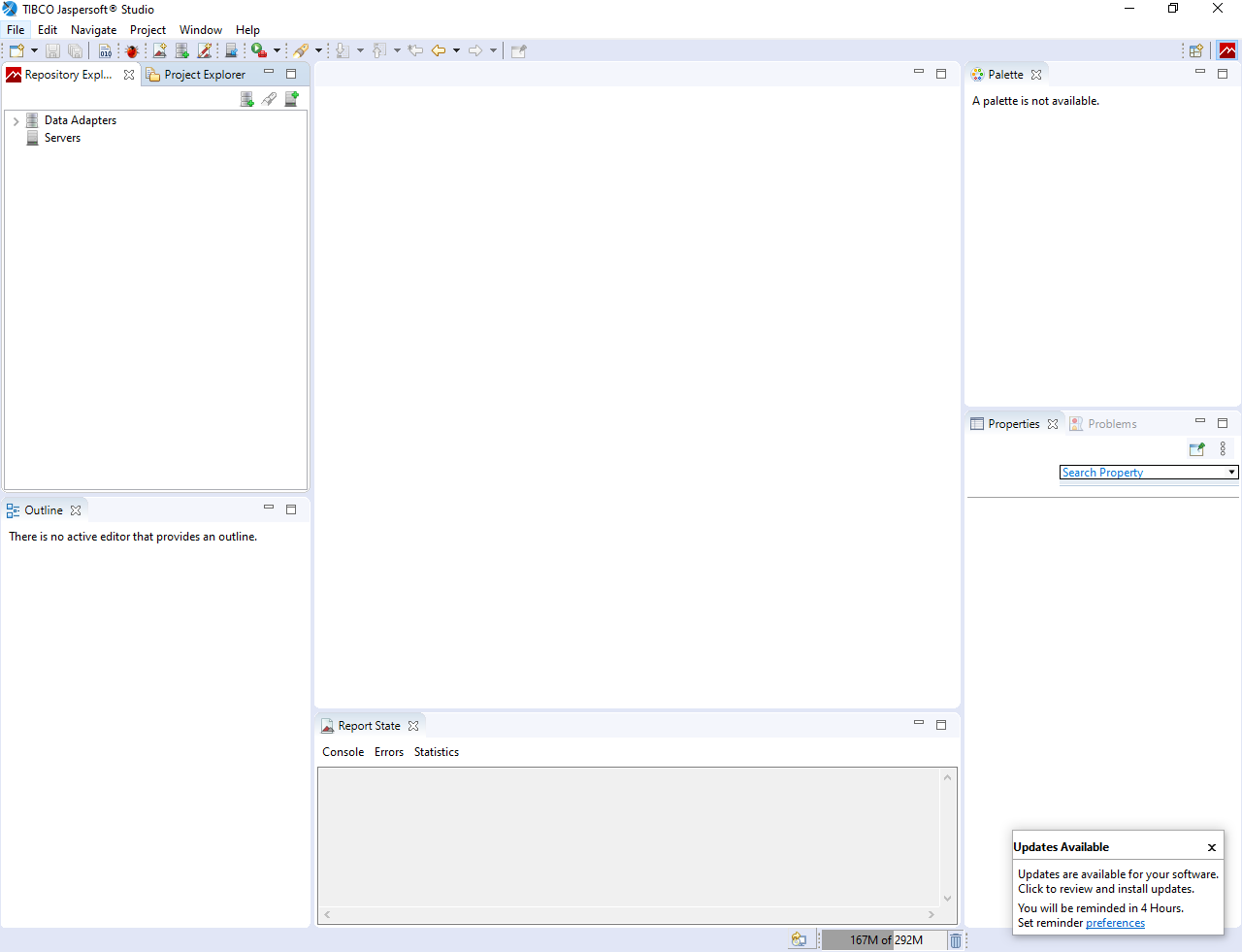
**Relatórios com JasperReports**

Sobre: <https://en.wikipedia.org/wiki/JasperReports>

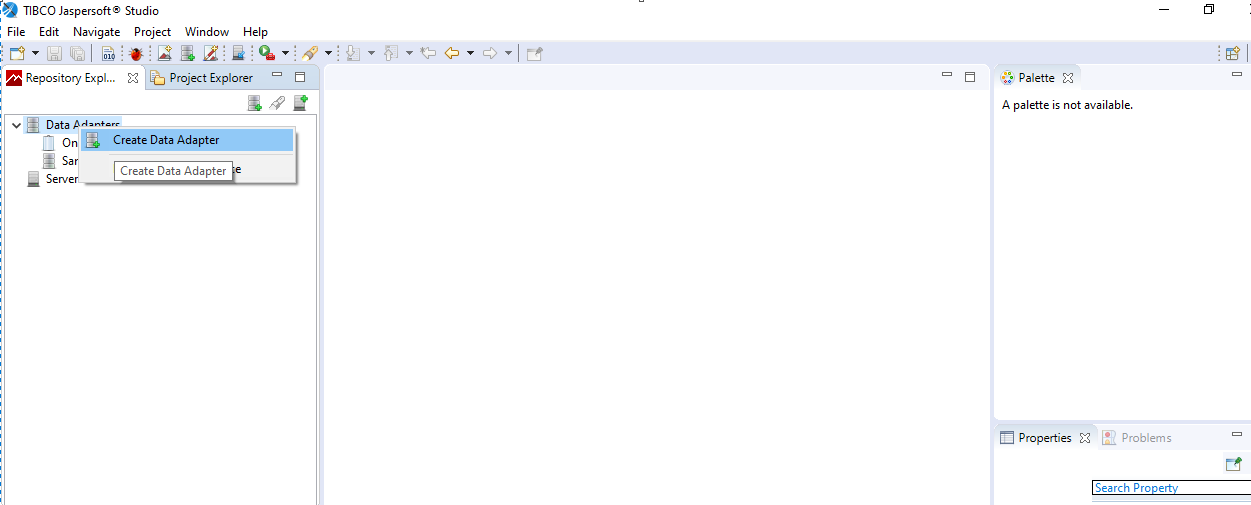
Ferramentas: <https://community.jaspersoft.com/community-download>

JasperSoft Studio Community Edition

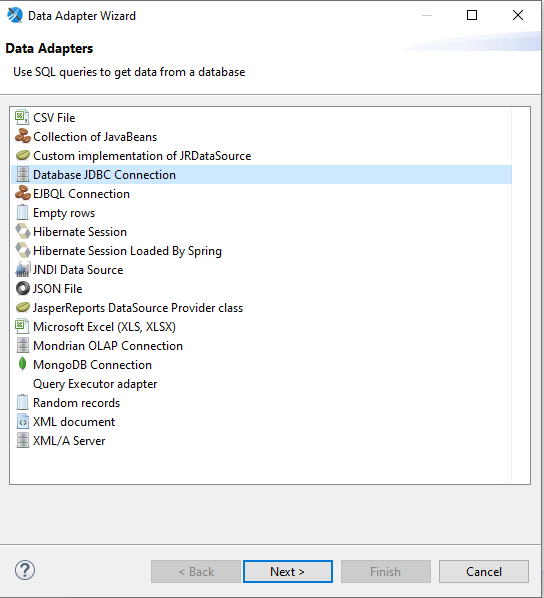
<https://community.jaspersoft.com/project/jaspersoft-studio>

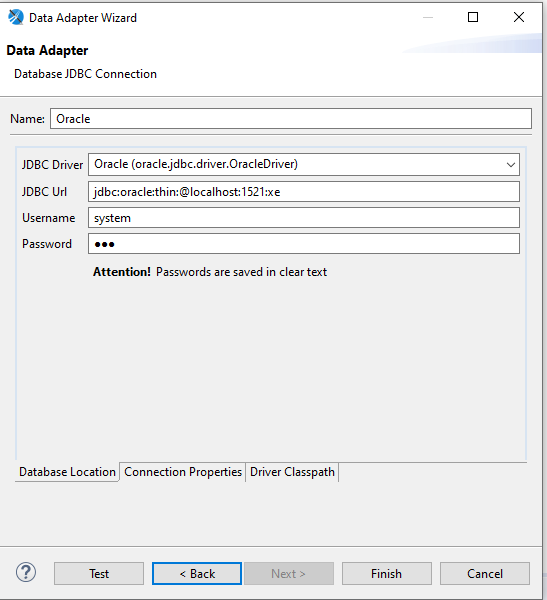


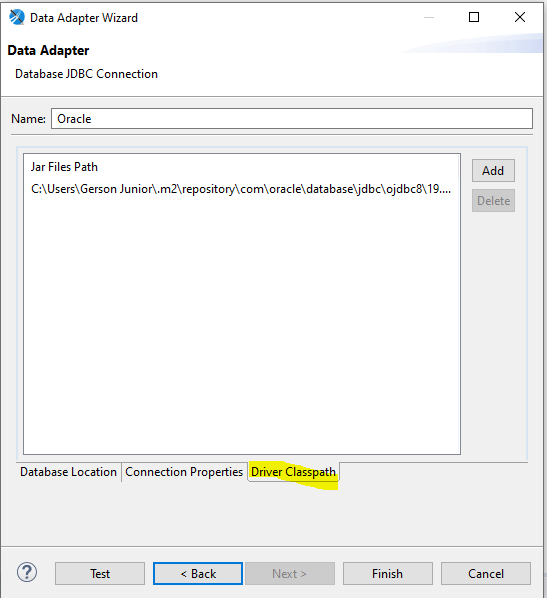
Criando Data Adapters

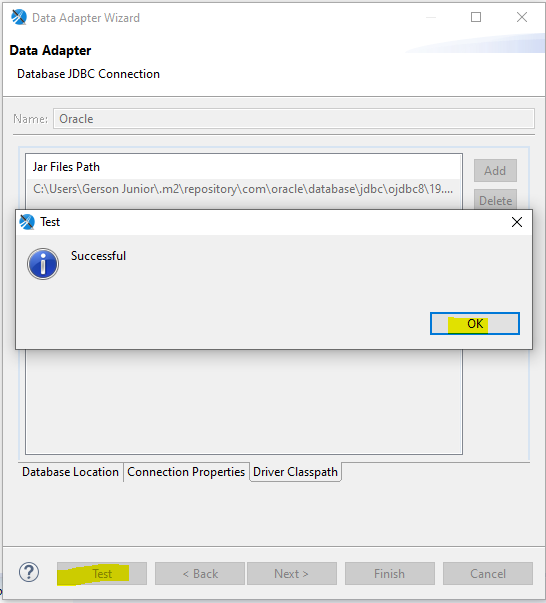


Tipo JDBC

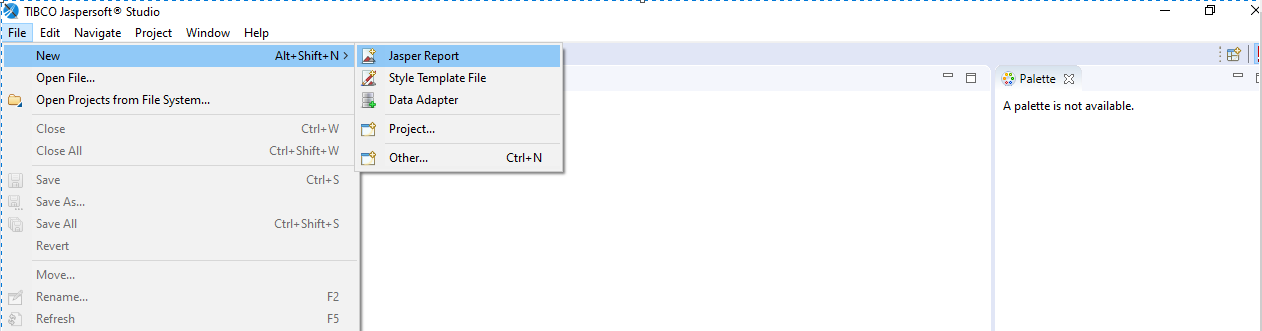


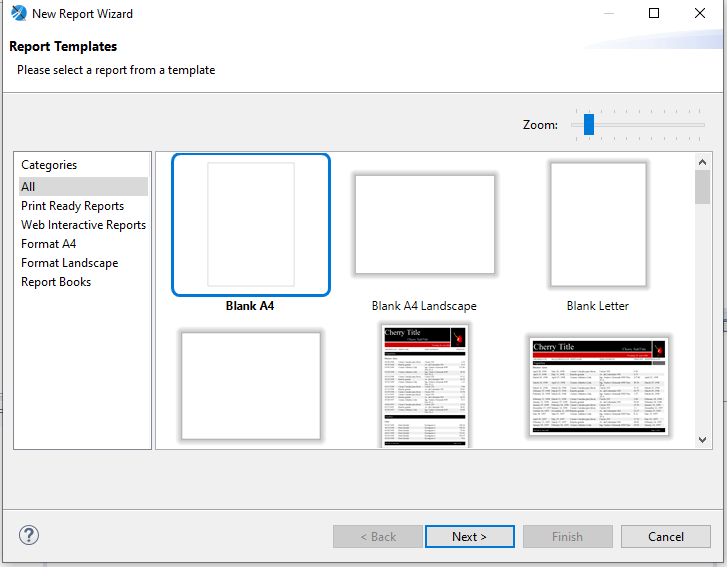


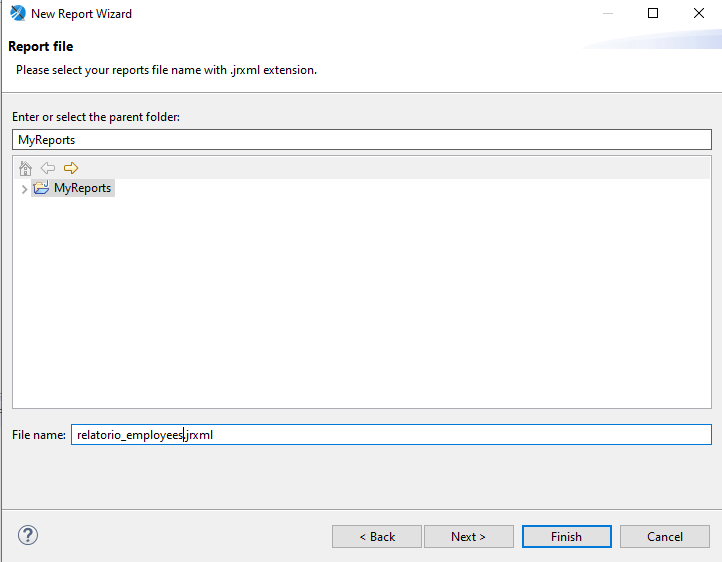


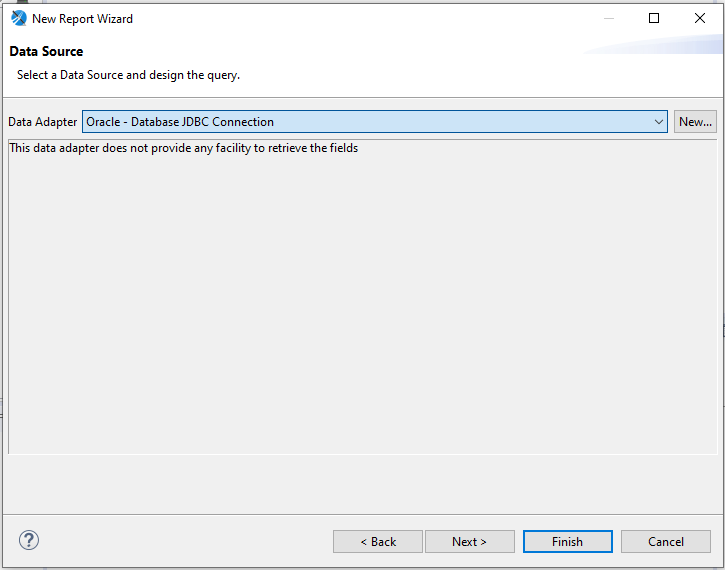


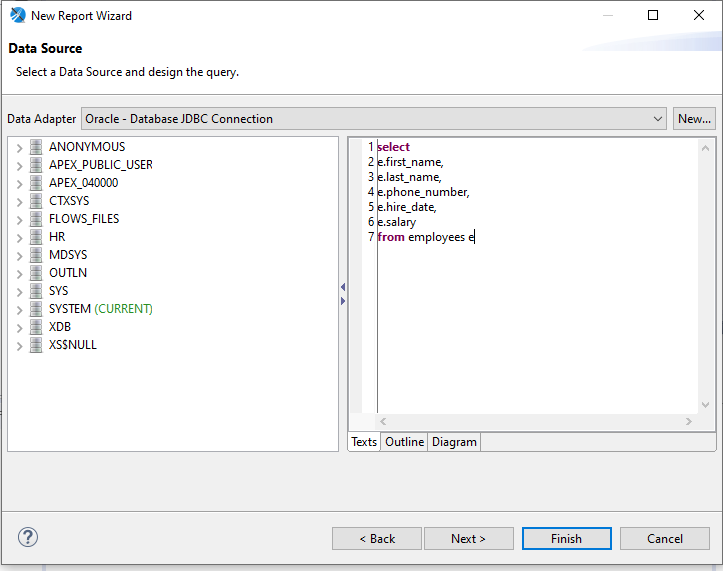
Criando novo projeto

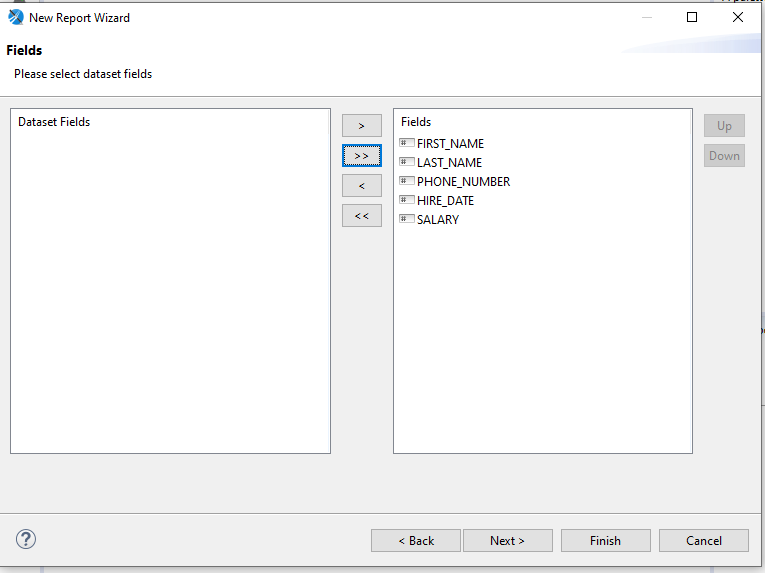


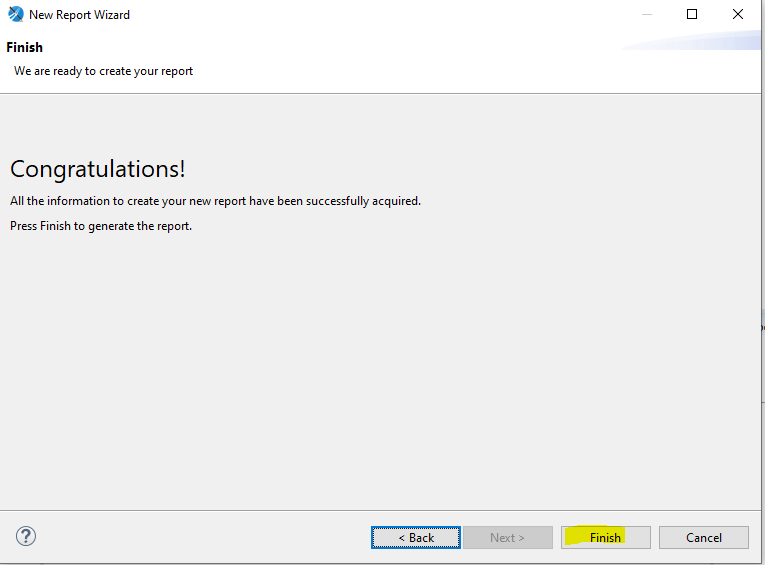


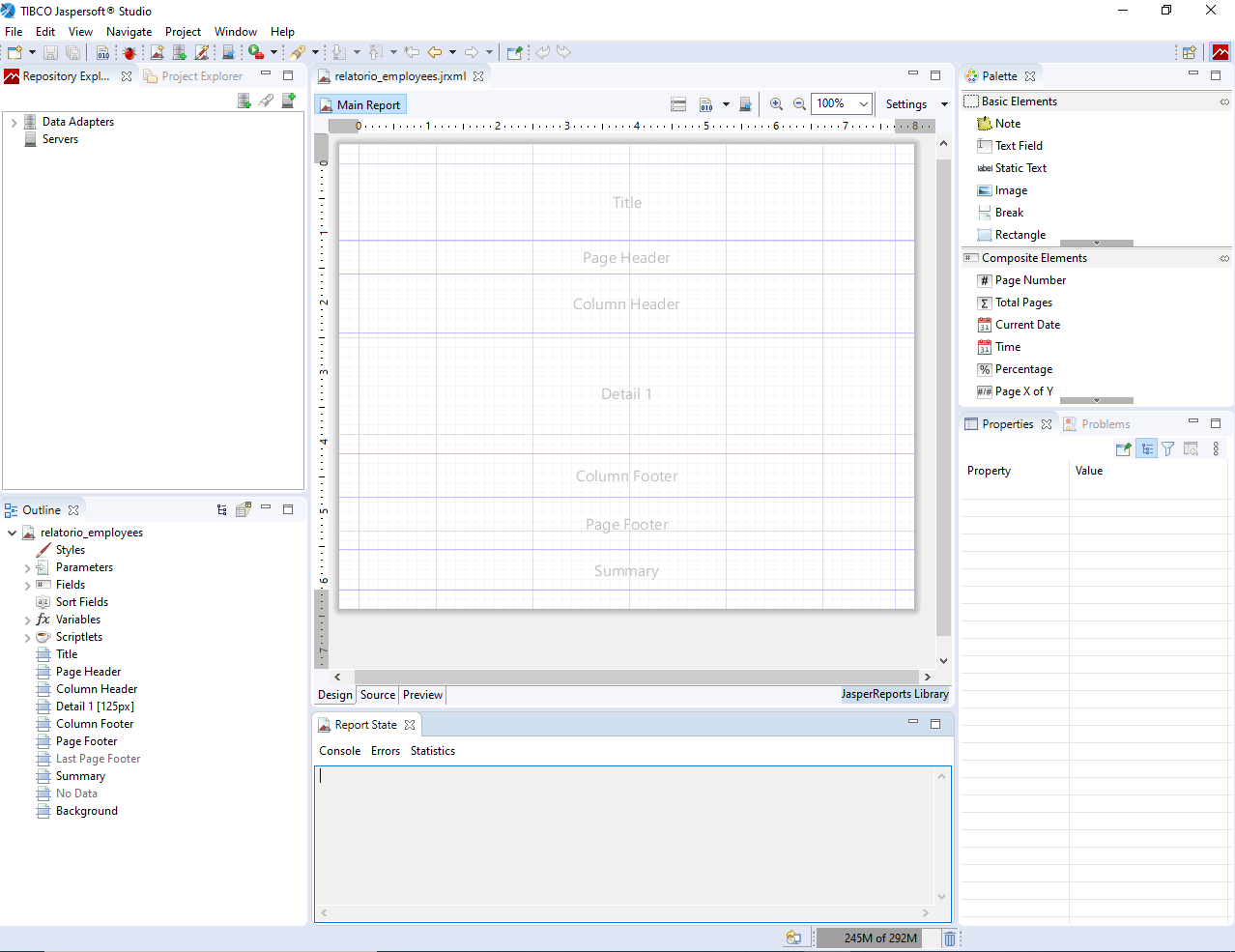






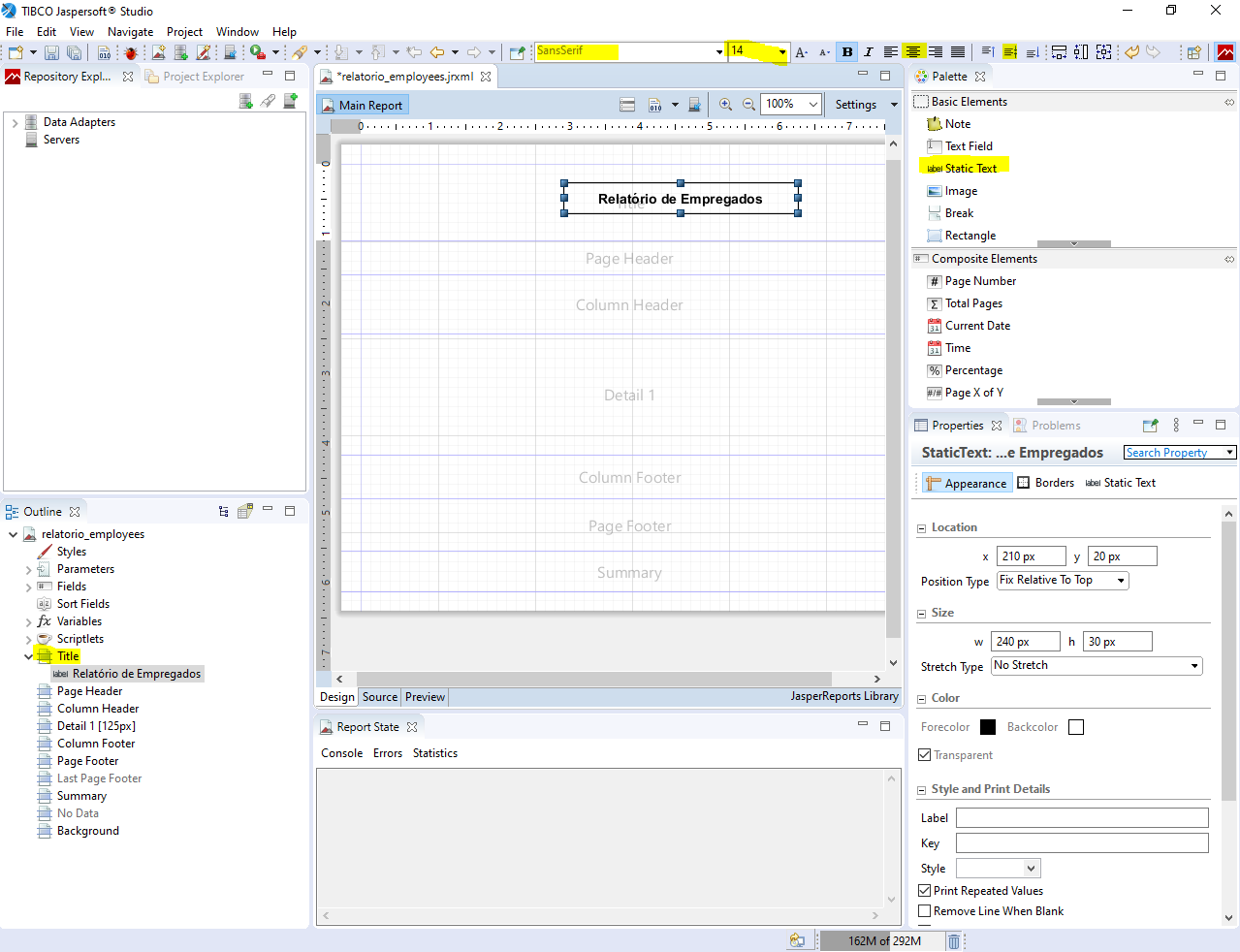




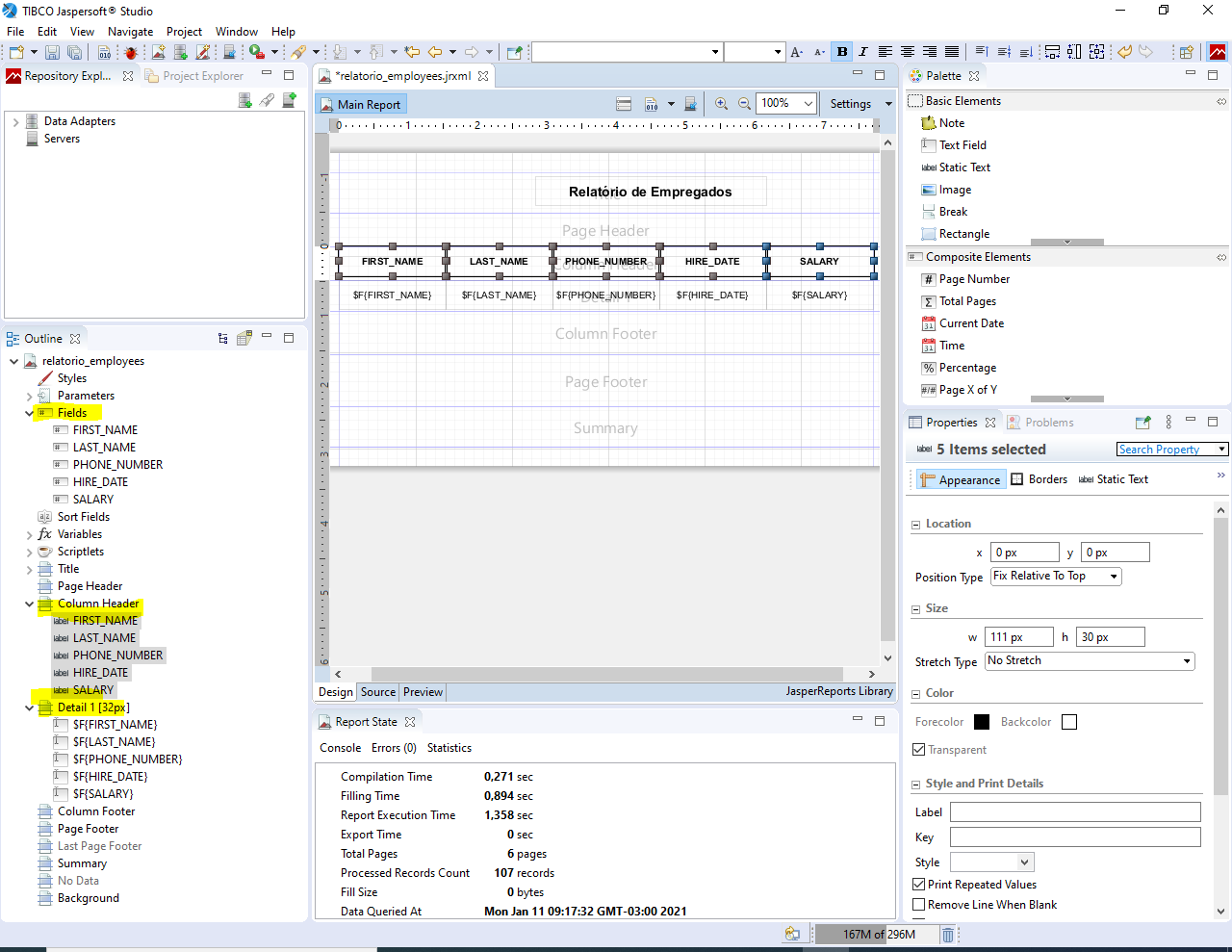


Configurando relatório

bundle Title

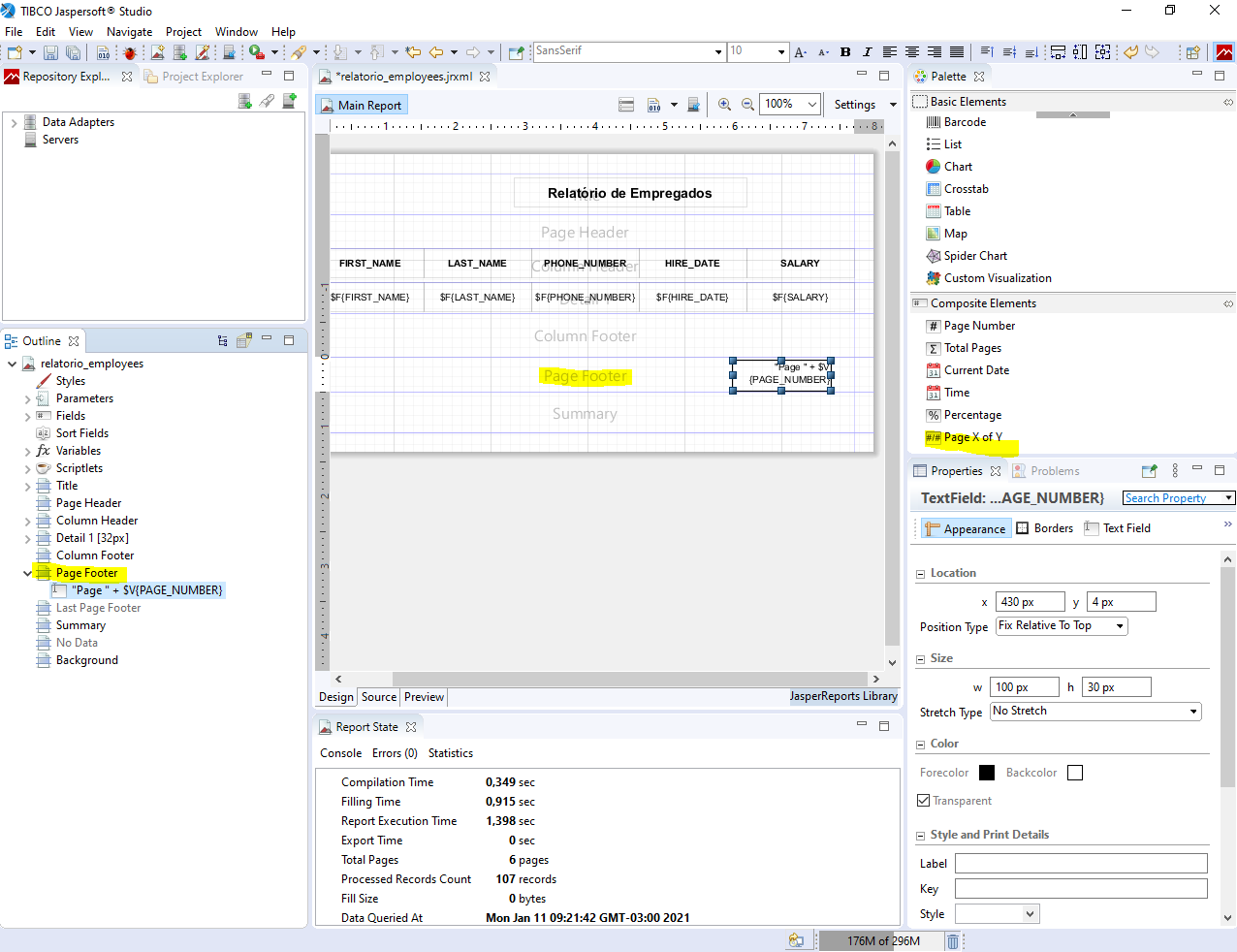


bundle detail, column header



Obs: A principal bundle é a detail, pois ela que itera os valores dos fields.

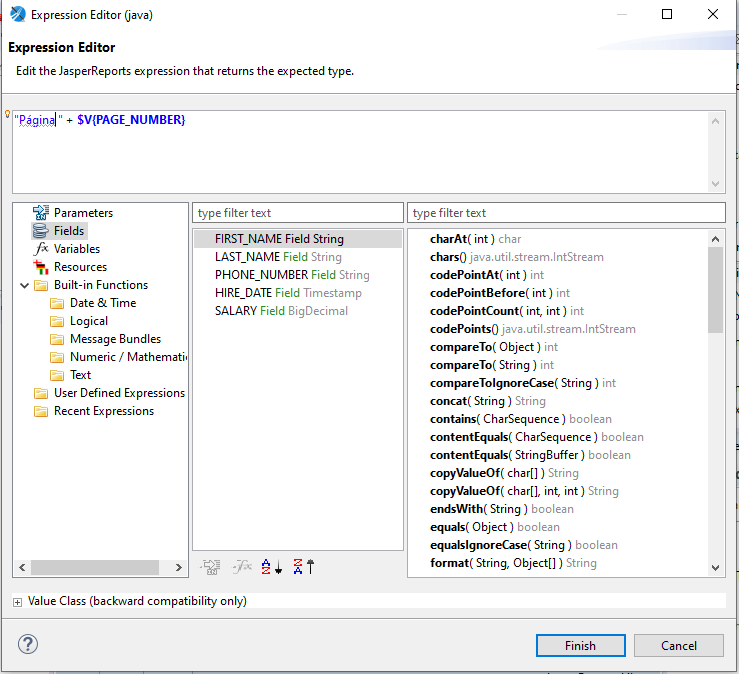
Vamos colocar o número de cada página na bundle page footer.



Editar fields do relatório

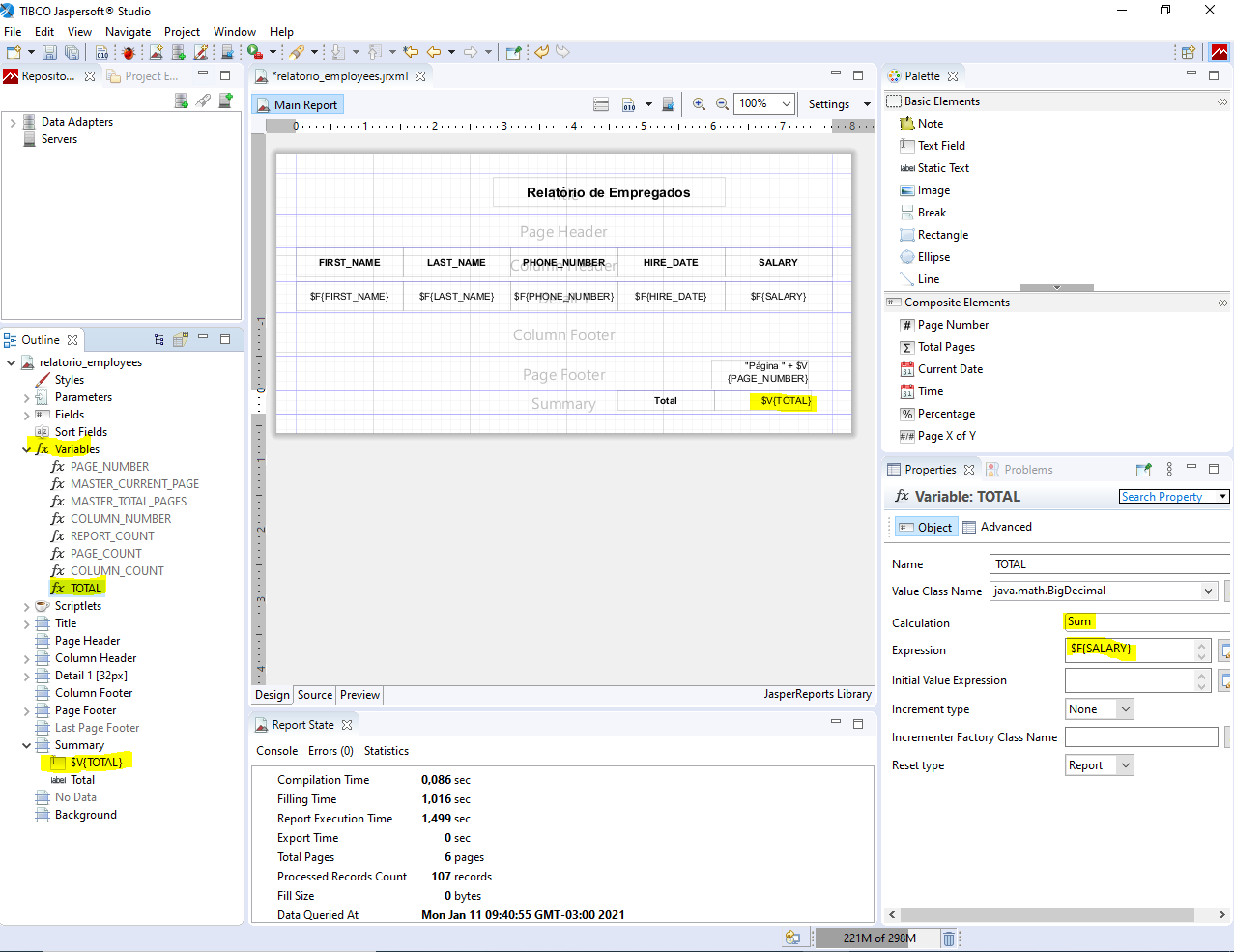
Clique duas vezes no elemento

Exemplo contador de páginas na banda page footer

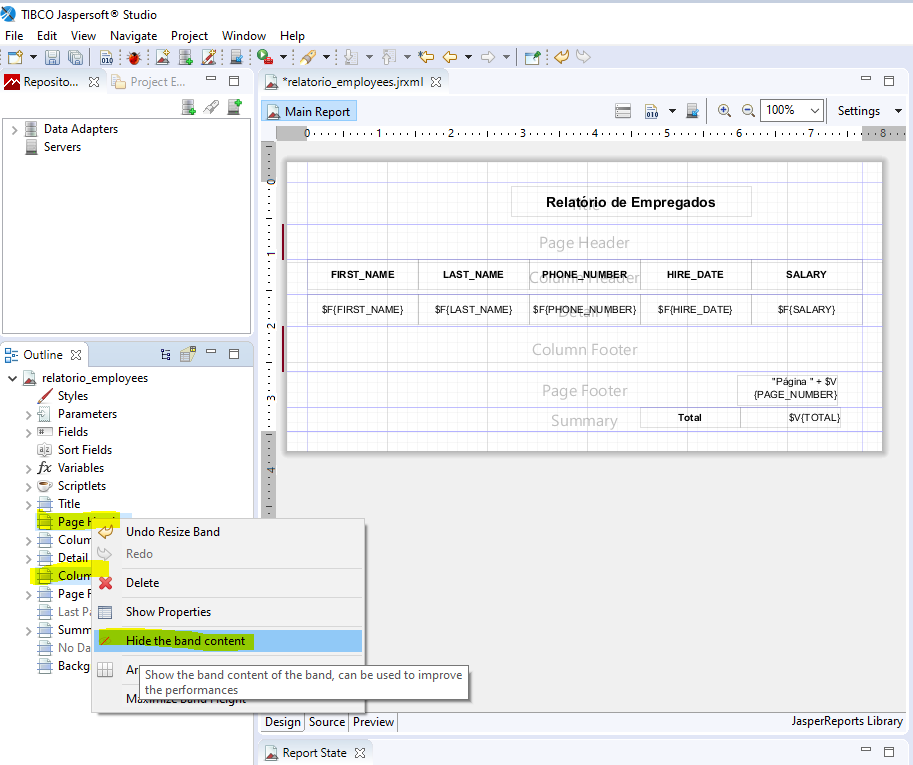


Criando total de salários

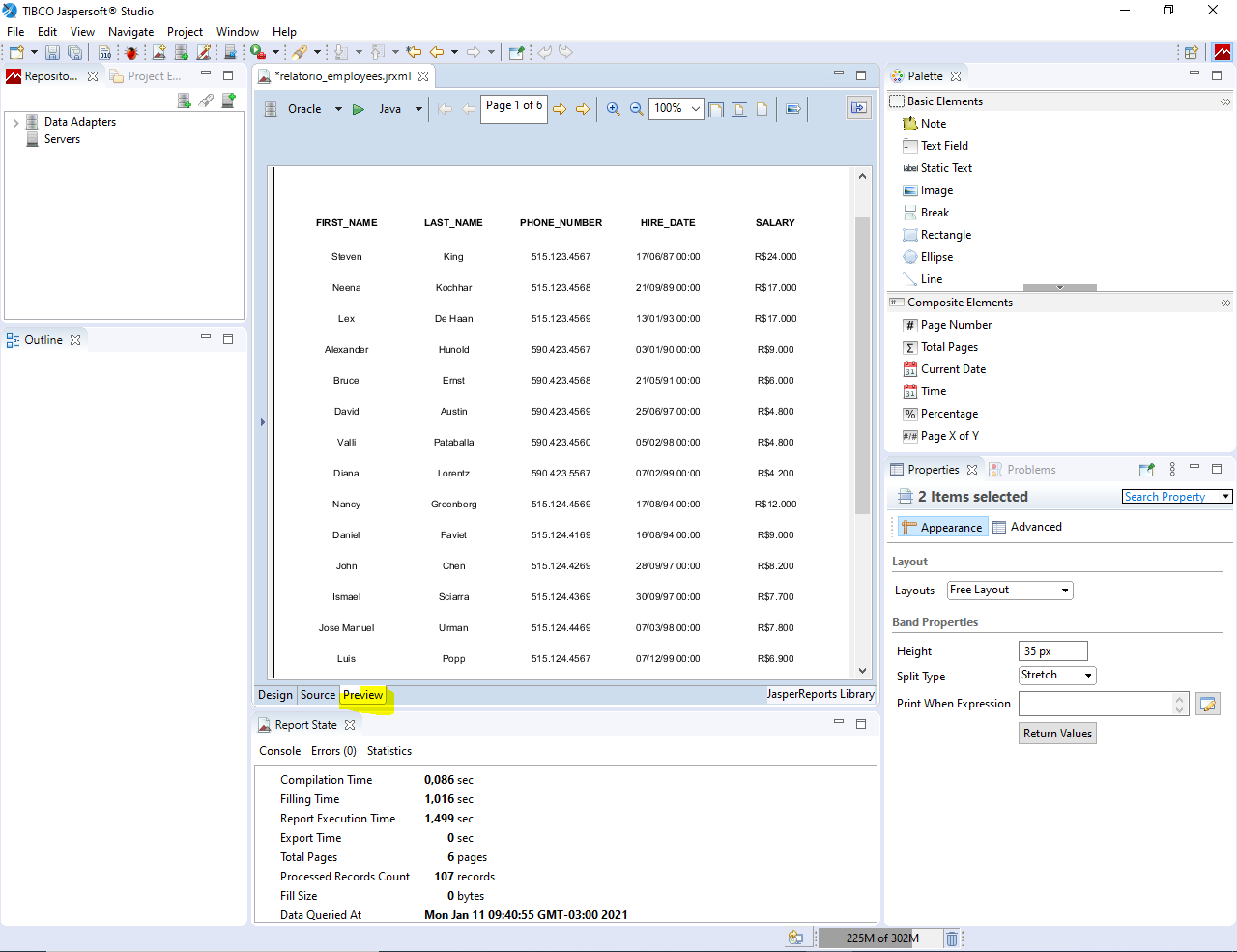
Após criar uma variável adicione o tipo no caso do exemplo BigDecimal e coloque o tipo de cálculo que ele irá fazer “Sum”, na expression referência o field que será somado, após isso adicione a variável total a bundle sumary.



Desativar bundles que não iremos usar

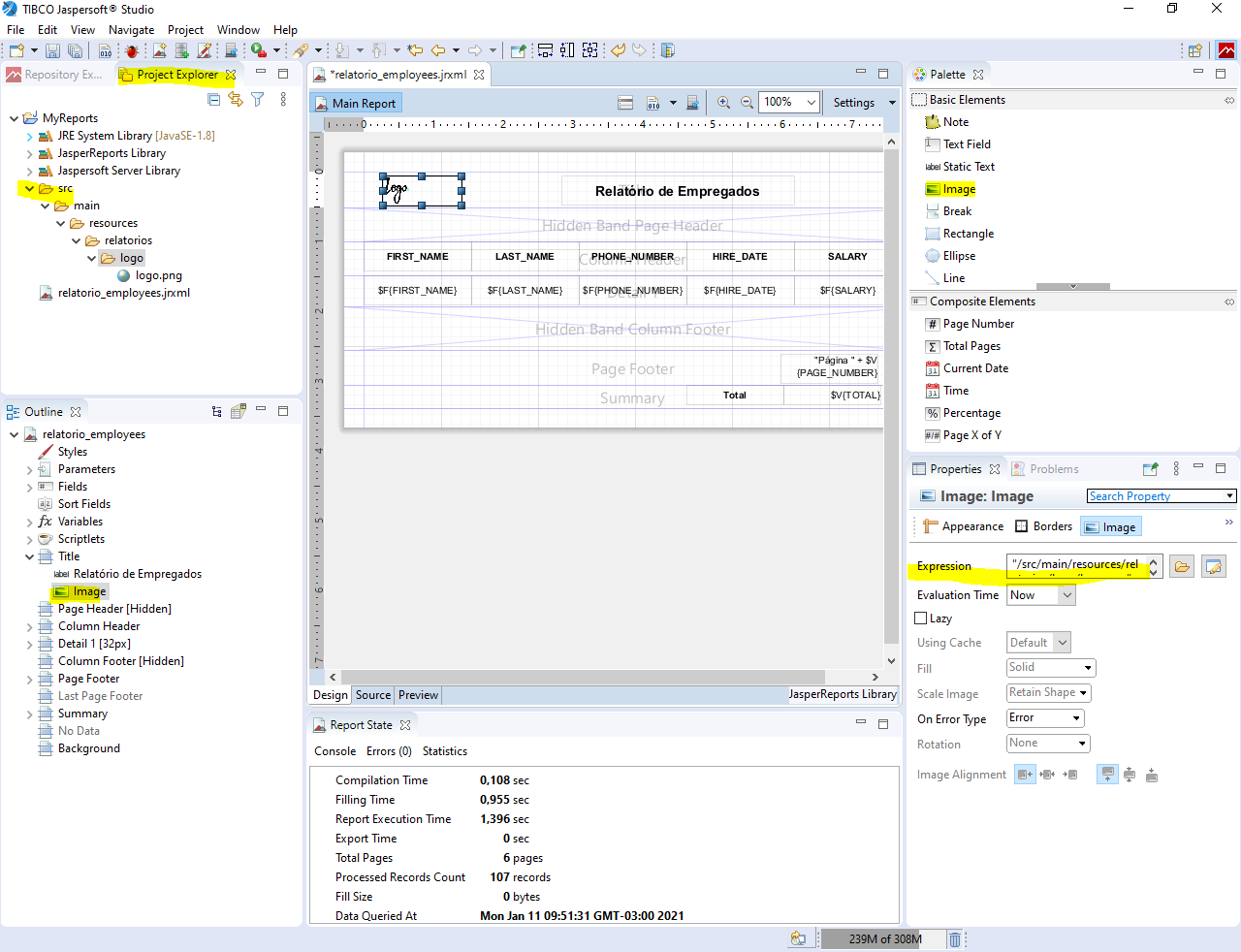


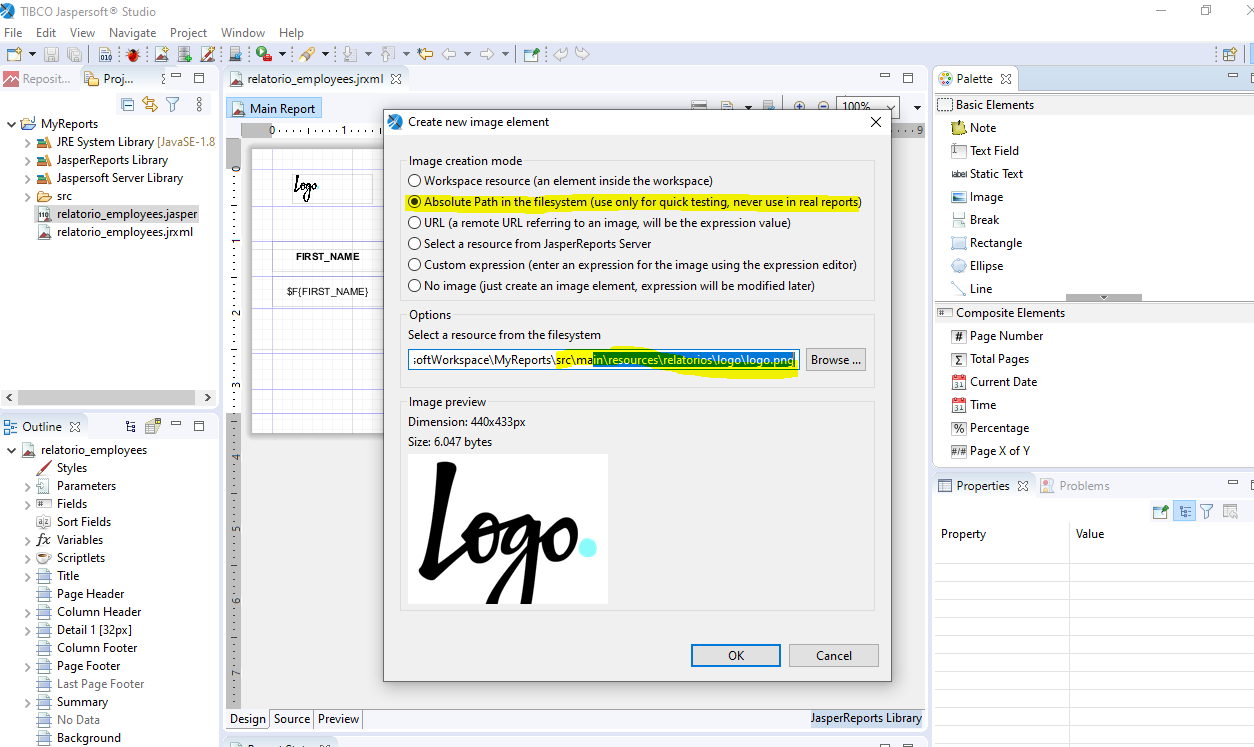
Visualizando relatório



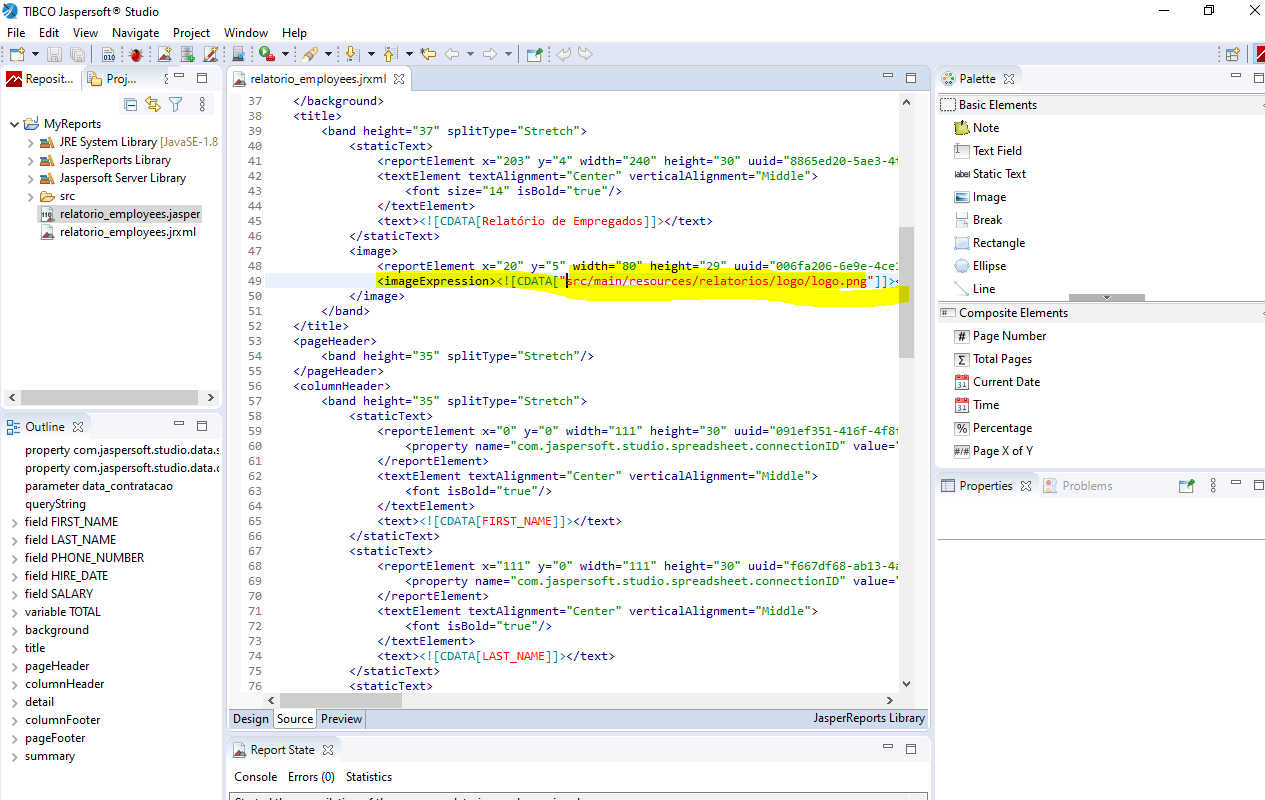
Adicionando imagem logo

Crie a estrutura de pasta no workspace do JasperSoft Studio src/main/resources/relatorios/logo e cole a imagem em png nesse diretório.



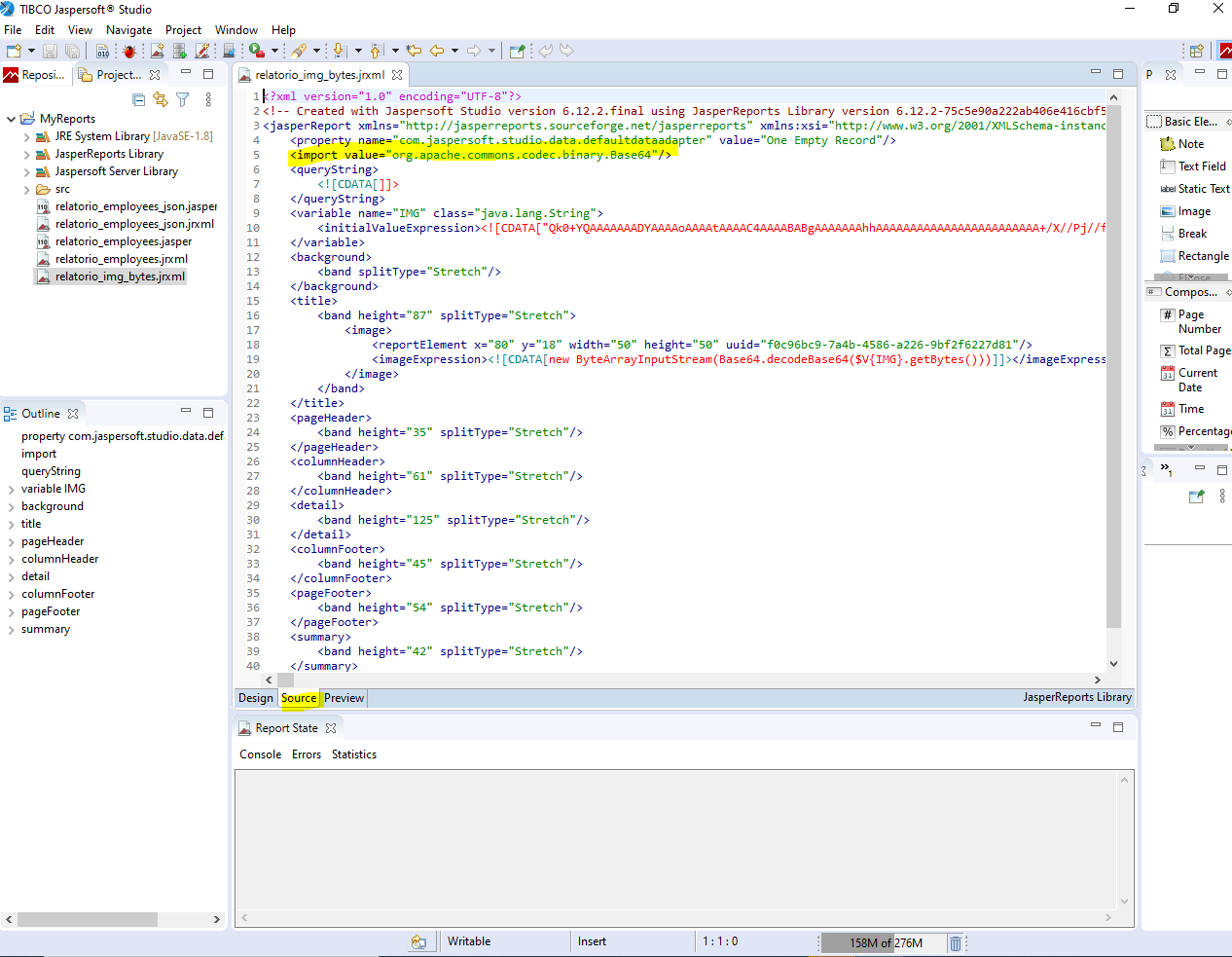


Retire a primeira barra antes do “src”

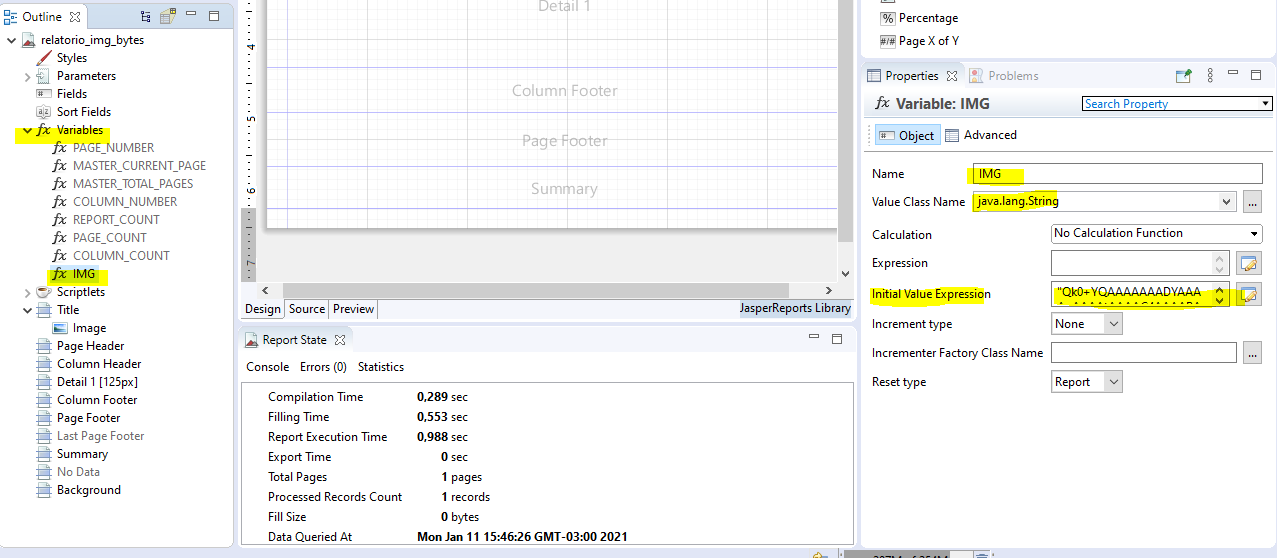


Adicionando imagem/logo em base64

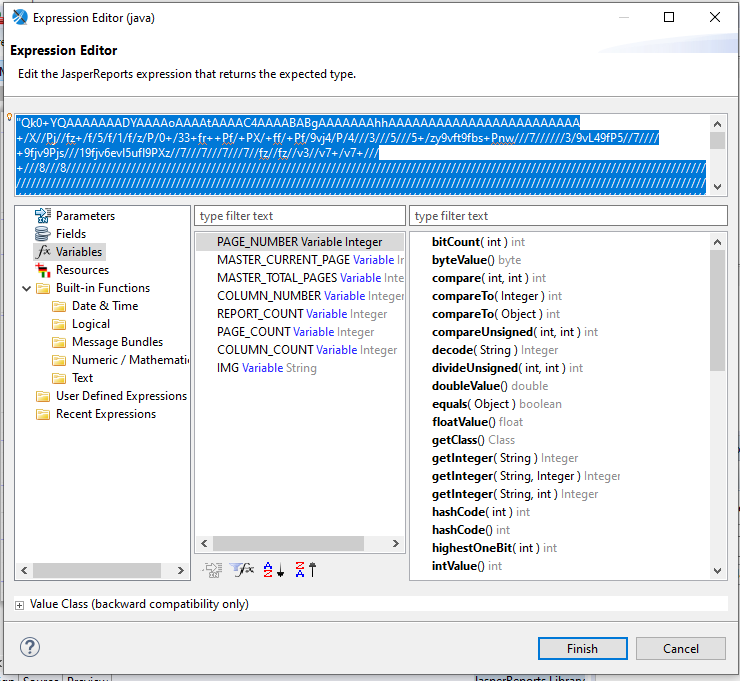
<import value="org.apache.commons.codec.binary.Base64"/>

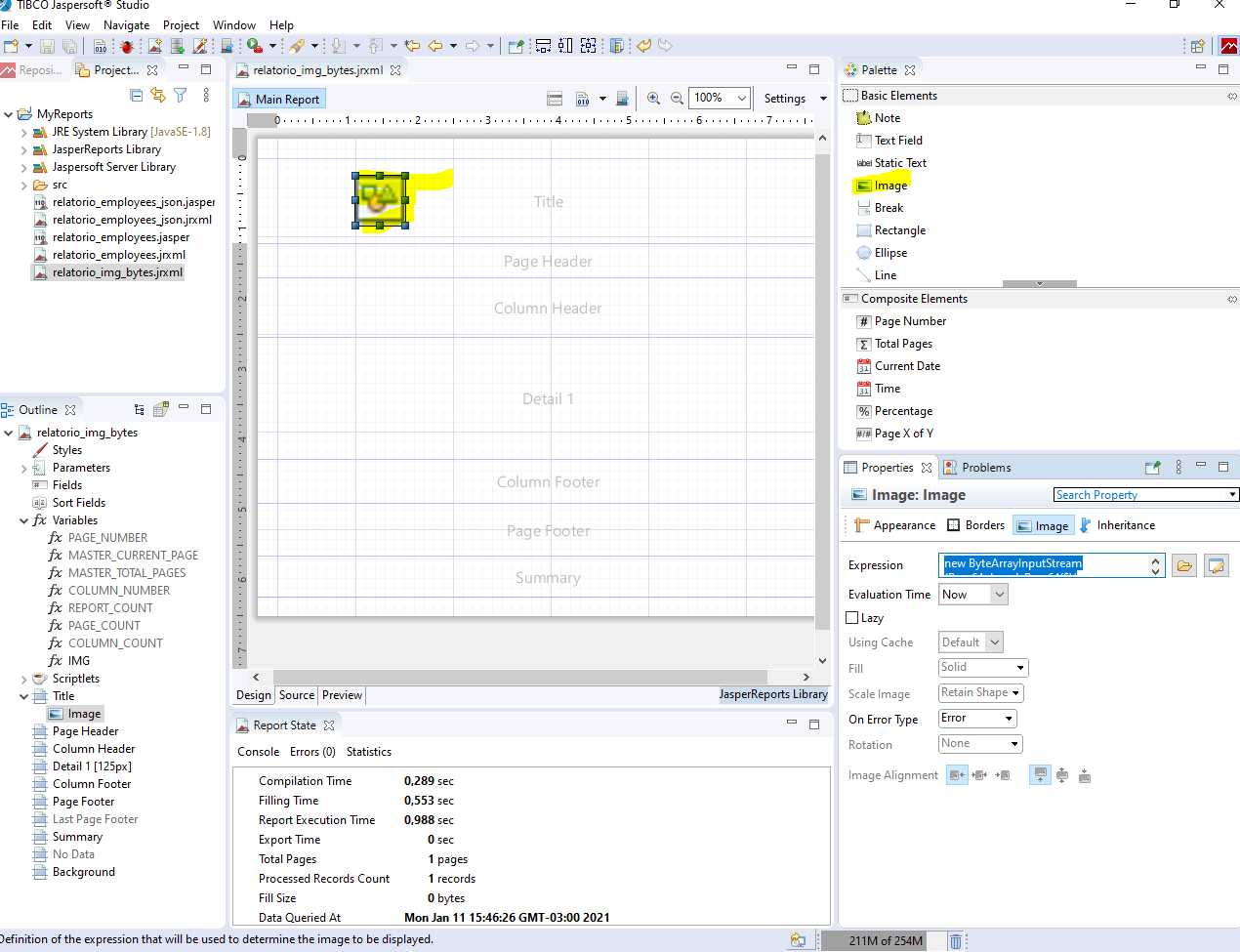


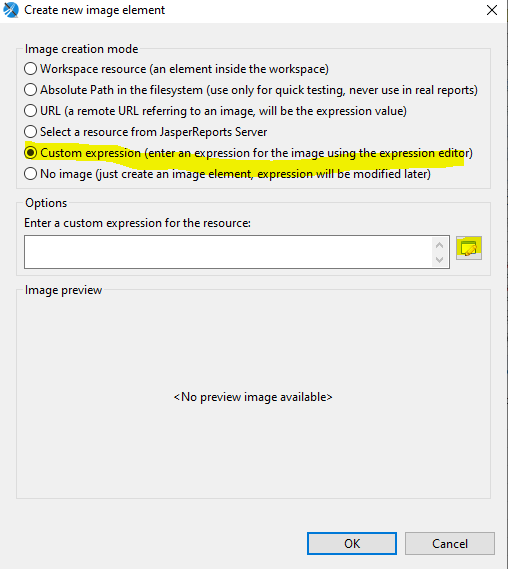
Crie uma variável do tipo String



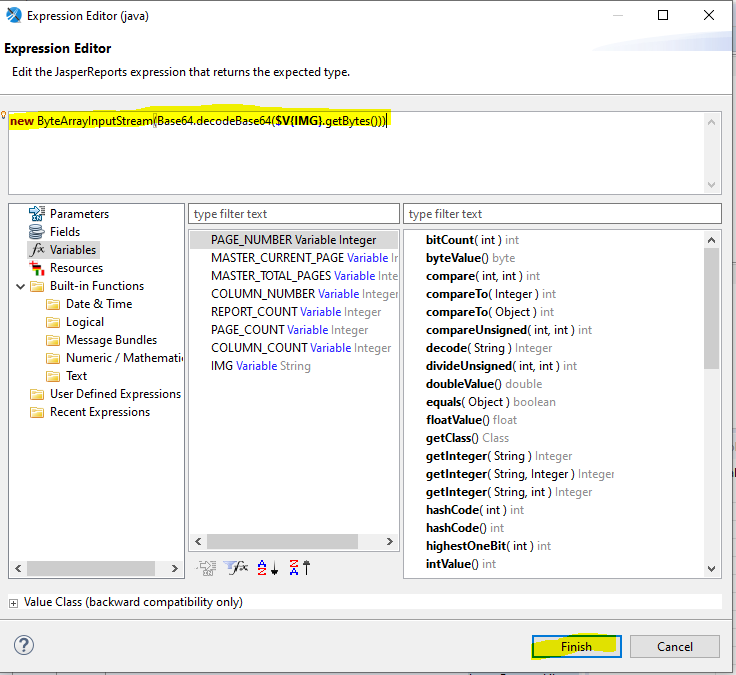
Cole o base64

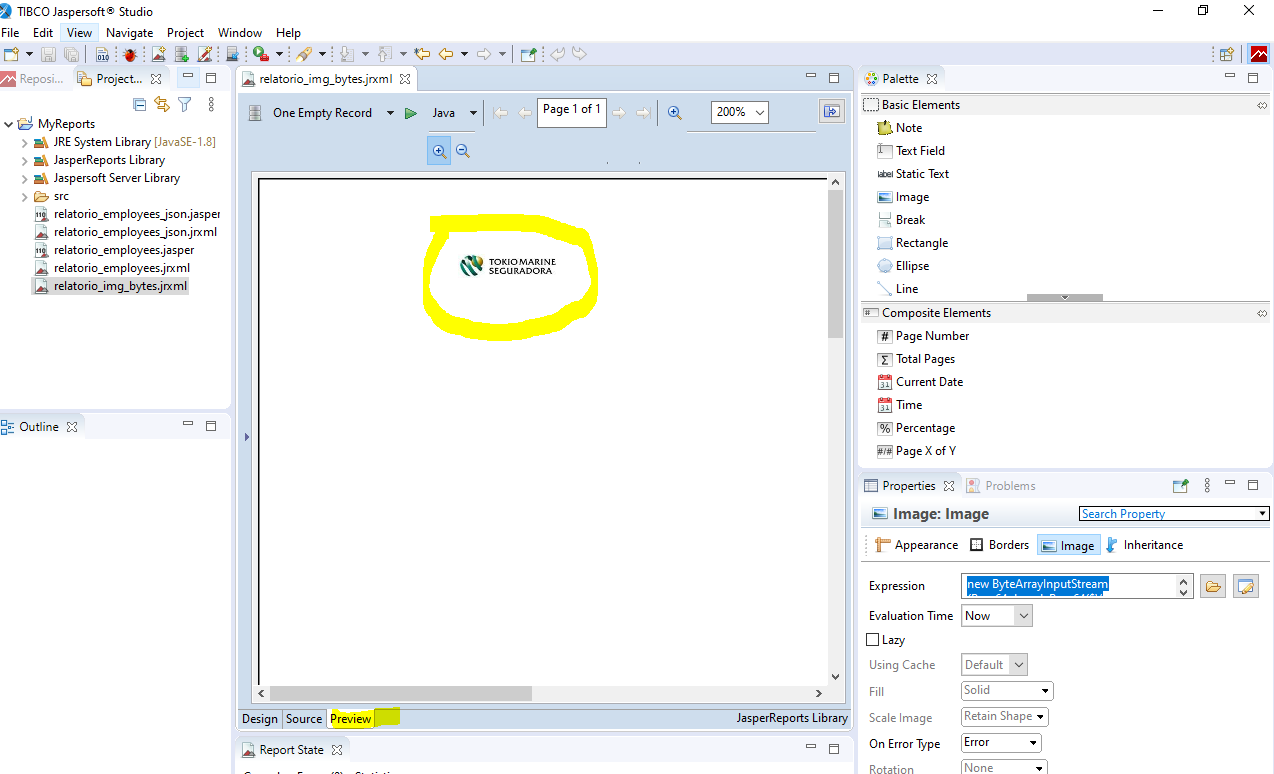




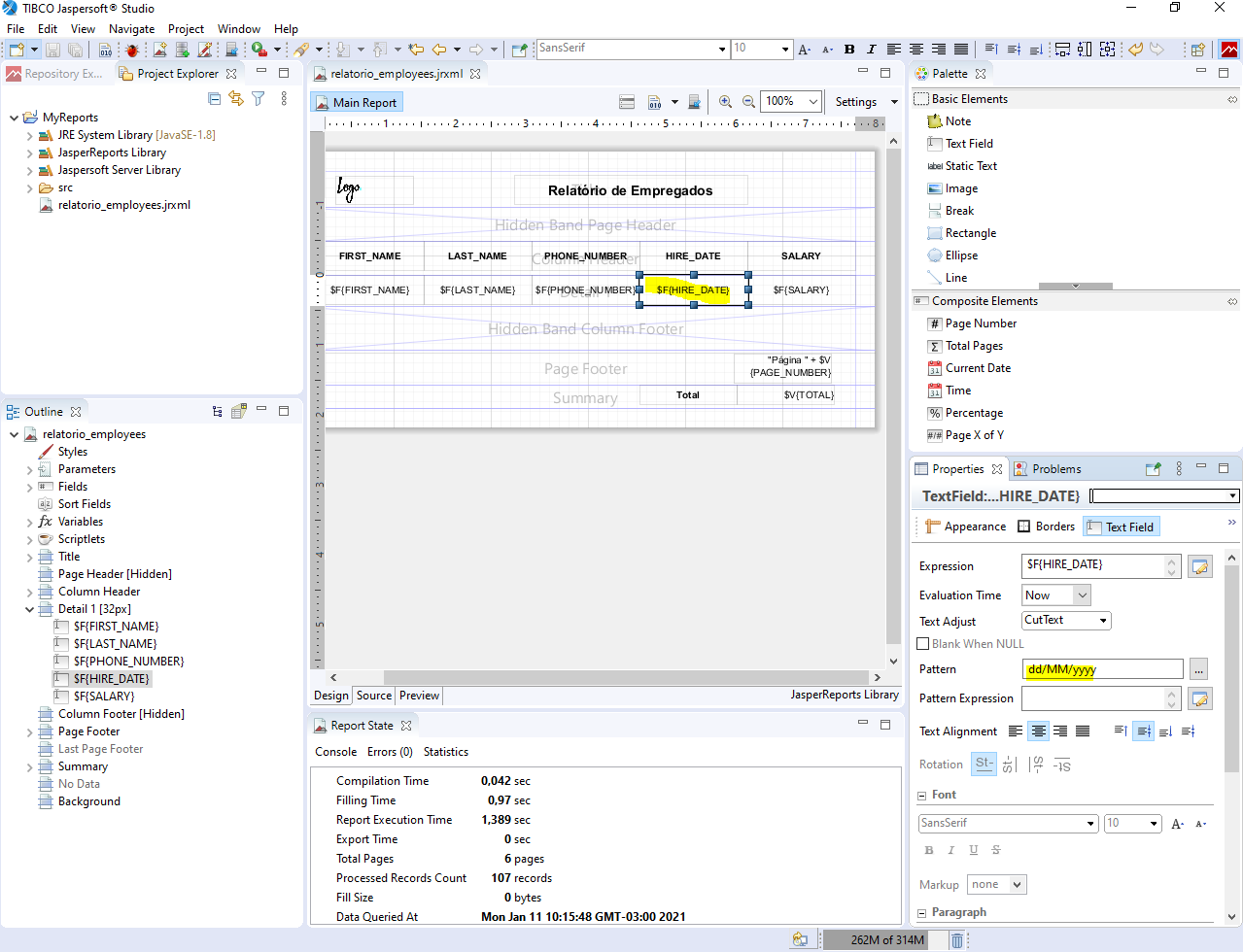


new ByteArrayInputStream(Base64.decodeBase64($V{IMG}.getBytes()))

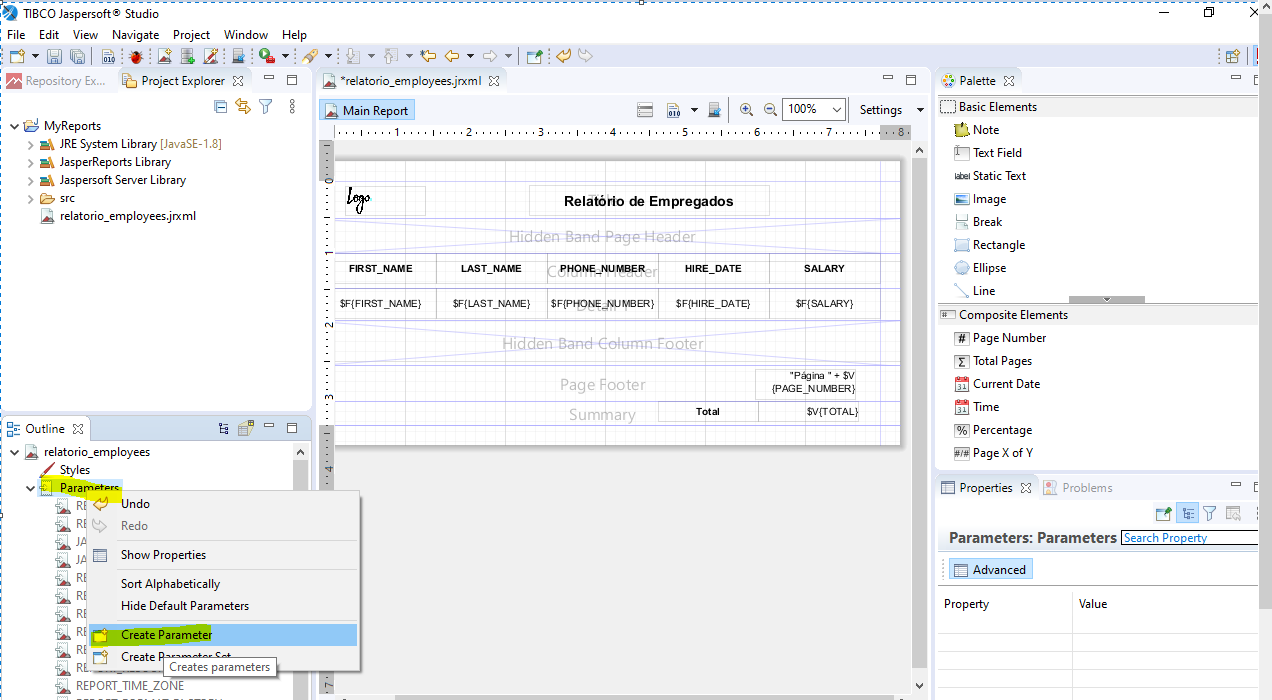


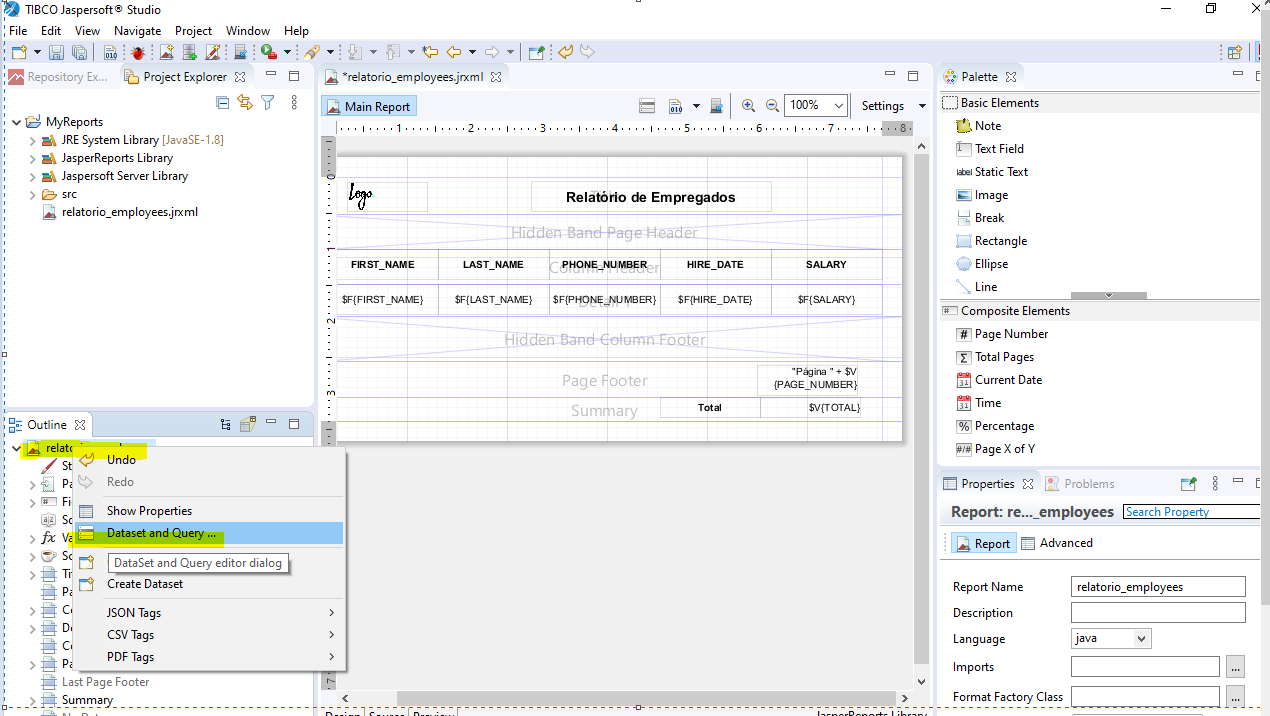


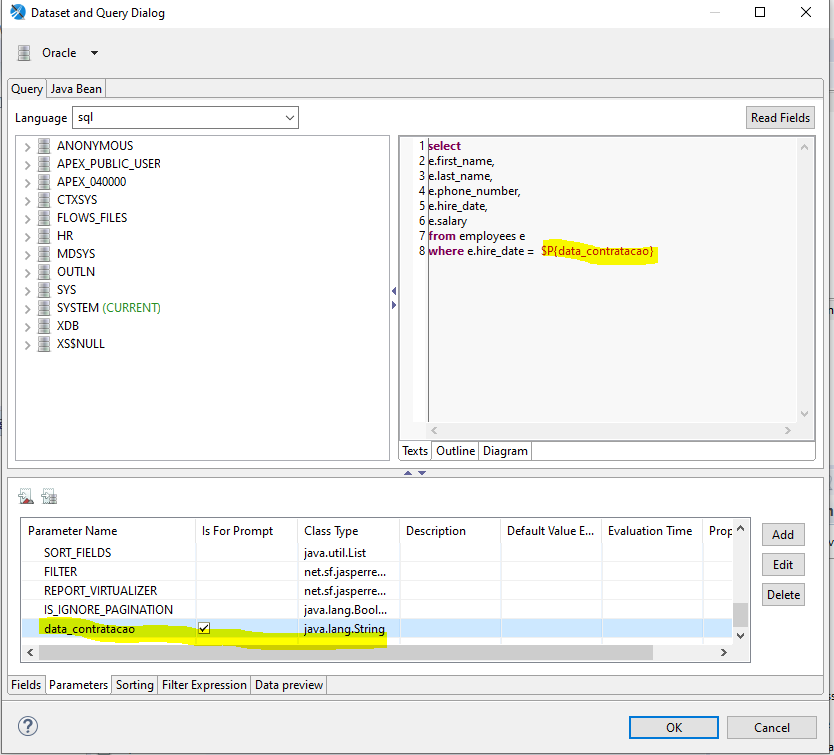
Formatando field de Datas



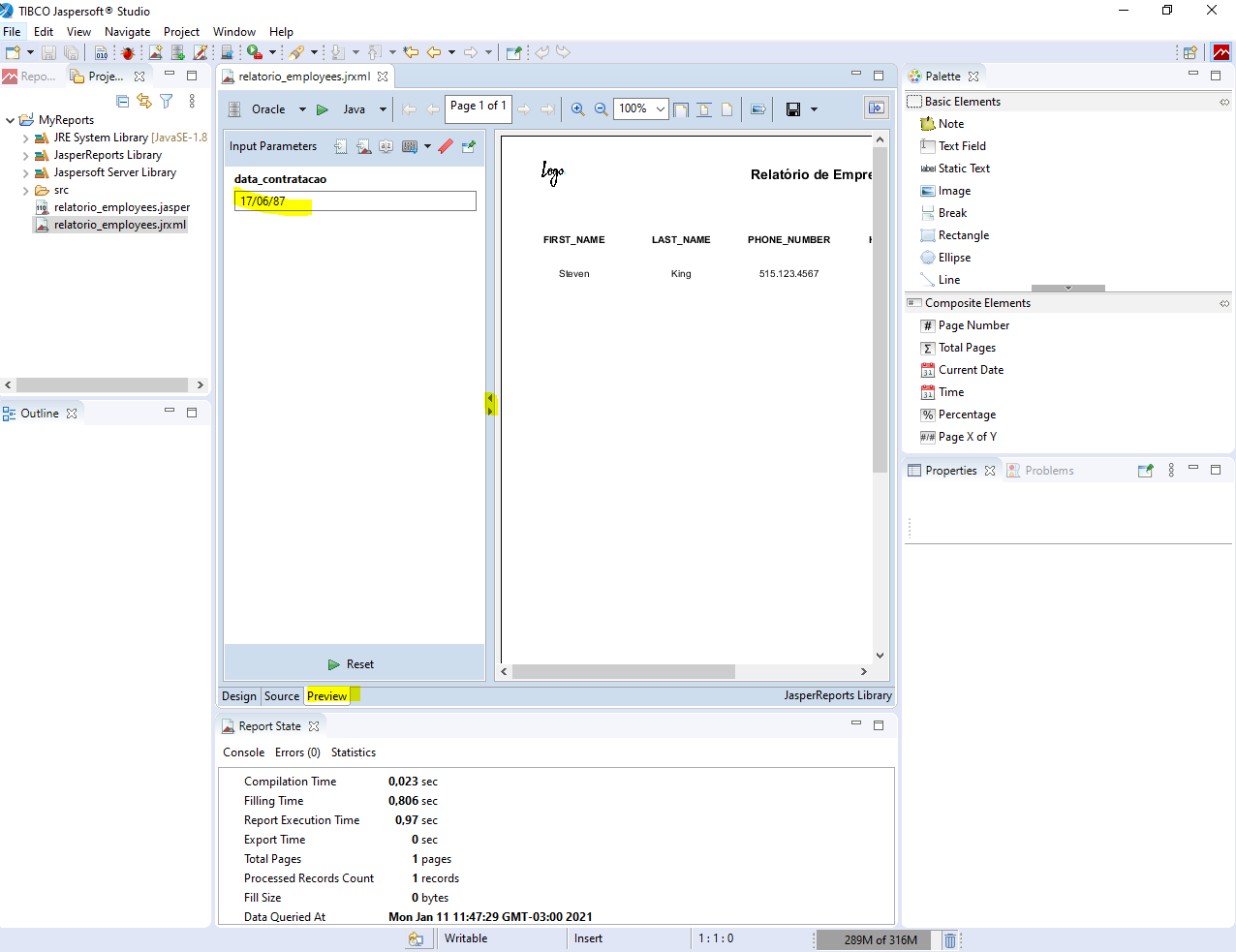
adicionando parâmetros



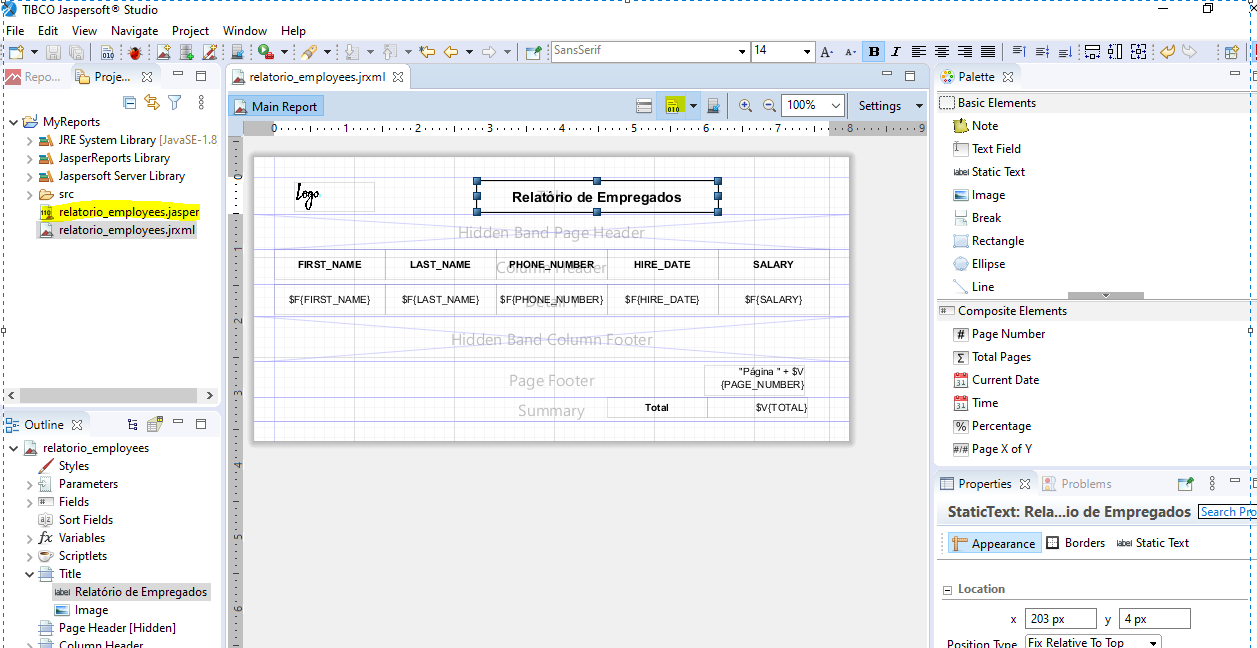




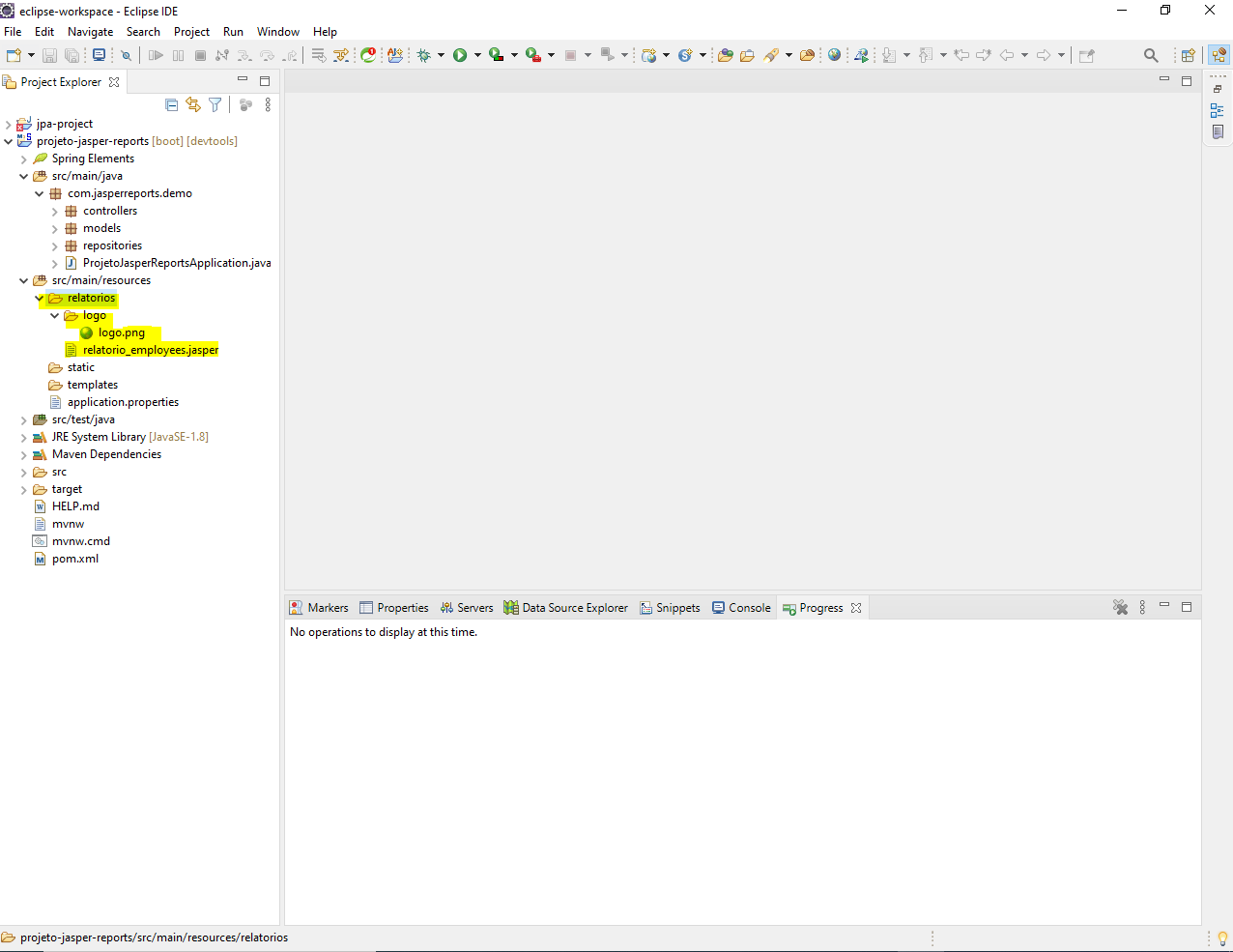
Testando parâmetros



Compilando projeto



Exportar relatórios usando java/spring



<!-- JasperReports -->

<dependency>

<groupId>net.sf.jasperreports</groupId>

<artifactId>jasperreports</artifactId>

<version>6.6.0</version>

</dependency>

<!-- JasperReports Fonts -->

<dependency>

<groupId>net.sf.jasperreports</groupId>

<artifactId>jasperreports-fonts</artifactId>

<version>6.0.0</version>

<scope>compile</scope>

</dependency>

**Controller**

@PostMapping("/db")

public ResponseEntity<byte[]> relatorioDb(@RequestBody ParametrosDTO dto) throws Exception {

byte[] relatorio = relatorioService.exportRelatorioDb(dto);

return ResponseEntity.ok()

.header("Content-Disposition", "attachment; filename=relatorio\_employees\_"+LocalDate.now()+".pdf")

.body(relatorio);

}

**Service**

public byte[] exportRelatorioDb(ParametrosDTO dto) throws SQLException, JRException {

Map<String, Object> parametros = new HashMap<>();

parametros.put("data\_contratacao", dto.getDataContratacao());

parametros.put("REPORT\_LOCALE", new Locale("pt", "BR"));

InputStream inputStream = this.getClass().getResourceAsStream("/relatorios/relatorio\_employees.jasper");

Connection con = this.dataSource.getConnection();

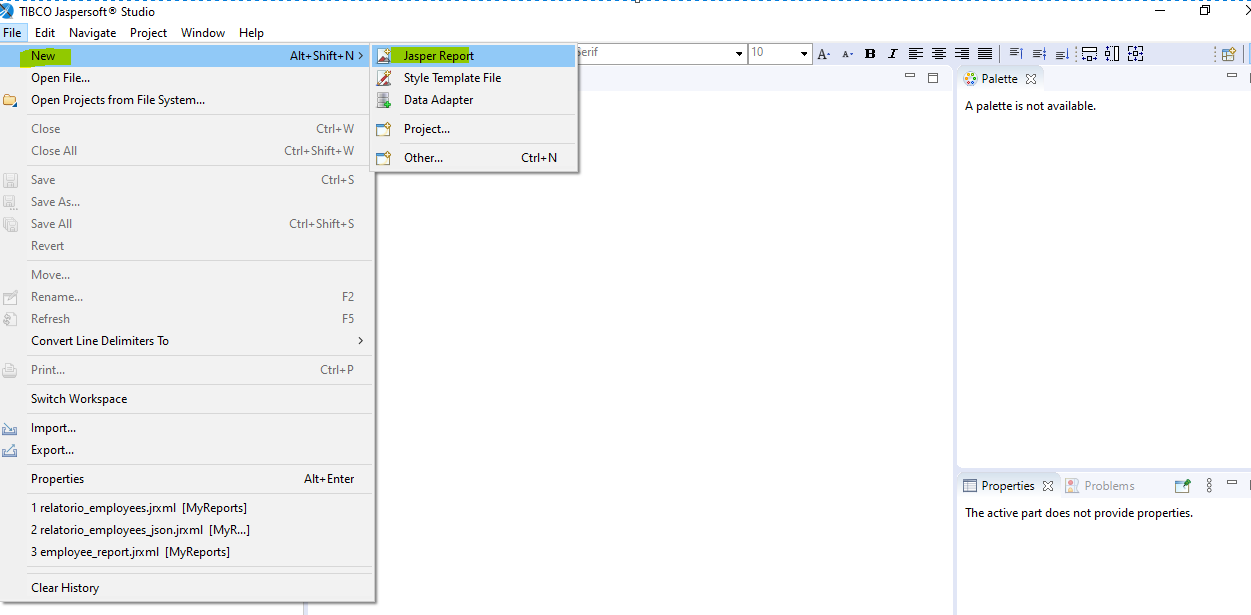
JasperPrint jasperPrint = JasperFillManager.fillReport(inputStream, parametros, con);

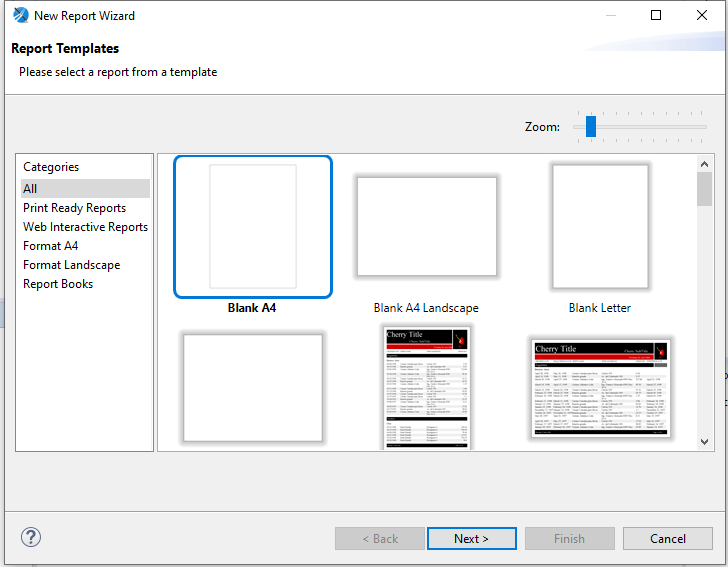
con.close();

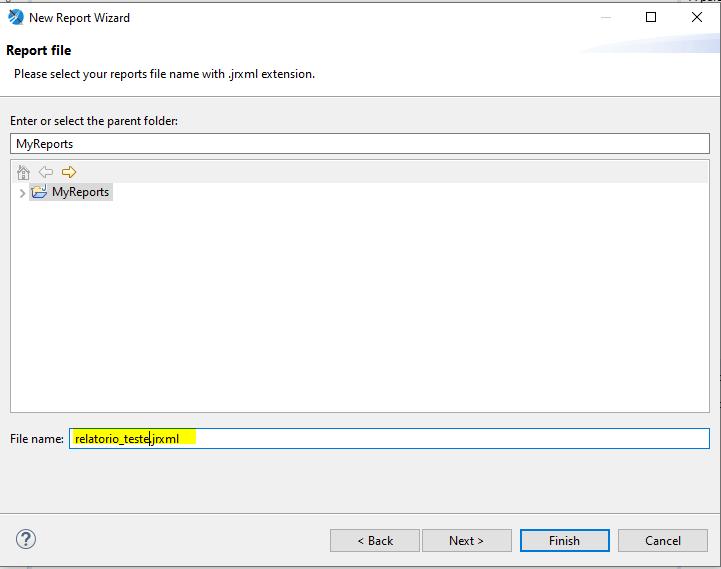
return JasperExportManager.exportReportToPdf(jasperPrint);

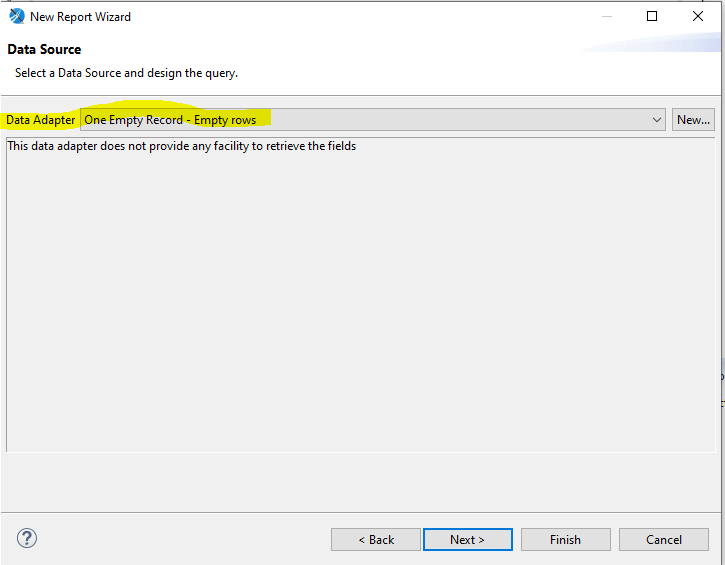
}

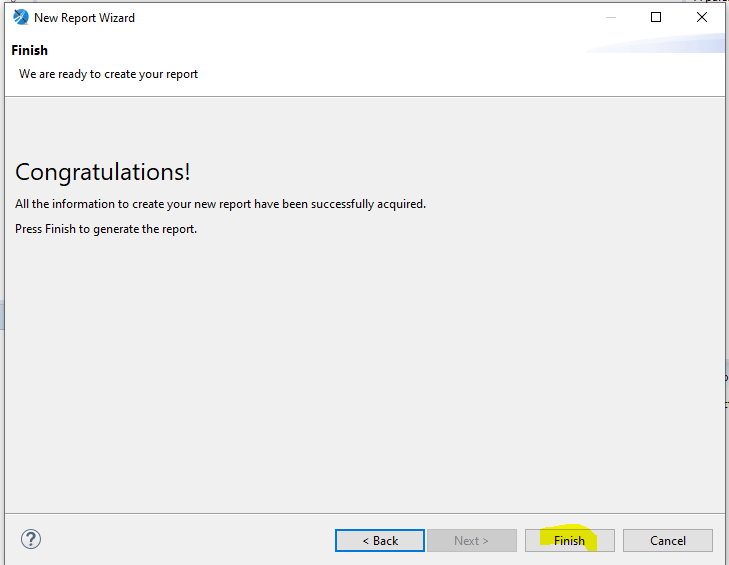
**Relatório com informações vindas de uma requisição rest**

****

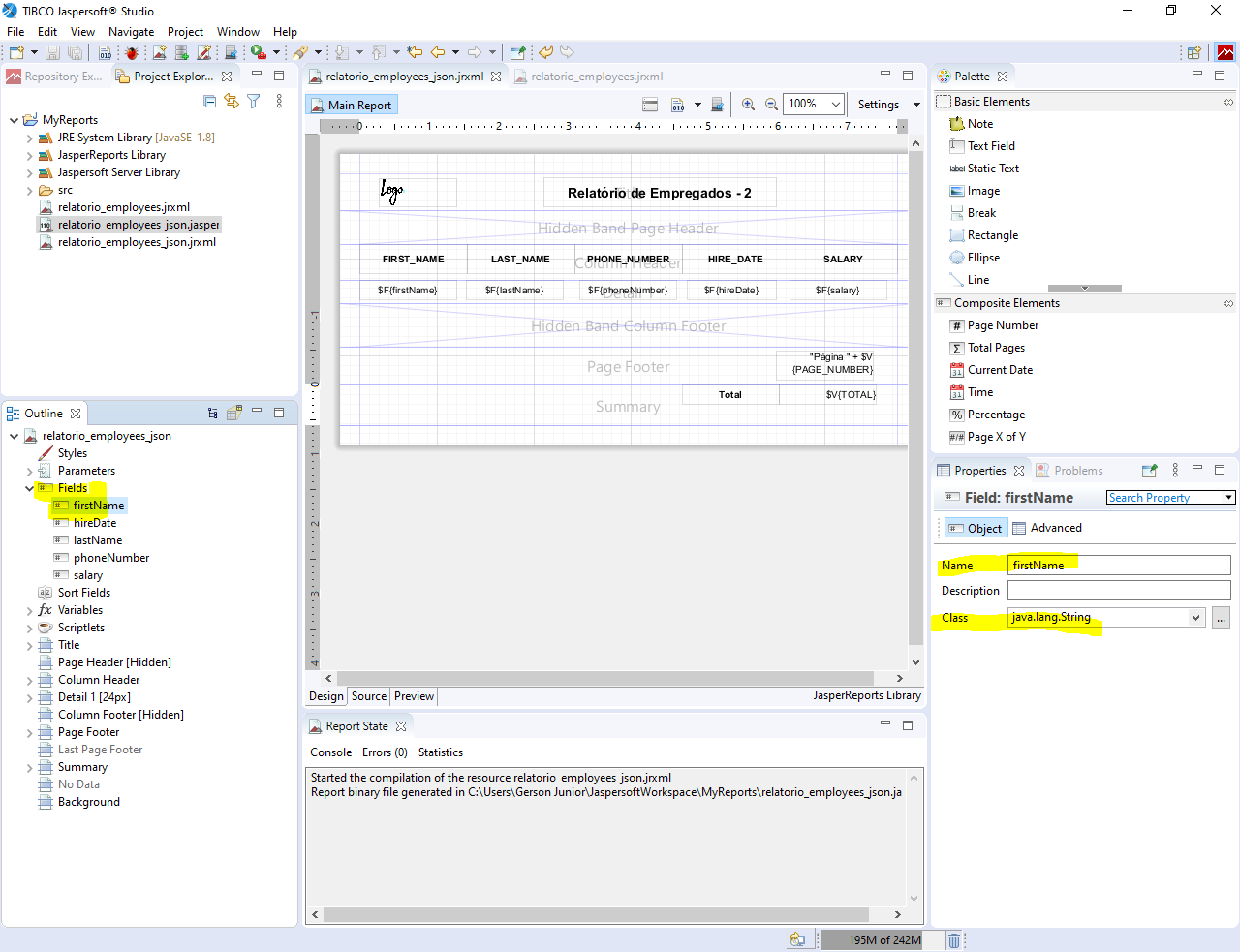
****

****

****

****

Crie os fields com tipo e nome de acordo com os dados retornados

****

**Controller**

@PostMapping("/json")

public ResponseEntity<byte[]> relatorioJson(@RequestBody ParametrosDTO dto) throws Exception {

byte[] relatorio = relatorioService.exportRelatorioJson(dto);

return ResponseEntity.ok()

.header("Content-Disposition", "attachment; filename=relatorio\_employees\_"+LocalDate.now()+".pdf")

.body(relatorio);

}

**Service**

public byte[] exportRelatorioJson(ParametrosDTO dto) throws SQLException, JRException, ParseException {

List<Employee> listEmployee = new ArrayList<Employee>();

List<EmployeeDTO> listEmployeeDTOs = new ArrayList<EmployeeDTO>();

if(dto.getDataContratacao() == null) {

listEmployee = employeeRepository.findAll();

}else {

SimpleDateFormat formato = new SimpleDateFormat("dd/MM/yy");

Date data = formato.parse(dto.getDataContratacao());

listEmployee = employeeRepository.findByHireDate(data);

}

listEmployee.forEach(employee -> {

listEmployeeDTOs.add(modelMapper.map(employee, EmployeeDTO.class));

});

Map<String, Object> parametros = new HashMap<>();

parametros.put("REPORT\_LOCALE", new Locale("pt", "BR"));

InputStream inputStream = this.getClass().getResourceAsStream("/relatorios/relatorio\_employees\_json.jasper");

JasperPrint jasperPrint = JasperFillManager.fillReport(inputStream, parametros,

new JRBeanCollectionDataSource(listEmployeeDTOs));

return JasperExportManager.exportReportToPdf(jasperPrint);

}