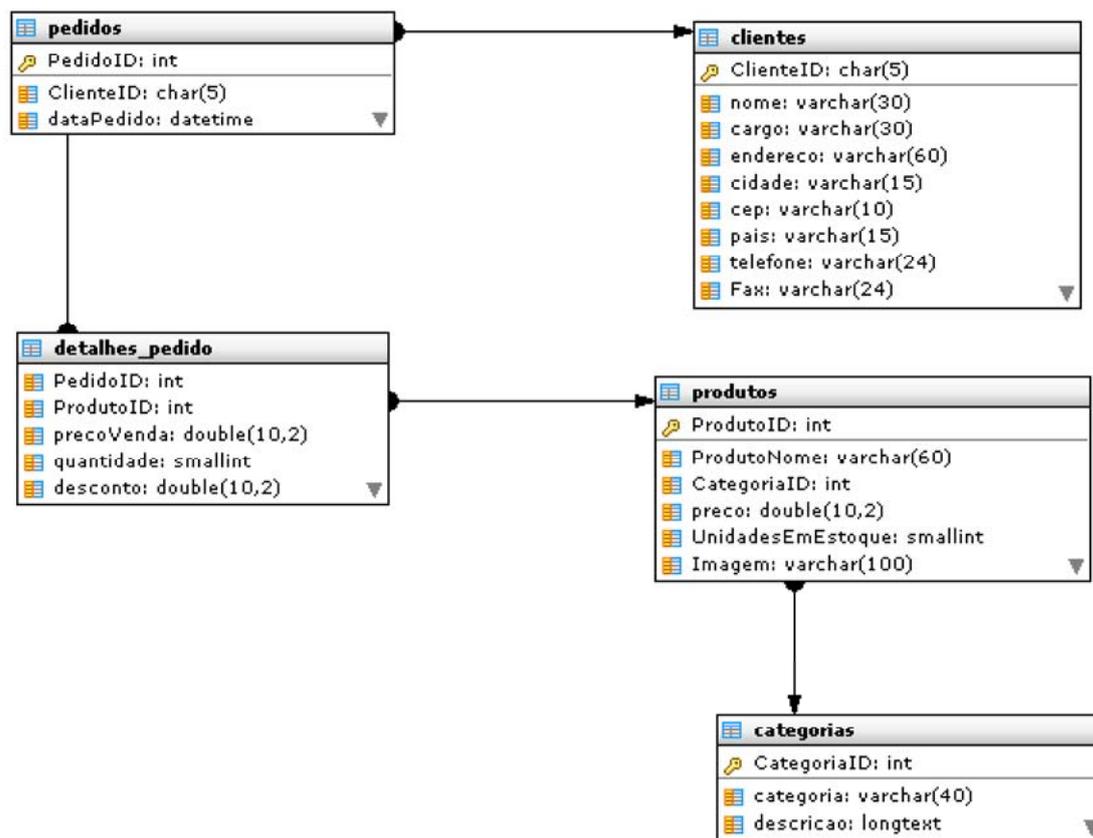


Figura 2 – Diagramação do banco de dados usado no livro

Observação: As tabelas foram baseadas no banco de dados de exemplo Northwind, encontrado no MS Office Access ® e no banco de dados MS SQL Server ®, ambos da Microsoft (MS).

Antes de criar o primeiro relatório

O idioma padrão utilizado pelo iReport é o configurado em seu sistema operacional. Se o seu sistema operacional estiver em português, como ocorre no meu caso, como autor, o iReport irá utilizar esta configuração.

O exemplo proposto será baseado em tal idioma, até que você aprenda a configurá-lo e então, utilizar o Inglês.

Observação: É escolha sua alterar ou não o idioma. Se quiser permanecer no idioma nativo, basta não configurá-lo. Mas vale lembrá-lo que, fora o idioma inglês, podem haver algumas falhas na tradução, ocasionando em reparo do idioma por sua parte.

O primeiro relatório

O banco de dados a ser criado é composto por cinco entidades, sendo quatro delas principais para o cenário proposto. Para o exemplo, você vai criar o primeiro relatório usando apenas a tabela de **produtos**.

Vá ao menu **Arquivo** (File) do **iReport** e selecione no menu o item **Assistente de Relatório...** (Report Wizard)

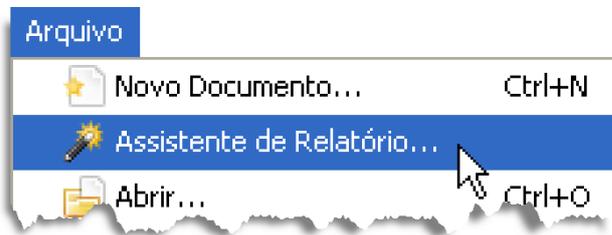


Figura 3 – Selecionando o Assistente de Relatório

Ao surgir à caixa de diálogo do assistente, chamada de **Assistente do iReport**, haverá a necessidade de criar uma conexão com o banco de dados. Para isso, como você ainda não fez nenhuma, clique no botão **Novo** (New), ao lado do campo **Conexões / Fonte de Dados** (Connections/Data Sources), como exhibe a **Figura 2.4** a seguir.

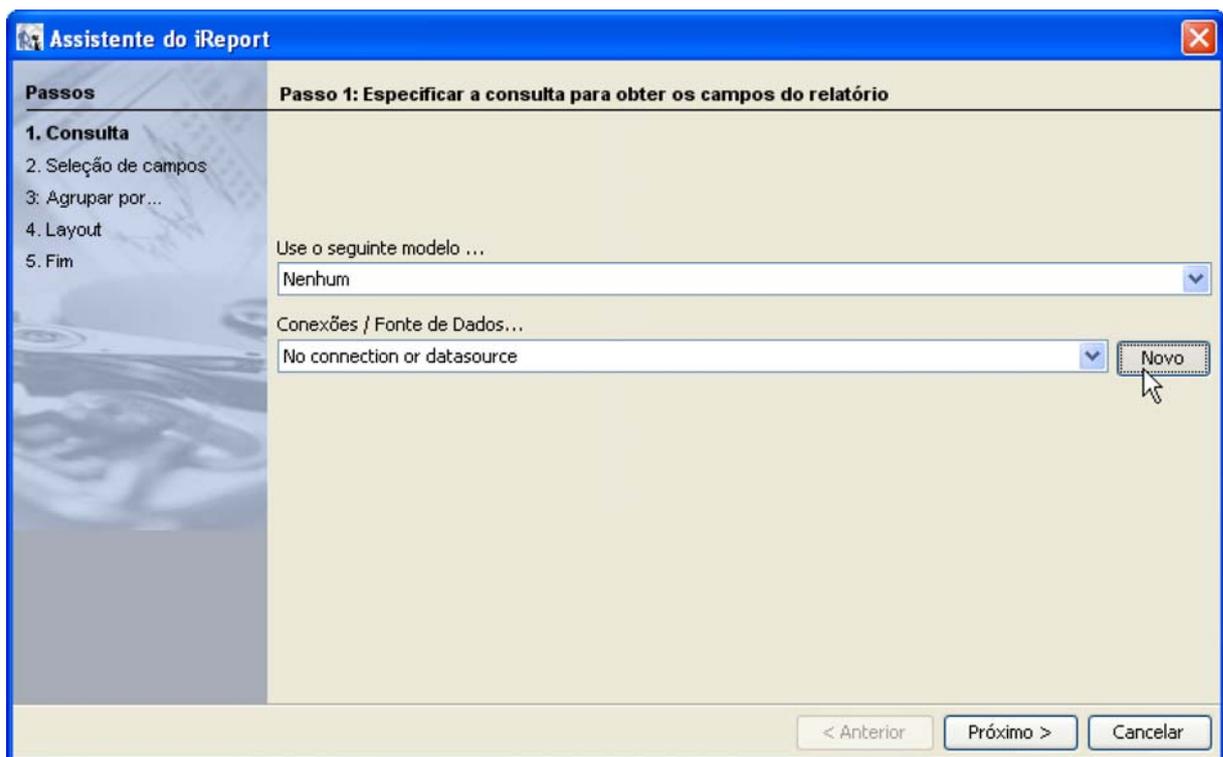


Figura 4 – Criando uma conexão

A caixa de diálogo **Propriedades de Conexão** (Connections properties) surgirá para ser definida a forma como será criada a conexão com o banco de dados. No caso, você vai utilizar a ponte JDBC, selecionando o item **Conexão de Banco de Dados JDBC** (Database JDBC connection). Clique no botão **Próximo** (Next) para prosseguir.

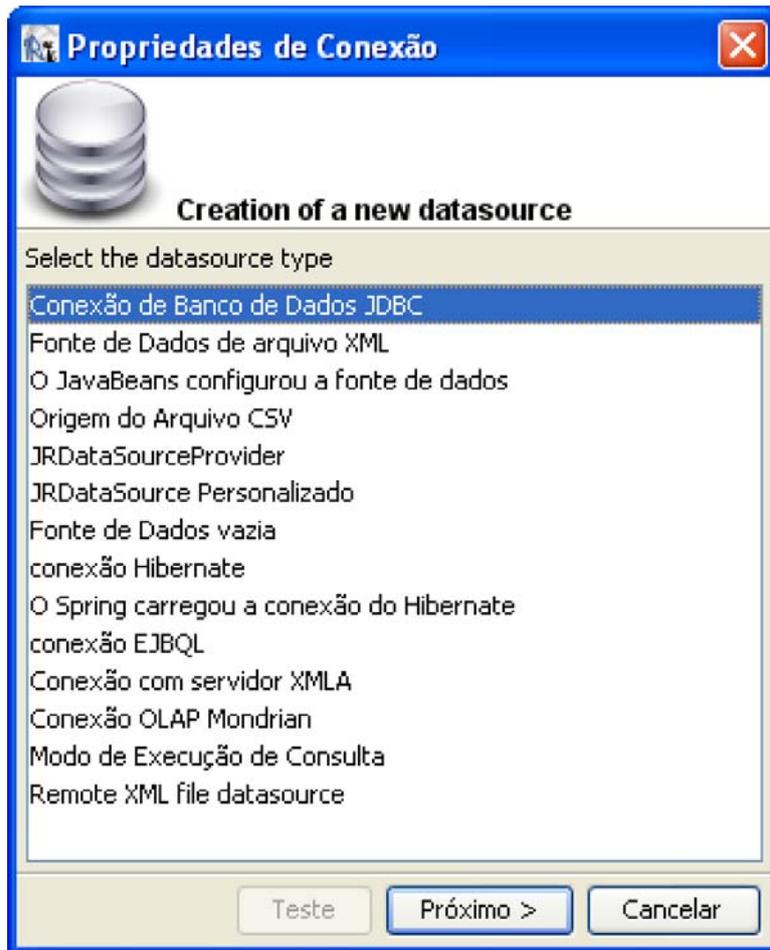


Figura 5 - Caixa de diálogo Propriedades de Conexão

Como o driver JDBC usado é do MySQL, por padrão, a segunda etapa das propriedades de conexão surgirá com esta opção selecionada no campo **Driver JDBC** (JDBC Driver), contendo **com.mysql.jdbc.Driver**. Não há necessidade, neste caso, de se adicionar a biblioteca JDBC na sua aplicação.

No campo **Nome** (Name) digite **relatorios**. No campo **Caminho do JDBC** (JDBC URL) complete deixando **jdbc:mysql://localhost/relatorios**. Coloque o nome de usuário no campo **Usuário** (User Name) e sua respectiva senha em **Senha** (Password). Marque a opção **Salvar senha** (Save password).



Figura 6 – Definindo as propriedades de conexão

Clique no botão **Teste** (Test). Caso apareça a mensagem **Conexão testada com sucesso!** (**Connection test successful!**), significa que você configurou tudo corretamente e que agora há uma comunicação com o banco de dados. Clique no botão **OK** para confirmar e no botão **Salvar** (Save) da caixa de diálogo **Propriedades de Conexão** (Connections properties) para guardar a conexão criada.

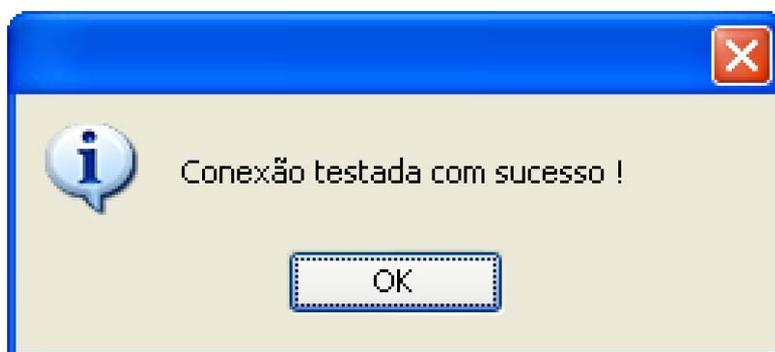


Figura 7 – Confirmação da conexão

Onde o iReport armazena minha conexão?

O iReport guarda as informações da configuração da conexão no diretório .ireport, dentro do arquivo **config.xml**.

Neste arquivo há o elemento `<iReportConnection />` que utiliza a classe `it.businesslogic.ireport.connection.JDBCConnection` para se conectar ao banco de dados utilizando JDBC. O nome da conexão fica no atributo **name**. As configurações são armazenadas no sub-elemento `<connectionParameter/>`.

Voltando ao assistente de relatório, na primeira etapa, concluída a conexão, você só precisa definir a tabela e dados que serão exibidos. No campo **Consulta SQL** (SQL query) digite a query a seguir que selecionará somente a tabela de produtos:

SELECT * FROM produtos

Clique no botão **Próximo** para prosseguir.

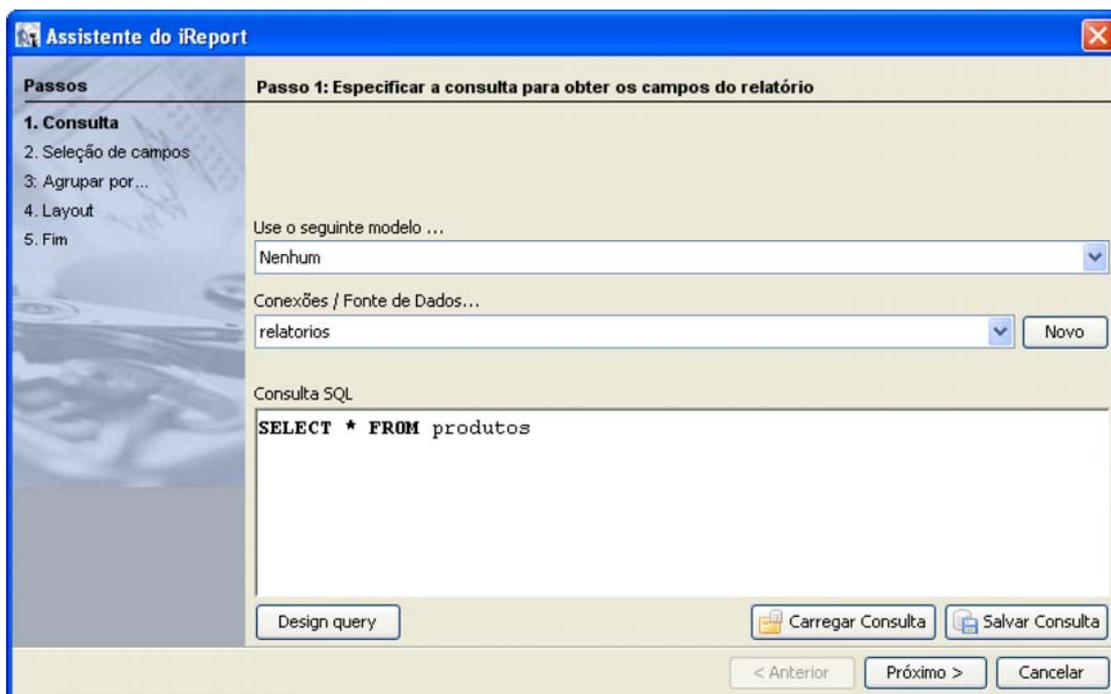


Figura 8 – Primeira etapa - Criando a consulta SQL

Na segunda etapa do assistente, em **Seleção de campos** (Fields selection), adicione todos os campos exceto **CategoriaID** (utilizando o botão “>”) e clique no botão **Próximo** (Next).

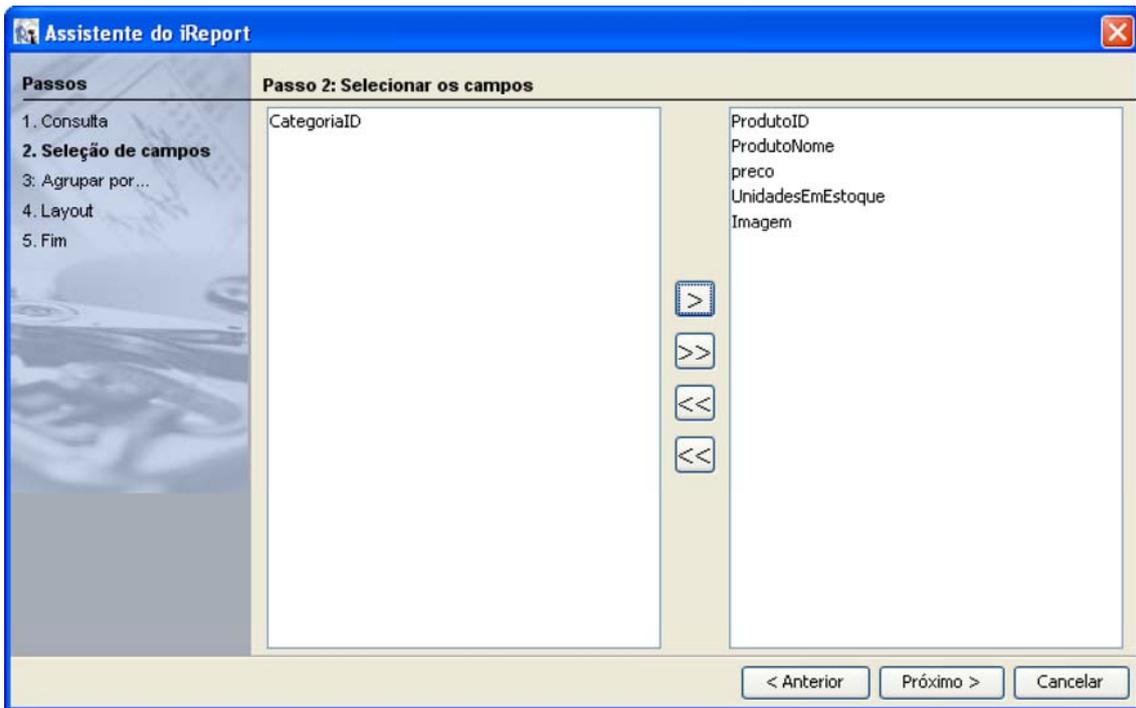


Figura 9 – Segunda etapa - Seleção de campos

Na terceira etapa simplesmente avance clicando no botão **Próximo** (Next).

Na quarta etapa, você vai selecionar o layout que deseja utilizar em seu relatório. Selecione um relatório na opção marcada **Layout colunar** (Columnar layout). Clique no botão **Próximo** (Next).

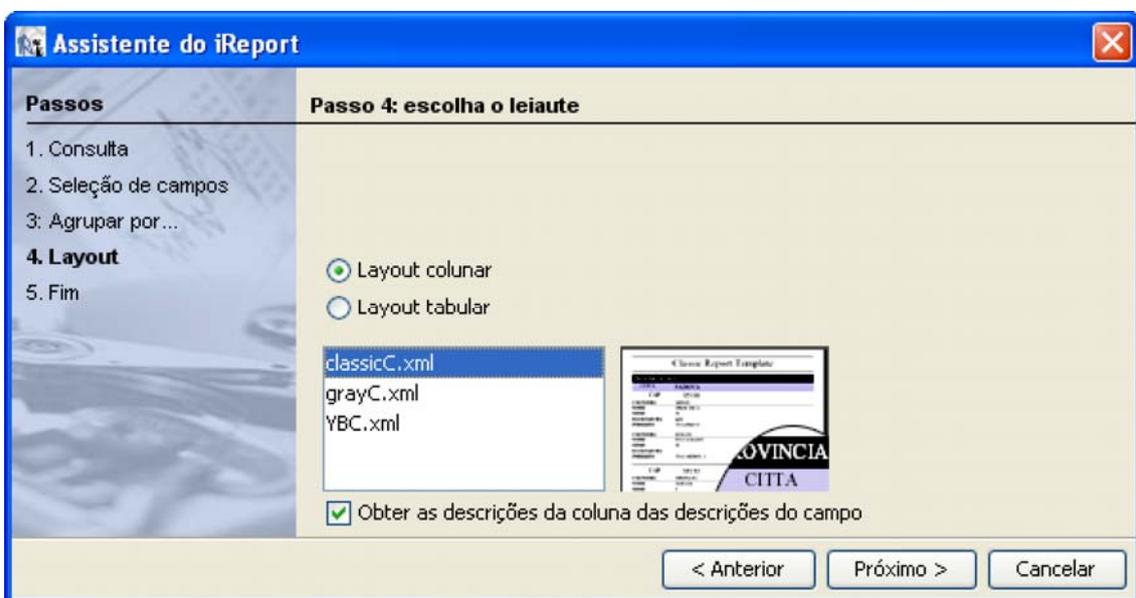


Figura 10 – Quarta etapa – Definindo seu Layout

Na quinta e última etapa, clique simplesmente no botão **Encerrar** (Finish) para concluir o assistente.



Figura 11 - Última etapa – Confirmando a criação

Ao final você terá um relatório gerado com os campos selecionados em formato tabular, contendo sua visualização apenas estrutural.

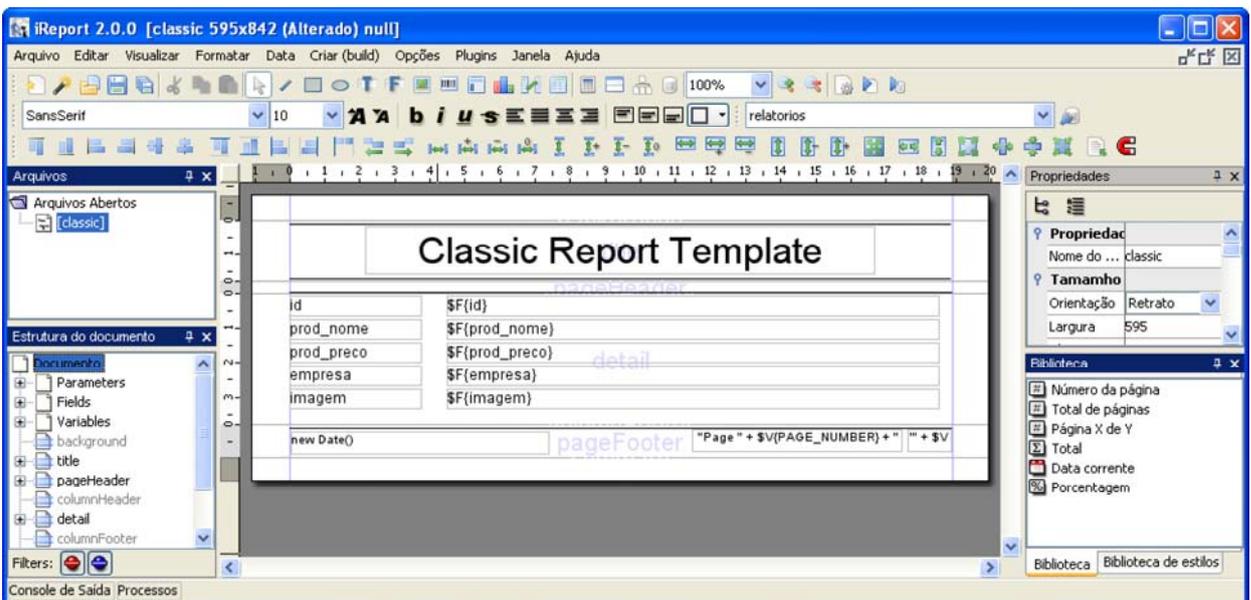


Figura 12 - O relatório gerado pelo assistente

Observação: O driver JDBC do MySQL já vem com o iReport, no diretório **lib** de sua instalação. Caso você deseje utilizar outro banco de dados ou atualizar a versão JDBC da biblioteca do MySQL, que se encontra na versão **mysql-connector-java-3.1.11-bin.jar**, adicione neste diretório.

Visualizando o primeiro relatório

Antes de iniciar qualquer alteração, você vai visualizar o relatório da forma bruta como foi criado.

Vá ao menu **Criar (build)** e clique na opção **Executar relatório (usar conexão ativa) – Execute (with active connection)**.

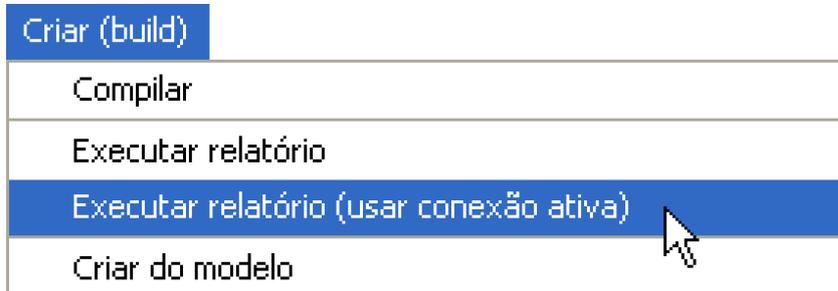


Figura 13 – Executando o relatório com a conexão ativa

Ao surgir a caixa de diálogo para salvar o arquivo, chamada de **Save report as XML jasperreports file**, dê um nome ao arquivo, mantenha na extensão **.jrxml** e clique no botão **Save**.

Surgirá o visualizador de relatórios próprio do JasperReports chamado de **JasperViewer**. Um ótimo visualizar desktop escrito em Java, contendo o básico de todo visualizador.

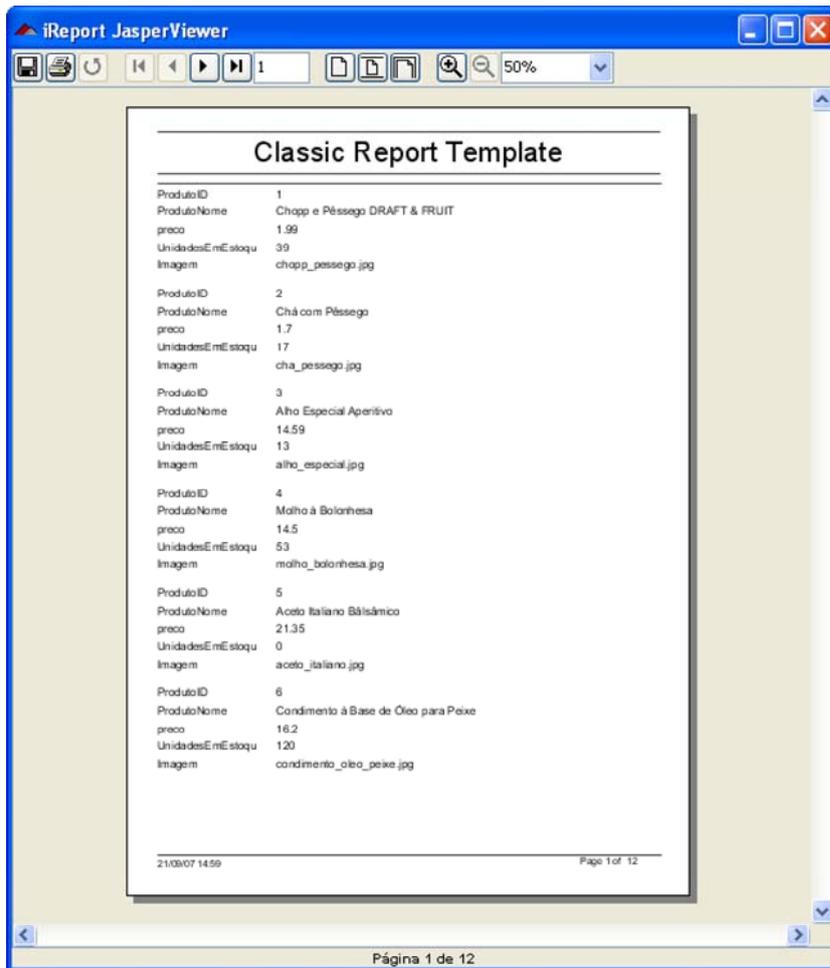


Figura 14- Visualizador de relatórios do JasperReports (JasperViewer)

Configurando o iReport

Agora com a criação de um relatório básico, você está preparado para configurar o iReport em algumas de suas características, incluindo o idioma.

Vá ao menu **Opções** (Options) e clique no item **Opções** (Settings).

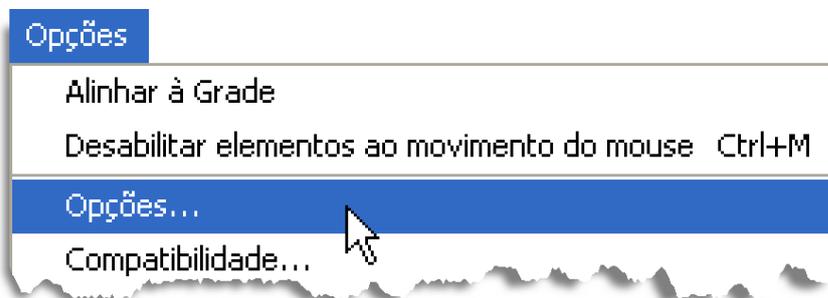


Figura 15 – Menu Opções

Ao surgir à caixa de diálogo **Opções** (Options), a primeira guia mostrada é a **Geral** (General). No grupo **Geral** (General), você pode fazer configurações destinadas à aparência do iReport.

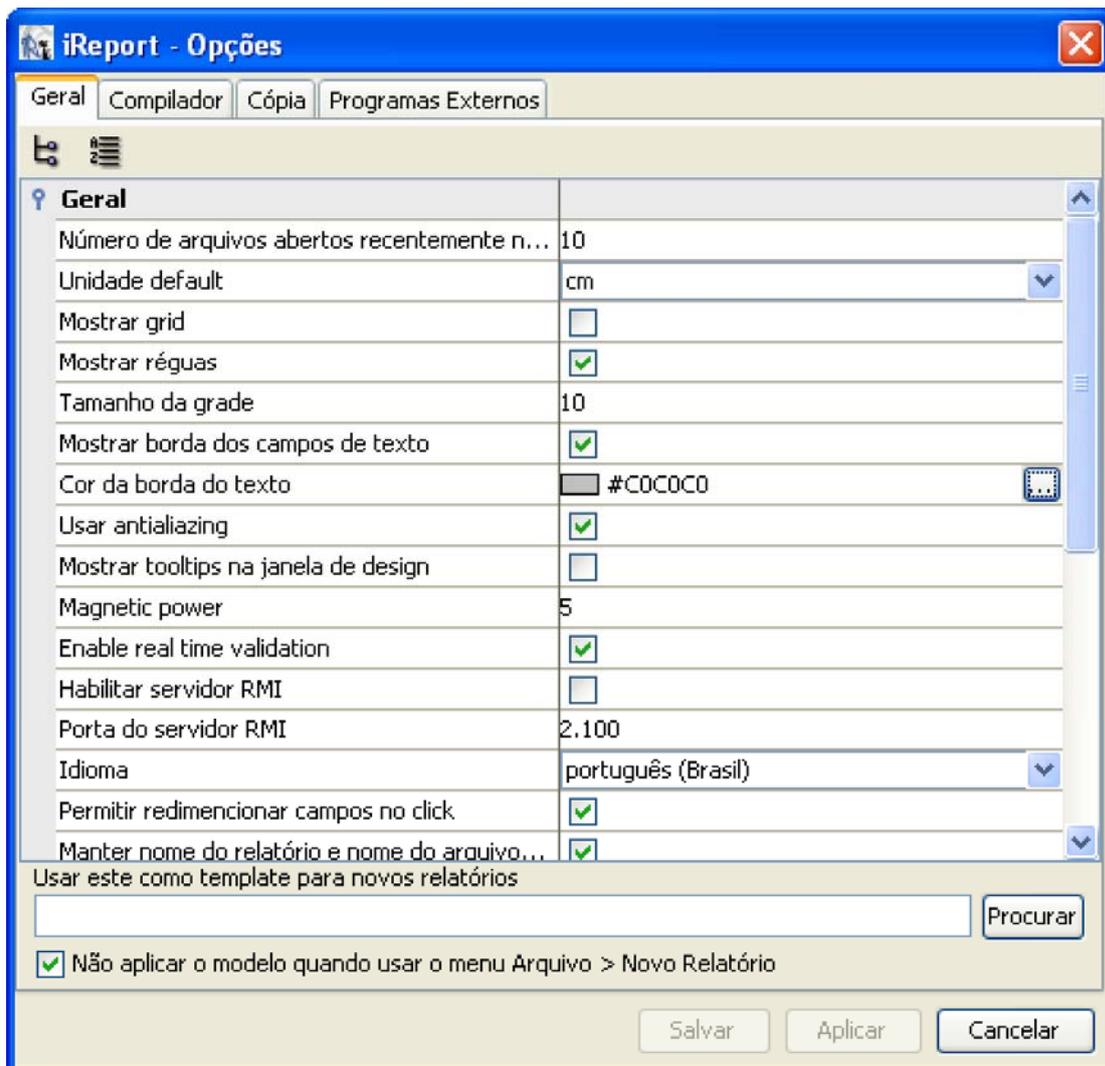


Figura 16 – Caixa de diálogo Opções

Como principais itens de configuração você tem:

Em **Geral**:

- **Unidade default** (Default Unit) – Indica a unidade utilizada para o trabalho com os relatórios. Podem ser: **mm**, **cm**, **pixels** e **polegadas**.
- **Mostrar grid** (View Grid) – Desabilitado por padrão, exibe um grid para melhorar o alinhamento dos componentes existentes na construção do relatório.
- **Mostrar réguas** (View Rules) – Habilitado por padrão, exibe a régua na área de desenho do iReport.
- **Tamanho da grade** (Grid Size) – O tamanho em pixels da grade (grid).
- **Cor da borda do texto** (Text Border Color) – Altera a cor da borda dos elementos existentes no relatório com relação a exibição de rótulos, textos dinâmicos e etc.
- **Magnetic Power** – Indica em pixels a distância de atração dos elementos para as linhas guias (você verá linhas guias mais adiante).
- **Idioma** (Language) – Utiliza as configurações regionais do sistema operacional para definir o idioma aqui. No caso, por padrão, está em português (Brasil), caso seu sistema esteja configurado neste idioma.

Atenção: Altere o idioma para English (Inglês) neste ponto. Clique no botão Aplicar (Apply) para prosseguir.

Em **Padrões do Relatório (Report Defaults)**:

- **Linguagem default para expressões** (Default Language for Expressions) – Por padrão é **Java**, mas pode ser alterado para **Groovy** (veja mais sobre Groovy em relatórios JasperReports no Capítulo 10 deste livro).
- **Estilo Padrão** (Default Style) – Determina um estilo padrão a ser utilizado em um relatório. Por padrão é **None** (Nenhum).

Em **LookAndFeel**:

- **Look and Feel** – O padrão é **System** (formato visual das janelas de acordo com o sistema operacional). Mas como em Java existem outros padrões visuais para aplicações desktop, você pode selecionar aqui o que mais lhe agrada. Escolha e clique no botão **Aplicar** para visualizar.
- **Fonte** (Font) – A fonte padrão usada pelo iReport.
- **Tamanho da fonte** (Font size) – O tamanho da fonte utilizada no iReport.

Na guia **Compilador** (Compiler) você pode definir onde o iReport poderá armazenar os arquivos **jasper** que são compilados. Por padrão, o iReport utiliza o seu diretório corrente para guardar os arquivos compilados. Para alterar, clique no botão **Procurar** (Browse) do campo **Diretório padrão de compilação** (Default compilation directory).

Caso deseje armazenar os arquivos compilados (.jasper) no mesmo diretório onde salvou o relatório .jrxml, marque a opção **Usar o diretório de relatório para os compilados** (Use Reports Directory to Compile).

Em **Compilador** (Compiler), você tem o **JasperReports default** selecionado. O JasperReports pode utilizar outro compilador para o seu relatório. Caso você queira usar o compilador Java (javac), você deve adicionar a biblioteca **tools.jar** ao diretório **lib** como já foi dito no início deste capítulo obrigatoriamente.

Outra opção interessante está no **Virtualizador de Relatório** (Report Virtualizer). Mais adiante você terá mais informações sobre esta opção na utilização de relatórios grandes.

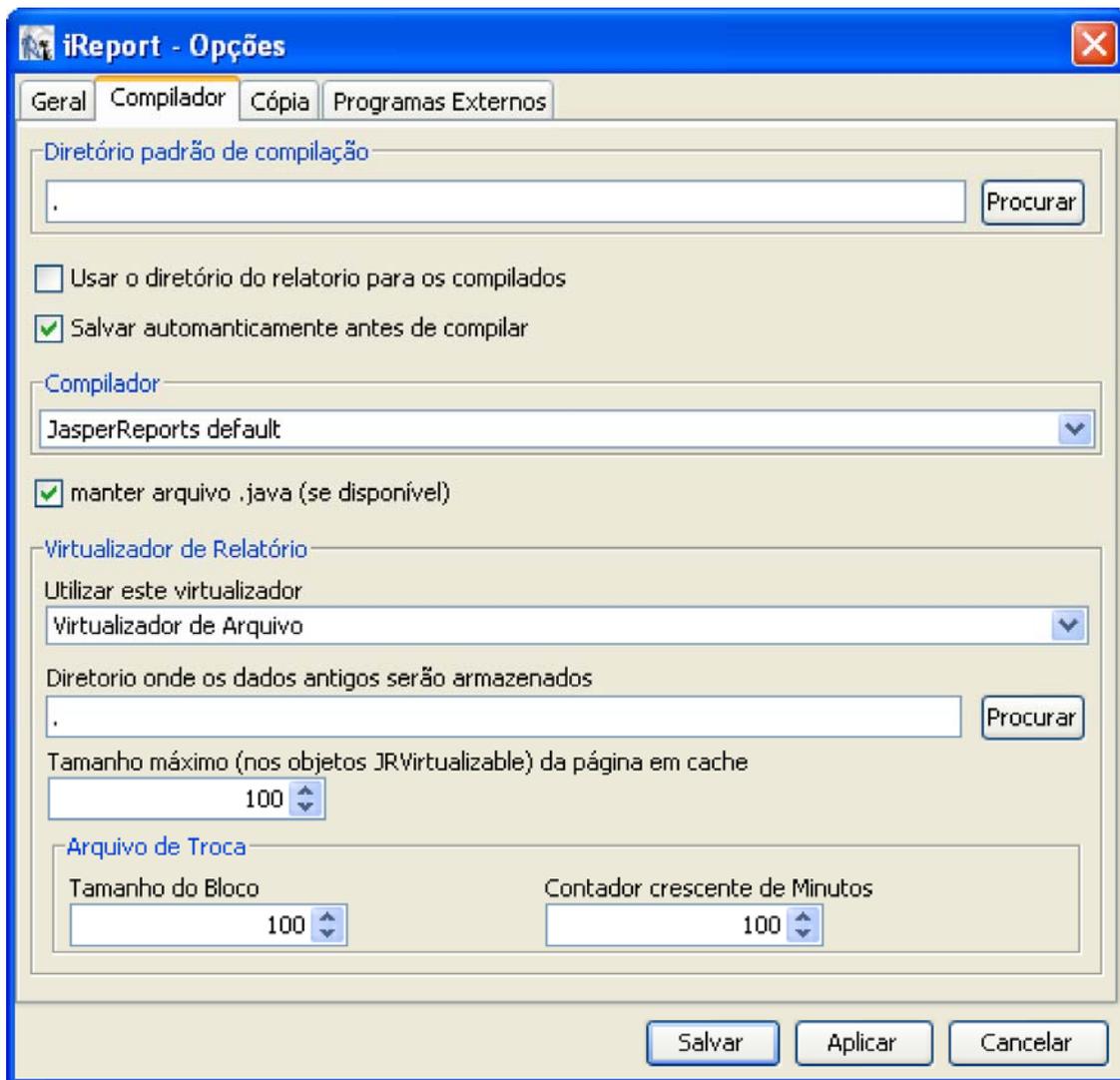


Figura 17 – Guia Compilador

Clicando na guia **Cópia** (Backup) você pode definir se deseja fazer ou não cópia de backup caso queira, além de definir o local onde serão armazenadas as cópias de backup. Por padrão, o iReport utiliza o seu diretório corrente para armazenar os arquivos de backup.

Na guia **Programas Externos** (External Programs) você pode selecionar cada um dos visualizadores que deseja utilizar quando compilar e exportar o relatório no determinado formato.

Adicione um programa que deseja utilizar para cada tipo de arquivo, principalmente o PDF que será muito utilizado ao longo do livro.

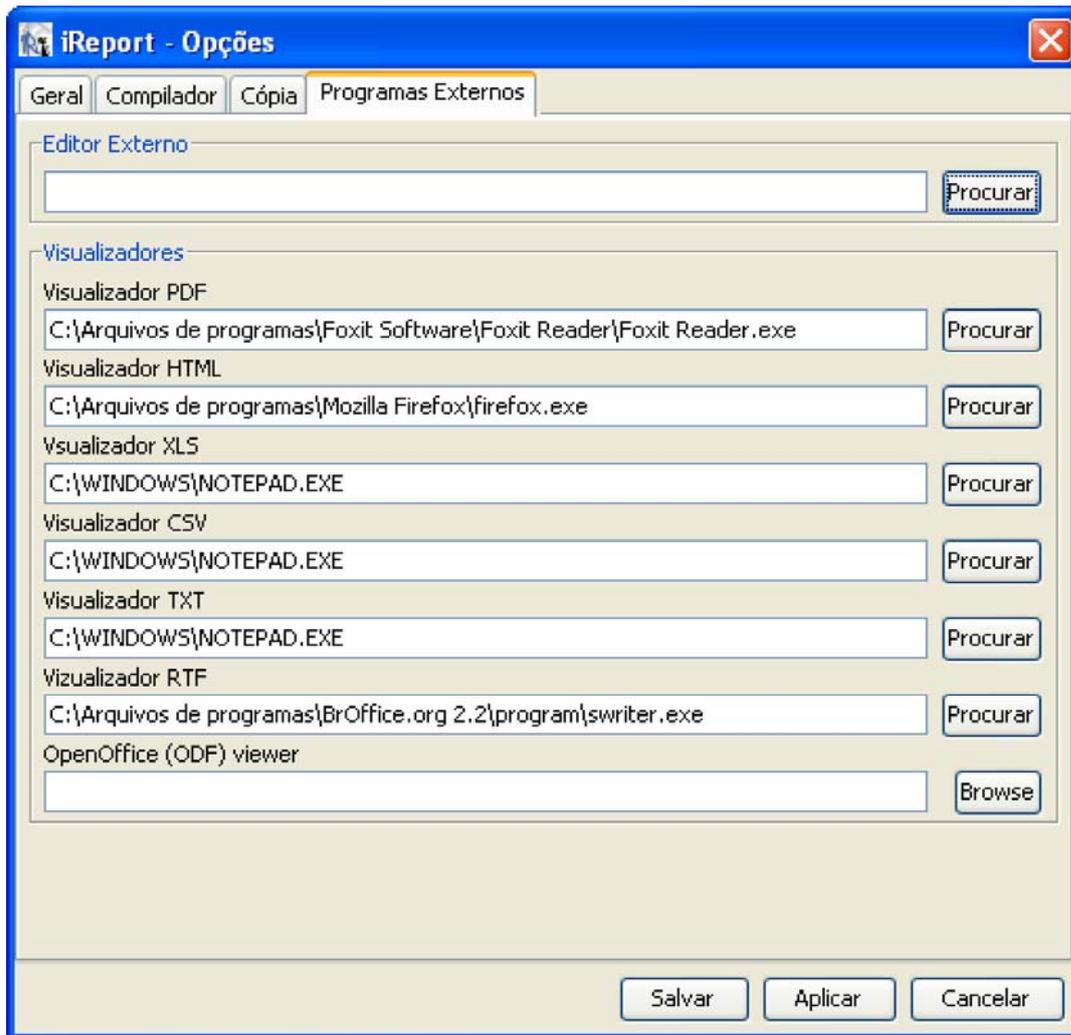


Figura 18 – A guia Programas Externos

Feita todas as alterações, clique no botão **Salvar** (Save) para guardar as configurações efetuadas e fechar a caixa de diálogo.

Visualizando o relatório compilado em outro aplicativo

Para finalizar, você pode querer visualizar o relatório compilado em outro aplicativo, como, por exemplo, em PDF. Vá ao menu **Criar (build)** e marque a opção **Visualizar PDF (PDF Preview)**. Retorne ao mesmo menu e clique no item **Executar relatório (usar conexão ativa) – Execute (with active connection)**.

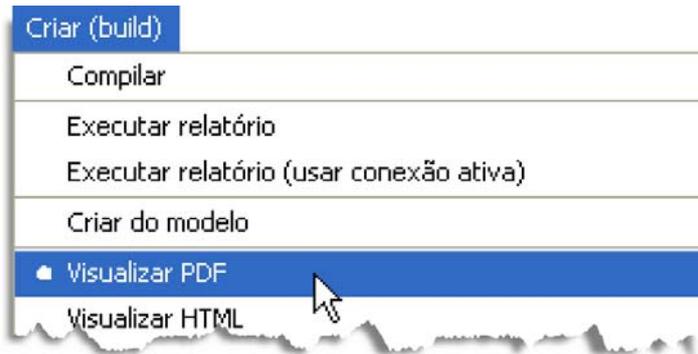


Figura 19 – Detalhe no item Visualizar PDF

Assim que compilar, o programa configurado no iReport para exibir arquivos no formato PDF e o relatório agora neste formato será aberto.

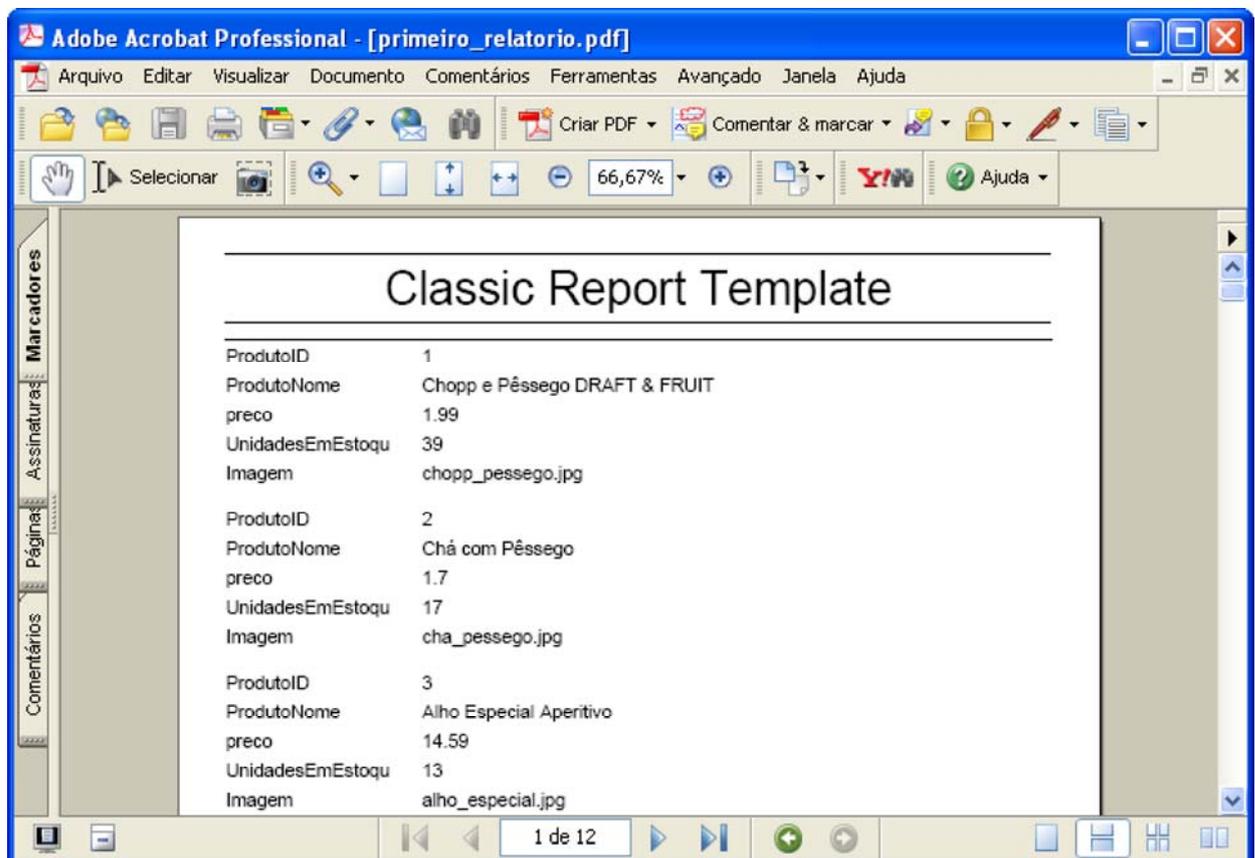


Figura 20 – O relatório compilado em PDF sendo exibido pelo Adobe® Acrobat

Capítulo 3

“A única hora que você não pode falhar é na última vez que tentar”.

Charles Kettering

Estrutura de um relatório e seu desenvolvimento

Um relatório é uma maneira agradável de apresentar os dados vindos de alguma base de informação com sua aparência formatada. Na maioria das vezes, relatórios são usados para agrupar as informações vindas de um grupo de dados, relacionados entre si, oriundos de diversas tabelas. Como resultado você tem apenas os campos determinados na construção do seu relatório e exposto como tal de forma a simplificar a compreensão das informações. Vindo em um formato padronizado para impressão, como um texto feito em um editor de textos, você só precisa mandar para a impressão ou exportá-los para outro formato de uso maior.

Neste capítulo você irá conhecer a estrutura de um relatório JasperReports na ferramenta iReport e suas propriedades.

A estrutura do relatório

Um relatório no JasperReports possui diversas seções, separadas por linhas horizontais ao qual são chamadas de **Bands (bandas)**.

O comportamento de cada Band depende da funcionalidade para a qual está preparada. Quando o relatório junta aos dados para rodar a impressão, estas seções são impressas na tela de diferentes maneiras com diferentes tempos. Por exemplo, o cabeçalho da página (**pageHeader**) é repetido em todas as páginas, uma vez em cada página, enquanto que os detalhes onde serão exibidos os dados (**detail**) é gerado em uma seqüência elaborada de acordo com os dados existentes (**query** ou **datasource**), repetindo-se uma vez para cada registro, incorporando-se ao desenho do relatório.

As bands de um relatório JasperReports é dividido em nove seções pré-definidas para que novos grupos possam ser adicionados.

Quando você inicia um novo relatório (New Document) sem o uso de assistente, a sua tela será como mostrada a **Figura 1** a seguir.

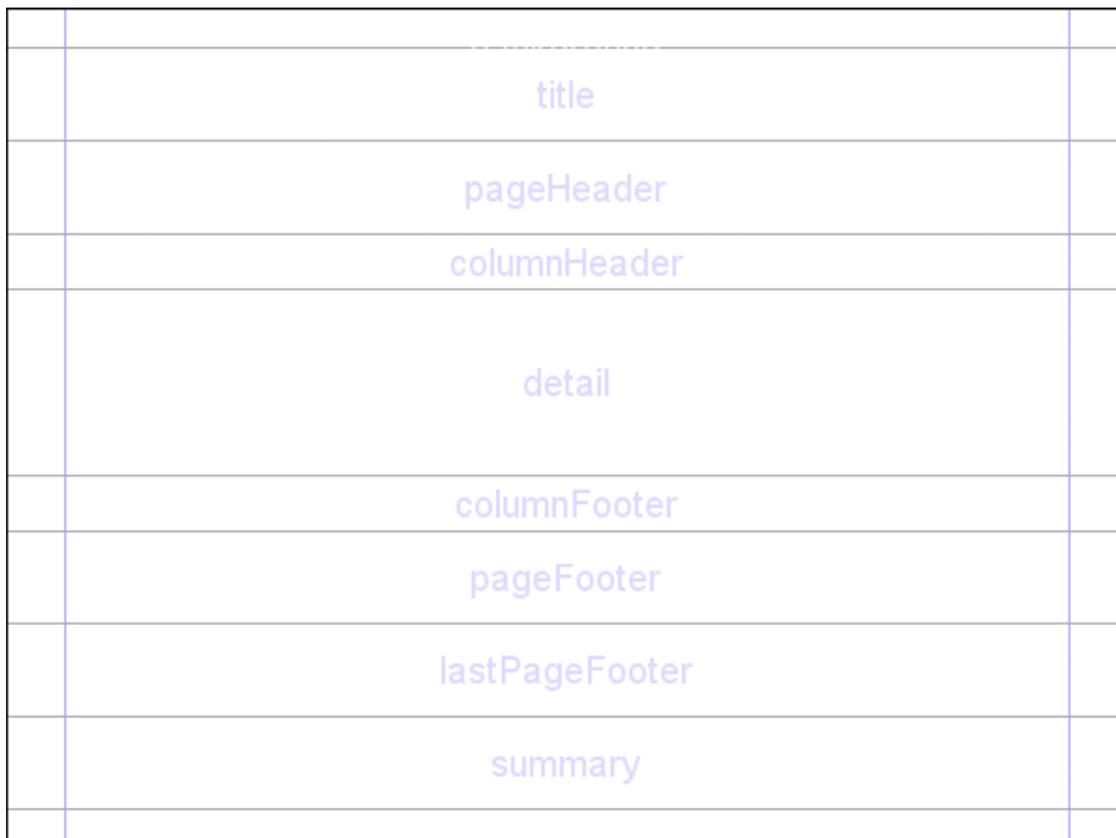


Figura 1 – Bands de um documento de relatório em branco no JasperReports

Embora as bands já estejam adicionadas e com espaçamentos pré-definidos inicialmente, você pode com o cursor, entre cada uma delas, nas linhas horizontais que as separam, e movê-las.

Ao arrastar, você percebe que a altura de cada band se expande, empurrando as demais, após a alterada, para baixo ou vice-versa. Embora o desenvolvedor possa alterar a sua altura, em casos como a band **detail**, sua alteração implica na visibilidade e espaçamento dos elementos impressos na geração do relatório. Porém, nem todas as bands são organizadas dinamicamente pelo conteúdo, como é o caso de **Column Footer**, **Page Footer** e **Last Page Footer**.

A seguir você tem a descrição de cada uma das bandas existentes em um relatório JasperReports:

Title

A band **title**. É a primeira band visível na construção de um relatório vazio. A seção de título. Esta seção só aparece uma vez no começo do relatório. Ex.: Relatório de Clientes.

No JasperReports é representado pelo trecho XML mostrado a seguir:

```
<title>
  <band height="50">
    <staticText>
      <reportElement x="180" y="0" width="200" height="20"/>
      <text>
```

```
<![CDATA[Title]]>
  </text>
</staticText>
</band>
</title>
```

Page Header

A band **pageHeader**. Esta seção aparece no começo de cada página impressa. A altura especificada durante a fase de desenho do relatório não se altera durante o processo de criação do mesmo (exceto que haja componentes redimensionáveis, como uma caixa de texto contendo um texto longo ou um sub-relatório). Neste local você pode ter data e hora e/ou nome ou informações da empresa. As bands Title e Summary não são inclusas no cabeçalho da página quando impressa em uma página separada.

No JasperReports é representado pelo trecho XML mostrado a seguir:

```
<pageHeader>
  <band height="20">
    <staticText>
      <reportElement x="180" y="30" width="200" height="20"/>
      <text>
        <![CDATA[Page Header]]>
      </text>
    </staticText>
  </band>
</pageHeader>
```

Column Header

A band da **columnHeader**. Esta seção só aparece no começo de cada interação com a band **detail**. Se o relatório tem uma só coluna definida, formando um relatório tabular, então são ignoradas as seções cabeçalhos e rodapés de coluna. Como exemplo você pode listar nomes escolhidos de campos específicos como: Nome do Cliente, Telefone, Data do Cadastro.

No JasperReports é representado pelo trecho XML mostrado a seguir:

```
<columnHeader>
  <band height="20">
    <staticText>
```

```
<reportElement x="180" y="50" width="200" height="20"/>
<text>
  <![CDATA[Column Header]]>
</text>
</staticText>
</band>
</columnHeader>
```

Detail

A band **detail** é o local de exibição dos dados de um objeto datasource ou query. Esta seção se repete enquanto houver linhas para serem colocadas no seu relatório de acordo com o resultado transmitido.

No JasperReports é representado pelo trecho XML mostrado a seguir:

```
<detail>
<band height="20">
  <textField>
    <reportElement x="10" y="0" width="600" height="20"/>
    <textFieldExpression class="java.lang.String">
      <![CDATA[${FieldName}]]>
    </textFieldExpression>
  </textField>
</band>
</detail>
```

Column Footer

A band **columnFooter**. Esta seção aparece abaixo de cada coluna. Vale o mesmo comentário da banda Column Header.

No JasperReports é representado pelo trecho XML mostrado a seguir:

```
<columnFooter>
<band height="20">
  <staticText>
    <reportElement x="0" y="0" width="200" height="20"/>
    <text>
      <![CDATA[Column Footer]]>
    </text>
  </staticText>
</band>
</columnFooter>
```

```
</text>
</staticText>
</band>
</columnFooter>
```

Page Footer

A band **pageFooter**. Representa o rodapé da página. Esta seção aparece no final de cada página. Esta band pode conter informações como o número da página, o total de páginas encontradas, endereço da empresa e etc.

No JasperReports é representado pelo trecho XML mostrado a seguir:

```
<pageFooter>
<band height="20">
<staticText>
<reportElement x="0" y="5" width="200" height="20"/>
<text>
<![CDATA[Page Footer]]>
</text>
</staticText>
</band>
</pageFooter>
```

Last Page Footer

A band **lastPageFooter**. A última seção do rodapé da página. Esta seção substitui o rodapé da página regular na última página do relatório. Se você quer que o último rodapé da página seja diferente dos demais, isto é possível usando esta band especial. Se esta band possuir o valor **zero** de altura, será completamente ignorada e o layout desenhado para as páginas comuns será usado também para a última página.

No JasperReports é representado pelo trecho XML mostrado a seguir:

```
<lastPageFooter>
<band height="20">
<staticText>
<reportElement x="0" y="5"
width="200" height="20"/>
<text>
```

```
<![CDATA[Last Page Footer]]>
</text>
</staticText>
</band>
</lastPageFooter>
```

Summary

A band **summary**. A seção de sumário, conhecido em outros sistemas como Report footer (Rodapé do relatório). Esta seção só aparece uma vez ao término do relatório e permite você adicionar um campo possuindo o total geral, por exemplo, de todo o relatório ou qualquer outra coisa que deseje no fim deste (como um gráfico).

No JasperReports é representado pelo trecho XML mostrado a seguir:

```
<summary>
<band height="20">
<staticText>
<reportElement x="0" y="5" width="200" height="20"/>
<text>
<![CDATA[Summary]]>
</text>
</staticText>
</band>
</summary>
```

Background

A band **background**. Esta seção possibilita a introdução de marcas d'água em um relatório, ou qualquer outra coisa de efeito similar (como linhas ao redor da página). Por padrão ela está com a altura em valor zero no iReport.

No JasperReports é representado pelo trecho XML mostrado a seguir:

```
<background>
<band height="745">
<image scaleImage="Clip" hAlign="Left" vAlign="Bottom">
<reportElement x="0" y="0" width="160" height="745"/>
<imageExpression>"image.gif"</imageExpression>
</image>
</band>
```

</background>

noData

A band **noData** foi introduzida no JasperReports 2.0.0 com o intuito de introduzir uma seção para melhorar o controle sobre relatórios e sub-relatórios preenchidos com fontes de dados (data sources) vazias.

No JasperReports é representado pelo trecho XML mostrado a seguir:

<noData>

```
<band height="5" isSplitAllowed="true" >
```

```
</band>
```

</noData>

Compreendendo as propriedades de um relatório

Antes de criar um relatório, e compreender as bands explicadas, você precisa aprender as configurações de um relatório através de suas propriedades iniciais, começando com um relatório sem o uso de assistentes.

Iniciando com um relatório em branco

Para criar um relatório em branco, vá ao menu **File** e clique no item **New Document**, ou utilize o ícone na barra de ferramentas principal **New Document** (atalho **Ctrl+N**).

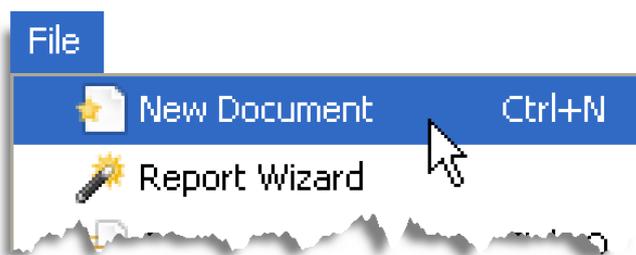


Figura 2 – Detalhe na criação de um Novo relatório em branco

Ao surgir à caixa de diálogo **Report properties**, você pode iniciar colocando o **Report Name**. Na área **Page size** você define o tamanho da folha que comportará seu relatório. Como qualquer outro editor de relatórios, você possui alguns **Preset Sizes** padronizados e a possibilidade de criar um personalizado, alterando a **Width** e **Height** da página, incluindo a unidade de medida. Em **Orientation** você pode colocar a página do seu relatório em **Portrait** (Retrato) ou **Landscape** (Paisagem).

Embora as dimensões sejam as mais importantes para a construção de um relatório, a unidade de medidas usada pelo iReport e JasperReports é em pixels (com uma resolução de 75 dpi), como você já deve ter concluído, uma vez que no Capítulo 2 foram mostradas as configurações onde a unidade do grid (Grid Unit) estava em pixels.

Para que você possa se orientar em um arquivo JRXML do JasperReports, a seguir a **Tabela 1** mostra o padrão de algumas páginas e seu valor em pixels.

Tabela 1 – Dimensões em pixels das páginas padronizadas

Padrão da página	Dimensão em Pixels
LETTER	612 x 792
A3	842 x 1190
A4	595 x 842

Atenção: A lista completa se encontra no livro **Dominando Relatórios JasperReports com iReport**.

A guia Margem da Página

Na guia **Page Margin**, você possui quatro posições que definem as margens de seu relatório: **Top**, **Bottom**, **Left** e **Right**. As margens podem ser personalizadas seguindo o padrão necessário para o desenvolvimento do seu relatório.

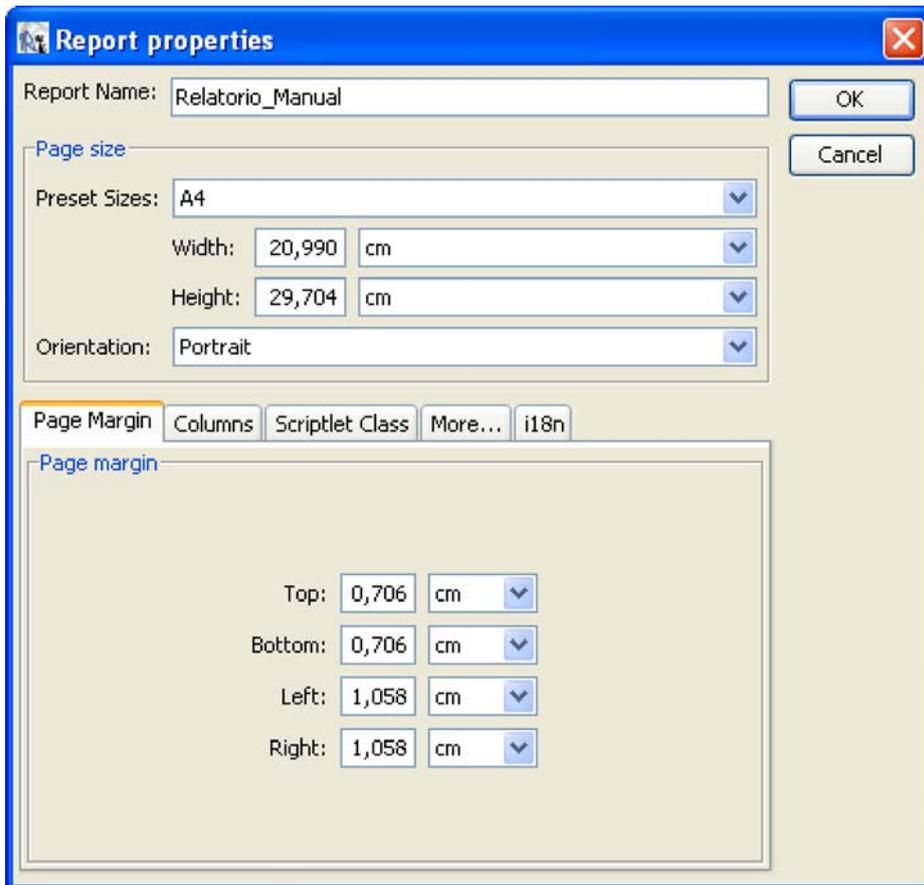


Figura 3 – Propriedades do relatório

A guia Columns

Na guia **Columns** você pode definir quantas colunas seu relatório possuirá em uma página. A página que compõe o relatório possui seções que são independentes de dados vindos de um datasource ou query (como o rodapé da página), mas possui em outras bands a possibilidade de criar colunas para exibir os dados que são preenchidos dinamicamente. Estas bands podem ser postas em colunas verticais definindo uma divisão de acordo com a largura da página.

O padrão é uma coluna, onde a sua largura compreende da margem esquerda à margem direita da página.

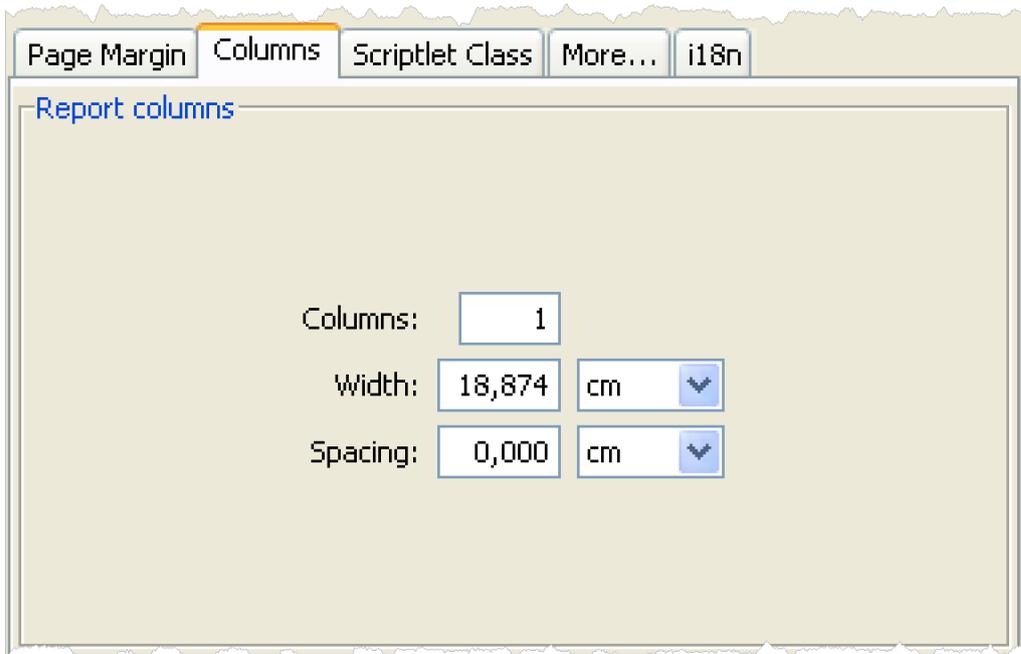
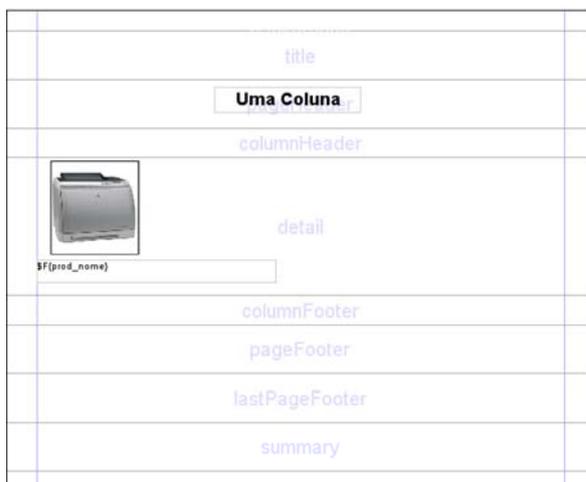


Figura 4 – Detalhe da guia Columns

Caso você execute um relatório contendo apenas uma coluna, os elementos vindos da fonte de dados serão postos um abaixo do outro em sua ordem, como mostra a **Figura 3.5** a seguir:

Na construção do relatório



Com o relatório gerado



Figura 5 – Relatório de uma coluna

Caso você adicione duas colunas, por exemplo, a largura máxima é dividida automaticamente pelo iReport. Obviamente, você também pode definir um espaçamento entre colunas. Ao adicionar o espaçamento, a largura da coluna é subtraída automaticamente, evitando assim problemas na criação do relatório.

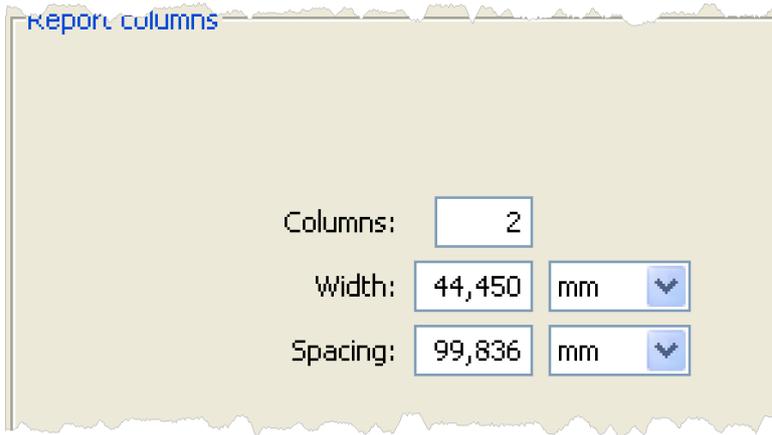


Figura 6 – Detalhe na adição de duas colunas na página

É interessante notar que as bandas **columnHeader** e **columnFooter** também ficam colunadas.

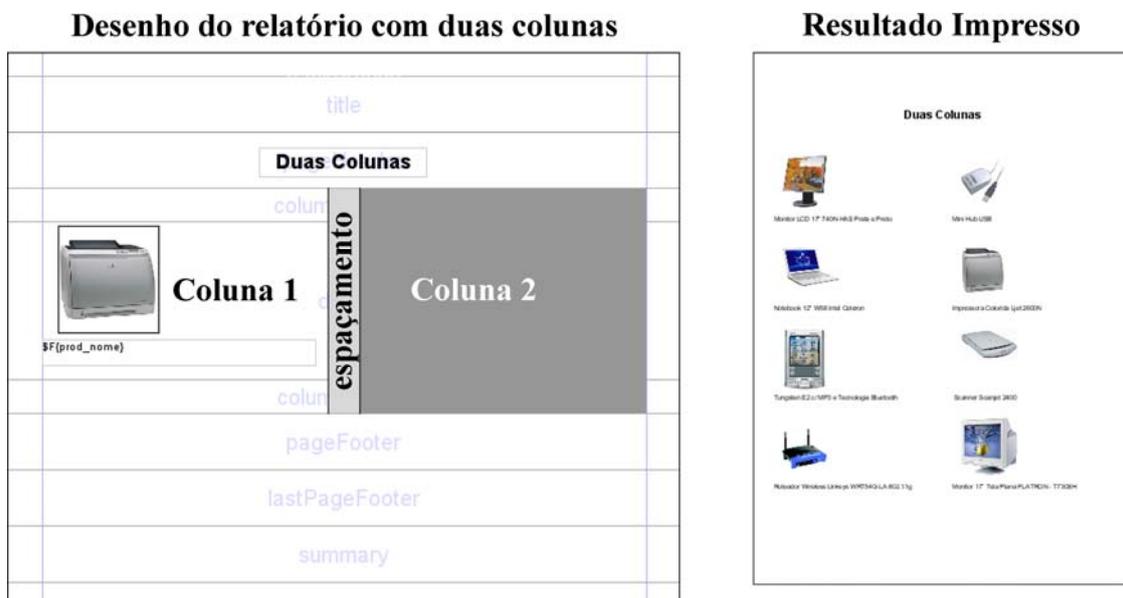


Figura 7 – Criação de um relatório com duas colunas

A guia Scriptlet Class

Nesta guia você define a utilização de um scriptlet customizado ou não. O scriptlet é uma classe Java cujos métodos são executados conforme são especificados os eventos durante a criação do relatório, como o início de uma nova página ou o fim de um grupo - **Mais adiante trataremos deste assunto**. Além desta opção, você também possui a linguagem (Language) que será usada para a ferramenta de expressões do iReport. Somente **Java** e **Groovy** são suportados nesta versão. É importante ressaltar que esta linguagem não está relacionada à classe scriptlet.

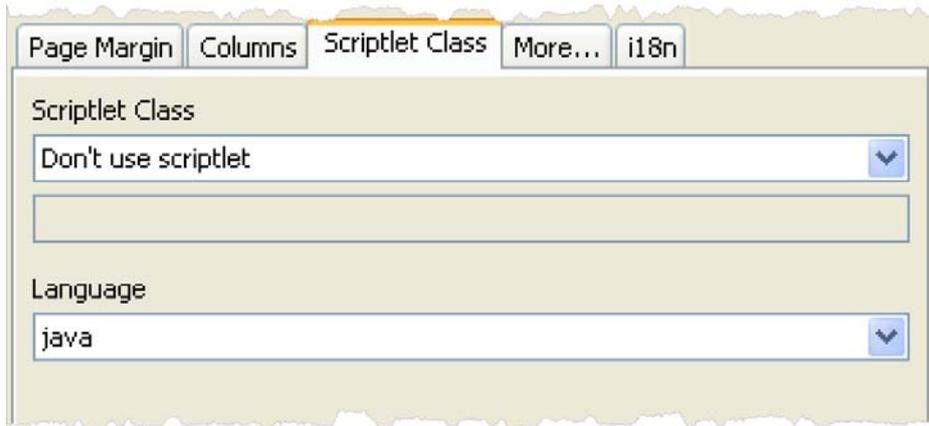


Figura 8 – Detalhe na guia Classe scriptlet

A guia More...

Na guia “More...” é possível especificar instruções para a impressão do relatório.

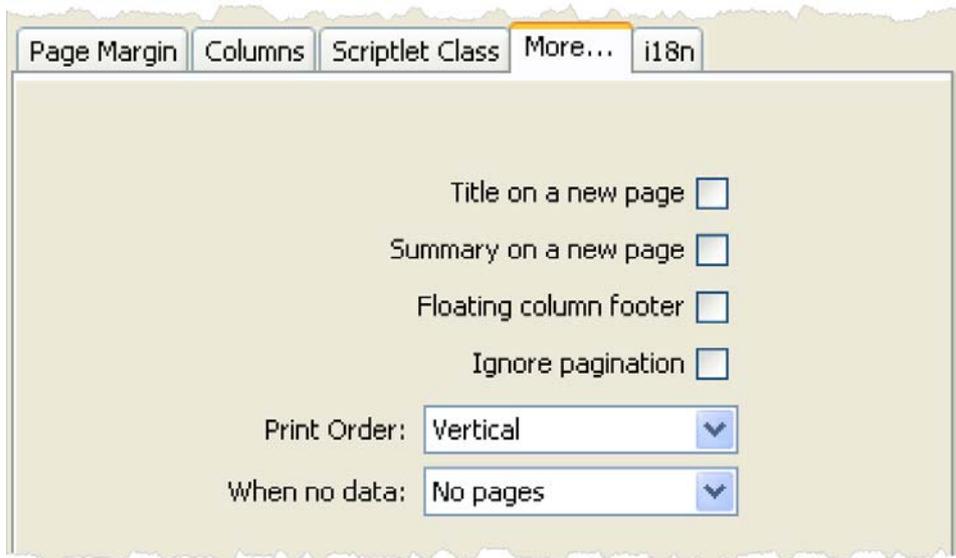


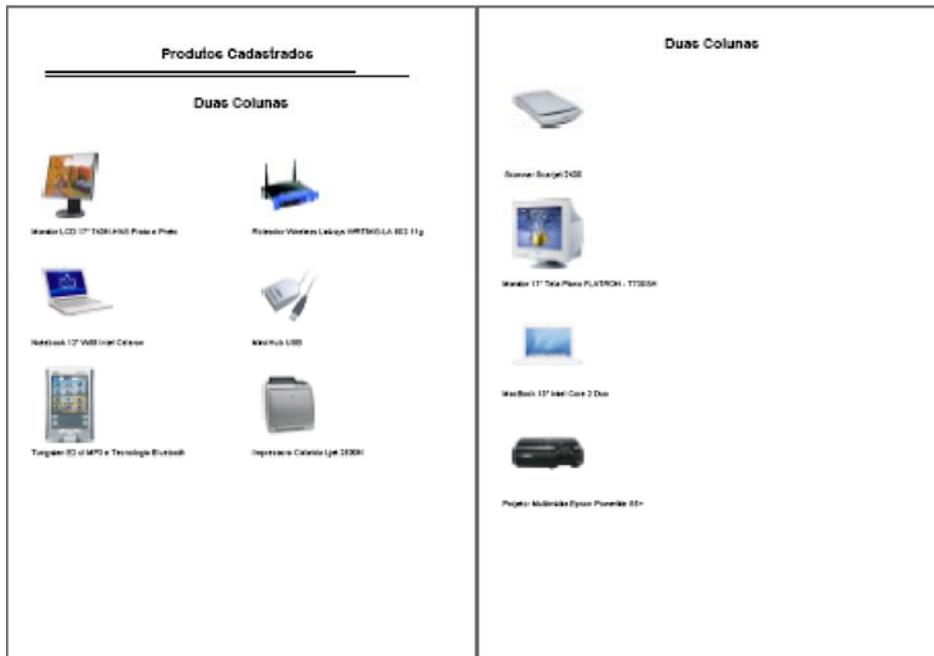
Figura 9 - Detalhe na guia Mais...

Na parte inicial desta guia, você possui quatro checkboxes e dois combos:

Title on a new page

Quando ativa, especifica se a banda **title** será impressa e em seguida criada uma nova página, gerando uma “quebra de página”. Seu comportamento pode ser visualizado na **Figura 10** a seguir.

Padrão com a banda title preenchida



Título a cada nova página selecionado

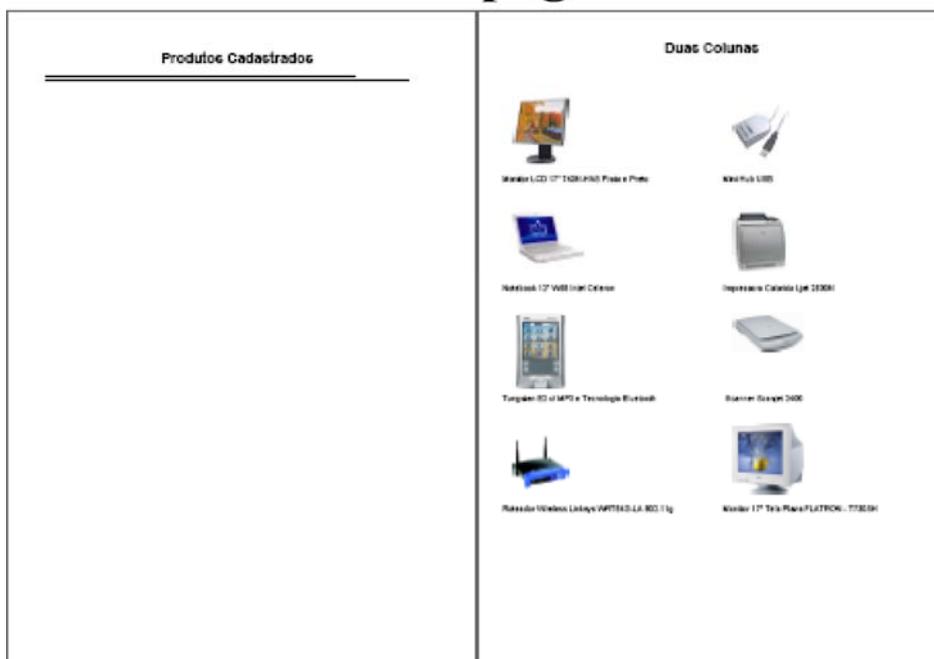


Figura 10 – Comportamento de Title on a new page

Summary on a new page

Esta opção é similar a anterior (**Title on a new page**), mas com a diferença de ser a banda **summary** que será impressa como a última página.

Floating column footer

As bandas **columnFooter** são impressas por padrão no fim das páginas. Se na impressão do relatório não houver dados suficientes para preencher uma página, um espaço em branco aparecerá entre a coluna e o rodapé da coluna (**columnFooter**). Se você quer que o texto encontrado na banda **columnFooter** seja impresso logo após a impressão da coluna, marque a opção **Floating column footer**. Esta opção permite forçar a impressão da banda **columnFooter** imediatamente após a última interação da banda **detail** e não do término de todas as colunas. Esta opção é geralmente usada quando você quer criar tabelas usando os elementos de relatórios.

Ignore pagination

Ignora a paginação ao preencher o relatório, impossibilitando a visualização das demais páginas caso haja mais que uma. Quando ativa, esta opção implica até mesmo na visualização correta dos dados caso haja mais de uma coluna.

Print Order

A ordem de impressão (Print Order) determina como os dados impressos na página serão organizados em mais de uma coluna. O padrão para a ordem de impressão é **Vertical**.

Com a ordem **Vertical**, primeiro os dados são impressos um embaixo do outro, passando para uma nova coluna somente quando a coluna está totalmente preenchida no final da página.

Na ordem **Horizontal** os dados são preenchidos em linha. Os dados passam por todas as colunas, adicionando uma nova linha somente quando a última coluna da página foi preenchida naquela determinada linha. A **Figura 11** a seguir ilustra as duas ordens de impressão.



Figura 11 – Ordem de impressão na Vertical e na Horizontal

When no data

Quando o número de dados para a impressão é vazio (query SQL sem dados, por exemplo), o relatório pode ter comportamentos diferentes para apresentar a página ao usuário. Neste caso existem quatro opções, listadas a seguir:

- **No pages (Sem Páginas)**– O padrão. O resultado final é um buffer vazio.

- **Blank page (Páginas em branco)** – Exibe uma página em branco caso não haja dados.
- **All sections, no detail (Todas as seções, nenhum detalhe)** – Mostra todas as seções existentes no relatório, menos os detalhes da banda **detail**.
- **No-data section** – Seção sem dados. Utiliza a band **noData**, ou seja, você adiciona um conteúdo estático (ou dinâmico sem usar o banco de dados) neste local. Caso não haja dados em uma determinada pesquisa ou sem conexão com o banco de dados, esta seção será exibida.

A guia i18n

A guia **i18n** é a responsável pelos recursos de internacionalização de seus relatórios.

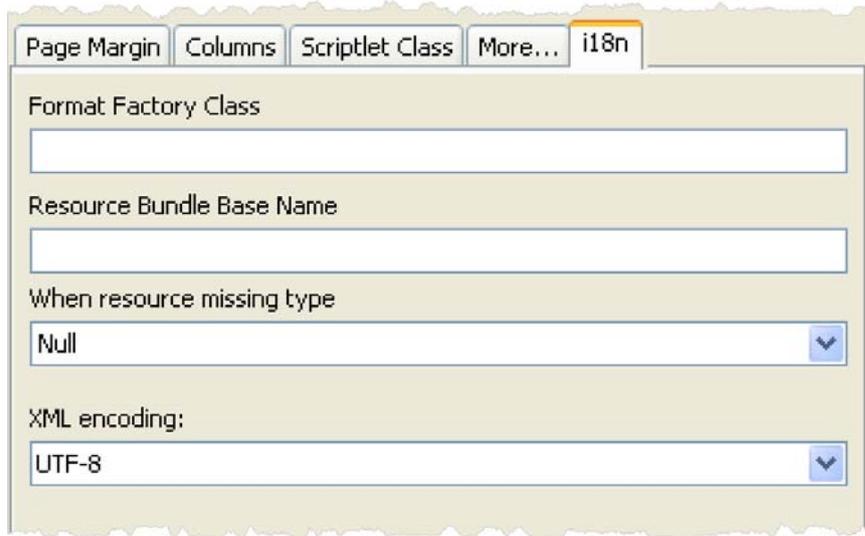


Figura 12 – Detalhe da guia i18n

Format Factory Class

Recebe uma string com o nome da classe de formatos, habilitando o uso de datas personalizadas e formatos de números. Representa o atributo **formatFactoryClass** de JasperReports e implementa a interface *net.sf.jasperreports.engine*.

Resource Bundle Base Name

É o nome do pacote de recursos usado para internacionalizar um relatório. É o Resource Bundle do Java, onde você define em um arquivo os textos traduzidos como rótulos. Cada linguagem corresponde a um arquivo específico. Veremos detalhes sobre internacionalização de relatórios em um capítulo posterior.

When resource missing type

Se um pacote de recurso (Resource Bundle) não está disponível, você escolhe uma das opções na caixa de combinação **Quando está faltando o tipo do recurso**. Representa o atributo **whenResourceMissingType** no JasperReports. Neste caso, você tem as seguintes opções:

- **Null (Nulo)** – Imprime a string “*Null*”. Esta é a opção padrão.

- **Empty (Vazio)** – Não imprime nada. Recebe como valor *Empty* no atributo *whenResourceMissingType* de JasperReports.
- **Key (Chave)** – Imprime o nome chave que falta. Recebe como valor *Key* no atributo *whenResourceMissingType* de JasperReports.
- **Error (Erro)** – Lança uma exceção parando o processo. Recebe como valor *Error* no atributo *whenResourceMissingType* de JasperReports.

XML encoding

O formato padrão para salvar os arquivos fontes do JasperReports no iReport é UTF-8. Em todo caso, se você desejar utilizar outro formato que não seja UTF-8, é neste local que deve ser especificado. O UTF-8 gerencia todas as letras acentuadas e o símbolo de euro. A outra codificação comum, ISO-8859-1, se encontra nesta caixa de combinação.

Curiosidade: Para construir um relatório com o processo de adaptação para vários idiomas e regiões, sem qualquer esforço de desenvolvimento adicional, utilizamos a internacionalização ou, **i18n** (“**internationalization**” é uma palavra de 20 caracteres que começa com a letra “**i**” e termina com “**n**”, tendo 18 letras entre elas).

Visualização do arquivo JRXML pelo iReport

Através do iReport é possível ver o arquivo XML gerado para o JasperReports. Vá ao menu **Edit** e clique no item **XML Source**. O visualizador será o definido em sua máquina para o formato (isso na versão atual em que este livro está sendo baseado).

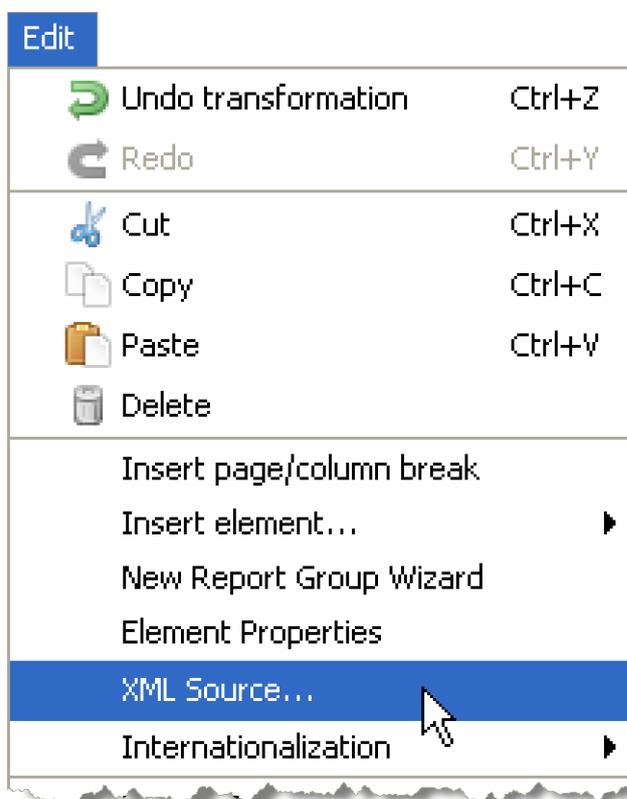


Figura 13 – Visualizando o arquivo JRXML pelo iReport

Capítulo 4

“A imaginação é mais poderosa que o conhecimento. Ela alarga a visão, estimula a inteligência, desafia o impossível. Sem a imaginação, o pensamento estagnará.”

Albert Einstein

Criando um relatório sem o uso de assistentes

Para uma maior compreensão dos diversos elementos que compõem um relatório no JasperReports, através do iReport, é importante a um desenvolvedor aprender a desenhá-lo manualmente, isto é, sem o uso de assistentes.

No capítulo anterior foram apresentadas as bandas e as configurações iniciais de um relatório em branco através de suas propriedades. Neste capítulo você irá aprender a criar um relatório manualmente, adicionando diversos elementos, configurando-os e trabalhando com banco de dados.

Iniciando um relatório manualmente

Para iniciar, você vai criar um **New Document (Ctrl+N)**. Dê um nome para o seu relatório. Os tamanhos predefinidos serão mantidos. Clique no botão **OK**.

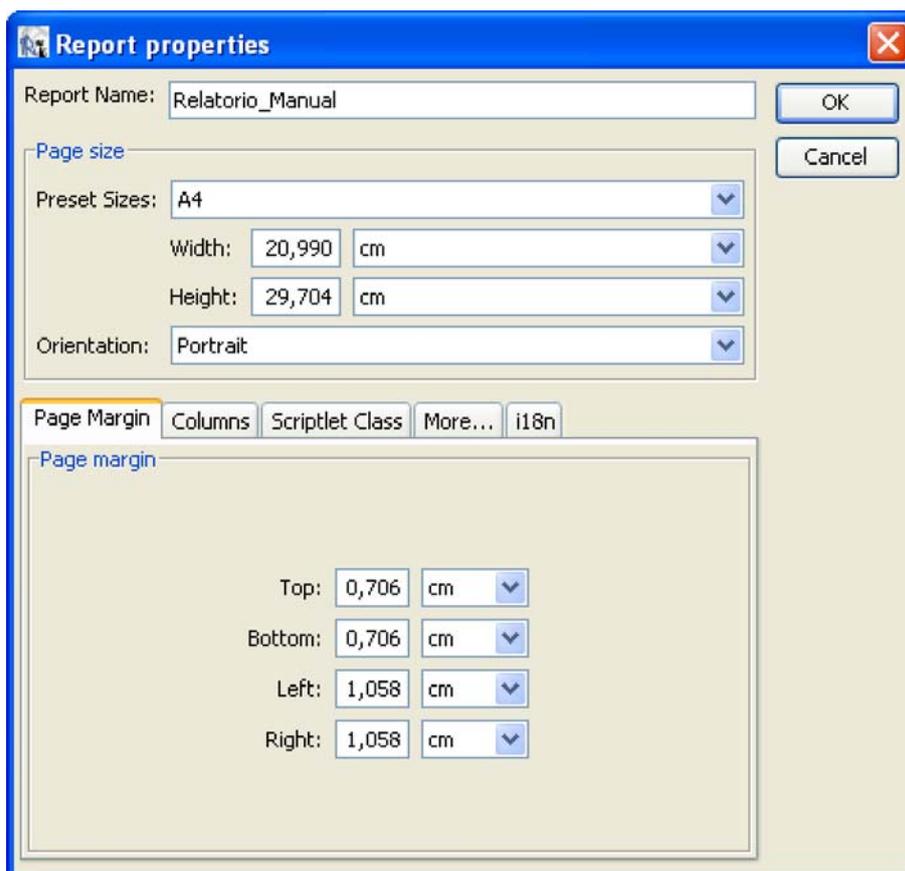


Figura 1 – Propriedades do relatório

Exibindo o Grid para desenho de relatórios

Caso você queira desenhar o Grid para desenhar o relatório, vá ao menu **View** e clique em **Show Grid**.



Figura 2 –Exibindo o Grid no Design do relatório

A configuração para o **Grid Size (Tamanho do Grid)** é visto na guia **General** da caixa de diálogo **Options**, encontrado no menu **Options** (explicado no Capítulo 2 deste livro, em **Configurando o iReport**).

As barras de ferramentas do iReport

O iReport possui três barras de ferramentas na parte superior de sua janela: a **Principal**, a **Text Formatting** (Formatação de Texto) e **Elements Formatting** (Formatar Elementos). Na primeira barra de ferramentas, a **Principal**, existe doze elementos que são muito usados na construção de um relatório. Estes ditos “*elementos*” são objetos gráficos que podem compor um relatório, como uma linha, rótulo ou uma Elipse.

A **Figura 3** apresenta as principais ferramentas que podem ser usadas tanto para a criação de elementos, como também para sua manipulação, em um relatório JasperReports.

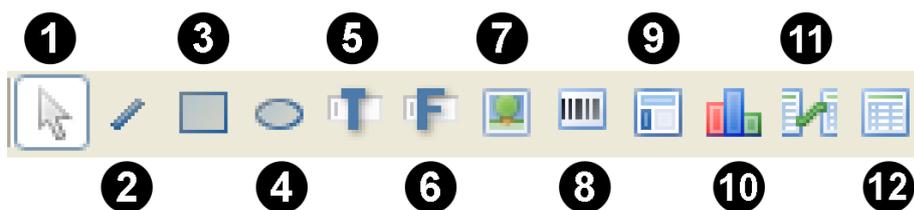


Figura 3 - Ferramentas para a criação de elementos no relatório

1. **Pointer** – A ferramenta Pointer é usada para as operações mais importantes no desenho de um elemento, como: selecionar, arrastar, dimensionar e etc. Por padrão é ativa, mas fica desativada quando outra ferramenta de desenho de elementos é selecionada.
2. **Line** – Em um relatório JasperReports uma linha é definida por um elemento retangular com uma linha diagonal. A linha pode ser desenhada em uma reta perfeita, na horizontal, vertical ou em ângulo se pressionada a tecla <Shift> antes de arrastar. A linha segue o grid.

3. **Rectangle** – Usado para desenhar retângulos ao redor de outros elementos, criando destaques como uma formatação de parágrafo de um programa de edição de textos, este elemento pode também conter retângulos com cantos arredondados.
4. **Ellipse** – Desenha elipses no relatório.
5. **Static Text** – Utilizado para criar rótulos de relatórios.
6. **Text Field** – Utilizado para criar os campos dinâmicos dos relatórios. É neste elemento que você se conecta a um determinado campo do banco de dados para exibir suas informações, por exemplo.
7. **Image** – Utilizado para exibir imagens em seu relatório. Podem ser imagens dinâmicas (preenchidas por um banco de dados, por exemplo) ou estáticas.
8. **Barcode tool** – Gera um código de barras baseado em uma string.
9. **Frame** – Cria quadros que podem conter outros elementos em seu interior, com bordas ao seu redor.
10. **Chart tool** – Gera um gráfico através de valores determinados transmitidos. Mais adiante haverá um capítulo exclusivamente para gráficos.
11. **Subreport** – Cria um relatório dentro de outro, o que chamamos de mestre-detalhe. Mais adiante haverá um capítulo abordando exclusivamente este assunto.
12. **Crosstab** – Desenvolve o que chamamos em programas de planilhas eletrônicas de tabela dinâmica (referência cruzada), muito utilizado em relatórios gerenciais (Business Intelligence). Mais adiante também haverá um capítulo abordando exclusivamente este assunto.

Adicionando uma conexão a um relatório em branco

Para a construção deste relatório, vamos utilizar o banco de dados do livro. O exemplo seguirá do pressuposto que você ainda não possui uma conexão criada com o iReport. Vá ao menu **Data** e clique em **Connections/Data Sources**, ou na barra de ferramentas **Text Formatting**, clique no botão **Datasources**, como mostra o detalhe da **Figura 4**.



Figura 4 – Criando uma nova conexão com o banco de dados

Na caixa de diálogo **Connections / Datasources**, clique no botão **New**.

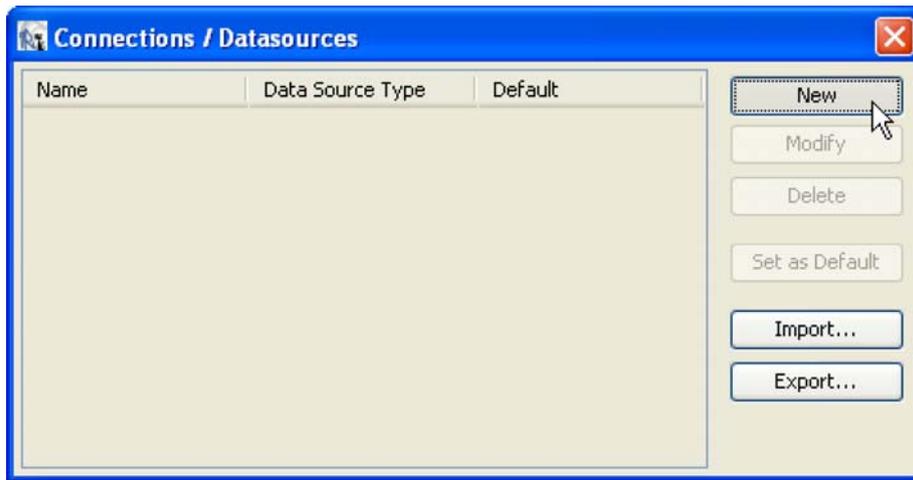


Figura 5 – Criando uma nova conexão com o banco de dados

Através dos passos seguidos no **Capítulo 2**, através da caixa de diálogo **Connections properties**, você vai utilizar a ponte JDBC, selecionando o item **Database JDBC connection** e clicar no botão **Next**. Mas desta vez você vai utilizar o assistente, **JDBC URL Wizard**, digitando **localhost:3306** no campo **Server Address** e em **Database** digite **relatorios**. Clique no botão **Wizard**. Perceba que o campo **JDBC URL** será preenchido com os dados digitados. **Teste** a conexão e **Save**.

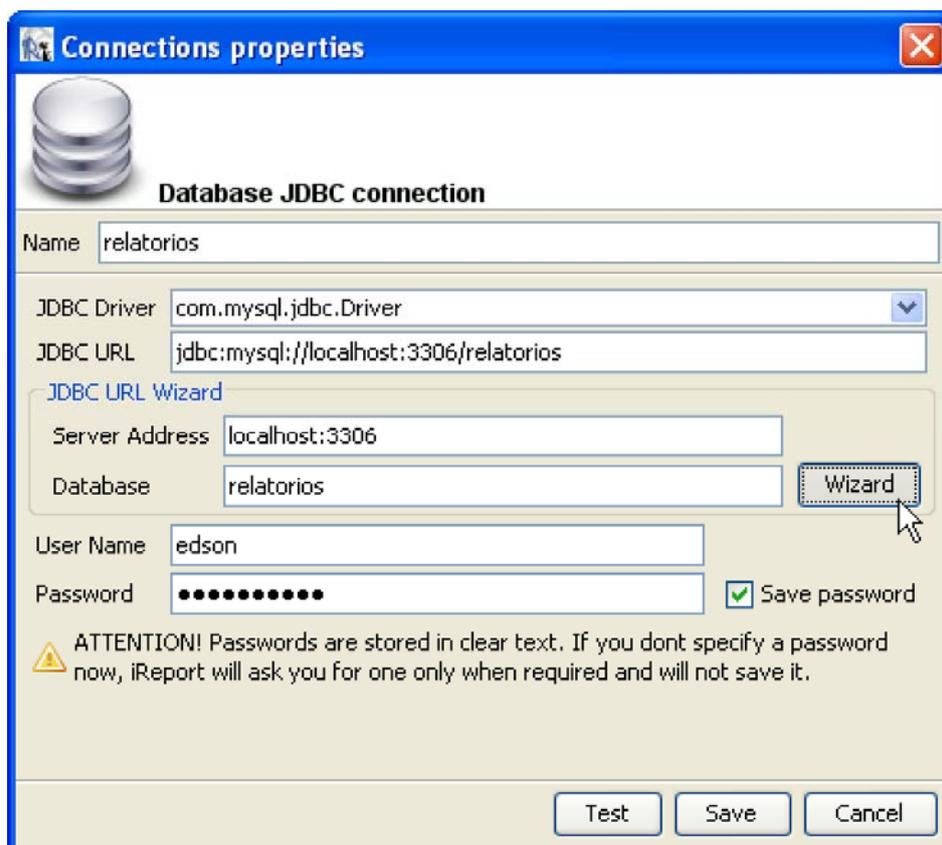


Figura 6 – Usando o Assistente para JDBC URL

Para uma modificação da conexão, na caixa de diálogo **Connections / Datasources** temos o botão **Modify**. Basta selecionar a linha da sua conexão que deseja alterar e clicar neste botão. A caixa de diálogo **Connections properties** reabre. Basta alterar e salvar (Save).

Em mais de uma conexão, você tem o botão **Set as Default**. Basta selecionar a conexão que deseja tornar padrão (Default) e clicar neste botão. Observe que na caixa de diálogo a coluna Default apresenta a conexão padrão com uma caixa de checagem ativa.

Caso você precise levar as configurações da sua conexão para outra máquina, ou quer guardá-la para uma futura reutilização, clique no botão **Export**. A caixa de diálogo para salvar a conexão aparecerá. Basta selecionar o local e dar o nome. Caso tenha mais de uma conexão, todas serão exportadas.

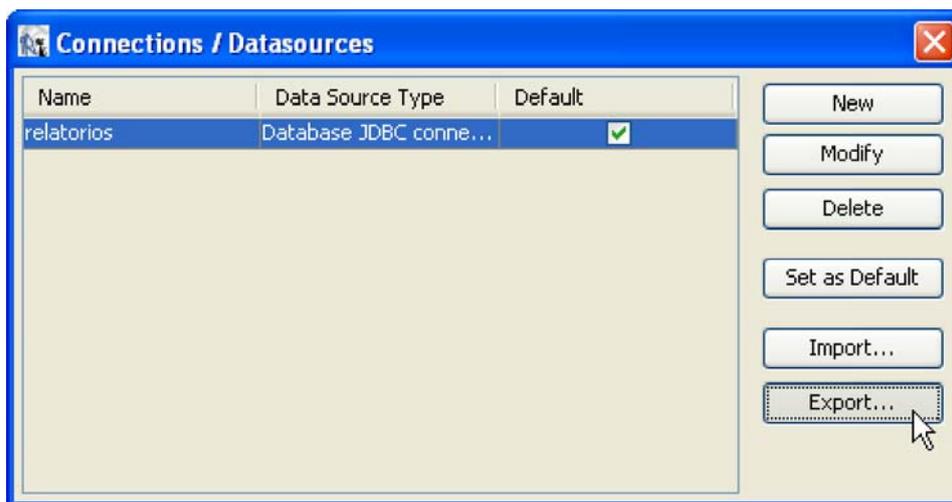


Figura 7 – Exportando a configuração de Connections / Datasources

O arquivo exportado é um XML aberto, no seguinte formato:

```
<?xml version="1.0"?>
<!-- iReport connections -->
<iReportConnectionSet>
  <iReportConnection name="relatorios"
connectionClass="it.businesslogic.ireport.connection.JDBCConnection">
    <connectionParameter name="ServerAddress">
      <![CDATA[localhost:3306]]>
    </connectionParameter>
    <connectionParameter name="SavePassword">
      <![CDATA[true]]>
    </connectionParameter>
    <connectionParameter name="Url">
      <![CDATA[jdbc:mysql://localhost:3306/relatorios]]>
    </connectionParameter>
  </iReportConnection>
</iReportConnectionSet>
```

```
<connectionParameter name="JDBCDriver">
    <![CDATA[com.mysql.jdbc.Driver]]>
</connectionParameter>
<connectionParameter name="Database">
    <![CDATA[relatorios]]>
</connectionParameter>
<connectionParameter name="Password">
    <![CDATA[integrator]]>
</connectionParameter>
<connectionParameter name="Username">
    <![CDATA[edson]]>
</connectionParameter>
</iReportConnection>
</iReportConnectionSet>
```

Isso significa que caso precise alterá-lo, antes de **Importar**, você pode modificar em um simples bloco de notas.

Depois de configurado, basta fechar a caixa de diálogo. A conexão padrão agora é exibida na caixa de seleção (**Figura 8**) na barra de ferramentas **Text Formatting**. É neste local também que você pode alternar entre outras conexões, caso necessite.



Figura 8 – Exibição da conexão ativa

Criando a query do relatório

O primeiro relatório que você vai criar manualmente trará informações dos produtos existentes no banco de dados, similar ao Wizard feito anteriormente no Capítulo 2.

Para criar a query do relatório, vá ao menu **Edit** e clique no item **Report Query**. Alternativamente você pode ir ao menu **Data** e clicar em **Report Query**.

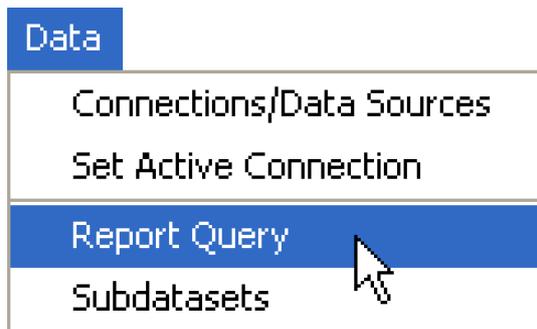


Figura 9 – Acessando Report Query através do menu Data

Ao abrir a caixa de diálogo **Report Query**, clique no botão **Query designer**. Outra caixa de diálogo surgirá, chamada de **SQL Query Designer**, mas desta vez para a criação da instrução SQL visual. Observe que há três quadros na janela. O primeiro, à esquerda, define a exibição da query de forma visual, separando os elementos padronizados de uma seleção de dados (SELECT, FROM, WHERE e etc.). A cada coluna de sua tabela selecionada, automaticamente a query adicionará na parte superior, em SELECT, o campo e um alias. Em FROM surgirá a, ou, as tabelas que estão relacionadas para tal seleção.

Como vamos evoluir a construção do relatório, primeiramente você vai criar um simples relatório contendo apenas uma tabela, a de produtos. Você vai selecionar, através do segundo quadro a esquerda, a tabela **produtos**, dando um duplo clique sobre seu respectivo nome. Esta seleção pode ser através do arrastar para o quadro da direita também, caso ache mais prático.

Existem duas formas de você selecionar todos os campos de uma tabela, no Query designer. A primeira é ir até a tabela (no caso produtos) e clicar no menu de mesmo nome, selecionando a opção **select all**.



Figura 10 – Selecionando todos os campos automaticamente

Ou simplesmente clicando com o direito do mouse e selecionando a opção **add expression**, no quadro da esquerda, em **SELECT**.

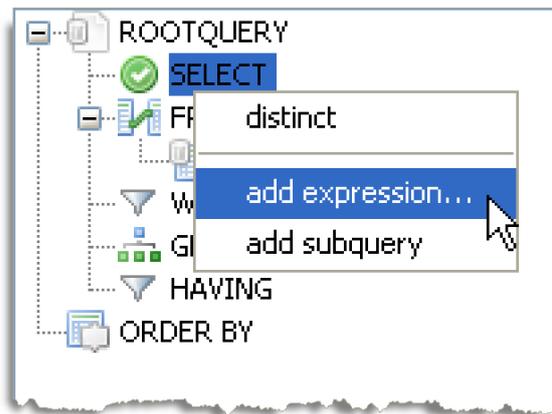


Figura 11 – Adicionando uma expressão

Neste caso, você digita o que deseja fazer, na caixa de diálogo **expression.edit**.

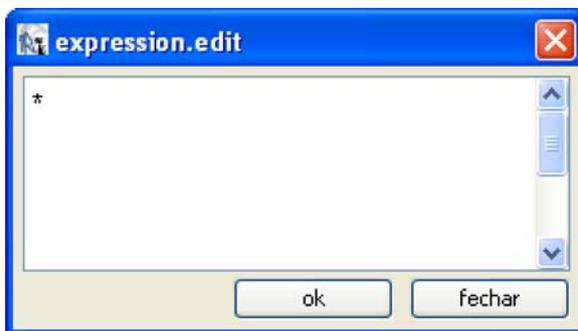


Figura 12 – Criando a expressão para a query SQL SELECT

Ao final, você possuirá uma configuração da sua query SELECT similar ao mostrado na **Figura 13** a seguir:

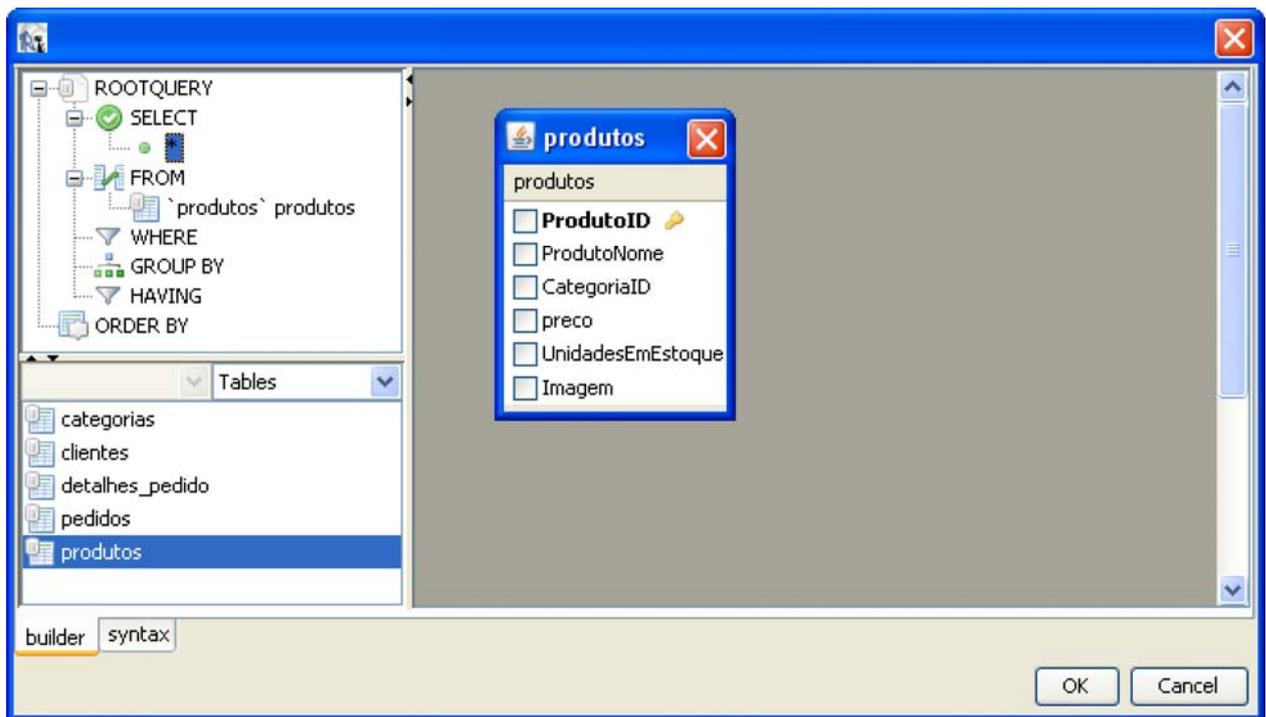


Figura 13 – Query final gerada na caixa de diálogo SQL Query Designer

Se você clicar em **syntax**, a guia abaixo, ao lado de **builder** (a atual), você verá a **query** construída. Confirme no botão **OK**.

Retornando a caixa de diálogo **Report Query**, você ainda pode fazer alterações na query criada.

Na parte superior da janela, existe o botão **Save query**, caso deseje armazenar a consulta construída para uso posterior em outro relatório. Se esse for o caso, o carregamento de uma consulta armazenada pode ser feito através do botão **Load query**.

Como a opção **Automatically Retrieve Fields** está selecionada, na parte inferior aparece cada campo selecionado, em uma tabela, com seus respectivos nomes e o tipo de dados em Java de cada um. É importante manter selecionada esta opção, uma vez que na digitação ou alteração de uma query, possíveis erros podem ser encontrados graças a não exibição dos campos nesta tabela.

Sem esta opção selecionada, qualquer alteração na query SQL, não reflete automaticamente neste local, o que se faz necessário clicar no botão **Read Fields**.

Observe que também existe a possibilidade de ter uma descrição de cada campo. Na tabela inferior, nos campos resultantes da query, dê um duplo clique em cada linha. Observe que a caixa de diálogo **Add/modify field** surgirá, com o nome e tipo de classe do campo. E também um campo de **descrição** em **Field Description**. Se houver necessidade, para melhor esclarecimento sobre o nome do campo encontrado na query, quanto a respeito do que ele armazena, basta digitar sua descrição e confirmar.

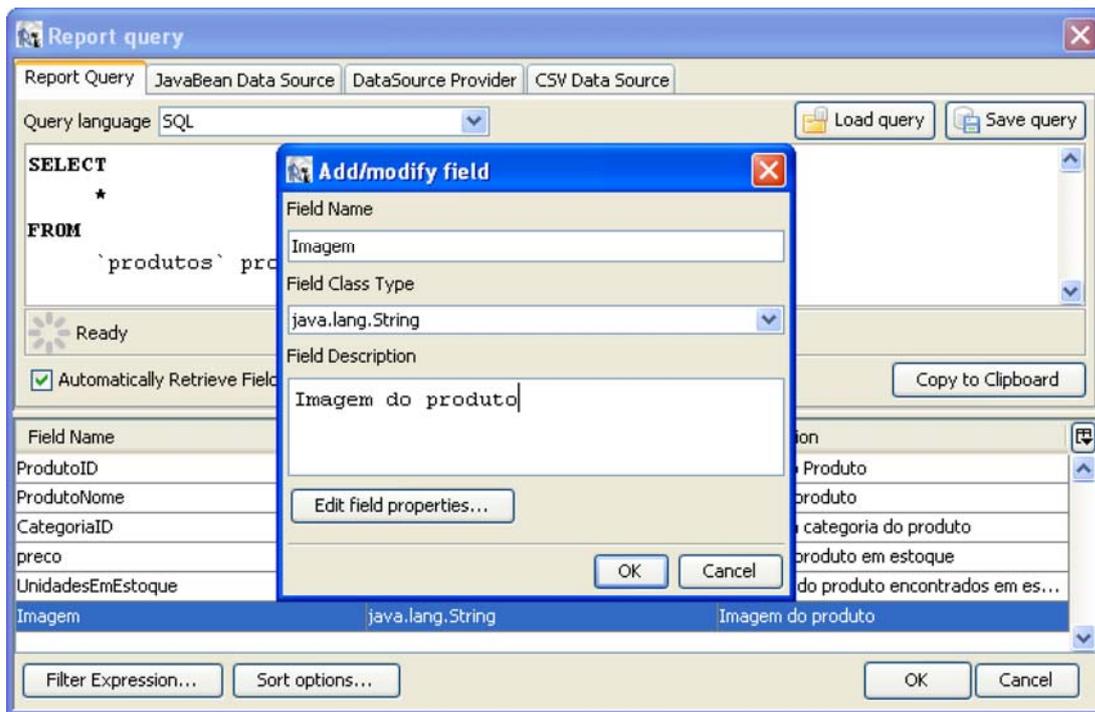


Figura 14 – Adicionando descrição em um campo

Como não vamos utilizar todos os campos descritos, vamos remover os que não serão usados.

Selecione os campos **CategoriaID** e **UnidadesEmEstoque**, com a tecla <Ctrl> pressionada e tecla <DELETE> para removê-los. Caso você erre, clique no botão **Cancel** e através do menu **Data** item **Report Query**, refaça novamente a seleção e a devida exclusão.

Field Name	Field Type	Description
ProdutoID	java.lang.Integer	Código do Produto
ProdutoNome	java.lang.String	Nome do produto
CategoriaID	java.lang.Integer	Código da categoria do produto
preco	java.lang.Double	Preço do produto em estoque
UnidadesEmEstoque	java.lang.Integer	Unidades do produto encontrados em estoque
Imagem	java.lang.String	Imagem do produto

Figura 15 – Detalhe dos campos selecionados da caixa de diálogo Consulta do Relatório

Depois de criado todos estes detalhes, confirme a caixa de diálogo **Report query** clicando no botão **OK**.

Os elementos da query registrados no JasperReports

Ao adicionar query criada, através do iReport, o arquivo JasperReports registrou elemento <queryString /> em seu arquivo jrxml, para armazenar a consulta SQL:

```
<queryString>
  <![CDATA[SELECT * FROM `produtos` produtos]]>
</queryString>
```

Nota: O texto contido por entre os elementos <queryString> possui um XML **CDATA**. Este bloco é ignorado pelo analisador XML. A justificativa está nos caracteres inválidos. Caso estes estivessem fora deste bloco, o XML conteria um erro e não seria analisado corretamente para a criação do relatório compilado criado JasperReports.

Para que seja possível saber quais campos você deseja trabalhar neste relatório, o JasperReports registra no elemento <field />, cada um dos que serão utilizados. Neste elemento, encontra-se em forma de atributos o nome da coluna e também o seu tipo mapeado pela JDBC conforme o driver do banco de dados. Como foram removidos dois campos, somente quatro ficaram registrados:

```
<field name="ProdutoID" class="java.lang.Integer"/>
<field name="ProdutoNome" class="java.lang.String"/>
<field name="preco" class="java.lang.Double"/>
<field name="Imagem" class="java.lang.String"/>
```

Este elemento é importante, pois campos podem ser mesclados em uma expressão do relatório para obter a saída necessária.

Desenhando o relatório

O desenho do relatório será uma etapa importante para a sua compreensão prática dos campos, rótulos, imagens, frames e também sobre as bands (bandas) em sua geração.

A **Figura 16** mostra a janela do iReport com a visibilidade de cinco painéis que facilitam o desenvolvimento do relatório JasperReports.

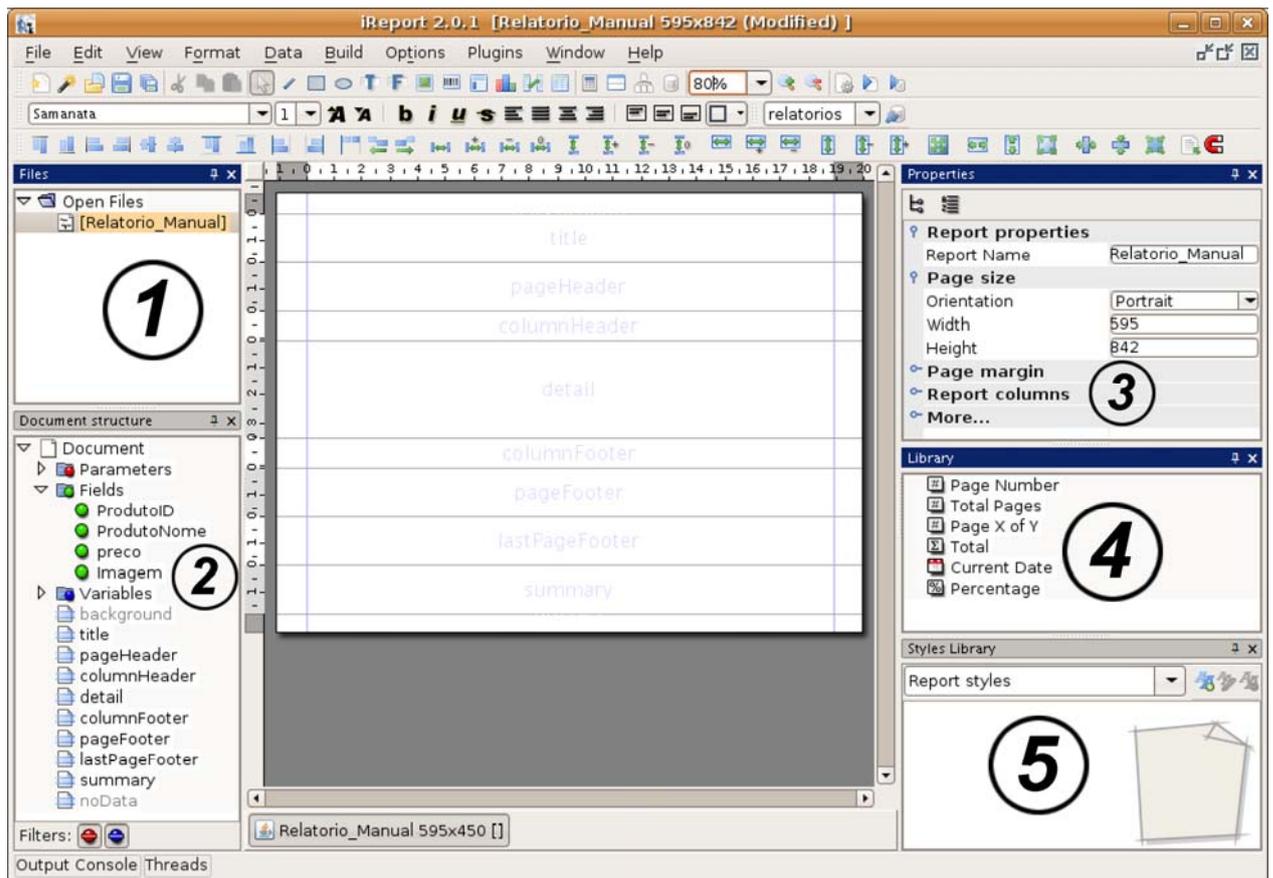


Figura 16 – A janela do iReport em um relatório vazio no Linux

Cada um destes painéis espalhados ao redor de seu relatório possui acessibilidade também de outra forma e, em algumas vezes, existem mais de um jeito de acessar. Estes painéis numerados na **Figura 16** são explicados na lista a seguir:

- (1)**File** – Exibe os relatórios abertos. O acesso é feito com um duplo clique sobre o relatório que deseja exibir na janela. O ícone apresentado ao lado do nome do relatório neste painel também indica modificação no arquivo. Caso esteja com dois riscos, é porque houve alteração e não foi salvo. Alternativamente você pode acessar outro relatório aberto através do menu **Window**.
- (2)**Document structure** – Exibe os parâmetros, campos da tabela da sua query, variáveis pré-definidas e criadas e as bands do relatório.
- (3)**Properties** – Exibe a propriedade de acordo com o elemento selecionado em sua estrutura (Estrutura do documento). As propriedades de cada elemento do relatório também pode ser acessível facilmente com o direito do mouse sobre, selecionando o item **Properties** no menu de contexto.
- (4)**Library** – Exibe algumas variáveis e funções pré-definidas pelo próprio iReport que são úteis e comuns na maioria dos relatórios.
- (5)**Styles Library** – Mostra os estilos criados para serem utilizados no desenvolvimento do relatório.

Configurando as Bands

Para começar na criação do desenho do relatório, você vai configurar inicialmente as bands que não serão exibidas. Com o direito do mouse sobre a página do relatório (design), selecione no menu de contexto o item **Band Properties**.

Observe que na caixa de diálogo, todas as bandas são exibidas. Cada band possui uma altura pré-estabelecida e pode ser alterada aqui, simplesmente mudando seu valor no campo **Band Height**.



Figura 17 – Propriedades das bands

Caso você queira **retirar a visibilidade** de uma destas **bands**, basta deixar o **número zero** no campo **Band Height**.

Portanto, as bands que não queremos devemos zerar sua altura e depois clicar no botão **Apply** antes de alterar a altura de outra band. As alturas de cada band para a criação do relatório é exibida conforme a **Tabela 1** mostrada a seguir:

Tabela 1 – Alturas das bands utilizadas na construção do relatório

Band	Altura da Banda
<i>title</i>	0
<i>pageHeader</i>	Mantenha como está
<i>columnHeader</i>	Mantenha como está
<i>detail</i>	Mantenha como está
<i>columnFooter</i>	0
<i>pageFooter</i>	Mantenha como está
<i>lastPageFooter</i>	0
<i>summary</i>	Mantenha como está

As outras bands não citadas na **Tabela 1** serão vistas em outro momento. Ao terminar, feche a caixa de diálogo.

Atenção: Você pode configurar as propriedades de cada band selecionando-a através do painel **Document structure** e alterando no painel **Properties**. Neste caso, a alteração não precisa ser aplicada (alterou o valor da propriedade, tirou o foco, mudou).

Conhecendo a barra de ferramentas Text Formatting

A barra de ferramentas **Text Formatting** possui as principais ferramentas para a formatação de um elemento de texto em um relatório JasperReports. A **Figura 4.18** ilustra a barra contendo a respectiva numeração pela ordem de exibição. Logo após o nome e descrição de cada ferramenta.

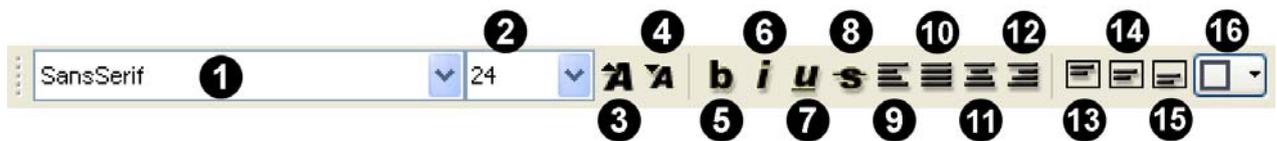


Figura 18 – Barra de ferramentas Formatação de Texto

1. **Font** – Altera a fonte do texto encontrado no elemento.
2. **Font Size** – Altera o tamanho da fonte do texto encontrado no elemento, em uma seleção numérica ou digitada (pressione <ENTER> para confirmar).
3. **Increase Font Size** – Altera o tamanho da fonte do texto encontrado no elemento aumentando sempre em dois pontos (2pt).
4. **Decrease Font Size** – Altera o tamanho da fonte do texto encontrado no elemento diminuindo sempre em dois pontos (2pt).
5. **Bold** – Aplica negrito ao texto encontrado no elemento.
6. **Italic** – Aplica itálico ao texto encontrado no elemento.
7. **Underlined** – Sublinha o texto encontrado no elemento.
8. **Strikethrough** – Desenha uma linha no meio do texto encontrado no elemento (tachado).
9. **Align Left** – Alinha o texto à esquerda em relação à área do elemento.
10. **Align Justify** – Alinha o texto às margens esquerda e direita da área do elemento.
11. **Align Center** – Alinha o texto ao centro em relação à área do elemento.
12. **Align Right** – Alinha o texto à direita em relação à área do elemento.
13. **Align Top** – Alinha o texto na parte superior vertical em relação à área do elemento.
14. **Align Vertical Axis** – Alinha o texto no centro vertical em relação à área do elemento.
15. **Align Bottom** – Alinha o texto à parte inferior vertical em relação à área do elemento.
16. **Border** – Adiciona uma borda ao redor do elemento.

Adicionando os textos estáticos

Antes de iniciar a criação do desenho, você tem na **Figura 19** como será o resultado final.

Produtos Cadastrados			
Código	Produto	Preço R\$	Imagem
\$F	\$F{ProdutoNome}	\$F{preco}	\$F{Imagem}
			\$V

Figura 19 – Design final do relatório

Para criar a aparência vista na **Figura 19**, os seguintes passos serão feitos:

1. Na barra de ferramentas principal, clique na ferramenta **Static Text** e arraste um tamanho qualquer na band **pageHeader**.
2. Dê um duplo clique neste elemento desenhado e digite: **Produtos Cadastrados**. Na barra de ferramentas formatação altere em **Font Size** para **24** e clique no botão **Bold**. Textos estáticos são representados pelo elemento JasperReports `<staticText>`. Alguns elementos são comuns a todos, portanto não será apresentado aqui, por enquanto.

`<staticText>`

...

```
<textElement verticalAlignment="Top" rotation="None">
```

```
  <font
```

```
    pdfFontName="Helvetica-Bold"
```

```
    size="24"
```

```
    isBold="true"
```

```
    isStrikeThrough="false" />
```

```
</textElement>
```

```
  <text><![CDATA[Produtos Cadastrados]]></text>
```

```
</staticText>
```

O sub-elemento `<textElement />`, de `<staticText />`, determina o alinhamento vertical e se há rotação de texto, através dos atributos **verticalAlignment** e **rotation**. Todas as configurações feitas na fonte do texto é adicionado em ``. Este elemento, por sua vez, possui alguns atributos que determinam o comportamento da fonte, como se é grrito (***isBold***) ou itálico (***isItalic***).

Em `<text />`, também sub-elemento de `<staticText />`, é o local onde o texto que será exibido no relatório é adicionado, também em um bloco CDATA.

3. Pegue a ferramenta **Line** e risque pouco abaixo de **Produtos Cadastrados**. A linha pode ser desenhada em uma reta perfeita, na horizontal, vertical ou em ângulo se pressionada a tecla <Shift> antes de arrastar. A linha segue o grid definido no relatório. O elemento <line /> é a sua representação em um relatório JasperReports. O atributo **direction** está em **TopDown** (opção **Line Direction** em **Top-down** no iReport), Indica qual direção das duas diagonais do retângulo a linha será representada e vista na compilação. Caso seja **Bottom-up**, no atributo **direction** você terá **BottomUp**. Isso é visto somente se a linha tiver uma determinada **altura** acima de **zero**. No caso do exemplo criado para o relatório, o valor é **zero**.

```
<line direction="TopDown">
```

...

```
</line>
```

4. Adicione um novo elemento **Static Text**, mas desta vez na band **columnHeader**. Digite código e altere seu tamanho para **12** e coloque em **Bold**. Digite **Código**.
5. Copie e cole três vezes. Observe que temos três elementos estáticos (Static Text), que cumpre o papel de rótulos, configurados identicamente com a palavra Código. Tudo que você deve fazer é alinhá-los e dimensioná-los para que fiquem por entre as margens do relatório.

Observação: Existem alguns detalhes interessantes de no iReport, com relação as suas características visuais. Uma delas é o fato do elemento ficar na divisão de bands. Note que a sua cor, ao redor, fica vermelha. No painel **Document Structure**, também fica vermelho.

Quando você arrasta um elemento, entra em ação a ferramenta **Magnetic**, ativa por padrão, que auxilia no alinhamento dos elementos no relatório ao arrastá-los. Linhas guias surgem, tanto na vertical como na horizontal, dependendo da situação para auxiliar o alinhamento no arraste do elemento ou no seu dimensionamento com relação aos demais encontrados.

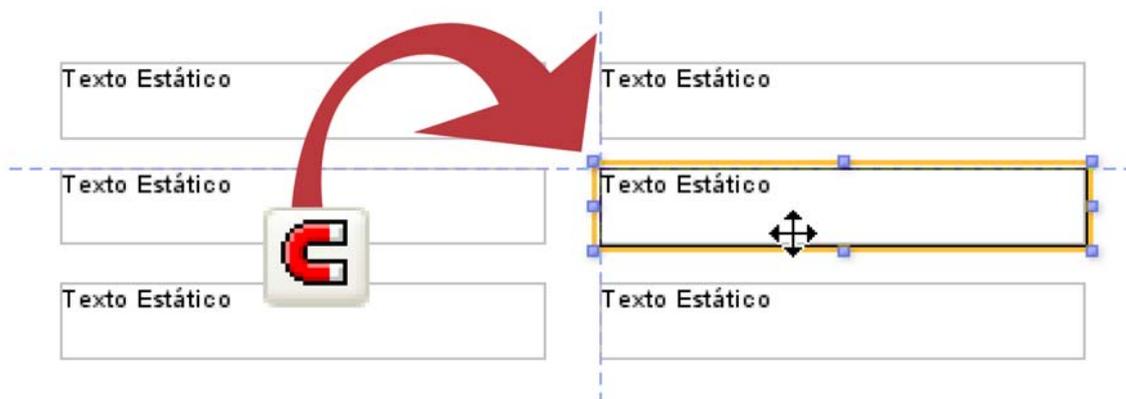


Figura 20 – A ferramenta Magnetic em ação. Note as linhas que guiam o alinhamento.

6. Digite os rótulos: **Produto**, **Preço R\$** e **Imagem**. Selecione todos os rótulos (textos estáticos) de **columnHeader** (utilizando <Shift> pressionado).



Figura 21 – Rótulos digitados no cabeçalho da coluna

7. Na barra de ferramentas **Elements Formatting** clique no ícone **Horizontal space: Make equal** (1). Isso fará com que o espaçamento por entre os elementos sejam idênticos na horizontal.



Figura 22 – Destaque na ferramenta **Horizontal space: Make equal**

8. Crie um retângulo sobre estes elementos, arrastando a ferramenta **Rectangle**. Faça com que este retângulo ocupe todo o tamanho da band **columnHeader**. No painel **Properties**, clique no botão com três pontos, em **Background**. Na caixa de diálogo escolha uma cor e confirme. Caso deseje adicionar a mesma cor na borda, em **Cor**, repita o mesmo processo.

Dica: Caso queira arredondar um retângulo, no iReport, clique com o direito do mouse sobre ele e no menu de contexto selecione **Properties**. Na guia **Rectangle**, em **Rectangle radius**, digite um número acima de zero.

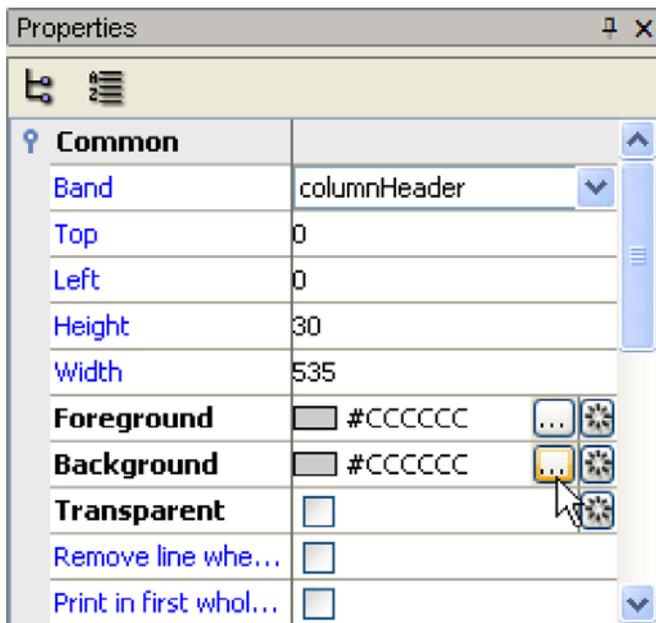


Figura 23 – Propriedades do Retângulo de columnHeader

O retângulo é representado no JasperReports pelo elemento `<rectangle />` :

`<rectangle>`

...

`</rectangle>`

9. Clique com o direito do mouse sobre o retângulo e no menu de contexto selecione **Send to Back**. Alternativamente você pode ir ao menu **Format** em **Send to Back**.

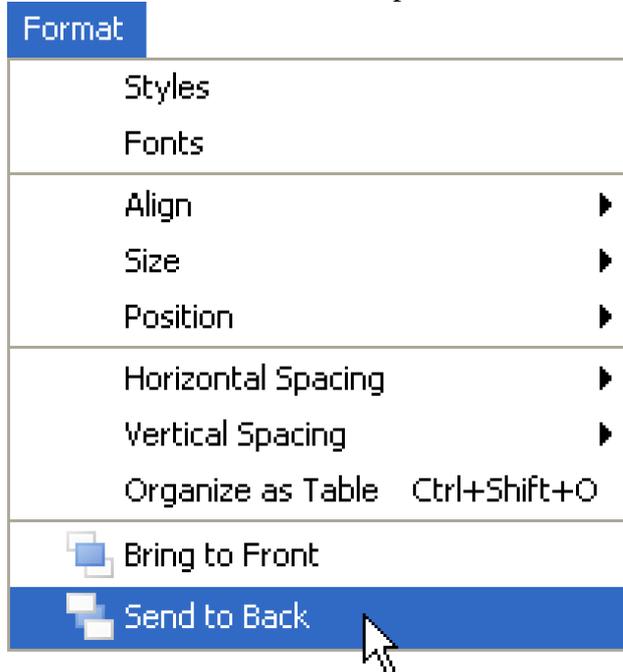


Figura 24– Item Enviar para trás do menu Formatar

10. Vá até a ferramenta **Frame**, na barra de ferramentas principal, e arraste na band **Detail** de uma margem a outra, mas não na mesma altura da band. O Frame é representado pelo elemento JasperReports `<frame />`. O Frame é usado para organizar outros elementos em seu interior.

`<frame>`

...

`</frame>`

11. Adicione da barra de ferramentas principal, quatro campos de texto (**Text Field**), sobre o Frame criado. Não copie e cole desta vez. Observe no painel **Document structure** se os Campos de texto (`textField-x`) estão todos dentro do frame.

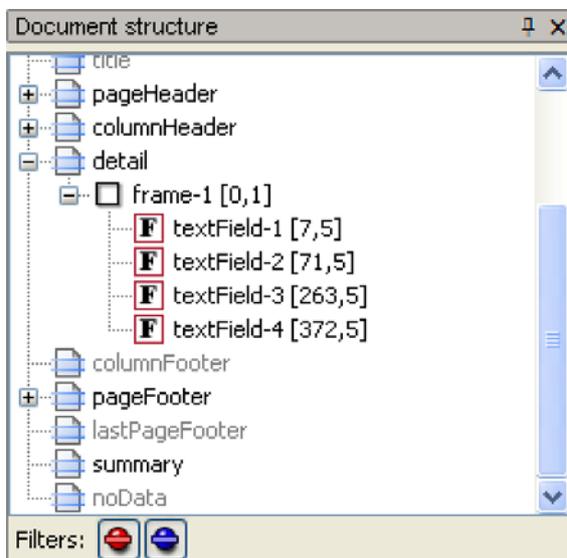


Figura 25 – Exibição de Text Fields no Frame

12. Selecione esses quatro *campos de texto* através do painel **Document structure**, clicando sobre o `textField-1` e com a tecla `<Shift>` pressionada, clicando no último, em `textField-4`. Na barra de ferramentas **Elements Formatting**, clique no ícone **Align Top** (1) e depois em **Same height** (2).



Figura 26 – Destaques das ferramentas para ajustar os Text Fields

13. Agora você vai vincular cada coluna do banco de dados com o Text Field. Clique com o direito do mouse sobre o primeiro Text Field. No menu de contexto selecione o item **Edit**

Expression. Na caixa de diálogo **Expression editor...**, apague o **\$F{Field}** e na parte inferior, no quadro central, dê um duplo clique em **ProdutoID**. Clique no botão **Apply** para confirmar.

14.

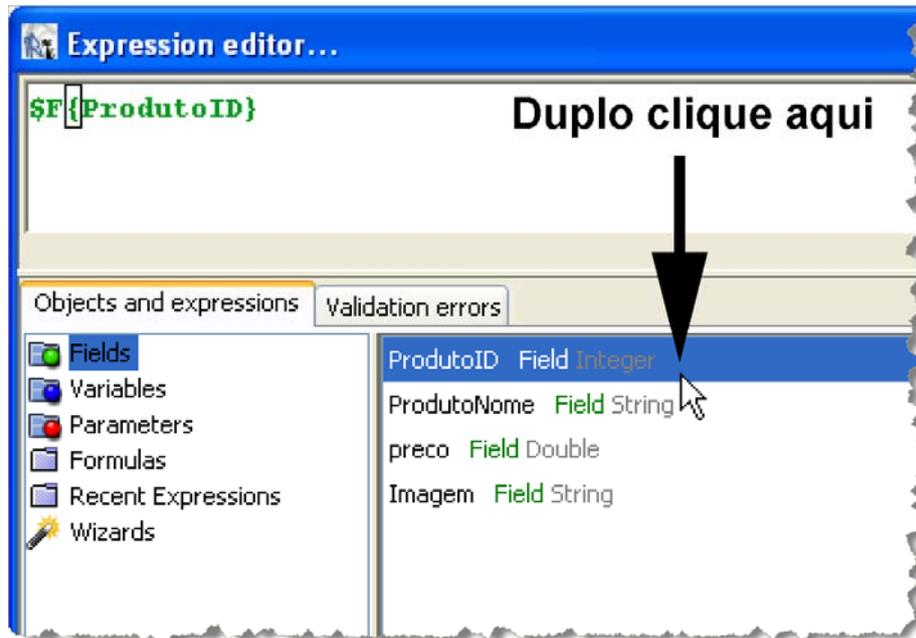


Figura 27– Vinculando o Text Field a coluna ProdutoID

15. Execute o mesmo processo para cada um dos demais campos.

16. Para o campo **ProdutoID** e **preco**, temos que fazer um último ajuste. Ambos são de tipos diferentes de String e precisam ser configurados para os seus respectivos tipos de dados, possibilitando a compilação pelo engine do JasperReports. Clique novamente com o direito do mouse sobre o Text Field **ProdutoID** e selecione no menu de contexto o item **Properties**. Na caixa de diálogo, na guia **Text Field**, altere em **Text Field Expression Class** para **java.lang.Integer**.

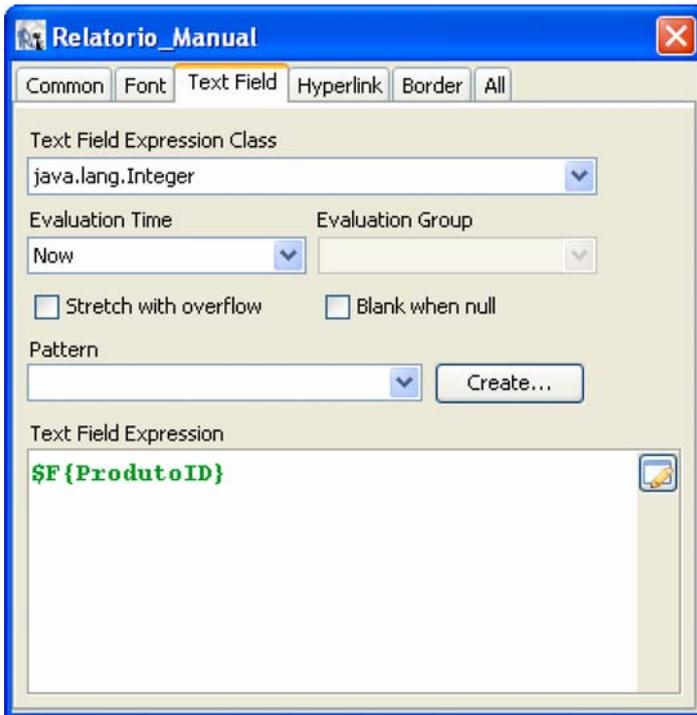


Figura 28 – Propriedades do campo ProdutoID

O mesmo processo deve ser feito no campo de texto **preco**. Em **Text Field Expression Class** selecione **java.lang.Double**. Em **Pattern** selecione a máscara contida na caixa de combinação. Caso não esteja um modelo desejado, basta clicar no botão **Create**. Existem diversos modelos de formatação de números.

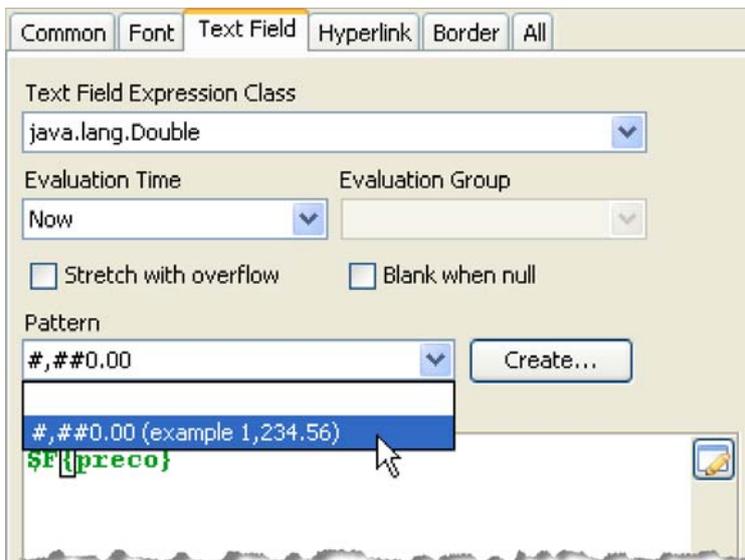


Figura 29 – Propriedades do campo preco com o Pattern configurado

17. Para auto-ajustar o tamanho da band **detail**, dê um duplo clique na linha divisória entre as bands.



Figura 30 – Auto-ajustando a altura da band detail

18. Do painel **Library**, arraste para a band **pageFooter** o item **Page Number**. Posicione-o a direita da página ou ao centro, se desejar.

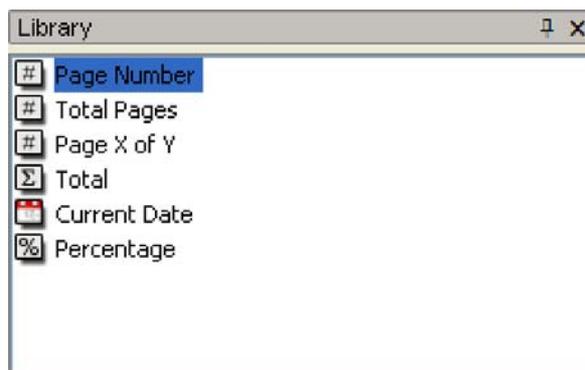


Figura 31– Número da página selecionado no painel Library

19. Vá ao menu **Build** e clique no item **Execute (with active connection)**.

Os símbolos para exibir o formato de números

Há pouco você ajustou o preço do seu produto a ser exibido através de um pattern para que houvesse a separação de casas decimais e de milhar. Esses “padrões” podem ser configurados e no caso agora, você vai conhecer os símbolos para números

Tabela 2 – Símbolos para criar os padrões de exibição numéricos

Símbolo	Para	Descrição
0	Número	Dígitos
#	Número	Dígitos, zeros mostrados como ausente.
.	Número	Separador decimal

-	Número	Sinal negativo (menos)
,	Número	Separador de milhar
;	Número	Separa valores positivos de negativos. Exemplo: #,##0.00;(#,##0.00), que resultará em 1.325,99 (-1.325,99) .
E	Número	Exponenciação
%	Prefixo ou sufixo	Multiplica por 100 e mostra o resultado em porcentagem
\u2030	Prefixo ou sufixo	Multiplica por 1000 e mostra o resultado por milha
¤ (\u00A4)	Prefixo ou sufixo	Símbolo de moeda
'	Prefixo ou sufixo	Usado para caracteres especiais. Por exemplo o símbolo # 99 será feito assim: '# '##

Conhecendo melhor o Text Field

O Text Field é bem complexo como você pôde perceber sobre o que já foi desenvolvido até o momento. Este possui algumas propriedades que o tornam um elemento capaz de trazer dados de uma tabela, de uma classe ou até mesmo executar cálculos. Este dinamismo é graças a suas expressões suportadas, além dos diversos tipos de dados que este elemento é capaz de suportar.

Temos em um Text Field as seguintes propriedades:

- **Text Field Expression Class** – Especifica o tipo de retorno para a expressão dada ao campo. Neste caso você tem objetos Java que manipulam dados para trabalhar com cálculos e textos assim como também com informações SQL. Caso o tipo não se encaixe nos encontrados nesta lista, você pode utilizar **java.lang.Object**.
- **Evaluation Time** – Determina em que fase do relatório a expressão contida no Text Field será elaborada.
- **Evaluation Group** – Determina o tempo de avaliação do Text Field com relação ao grupo. Grupos serão abordados no *Capítulo 5*.
- **Stretch with overflow** – Quando selecionada, permite ao Text Field se adaptar ao conteúdo na vertical. Isso é ótimo quando o tamanho determinado ao Text Field é menor que o texto a ser exibido.
- **Blank when null** – Permite exibir o campo mesmo quando a expressão retornar um objeto nulo.
- **Pattern** – Possibilita a criação de um padrão de exibição dos dados a serem impressos. É utilizado para formatar com uma “máscara” desde valores monetários até datas, horas ou números científicos. Veja detalhes sobre o Pattern Date no próximo capítulo.
- **Text Field Expression** – Local onde você pode digitar a expressão do Text Field ou simplesmente exibir a expressão criada pelo Expression editor.

Exibindo imagens em seu relatório

A última modificação que será feita em seu relatório manual é da adição de imagens. Seu relatório ficará similar a **Figura 32** mostrada a seguir.

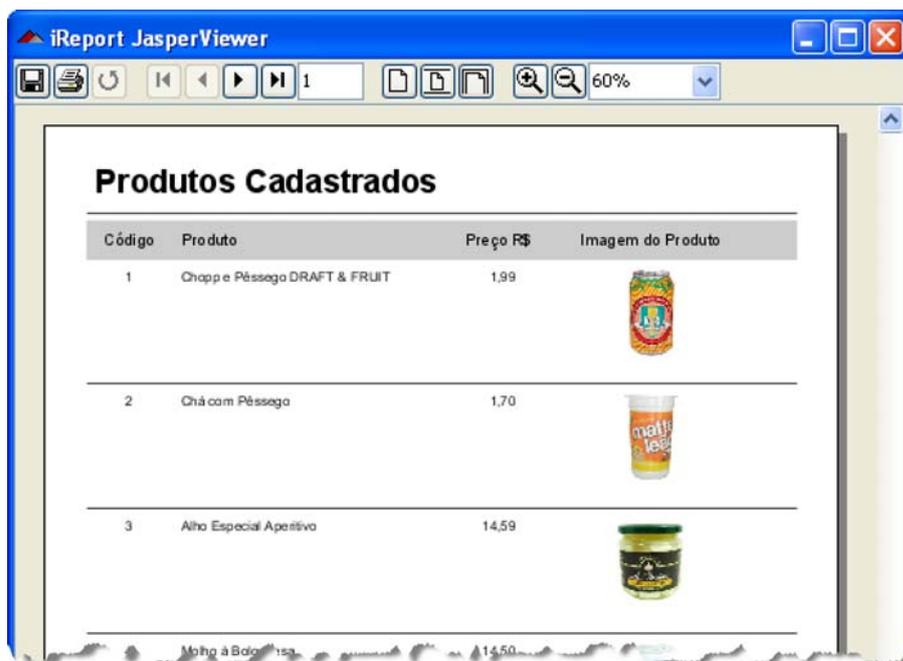


Figura 32 – Imagens no relatório com dados da tabela

Para essa alteração, salve com o nome de **Relatorio_Manual_Imagem**.

Apague o **Text Field** da imagem e o substitua pelo elemento **Image**. Ajuste a imagem para que ela tenha a **Height** (Altura) de **70** e a **Width** (Largura) de **85**, em **Properties**. Ainda no painel **Properties**, em **Image**, clique no botão com três pontos da propriedade **Image expression**.

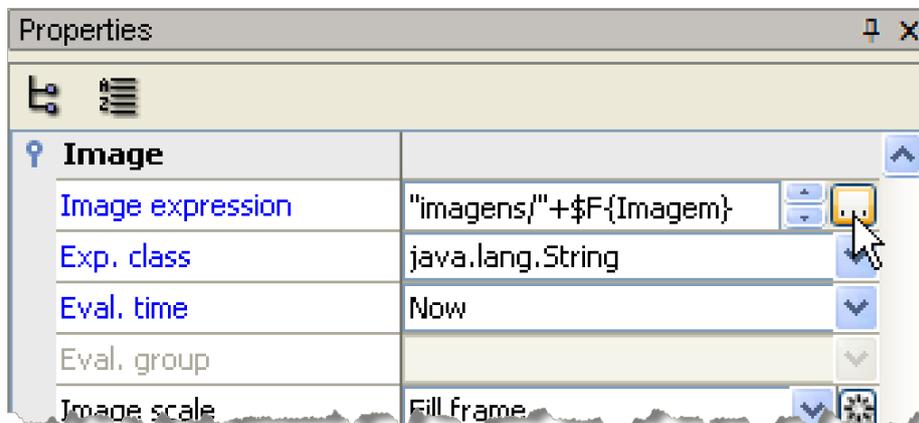


Figura 33 – Adicionando Expressão da imagem

Na caixa de diálogo **Expression editor**, coloque entre “aspas” o caminho para o diretório onde se encontra as imagens. Depois, concatene (em Java é o sinal de “+”) com o campo imagem do banco de dados, que já está com o nome de cada produto armazenado. Por exemplo, no caso de suas imagens estarem em um diretório chamado de **imagens**, você faria a seguinte expressão:

"imagens/" + \$F{Imagem}

Perceba que não foi preciso passar o caminho completo neste caso. Aplique a alteração.

Ajuste a band **detail** e se desejar, coloque ao fim desta uma **Line** para criar uma separação, como no exemplo mostrado pela **Figura 32**.

Um pouco mais sobre o elemento image

O elemento **image** é capaz de exibir objetos gráficos complexos e possui o suporte a várias extensões de imagens (GIF, PNG ou JPEG).

O tipo que você utilizou no exemplo para renderizar a imagem vinda do diretório foi **java.lang.String**. Mas a imagem possui outros cinco tipos, que estão listados na **Tabela 3**.

Tabela 3 – Image Expression Class do elemento Image

Tipo	Descrição
java.lang.String	O mais comum, onde uma string é interpretada como um nome de arquivo, similar ao que os desenvolvedores Web fazem com imagens em diretórios guardando somente o nome do arquivo no banco de dados. Exemplos de sua expressão: "c:\aplicativo\images\photo.jpg" "br/com/integrator/images/ "+\${Image}
java.io.File	Especifica o arquivo como objeto File para ser carregado como imagem. Exemplo de sua expressão: new java.io.File("c:\photo.jpg")
java.net.URL	Utiliza o objeto java.net.URL para determinar o caminho da imagem. Adequado a relatórios que possuem exportação para o formato HTML. Exemplo de sua expressão: new java.net.URL("http://integrator.com.br/photo.jpg")
java.io.InputStream	Utiliza o objeto java.io.InputStream para a exibição da imagem, verificando se está pronto para ler. Pode ser usado como leitura de um determinado campo do banco de dados retornando o InputStream para leitura. Exemplo de sua expressão: <i>geraStream.getInputStream(\${Image})</i> – onde a classe é criada pelo desenvolvedor
java.awt.Image	Utiliza o objeto java.awt.Image. Sua utilização provável é para a geração de uma imagem dinamicamente, como ocorre em um gráfico de dados. Exemplo de sua expressão: <i>gerarObjeto.criaGrafico()</i>
JRRenderable	Usa a interface do JasperReports

	net.sf.jasperreports.engine. JRRenderable.
--	---

As imagens são objetos complexos, o que não seria diferente em seu tratamento utilizando o JasperReports em um relatório.

Quando você acessa suas propriedades, como por exemplo, direito do mouse, item **Properties**, você tem uma guia chamada **Image** que possui diversos itens referentes a este elemento.

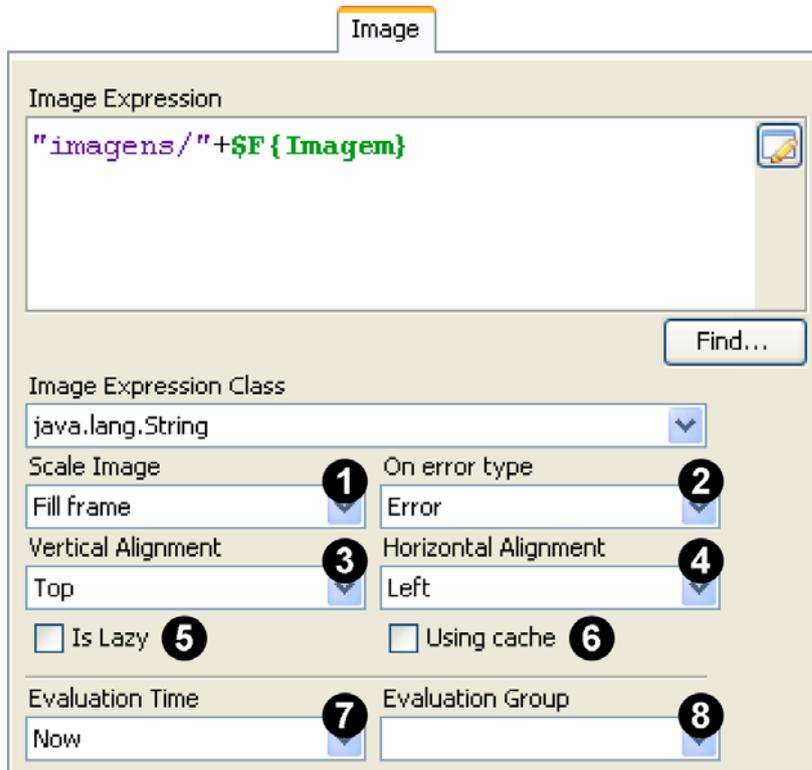


Figura 34 – Propriedades específicas do elemento Image

Observando a **Figura 34** temos alguns pontos numerados indicando comportamentos da imagem em determinada situação na geração do relatório.

1. **Scale Image** – Determina como a imagem irá se adaptar a dimensão do elemento. Neste caso temos três possibilidades:
 - a. **Clip**: A dimensão da imagem não é alterada.
 - b. **Fill frame**: A imagem se adapta a dimensão do elemento, incluindo deformidades (padrão).
 - c. **Retain shape**: A imagem se adapta ao elemento mantendo as proporções originais. Isso significa que a imagem irá se alterar em proporções idênticas, ocorrendo às vezes o não preenchimento total determinado pela área do elemento Image.
2. **On error type** – Determina o que ocorrerá caso a imagem não consiga ser carregada. Existem neste caso três possibilidades:
 - a. **Error**: Gera uma exceção Java, impedindo a renderização do relatório. É o valor padrão.
 - b. **Blank**: A imagem não é exibida, ficando em seu lugar um espaço em vazio.
 - c. **Icon**: Um ícone é impresso no lugar da imagem.
3. **Vertical Alignment** – Determina o alinhamento vertical da imagem com relação à área desenhada do elemento. Assim como os anteriores, possui três opções:

- a. **Top:** Alinha na parte superior do elemento.
- b. **Middle:** Alinha no centro do elemento.
- c. **Bottom:** Alinha na parte inferior do elemento.
4. **Horizontal Alignment** - Determina o alinhamento horizontal da imagem com relação à área desenhada do elemento. Assim como os anteriores, possui três opções:
 - a. **Left:** Alinha a imagem à esquerda do elemento.
 - b. **Center:** Alinha a imagem ao centro do elemento.
 - c. **Right:** Alinha a imagem à direita do elemento.
5. **Is Lazy** – Ótima opção quando estamos carregando uma imagem de uma URL. Faz com que a imagem seja carregada quando o relatório for exportado. Boa opção para relatórios que serão exportados no formato HTML.
6. **Using cache** - Esta opção mantém a imagem em memória para usar novamente se o elemento for reimpresso. A imagem só é mantida em cache se o campo *Image Expression Class* for *java.lang.String*.
7. **Evaluation Time** - Define durante qual fase de criação do relatório a expressão da imagem tem que ser processada. Ocorre que a avaliação de uma expressão pode ser feita quando o engine do JasperReports encontra o elemento durante a criação do relatório, ou também pode ser adiado em alguns casos particulares, como por exemplo, sobre um determinado cálculo. Existem seis possibilidades neste caso:
 - a. **Now:** A expressão é avaliada imediatamente.
 - b. **Report:** A expressão é avaliada somente no final do relatório.
 - c. **Page:** A expressão é avaliada somente no final da página.
 - d. **Column:** Avalia a expressão ao término da coluna.
 - e. **Group:** Avalia a expressão do grupo que é determinado na seleção *Evaluation Group* (8).
 - f. **Band:** Avalia esta expressão após a avaliação da band atual (usado para avaliar expressões de sub-relatórios que retornam valores).
8. **Evaluation Group** – Determina o tempo de avaliação de um determinado grupo para a imagem a ser exibida.

Capítulo 5

"Tudo que uma pessoa pode imaginar, outras podem tornar real."

Júlio Verne

Gerando relatórios com grupos, parâmetros e executando por uma aplicação desktop

Os relatórios mais utilizados em aplicações, sejam desktop ou Web, é com toda certeza, os dinâmicos. Aqueles relatórios que transmitimos uma determinada informação e este nos trazem aqueles determinados dados.

Poucas vezes também criamos relatórios com somente uma tabela. Isso se deve ao fato de que toda aplicação possuir sempre diversas entidades relacionais e de que os dados sempre estão espalhados por entre estas entidades.

Neste capítulo você aprenderá a criar um relatório mais complexo, utilizando mais de uma entidade (tabela) no desenvolvimento, agrupando informações e filtrando-as através de parâmetros. Por fim, aprenderá a chamar este relatório utilizando uma aplicação Desktop.

O relatório que será desenvolvido

O relatório final será composto por quatro tabelas, resultando em um grupo formado por pedidos efetuados com seus respectivos produtos adquiridos. A **Figura 1** ilustra o resultado do relatório.



Pedidos Efetuados			
Cliente: Paul Henriot			
Pedido N.º: 10248		Data: 04/07/1996	
Produto	Qtd.	Preço Unitário R\$	Total Parcial R\$
Lombo Condimentado Fatiado	12	12,00	144,00
Sopa Americana de Cogumelo	10	8,99	89,90
Manteiga com Sal	5	6,99	34,95
Total do Pedido:			268,85

Figura 1 – Exemplo final do relatório com grupos

Gerando o relatório

Para começar, pegue o relatório zebrado feito no capítulo anterior. Salve como **Relatorio_Agrupado**.

Apague os itens existentes e mantenha apenas os elementos das bands **pageHeader**, **columnHeader** e **pageFooter**. Em **detail** mantenha apenas o **Frame**.

Em **pageHeader** altere o rótulo para **Pedidos Efetuados**. Quanto aos quatro rótulos encontrados na **band columnHeader** altere como mostra a **Tabela 1** a seguir.

Tabela 1 – Alterações nos rótulos

Static Text em columnHeader	Texto a ser exibido
staticText-2	Produto
staticText-3	Qtd.
staticText-4	Preço Unitário R\$
staticText-5	Total Parcial R\$

Adicione mais uma linha separadora, através da ferramenta **Line**, para ficar abaixo dos rótulos de **columnHeader**. Deixe a formatação dos elementos apenas para o final do relatório construído.

A **Figura 2** ilustra como ficará seu relatório com os elementos existentes e adicionados até o momento.

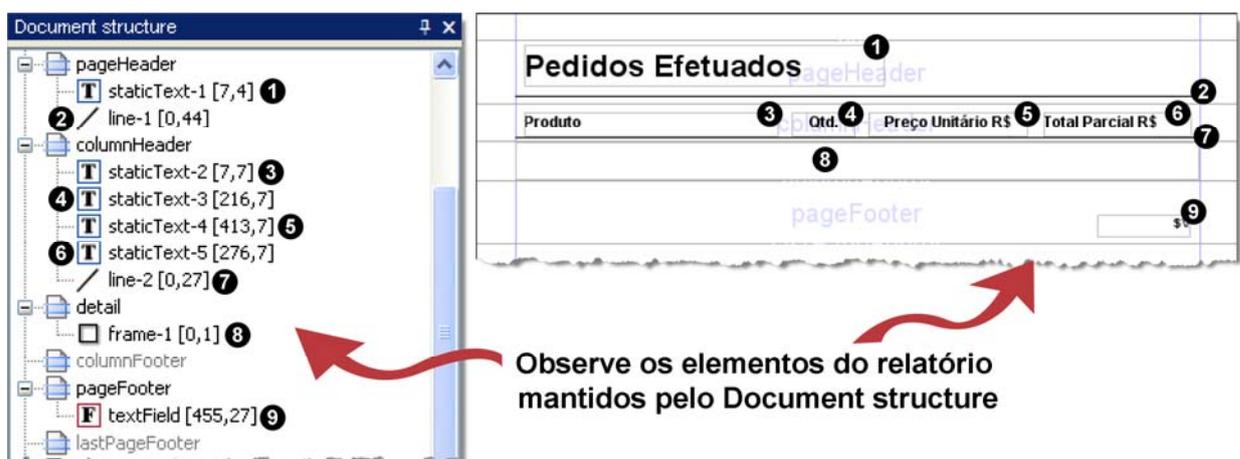


Figura 2 – Detalhe no painel Document structure e relatório

Criando a query do relatório

A query utilizará quatro das cinco tabelas existentes. Para alterar a atual query, vá ao menu **Data** e clique em **Report Query**. Na caixa de diálogo **Report query** clique no botão **Query designer**.

Em **Query designer**, através do segundo quadro a esquerda, selecione as tabelas **clientes**, **detalhes_pedido**, **produtos** e **pedidos** através de um duplo clique sobre seus respectivos nomes. Esta seleção pode ser através do arrastar para o quadro da direita também, caso ache mais prático.

Com as quatro tabelas adicionadas no quadro direito, marque os seguintes campos, mostrados na **Tabela 2** a seguir:

Tabela 2 – Campos selecionados das tabelas para gerar a query

Tabela	Campo
pedidos	PedidoID
	dataPedido
clientes	nome
produtos	ProdutoNome
	Imagem
detalhes_pedido	precoVenda
	quantidade

Veja os campos selecionados na **Figura 3**.

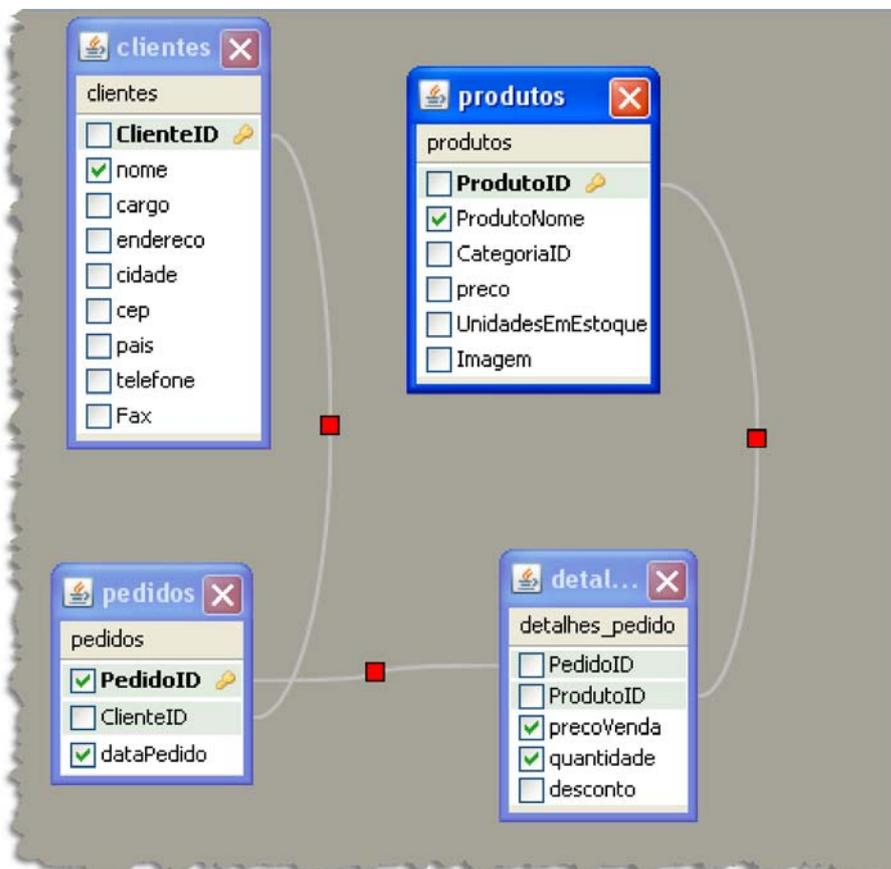


Figura 3 – Criando a query visualmente através do Query designer

Para completar a query, você deverá adicionar uma ordenação pelo código do pedido (PedidoID). Clique com o direito no campo **PedidoID**, no desenho da query do primeiro quadro, e selecione no menu de contexto o item **add to order-by**. Confirme logo após.

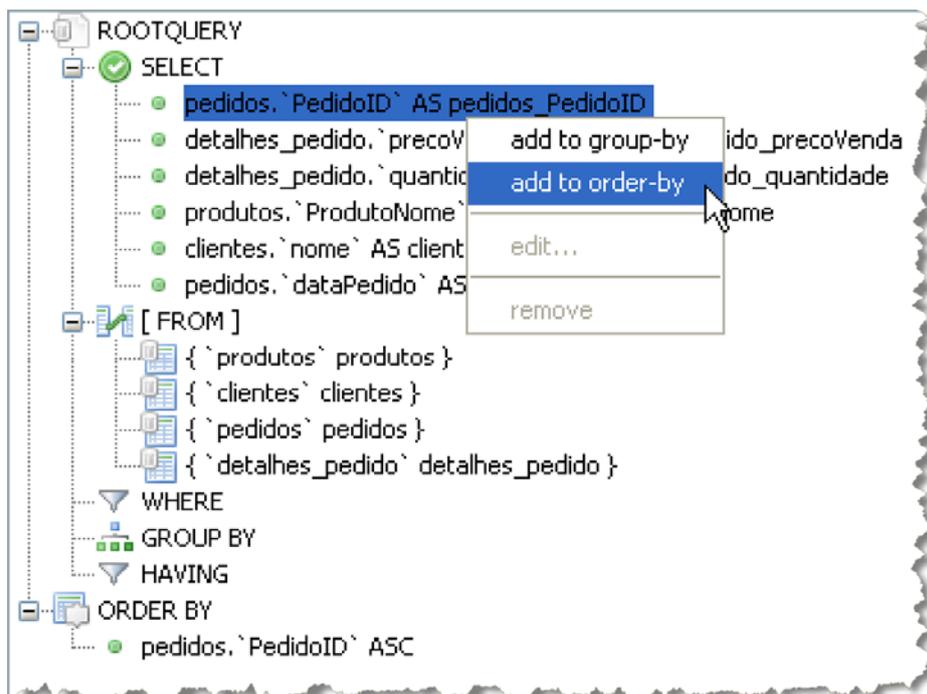


Figura 4 – Adicionando uma ordenação através do query designer

Nota: Aproveite e retire o “*” encontrado do relatório antigo, clicando com o direito do mouse sobre ele e selecionando o item **remove**.

Retornando ao **Report query**, modifique o nome dos “**alias**” retirando os prefixos referentes aos nomes das tabelas. Confirme logo após. Veja na **Figura 5** como ficará:

```

SELECT
    pedidos.`PedidoID` AS PedidoID,
    detalhes_pedido.`precoVenda` AS precoVenda,
    detalhes_pedido.`quantidade` AS quantidade,
    produtos.`ProdutoNome` AS ProdutoNome,
    clientes.`nome` AS nome,
    pedidos.`dataPedido` AS dataPedido
FROM

```

Figura 5 - Detalhe dos alias dos campos encontrados na query SELECT

Criando o grupo

Para que o relatório exiba os produtos por pedido, você tem que criar um grupo por cada campo **PedidoID** da tabela **pedidos**.

Vá ao menu **View** e clique no item **Report Groups**. Alternativamente pode acessar **Report Groups** pelo ícone na barra de ferramentas principal.

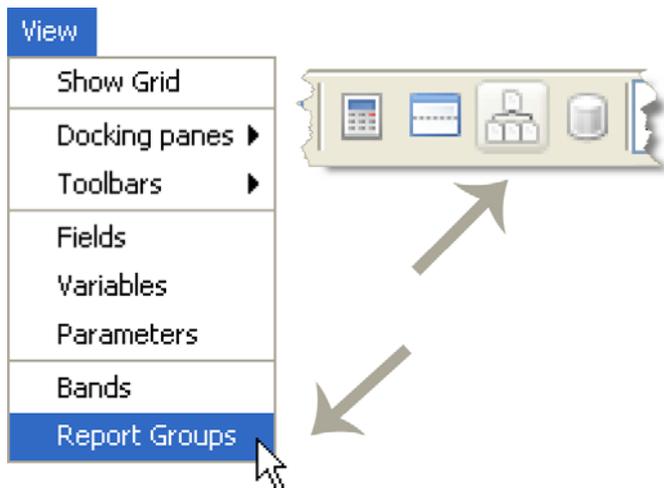


Figura 6 – Item Report Groups do menu View e da barra de ferramentas principal

Na caixa de diálogo clique no botão **New**. Na caixa de diálogo **Add/modify group** digite **PedidoGroup** em **Group Name**. Em **Group Expression** clique no ícone **Open the expression editor**. Na caixa de diálogo **Expression editor** dê um duplo clique em **PedidoID** e confirme no botão **Apply**. Confirmado o grupo feito pelo campo **PedidoID** da tabela Pedidos, confirme no botão **OK** de **Add/modify group**.

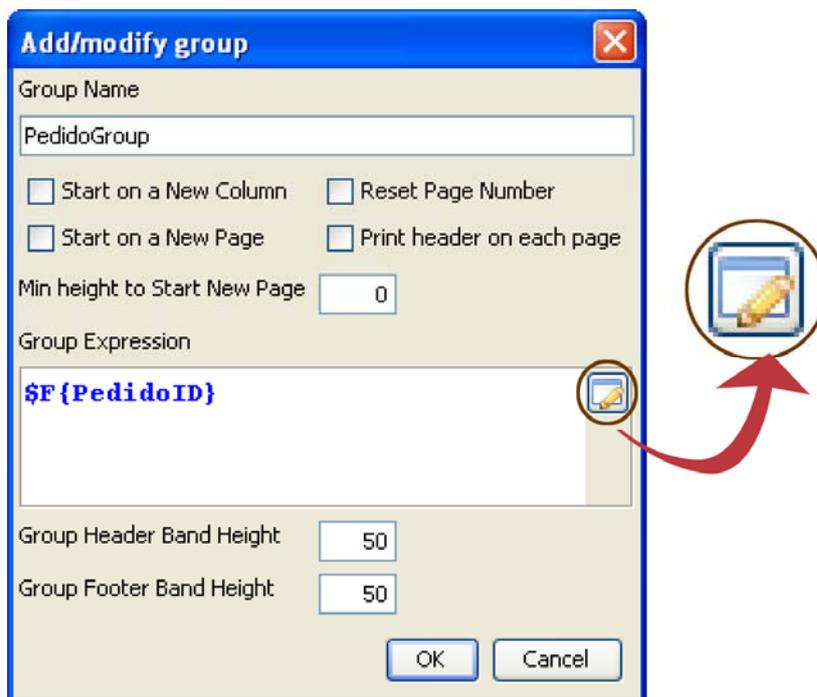


Figura 7 – Detalhe do botão Open the expression editor

Podemos ter diversos grupos em um determinado relatório. Todos eles podem ser modificados, caso haja necessidade, através desta caixa de diálogo. Como você possui apenas um, se houver necessidade de alteração, selecione o grupo e clique no botão que se ativará chamado de **Modify**. Para confirmar, apenas feche essa caixa.

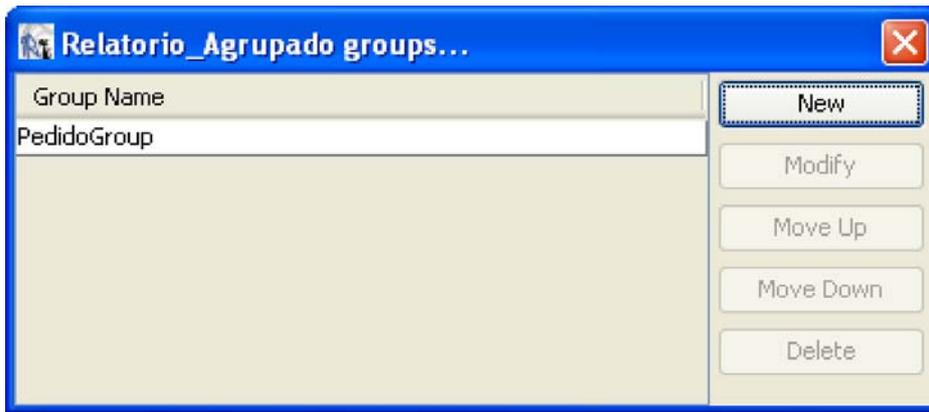


Figura 8 – Grupos existentes no relatório

Uma alternativa para criar um grupo

Há outra forma de se criar um grupo. É um assistente que talvez você ache mais prático. Vá ao menu **Edit** e clique no item **New Report Group Wizard**.

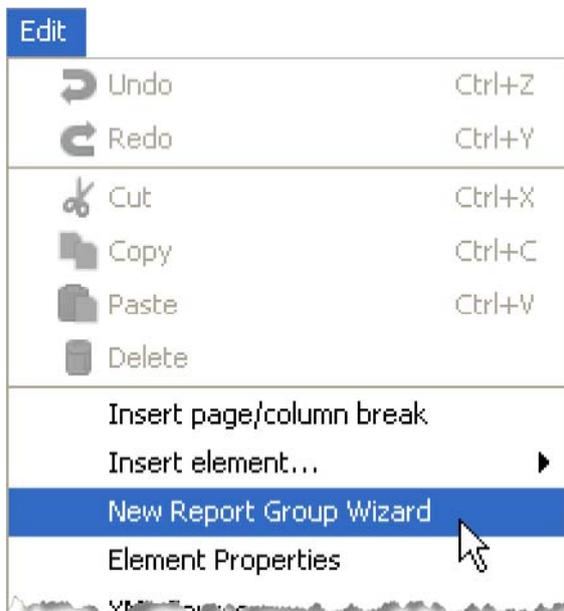


Figura 9 – Acessando o assistente de criação de grupos no iReport

Ao surgir o assistente de grupos, na primeira etapa, em **Criteria**, você determina o nome do grupo, no campo **Group name** e seleciona o campo que deseja usar para agrupar em **Group by following report object**. Em **Next** você encontra a segunda etapa, que determina se você vai criar apenas o grupo de cabeçalho, de rodapé ou os dois (padrão). Terminado o assistente, basta finalizar com o botão **Finish**.

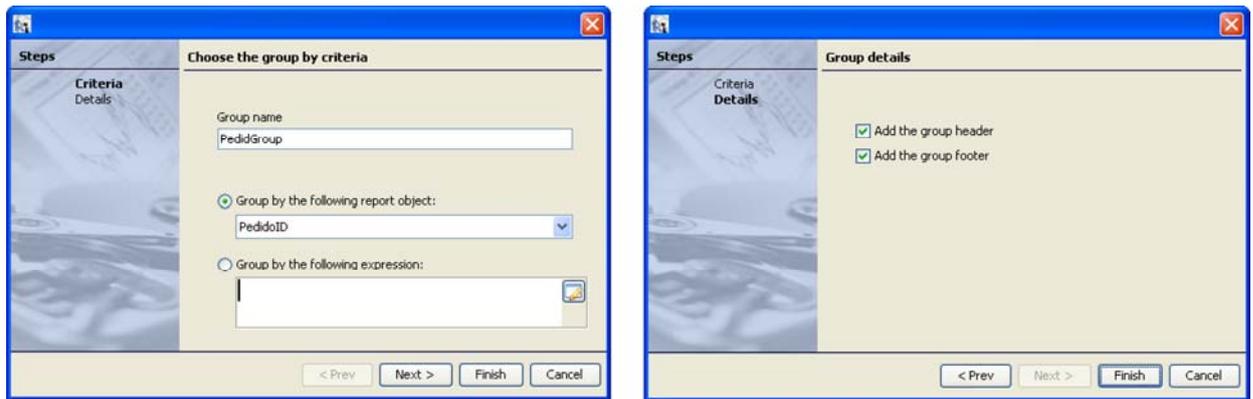


Figura 10 – Assistente de grupos do iReport com duas etapas (Criteria e Details)

Ao adicionar o grupo, seja por assistente ou não, você verá duas novas bands adicionadas ao seu design do relatório. A band superior, chamada de **PedidoGroupHeader** e a inferior, **PedidoGroupFooter**.

Pedidos Efetuados			
Produto	Qty.	Preço Unitário R\$	Total Parcial R\$
PedidoGroupHeader		1	
PedidoGroupFooter		2	
pageFooter			\$V
summary			

Figura 11 – Adição das bands PedidoGroupHeader (1) e PedidoGroupFooter (2)

Adicionando os elementos na band PedidoGroupHeader

Na band **PedidoGroupHeader** adicione o elemento **Rectangle** fazendo-o ocupar todo o espaço disponível. Altere suas propriedades **Background** e **Foreground** para uma determinada cor.

Adicione em pares, um ao lado do outro, três **Static Text** e **Text Fields** respectivamente. Na **Tabela 3** a seguir, você tem as propriedades (painel Properties) de cada elemento que deverão ser alterados.

Tabela 3 – Alterações dos elementos encontrados em PedidoGroupHeader

Elemento	Propriedade	Valor
Static Text 1	Text	Pedido N.º:
	Bold	True (checar)
Text Field 1	Expression	\$F{PedidoID}
	Exp. class	java.lang.Integer
Static Text 2	Text	Cliente:
	Bold	True
Text Field 2	Expression	\$F{nome}
	Exp. class	java.lang.String
Static Text 3	Text	Data:
	Bold	True
Text Field 3	Expression	\$F{dataPedido}
	Exp. class	java.util.Date
	Pattern	dd/MM/yyyy

Sempre que utilizar um **Pattern** para representar um determinado valor, clique nos três botões da propriedade no painel **Properties**.

Existem diversos padrões que podem ser selecionados para que os dados sejam exibidos corretamente no relatório.

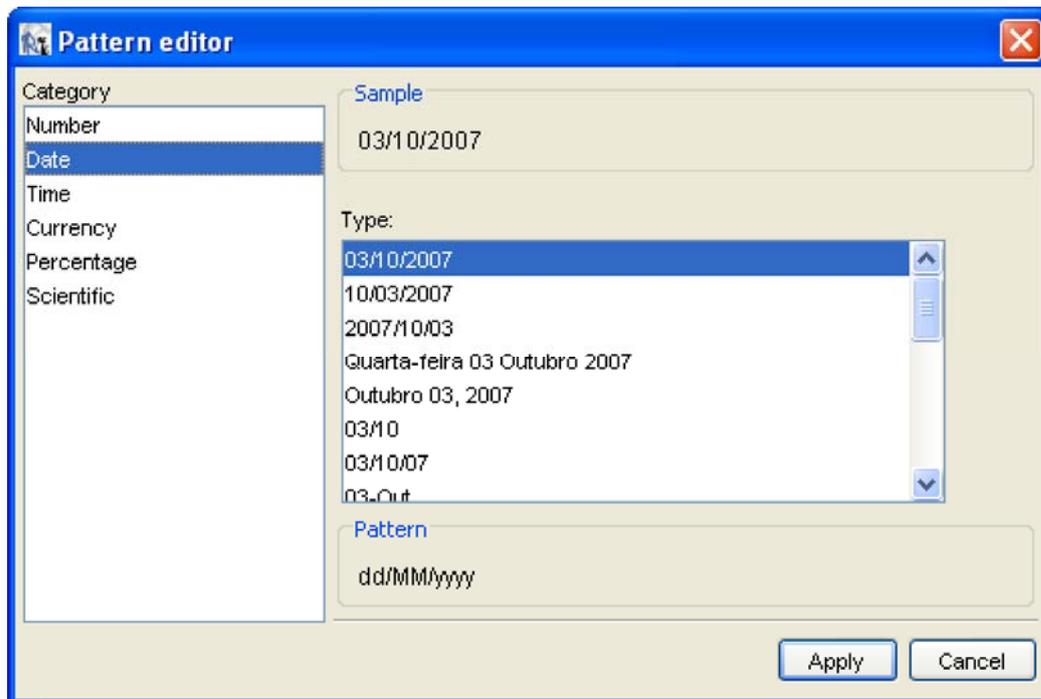


Figura 12 – Caixa de diálogo Pattern editor com Date selecionado

Mas caso não encontre exatamente o pattern que deseja utilizar para desenvolver a exibição da data, você pode personalizar. A seguir você tem a **Tabela 4** para a criação de “padrões” de letras para datas e horas.

Tabela 4 – O Pattern para Datas e horas

Letra	Para que serve?	Exibe por exemplo
G	Designa a Era	AD
y	Ano	2007, 07
M	Mês no ano	04, April
w	Número da semana no ano	27
W	Número da semana no mês	2
D	Número do dia no ano	152
d	Dia no mês	5
F	Número do dia da semana no mês	2
E	Dia da semana	Tuesday
a	AM/PM	AM
H	Formato de hora de 0 à 23	0
k	Formato de hora de 1 à 24	24
K	Hora em AM/PM (0 a 11)	0
h	Hora em AM/PM (1 a 12)	12
m	Minutos em uma hora	25
s	Segundos no minuto	30
S	Milissegundos	150
z	Time zone	Pacific Standard Time, PST, GMT-08:00
Z	Time zone	-0800

Dessa forma, você pode criar vários tipos de padrões para exibir o tempo em seus relatórios. A **Tabela 5** exibe alguns exemplos para melhor compreensão:

Tabela 5 – Exemplos de padrões para formatos de tempo no relatório JasperReports

Formato	Como exhibe (Locale Brasil)
dd/MM/yyyy	25/04/2007
dd/MMMM/yy	25/Abril/07
hh:mm	12:25
hh:mm:ss	12:25:35
dd' de 'MMMM' de 'yyyy	25 de Abril de 2007

Criando variáveis com cálculos no iReport

Cada venda efetuada possui uma determinada quantidade e um valor que foi vendido. Como o relatório visa não somente mostrar a quantidade e o preço, mas também o total parcial ganho por cada produto, você deverá fazer um cálculo. A idéia é simples:

Quantidade X Preço da Venda

Para criar a variável, vá ao menu **View** e clique em **Variables**. Alternativamente você pode acessar através da barra de ferramentas principal o ícone **Report fields, variables and parameters**.

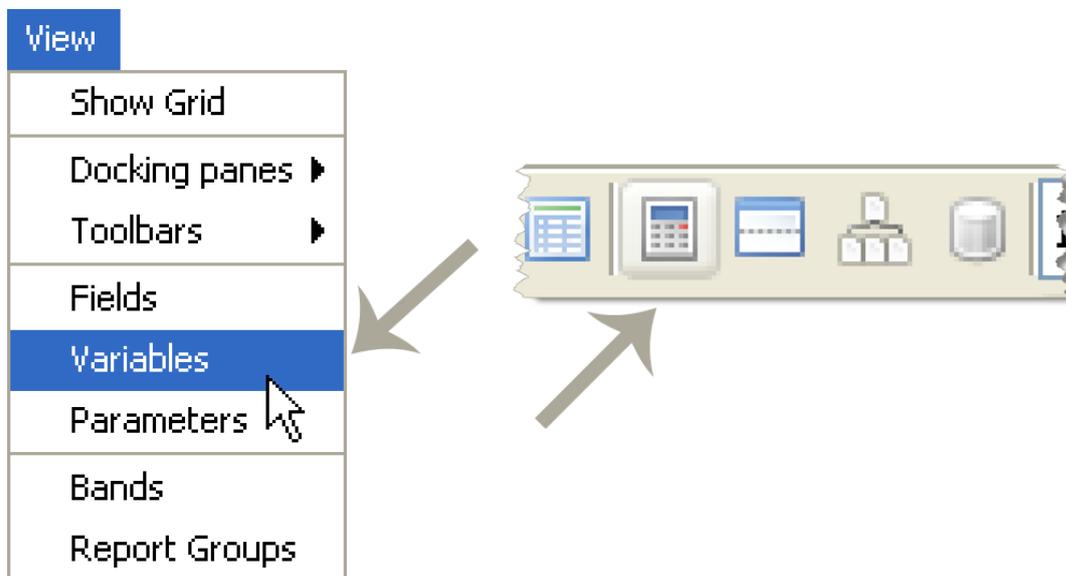


Figura 13 – Detalhe de Variables no menu View e do ícone Report fields, variables and parameters

Ao surgir a caixa de diálogo, você possui três guias: **Fields**, **Variables** e **Parameters**. Para criar uma nova variável, coloque em **Variables** e clique no botão **New**.

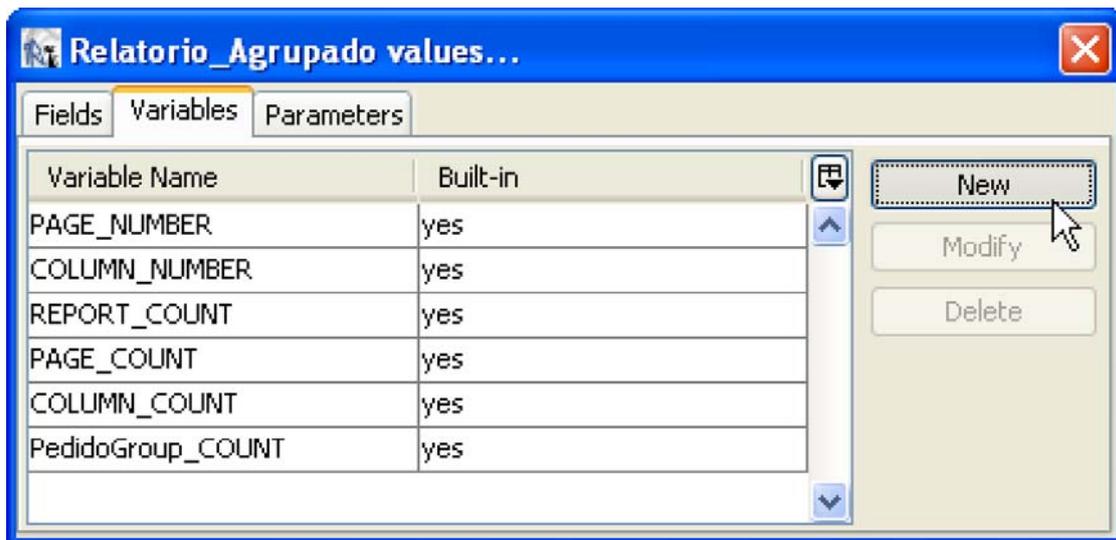


Figura 14 – Adicionando uma nova variável ao iReport

Outra forma de adicionar uma variável é com o direito do mouse sobre um dos elementos encontrados no painel **Document structure**. No menu de contexto selecione **Add** e clique sobre **Variable**.

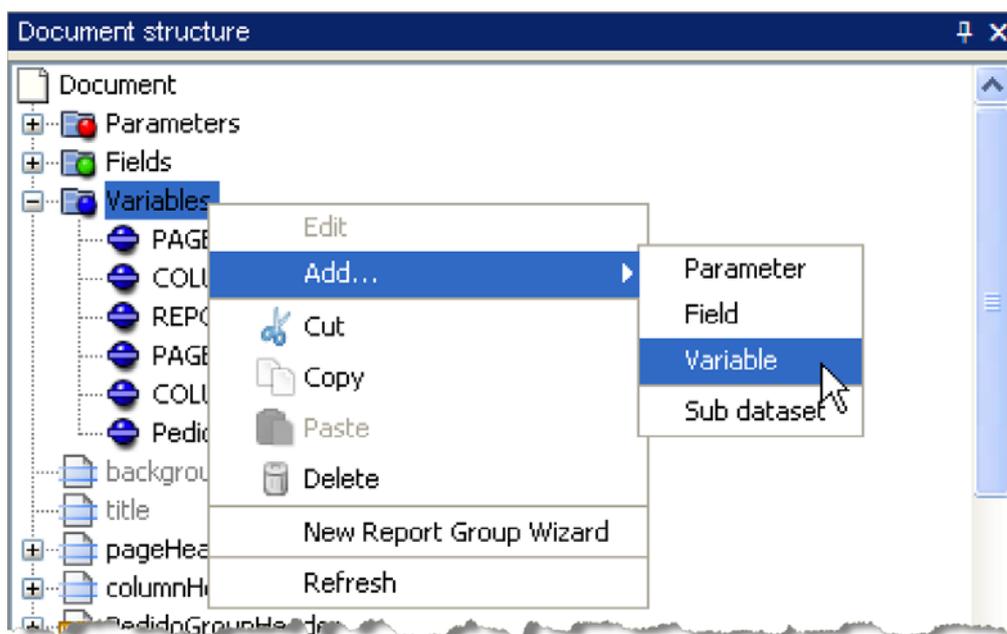


Figura 15 – Adicionando variáveis através do painel Document structure

Na caixa de diálogo **Add/modify variable** digite **MULT_QTD_PRECOVND** em **Variable Name**. Selecione em **Variable Class Type** o tipo **java.lang.Double**. Clique no botão **Open the expression editor** no campo **Variable Expression**.

No **Expression editor** selecione **Wizards** em **Objects and expressions** e dê um duplo clique no item **Multiplication (*)**. Este assistente ajudará você a criar a expressão de multiplicação necessária para a variável em questão.

Na caixa de diálogo **Arithmetic operation** selecione **quantidade** em **Value 1** e **precoVenda** em **Value 2**. Observe que ao centro, entre um valor e outro existe uma caixa de combinação contendo o símbolo de asterisco, caracterizando a multiplicação. Evidentemente existem os demais operadores aritméticos ali. Confirme no botão **OK**.

Observe o resultado efetuado pelo assistente:

```
new Double( ($F{quantidade}.intValue()) *($F{precoVenda}.doubleValue()) )
```

Clique no botão **Apply** para confirmar a expressão criada.

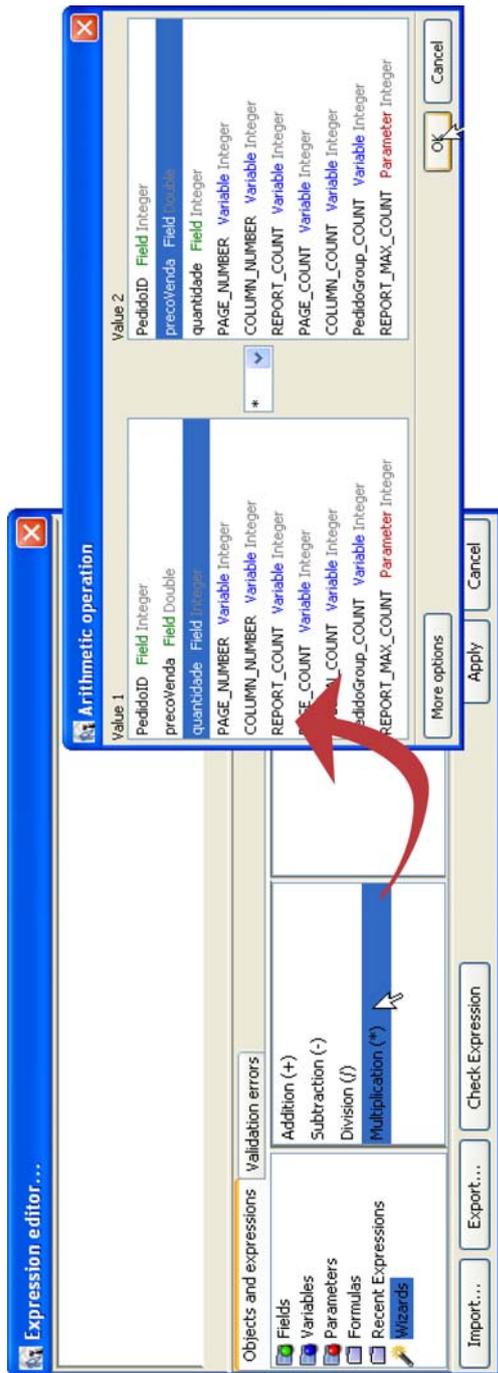


Figura 16 – Criando uma expressão com operação aritmética de multiplicação

Retornando a caixa de diálogo **Add/modify variable**, você deve ter uma configuração semelhante à **Figura 17** mostrada a seguir. Confirme novamente clicando no botão **OK**.

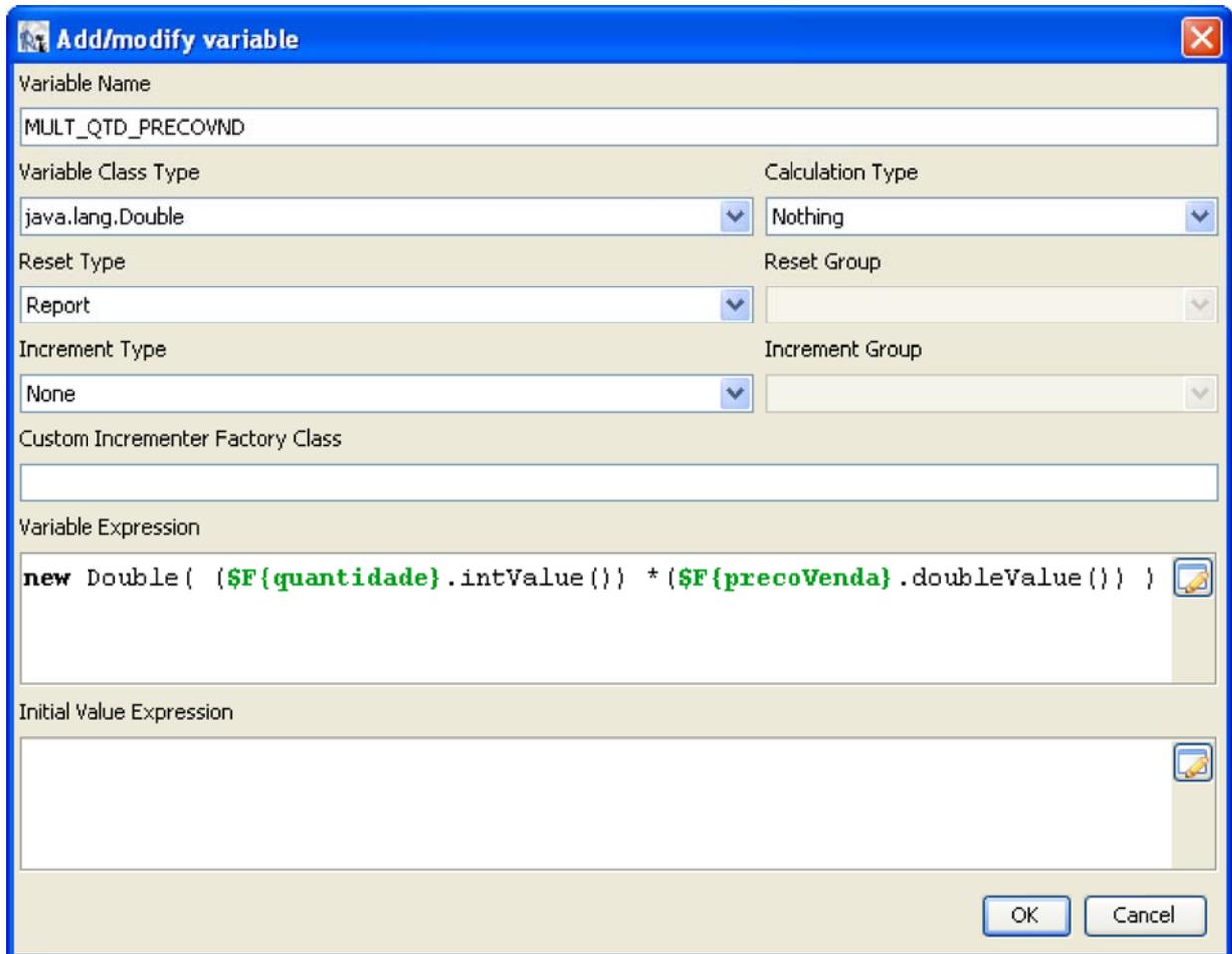


Figura 17 – Variável configurada completamente com a expressão definida

Observe que no painel **Document structure** agora existe em **Variables** a variável criada. Neste mesmo painel, no caso de editar esta variável, clique com o direito do mouse e selecione no menu de contexto o item **Edit**. Se precisar excluir, clique com o direito do mouse e selecione o item **Delete**.

Atenção: Como estamos usando um relatório criado anteriormente com outra query, é possível que variáveis indesejáveis estejam ainda neste relatório, como por exemplo você pode ter a variável **SUM_ProdutoID_1**. Esta variável poderá atrapalhar na compilação do relatório, uma vez que ela não possui mais o campo **ProdutoID**, portanto faça o **Delete** pelo menu de contexto do painel **Document structure**.

Adicionando os elementos na band detail

Na band **detail** está o Frame deixado do relatório anterior. Este frame possui um estilo com o efeito de cores alternadas, o que será perfeito para separar os diversos produtos adquiridos em cada pedido.

Adicione quatro **Text Fields** sobre o **Frame** da band **detail**. Altere suas propriedades conforme a **Tabela 6** a seguir:

Tabela 6 – Configurando os Text Fields da band detail

Elemento	Propriedade	Valor
Text Field 4	Expression	$\$F\{\text{ProdutoNome}\}$
	Exp. class	java.lang.String
Text Field 5	Expression	$\$F\{\text{quantidade}\}$
	Exp. class	java.lang.Integer
Text Field 6	Expression	$\$F\{\text{precoVenda}\}$
	Exp. class	java.lang.Double
	Pattern	#,##0.00
Text Field 7	Expression	$\$V\{\text{MULT_QTD_PRECOVND}\}$
	Exp. class	java.lang.Double
	Pattern	#,##0.00

Observe que no último **Text Field** houve a mudança de Field ($\$F\{\}$) para Variable ($\$V\{\}$). Portanto, em **Expression editor**, selecione **Variables** em **Objects and expressions** para selecionar a variável **MULT_QTD_PRECOVND**.

Ajuste os elementos e execute o relatório - **Build > Execute (with active connection)** -para verificar se está sendo exibido tudo corretamente.

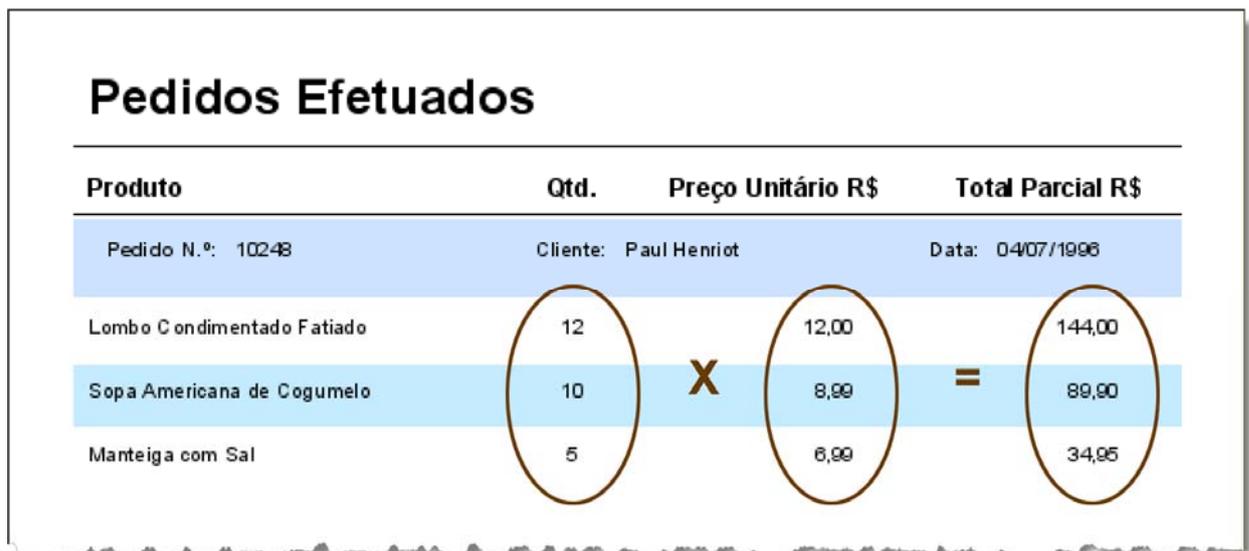


Figura 18 – Detalhe da multiplicação da quantidade pelo preço unitário

Totalizando por pedido

Como você já possui o total parcial por produto comprado, agora nos resta criar o total geral do pedido.

No painel **Library** arraste **Total** para a band **PedidoGroupFooter**.

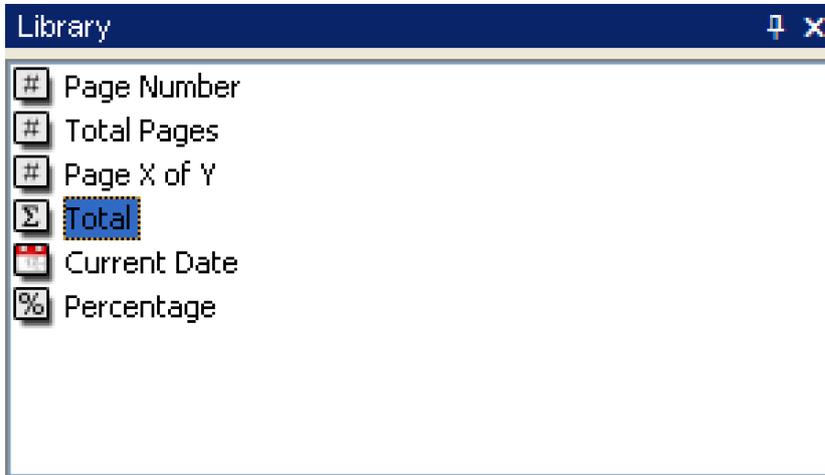


Figura 19 – Painel Library com Total selecionado

Na caixa de diálogo, em **Select Object to Sum** (selecione o objeto para Soma), dê um duplo clique em **Variables** para expandir. Clique na variável criada **MULT_QTD_PRECOVND**. Confirme no botão **OK**.

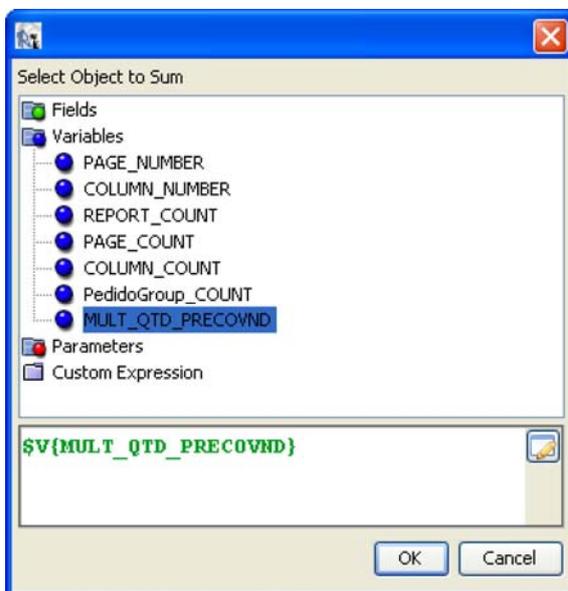


Figura 20– Selecionando a variável MULT_QTD_PRECOVND para somar

Pelo painel **Document structure** você observará que fora criada a variável **SUM_MULT_QTD_PRECOVND_1**.

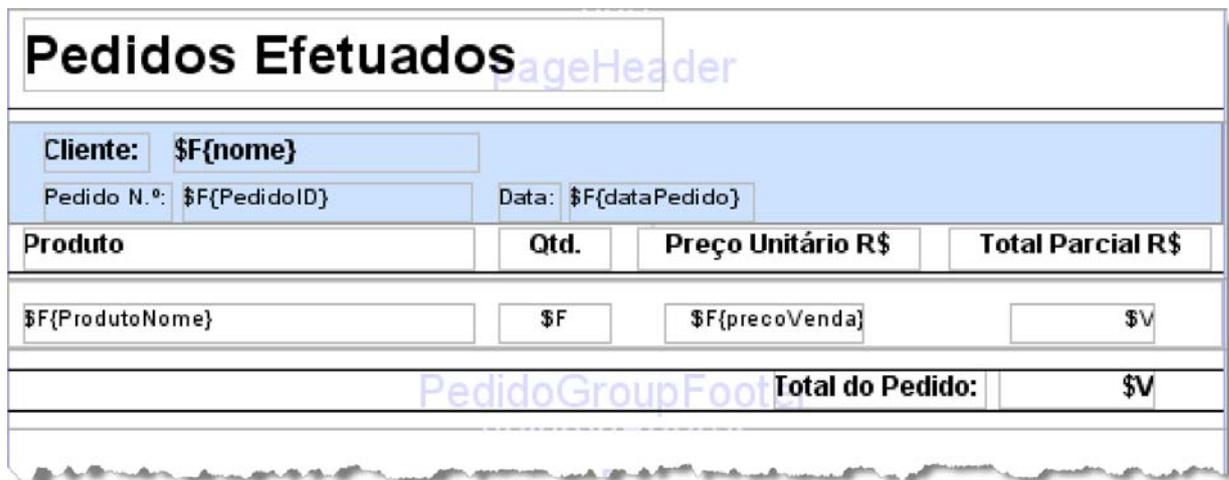
Ajuste o campo na band **PedidoGroupFooter** e adicione um **Static Text**. Digite **Total do Pedido** na propriedade **Text**.

Finalizando a aparência do relatório

Para finalizar a aparência do relatório, execute os passos mostrados a seguir:

1. Aumente o tamanho da band **PedidoGroupHeader** e mova os Static Texts que estão na band **columnHeader** para baixo do retângulo;
2. Altere **Band Header** para **0** nas propriedades de **columnHeader**;
3. Adicione linhas com a ferramenta Line e alinhe os Text Fields e Static Texts para que tenham uma aparência similar ao mostrado no início do capítulo, na **Figura 1**;

Caso queira ter uma base, a **Figura 21** demonstra o relatório no formato final pelo design.



Pedidos Efetuados			
Cliente: \${nome}			
Pedido N.º: \${PedidoID}		Data: \${dataPedido}	
Produto	Qty.	Preço Unitário R\$	Total Parcial R\$
\${ProdutoNome}	\$F	\${precoVenda}	\$V
Total do Pedido:			\$V

Figura 21 – Aparência do design do relatório

Adicionando parâmetros ao relatório

Antes de construirmos um aplicativo para chamar o relatório, você vai criar um parâmetro para exibir um determinado pedido.

Assim como adicionar variáveis, você pode ir ao menu **View** e clicar em **Parameters**, ou pela barra de ferramentas. Mas no caso vamos utilizar o painel **Document structure**, com o direito do mouse sobre um dos elementos, selecionando no menu de contexto **Parameter** em **Add**.

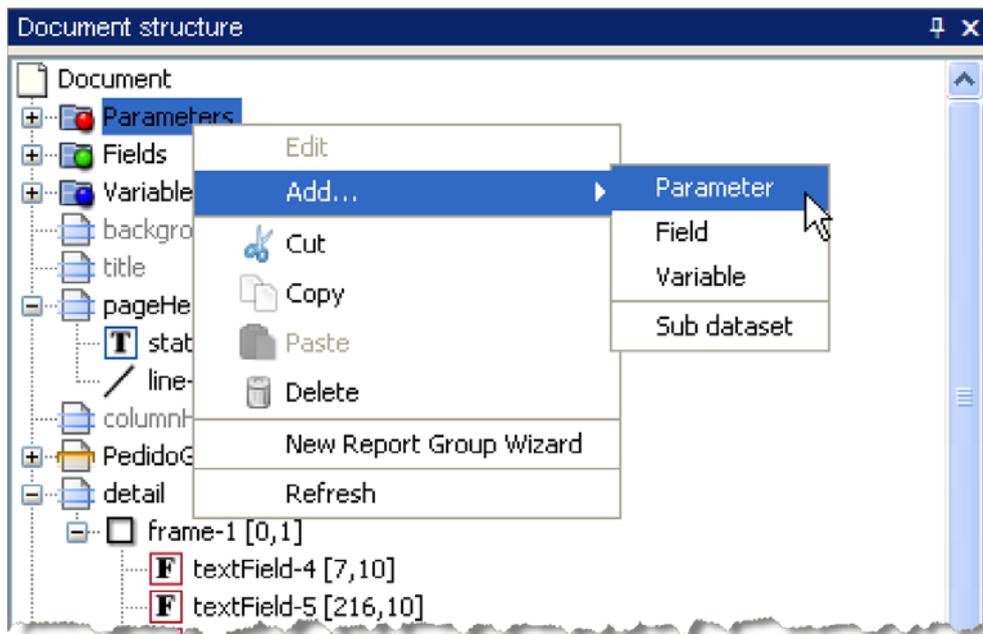


Figura 22– Adicionando parâmetros pelo painel Document structure

Na caixa de diálogo **Add/modify parameter** digite **PAR_PEDID**. Selecione **java.lang.Integer** em **Parameter Class Type**. Isso significa que até o momento você terá um parâmetro chamado **PAR_PEDID** com o tipo **Integer**. Como precisamos testar o relatório, você vai selecionar o item **Use as a Prompt**. Isso fará com que um prompt apareça para que você digite o código do pedido para testá-lo.

Existe um campo chamado **Default Value Expression**. É a opção padrão para o valor do prompt, caso não se lembre de nenhum código de pedido. Neste caso, você vai adicionar o número do pedido **10248** como padrão. Por se tratar de um **int**, converta para **java.lang.Integer**, digitando neste campo o detalhe a seguir:

```
new Integer( 10248 )
```

Caso queira completar com uma descrição para o parâmetro criado, basta adicioná-lo no campo **Parameter Description**, como é mostrado na **Figura 23** e confirme. Esta descrição aparecerá na caixa de diálogo que será exibida quando chamar o relatório pelo iReport.

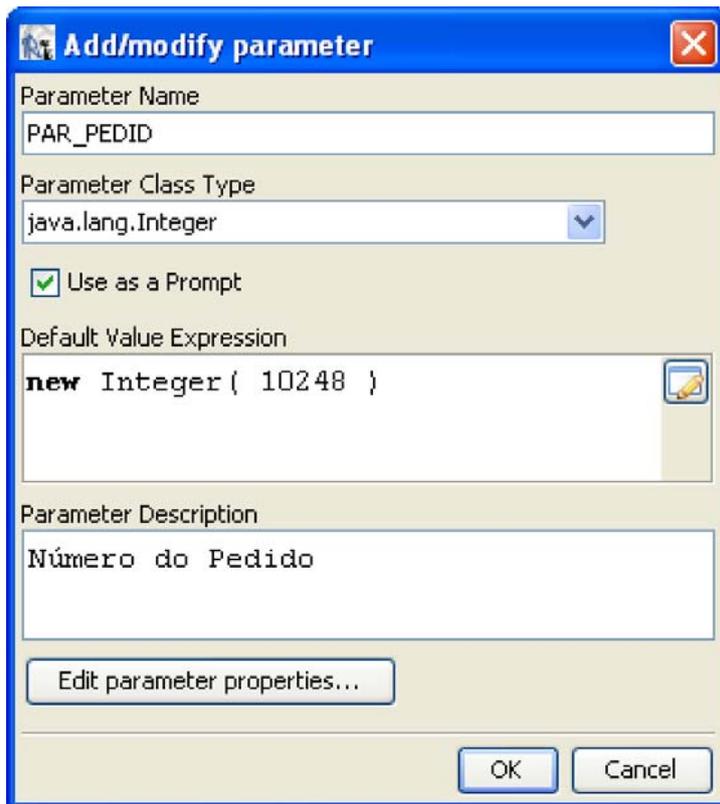


Figura 23 – A adição do parâmetro PAR_PEDID

Resta agora adicionar este parâmetro á query. Vá ao menu **Data** e clique na opção **Report Query**. Na caixa de diálogo **Report query** clique no botão **Query Designer**.

Na caixa de diálogo **SQL Query Designer**, no quadro a direita, na tabela **pedidos**, clique com o botão direito do mouse sobre o campo **PedidoID**. No menu de contexto (veja **Figura 24**), selecione o item **add where condition**.



Figura 24 – Adicionando um campo à condição SQL WHERE pelo SQL Query Designer

Ao surgir a caixa de diálogo **condition.edit**, a parte superior estará preenchida com o campo **PedidoID**. Resta apenas manter a condição de igualdade e digitar na parte inferior o parâmetro criado. O parâmetro começa com **\$P{PAR_PEDID}**. Sua caixa de diálogo se parecerá com a da **Figura 25**.

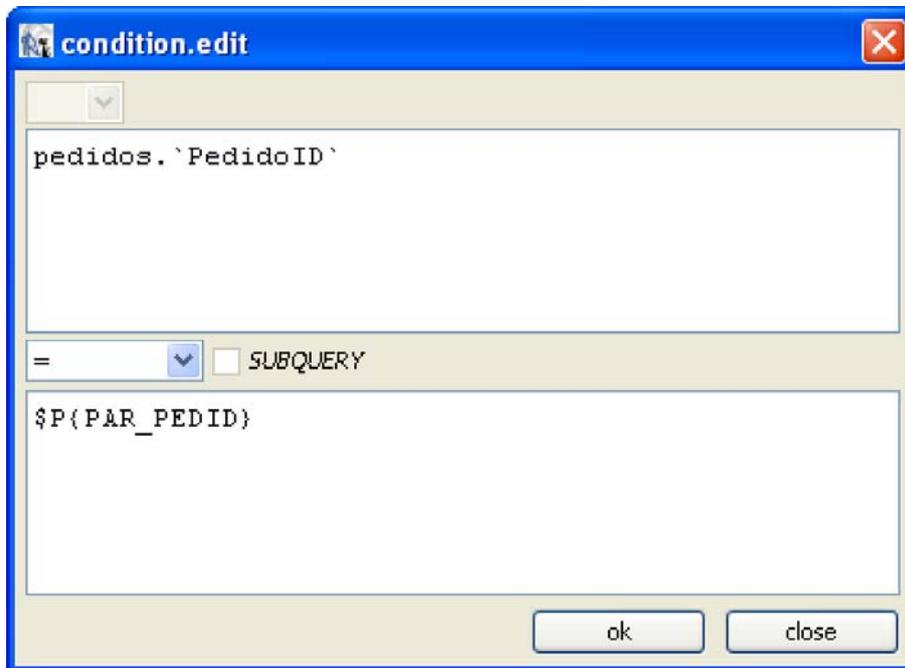


Figura 25 – O parâmetro adicionado a condição de pesquisa sobre o número de pedido

Após a confirmação, você verá na construção da query a condição **WHERE** contendo a comparação do número do pedido com o parâmetro (**Figura 26**).

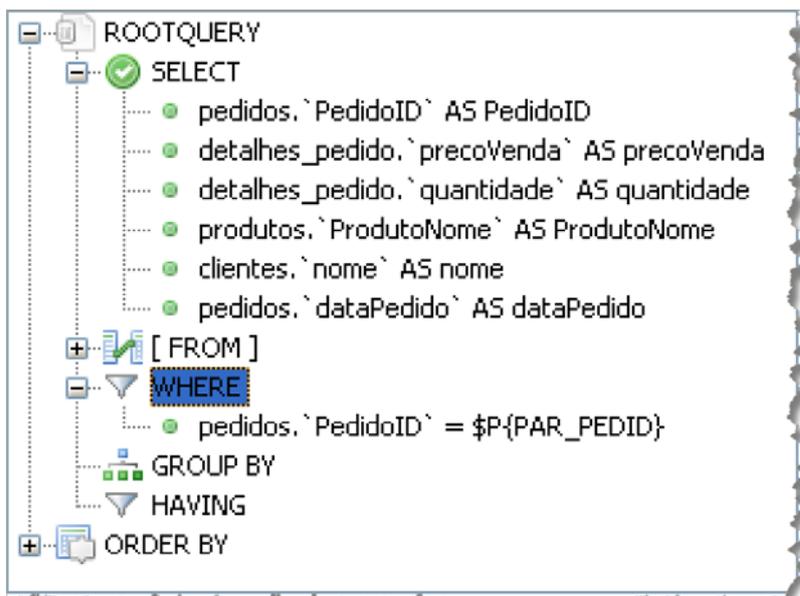


Figura 26 – Condição WHERE configurada com o parâmetro

Voltando a caixa de diálogo **Report query** você terá a sua instrução SQL com a condição WHERE contendo o PedidoID comparada ao parâmetro **\$P{PAR_PEDID}**. Confirme a caixa de diálogo.

...

WHERE

```
pedidos.`PedidoID` = $P{PAR_PEDID}
```

...

Execute o relatório. Observe que antes de exibir o relatório, o iReport exibe um prompt chamado de **Parameter prompt**. O campo já vem preenchido com o valor padrão, esperando digitar um novo número. Abaixo aparece o tipo aceito e a descrição, caso tenha feito, do parâmetro. Mesmo que apague o parâmetro padrão, se precisar testar o relatório com este número, clique no botão **Use default**.

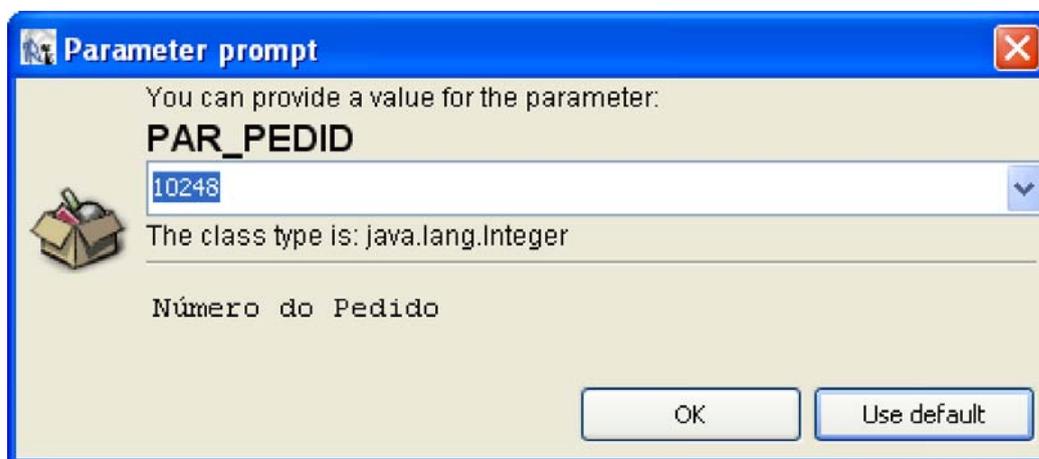


Figura 27 – Parameter prompt com o código padrão e a descrição abaixo

Chamando o relatório por uma aplicação desktop

Assim como criar um relatório, é muito fácil chamá-lo por uma aplicação. O primeiro exemplo utilizará uma aplicação desktop simples, sem interface gráfica, apenas com o intuito de ensiná-lo a chamar sua aplicação.

Para compilar sua aplicação, você vai precisar de sete arquivos **JARs**, **por enquanto**, listados a seguir:

1. commons-beanutils-1.7.jar
2. commons-collections-2.1.jar
3. commons-digester-1.7.jar
4. commons-javaflow-20060411.jar
5. commons-logging-api-1.0.2.jar
6. itext-1.3.1.jar
7. JasperReports-2.0.2.jar

Este sete arquivos se encontram no diretório **lib** do iReport. É claro que como você está trabalhando com banco de dados, haverá a necessidade de ter o JAR da ponte JDBC do seu banco de dados. Como no livro estamos usando o MySQL, você pode utilizar o que está dentro do diretório **lib** do próprio iReport ou pegar um mais novo.

A versão da biblioteca JDBC do MySQL no iReport é antiga, e no momento em que este livro é escrito, é representado pelo arquivo **mysql-connector-java-3.1.11-bin.jar**.

Caso você queira trabalhar com uma versão mais moderna, entre no endereço a seguir:

<http://dev.mysql.com/downloads/>

Procure pelo **MySQL Connector/J**. Você terá a versão 5.0, que pode ser baixada compactada em formato .ZIP ou Tarball. Dentro do arquivo compactado, você extrai o arquivo **mysql-connector-java-5.0.7-bin.jar**.

O código para executar o relatório

O código para executar o relatório e gerar sua saída é uma chamada ao arquivo compilado, no caso, chamado de **Relatorio_Agrupado.jasper**.

A seguir você tem o código na integra da classe Main.

Main.java

```
package br.com.integrator;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.HashMap;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;

public class Main {

    public static void main(String[] args) {
        Connection conn = null;

        try {
```

```
//cria a conexão com o banco de dados
Class.forName("com.mysql.jdbc.Driver");
String db = "jdbc:mysql://localhost:3306/relatorios";
conn = DriverManager.getConnection(db,"edson","integrator");

System.out.println("Gerando relatório...");

HashMap<String, Integer> parameterMap =
    new HashMap<String, Integer>();
//o Nome do parâmetro e o valor é passado ao HashMap
parameterMap.put("PAR_PEDID", 10248);

//pega o caminho físico até o arquivo .jasper
String arquivo = System.getProperty("user.dir") +
"/src/br/com/integrator/relatorio/Relatorio_Agrupado.jasper";

//chama fillReport
JasperPrint jp = JasperFillManager.fillReport(arquivo,
    parameterMap, conn);

//exibe o relatório com viewReport
JasperViewer.viewReport(jp, false);

    } catch (SQLException ex) {
ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
ex.printStackTrace();
    } catch (JRException e) {
e.printStackTrace();
    } finally {
try {
    if (!conn.isClosed()) {
conn.close();
    }
    System.out.println("Finalizado!");
} catch (SQLException ex) {}
}
}
```

```
}  
}
```

O processo de gerar um relatório no JasperReports, seja por um template ou por um arquivo Jasper, é conhecido como “*filling*”. O relatório é “ocupado” programaticamente, por um método chamado **fillReport()**, de **net.sf.jasperreports.engine.JasperFillManager**. Esta “ocupação” é feita por três parâmetros:

1. **Arquivo** – o arquivo é colocado em um caminho físico ou relativo, depende da situação, para que seja “montado” o relatório. No caso, o caminho físico foi obtido por um relatório compilado;
2. **Parâmetro** – o parâmetro passado é um **java.util.HashMap()**, que define se haverá um ou mais parâmetros. O parâmetro criado no relatório possui um nome, e é este que deve ser utilizado neste HashMap: **PAR_PEDID** (referindo-se a **#{PAR_PEDID}**). Como se trata de uma chave string e de um valor numérico inteiro, graças ao método *put(Key, Value)*, o exemplo passa o pedido de número **10248** da seguinte forma: **parameterMap.put("PAR_PEDID", 10248)**. Caso não haja parâmetros, um **HashMap()** vazio é passado ao método **fillReport**:

JasperFillManager.fillReport(arquivo, new HashMap(), conn);

3. **Conexão** – a conexão com o banco de dados é transmitida no terceiro parâmetro. Em caso de não haver conexão alguma, por não precisar de acesso a banco de dados, este terceiro parâmetro recebe o método **JREmptyDataSource()** de **net.sf.jasperreports.engine.JREmptyDataSource**.

Capítulo 6

Desenvolvendo Relatórios com NetBeans IDE

Este capítulo o instruirá no desenvolvimento de relatórios JasperReports básicos utilizando o novo plugin iReport for NetBeans, que o NetBeans IDE 6 agora possui. No momento em que este livro é escrito, sua versão ainda não é final.

Nota: Este capítulo é parte integrante do livro *Desenvolvendo aplicações Web com NetBeans IDE 6*, portanto, os exemplos aqui ditos são baseados em outras tabelas.

Desenvolvendo relatórios com NetBeans

O iReport é um bem-sucedido programa Open Source, capaz de criar visualmente os mais complexos relatórios para aplicações Java no formato da biblioteca JasperReports.

Atualmente, além de ser um programa gráfico independente, há também uma versão disponível que se integra ao NetBeans IDE 6.0.

Através de uma interface gráfica e intuitiva, o desenvolvedor é capaz de criar qualquer tipo de relatório de forma simples e rápida. O iReport para NetBeans desenvolve um formato XML usado pelo JasperReports, evitando naturalmente a edição no código.

Neste capítulo você aprenderá onde obter, como instalar, criar seu relatório e integrá-lo a suas aplicações Web conhecendo algumas características do iReport.

Obtendo e instalando o iReport para NetBeans

Para obter o iReport para NetBeans, primeiramente você precisará baixar o plugin. Para isso, o endereço oficial para baixá-lo é: <http://plugins.netbeans.org/PluginPortal/faces/PluginDetailPage.jsp?pluginid=4424>. Clique no botão **Download**. Você obterá o arquivo **.nbm** para instalação.

Atenção: No momento em que este livro está sendo escrito, este plugin ainda se encontra em versão **BETA**, sem algumas funcionalidades encontradas no iReport.

Vá ao menu **Tools** e clique no item **Plugins**. Na caixa de diálogo **Plugins**, vá à guia **Downloaded** e clique no botão **Add Plugins**, selecionando, através da caixa de diálogo o arquivo de extensão **.nbm** pertencente ao iReport para NetBeans. Ao fazê-lo, clique no botão **Install**.

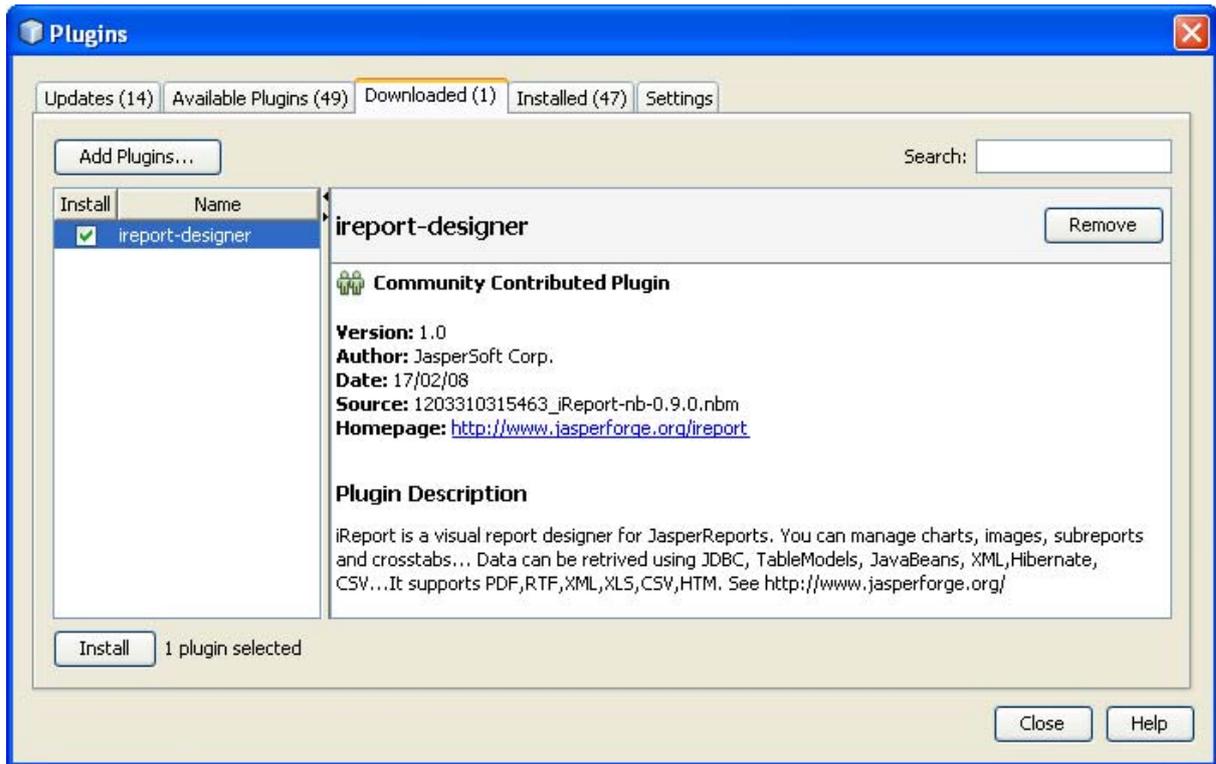


Figura 1 – Adição do plugin ireport-designer

Ao clicar no botão **Install**, a caixa de diálogo **NetBeans IDE Installer** surgirá. Clique no botão **Next** para prosseguir com a instalação.

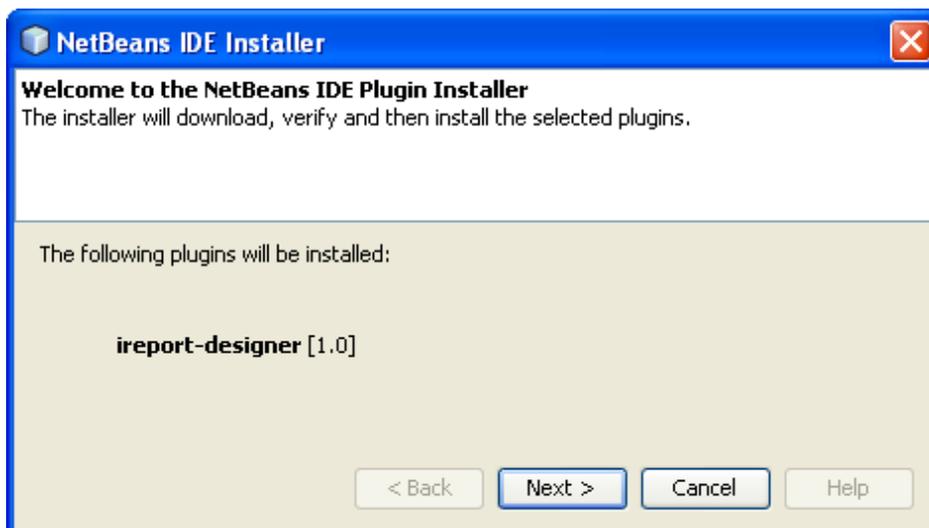


Figura 2 – Início da instalação do plugin ireport-designer

Na etapa seguinte, você deverá aceitar os termos de licença. Clique na opção **I accept the terms in all of the license agreements** e clique no botão **Install**.

A caixa de diálogo **Validation Warning** surgirá, dizendo que o plugin a ser instalado não foi assinado e que é potencialmente inseguro. Clique em **Continue**.

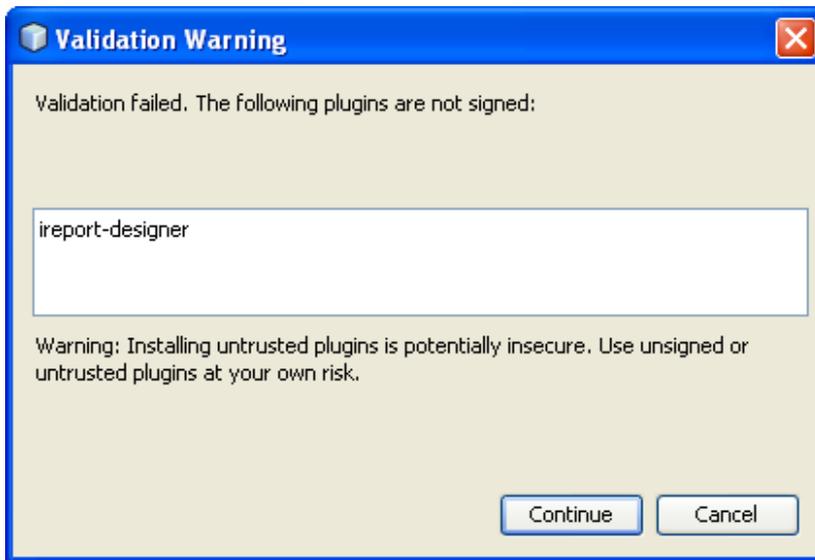


Figura 3 – Validação do plugin

Ao terminar a instalação, clique no botão **Finish**. Retornando a caixa de diálogo **Plugins**, observe, na guia **Installed**, que o plugin **ireport-designer** está instalado. Clique no botão **Close** para fechar a janela.

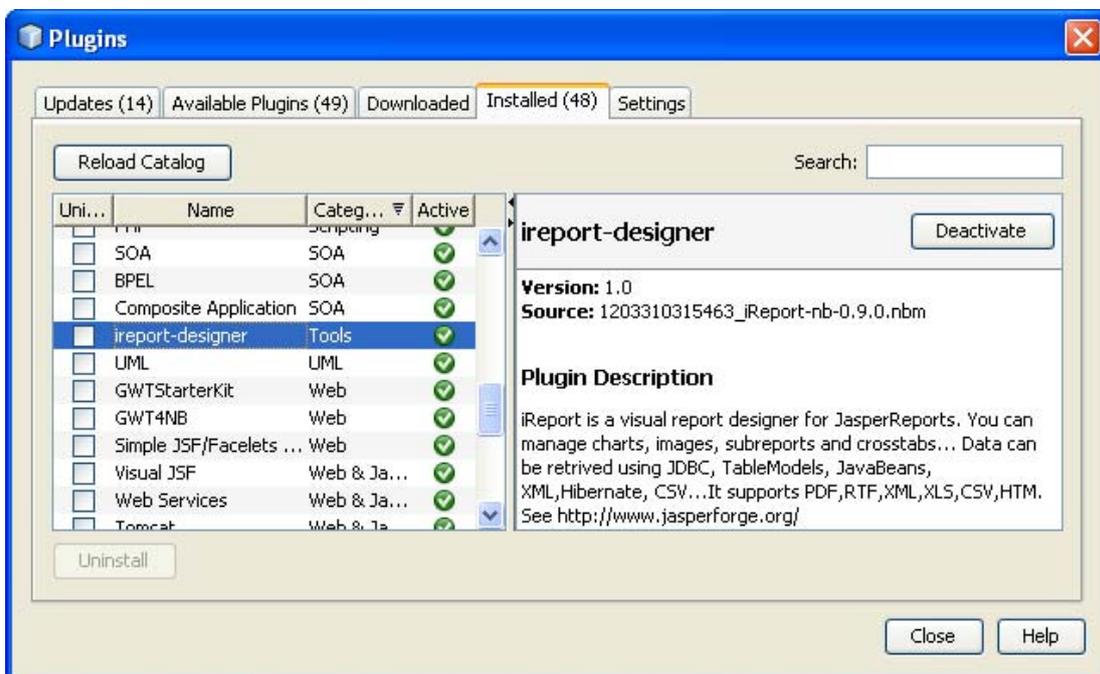


Figura 4 – Plugin instalado

Criando um relatório com o plugin

Crie um projeto, pode ser Web. Em **Web Pages**, por exemplo, crie um diretório chamado **relatorios**. Clique com o direito neste diretório e no menu de contexto selecione **New** e clique em **Other**. Na caixa de diálogo **New File** selecione **JrxmlTemplate.jrxml**, em **Reports**. Em alternativa, caso esteja aparecendo a opção **JrxmlTemplate.jrxml** no menu **New**, selecione-a. Na segunda etapa, digite o nome do seu relatório. Se preferir seguir o livro, chame-o de **relatório** e conclua o assistente.

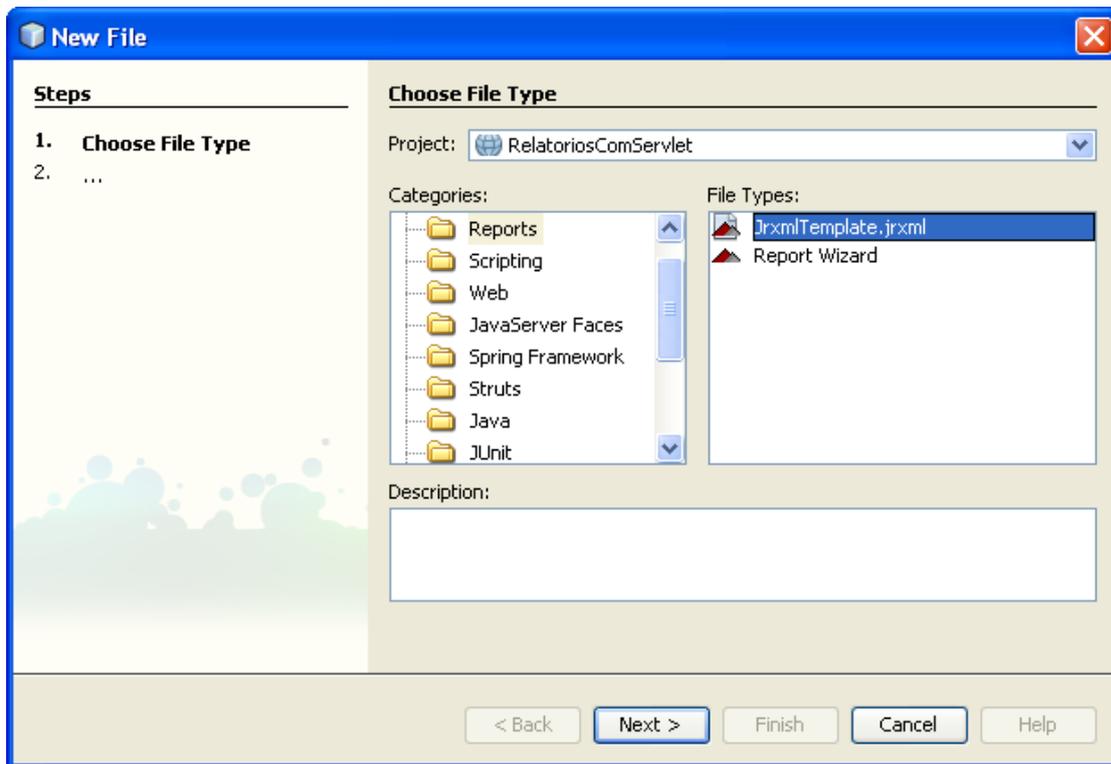


Figura 5 – Criando um documento JasperReports

A estrutura e propriedades de um relatório JasperReports

Ao finalizar o assistente, o **Designer** do iReport para NetBeans surgirá. Além do Design, você possui a possibilidade de visualizar em formato **XML** e fazer um **Preview**.

Um relatório no JasperReports possui diversas seções, separadas por linhas horizontais ao qual são chamadas de **Bands (bandas)**.

O comportamento de cada Band depende da funcionalidade para a qual está preparada. Quando o relatório junta aos dados para fazer a impressão, estas seções são impressas na tela de diferentes maneiras com diferentes tempos. Por exemplo, o cabeçalho da página (**pageHeader**) é repetido em todas as páginas, uma vez em cada página, enquanto que os detalhes onde serão exibidos os dados (**detail**) é gerado em uma seqüência elaborada de acordo com os dados existentes (**query** ou **datasource**), repetindo-se uma vez para cada registro, incorporando-se ao desenho do relatório.

As bands de um relatório JasperReports é dividido em dez seções pré-definidas para que novos grupos possam ser adicionados.

Embora as bands já estejam adicionadas e com espaçamentos pré-definidos inicialmente, você pode com o cursor, entre cada uma delas, nas linhas horizontais que as separam, e movê-las.

Ao arrastar, você percebe que a altura de cada band se expande, empurrando as demais, após a alteração, para baixo ou vice-versa. Embora o desenvolvedor possa alterar a sua altura, em casos como a band **detail**, sua alteração implica na visibilidade e espaçamento dos elementos impressos na geração do relatório. Porém, nem todas as bands são organizadas dinamicamente pelo conteúdo, como é o caso de **Column Footer**, **Page Footer** e **Last Page Footer**.

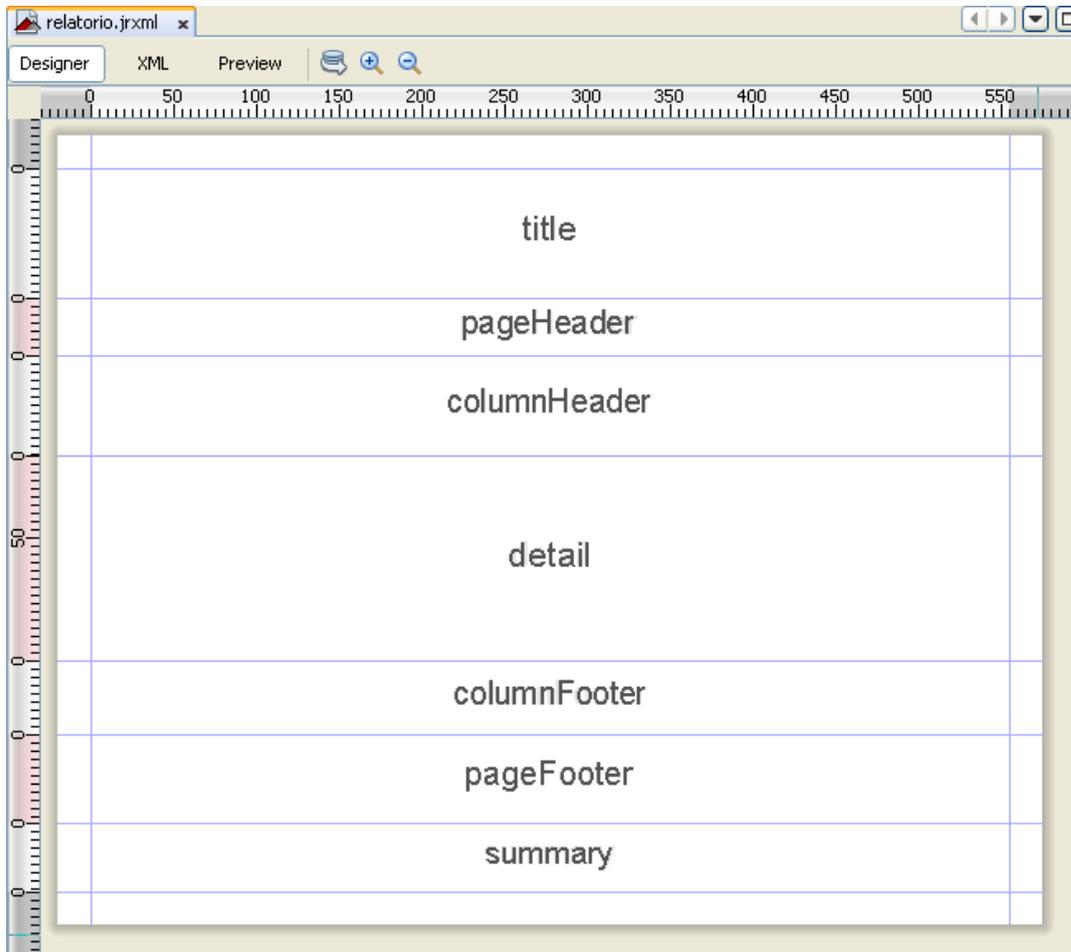


Figura 6 – Estrutura do relatório JasperReports

Na janela **Outline Window**, você encontra os parâmetros (Parameters), campos da tabela da sua query (Fields), variáveis pré-definidas e criadas (Variables) e as bands do relatório. Por padrão, algumas bands estão desativadas ou com sua altura no valor **zero**, o que a torna invisível no design (como a band **background**).

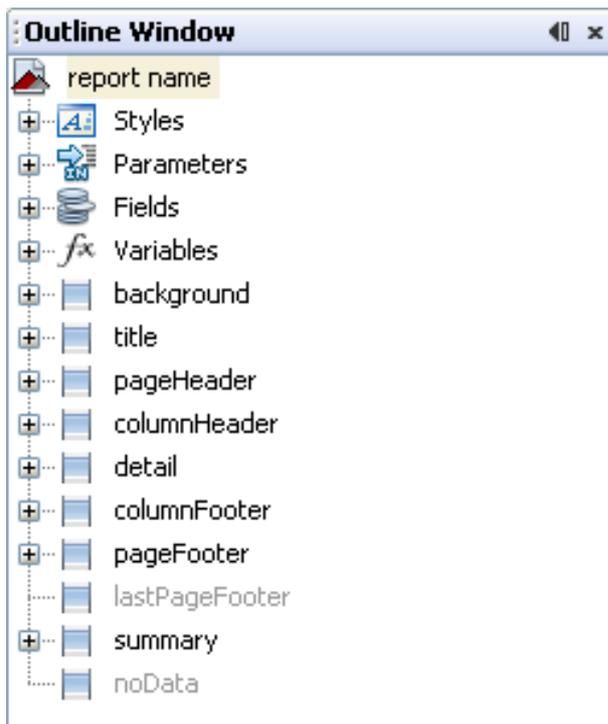


Figura 7 – A janela Outline e os componentes do relatório

Na janela **Palette** você encontra os elementos de um relatório.

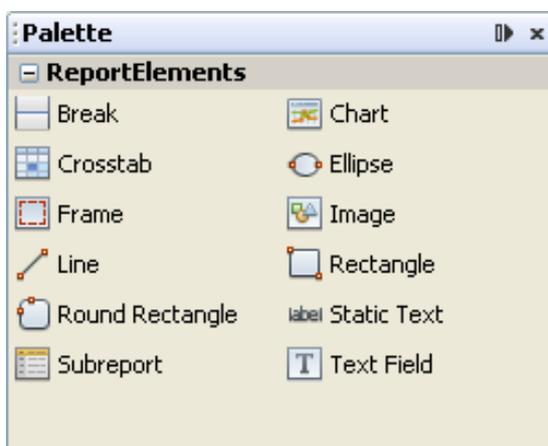


Figura 8 – A janela Palette e os ReportElements

Seguindo a imagem da **Figura 8** você tem as seguintes ferramentas:

- **Chart** – Gera um gráfico através de valores determinados transmitidos. Mais adiante haverá um capítulo exclusivamente para gráficos.
- **Image** – Utilizado para exibir imagens em seu relatório. Podem ser imagens dinâmicas (preenchidas por um banco de dados, por exemplo) ou estáticas.
- **Rectangle** – Usado para desenhar retângulos ao redor de outros elementos, criando destaques como uma formatação de parágrafo de um programa de edição de textos, este elemento pode também conter retângulos com cantos arredondados.

- **Text Field** – Utilizado para criar os campos dinâmicos dos relatórios. É neste elemento que você se conecta a um determinado campo do banco de dados para exibir suas informações, por exemplo.
- **Ellipse** – Desenha elipses no relatório.
- **Line** – Em um relatório JasperReports uma linha é definida por um elemento retangular com uma linha diagonal. A linha pode ser desenhada em uma reta perfeita, na horizontal, vertical ou em ângulo se pressionada a tecla <Shift> antes de arrastar. A linha segue o grid.
- **Static Text** – Utilizado para criar rótulos de relatórios.

Na parte inferior, da janela **Palette** se encontra as **Properties**. Quando um elemento é selecionado aparece nesta janela suas propriedades para serem alteradas.

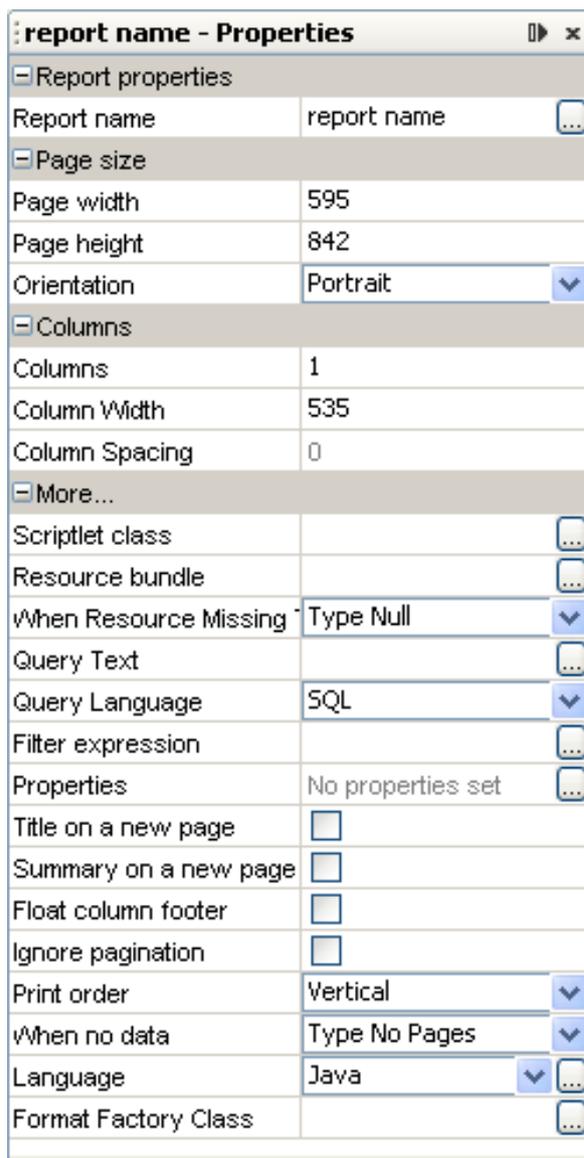


Figura 9 – A janela Properties do relatório

Nas propriedades do relatório, você possui o grupo **Page size**. Neste grupo você pode definir a largura da página (**Page width**) e a altura da página (**Page height**), bem como a Orientação (**Orientation**). Em **Orientation**, temos **Portrait** (Retrato) ou **Landscape** (Paisagem).

No grupo **Columns** você pode definir quantas colunas seu relatório possuirá em uma página, no item **Columns**. **Column Width** automaticamente se altera quando você adiciona mais de uma coluna, dividindo sem o valor em partes iguais. **Column Spacing** determina o espaçamento entre colunas, que também influi em **Column Width** automaticamente.

No grupo “**More...**” é possível especificar instruções para a impressão do relatório. Nele se encontram:

- **Scriptlet Class** - Onde você define um scriptlet, que nada mais é que uma classe Java cujos métodos são executados conforme são especificados os eventos durante a criação do relatório, como o início de uma nova página ou o fim de um grupo.
- **Resource Bundle** - É o nome do pacote de recursos usado para internacionalizar um relatório. É o Resource Bundle do Java, onde você define em um arquivo os textos traduzidos como rótulos. Cada linguagem corresponde a um arquivo específico.
- **When Resource Missing Type** – Se um pacote de recurso (Resource Bundle) não está disponível, você escolhe uma das opções na caixa de combinação. Representa o atributo **whenResourceMissingType** no JasperReports. Neste caso, você tem as seguintes opções:

1. **Type Null (Nulo)** – Imprime a string “*Null*”. Esta é a opção padrão.
2. **Type Empty (Vazio)** – Não imprime nada. Recebe como valor *Empty* no atributo **whenResourceMissingType** de JasperReports.
3. **Type the Key (Chave)** – Imprime o nome chave que falta. Recebe como valor *Key* no atributo **whenResourceMissingType** de JasperReports.
4. **Rise an Error (Erro)** – Lança uma exceção parando o processo. Recebe como valor *Error* no atributo **whenResourceMissingType** de JasperReports.

- **Query Text** – Determina a query criada para o relatório.
- **Query Language** – Determina a linguagem utilizada para a instrução. Pode ser: **SQL**, **HQL(Hibernate Query Language)**, **XPath**, **EJBQL**, **MDX** e **XMLA-MDX**.
- **Properties** – Determina as propriedades para o objeto selecionado.
- **Title on a new page** - Quando ativa, especifica se a banda **title** será impressa e em seguida criada uma nova página, gerando uma “quebra de página”.
- **Summary on a new page** - Esta opção é similar a anterior (**Title on a new page**), mas com a diferença de ser a banda **summary** que será impressa como a última página.
- **Floating column footer** - As bandas **columnFooter** são impressas por padrão no fim das páginas. Se na impressão do relatório não houver dados suficientes para preencher uma página, um espaço em branco aparecerá entre a coluna e o rodapé da coluna (**columnFooter**). Se você quer que o texto encontrado na banda **columnFooter** seja impresso logo após a impressão da coluna, marque a opção **Floating column footer**. Esta opção permite forçar a impressão da banda **columnFooter** imediatamente após a última interação da banda **detail** e não do término de todas as colunas. Esta opção é geralmente usada quando você quer criar tabelas usando os elementos de relatórios.
- **Ignore pagination** - Ignora a paginação ao preencher o relatório, impossibilitando a visualização das demais páginas caso haja mais que uma. Quando ativa, esta opção implica até mesmo na visualização correta dos dados caso haja mais de uma coluna.
- **Print Order** - A ordem de impressão (Print Order) determina como os dados impressos na página serão organizados em mais de uma coluna. O padrão para a ordem de impressão é **Vertical**. Com a ordem **Vertical**, primeiro os dados são impressos um embaixo do outro, passando para uma nova

coluna somente quando a coluna está totalmente preenchida no final da página. Na ordem **Horizontal** os dados são preenchidos em linha. Os dados passam por todas as colunas, adicionando uma nova linha somente quando a última coluna da página foi preenchida naquela determinada linha.

• **When no data** - Quando o número de dados para a impressão é vazio (query SQL sem dados, por exemplo), o relatório pode ter comportamentos diferentes para apresentar a página ao usuário. Neste caso existem quatro opções, listadas a seguir:

1. **Type No Pages (Sem Páginas)**– O padrão. O resultado final é um buffer vazio.
2. **Type a Blank Page (Páginas em branco)** – Exibe uma página em branco caso não haja dados.
3. **Type All Sections, No Detail (Todas as seções, nenhum detalhe)** – Mostra todas as seções existentes no relatório, menos os detalhes da banda **detail**.
4. **Type “No Data” section** – Seção sem dados. Utiliza a band **noData**, ou seja, você adiciona um conteúdo estático (ou dinâmico sem usar o banco de dados) neste local. Caso não haja dados em uma determinada pesquisa ou sem conexão com o banco de dados, esta seção será exibida.

• **Language** – Determina a linguagem a ser utilizada no relatório. Neste caso, somente as duas linguagens oficiais do JVM são possíveis, por enquanto: **Java** ou **Groovy**.

• **Format Factory Class** - Recebe uma string com o nome da classe de formatos, habilitando o uso de datas personalizadas e formatos de números. Representa o atributo **formatFactoryClass** de JasperReports e implementa a interface *net.sf.jasperreports.engine*.

Exibindo o Grid

Você pode exibir o Grid para desenhar o relatório, vá ao menu **Designer** e clique em **Show Grid**. Também é possível alinhar os elementos dentro do relatório se selecionar a opção **Snap To Grid**.



Figura 10 – Exibindo o Grid no Design

Adicionando uma conexão a um relatório em branco

Para a construção deste relatório, vamos utilizar o banco de dados livraria. Vá ao menu **Data** e clique em **Connections/Data Sources**, ou na barra de ferramentas **Designer**.



Figura 11 – Connections / Datasources no menu Data e ferramentas Designer

Na caixa de diálogo **Connections / Datasources**, clique no botão **New**. Em **Datasource** selecione o item **NetBeans Database JDBC connection** e clique no botão **Next**.

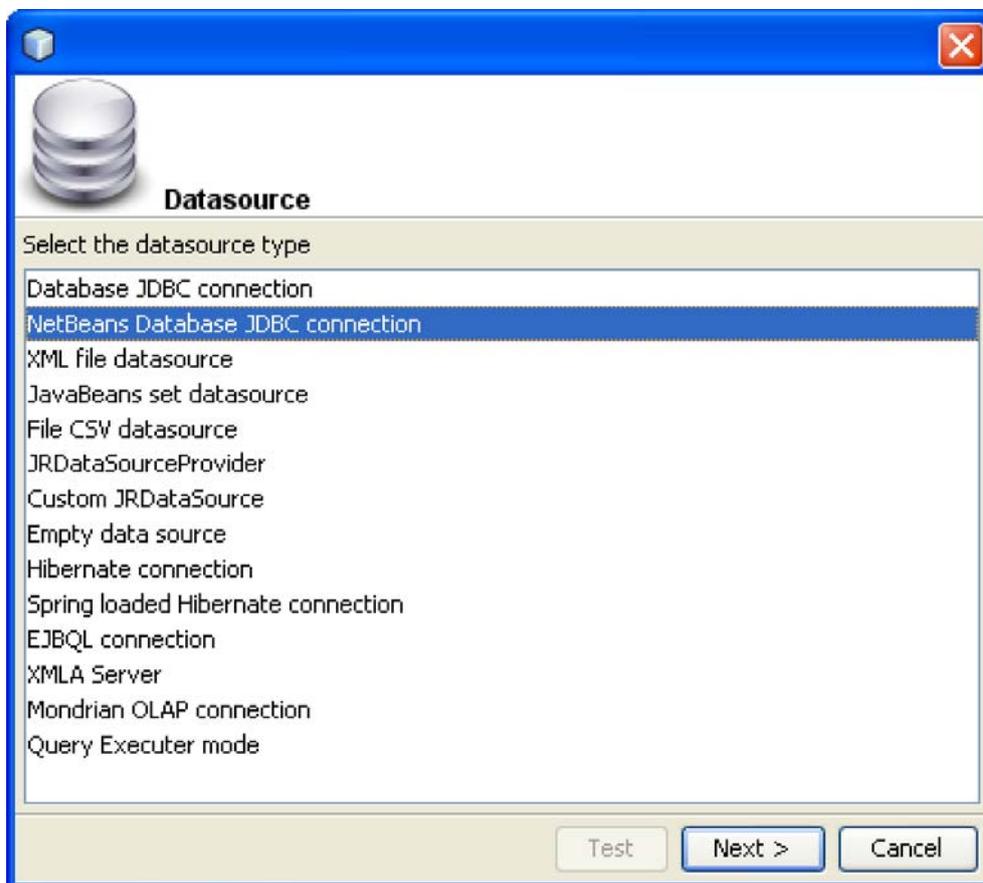


Figura 12 – Seleção do datasource NetBeans Database JDBC connection

Na segunda etapa, selecione em **Connection** a conexão **livraria** usada neste livro e digite **livraria** em **Name**. Clique no botão **Test**. Se a caixa de diálogo “**Connection test successful!**” surgir, confirme e clique no botão **Save**.

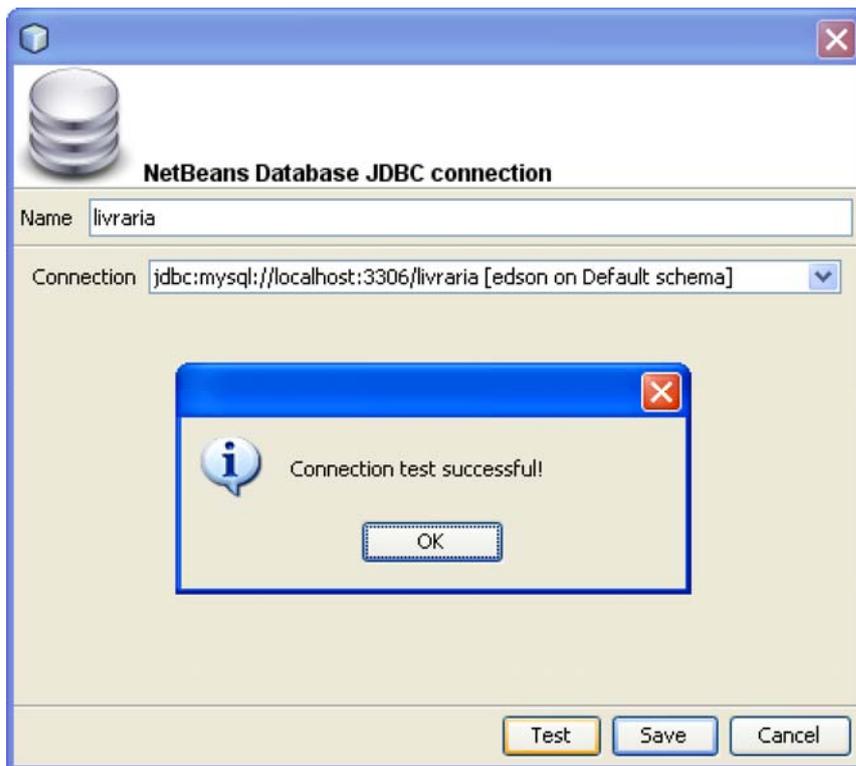


Figura 13 – Configurando a conexão livraria

Para uma modificação da conexão, na caixa de diálogo **Connections / Datasources** temos o botão **Modify**. Basta selecionar a linha da sua conexão que deseja alterar e clicar neste botão. A caixa de diálogo **de propriedades da conexão** reabre. Basta alterar e salvar (Save).

Em mais de uma conexão, você tem o botão **Set as default**. Basta selecionar a conexão que deseja tornar padrão (Default) e clicar neste botão. Observe que na caixa de diálogo a coluna Default apresenta a conexão padrão com uma caixa de checagem ativa.

Caso você precise levar as configurações da sua conexão para outra máquina, ou quer guardá-la para uma futura reutilização, clique no botão **Export**. A caixa de diálogo para salvar a conexão aparecerá. Basta selecionar o local e dar o nome. Caso tenha mais de uma conexão, todas serão exportadas. Se precisar importar uma conexão do próprio iReport, basta clicar no botão **Import**. Depois de configurado, basta fechar a caixa de diálogo.

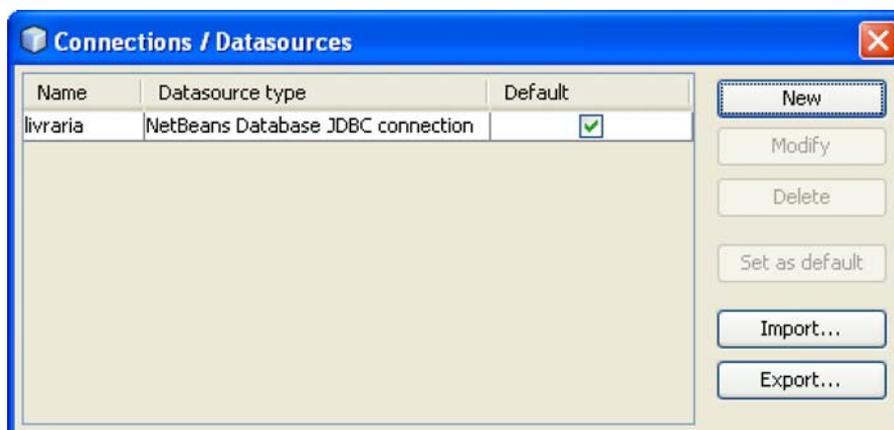


Figura 14 – Conexão criada e disponível

Criando a query do relatório

Para criar a query do relatório, vá ao menu **File** e clique no item **Report query**. Alternativamente você pode ir ao ícone **Report query**, acima do design do relatório.



Figura 15 – Abrindo o Report query

Ao abrir a caixa de diálogo **Report Query**, clique no botão **Query designer**. Outra caixa de diálogo surgirá, chamada de **SQL Query Designer**, mas desta vez para a criação da instrução SQL visual.

Observe que há três quadros na janela. O primeiro, à esquerda, define a exibição da query de forma visual, separando os elementos padronizados de uma seleção de dados (SELECT, FROM, WHERE e etc.). A cada coluna de sua tabela selecionada, automaticamente a query adicionará na parte superior, em SELECT, o campo e um alias. Em FROM surgirá a, ou, as tabelas que estão relacionadas para tal seleção.

Você vai selecionar, através do segundo quadro a esquerda, a tabela **autores**, dando um duplo clique sobre seu respectivo nome. Esta seleção pode ser através do arrastar para o quadro da direita também, caso ache mais prático.

Existem duas formas de você selecionar todos os campos de uma tabela, no Query designer. A primeira é ir até a tabela (no caso autores) e clicar no menu de mesmo nome, selecionando a opção **select all**. A segunda forma é clicando com o direito do mouse e selecionando a opção **add expression**, no quadro da esquerda, em **SELECT**.

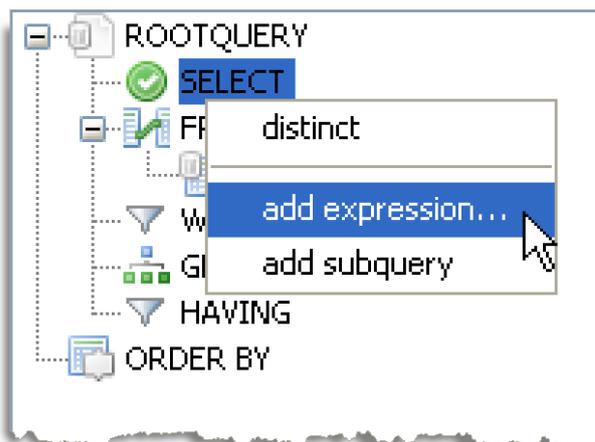


Figura 16 – Adicionando uma expressão

Neste caso, você digita o que deseja fazer, na caixa de diálogo **expression.edit**. Para o caso, digite “*” e confirme.

Ao final, você possuirá uma configuração da sua query SELECT similar ao mostrado na **Figura 17** a seguir:

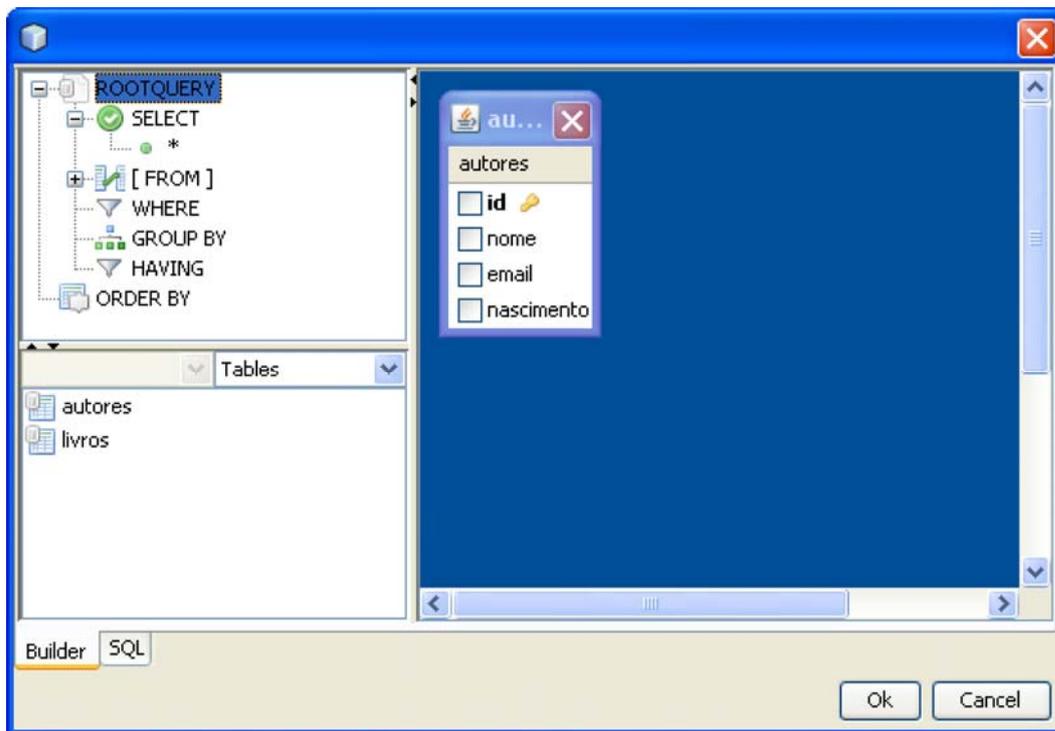


Figura 17 – Query desenhada na caixa de diálogo SQL Query Designer

Se você clicar em **SQL**, a guia abaixo, ao lado de **Builder** (a atual), você verá a **query** construída. Confirme no botão **OK**.

Retornando a caixa de diálogo **Report query**, você ainda pode fazer alterações na query criada.

Na parte superior da janela, existe o botão **Save query**, caso deseje armazenar a consulta construída para uso posterior em outro relatório. Se esse for o caso, o carregamento de uma consulta armazenada pode ser feito através do botão **Load query**.

Como a opção **Automatically Retrieve Fields** está selecionada, na parte inferior aparece cada campo selecionado, em uma tabela, com seus respectivos nomes e o tipo de dados em Java de cada um. É importante manter selecionada esta opção, uma vez que na digitação ou alteração de uma query, possíveis erros podem ser encontrados graças à exibição dos campos nesta tabela. Caso esta opção não esteja selecionada, qualquer alteração na query SQL não reflete automaticamente abaixo, o que necessitará clicar no botão **Read Fields**. Confirme logo após no botão **OK**.

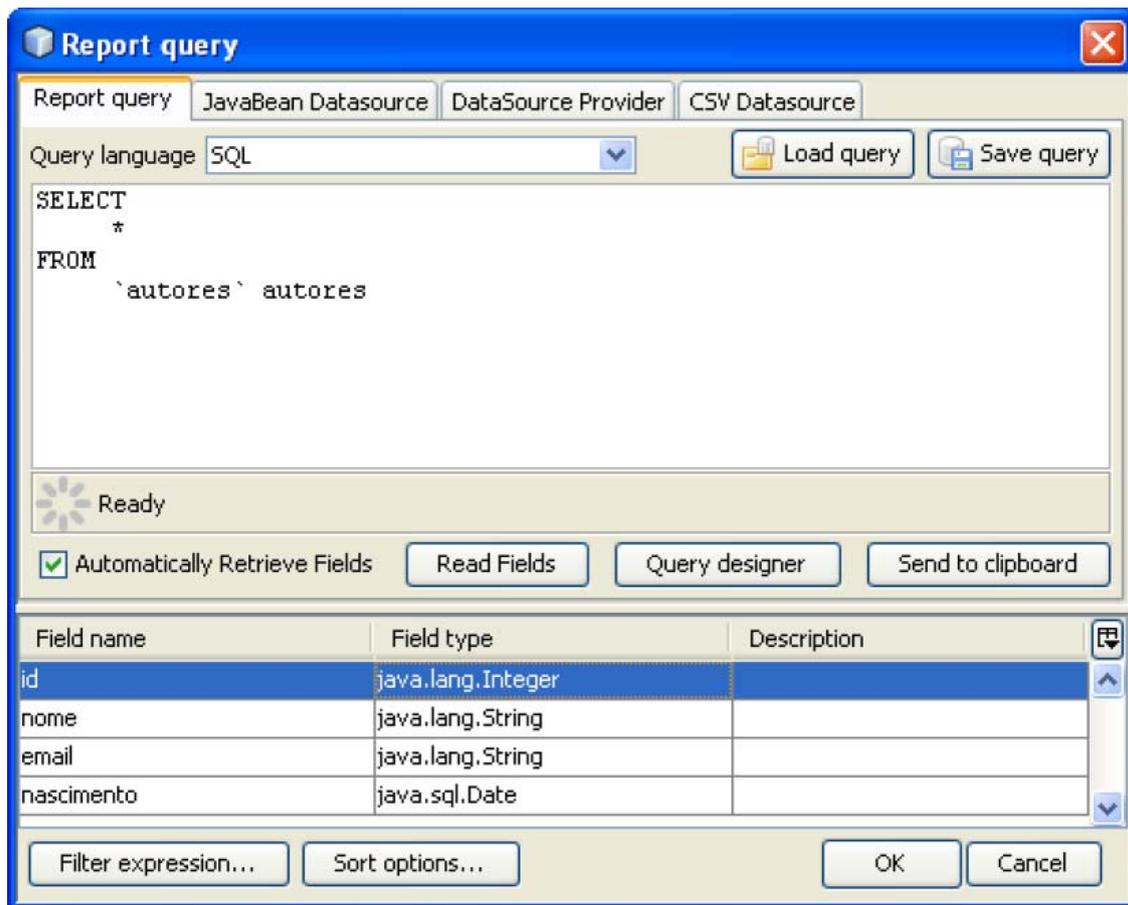


Figura 18 – Query final gerada pela caixa de diálogo SQL Query Designer

Desenhando o relatório

O desenho do relatório será uma etapa importante para a sua compreensão prática dos elementos existentes para se criar um e também sobre as bands (bandas) em sua geração.

Para começar na criação do desenho do relatório, você vai configurar inicialmente as bands que não serão exibidas. Com o cursor entre uma band e outra, você deve arrastar para cima, até que ela se encontre com a superior. Isso diminui a sua altura até zero, o que é exatamente o valor necessário para esta desaparecer.

As alturas de cada band para a criação do relatório é exibida conforme a **Tabela 2** mostrada a seguir:

Tabela 2 – Alturas das bands utilizadas na construção do relatório

Band	Altura da Banda
<i>title</i>	0
<i>pageHeader</i>	Mantenha como está
<i>columnHeader</i>	Mantenha como está
<i>detail</i>	Mantenha como está
<i>columnFooter</i>	0

<i>pageFooter</i>	Mantenha como está
<i>summary</i>	Mantenha como está

Atenção: Você pode configurar as propriedades de cada band selecionando-a através da janela **Outline Window** e alterando em **Properties**, no campo **Band height**. Neste caso, a alteração não precisa ser aplicada (alterou o valor da propriedade, confirmou com a tecla ENTER, mudou).

Adicionando os textos estáticos

Antes de iniciar a criação do desenho, você tem na **Figura 19** como será o resultado final.



Autores Cadastrados			
ID	Nome	E-mail	Nascimento
\$F{id}	\$F{nome}	\$F{email}	\$F{nascimento}
			Página:\$V

Figura 19 – Aparência final do relatório no Designer

Para criar a aparência vista na **Figura 19**, os seguintes passos serão feitos:

1. Na janela **Palette**, arraste o elemento **Static Text** (Label) e o dimensione a um tamanho qualquer na band **pageHeader**. Dê um duplo clique neste elemento desenhado e digite: **Autores Cadastrados**. Na janela **Properties** altere em **Font Size** para **24** e clique no botão **Bold**. Mude também para **Center** em **Horizontal Alignment** e para **Middle** em **Vertical Alignment**.
2. Na categoria **Box properties**, na janela **Properties**, com o label **Autores Cadastrados** selecionado, altere **Border** para **1 Point** e coloque **None** em **Left Border** e **Right Border**.
3. Adicione um elemento **Rectangle** na band **columnHeader** e altere a cor em **Background** a sua escolha, na janela **Properties**.
4. Adicione quatro elementos **Static Text** sobre o elemento **Rectangle** adicionado. Digite **ID**, **Nome**, **E-mail** e **Nascimento** respectivamente (ver **Figura 19**). Altere seu tamanho a sua escolha e coloque em **Bold**. Para alinhar, o editor visual do iReport para NetBeans possui linhas auxiliares para o alinhamento, herdado do **Sun's Visual Library API** do NetBeans.
5. Na janela **Outline**, expanda **Fields** e arraste cada campo para a band **detail**, no **Design**. Olhe a posição conforme a **Figura 19** apresenta. Dimensione a largura de cada componente a gosto. Observe os campos arrastados são **Text Fields**.



Figura 20 – Fields criados a partir da instrução SQL gerada

6. Selecione, no **Design**, o **Text Field** que representa o campo **nascimento**. Na janela **Properties**, vá até o item **Pattern** e de um clique no botão com três pontos ou em sua seleção, pressione **Ctrl+Space**. Na caixa de diálogo **Text field – Pattern**, selecione **Date** em **Category** e em **Type** selecione o formato de data usado no Brasil. Observe na parte inferior, em **Pattern**, o formato **dd/MM/yyyy**.

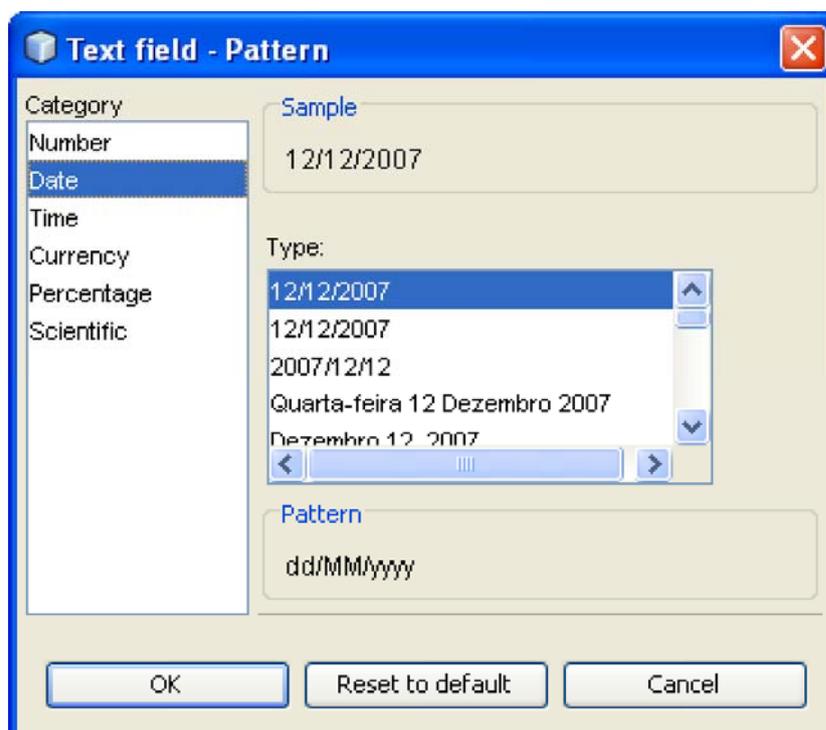


Figura 21 – Seleção do pattern para o formato de data

7. Expanda **Variables**, na janela **Outline**, e arraste para a band **pageFooter** e o posicione no canto direito.

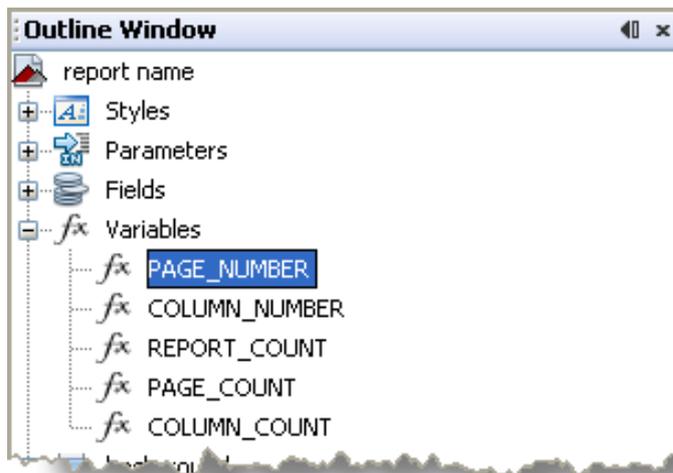


Figura 22 – A variável PAGE_NUMBER selecionada

8. Para finalizar, adicione um Static Text ao lado desta variável **PAGE_NUMBER** e digite **Página:**.

Clique em **Preview** para visualizar o relatório. Observe que os dados encontrados na band **detail** serão repetidos até o seu fim, criando uma nova página sucessivamente até a última linha encontrada em sua tabela. Na parte inferior você vê o número da página.

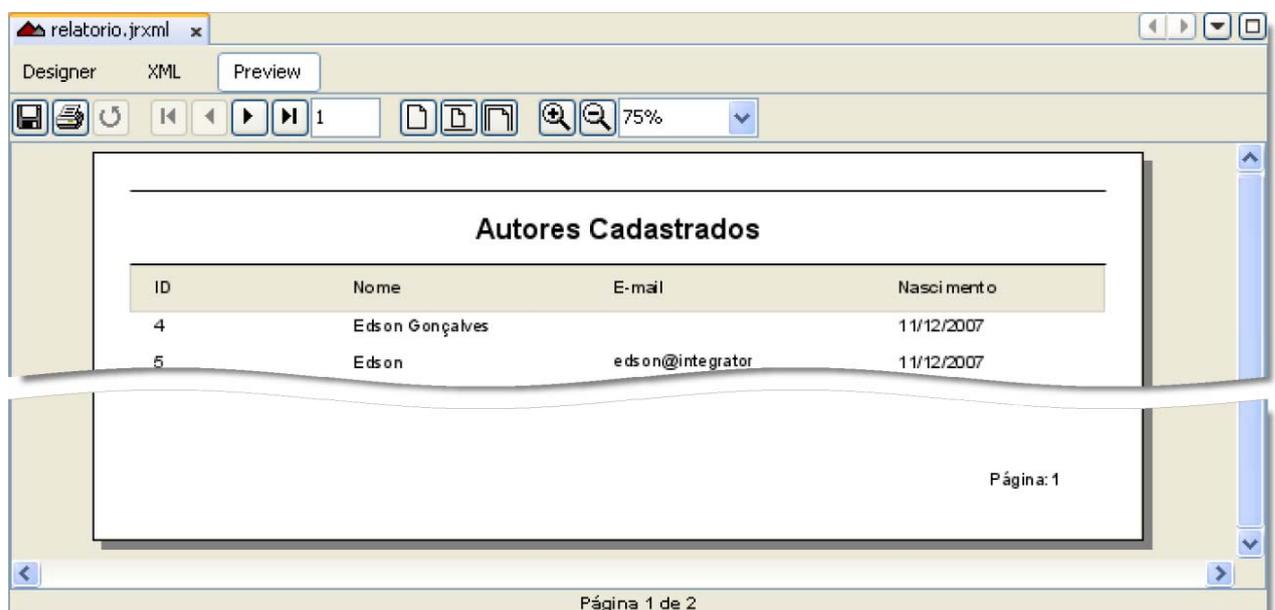


Figura 23 – Preview do relatório criado

Em caso de erro, observe a saída encontrada na parte inferior, na janela **iReport output**.



Figura 24 – Saída executada pelo iReport para NetBeans na compilação e exibição do relatório

Apêndice A

O MySQL

MySQL é um sistema de gerenciamento de banco de dados relacional, multiencadeado, de código-fonte aberto e nível corporativo.

O MySQL foi desenvolvido por uma empresa de consultoria na Suécia chamada inicialmente de TcX, depois, com a popularidade do MySQL, passou a se chamar MySQL AB.

Seu desenvolvimento ocorreu quando estavam precisando de um sistema de banco de dados que fosse extremamente rápido e flexível. Foi, assim então, que eles criaram o MySQL, que é vagamente baseado em outro sistema de gerenciamento de banco de dados chamado de mSQL.

O MySQL é rápido, flexível e confiável. É utilizado em muitos lugares por todo o mundo.

Obs.: A propósito, à parte "AB" do nome da companhia é o acrônimo para a palavra sueca "aktiebolag", ou "sociedade anônima". Ela é traduzida para "MySQL, Inc." De fato, MySQL Inc. e MySQL GmbH são exemplos de subsidiárias da MySQL AB. Elas estão localizadas nos EUA e Alemanha, respectivamente.

O que é um banco de dados relacional?

Um banco no mundo de cimento e tijolo é o lugar onde guardamos dinheiro. Um banco de dados também guarda, só que neste caso são dados.

Chamamos de dados tudo que possamos inserir no computador, números, letras, caracteres, imagens e etc.

Um banco de dados relacional é uma composição de tabelas e colunas que se relacionam entre si. Esses relacionamentos são baseados em um valor-chave que é contido em cada tabela, em uma coluna.

Instalando o banco de dados

O MySQL tem diferentes formas de instalação quando se trata de sistemas operacionais. No caso do Windows, você pode baixar a última distribuição através do site:

<http://www.mysql.com/downloads>.

Instalando no Windows

Procure pelo formato executável. O arquivo vem compactado no formato **.zip**.

Descompacte e instale. A instalação, como não poderia deixar de ser, é feita por um assistente. Siga os passos até a finalização.

Caso sua máquina tenha o sistema operacional Windows pertencente a família NT(NT, 2000 ou XP), o MySQL é instalado como serviço. Então basta iniciar ou parar o serviço, encontrado no **Painel de Controle>Ferramentas Administrativas>Serviços**.

Você também pode utilizar o comando pelo prompt, desde que você saiba o nome do serviço do seu MySQL:

Para iniciar o serviço:

```
net start mysql
```

Para parar o serviço:

```
net stop mysql
```

Instalando o MySQL no Linux

O MySQL Server pode ser instalado no Linux de várias formas. A forma recomendada é a que está em formato RPM.

Você deve baixar dois arquivos para instalar o MySQL na sua máquina. Esses arquivos são:

MySQL-server-[versão].i386.rpm – para instalar o servidor mysqld no Linux

MySQL-client-[versão].i386.rpm – para instalar o cliente mysql para executar os comandos no Linux.

A instalação poderá ser feita através do comando rpm, no Shell do seu Linux. Um exemplo seria:

```
Shell> rpm -ivh MySQL-server-5.0.1.i386.rpm MySQL-client-5.0.1.i386.rpm
```

A versão RPM já vem com pré-configurações e assim que ocorrer a instalação, para iniciar ou parar o servidor, a seguinte sintaxe poderá ser feita:

```
Shell>/etc/init.d/./mysql start – para iniciar o servidor MySQL
```

```
Shell>/etc/init.d/./mysql stop – para parar o servidor MySQL
```

Acessando o banco de dados MySQL

No Windows

Se você estiver usando o sistema operacional Windows e utilizou a instalação padrão do programa, abra o prompt de comando e digite a seqüência:

```
cd\mysql\bin
```

Lembrando que você deve estar no drive em que o MySQL está instalado. Por padrão você o instala no drive C.

Digitando o comando a seguir você entra no MySQL.

```
mysql -u root -p
```

Tecele ENTER e receberá o pedido de senha:

password

Digite a senha que você configurou na instalação e tecele ENTER novamente.

Nota: Versões mais modernas do MySQL para o sistema operacional Windows não necessitam de tantos passos para iniciar, bastando ir até o atalho encontrado no menu Iniciar do sistema e no atalho do MySQL iniciar o prompt de comando encontrado neste local.

No Linux

Se você utilizou a instalação binária, em rpm (recomendado), basta abrir o terminal e digitar a seqüência:

```
shell>mysql -u root -p
```

Se já estiver logado como **root**, no seu sistema operacional, não há necessidade de colocar o **-u root -p** depois do comando **mysql**. Somente **-p** caso haja uma senha.

Os comandos CREATE e DROP

Quando pensar nos comandos **CREATE** e **DROP**, você deve imaginar equipamentos de terraplanagem, caminhões basculantes e guindastes, porque são ferramentas que você utiliza para criar o seu banco de dados. Esses comandos, embora raramente utilizados, são os mais importantes.

O comando CREATE

Há muitas maneiras diferentes de criar banco de dados no MySQL. Ao criar um banco de dados, você normalmente terá o layout inteiro pronto. Normalmente adicionaria as tabelas imediatamente depois de criar o banco de dado, mas, teremos uma etapa por vez.

A primeira etapa para criar um banco de dados no MySQL é inserir o comando **CREATE DATABASE nome_banco_de_dados** da SQL (Structured Query Language) no monitor MySQL, onde **nome_banco_de_dados** é o nome do banco de dados que você está criando.

No prompt de comando, no monitor do MySQL, insira o seguinte comando:

```
mysql> CREATE DATABASE relatorios;
```

Note que não foi utilizado acentuação e em casos de palavras compostas não insira espaços, se for o caso insira sublinhado “_”.

O comando USE

Depois de confirmado a criação do banco de dados, você deverá utilizar o comando **USE** para utilizar o banco de dados **relatorios**.

USE relatorios;

Um ponto importante é que o MySQL não torna ativo o banco de dados que você criou, isso deve ser implícito.

O comando DROP

O comando **DROP** é semelhante ao comando **CREATE**. Enquanto o último cria um banco de dados, o primeiro exclui. O comando **DROP** do **SQL** é imperdoável. Não há caixas de confirmação para ver se você tem certeza. Este comando exclui o banco de dados e tudo o que estiver nele.

É só ir até o prompt de comando e no monitor do MySQL e digitar:

```
mysql> DROP DATABASE relatorios;
```

Isso excluirá o banco de dados veículos e tudo o que estiver nele.

Criando tabelas

Criar tabela no MySQL é uma tarefa relativamente fácil. Para se criar uma tabela basta usar a seqüência:

```
shell>mysql -u root -p  
shell>DIGITE AQUI SUA SENHA
```

Após estar no monitor do MySQL digite a seguinte seqüência:

```
mysql> CREATE DATABASE relatorios;
```

```
mysql> USE relatorios;
```

```
mysql> CREATE TABLE categorias (  
-> CategoriaID int(11) NOT NULL auto_increment,  
-> categoria varchar(40) NOT NULL,  
-> descricao longtext,  
-> PRIMARY KEY (CategoriaID),  
-> KEY categoria (categoria)  
-> ) ENGINE=InnoDB
```

O comando SHOW

Assim que criada sua primeira tabela. Para ver o resultado basta digitar a seqüência:

SHOW TABLES FROM relatorios;

Para ver as colunas que existem na sua tabela digite:

SHOW COLUMNS FROM categorias;

Ou **DESCRIBE**.

O comando DESCRIBE

Se preferir, o comando **DESCRIBE** faz a mesma coisa que **SHOW**, mostrando as colunas existentes em sua tabela.

DESCRIBE categorias;

Ou simplesmente:

DESC categorias;

Alterando tabelas existentes

Agora que você criou a sua tabela o que aconteceria se você precisasse alterar algo que fez?

Confira os seguintes exemplos para alterar o nome da tabela, tipo de dados e o nome da coluna:

Alterando o nome da coluna

Usando a cláusula **CHANGE** você pode alterar o nome da coluna da sua tabela:

ALTER TABLE produtos CHANGE prod_nome produto_nome VARCHAR(150);

Alterando o tipo de coluna

O tipo pode ser alterado usando a cláusula **MODIFY**:

ALTER TABLE categorias

MODIFY CategoriaID VARCHAR(200) NOT NULL;

Renomeando uma tabela

Renomear o nome de uma tabela, também se usa **ALTER TABLE**, mas com a cláusula **RENAME**:

ALTER TABLE produtos RENAME tb_categorias;

Excluindo / adicionando colunas e tabelas

Como você pode ver quando uma coluna é criada ou uma tabela estas não são escritas na pedra e podem ser alteradas facilmente. Isso também implica em adicionar colunas em uma tabela existente ou excluí-la.

Eliminando tabelas e colunas

O comando **DROP** também é utilizado para eliminar as colunas de uma tabela.

Para excluir uma tabela existente execute a seguinte seqüência:

DROP TABLE categorias;

Para excluir somente uma coluna execute a seguinte seqüência:

ALTER TABLE categorias DROP descricao ;

Isso excluirá a coluna e todas as informações que você armazenou.

Adicionando colunas

O comando **ADD** é o responsável pela inserção de uma nova coluna.

ALTER TABLE categorias ADD descricao TEXT;

Adicionando colunas após uma outra determinada

O comando **AFTER** adiciona a nova coluna na tabela após o nome mencionado.

ALTER TABLE categorias ADD descricao TEXT AFTER categoria;

Utilizando índices

Um índice é um arquivo estruturado que facilita o acesso a dados.

Isso significa que um índice na coluna correta aumentará a velocidade de uma consulta consideravelmente. Um índice trabalha da mesma forma que pastas com separador alfabético em um gabinete de arquivo ele permite pular para a parte do alfabeto que você está procurando.

Decidindo quais colunas incluir no índice

Você deve colocar um índice na(s) coluna(s) que utilizará com mais freqüência como filtro em suas consultas.

Os índices também funcionam melhor em colunas que contêm dados únicos. Essa é uma das razões pela as quais chaves são normalmente suas melhores escolhas para índices. Essa também pode ser uma das razões que as pessoas confundem chaves e índices. Uma chave ajuda a definir a estrutura de um banco de dados, ao passo que índice apenas aprimora o desempenho.

Um índice pode ser composto de uma ou mais colunas. Você também pode ter mais de um índice em uma tabela.

Criando um índice

Por padrão, o MySQL cria um índice se você declara uma coluna como uma chave primária. Não há necessidade de criar um índice nessa coluna; caso contrário você teria dois índices em uma mesma coluna.

A sintaxe para criar um índice em uma coluna:

```
ALTER TABLE categorias ADD INDEX idx_cat(categoria);
```

Excluindo índices

Excluir um índice é tão simples quanto criar. A sintaxe é a mesma que excluir uma coluna ou uma tabela:

```
DROP INDEX nomedoindice ON nomedatabela;
```

Ou...

```
ALTER TABLE nomedatabela DROP INDEX nomedoindice;
```

Para alterar uma tabela eliminando uma chave primária, utilize a seguinte sintaxe:

```
ALTER TABLE nomedatabela DROP PRIMARY KEY;
```

Nota: Se você estiver usando uma coluna com **AUTO_INCREMENT**, você não excluirá a chave primária enquanto não retirar esse modificador.

Tipos de tabelas

O MySQL possui uma característica um pouco diferente dos outros sistemas gerenciadores de banco de dados, uma vez que no MySQL é possível escolher o tipo da tabela no momento da criação. O formato de armazenamento dos dados, bem como alguns recursos do banco de dados são dependentes do tipo de tabela escolhido.

A definição do tipo de tabela pode ser feita na criação da tabela, como você pode ver a seguir:

```
CREATE TABLE teste (  
  id INT NOT NULL,  
  nome VARCHAR(30) NOT NULL,  
  PRIMARY KEY (id)  
) ENGINE=MyISAM;
```

No comando criado, o tipo da tabela, indicado em `TYPE=MyISAM`, significa que você está criando uma tabela com o tipo MyISAM, que é o padrão das tabelas, caso não seja informado o `TYPE`. A partir da versão 4.0.18 você pode utilizar `ENGINE` como sinônimo de `TYPE`.

Os tipos mais comuns de tabelas criadas são o MyISAM (padrão) e o InnoDB (suporta transações):

O tipo MyISAM

Este é o tipo de tabela padrão do MySQL. Caso não seja informado o tipo de tabela, o MySQL criará a tabela do tipo MyISAM. O tipo de tabela padrão pode ser alterado incluindo-se no arquivo de configuração, chamado de `my.cnf` (no Linux) ou `my.ini` (no Windows), a opção a seguir:

```
default-storage-engine=INNODB
```

As tabelas MyISAM são armazenadas em 3 arquivos, com o mesmo nome da tabela, mas com extensões diferentes:

- **.FRM** que armazena a definição da tabela.
- **.MYD** que contém os dados.
- **.MYI** contendo os índices.

Estas tabelas são de grande desempenho para leitura, uma vez que os seus índices são armazenados em árvores binárias balanceadas, o que provê um ganho para o acesso às informações. O MyISAM não trabalha com transações (commit ou rollback) e também não possui integridade referencial, isto é, ao incluir uma chave estrangeira com alguns constraints, esta servirá apenas como documentação, mas as restrições não serão respeitadas pelo banco.

O tipo InnoDB

O tipo **InnoDB** é do tipo de tabela transacional, desenvolvido pela **InnoDBBase Oy**. A partir da versão 4.0 do MySQL ele passa a ser parte integrante das distribuições do MySQL. O InnoDB apresenta, além da capacidade transacional, outros recursos que são realmente úteis na utilização de tabelas:

- Integridade referencial, com implementação dos constraints **SET NULL**, **SET DEFAULT**, **RESTRICT** e **CASCADE**;
- Ferramenta de backup on-line (ferramenta comercial, não GPL);
- Lock de registro, como Oracle, DB2, etc;
- Níveis de isolamento;
- Armazenamentos de dados em tablespace.

Por se tratar de um tipo de tabela com recursos mais avançados, requer mais espaço em memória e disco, além de se apresentar, em determinadas situações, um pouco mais lento que tabelas do tipo MyISAM. Apesar disto, o tipo InnoDB tem se mostrado extremamente rápido se comparado com outros SGBDs transacionais.

Alterando o tipo de uma tabela

Com o comando **ALTER TABLE** não é possível alterar o tipo da tabela, por isso, você pode alterar da seguinte maneira:

ALTER TABLE departamentos ENGINE=INNODB;

Tipo de dados

Como a maioria dos sistemas de gerenciamento de banco de dados relacional (Relational Database Management Systems – RDBMS), o MySQL tem tipos de dados específicos de coluna.

O MySQL tem vários tipos de dados que suportam funções diferentes. Um tipo de dados é a definição das informações que uma coluna armazenará. Pode haver muitos tipos de dados em uma tabela, mas cada coluna armazenará seu próprio tipo de informações específicas.

Há quatro tipos de grupos de formatos de dados. O primeiro é o numérico. O segundo tipo é o formato de caractere ou string. Esse formato consiste em letras e números ou qualquer coisa que você coloque entre aspas. O terceiro grupo é formado por datas e horas. O tipo final é uma forma de miscelânea. Ele consiste em tudo que não se encaixa em qualquer uma das outras categorias.

Tipos numéricos

Os tipos numéricos destinam-se somente a números. Os diferentes tipos de números ocupam uma quantidade diferente de espaço na memória.

Um bom exemplo é você tentando comprar um chocolate em uma loja e ao passar no caixa a você descobre que deve pagar pela caixa inteira. Você diz que não precisa de tudo, mas é atacado e só é vendido de caixa. Se você vai utilizar 3 números, por que ocupar um espaço na memória como se estivesse utilizando 100?

Lembre-se: você só deve pagar pelo que vai usar.

Tabela 1 - Armazenamento numérico

Nome do tipo	Espaço na memória
<i>TINYINT</i>	1 byte
<i>SMALLINT</i>	2 bytes
<i>MEDIUMINT</i>	3 bytes
<i>INT</i>	4 bytes
<i>BIGINT</i>	8 bytes
<i>FLOAT(Inteiro,Decimal)</i>	4 bytes
<i>DOUBLE(Inteiro,Decimal)</i>	8 bytes
<i>DECIMAL(Inteiro,Decimal)</i>	O valor de bytes Inteiro + 2

Se a coluna é numérica e declarada **UNSIGNED**, o intervalo dobra para o tipo dado. Por exemplo, se você declara que uma coluna que é **UNSIGNED TINYINT**, o intervalo dessa coluna é de **0** a **255**. Declarando dessa forma você faz com que essa coluna tenha somente valores positivos.

Tabela 2 -Tipos numéricos

Nome do tipo	Intervalo de valor	Sem sinal
<i>TINYINT</i>	-128 a 127	0 – 255
<i>SMALLINT</i>	-32768 a 32767	0 – 65535
<i>MEDIUMINT</i>	-8388608 a 8388607	0 - 16777215
<i>INT</i>	-2147483648 a 2147483647	0 - 4294967295
<i>BIGINT</i>	-9223372036854775808 a 9223372036854775807	0 - 18446744073709550615
<i>FLOAT(Inteiro,Decimal)</i>	Varia dependendo dos valores	
<i>DOUBLE(Inteiro,Decimal)</i>	Varia dependendo dos valores	
<i>DECIMAL(Inteiro,Decimal)</i>	Varia dependendo dos valores	

FLOATs, **DOUBLEs** e **DECIMALs** são tipos numéricos que podem armazenar frações. Os outros não.

Utilize **DECIMAL** para números realmente grandes. **DECIMALs** são armazenados de maneira diferente e não têm limites.

Modificadores AUTO_INCREMENT, UNSIGNED e ZEROFILL

Esses modificadores só podem ser utilizados com tipos de dados numéricos. Eles utilizam operações que somente podem ser feitas com números.

AUTO_INCREMENT

O modificador de coluna **AUTO_INCREMENT** automaticamente aumenta o valor de uma coluna adicionando 1 ao valor máximo atual. Ele fornece um contador que é ótimo para criar valores únicos.

Você também pode incluir um número. Se quiser que uma coluna **AUTO_INCREMENT** inicie com **9.000**, por exemplo, é só declarar explicitamente um ponto inicial utilizando a seguinte sintaxe:

```
mysql> CREATE TABLE teste (
-> id INT NOT NULL PRIMARY KEY AUTO_INCREMENT
-> ) AUTO_INCREMENT=9000;
```

UNSIGNED

UNSIGNED depois de um tipo inteiro significa que ele só pode ter um zero ou valor positivo.

ZEROFILL

O modificador de coluna **ZEROFILL** é utilizado para exibir zeros à esquerda de um número com base na largura de exibição.

Como todos os tipos de dados numéricos têm uma largura de exibição opcional, se você declara um **INT(8) ZEROFILL** e o valor armazenado é **23**, ele será exibido como **00000023**.

Para isso utilize a seguinte sintaxe:

CREATE TABLE teste (id INT(4) ZEROFILL);

Tipos de caractere ou de dados de string

O outro grupo importante de tipo de dados são os tipos de strings ou de caractere.

Uma string é um conjunto de caracteres. Um tipo de string pode armazenar dados como São Paulo ou Avenida São João, n.º 255. Qualquer valor pode ser armazenado em um tipo de dados de string.

Tabela 3 - Tipos string

Nome de tipo	Tamanho máximo	Espaço de armazenamento
<i>CHAR(X)</i>	255 bytes	X bytes
<i>VARCHAR(X)</i>	255 bytes	X + 1 byte
<i>TINYTEXT</i>	255 bytes	X + 1 byte
<i>TINYBLOB</i>	255 bytes	X + 2 bytes
<i>TEXT</i>	65.535 bytes	X + 2 bytes
<i>BLOB</i>	65.535 bytes	X + 2 bytes
<i>MEDIUMTEXT</i>	1,6 MB	X + 3 bytes
<i>MEDIUMBLOB</i>	1,6 MB	X + 3 bytes
<i>LONGTEXT</i>	4,2 GB	X + 4 bytes
<i>LOBLOB</i>	4,2 GB	X + 4 bytes

CHAR e VARCHAR

Fora todos esses tipos, os tipos **VARCHAR** e **CHAR** são os mais utilizados. A diferença entre eles é que o **VARCHAR** tem um comprimento variável e o **CHAR** não. Os tipos **CHAR** são utilizados para comprimentos fixos. Você utilizará esse tipo quando os valores não variam muito. Se você declara um **CHAR(10)**, todos os valores armazenados nessa coluna terão **10 bytes** de comprimento, mesmo se ele tiver **3 bytes** de comprimento. O MySQL preenche esse valor para ajustar o tamanho que foi declarado. O tipo **VARCHAR** faz o contrário. Se você declara um **VARCHAR(10)** e armazena um valor que tem somente **3** caracteres de comprimento, a quantidade total de espaço de armazenamento é de **4 bytes** (o comprimento mais um).

A vantagem de utilizar os tipos **CHAR** é que as tabelas que contêm esses valores fixos são processadas mais rapidamente que aquelas que são compostas pelo tipo **VARCHAR**. A desvantagem de utilizar o tipo **CHAR** é o espaço desperdiçado.

De um modo geral não se pode utilizar os dois na mesma tabela, pois quando feito o MySQL converte automaticamente uma coluna com o tipo **CHAR** em **VARCHAR**.

A única exceção é quando você declara uma coluna como **VARCHAR(3)**, o MySQL converte automaticamente em **CHAR(3)**. Isso acontece porque valores de 4 caracteres ou menores são muito pequenos para o tipo **VARCHAR**.

TEXT e BLOB

TEXT e **BLOB** (*Binary Large Object*) são tipos variáveis de comprimento que podem armazenar grandes quantidades de dados. Você utilizará esses tipos quando quiser armazenar *imagens*, *sons* ou *grandes* quantidades de *textos*, como páginas da *Web* ou *documentos*.

Um bom exemplo é se você estiver querendo armazenar valores de uma *<TEXTAREA>* de uma sessão de comentários em uma página da *Web*, o tipo **TEXT** seria uma boa escolha.

Tipos variados

Há basicamente três tipos variados; os tipos *ENUM*, *SET* e *DATE/TIME*.

Tipo ENUM

O tipo **ENUM** é uma lista ENUMerada. Significa que essa coluna pode armazenar apenas um dos valores que estão declarados na lista dada.

Você pode ter até 65.535 *itens* em sua lista enumerada. É uma boa escolha para caixas de combinação.

Tipo SET

O tipo **SET** é muito parecido com o tipo **ENUM**. O tipo **SET**, como o tipo **ENUM**, armazena uma lista de valores. A diferença é que no tipo **SET**, você pode escolher mais de uma opção para armazenar. Um tipo **SET** pode conter até 64 *itens*. O tipo **SET** é uma boa escolha para opções em uma página da *Web* em que o usuário pode escolher mais de um valor.

Tipos de data e hora (DATE/TIME)

O MySQL suporta vários tipos de data e hora. Esses são mostrados na tabela a seguir:

Tabela 4 - Tabela de data e hora

Tipo	Intervalo	Descrição
<i>DATE</i>	1000-01-01 a 9999-12-31	Datas. Será exibida como YYYY-MM-DD
<i>TIME</i>	-838:59:59 a 838:59:59	Hora. Será exibida como HH:MM:SS
<i>DATETIME</i>	1000-01-01 00:00:00 a 9999-12-31 23:59:59	Data e Hora. Será exibida como YYYY-MM-DD HH:MM:SS
<i>TIMESTAMP[(F)]</i>	1970-01-01 00:00:00	Um registro de data/hora, útil para relatório de transação. O formato de exibição depende do formato de F.
<i>YEAR[(2 4)]</i>	70-69 (1970-2069) 1901-2155	Um ano. Você pode especificar 2 ou 4 formatos de dígitos. Cada um desses tem um intervalo diferente, como mostrado.

Modificadores adicionais de coluna

O MySQL tem várias palavras-chave que modificam a maneira como uma coluna funciona.

Como vimos acima, temos **AUTO_INCREMENT** e **ZEROFILL** e como eles afetam a coluna em que são utilizados. Alguns modificadores se aplicam apenas em colunas de um certo tipo de dado.

Tabela 5 - Tabela de Modificadores

Nome de modificador	Tipos aplicáveis
<i>AUTO_INCREMENT</i>	Todos os tipos INT
<i>BINARY</i>	CHAR, VARCHAR
<i>DEFAULT</i>	Todos, exceto BLOB, TEXT
<i>NOT NULL</i>	Todos os tipos
<i>NULL</i>	Todos os tipos
<i>PRIMARY KEY</i>	Todos os tipos
<i>UNIQUE</i>	Todos os tipos
<i>UNSIGNED</i>	Tipos numéricos
<i>ZEROFILL</i>	Tipos numéricos

O modificador **BINARY** faz com que os valores armazenados sejam tratados como strings binárias, fazendo-os distinguir letras maiúsculas e minúsculas. Ao classificar ou comparar essas strings, a distinção entre maiúsculas e minúsculas será considerada.

Por padrão os tipos **CHAR** e **VARCHAR** não são armazenados como binários.

O modificador **DEFAULT** permite especificar o valor de uma coluna se não existir um valor.

Os modificadores **NULL** e **NOT NULL** especifica se na coluna deve haver um valor ou não.

Por exemplo; se você especificar a coluna como **NOT NULL** você é forçado a colocar um valor, pois esse campo é requerido.

PRIMARY KEY é um índice que não deve conter valores nulos (NULL). Cada tabela deve conter uma chave primária, isso facilita uma consulta de dados. Abordarei essa questão mais adiante.

O modificador **UNIQUE** impõe a regra que todos os dados dentro da coluna declarada devem ser únicos, mas aceita valores nulos.

Sintaxe básica da SQL

A primeira coisa que devemos fazer quando criamos um banco de dados e depois uma tabela e utilizá-la inserindo dados.

Comando INSERT

O comando **INSERT INTO** adiciona dados em uma tabela. A sua sintaxe é:

```
mysql> INSERT INTO departamentos (dept_nome)
-> VALUES ('Livros');
```

O nome da tabela em que você irá inserir deverá ser declarada logo no início INSIRA DENTRO nomedatabela (colunas) VALORES ('valores inseridos dentro de cada coluna');

É importante salientar que strings ficam entre aspas ou apóstrofos e valores numéricos (declarados como tipo de dados numéricos) não precisam de “aspas” ou ‘apóstrofos’.

Para inserir mais de um valor separe-os por vírgula:

```
mysql> INSERT INTO nomedatabela(colunas) VALUES ('valores inseridos 1'),
->('valores inseridos 2'),
->('e assim por diante');
```

Comando SELECT

A instrução SELECT é provavelmente a mais utilizada de todas as instruções de SQL. A instrução SELECT somente retornará os dados que são armazenados no banco de dados dentro de uma tabela. O MySQL realiza essa instrução mais rápido que qualquer outro banco de dados do mercado.

A sintaxe é:

```
SELECT nomedacoluna FROM nomedatabela WHERE condições;
```

No caso do nosso banco de dados relatorios:

```
mysql>SELECT * FROM categorias; # o asterisco indica todas as colunas
```

A cláusula WHERE

Com a cláusula WHERE você filtra informações de acordo com a condição passada:

```
mysql>SELECT * FROM categorias WHERE id=3;
```

Nesse caso foi colocada uma condição que dentre todos os registros só deverá aparecer os dados ONDE a coluna **id** for igual à **3**.

A cláusula **WHERE** especifica o critério utilizado para selecionar linhas particulares. O único sinal igual é utilizado para testar igualdade – observe que isso é diferente do Java e é fácil se confundir.

Além da igualdade, o MySQL suporta um conjunto completo de operadores e expressões regulares. Na tabela a seguir estão listadas as mais utilizadas por você:

Tabela 6 - Tabela de Operadores no MySQL

Operador	Nome	Exemplos	Descrição
=	igual à	autor_id = 1	Testa se os dois valores são iguais
>	maior que	Quantidade > 50	Testa se um valor é maior que o outro
<	menor que	Quantidade < 50	Testa se um valor é menor que o outro
>=	maior ou igual a	Quantidade >= 50	Testa se um valor é maior ou igual ao outro
<=	menor ou igual a	Quantidade <= 50	Testa se um valor é menor ou igual ao outro
!= ou <>	diferente de	Quantidade !=0	Testa se um valor é diferente do outro
IN		cidade in ('São Paulo', 'Minas Gerais')	Testa se o valor está em um conjunto particular
NOT IN		cidade not in ('São Paulo', 'Minas Gerais')	Testa se o valor não está em um conjunto particular
IS NOT	Endereço não é nulo		
IS NULL	Endereço é nulo	promocao is null	Testa se o campo não contém um valor
BETWEEN	Quantidade entre um valor e outro	valor BETWEEN 200 AND 350	Testa se o campo tem valores entre um e outro

Funções que trabalham com a instrução SELECT

Vistas ao longo desse livro, sejam aplicadas ou comentadas, você tem a explicação mais detalhada a seguir:

LIMIT

Função que limita resultados exibidos na tela.

```
SELECT * FROM tabela LIMIT 2;
```

Limita a visualização de 2 linhas de dados.

```
SELECT * FROM tabela LIMIT 2,5;
```

Limita a visualização da linha 2 a linha 5 de dados. Esta instrução foi utilizada para paginação de dados.

COUNT()

Conta a quantidade de linhas encontradas, de acordo com a coluna passada entre os parênteses.

Para contar uma determinada quantidade de dados em uma coluna:

```
SELECT COUNT(coluna) FROM tabela;
```

Isso não reflete a quantidade total existente na tabela, pois um valor NULL pode existir.

Para contar o total de linhas existentes em uma tabela, use:

```
SELECT COUNT(*) FROM tabela;
```

Conta quantas linhas de dados existem em todas as linhas.

Nota: Em caso de fazer a contagem em campo de valor **NULL** a contagem será diferente da no valor total.

ORDER BY

Ordena os resultados de acordo com a coluna estabelecida (crescente ou decrescente):

```
SELECT * FROM tabela ORDER BY coluna;
```

Ordena de forma crescente pela coluna dada.

ORDER BY ... DESC

```
SELECT * FROM tabela ORDER BY coluna DESC;
```

Coloca os dados selecionados em ordem decrescente pela coluna.

LIKE

Usado para filtrar dados em uma coluna que armazena strings (varchar, text e etc...). Sua sintaxe é como mostrada a seguir:

```
mysql>SELECT * FROM clientes WHERE nome LIKE 'em%';
```

Neste caso pode-se fazer uma busca por apenas a inicial do valor desejado.

O sinal de %(porcentagem) é o caractere curinga que significa qualquer caractere.

```
mysql>SELECT * FROM clientes WHERE nome LIKE '%em%';
```

Colocando a % no início e no fim, com um valor no meio, é possível buscar todos os valores que contenham as letras **em**, seja no começo, meio ou fim. Esta instrução foi feita para os exemplos que utilizavam um sistema de auto-completar (caixa de sugestões).

Comando UPDATE

O comando UPDATE permite editar os valores de dados existentes. A sintaxe para modificar os valores é:

```
UPDATE tabela SET coluna= 'valor' WHERE coluna='valor';
```

Atualiza os dados da coluna determinada em SET na condição passada em WHERE.

Comando DELETE

A instrução DELETE é muito semelhante á instrução SELECT. A única diferença em vez de selecionar registros para visualizar, essa instrução exclui esses registros.

A instrução DELETE tem a seguinte sintaxe:

```
DELETE FROM tabela WHERE coluna='valor';
```

Trabalhando com Junções

As junções são uma parte integrante de um banco de dados relacional. As junções permitem ao usuário de banco de dados tirar proveito dos relacionamentos que foram desenvolvidos na fase do projeto do banco de dados.

Uma *JUNÇÃO* é o termo utilizado para descrever o ato em que uma ou mais tabelas são “unidas” entre si para recuperar dados necessários com base nos relacionamentos que são compartilhados entre elas.

Criando uma junção com INNER JOIN

A seguinte sintaxe cria uma junção:

```
SELECT tabela1.coluna, tabela2.coluna FROM tabela1 INNER JOIN tabela2 on  
tabela1.coluna_de_valor_identico=tabela2.coluna_de_valor_identico;
```

INNER JOIN's são provavelmente as mais comuns de todas as junções.

Uma **INNER JOIN** significa que todos que todos os registros que estão sem correspondência são descartados. Somente as linhas correspondidas serão exibidas no conjunto de resultados. Os dados aparecem na ordem em que você especifica.

Chaves variadas do MySQL

O que é uma chave?

Uma chave em uma tabela em um banco de dados fornece um meio de localizar rapidamente informações específicas. Embora uma chave não precise significar qualquer coisa para o usuário humano do banco de dados, as chaves são uma parte vital da arquitetura de banco de dados e pode influenciar significativamente o desempenho.

Como as chaves funcionam

Uma chave existe como uma tabela extra no banco de dados, embora pertença à sua tabela pai. Ela ocupa espaço físico no disco rígido (ou outras áreas de armazenamento) do banco de dados. Pode ser tão grande quanto a tabela principal e, teoricamente, até maior.

Você define a chave para se relacionar com uma ou várias colunas em uma tabela específica. Como os dados em uma chave são totalmente derivados da tabela, você pode eliminar e recriar uma chave sem qualquer perda de dados.

Benefícios de usar uma chave

A utilização adequada de chaves pode aprimorar significativamente o desempenho do banco de dados. Para utilizar a analogia de um índice de livro, considere o pouco número de páginas que é necessário no índice de um livro para dar visão rápida dos temas importantes. Compare quanto tempo você levaria se estivesse pesquisando pelo volume, página por página.

Suporte de chaves do MySQL

O MySQL suporta os seguintes comandos para criar chaves nas tabelas existentes:

```
ALTER TABLE nome_tabela ADD (KEY | INDEX) nome_do_índice (nome_da_coluna [...]);
```

```
ALTER TABLE nome_tabela ADD UNIQUE nome_do_índice (nome_da_coluna[,...]);
```

```
ALTER TABLE nome_tabela ADD PRIMARY KEY nome_do_índice (nome_da_coluna[,...]);
```

Observe que no MySQL, chave e índice são sinônimos.

Esses são os formatos preferidos para adicionar chaves a tabelas existentes. Para compatibilidade com outras implementações de SQL, o MySQL também suporta os seguintes:

```
CREATE INDEX nome_do_índice ON nome_tabela (nome_da_coluna[,...]);
```

```
CREATE UNIQUE INDEX [nome_do_índice] ON nome_tabela (nome_da_coluna[,...]);
```

```
CREATE PRIMARY KEY ON nome_tabela (nome_da_coluna,...);
```

Você pode definir as chaves quando cria uma tabela:

```
CREATE TABLE nome_da_tabela (nome_da_coluna tipo_de_campo [NULL | NOT NULL], KEY col_index (nome_da_coluna));
```

Chaves primárias (Primary Key)

Uma chave primária é semelhante em princípio a uma chave única, seus dados devem ser únicos, mas a chave primária de uma tabela tem um status mais privilegiado. Apenas uma chave primária pode existir para cada tabela e seus valores de campo nunca podem ser nulos.

Uma chave primária é geralmente utilizada como um link estrutural no banco de dados, definindo o relacionamento entre as tabelas diferentes. Sempre que quiser unir uma tabela a outra, você deve ter a chave primária dessa tabela.

O MySQL não requer que você especifique que a coluna em que estiver a chave primária seja NOT NULL(não nula) *, mas porém se tentar colocar um valor idêntico na coluna chave, esta retornará um erro que não pode haver duplicação.

* Este caso é somente para chaves primárias em tabelas cuja coluna selecionada seja INT ou semelhante. Em casos de ser VARCHAR, CHAR e etc, é exigida a utilização do NOT NULL. Caso isso não ocorra, você terá como resultado um erro. Se desejar que seja nulo o campo, coloque uma chave UNIQUE.

Chaves estrangeiras (Foreign Key)

As chaves estrangeiras não são atualmente suportadas no MySQL pelas tabelas do tipo MyIsam. A mais usada e recomenda para transações são as tabelas do tipo **InnoDB**.

A seguir você tem o comando necessário para criar uma chave estrangeira no seu banco de dados:

```
ALTER TABLE produtos ADD CONSTRAINT fk_categoria  
FOREIGN KEY (CategoriaID) REFERENCES  
categorias (CategoriaID) ON DELETE CASCADE ON UPDATE CASCADE;
```

Excluindo uma chave estrangeira

Para excluir uma chave estrangeira, use o comando:

```
ALTER TABLE produtos DROP FOREIGN KEY fk_categoria;
```

Administrando o MySQL

Um sistema de MySQL pode ter muitos usuários. O usuário root geralmente deve ser utilizado somente para propósitos de administração, por razões de segurança. Para cada usuário que precisar utilizar o sistema, você precisará configurar uma conta e senha.

Não é obrigatório configurar senhas para usuários, mas recomendo que você configure senhas para todos os usuários que forem criados.

Entendendo o sistema de privilégios do MySQL

O MySQL suporta um sofisticado sistema de privilégios. Um privilégio é um direito que um usuário tem para realizar uma ação particular em um objeto particular.

Quando você cria um usuário no MySQL, você concede a ele um conjunto de privilégios para especificar o que ele pode e não pode fazer dentro do sistema.

Configurando usuários

Os comandos GRANT e REVOKE são utilizados para fornecer e retirar direitos dos usuários do MySQL. Ele pode ser concedido nos seguintes níveis:

- Global
- Banco de dados
- Tabela
- Coluna

O comando para criar um usuário com privilégios é como mostrado a seguir:

```
GRANT privilégios [colunas] ON item  
TO nome_do_usuario [IDENTIFIED BY 'senha']  
[WITH GRANT OPTION]
```

As cláusulas entre colchetes são opcionais.

Para conceder privilégios ao um usuário no banco livraria, você deve criar um usuário com os seguintes privilégios:

```
mysql> grant all  
      -> on relatorios.*  
      -> to edson identified by 'integrator';
```

Com isso você concede todos os privilégios de manipulação do banco de dados livraria somente ao usuário **edson**, com a senha **integrator**.

Confirmando o novo usuário

Para confirmar a criação do novo usuário, você deve executar o comando a seguir:

```
flush privileges;
```

Revogando privilégios

Para revogar esse privilégio você deve fazer o seguinte comando:

```
mysql> revoke all  
-> on relatorios.*  
-> from edson;
```

Obtendo informações com SHOW

Se você desejar visualizar todos os privilégios que um usuário tem, execute o seguinte comando:

```
SHOW GRANTS FOR edson;
```

Para visualizar todos os usuários existentes no seu MySQL execute;

```
SHOW GRANTS;
```

Apêndice B

Contexto de desenvolvimento Web em Java

Para desenvolver os exemplos das aplicações Web escritas em Java, você primeiro precisa criar um contexto de desenvolvimento de sua aplicação Web, para que tudo funcione perfeitamente.

Crie um diretório chamado de **Site** no local que desejar. Para que você não tenha dúvidas, se você estiver usando o *Windows* crie, por exemplo, no drive **C**. No *Linux* pode ser criado dentro do diretório **home** do seu usuário. Dentro desse diretório, crie os demais diretórios como apresentados:

Site/

WEB-INF/

classes/

lib/

Note que dentro do diretório **Site** existe o diretório **WEB-INF**. No diretório **WEB-INF**, existem dois diretórios, um chamado de **classes** e outro de **lib**.

No diretório WEB-INF

No diretório **WEB-INF** adicione o arquivo **web.xml**:

web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
  version="2.4">
  <display-name>Texto mostrado no manager</display-name>
  <description>
    Descritor do contexto de desenvolvimento.
  </description>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

O arquivo WEB-INF/**web.xml** é o descritor de contexto de uma aplicação web, segundo a especificação Java Servlet/J2EE. As informações nele contidas são as configurações específicas da aplicação.

Instalando uma aplicação Web em outro local

Talvez você queira criar uma aplicação Web em um caminho diferente do habitual webapps do Tomcat. Nesse caso, você deverá configurar um arquivo de contexto, contendo o nome da aplicação e o caminho onde se encontra:

context.xml

```
<Context path="/Site" docBase="C:\Site"
    debug="0">

    <!-- Link to the user database we will get roles from -->
    <ResourceLink name="users" global="UserDatabase"
        type="org.apache.catalina.UserDatabase"/>

</Context>
```

No atributo **docBase** você coloca o caminho físico da sua aplicação Web, no Windows, como mostrado a seguir ou no Linux, por exemplo: **docBase="/var/www/Site"**.

No URL do navegador, digite como mostrado a seguir, nos exemplos:

No Windows

<http://localhost/manager/deploy?path=/Site&config=file:/C:/Site/context.xml>

No Linux

<http://localhost/manager/deploy?path=/Site&config=file:/home/Site/context.xml>

A mensagem a seguir será mostrada no navegador, o que indicará que a aplicação foi configurada com sucesso:

OK - Deployed application at context path /Site

Apêndice C

Trabalhando com Datas

Um problema comum no desenvolvimento de aplicações Web é a formatação de Datas, que pela complexidade da mesma, contém diversas formas de trabalho.

Para formatar a representação de um objeto Date você pode utilizar a classe **SimpleDateFormat**. Essa classe fornece um conjunto de caracteres padrão para formatação do objeto **Date**. Veja a seguir uma tabela retirada da documentação do Java onde cada campo de uma data tem uma representação String.

Tabela 1

Símbolo	Significado	Apresentação	Exemplo
G	Era	(Text)	AD
y	ano	(Number)	1996
M	Mês no ano	(Text & Number)	July & 07
d	Dia no mês	(Number)	10
h	Hora em am/pm (1~12)	(Number)	12
H	Hora em um dia (0~23)	(Number)	0
m	minute in hour	(Number)	30
s	segundos	(Number)	55
S	milissegundos	(Number)	978
E	Dia em uma semana	(Text)	Tuesday
D	Dia em um ano	(Number)	189
F	Dia da semana no mês	(Number)	2 (2nd Wed in July)
w	Semana no ano	(Number)	27
W	Semana no mês	(Number)	2
a	am/pm	(Text)	PM
k	Hora (1~24)	(Number)	24
K	Hora em am/pm (0~11)	(Number)	0
z	time zone	(Text)	Pacific Standard Time

Para melhor entendimento, você tem a seguir alguns exemplos de formatações de datas:

dd/MM/yy = 10/09/06

dd/MMM/yyyy = 23/MAI/2006

Até 3 dígitos você tem um valor resumido, acima de quatro a representação de strings é sem abreviação.

Neste exemplo abaixo você pode conferir como é simples utilizar o objeto **SimpleDateFormat** para melhorar a saída dos seus programas Java:

```
Date hoje = new Date( );
SimpleDateFormat formato = new SimpleDateFormat("dd/MM/yyyy");
System.out.println("A data formatada é: "+ formato.format(hoje));
formato = new SimpleDateFormat("EEEE, dd de MMMM de yyyy");
System.out.println("Hoje é: "+ formato.format(hoje));
```

A saída da execução deste programa é:

A data de hoje é: **Sun Sep 10 22:56:13 GMT-03:00 2006**

A data formatada é: 10/09/2006

Hoje é: Domingo, 10 de Setembro de 2006

A classe **SimpleDate** pode ser usada para formatar a representação de um objeto Date para a data do seu usuário.

O trecho de código a seguir apresenta a utilização dessa classe:

```
Locale brasil = new Locale ("pt", "BR");
DateFormat df = DateFormat.getDateInstance(DateFormat.LONG, brasil);
System.out.println("Brasil: "+ df.format(hoje));
df = DateFormat.getDateInstance(DateFormat.LONG, Locale.US);
System.out.println("USA: "+ df.format(hoje));
```

Após criar um objeto Locale que representa o Brasil, você pode instanciar um objeto DateFormat e utilizar os métodos para formatação representando a localidade.

Apêndice D

O Tomcat

Tomcat tem suas origens no início da tecnologia servlet. A Sun Microsystems criou o primeiro contêiner servlet, o Java Web Server, para demonstrar a tecnologia, mas não era um servidor robusto, para uso na Web como se necessitava. Ao mesmo tempo, o Apache Software Foundation (ASF) criou JServ, um servlet engine que integrava com o servidor Web Apache.

Em 1999, a Sun Microsystems doou o código do Java Web Server para o ASF, e os dois projetos se fundiram para criar o Tomcat. A versão 3.x foi à primeira da série Tomcat e teve a descendência direta do código original que a Sun Microsystems doou ao ASF.

Em 2001, o ASF liberou a versão 4.0 do Tomcat, que era uma completa arquitetura redesenhada do Tomcat no qual teve um novo código base. A série Tomcat, versão 4.x é um RI (reference implementation) de especificações Servlet 2.3 e JSP 1.2.

A versão Tomcat 6.0.x é a atual e é a RI de especificações Servlet 2.5 e JSP 2.1, no momento em que este livro foi escrito.

Tecnicamente, o Tomcat é um Container Web, mas tem a capacidade de atuar também como servidor Web/HTTP assim como pode funcionar integrado a um servidor web dedicado como o Apache ou o Microsoft IIS. O Tomcat, porém, não implementa até o momento um container EJB.

Servlets e JSP

J2SE (Standard Edition) é planejado para desenvolvimento do lado cliente e pode ser usada para desenvolver as tradicionais aplicações baseadas em cliente/servidor.

As aplicações para Web estão alçadas em Java EE (Enterprise Edition). A implementação de aplicações Web usando o padrão Java EE é bastante complexa.

Servlets e JSP são duas tecnologias desenvolvidas pela Sun Microsystems para desenvolvimento de aplicações Web a partir de componentes Java que executem do lado servidor.

Instalando o Tomcat

Para instalar o Tomcat, inicialmente, você deverá ter também instalado o Java em sua máquina.

Se você apenas for rodar aplicações no Tomcat, basta ter o JRE (máquina virtual do Java) instalado. Mas se você desenvolve, o ambiente JDK deverá estar instalado.

O Tomcat 6 que é focado no J2SE 5.0 ou superior, com uma nova modelagem de vários componentes estruturais visando maior desempenho, estabilidade e facilidade de manutenção. Para quem utiliza o J2SE 1.4, o Tomcat 5.0.x ainda é o mais recomendado.

Baixando e instalando o Apache Tomcat

Instalar o Tomcat é simples como no caso do JDK. Você deverá baixar a versão desejada no site. Entre no link <http://tomcat.apache.org/download-60.cgi> e selecione a distribuição binária desejada.

No Windows

Existe mais de uma forma de instalar o Tomcat na sua máquina, em Windows. Em **Binary Distributions**, na página, você tem o item **Core**, que permite selecionar o arquivo compactado "zipado", "tarball" ou como **Windows Executable**.

Como executável você terá a instalação em forma de assistente. No caso, iremos dar preferência ao arquivo compactado, baixando para o Windows no formato **zip**.

Descompacte-o e coloque-o em um diretório chamado de **Tomcat**, tornando assim fácil o seu acesso ao diretório.

Se você der preferência pela instalação por assistente, você poderá rodá-lo como serviço ou não, dependendo de você.

Adicionando a variável CATALINA_HOME

%CATALINA_HOME% é o diretório onde você escolheu para instalar o seu Tomcat. No Windows, em versão NT (2000, XP e Vista), entre no Painel de controle e abra Sistema:

Iniciar > Configurações > Painel de Controle > Sistema

Crie uma variável chamada de **CATALINA_HOME** e coloque em seu valor o caminho da instalação.

No Windows 9x, você deve entrar no arquivo **autoexec.bat** e adicionar a linha:

```
set CATALINA_HOME=C:\tomcat
```

Lembrando que o caminho apresentado é o local onde eu escolhi para instalação do Tomcat.

No Linux

No Linux você tem a possibilidade de instalação como no Windows, baixando o arquivo com extensão **tar.gz** (tarball) ou **.zip** se assim preferir.

Descompacte o arquivo com o seguinte comando, no diretório onde se encontra o arquivo baixado:

```
shell# tar -xpvzf apache-tomcat-6.0.14.tar.gz /usr/local/tomcat
```

Após a descompactação, no arquivo **/etc/profile** adicione a variável **CATALINA_HOME**:

```
CATALINA_HOME=/usr/local/tomcat
```

```
export CATALINA_HOME
```

Iniciando o Tomcat

Após instalar o Tomcat, você deverá iniciá-lo. Embora existam dois sistemas operacionais sendo abordados, a inicialização no caso será idêntica, desde que não haja a instalação por assistente no Windows.

No Windows

Para iniciar no Windows, vá até o diretório `%CATALINA_HOME%` e no diretório **bin** dê um duplo clique no arquivo **startup.bat**.

Se você quiser rodar pelo prompt de comando, execute a seguinte linha:

```
%CATALINA_HOME%\bin\startup.bat
```

No Linux

Para o Linux, você deve iniciar através do comando:

```
shell # $CATALINA_HOME/bin/startup.sh
```

Parando o Tomcat

Assim como você inicia, você também pode parar o Tomcat.

No Windows

Para parar o Tomcat no Windows use o comando:

```
%CATALINA_HOME%\bin\shutdown.bat
```

Se preferir, você pode ir até onde o Tomcat está instalado e no diretório **bin** dar um duplo clique no arquivo **shutdown.bat**.

No Linux

Pare o Tomcat com o comando a seguir no Linux:

```
$CATALINA_HOME/bin/shutdown.sh
```

O Tomcat em operação

Depois de instalado e iniciado o Tomcat, você pode vê-lo trabalhando abrindo um navegador e digitando em sua barra de endereços:

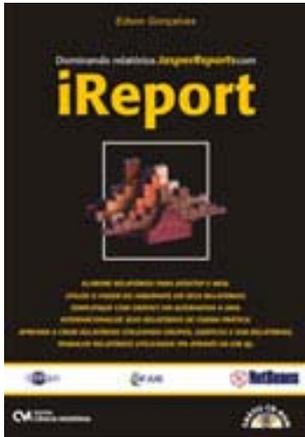
```
http://localhost:8080
```

O que você encontra no livro Dominando Relatórios JasperReports com iReport

Este e-book é um pequeno resumo do livro Dominando Relatórios JasperReports com iReport.

No livro, como chamar seu relatório por sua aplicação Web, seja usando NetBeans IDE ou Eclipse.

Além disso, há também como utilizar o framework JavaServer Faces para chamar o relatório e resolução de problemas.



Livro Dominando Relatórios JasperReports com iReport

Sub-relatórios

Os sub-relatórios são relatórios dentro de outros, também conhecidos como relatórios do tipo “mestre-detalhe”.

A idéia básica para a criação de um relatório mestre-detalhe é ter duas entidades relacionadas, onde a primeira possui um cadastro principal da qual a outra depende e origina-se diversos outros cadastros.

Estes relatórios, embora possam parecer complexos, são simples de desenvolver em JasperReports, principalmente se utilizada a ferramenta iReport.

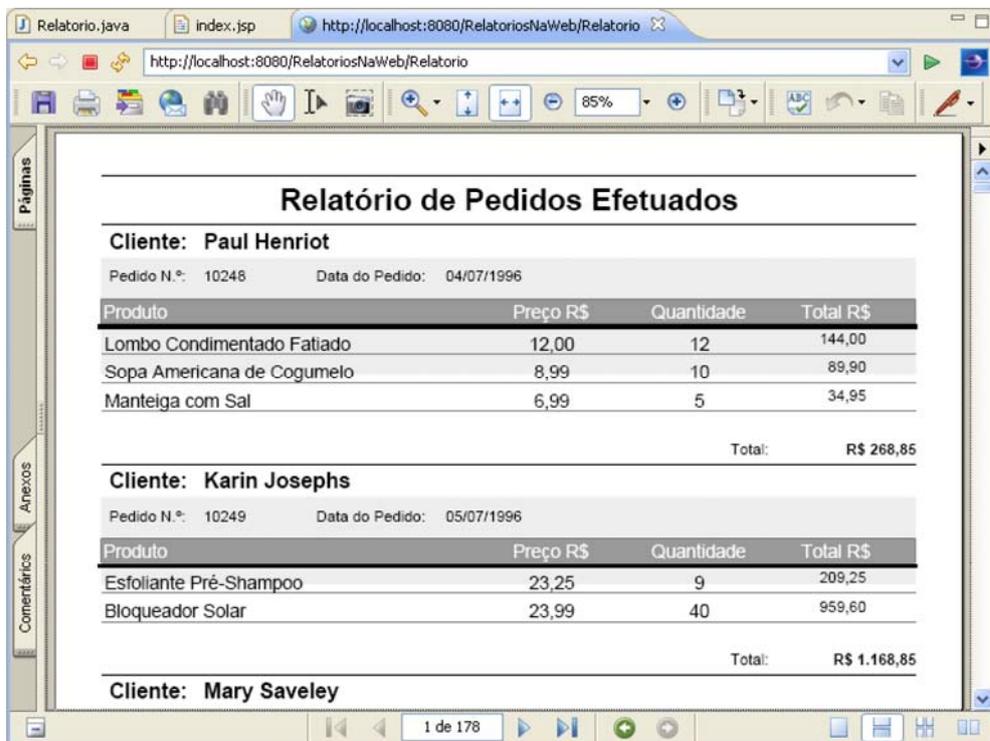
Neste capítulo você aprenderá a criar sub-relatórios em seu relatório usando o iReport e acesso a dados, conhecerá a `band noData` na prática e depois o chamará através de uma aplicação Web com JSP e Servlets.

O relatório mestre-detalhe que será desenvolvido

O relatório final, a ser desenvolvido, possui como semelhança do Capítulo 5 sua estrutura, uma vez que fará uso das mesmas tabelas, mas em uma situação completamente diferente. Antes, você criou o relatório com agrupamento. Desta vez, será criado com sub-relatório. A **Figura 1** ilustra o resultado final do relatório.

Relatório de Pedidos Efetuados			
Cliente: Paul Henriot			
Pedido N.º: 10248		Data do Pedido: 04/07/1996	
Produto	Preço R\$	Quantidade	Total R\$
Lombo Condimentado Fatiado	12,00	12	144,00
Sopa Americana de Cogumelo	8,99	10	89,90
Manteiga com Sal	6,99	5	34,95
			Total: R\$ 268,85
Cliente: Karin Josephs			
Pedido N.º: 10249		Data do Pedido: 05/07/1996	
Produto	Preço R\$	Quantidade	Total R\$
Esfoliante Pré-Shampoo	23,25	9	209,25
Bloqueador Solar	23,99	40	959,60
			Total: R\$ 1.168,85

Figura 1 – Exemplo final do relatório mestre-detilhe



The screenshot shows a web browser window with the URL <http://localhost:8080/RelatoriosNaWeb/Relatorio>. The browser displays a PDF report with the following content:

Relatório de Pedidos Efetuados			
Cliente: Paul Henriot			
Pedido N.º: 10248		Data do Pedido: 04/07/1996	
Produto	Preço R\$	Quantidade	Total R\$
Lombo Condimentado Fatiado	12,00	12	144,00
Sopa Americana de Cogumelo	8,99	10	89,90
Manteiga com Sal	6,99	5	34,95
			Total: R\$ 268,85
Cliente: Karin Josephs			
Pedido N.º: 10249		Data do Pedido: 05/07/1996	
Produto	Preço R\$	Quantidade	Total R\$
Esfoliante Pré-Shampoo	23,25	9	209,25
Bloqueador Solar	23,99	40	959,60
			Total: R\$ 1.168,85
Cliente: Mary Saveley			

Figura 2 – Exibição do relatório gerado em PDF

Relatórios com Hibernate

Criar relatórios com o uso do Hibernate como camada de acesso a banco de dados é um caso comum no desenvolvimento de aplicações Web.

O Hibernate é hoje o mais popular projeto de gerenciamento de dados persistentes em Java e também o que possui uma gama de soluções para problemas comuns.

Neste capítulo você aprenderá a desenvolver seus relatórios utilizando o Hibernate para acessar os dados, incluindo um exemplo com sub-relatório.

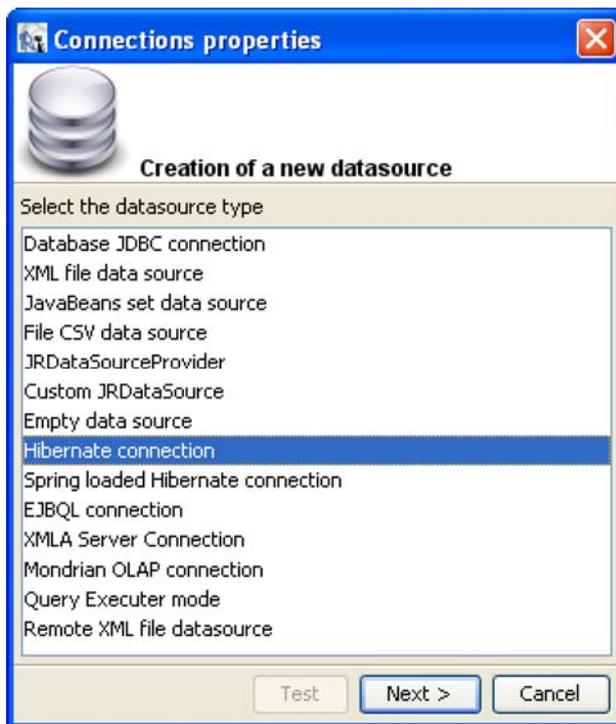


Figura 3 – Seleção do item **Hibernate connection**

Na segunda etapa, digite o nome de sua conexão em **Name** e clique no botão **Test** logo após.

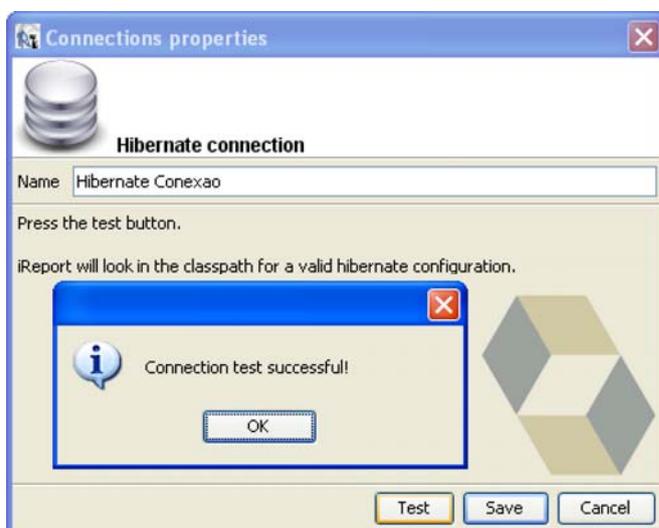


Figura 4 – Criando e testando a conexão com o **Hibernate**

O primeiro relatório utilizando o Hibernate é mostrado na **Figura 5**.

Produtos Cadastrados			
Código	Produto	Preço R\$	Imagem do Produto
1	Chopp e Pêssego DRAFT & FRUIT	1,99	chopp_pessego.jpg
2	Chá com Pêssego	1,70	cha_pessego.jpg
3	Alho Especial Aperitivo	14,99	alho_especial.jpg
4	Molho à Bolonhesa	14,50	molho_bolonhesa.jpg
5	Aceto Italiano Balsâmico	21,35	aceto_italiano.jpg

Figura 5 – Resultado final do primeiro relatório com Hibernate

Sub-relatório com Hibernate

O segundo relatório utilizando o Hibernate é mostrado na **Figura 6**.

Relatório de Produtos Cadastrados			
Categoria: Carne, A vícula e Peixes			
Categoria N°: 6		Descrição: Carnes bovinas, carnes suínas, carnes de aves, peixes	
Produto	Preço	Qtd. Estoque	
Baby Beef de Alcatra Alcatra Bovina	18,65	29	
Coelho Congelado em Cortes	18,47	0	
Lingüiça de Carne Suína	10,49	0	
Bacalhau	41,19	0	
Lombo suíno temperado e congelado	18,89	21	
Assado de tira (costela bovina)	17,12	115	
Categoria: Higiene e Perfumaria			
Categoria N°: 7		Descrição: Produtos de Higiene e Perfumaria	
Produto	Preço	Qtd. Estoque	
Escova Dental COLGATE Micro Soric	19,19	15	

Figura 6 – Relatório mestre-detalle gerado com o uso do Hibernate

Relatórios usando EJB QL

O uso de Java Persistence API se tornou muito intenso no último ano, e como não poderia deixar de ser, o JasperReports possui suporte a esta especificação através de EJB QL. Esta linguagem de queries é muito similar à HQL do Hibernate e também a tradicional instrução SQL.

Com a especificação do EJB3, na introdução da JPA, Java Persistence API, ficou muito popular o uso de JPQL, Java Persistence Query Language, que é uma extensão da EJB QL (JavaBeans Query Language).

Neste capítulo você vai aprender a configurar um arquivo de persistência e criar um relatório com sub-relatório utilizando a EJB QL.

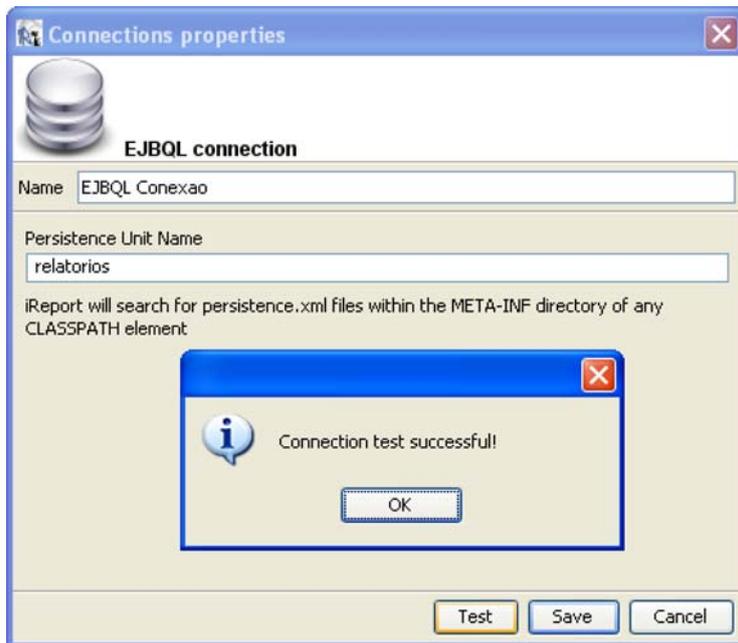


Figura 7 – Criando e testando a conexão EJBQL

Desenvolvendo o relatório

O relatório que será criado é com sub-relatório.

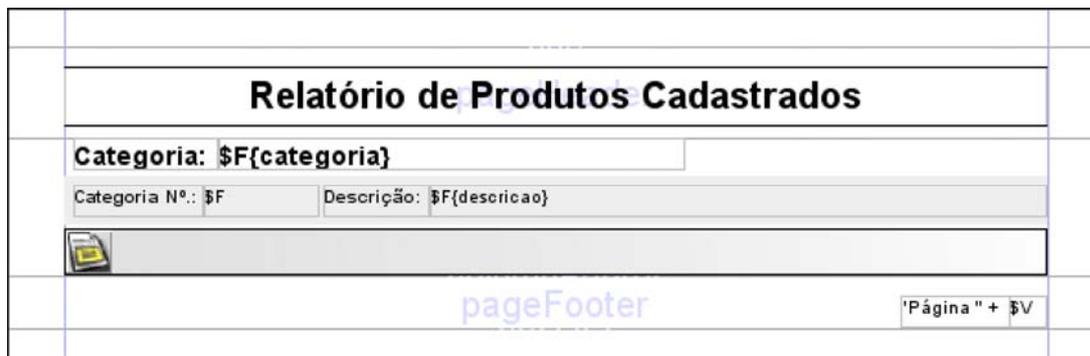


Figura 8 – Design do relatório mestre-detalle feito para utilizar EJBQL

Apenas para que você possa examinar, seu relatório mestre, quando executado, poderá ser semelhante ao mostrado na **Figura 9**, mostrada a seguir.



Figura 9 – Exibição final do relatório mestre

Exemplo da JP-QL utilizada:

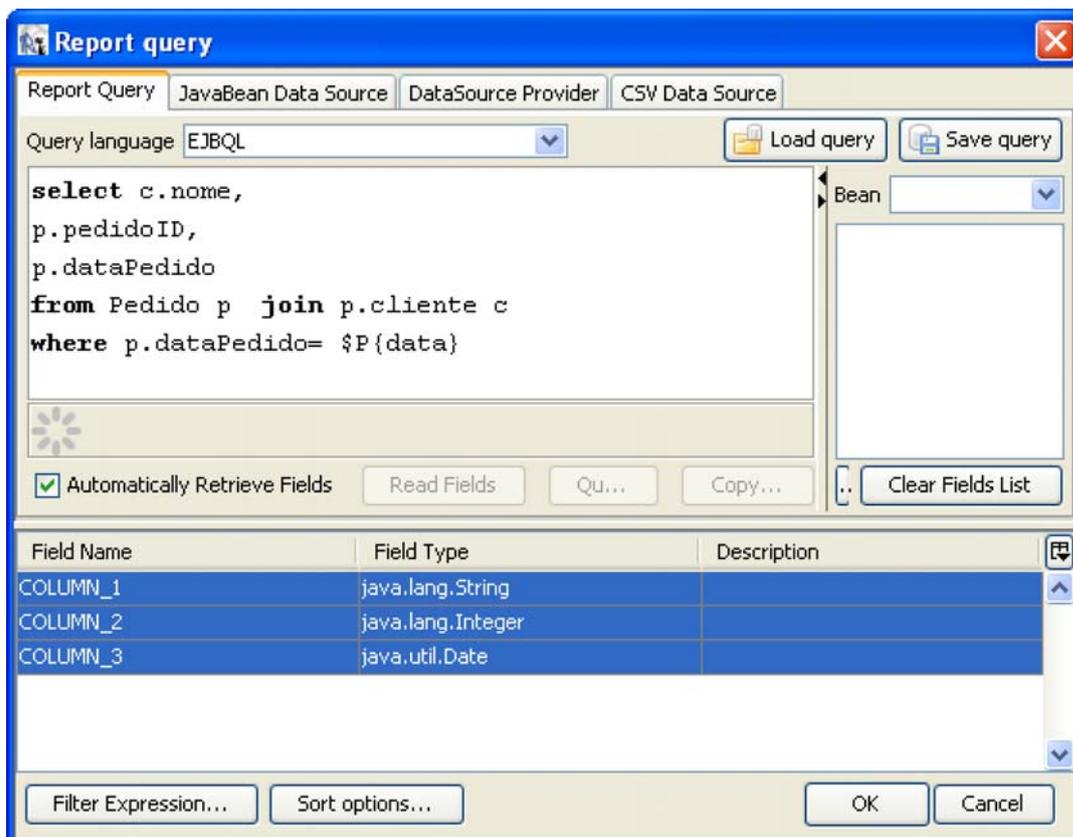


Figura 10 – Uma query com apenas alguns objetos/valores

Gerando relatórios com outras fontes de dados

O JasperReports é rico no desenvolvimento de relatórios e fornece ao desenvolvedor uma gama de outras fontes de dados acessíveis para que o mesmo possa criar relatórios.

Diversos sistemas costumam exportar dados em diversos tipos, sendo os mais comuns o formato CSV e XML.

Este capítulo tratará de outras formas de acesso que não utilizam um banco de dados diretamente.

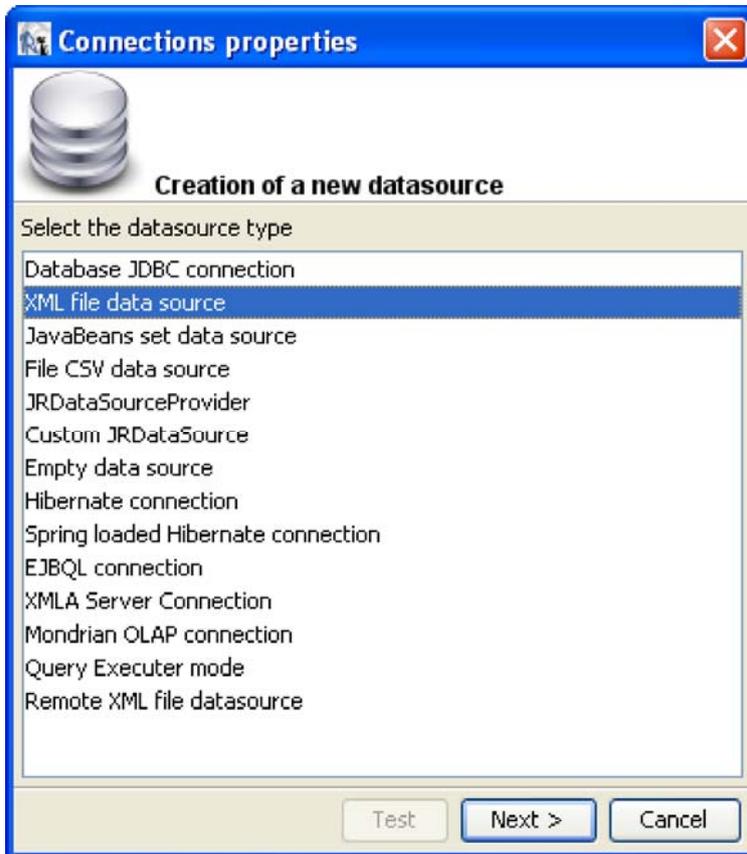


Figura 11 – Selecionando XML file data source

O design do relatório com XML mostrado na **Figura 12**.

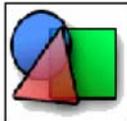
Produtos Cadastrados			
Produto	Quantidade	Preço	Imagem
<code>#{nome}</code>	<code>#{categoria}</code>	<code>#{preco}</code>	
<code>detail</code>			
<code>pageFooter</code>			<code>'Página " + #{V}</code>

Figura 12 – O design do relatório usando XML

O resultado pode ser visto na **Figura 13**.

Produtos Cadastrados			
Produto	Quantidade	Preço	Imagem
Chopp e Pêssego DRAFT & FRUIT	1	199,00	
Chá com Pêssego	1	170,00	
Alho Especial Aperitivo	2	1.459,00	

Figura 13 – Resultado compilado do relatório utilizando XML

Relatório utilizando CVS:

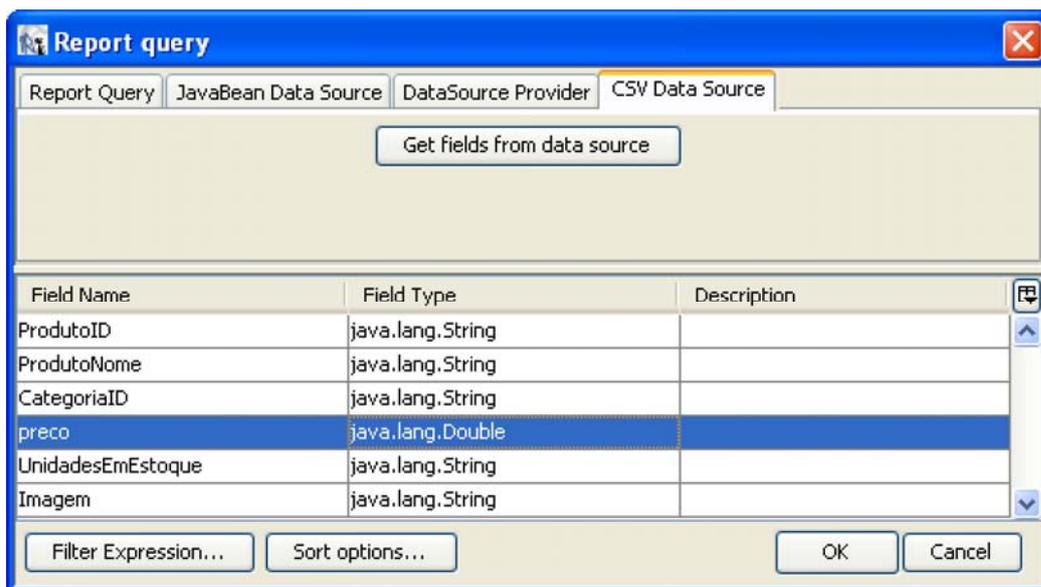


Figura 14 – Definindo os campos de um arquivo CSV

Relatório sem Fonte de dados existente.

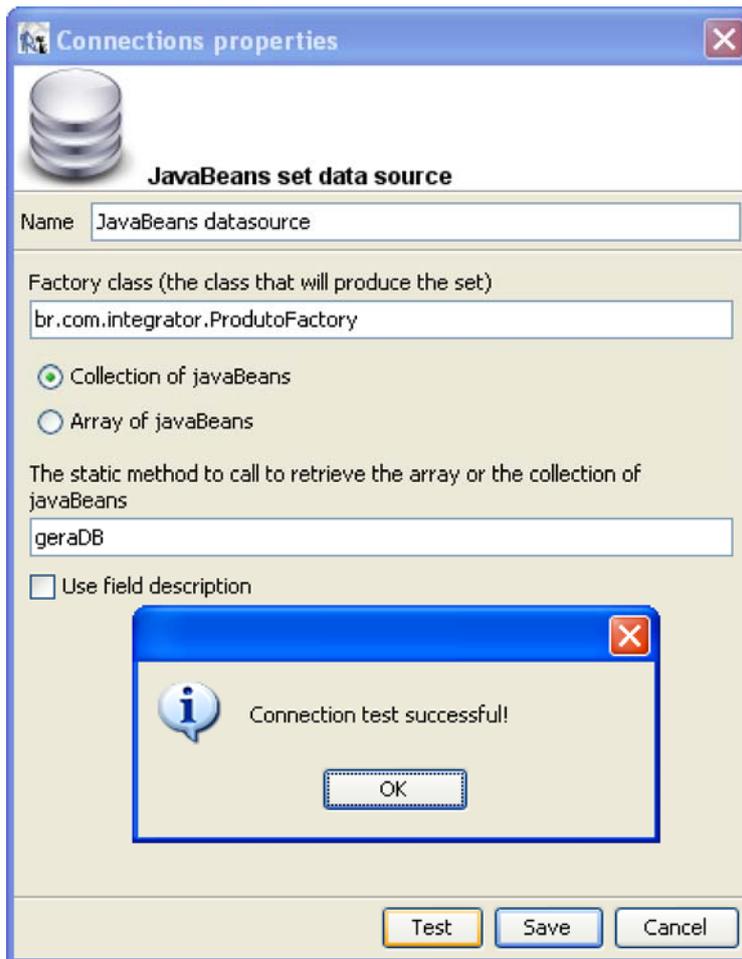


Figura 15 – Configurada e testada a conexão JavaBeans como fonte de dados

O relatório é simples e o design proposto está na **Figura 16** mostrado a seguir.

Produtos Cadastrados		
ID	Nome do Produto	Preço
#{id}	#{nome} detail	#{preco}
pageFooter		'Página " + #V

Figura 16 – Design montado com o JavaBean Produto

Simplificando relatórios com Groovy

Groovy é uma linguagem dinâmica escrita para a plataforma Java, inspirada em linguagens como Ruby, Python e Smalltalk. Embora seja simples de se manipular e muito similar a linguagens de scripting, Groovy é um pouco mais complexo. Aplicações com Groovy podem ser pré-compiladas em bytecodes Java, para ser integrada à suas aplicações Java.

Graças a um Java Specification Request (JSR-241), Groovy se tornou a segunda linguagem oficial padrão do Java Virtual Machine (JVM), depois de Java, claro.

Desde a versão 0.6.6 do JasperReports, há suporte a linguagem Groovy. Com isso, quem ganha é o desenvolvedor, que tem em suas mãos uma maior simplicidade no desenvolvimento de expressões em seus relatórios. Por outro lado, há também uma aproximação de não-desenvolvedores, onde a utilização do iReport como ferramenta de relatórios passa a ser mais amigável em BI (Business Intelligence).

Neste capítulo você vai construir um relatório utilizando Groovy, trabalhando com expressões e aprenderá como rodar o relatório criado em suas aplicações também usando Groovy.

O resultado final do relatório utilizando a linguagem Groovy é mostrada a seguir, na **Figura 17**.



Produto	Preço	Qtd. Vendida	Total
Lombo Condimentado Fatiado	12,00	12	144,00
Sopa Americana de Cogumelo	8,99	10	89,90
Manteiga com Sal	6,99	5	34,95

Figura 17 – Resultado final do relatório com a linguagem Groovy

Criando gráficos e subdataset em seus relatórios

Como todo bom relatório, não poderia faltar os gráficos. Simples e eficazes, os gráficos exibem uma situação visualmente impactante, possibilitando um entendimento muito maior, e mais rico, do que apenas números e textos.

O JasperReports possui um suporte a vários formatos de gráficos, facilitando o desenvolvimento do relatório.

Neste capítulo será apresentado como criar gráficos em um relatório JasperReports usando o iReport, além de aprender a trabalhar com consultas diferentes no mesmo relatório, o que chamamos de Subdataset.

Criando um gráfico simples

O primeiro exemplo na geração de relatórios com gráficos um gráfico no formato de pizza 3D (Pie 3D). A **Figura 18** a seguir ilustra o resultado final.

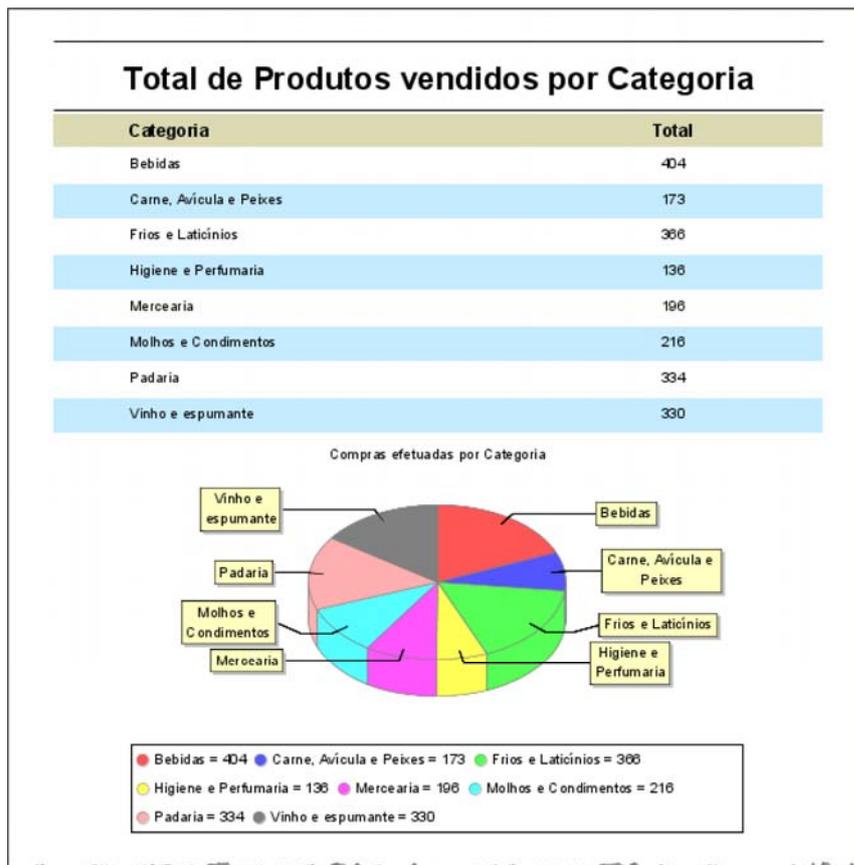


Figura 18 – Relatório gerado completo

Subdatasets

Subdatasets são fontes de dados secundárias que são adicionadas dentro de um relatório. São utilizadas até o momento para criar gráficos em relatórios cuja query principal está focada exclusivamente em alimentar o relatório.

Esta fonte de dados secundária tem seus próprios parâmetros, variáveis, fields, outro data source, além de sua própria instrução SQL (HQL, EJBQL, etc.).

Seu conceito é similar ao de sub-relatórios, onde a fonte de dados pode ser agrupada em um ou mais grupos, o que significa que podemos ter mais de um subdataset em um único relatório.

Utilizando Gráficos com Subdatasets em um relatório com Hibernate

Para criar o mesmo exemplo, mas utilizando o Hibernate, os passos são muito similares, mas com a diferença das entidades.

Internacionalizando relatórios e imagens de fundo

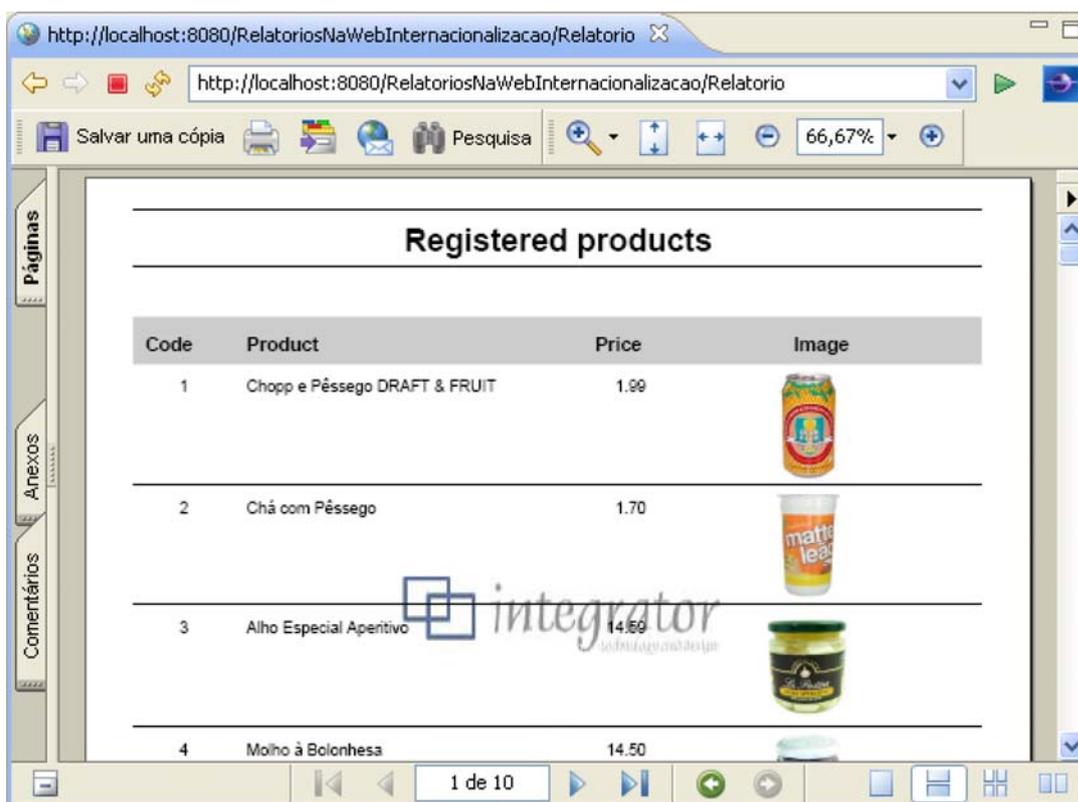
Internacionalizar aplicações atualmente é uma situação muito comum entre os desenvolvedores. Não seria diferente no caso do desenvolvimento de relatórios. O JasperReports introduziu esta característica a partir da versão 0.6.2 e desde então seu suporte foi acrescentado ao iReport, facilitando mais ainda seu desenvolvimento.

Outra situação comum é a criação de relatórios com imagens de fundo, no estilo marca d'água, onde muitas empresas a colocam para exibir sua logo marca.

Este capítulo ensinará você a criar relatórios internacionalizados e também a adicionar imagens de fundo em seus relatórios.

Registered products			
Code	Product	Price	Image
73	Português Branco DÃO TERRAS ALTAS	21.99	

Figura 19 – Relatório com a exibição a internacionalização aplicada



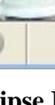
Registered products			
Code	Product	Price	Image
1	Chopp e Pêssego DRAFT & FRUIT	1.99	
2	Chá com Pêssego	1.70	
3	Alho Especial Aperitivo	14.50	
4	Molho à Bolonhesa	14.50	

Figura 20 – Execução final do relatório no navegador dentro do Eclipse IDE

Crosstabs

Crosstabs (cross-tabulation), no JasperReports, são tabelas cruzadas, muito comuns no desenvolvimento de relatórios com a utilização de planilhas eletrônicas, também conhecidas por tabelas dinâmicas ou tabelas de referência cruzada.

Diferente dos demais relatórios criados até o momento no livro, um relatório crosstab possui uma tabela contendo um número de linhas e colunas que não se conhece em tempo de design.

Graças ao uso do iReport, você pode contar com o uso de um assistente, tornando esta tarefa não muito complicada.

Neste capítulo você aprenderá a criar crosstabs no iReport, modificar, formatar e usar também subdatasets para relatórios mais complexos.

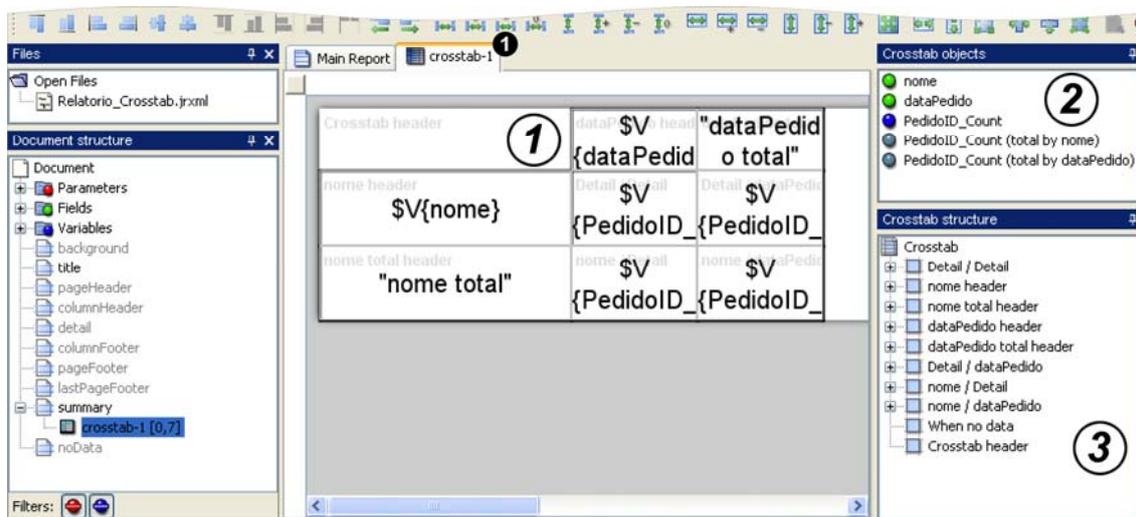


Figura 21 – Detalhes da guia crosstab

	1996	1997	1998	Total
Pirkko Koskitalo	4	10	1	15
Renate Messner	3	8	4	15
Rene Phillips	2	4	4	10
Yvonne Moncada	0	2	3	5
Zbyszek Piestrzeniewicz	1	2	4	7
nome total	152	408	270	830

Figura 22 – Resultado das alterações na execução do relatório

Ao executar o relatório, você terá um resultado similar ao mostrado na **Figura 23** a seguir:

	1996							1997	
	07	08	09	10	11	12	Total	01	02
Pedro Afonso	0	1	0	0	0	0	1	0	0
Peter Franken	1	0	0	2	0	1	4	0	0
Philip Cramer	0	0	0	2	0	0	2	0	2
Yvonne Moncada	0	0	0	0	0	0	0	1	0
Zbyszek Piestrzeniewicz	0	0	0	0	0	1	1	0	0
Total Geral	22	25	23	26	25	31	152	33	29

Figura 23 – Resultado final na execução do relatório

Criando um Crosstab com Subdataset

Para este exemplo, você precisará criar um novo relatório. Vamos partir da hipótese que você deseja visualizar um relatório contendo as compras efetuadas por cada cliente, agrupados por produto.

Relatório de Vendas

Relatório de compras agrupadas por produto

Compras do produto Aceto Italiano Bâlsâmico

	1996	1997	1998	Total
Ann Devon	0	0	1	1
Elizabeth Brown	0	0	1	1
Giovanni Rovelli	0	1	0	1
Jose Pavardti	0	0	1	1
Liz Nixon	0	1	0	1
Paula Wilson	1	0	0	1
Pedro Afonso	1	0	0	1
Roland Mendel	2	0	0	2
Yvonne Moncada	0	0	1	1
Total Geral	4	2	4	10

Compras do produto Alho Especial Aperitivo

	1996	1997	1998	Total
Christina Berglund	0	0	1	1

Figura 24 – Relatório contendo Crosstab e Subdataset

Outras características da geração de relatórios

Gerar relatório JasperReports com o iReport é uma tarefa muito fácil, como você vem acompanhando ao longo do livro.

Neste capítulo você conhecerá algumas outras características da ferramenta iReport, que não foram apresentadas até o momento, como tradução do programa, compilação de relatórios grandes, formatação tanto com estilos próprios do JasperReports como em HTML, além da resolução de problemas comuns na construção de relatórios.

Tratando relatórios grandes

O virtualizador de relatórios é uma forma de minimizar a criação de relatórios JasperReports grandes. Esta opção ajuda a minimizar a limitação física, economizando na memória usada e empregando outras formas avaliadas (geralmente o disco rígido) para salvar páginas completas durante a impressão de um relatório. Isso evita que relatórios grandes recebam a mensagem de exceção “**Out of Memory**”.

Exibindo somente um grupo por página

Você aprendeu a criar grupos e percebeu o quanto isto é simples. Mas talvez você queira exibir cada grupo por página.

Reiniciando o número de páginas

Caso o seu grupo seja efetuado por datas, por exemplo, talvez você queira reiniciar o número de páginas e total de páginas a cada novo grupo.

Alinhamento de objetos no design do relatório

O iReport possui uma característica comum em ferramentas de desenho, ao qual podemos utilizar linhas guias (*guidelines*), para colocar os elementos dentro do relatório perfeitamente alinhados às réguas.

Formatação entre textos no seu relatório

Em alguns casos, você vai precisar embeber em um texto uma formatação de cor, tamanho, negrito ou até mesmo sublinhado.

```
Tags HTML Suportadas :
<font size="12">Tamanho</font>
<font color="red">Cor</font>
<font face="Comic Sans MS">Tipo de Fonte</font>
<b>bold</b>
<i>italic</i>
<u>underlined</u>
Quebra
<br/>
A bulleted list:
<li>item 1</li>
```

Tags HTML Suportadas :

Tamanho
Cor
 Tipo de Fonte
 bold
italic
underlined
 Quebra

```
Este é um <style isBold="true" isItalic="true" isUnderline="true" pdfFontName="Helvetica-BoldOblique">Static Text</style> formatado. O <style isStrikeThrough="true">Preço</style> foi cortado.
```

Este é um Static Text formatado. O ~~Preço~~ foi cortado.

```
'Tags HTML dentro do Text Field :<br/>'+
'<font size="12">Tamanho</font><br/>'+
'<font color="red">Cor</font><br/>'+
'<font face="Comic Sans MS">Tipo de Fonte</font><br/>'+
'<b>bold</b>'
```

Tags HTML dentro do Text Field :

Tamanho
Cor
 Tipo de Fonte
 bold

Figura 25 – Campos no Design e na execução do relatório

Apêndice A - Exportando para outros formatos

Este apêndice é uma referência sobre a API do JasperReports na exportação para outros formatos através de sua aplicação, seja desktop ou Web.

Os formatos discutidos aqui não são todos os suportados para exportação pelo JasperReports, mas são os principais: **RTF, XLS, CSV, HTML e ODF**.