UNIVERSIDADE DO ESTADO DE SANTA CATARINA
CENTRO DE EDUCAÇÃO SUPERIOR DO ALTO VALE DO ITAJAÍ - CEAVI
BACHARELADO EM ENGENHARIA DE SOFTWARE

WILLESON THOMAS DA SILVA

RUAN FELIPE MAÇANEIRO

MANUAL DE USO DO FRAMEWORK

IBIRAMA

2017

WILLESON THOMAS DA SILVA

RUAN FELIPE MAÇANEIRO

MANUAL DE USO DO FRAMEWORK

Manual do Trabalho 03 de Padrões de Projeto apresentado ao curso de Engenharia de Software do Centro de Educação Superior do Alto Vale do Itajaí (CEAVI), da Universidade do Estado de Santa Catarina (UDESC)

IBIRAMA

2017

SUMÁRIO

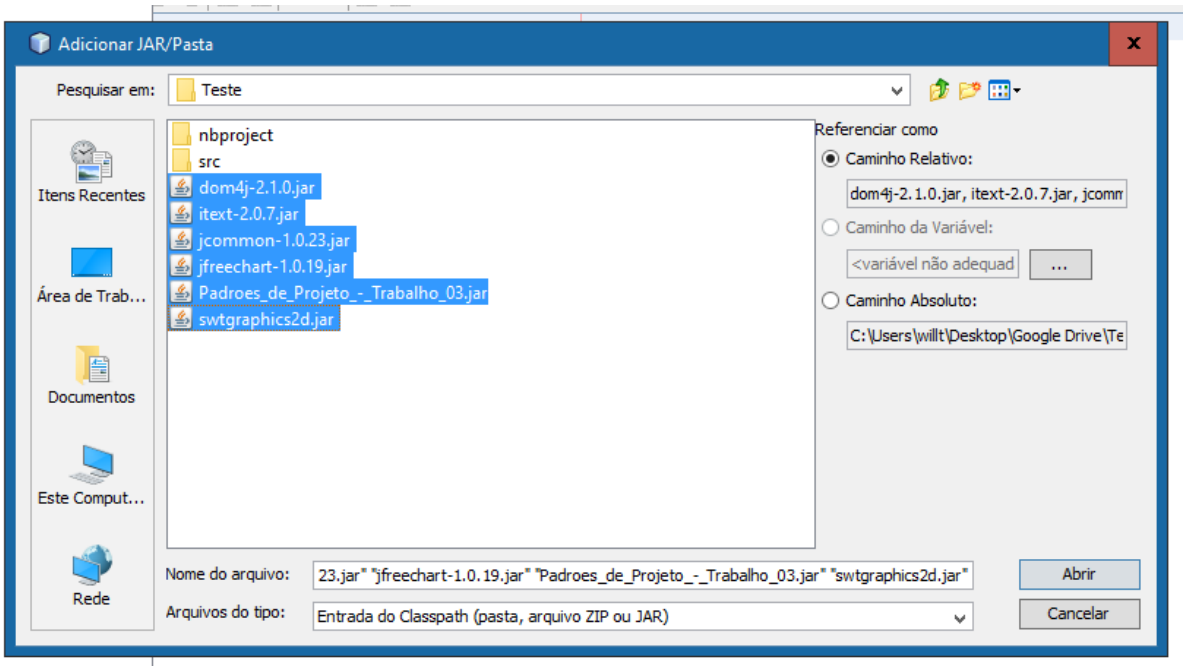
1 FRAMEWORK PARA GERAÇÃO DE RELATORIOS

O Manual refere-se a um framework que visa gerar relatórios nos formatos **HTML, PDF e RTF.** Com esse framework, pode-se gerar relatórios com **gráficos, tabelas, imagens de cabeçalho e rodapé, além de permitir alterações de estilo.**

Com o framework, existe a possibilidade de gerar um relatório padrão com configurações de estilo default. Pode-se também gerar o HTML, PDF e RTF a partir da **leitura de um arquivo XML,** no qual tem-se atributos de estilos e outras configurações, permitindo a geração de relatórios no formato que melhor convir.

1.1 IMPORTAR FRAMEWORK PARA O PROJETO

Para o devido uso do framework para geração de relatórios, deve-se primeiramente importa-lo para o projeto que irá usá-lo, além de importar os JARS dependentes que estão na pasta chamada “Manual”.



Após adicionar as importações necessárias, o usuário deve na raiz do projeto os seguintes itens adicionar os seguintes itens:

- **Arquivo:** configuração.xml
- **Imagem:** udesc.png
- **Pasta:** imagens

2 CRIAR CLASSES

Primeiramente, deve-se criar uma classe que será utilizada para geração de gráficos e tabelas no relatório. Neste documento, tem-se como exemplo a classe **"Pessoa",** que possui como atributos, **nome, salario e profissao.**

```
11  public class Pessoa {
12      private String nome;
13      private String salario;
14      private String profissao;
15
16      public Pessoa() {
17      }
18
19      public Pessoa(String nome,
20      String salario, String profissao) {
21          this.nome = nome;
22          this.salario = salario;
23          this.profissao = profissao;
24      }
25
26      //Getter e Seter... de nome e salario
```

Figura 1

Após criar a classe **"Pessoa"**, o usuário deve preocupar-se em criar uma classe que será o relatório de **"Pessoa"**, que pode ser chamada de **"RelatorioPessoa"**. ***Essa classe estende a classe "Relatorio" do framework***.

```
16  public class RelatorioPessoa extends Relatorio {
17
18      public RelatorioPessoa() {
19      }
20
21      public RelatorioPessoa(String nomeArquivo, String cabecalho,
22      String data, String rodape, String fontTabelaCabecalho,
23      String fontSizeTabelaCabecalho, String fontTabelaLinha,
24      String fontSizeTabelaLinha, String fontCabecalho,
25      String fontSizeCabecalho, String fontRodape,
26      String fontSizeRodape) {
27
28       super(nomeArquivo, cabecalho, data, rodape, fontTabelaCabecalho,
29       fontSizeTabelaCabecalho, fontTabelaLinha, fontSizeTabelaLinha,
30       fontCabecalho, fontSizeCabecalho, fontRodape, fontSizeRodape);
31      }
32
33  }
```

Figura 2

3 ADICIONAR INFORMAÇÕES AO RELATORIO

Para adicionar informações no relatório, com o intuito de gerar tabelas e gráficos, deve-se criar um “ArrayList”, referente a classe com a qual está trabalhando e adicionar os dados.

```
17  public class Teste01 {
18
19      public static void main(String[] args) {
20          //Relatorio Padrão
21          try {
22              ArrayList<Pessoa> listal = new ArrayList<>();
23              listal.add(new Pessoa("Albert Einsten", "4000", "Físico"));
24              listal.add(new Pessoa("Issac Newton", "6000", "Físico e Matemático"));
25              listal.add(new Pessoa("Neil armstrong", "9000", "Astronauta"));
26              listal.add(new Pessoa("Willeson", "2000", "Estudante"));
27              listal.add(new Pessoa("Ruan", "1000", "Estudante"));
28              RelatorioPessoa pl = new RelatorioPessoa();
29              pl.addColunaTabela("Nome", "Salario", "Profissão");
30              for (int i = 0; i < listal.size(); i++) {
31                  pl.addLinhaTabela(listal.get(i).getNome(),
32                  listal.get(i).getSalario(), listal.get(i).getProfissao());
33              }
```

Figura 3

No exemplo apresentado, cria-se um ”ArrayList” de **“Pessoa”** e na **linha 28**, tem-se a criação de um objeto do tipo “RelatorioPessoa”.

Das **linhas 29 a 33**, é adicionado dados na tabela, nesse instante o framework já entende que deseja-se gerar uma tabela com os dados listados. No entanto, caso as **linhas 29 até 33** não for executada, não será gerado as tabelas.

Além de adicionar valores na tabela, pode-se mudar as informações referentes aos seguintes atributos:

- **Nome do arquivo.**
- **Cabeçalho.**
- **Data.**
- **Rodapé.**
- **Imagem no cabeçalho.**
- **Imagem no rodapé.**
- **Tamanho de Página (A4 ou Carta)**

Para atribuir valores aos atributos citados, deve-se proceder com o seguinte código:

```
36          pl.setNomeArquivo("Relatorio - Teste 02 (Carro 02)");
37          pl.setCabecalho("Cabecalho - Teste 02 (Carro 02)");
38          pl.setData("29/12/2017");
39          pl.setTamanhoPagina("A4");
40          pl.setRodape("Rodape - Teste 02 (Carro 02)");
41          pl.setImagemCabecalho("udesc.jpg");
42          pl.setImagemRodape("udesc.jpg");
```

Figure 4

Caso não seja adicionado valores como nome do arquivo, imagens, tipo de fonte e dentre outras opções, **será gerado um relatório padrão**, com valores pré-definidos para tais atributos.

No caso de adicionar imagens e, **não necessitar desta** no cabeçalho ou rodapé, então deve-se **"setar" como "null",** pois caso isso não for especificado, o framework por padrão adiciona uma imagem ao local que não foi especificado, tanto no rodapé quanto cabeçalho.

Assim, segue o código para o caso adicionar imagem apenas no cabeçalho.

```
36          pl.setNomeArquivo("Relatorio - Teste 02 (Carro 02)");
37          pl.setCabecalho("Cabecalho - Teste 02 (Carro 02)");
38          pl.setData("29/12/2017");
39          pl.setTamanhoPagina("A4");
40          pl.setRodape("Rodape - Teste 02 (Carro 02)");
41          pl.setImagemCabecalho("udesc.jpg");
42          pl.setImagemRodape(null);
```

Figure 5

4 ADICIONAR ESTILOS AO RELATORIO

No framework existe dois tipos de estilos, um para HTML e outro para PDF e RTF, no qual pode-se proceder da seguinte forma.

Gerar estilo para o HTML.

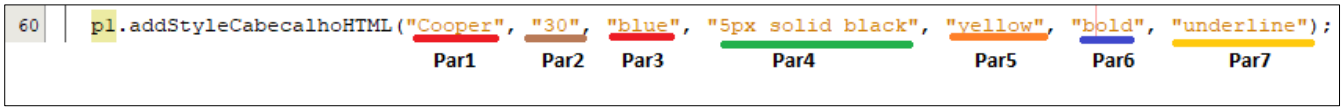


Figure 6

- **Par1:** Refere-se ao **tipo de fonte**.
- **Par2:** Refere-se ao **tamanho da fonte**.
- **Par3:** Refere-se a **cor da fonte.**
- **Par4:** Refere-se **cor e tipo da borda**.
- **Par5:** Refere-se a **cor de fundo**.
- **Par6:** Determina se a fonte será **negrito.**
- **Par7:** Determina se a fonte será **sublinhada.**

Gerar estilo para PDF e RTF.

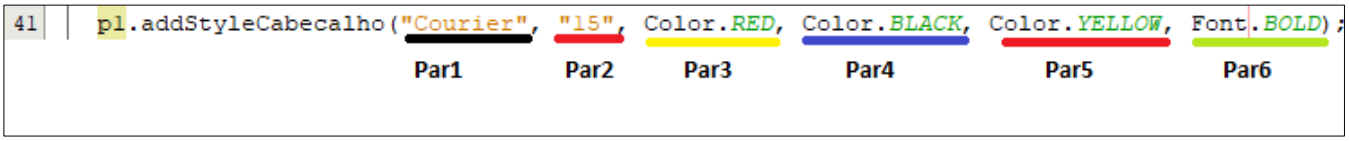


Figura 7

- **Par1:** Refere-se ao **tipo de fonte**.
- **Par2:** Refere-se ao **tamanho da fonte**.
- **Par3:** Refere-se a **cor da fonte.**
- **Par4:** Refere-se **cor da borda**.
- **Par5:** Refere-se a **cor de fundo**.
- **Par6:** Determina se a fonte será **negrito ou sublinhado.**

Sabendo o significado de cada atributo, para adicionar estilos ao relatório procede-se com o seguinte código para **HTML (Figura 8), PDF e RTF (Figura 9).**

```
60  pl.addStyleCabecalhoHTML("Cooper", "30", "blue", "5px solid black", "yellow", "bold", "underline");
61  pl.addStyleRodapeHTML("Cooper", "30", "red", "5px solid black", "grey", "bold", "");
62  pl.addStyleTabelaColunaHTML("Arial", "20", "blue", "", "underline");
63  pl.addStyleTabelaLinhaHTML("Arial", "20", "red", "", "");
```

Figura 8

```
41  pl.addStyleCabecalho("Courier", "15", Color.RED, Color.BLACK, Color.YELLOW, Font.BOLD);
42  pl.addStyleRodape("Courier", "15", Color.BLUE, Color.BLACK, Color.PINK, Font.BOLD);
43  pl.addStyleTabelaColuna("Courier", "10", Color.RED, Font.BOLDITALIC);
44  pl.addStyleTabelaLinha("Courier", "10", Color.BLUE, Font.NORMAL);
```

Figure 9

.

4.1 LISTA DE FONTES SUPORTADAS

Para o framework apresentado no presente manual, tem-se apenas alguns tipos fontes suportadas, sendo listadas a seguir.

- **COURIER**
- **HELVETICA**
- **TIMES_ROMAN**
- **SYMBOL**
- **ZAPFDINGBATS**
- **NORMAL**
- **BOLD**
- **ITALIC**
- **UNDERLINE**
- **STRIKETHRU**
- **BOLDITALIC**
- **UNDEFINED**
- **DEFAULTSIZE**

5 ADICIONAR GRAFICO AO RELATORIO

Para adicionar gráfico no relatório deve-se tomar atenção apenas com os parâmetros que estão sendo passados. Primeiramente deve criar o tipo de relatório que será gerado.

O **“TipoRelatorio” é uma classe abstrata** da qual pode ser usada para instanciar os três tipos de relatórios **(PDF, RTF e HTML)**. Deve-se proceder:

- TipoRelatorio pdf = new RelatorioPDF(p1);
- TipoRelatorio rtf = new RelatorioRTF(p1);
- TipoRelatorio html = new RelatorioHTML(p1);

Em ambos os formatos, deve-se passar como parâmetro o objeto criado com a classe **“RelatorioPessoa”** **(Figura 2)****.**

Agora, o método para geração de gráficos, possui alguns parâmetros que merece atenção.

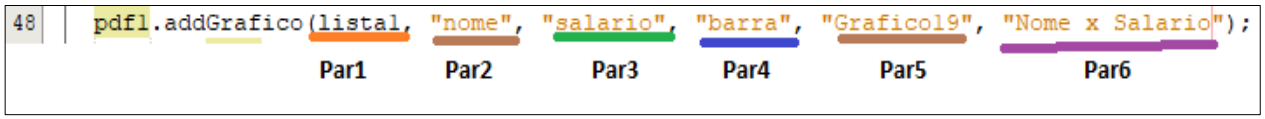


Figura 10

- **Par1:** Refere-se a lista de **“Pessoa”** criada no para gerar o gráfico.
- **Par2:** Refere-se a um atributo da classe da **“Pessoa”** que será alocado no gráfico.
- **Par3:** Refere-se a um atributo da classe da **“Pessoa”** que será alocado no gráfico, **nesse caso o atributo deve ser sempre um número.**
- **Par4:** Refere-se ao tipo de gráfico, no caso só existem dois tipos “barra”, “pizza”.
- **Par5:** Refere-se ao **nome da imagem** do gráfico.
- **Par6:** Refere-se ao **título do gráfico.**

Com isso, pode-se adiciona inúmeros gráficos ao relatório, informando os valores corretos.

Para gerar os relatórios com gráfico deve-se proceder da seguinte forma.

```
35          //Relatorio pdf
36          TipoRelatorio pdf1 = new RelatorioPDF(p1);
37          pdf1.addGrafico(lista1, "nome", "salario", "barra", "Grafico1", "Nome x Salario");
38          pdf1.addGrafico(lista1, "nome", "salario", "pizza", "Grafico2", "Nome x Salario");
39          pdf1.gerarRelatorio();
40
41          //Relatorio rtf;
42          TipoRelatorio rtf1 = new RelatorioRTF(p1);
43          rtf1.addGrafico(lista1, "nome", "salario", "pizza", "Grafico3", "Nome x Salario");
44          rtf1.addGrafico(lista1, "nome", "salario", "barra", "Grafico4", "Nome x Salario");
45          rtf1.addGrafico(lista1, "nome", "salario", "barra", "Grafico5", "Nome x Salario");
46          rtf1.addGrafico(lista1, "nome", "salario", "pizza", "Grafico6", "Nome x Salario");
47          rtf1.gerarRelatorio();
48
49          //Relatorio html
50          TipoRelatorio html1 = new RelatorioHTML(p1);
51          html1.addGrafico(lista1, "nome", "salario", "barra", "Grafico7", "Nome x Salario");
52          html1.addGrafico(lista1, "nome", "salario", "pizza", "Grafico8", "Nome x Salario");
53          html1.addGrafico(lista1, "nome", "salario", "barra", "Grafico9", "Nome x Salario");
54          html1.gerarRelatorio();
```

Figura 11

6 CRIAR RELATORIO PADRÃO

Para criar um relatório padrão com o exemplo “Pessoa”, deve-se criar as classes das figuras 1 e 2.

```
public static void main(String[] args) {
    //Relatorio Padrão
    try {
        ArrayList<Pessoa> listal = new ArrayList<>();
        listal.add(new Pessoa("Albert Einsten", "4000", "Físico"));
        listal.add(new Pessoa("Issac Newton", "6000", "Físico e Matemático"));
        listal.add(new Pessoa("Neil armstrong", "9000", "Astronauta"));
        listal.add(new Pessoa("Willeson", "2000", "Estudante"));
        listal.add(new Pessoa("Ruan", "1000", "Estudante"));
        RelatorioPessoa pl = new RelatorioPessoa();
        pl.addColunaTabela("Nome", "Salario", "Profissão");
        for (int i = 0; i < listal.size(); i++) {
            pl.addLinhaTabela(listal.get(i).getNome(),
            listal.get(i).getSalario(), listal.get(i).getProfissao());
        }
 //Relatorio pdf
TipoRelatorio pdfl = new RelatorioPDF(pl);
pdfl.addGrafico(listal, "nome", "salario", "barra", "Grafico1", "Nome x Salario");
pdfl.addGrafico(listal, "nome", "salario", "pizza", "Grafico2", "Nome x Salario");
pdfl.gerarRelatorio();
//Relatorio rtf;
TipoRelatorio rtfl = new RelatorioRTF(pl);
rtfl.addGrafico(listal, "nome", "salario", "pizza", "Grafico3", "Nome x Salario");
rtfl.addGrafico(listal, "nome", "salario", "barra", "Grafico4", "Nome x Salario");
rtfl.gerarRelatorio();
//Relatorio html
TipoRelatorio htmll = new RelatorioHTML(pl);
htmll.addGrafico(listal, "nome", "salario", "barra", "Grafico7", "Nome x Salario");
htmll.gerarRelatorio();
```

Figura 12

Ao criar a classe da figura 12 com o método “main”. Pode-se observar que não foi especificado nenhum valor de estilo, nome do arquivo, data, imagens e dentre outros. Nesse caso, por padrão, ao gerar o relatório, será atribuído valores as variáveis não especificadas.

- PDF
- RTF
- HTML

7 CRIAR RELATORIO COM ATRIBUIÇÃO DE ESTILOS

Para criar um relatório com atribuição de estilos no exemplo da classe “Pessoa”, deve-se criar as classes das figuras 1 e 2.

```
24      public static void main(String[] args) throws BadElementException, IOException,
25      FileNotFoundException, ClassNotFoundException, NoSuchFieldException, IllegalArgumentException, IllegalAccessException {
26
27          ArrayList<Pessoa> listal = new ArrayList<>();//Adiciona elementos ao Relatorio para gerar Graficos
28          listal.add(new Pessoa("Albert Einsten", "2000", "Físico"));
29          listal.add(new Pessoa("Issac Newton", "3000", "Físico e Matemático"));
30          listal.add(new Pessoa("Neil armstrong", "4000", "Astronauta"));
31          listal.add(new Pessoa("Willeson", "1000", "Estudante"));
32          RelatorioPessoa pl = new RelatorioPessoa();
33          pl.addColunaTabela("Nome", "Salario", "Profissão");
34          for (int i = 0; i < listal.size(); i++) {
35              pl.addLinhaTabela(listal.get(i).getNome(), listal.get(i).getSalario(), listal.get(i).getProfissao());
36          }
37          pl.setNomeArquivo("Relatorio - Teste 02 (Pessoa 01)");
38          pl.setCabecalho("Cabecalho - Teste 02 (Pessoa 01)");
39          pl.setData("15/12/2017");
40          pl.setRodape("Rodape - Teste 02 (Pessoa 01)");
41          pl.setImagemCabecalho("udesc.jpg");
42          pl.setImagemRodape(null);
43          pl.addStyleCabecalhoHTML("Cooper", "30", "blue", "5px solid black", "yellow", "bold", "underline");
44          pl.addStyleRodapeHTML("Cooper", "30", "red", "5px solid black", "grey", "bold", "");
45          pl.addStyleTabelaColunaHTML("Arial", "20", "blue", "", "underline");
46          pl.addStyleTabelaLinhaHTML("Arial", "20", "red", "", "");
47          pl.setImagemCabecalho("udesc.jpg");
48          //Relatorio html
49          TipoRelatorio html1 = new RelatorioHTML(pl);
50          html1.addGrafico(listal, "nome", "salario", "barra", "Grafico25", "Nome x Salario");
51          html1.addGrafico(listal, "nome", "salario", "pizza", "Grafico26", "Nome x Salario");
52          html1.addGrafico(listal, "nome", "salario", "barra", "Grafico27", "Nome x Salario");
53          html1.gerarRelatorio();
```

```
55              pl.addStyleCabecalho("Courier", "15", Color.RED, Color.BLACK, Color.YELLOW, Font.BOLD);
56              pl.addStyleRodape("Courier", "15", Color.BLUE, Color.BLACK, Color.PINK, Font.BOLD);
57              pl.addStyleTabelaColuna("Courier", "10", Color.RED, Font.BOLDITALIC);
58              pl.addStyleTabelaLinha("Courier", "10", Color.BLUE, Font.NORMAL);
59
60              //Relatorio pdf
61              TipoRelatorio pdf1 = new RelatorioPDF(pl);
62              pdf1.addGrafico(listal, "nome", "salario", "barra", "Grafico19", "Nome x Salario");
63              pdf1.addGrafico(listal, "nome", "salario", "pizza", "Grafico20", "Nome x Salario");
64              pdf1.gerarRelatorio();
65
66              //Relatorio rtf;
67              TipoRelatorio rtf1 = new RelatorioRTF(pl);
68              rtf1.addGrafico(listal, "nome", "salario", "pizza", "Grafico21", "Nome x Salario");
69              rtf1.addGrafico(listal, "nome", "salario", "barra", "Grafico22", "Nome x Salario");
70              rtf1.addGrafico(listal, "nome", "salario", "barra", "Grafico23", "Nome x Salario");
71              rtf1.addGrafico(listal, "nome", "salario", "pizza", "Grafico24", "Nome x Salario");
72              rtf1.gerarRelatorio();
```

Figura 13

Na figura 13 é criado uma classe com o método “main”. Pode-se observar que nesse caso foi especificado valores de estilo, nome do arquivo, data, imagens e dentre outros. Assim, será atribuído os valores especificados às variáveis.

- PDF
- RTF
- HTML

8 CRIAR RELATORIO POR LEITURA XML

XML é uma recomendação da W3C para gerar linguagens de notação para necessidades especiais. É um dos subtipos da SGML capaz de descrever diversos tipos de dados.

Framework no presente manual, o arquivo é composto por dois estilos de TAGS: “**relatorio” e “gerarRelatorio”.**

A TAG “**relatorio”** tem como atributos:

- **Id:** Identificador da classe.
- **Class:** nome da classe com a qual está trabalhando.
- **Singleton:** aceita apenas valores booleanos (true ou false).

Nessa TAG, tem-se configurações de estilos, nomeArquivo, data e dentre outras do relatório, que pode ser alterado seus valores.

```
4   <relatorio id="1" class="RelatorioPessoa" singleton="true">
5       <nomeArquivo>Relatorio - Pessoa 01</nomeArquivo>
6       <cabecalho>Titulo do Relatorio - Pessoa 01</cabecalho>
7       <data>10/11/2017</data>
8       <rodape>Rodape - Pessoa 01</rodape>
9       <fontTabelaCabecalho>Cooper</fontTabelaCabecalho>
10      <fontSizeTabelaCabecalho>20</fontSizeTabelaCabecalho>
11      <fontTabelaLinha>Cooper</fontTabelaLinha>
12      <fontSizeTabelaLinha>15</fontSizeTabelaLinha>
13      <fontCabecalho>Cooper</fontCabecalho>
14      <fontSizeCabecalho>20</fontSizeCabecalho>
15      <fontRodape>Cooper</fontRodape>
16      <fontSizeRodape>20</fontSizeRodape>
17  </relatorio>
```

Figura 14

Já o “**gerarRelatorio”** é constituído dos seguintes atributos:

- **Id:** Identificador do tipo e relatório (PDF, RTF, HTML).
- **Class:** classe do framework que será referenciada (RelatorioPDF, RelatorioRTF, RelatorioHTML)

- **IdGerar:** Identifica a classe da TAG “relatorio”, com a qual será trabalhada para geração do relatorio.

```
65      <gerarRelatorio id="100" class="RelatorioRTF" idGerar="1"/>
66      <gerarRelatorio id="600" class="RelatorioPDF" idGerar="1"/>
67      <gerarRelatorio id="1000" class="RelatorioHTML" idGerar="1"/>
68
69      <gerarRelatorio id="200" class="RelatorioRTF" idGerar="2"/>
70      <gerarRelatorio id="700" class="RelatorioPDF" idGerar="2"/>
71      <gerarRelatorio id="2000" class="RelatorioHTML" idGerar="2"/>
72
73      <gerarRelatorio id="300" class="RelatorioHTML" idGerar="3"/>
74      <gerarRelatorio id="3000" class="RelatorioRTF" idGerar="3"/>
75      <gerarRelatorio id="500" class="RelatorioPDF" idGerar="3"/>
76
77      <gerarRelatorio id="400" class="RelatorioRTF" idGerar="4"/>
78      <gerarRelatorio id="800" class="RelatorioPDF" idGerar="4"/>
79      <gerarRelatorio id="900" class="RelatorioHTML" idGerar="4"/>
```

Figura 15

Para gerar um relatorio, usando leitura de um XML, deve-se criar as classes “Pessoa” e “RelatorioPessoa” das figuras 2 e 3 respectivamente, apresentadas no item 2 do presente manual. Após criar as classes solicitadas, os procedimentos apresentados nos item 3,4 e 5 são validos para geração de relatórios com XML, apresentando apenas algumas alterações na sua implementação, que serão apresentadas a seguinte código.

```
20  public static void main(String[] args) throws Exception {
21      try {
22  (1)     Container container = new Container("configuracao.xml");
23
24          ArrayList<Pessoa> listal = new ArrayList<>();
25          listal.add(new Pessoa("Albert Einsten", "4000", "Físico"));
26          listal.add(new Pessoa("Issac Newton", "6000", "Físico e Matemático"));
27          listal.add(new Pessoa("Neil armstrong", "9000", "Astronauta"));
28          listal.add(new Pessoa("Willeson", "2000", "Estudante"));
29          listal.add(new Pessoa("Ruan", "1000", "Estudante"));
30
31  (2)     TipoRelatorio pdfl = container.criarRelatorio("600");
32          TipoRelatorio rtfl = container.criarRelatorio("100");
33          TipoRelatorio html1 = container.criarRelatorio("1000");
34
35  (3)     RelatorioPessoa p1 = pdfl.getRel();
36          RelatorioPessoa p2 = rtfl.getRel();
37          RelatorioPessoa p3 = html1.getRel();
38
39          p1.addColunaTabela("Nome", "Salario", "Profissão");
40          for (int i = 0; i < listal.size(); i++) {
41              p1.addLinhaTabela(listal.get(i).getNome(),
42              listal.get(i).getSalario(), listal.get(i).getProfissao());
43          }
44          pdfl.addGrafico(listal, "nome", "salario", "pizza", "Grafico39", "Nome x Salario");
45          pdfl.addGrafico(listal, "nome", "salario", "barra", "Grafico40", "Nome x Salario");
46          pdfl.addGrafico(listal, "nome", "salario", "barra", "Grafico41", "Nome x Salario");
47          pdfl.addGrafico(listal, "nome", "salario", "pizza", "Grafico42", "Nome x Salario");
48          pdfl.gerarRelatorio();
```

Figura 16

Os pontos do código marcados por retângulos vermelhos, mostram as mudanças mais significativas no código para gerar relatórios em relação aos códigos apresentados nos itens 6 e 7.

1. Nessa linha de código, é necessário instanciar a classe container, na qual terá como atributo, o nome do arquivo XML, que será usado para gerar os respectivos relatórios.
2. Nesse item, usa-se a classe abstrata “TipoRelatorio”, para receber o formato do relatorio que pretende-se gerar (**RelatorioHTML, RelatorioPDF, RelatorioRTF**), passando como parâmetro o id do tipo de relatorio, que irá ser verificado nas TAGS “gerarRelatorio”.
3. Com base no tipo de relatorio retornado no item “2” é feito injeção em RelatorioPessoa, o qual será usado para adicionar as colunas e linhas na tabela, além de outras funções. Já o objeto de “TipoRelatorio” será usado para adicionar gráficos e gerar os relatorios.

   Assim, o seguinte código gera relatórios, por leitura do arquivo XML.

```
20      public static void main(String[] args) throws Exception {
21          try {
22              Container container = new Container("configuracao.xml");
23              ArrayList<Pessoa> listal = new ArrayList<>();
24              listal.add(new Pessoa("Albert Einsten", "4000", "Físico"));
25              listal.add(new Pessoa("Issac Newton", "6000", "Físico e Matemático"));
26              listal.add(new Pessoa("Neil armstrong", "9000", "Astronauta"));
27              listal.add(new Pessoa("Willeson", "2000", "Estudante"));
28              listal.add(new Pessoa("Ruan", "1000", "Estudante"));
29              TipoRelatorio pdfl = container.criarRelatorio("600");
30              System.out.println(pdfl);
31              TipoRelatorio rtfl = container.criarRelatorio("100");
32              TipoRelatorio html1 = container.criarRelatorio("1000");
33
34              RelatorioPessoa p1 = pdfl.getRel();
35              RelatorioPessoa p2 = rtfl.getRel();
36              RelatorioPessoa p3 = html1.getRel();
37              p1.addColunaTabela("Nome", "Salario", "Profissão");
38              for (int i = 0; i < listal.size(); i++) {
39                  p1.addLinhaTabela(listal.get(i).getNome(),
40                  listal.get(i).getSalario(), listal.get(i).getProfissao());
41              }
42              pdfl.addGrafico(listal, "nome", "salario", "pizza", "Grafico39", "Nome x Salario");
43              pdfl.addGrafico(listal, "nome", "salario", "barra", "Grafico40", "Nome x Salario");
44              pdfl.gerarRelatorio();
45              rtfl.addGrafico(listal, "nome", "salario", "pizza", "Grafico39", "Nome x Salario");
46              rtfl.addGrafico(listal, "nome", "salario", "barra", "Grafico40", "Nome x Salario");
47              rtfl.addGrafico(listal, "nome", "salario", "barra", "Grafico41", "Nome x Salario");
48              rtfl.gerarRelatorio();
49              html1.addGrafico(listal, "nome", "salario", "barra", "Grafico43", "Nome x Salario");
50              html1.gerarRelatorio();
```

Figure 6

Ao analisar o arquivo XML, percebe-se configurações de estilo tanto dos cabeçalho e rodapé quanto de colunas e linhas da tabela. Nesse contexto, não tem-se necessidade de fazer tais configurações no método “main”, tendo em vista que as mesmas estão no XML.

- PDF
- RTF
- HTML