

Maven

```
<dependency>  
  <groupId>mysql</groupId>  
  <artifactId>mysql-connector-java</artifactId>  
  <version>8.0.21</version>  
</dependency>  
  
<dependency>  
  <groupId>net.sf.jasperreports</groupId>  
  <artifactId>jasperreports</artifactId>  
  <version>6.16.0</version>  
</dependency>
```

Maven

artifactId é o nome do jar sem versão. Se você o criou, pode escolher o nome que quiser com letras minúsculas e sem símbolos estranhos. Se for um frasco de terceiros, você deve levar o nome do frasco conforme é distribuído. por exemplo. maven, matemática comum

groupId identificará seu projeto exclusivamente em todos os projetos, portanto, precisamos aplicar um esquema de nomenclatura. Ele deve seguir as regras de nome de pacote, o que significa que deve ser pelo menos um nome de domínio que você controla, e você pode criar quantos subgrupos desejar. Veja Mais informações sobre nomes de pacotes. por exemplo:

unc.com.br

br.com.unc.sistemaacademico

Jasper Studio

<https://community.jaspersoft.com/project/jasperreports-library/releases>

<https://sourceforge.net/projects/jasperreports/files/latest/download>

Fluxo

Arquivo .jrxml



Arquivo .jasper

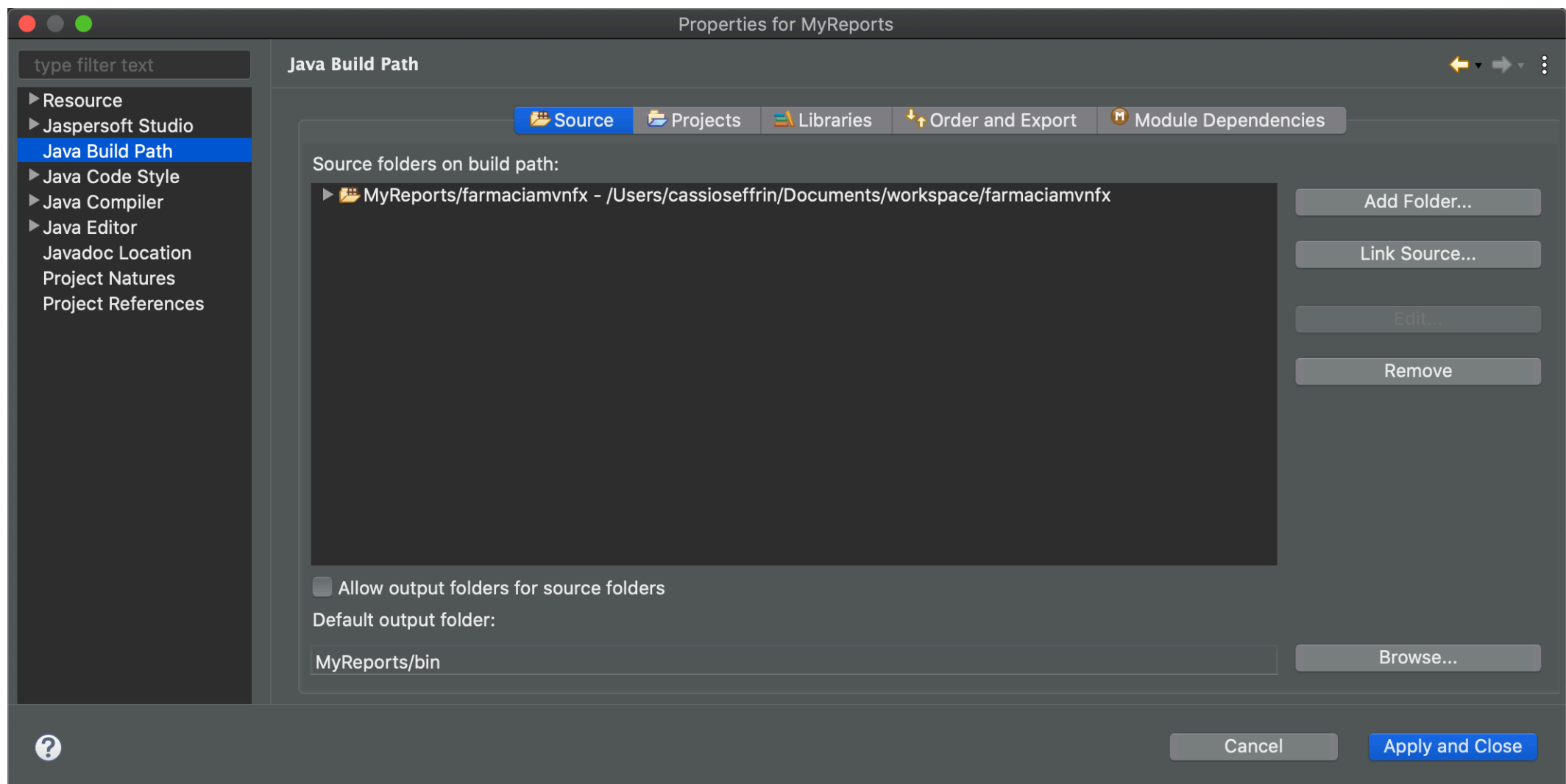


Arquivo .jrprint



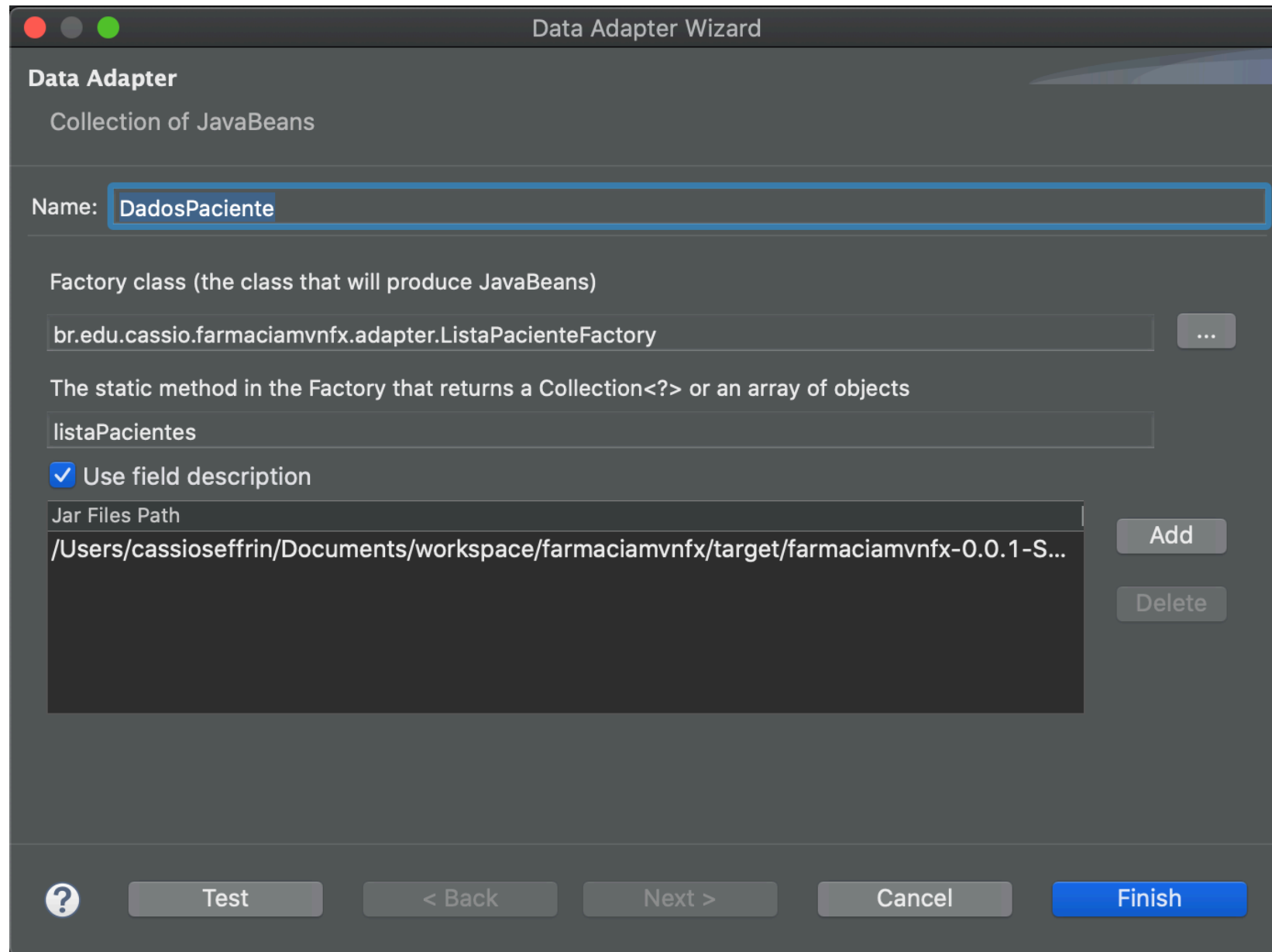
Jasper

**Adicionando projeto ao source path,
Project -> Properties -> tab Source -> opção Java Build Path,
clicar o botão Link Source e adicionar a pasta do projeto.
Com isso será possível criar os relatórios diretamente no projeto
do Netbeans ou Eclipse.**



Jasper

Tools -> Options -> Classpath



The screenshot shows the 'Data Adapter Wizard' dialog box. The title bar reads 'Data Adapter Wizard'. The main section is titled 'Data Adapter' with the subtitle 'Collection of JavaBeans'. The 'Name' field is filled with 'DadosPaciente'. The 'Factory class (the class that will produce JavaBeans)' field contains 'br.edu.cassio.farmaciamvnfx.adapter.ListaPacienteFactory'. The 'The static method in the Factory that returns a Collection<?> or an array of objects' field contains 'listaPacientes'. The 'Use field description' checkbox is checked. The 'Jar Files Path' section shows a list of paths, with the first one being '/Users/cassioseffrin/Documents/workspace/farmaciamvnfx/target/farmaciamvnfx-0.0.1-S...'. There are 'Add' and 'Delete' buttons next to the list. At the bottom, there are buttons for '?', 'Test', '< Back', 'Next >', 'Cancel', and 'Finish'.

Data Adapter Wizard

Data Adapter
Collection of JavaBeans

Name:

Factory class (the class that will produce JavaBeans)
 ...

The static method in the Factory that returns a Collection<?> or an array of objects

☒ Use field description

Jar Files Path

Add
Delete

? Test < Back Next > Cancel Finish

Jasper

Criando um data adapter -> Collection of JavaBeans

The screenshot shows the 'Data Adapter Wizard' dialog box. The title bar reads 'Data Adapter Wizard'. The main section is titled 'Data Adapter' with the subtitle 'Collection of JavaBeans'. The 'Name' field contains 'DadosPaciente'. The 'Factory class (the class that will produce JavaBeans)' field contains 'br.edu.cassio.farmaciamvnfx.adapter.ListaPacienteFactory'. The 'The static method in the Factory that returns a Collection<?> or an array of objects' field contains 'listaPacientes'. The 'Use field description' checkbox is checked. The 'Jar Files Path' field contains '/Users/cassioeffrin/Documents/workspace/farmaciamvnfx/target/farmaciamvnfx-0.0.1-S...'. There are 'Add' and 'Delete' buttons next to the 'Jar Files Path' field. At the bottom, there are buttons for '?', 'Test', '< Back', 'Next >', 'Cancel', and 'Finish'.

Data Adapter Wizard

Data Adapter
Collection of JavaBeans

Name:

Factory class (the class that will produce JavaBeans)
 ...

The static method in the Factory that returns a Collection<?> or an array of objects

☒ Use field description

Jar Files Path
 Add Delete

? Test < Back Next > Cancel Finish

Jasper Studio - JRBeanCollectionDataSource

```
private void handleRelatorio(ActionEvent event) throws JRException {  
    DatabaseMySQL db = new DatabaseMySQL();  
    Connection conexao = db.conectar();  
    MedicoDao pdao = new MedicoDao();  
    pdao.setConnection((Connection) conexao);  
    List<Medico> lst = pdao.listar();  
  
    URL url = getClass().getResource("/relatorios/medico.jasper");  
    JasperReport jasperReport = (JasperReport) JRLoader.loadObject(url);  
  
    JRBeanCollectionDataSource dsMedicos = new JRBeanCollectionDataSource(lst);  
    JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, null, dsMedicos);  
    JasperViewer jasperViewer = new JasperViewer(jasperPrint, false);  
    jasperViewer.setVisible(true);  
}
```


Jasper Studio - Spring 3,4

```
@RequestMapping(value = "/parcelas", method = RequestMethod.GET)  
public ModelAndView parcelas() {  
    ModelAndView modelAndView = new ModelAndView();  
    modelAndView.setViewName("/alunos");  
    return modelAndView;  
}
```

Jasper Studio - Spring 5

```
@RequestMapping(value = "alunos", method = RequestMethod.GET)
@ResponseBody
public void alunosInad(HttpServletRequest response) throws JRException, IOException {

    InputStream jasperStream = getClass().getResourceAsStream("/alunos.jasper");
    JasperReport jasperReport = (JasperReport) JRLoader.loadObject(jasperStream);
    JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, null, new
JREmptyDataSource());

    response.setContentType("application/x-pdf");
    response.setHeader("Content-disposition", "inline; filename=alunos.pdf");

    final OutputStream outputStream = response.getOutputStream();
    JasperExportManager.exportReportToPdfStream(jasperPrint, outputStream);
}
```

Jasper Studio - Spring 5

1. Criando relatório com DataSet do banco de dados
2. Collection de JavaBeans.
3. Montar o relatório consumindo uma REST API