

# Mini-Tutorial

## Como criar relatórios Java para Web com JasperReports e iReport

por Roberto J. Furutani  
20/07/2005  
www.furutani.eti.br

### 1. Introdução

O objetivo deste trabalho é mostrar de forma prática como criar relatórios e subrelatórios para o ambiente Web usando Java. Será apresentado neste mini-tutorial o JasperReports e iReport que são os componentes mais conhecidos para a geração de relatórios em java, por serem open-source e terem muitos recursos.

O iReport é uma ferramenta para facilitar os testes e principalmente facilitar a criação do layout dos relatórios. Mas quem gera de fato os relatórios em pdf, xls, html e outros formatos é o JasperReports.

### 2. Obtendo os programas

A versão do iReport que será usado nesse mini-tutorial é 0.4.0 e do JasperReports (que acompanha o iReport) é 0.6.1.

Para fazer o download do iReport acesse <http://ireport.sourceforge.net/> ou diretamente no SourceForge [http://sourceforge.net/project/showfiles.php?group\\_id=64348](http://sourceforge.net/project/showfiles.php?group_id=64348)

Após o download descompacte o arquivo iReport-0.4.0.zip em algum diretório, neste trabalho será usado D:\iReport-0.4.0

Entre no site [www.mysql.org](http://www.mysql.org) e faça o download do MySQL database server 4.1.

No site [www.eclipse.org](http://www.eclipse.org) pegue o Eclipse 3.1 ou mais recente.

OBS: Nesta versão do iReport a versão do driver do MySQL é antiga, para atualizar, na pasta D:\iReport-0.4.0\lib apague o *mysql-connector-java-3.0.8-stable-bin.jar* e insira o *mysql-connector-java-3.0.16-ga-bin.jar* que acompanha os fontes.

### 3. Criando as tabelas

Os dados usados serão bem simples, são apenas 3 tabelas (livro, livro\_autor e autor), suficientes para mostrar como gerar. As tabelas estão no banco de dados *relatório*.

Abaixo está o conjunto de comandos SQL para a criação das tabelas. Os fontes estão disponíveis em [http://www.furutani.eti.br/tutoriais/relatorio\\_jasperreports.zip](http://www.furutani.eti.br/tutoriais/relatorio_jasperreports.zip)

```
CREATE TABLE `autores` (
  `cd_autor` int(11) unsigned NOT NULL auto_increment,
  `Nome` varchar(60) default NULL,
  PRIMARY KEY (`cd_autor`),
  UNIQUE KEY `cd_autor` (`cd_autor`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `livro_autor` (
  `cd_livro` int(11) unsigned NOT NULL default '0',
  `cd_autor` int(11) unsigned NOT NULL default '0',
  PRIMARY KEY (`cd_livro`,`cd_autor`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `livros` (
  `cd_livro` int(11) unsigned NOT NULL auto_increment,
  `Titulo` varchar(100) default NULL,
  `ISBN` int(15) default NULL,
```

```

`DataLancamento` datetime default NULL,
PRIMARY KEY (`cd_livro`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

INSERT INTO `autores` (`cd_autor`, `Nome`) VALUES
(1,'Kathy Sierra'),
(2,'Bryan Basham'),
(3,'Bert bates'),
(4,'Carlos H. Poderoso de Oliveira');

INSERT INTO `livros` (`cd_livro`, `Titulo`, `ISBN`, `DataLancamento`) VALUES
(1,'Head First Servlets & JSP',596005407,'2004-07-01 00:00:00'),
(2,'SQL Curso Prático',857522041,'2005-02-10 00:00:00');

INSERT INTO `livro_autor` (`cd_livro`, `cd_autor`) VALUES
(1,1),
(1,2),
(1,3),
(2,4);

```

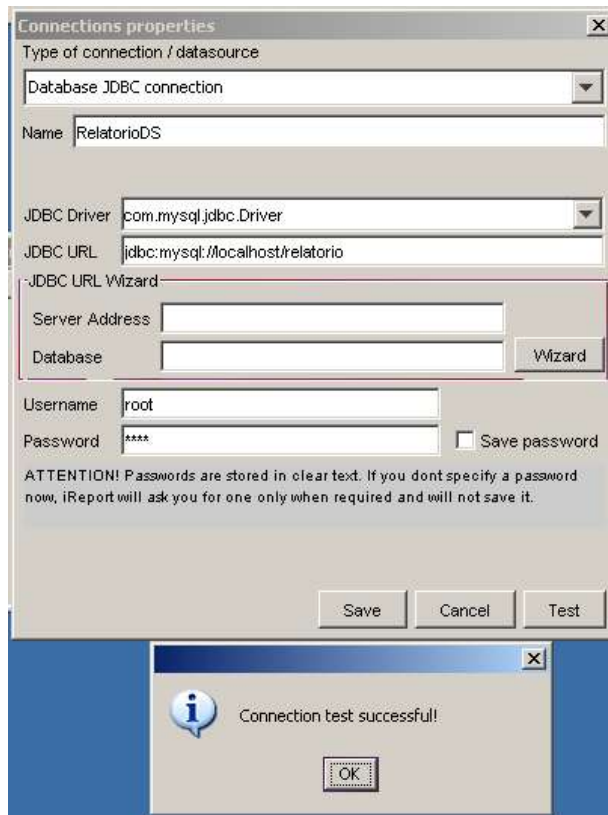
#### 4. Montando o relatório

Antes de criar o relatório master ou relatório principal, é preciso configurar um data source (fonte de dados) que neste caso será o mysql.

Vá ao menu fonte de dados e selecione Conexões/Fonte de dados.



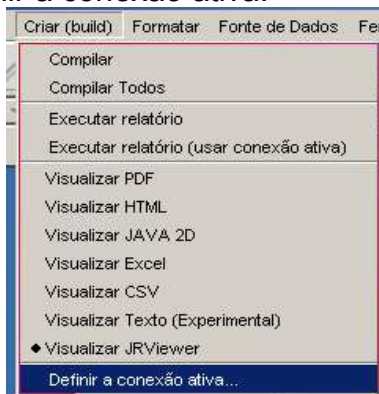
Na janela que se abre clique em new e configure as propriedades conforme a figura abaixo:



Se o teste foi bem sucedido, clique em save.

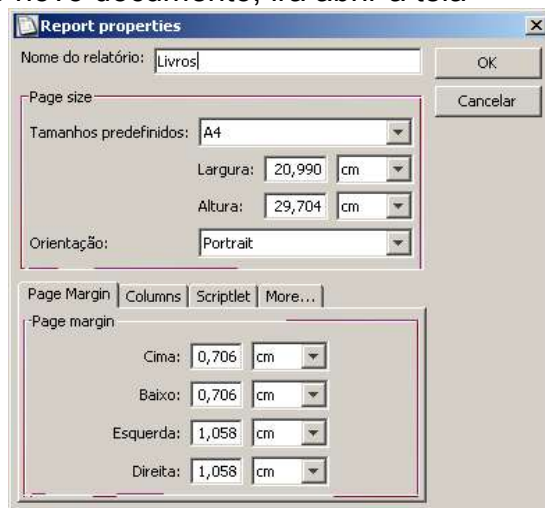


Agora é preciso definir uma conexão ativa, que será a conexão criada anteriormente. No menu Criar(build) clique em Definir a conexão ativa.





Selecione RelatorioDS e clique em OK.

Agora é possível prosseguir com a criação do relatório principal. No menu arquivo selecione novo documento, irá abrir a tela



Altere somente o nome do relatório.

Na barra de ferramentas procure por static text tool , esse ferramenta possibilita inserir texto estáticos, ou seja, textos que não virão do banco de dados, no relatório. Seu uso é simples, basta clicar nele e depois desenhar no relatório.

Ao lado do static text tool tem o textfield tool , este sim serve para pôr textos que virão do banco de dados.

Coloque no relatório os texto estáticos “Relatório de Livros” na banda title e os texto ”Titulo:”, “ISBN:” e “Data de Lançamento” na banda detail. A banda detail é onde colocamos os textos que se repetirão para cada linha retornada pelo banco de dados.

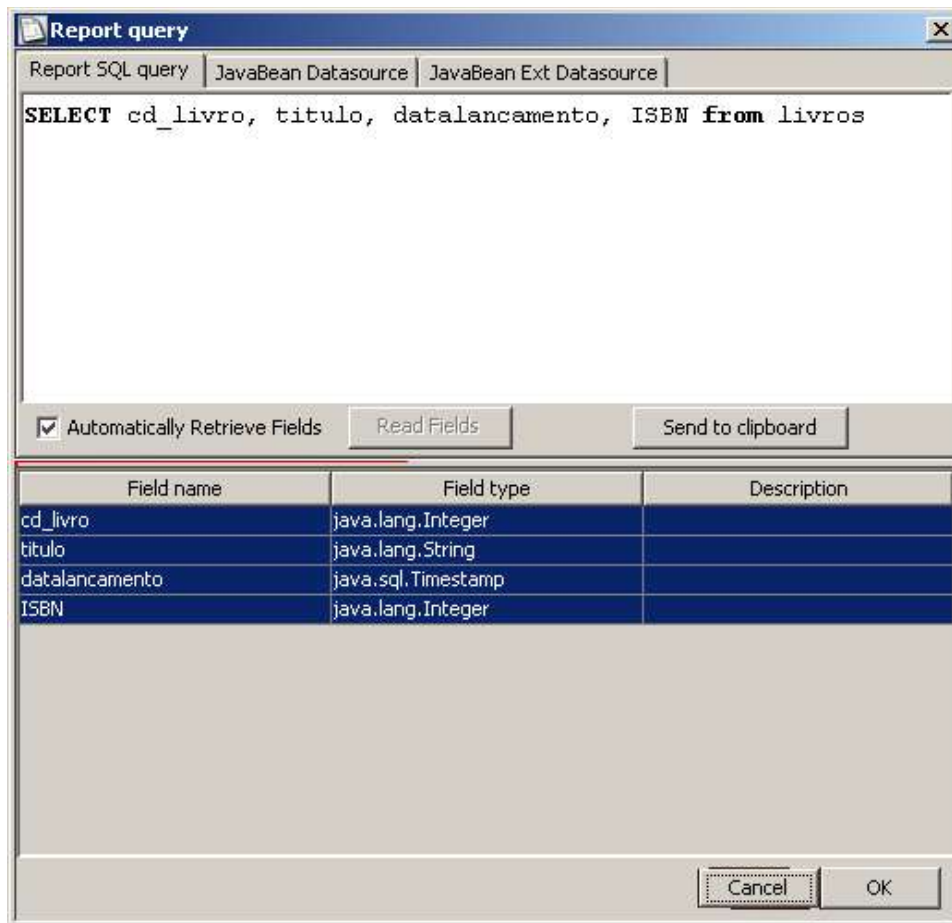
Coloque também três textfields.

Faça algo parecido com a figura abaixo.



Precisamos dizer a esses três textfields o que eles mostrarão.

Para isso vamos configurar um comando SQL para puxar os dados do mysql. Vá ao menu visualizar, selecione Consultas do Relatório, vai abrir a janela:



Provavelmente quando estiver digitando o comando SQL vai abrir uma janela com o desenho de duas chaves, ele está pedindo a senha do mysql. Digite-a.

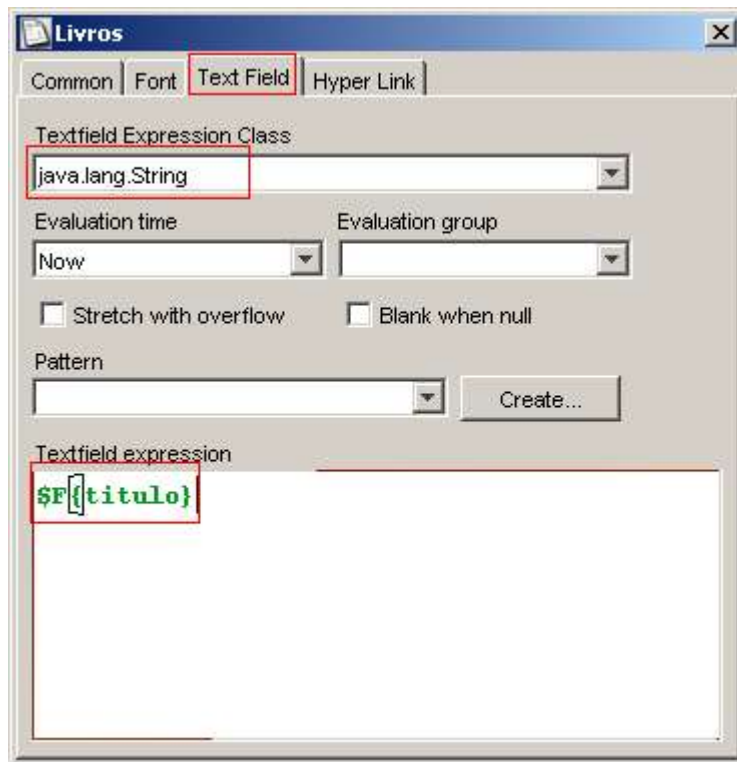
No final da digitação do SQL, deverá aparecer em baixo quatro campos e seus respectivos tipos, guarde bem o nome deles você irá utiliza-los mais para frente, senão aparecer, o SQL deve estar errado.

Clique em OK.

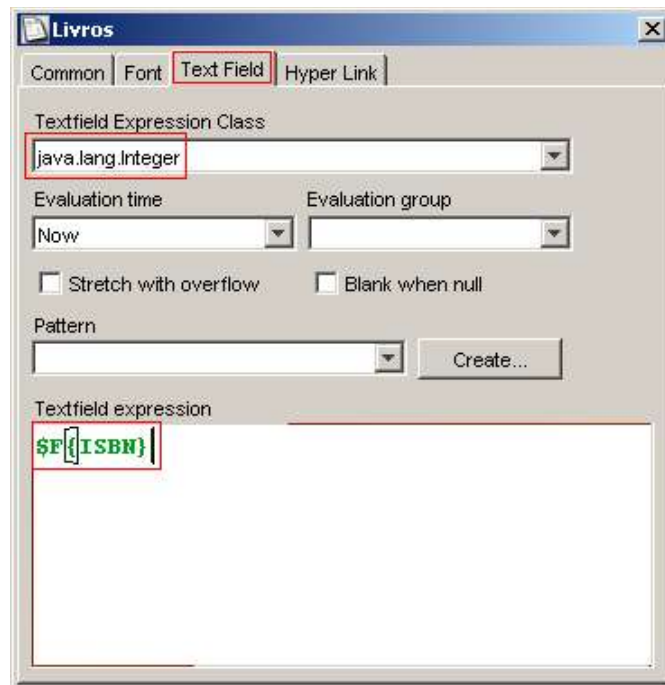
Agora dê um duplo clique em cima do text field referente ao “título” do livro.

A janela abaixo vai se abrir:

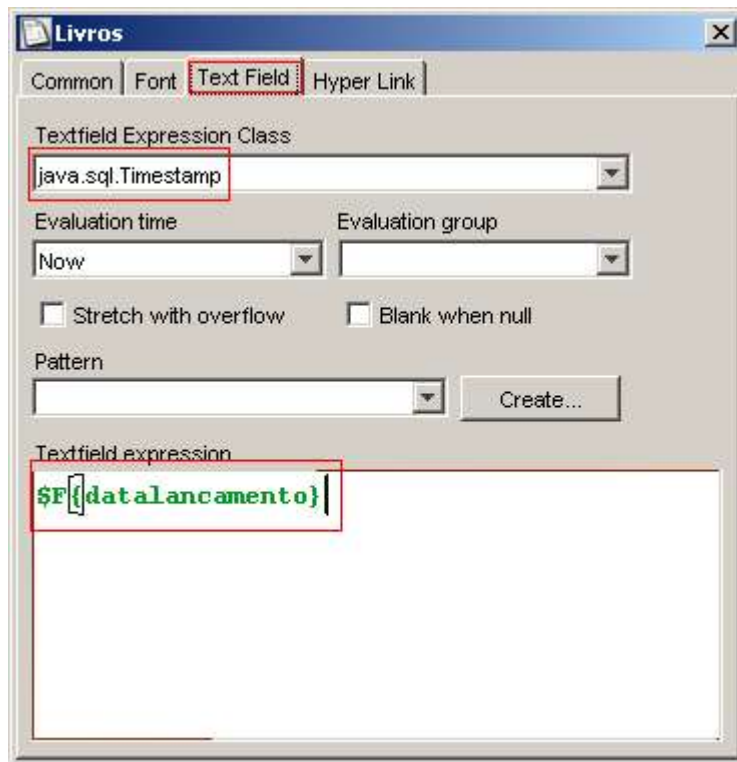
Faça as configurações destacadas na figura.



Feche essa janela e dê um duplo clique no textfield referente ao ISBN e faça como mostrado na figura abaixo.




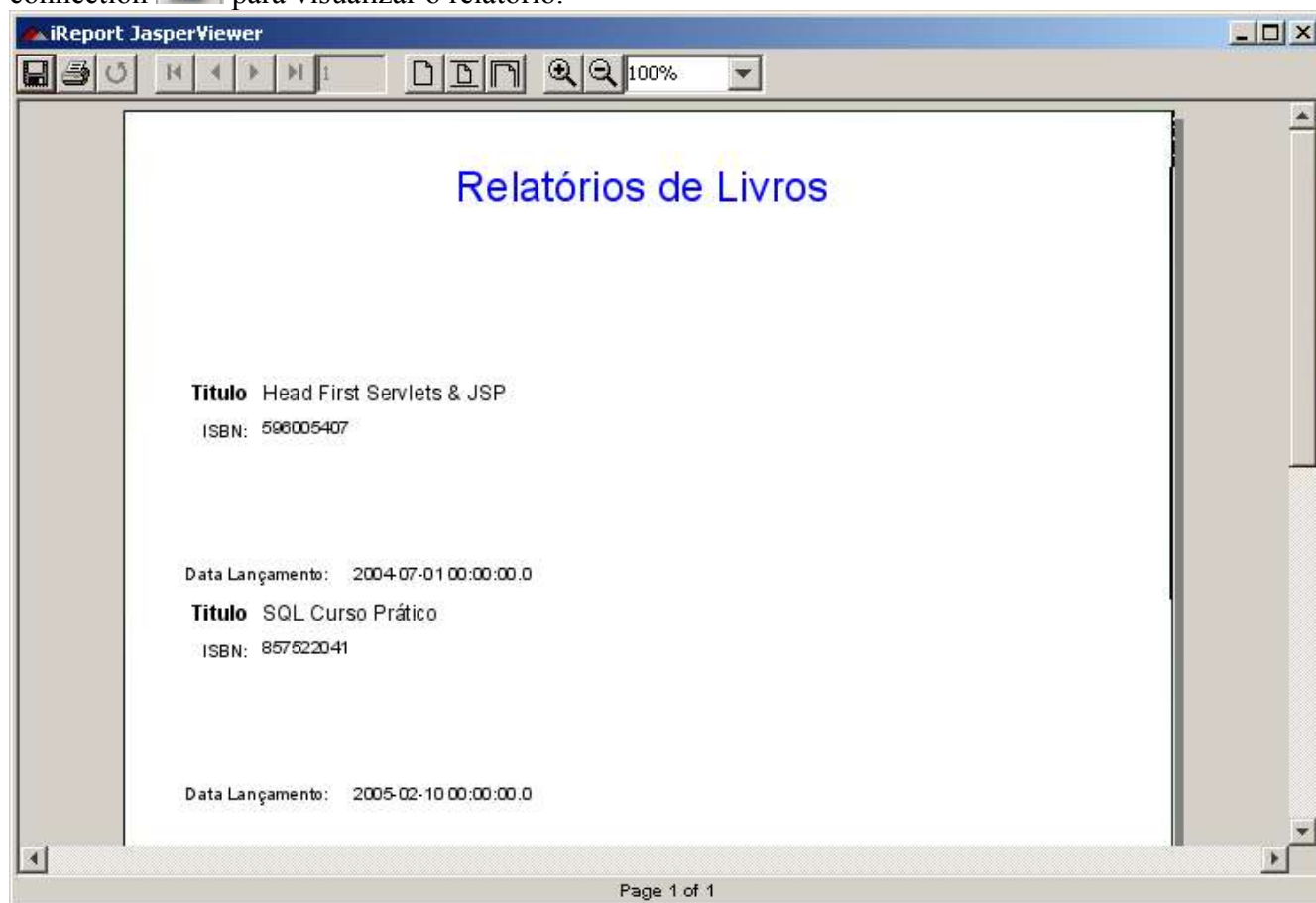
Faça o mesmo para o textfield dataLancamento



O conteúdo da propriedade textfield expression deve coincidir com o nome do campo obtido quando digitamos o SQL. O Relatório deve ficar assim:

<b>Relatórios de Livros</b>	
pageHeader	
columnHeader	
<b>Título:</b> \$F{titulo}	detail
<b>ISBN:</b> \$F{ISBN}	
<b>Data Lançamento:</b> \$F{dataLancamento}	
columnFooter	
pageFooter	
lastPageFooter	
summary	

Neste momento o relatório já está funcionando mas sem o subreport, clique em Run a report using a connection  para visualizar o relatório.



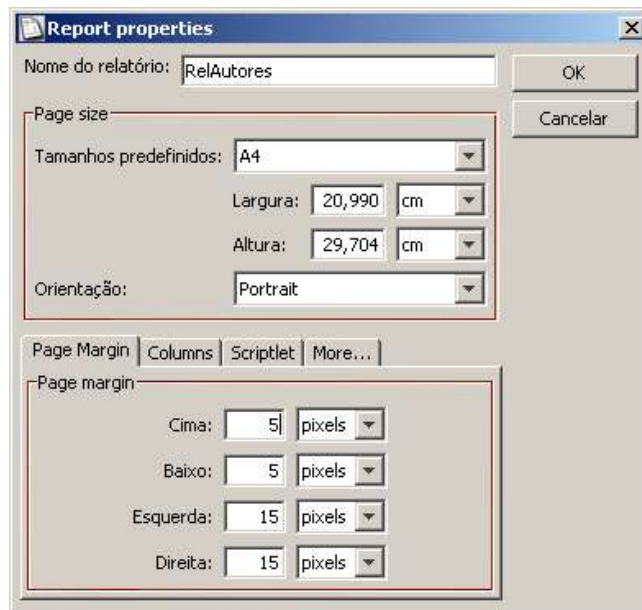
O relatório principal já está pronto.

## 5. Montando o subrelatório

O subrelatório (subreport) é simples, só terá um textfield para o nome do autor.

Crie um novo relatório, menu arquivo selecione novo documento.



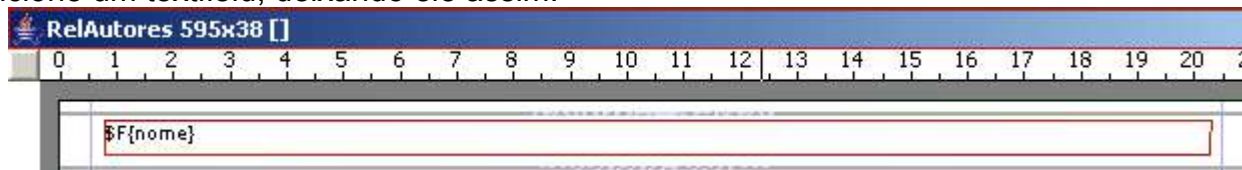


Altere as propriedades das margens da página (Page Margin).

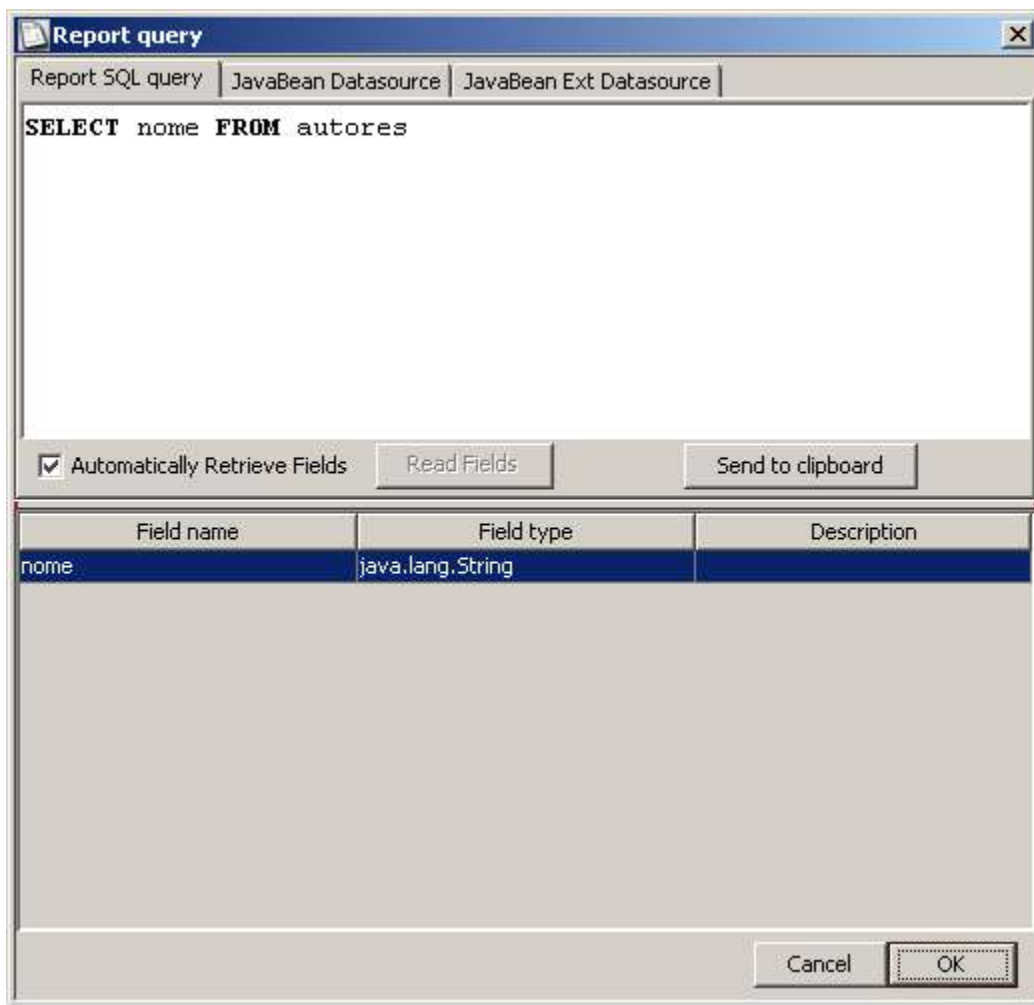
Defina a conexão ativa (já foi mostrado).

Reduza o tamanho de todos as bands, para ficar só o detail visível.

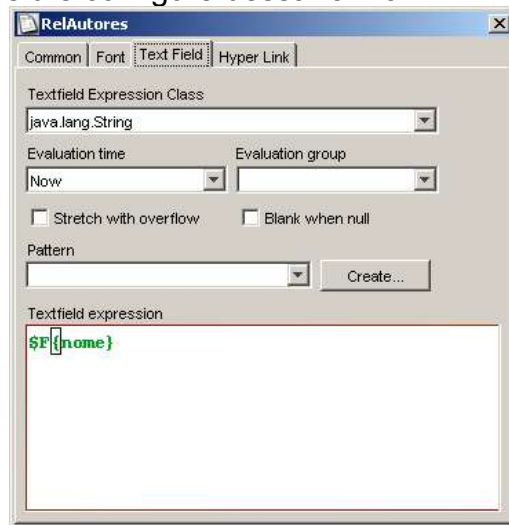
Adicione um textfield, deixando ele assim:




Vá ao menu Visualizar selecione Consultas do Relatório e insira o código SQL mostrado abaixo.



Se tudo estiver certo clique em OK.  
Dê um duplo clique no text field e configure dessa forma:

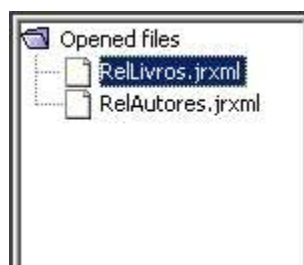


O relatório já deve estar funcionando, clique em Run a report using a connection .  
O resultado será:



Desta forma, o subrelatório irá mostrar todos os autores, o que se deseja é mostrar os autores de acordo com o livro. Veremos como fazer isso no próximo tópico.

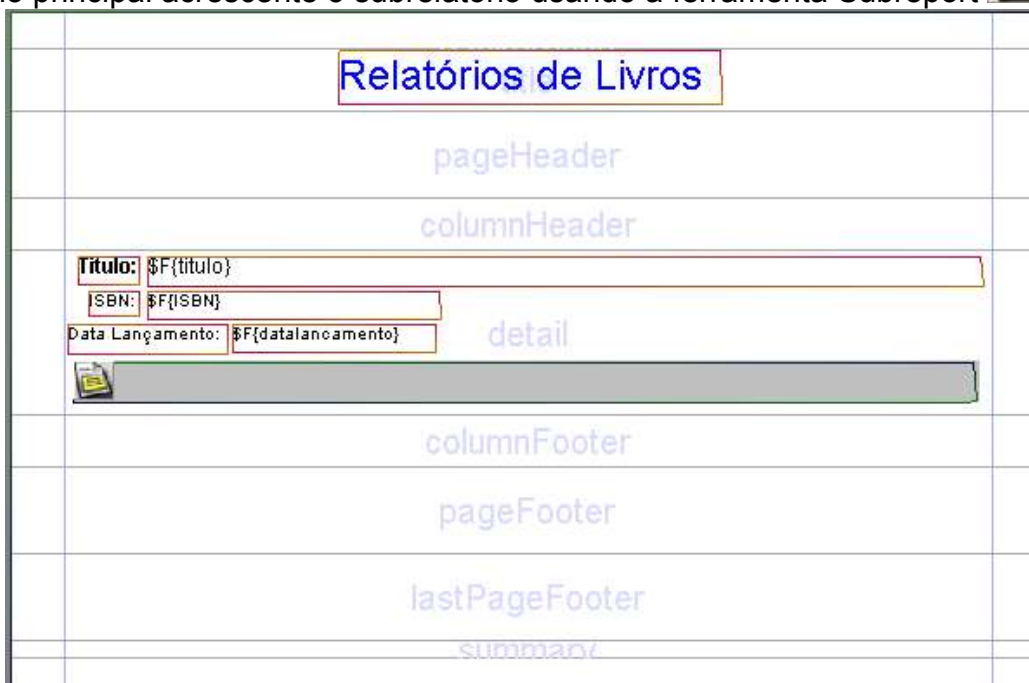
Observação: Para alternar entre um relatório e outro dê um duplo clique no nome dele na árvore que fica do lado esquerdo



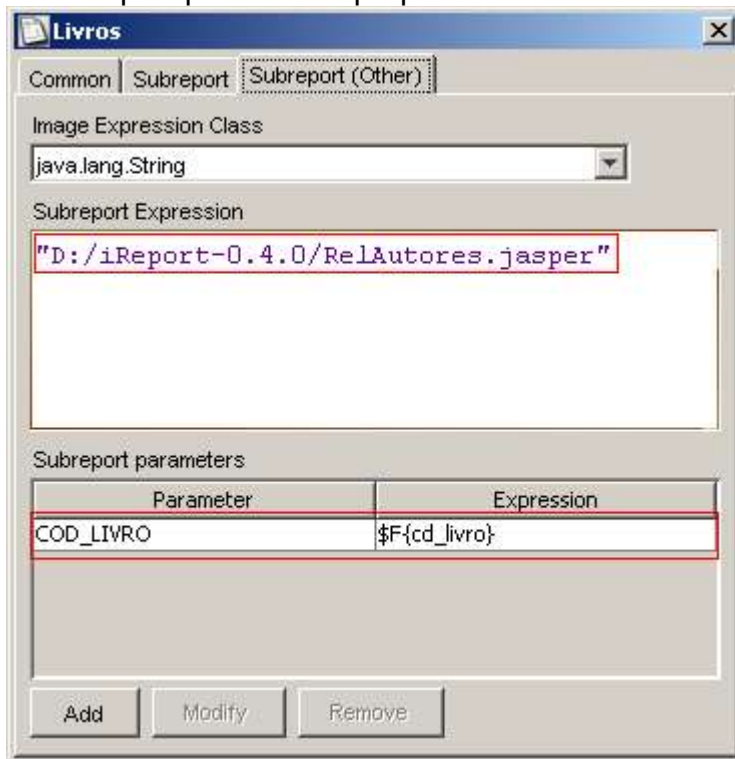
## 6. “Ligando” o relatório ao subrelatório

Agora chegou a parte mais legal do tutorial.

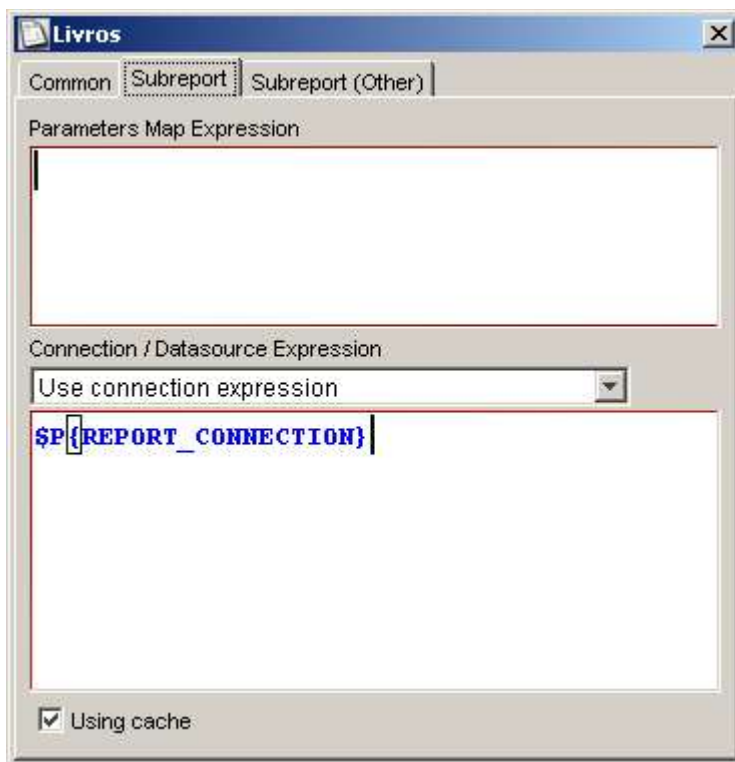
No relatório principal acrescente o subrelatório usando a ferramenta Subreport



Dê um duplo clique no subreport para ver as propriedades dele.



No campo Subreport Expression informe o caminho do relatório compilado (.jasper). Logo abaixo é configurado os parâmetros que serão **passados** para o subrelatório. Neste caso estamos passando o código do livro (\$F{cd\_livro}) com o nome de COD\_LIVRO. Ainda nessa janela selecione a aba Subreport e selecione Use connection expression.

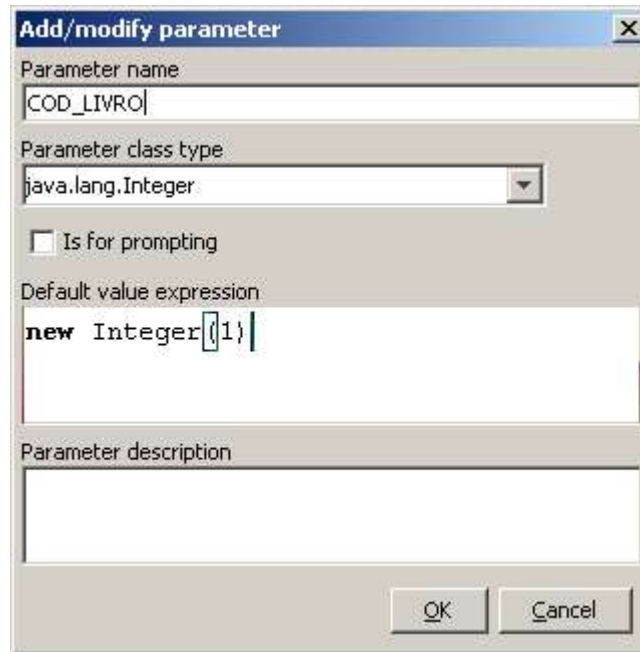


Volte ao subreport, pois é necessário fazer uma modificação.

O `$P{COD_LIVRO}` é o nome do parâmetro que foi definido anteriormente.

Para que o subrelatório aceite o parâmetro `$P{COD_LIVRO}` é preciso configurá-lo então, vá ao menu visualizar selecione parâmetros do relatório.

E configure assim:



**Add/modify parameter**

Parameter name  
COD\_LIVRO

Parameter class type  
java.lang.Integer

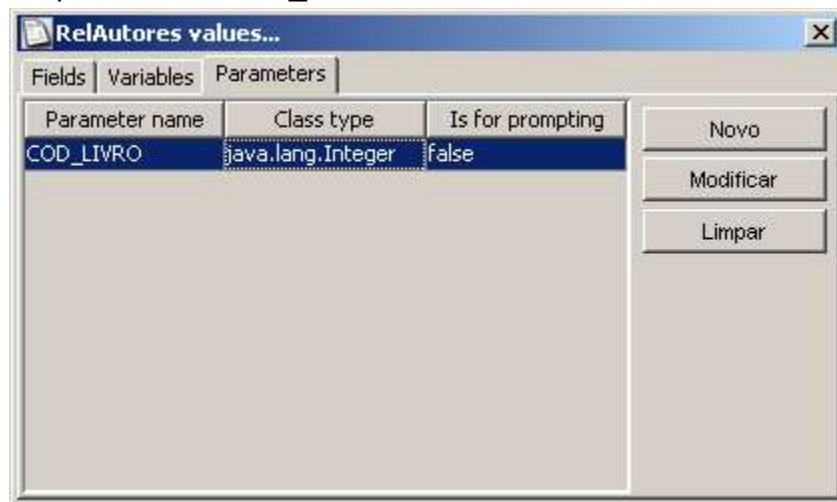
Is for prompting

Default value expression  
new Integer({1})

Parameter description

OK Cancel

O valor para o campo Default value expression é somente para poder testar o comando SQL abaixo, caso não coloque um valor padrão para o parâmetro verá um erro: Please set a default value for the parameter 'COD\_LIVRO'.



**RelAutores values...**

Fields | Variables | Parameters

Parameter name	Class type	Is for prompting
COD_LIVRO	java.lang.Integer	false


Novo  
Modificar  
Limpar

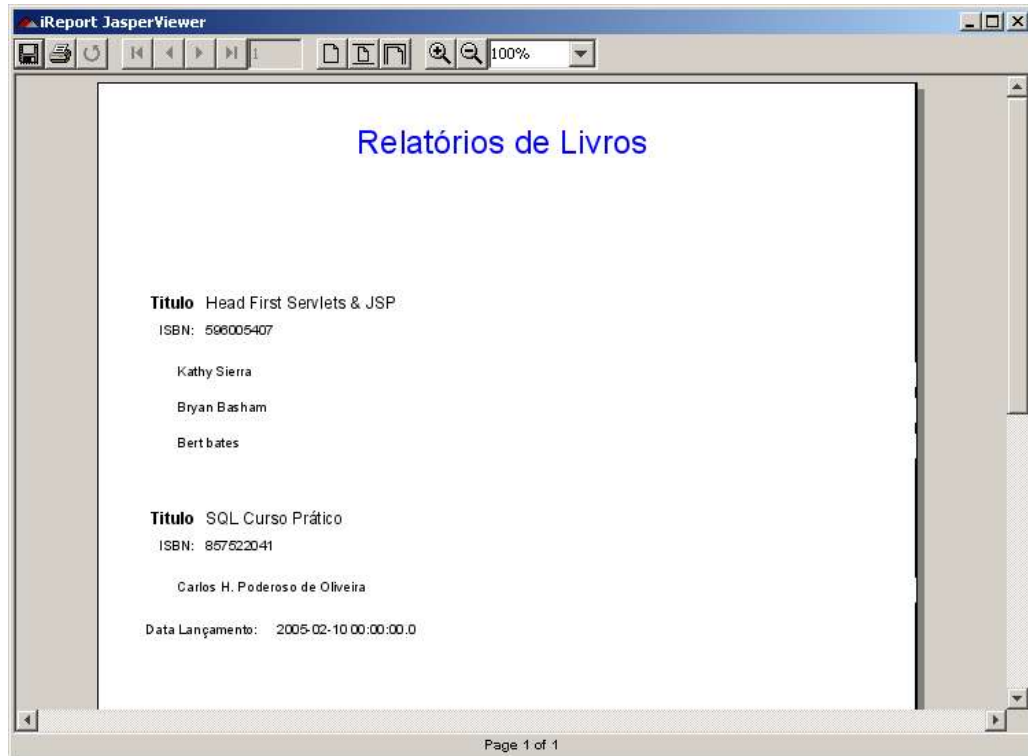
A depois mude o SQL para:

```
SELECT at.nome FROM `autores` at
INNER JOIN `livro_autor` la ON la.cd_autor = at.cd_autor
INNER JOIN `livros` lv ON lv.cd_livro = la.cd_livro
WHERE lv.cd_livro = $P{COD_LIVRO}
```

Compile o relatório clicando em



Volte ao relatório principal, clique em Run a report using a connection , se tudo correr bem esse será o resultado

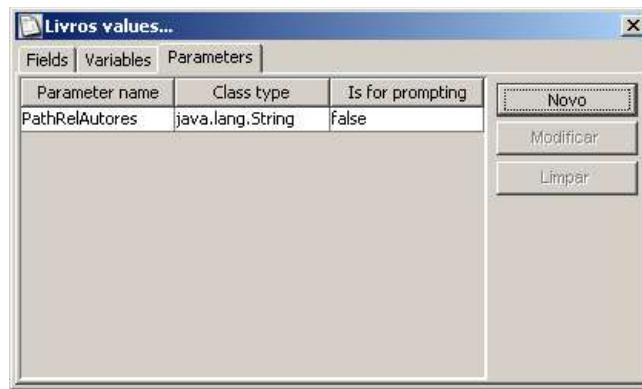


## 7. Colocando na web

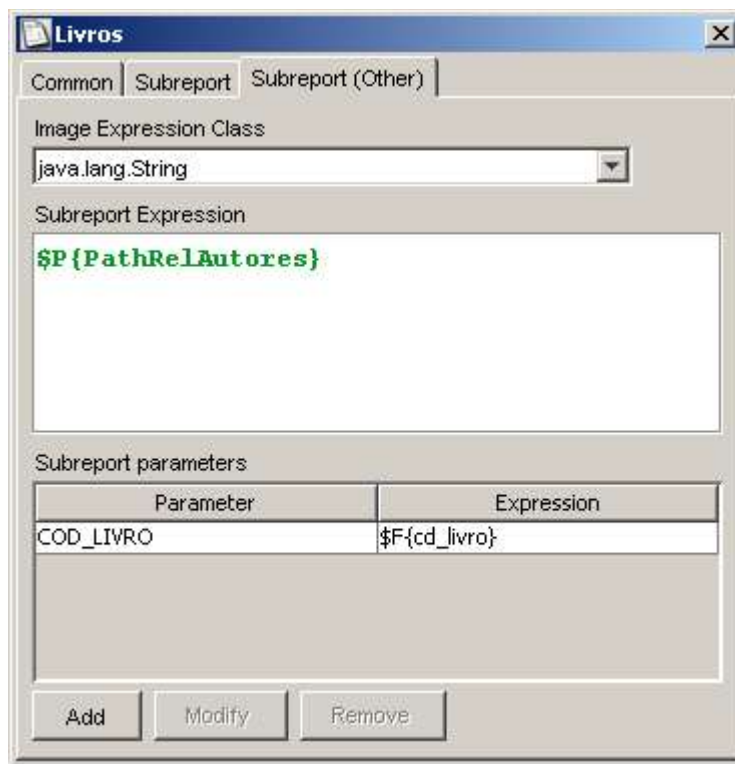
8. A parte mais fácil do tutorial é agora.  
Colocar esse relatório para ser gerado na web.  
Saber um pouco de servlets e JSP é pré-requisito para este tópico.


Lá atrás colocamos como caminho para o subreport a constante "[D:/iReport-0.4.0/RelAutores.jasper](#)", agora não pode mais ser assim. É preciso passar o caminho para o RelAutores.jasper por parâmetro, pois o nome do contexto entre outros fatores podem alterar esse caminho.

No relatório principal vá ao menu visualizar, selecione Parâmetros do Relatório e crie um novo parâmetro.



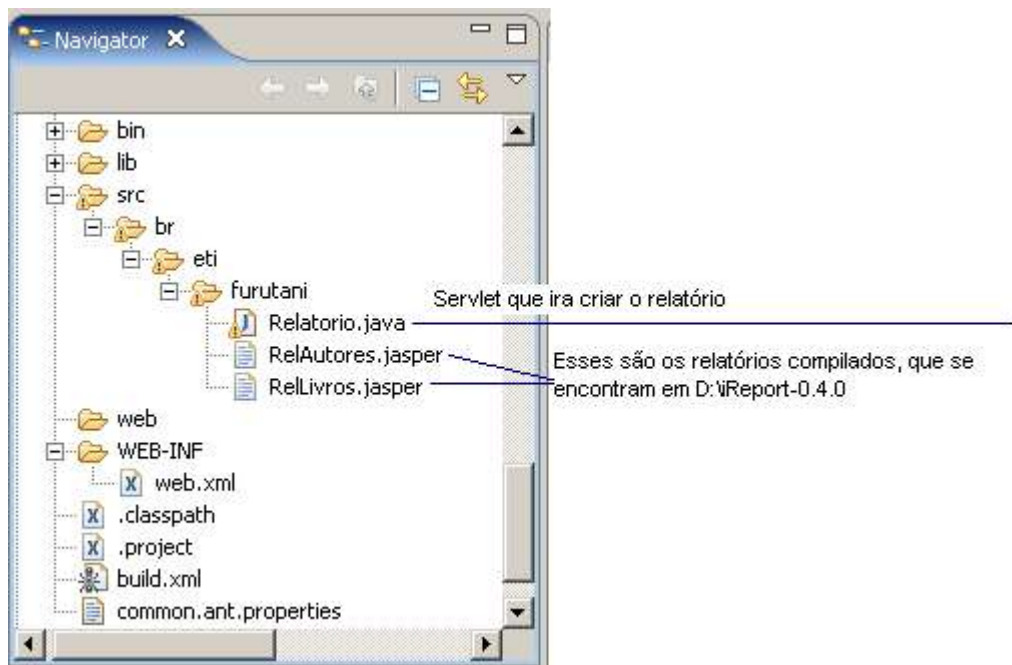
Dê um duplo clique no subreport, vá na aba Subreport (other) e substitua “[D:/iReport-0.4.0/RelAutores.jasper](#)” por `$P{PathRelAutores}`



Compile o relatório clicando em . Não rode. Se você tentar rodar vai dar um erro ou vai criar um relatório sem páginas.

Nessa fase utilizaremos o Eclipse 3.1 para codificar um Servlet e o tomcat 5.5.7.

Foi criado a seguinte estrutura de diretórios  
(Os detalhes podem ser analisados baixando os fontes em [http://www.furutani.eti.br/tutoriais/relatorio\\_jasperreports.zip](http://www.furutani.eti.br/tutoriais/relatorio_jasperreports.zip))



O servlet que foi criado para gerar o relatório está listado abaixo com alguns comentários.

```

package br.eti.furutani;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperManager;
import net.sf.jasperreports.engine.JasperPrint;

public class Relatorio extends HttpServlet {

    public static Connection con;
    public static String banco = "relatorio"; // Nome do banco de dados
    public static String usuario = "root"; // Usuario do banco
    public static String senha = "4122"; // Senha

    protected void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        conexao(); // Faz a conexão

        // Na variavel pathJasper ficara o caminho do diretório para
        // os relatórios compilados (.jasper)
        String pathJasper = getServletContext().getRealPath(
            "/WEB-INF/classes/br/eti/furutani/"
            + "/");

        // A variavel path armazena o caminho real para o contexto
        // isso é util pois o seu web container pode estar instalado em lugares diferentes
        String path = getServletContext().getRealPath("/");

        // Parametros do relatorio

```



```

Map parametros = new HashMap();
// O único parametro que devemos passar é o PathRelAutores
// é o caminho pro RelAutores.jasper (que foi substituido pelo
// valor fixo "D:/iReport-0.4.0/RelAutores.jasper")
parametros.put("PathRelAutores", pathJasper + "RelAutores.jasper");

try {
    // Aqui ele cria o relatório
    JasperPrint impressao = JasperFillManager.fillReport(pathJasper
        + "RelLivros.jasper", parametros, con);

    // Grava o relatório em disco em pdf
    JasperManager.printReportToPdfFile(impressao, path
        + "/RelatorioLivros.pdf");

    // Redireciona para o pdf gerado
    res.sendRedirect("RelatorioLivros.pdf");

} catch (Exception e) {
    res.getWriter().println("Erro ao gerar o relatório: " + e);
}

// Cria a conexão
public void conexao() {
    try {
        if (con == null || con.isClosed()) {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/" + banco, usuario, senha);
        }
    } catch (Exception e) {
        System.out.println("não foi possível conectar ao banco ->");
        e.printStackTrace();
    }
}

protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doPost(request, response);
}
}

```

Só falta fazer o mapeamento desse servlet no web.xml

```

<servlet>
  <servlet-name>RelatorioServlet</servlet-name>
  <servlet-class>br.eti.furutani.Relatorio</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>RelatorioServlet</servlet-name>
  <url-pattern>/RelLivro</url-pattern>
</servlet-mapping>

```

Faça o deploy da aplicação e acesse <http://127.0.0.1:8080/exemplo/RelLivro>

OBS: Para quem for usar o build do ANT, não se esqueça de trocar o valor do atributo `Deploy.Dir` no `common.ant.properties`

## 8. Referências

- <http://ireport.sourceforge.net/>
- <http://jasperreports.sourceforge.net/>
- [www.guj.com.br](http://www.guj.com.br)
- [www.javafree.com.br](http://www.javafree.com.br)
- [http://geocities.yahoo.com.br/robertofurutani/java/Tutorial\\_JasperReports/](http://geocities.yahoo.com.br/robertofurutani/java/Tutorial_JasperReports/)
- [www.eclipse.org](http://www.eclipse.org)