

DEFESA TCC

DashGen: Gerador de quadros de apresentação de dados em formato de dashboards.



**INSTITUTO
FEDERAL**

Pará

Campus
Belém

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Aluno: Glauber Matteis Gadelha

Orientador: Prof. Me. Claudio Roberto de Lima Martins

Introdução

- *Motivação da pesquisa:* Quadros de exibição de informações em formato de gráficos, ou Dashboards, facilitam o entendimento de conjuntos de dados volumosos. Porém ainda é bastante trabalhoso construí-los um a um.
- Como facilitar a geração destes quadros com engenharia baseada em reúso de software?

Objetivo do Trabalho

- **Desenvolver aplicação em Java 8 SE que:**
- Permita selecionar um arquivo CSV;
- Possibilite escolher entre alguns tipos de gráficos e associá-los aos atributos do arquivo;
- Com base em um gabarito predefinido e algumas de dependências, gere um Dashboard HTML5 que permita aplicação de filtros dinâmicos para análise.

Metodologia

- Pesquisa bibliográfica em artigos, teses e trabalhos científicos a respeito de reuso de software e geração de código fonte, além de bibliotecas gratuitas para geração de gráficos em páginas HTML5;
- Análise e modelagem do protótipo
- Desenvolvimento da aplicação e validação do Dashboard de saída

Pesquisa

REÚSO DE SOFTWARE

- (KRUEGGER, 1992) Reuso de software é o processo de se criar software a partir de software existente.

(SOFTWARE = código fonte, diagramas, documentação, bibliotecas ou até mesmo sistemas completos).

GERADORES DE PROGRAMAS

- (SOMMERVILLE, 2013) Geradores de programas são software que geram outros softwares a partir de especificações de alto Nível.

Pesquisa

Abordagem escolhida para geração do Dashboard

- **Programação generativa baseada em gabaritos**

Pesquisa

- Gabaritos consistem em partes parcialmente prontas do produto, com marcações que são substituídas por parâmetros fornecidos ao gerador, que faz a composição, gerando o produto final concluído (LUCRÉDIO, 2009).

Para aplicar a geração de código baseada em gabaritos, utilizamos

MOTORES DE GABARITOS (TEMPLATE ENGINES)

Motores de Gabaritos

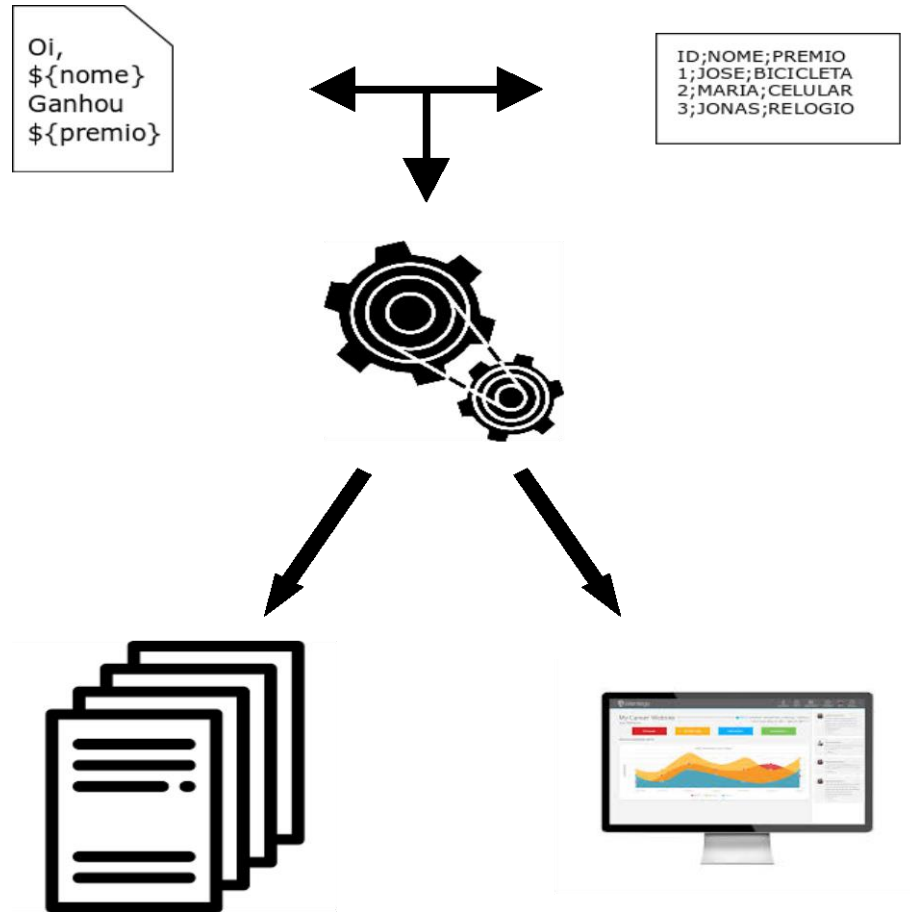
- partes ou componentes de software que têm a função de combinar um ou mais gabaritos com um dado modelo de dados, gerando um ou mais artefatos de saída como resultado de seu processamento(WIKIPEDIA.ORG, 2020).

Motores de Gabaritos

Composição de um Sistema de processamento de Gabaritos

- Um modelo de dados associado
- Um ou mais gabaritos fonte
- Um processador ou motor de gabaritos
- Artefato final gerado(WIKIPEDIA.ORG, 2020).

Motores de Gabaritos



Motores de Gabaritos

- Várias linguagens de programação possuem motores de gabaritos desenvolvidos para facilitar a implementação de aplicações com padrão arquitetural MVC

Entre várias soluções Java, foram analisados:

- Apache Velocity
- **Apache Freemarker**

Fundamentação Teórica

Apache Freemarker

- motor de gabaritos para Java;
- versão inicial disponibilizada em 2000, desenvolvida por Benjamin Geer e Mike Bayer (WIKIPEDIA.ORG, 2018).
- linguagem especializada chamada FTL (Freemarker Template Language)
- desenvolvido com a finalidade de geração de páginas HTML em frameworks que adotassem o padrão arquitetural MVC (Modelo-Visão-Controle)

Freemarker Template Language

Conteúdo FTL

Iterador

Coleção (modelo de dados)

```
<#list graficos as grafico>
<div id="grafico${grafico_index +1}" class='col-xs-12 col-md-12 col-lg-4'>
    <strong>${grafico.nome}</strong>
    <span class="reset" style="display: none;">Selected:<span
class="filter"></span></span>
    <a class="reset" href="javascript:grafico${grafico_index
+1}.filterAll();dc.redrawAll();" style="display: none;">reset</a>
    <div class="clearfix"></div>
</div>
</#list>
```

Acesso a atributo

Freemarker Template Language

Saída

Processador de gabarito executa as iterações, gerando a <div> final com as substituições para todos os itens da coleção

```
<div id="grafico1" class='col-xs-12 col-md-12 col-lg-4'><strong>Sexo</strong>
  <span class="reset" style="display: none;">Selected: <span
  class="filter"></span></span>
  <a class="reset" href="javascript:grafico1.filterAll();dc.redrawAll();"
  style="display:none;">reset</a>
  <div class="clearfix"></div>
</div>

<div id="grafico2" class='col-xs-12 col-md-12 col-lg-4'><strong>Religiao</strong>
  <span class="reset" style="display: none;">Selected: <span
  class="filter"></span></span>
  <a class="reset" href="javascript:grafico2.filterAll();dc.redrawAll();"
  style="display: none;">reset</a>
  <div class="clearfix"></div>
</div>

<div id="graficoN"...
```

Bibliotecas de suporte do DashGen

- **Apache Commons CSV**
 - (Acesso e iteração no conteúdo de arquivos padrão CSV)
- **Apache Commons IO**
 - (Acesso a arquivos e cópia recursiva do conteúdo de diretórios)
- **ZeroTurnaround ZT-ZIP**
 - (compactação de arquivos e diretórios recursivamente)
- **Apache Maven**
 - (gerenciador de dependências)

Bibliotecas de suporte para o Dashboard

- DC.js
 - (Geração de gráficos)
- Dependências
 - Crossfilter (filtros cruzados dinâmicos)
 - D3 (Data-Driven Documents) (renderização de gráficos SVG)

Desenvolvimento da Aplicação

Fase de análise

- **Análise de Requisitos**
 - (Levantados os requisitos funcionais e não funcionais do DashGen)
- **Análise de Casos de Uso**
 - (Definido e especificado o caso de uso a partir da perspectiva do usuário do DashGen)
- **Diagramas de Classes de Análise e de Implementação**
 - (Identificação das classes candidatas e posteriormente o modelo de implementação com atributos e operações)
- **Diagramas de sequência: Geral e da rotina do Gerador**
 - (Análise da dinâmica da aplicação)

Desenvolvimento da Aplicação

Fase de Implementação

- Aplicação aderente ao padrão arquitetural MVC (Modelo – Visão – Controle);
- Decidido o uso do JavaFX 8 para a GUI. Aplicação de tela única onde o usuário especifica todas as informações desejadas e finaliza o dashboard.
- Saída é copiada para o diretório de destino especificado e é gerado um arquivo compactado em formato ZIP.

Desenvolvimento da Aplicação GUI

Gerador de Dashboards

IFPA DashGen

C:\Users\glaubergad\Desktop\teste_salario.csv

Arquivo CSV

C:\Users\glaubergad\Desktop\Dashboard

Selecionar Destino

Titulo do Dashboard

Meu Dashboard

Graficos

Titulo do Grafico

Renda Bruta por Religião

Tipo de Gráfico

dc.rowChart

Atributo Dimensão

Atributo do CSV

Atributo Somatória

Atributo do CSV

Redução

☐ Contagem

☒ Somatória

Graficos Adicionados

SEXO|SEXO|Sexo|dc.pieChart

RELIGIAO|RELIGIAO|Religião|dc.rowChart

IDADE|IDADE|Idade|dc.lineChart

IDADE|SALARIO|Renda Bruta por Idade|dc.rowChart

SEXO|SALARIO|Renda Bruta por Sexo|dc.rowChart

RELIGIAO|SALARIO|Renda Bruta por Religião|dc.rowChart

Limpar Gráficos

Adicionar Gráfico

Finalizar Dashboard

Desenvolvimento da Aplicação

The screenshot shows the 'Gerador de Dashboards' application window. The title bar reads 'Gerador de Dashboards'. The main title is 'IFPA DashGen'. The interface includes several input fields and buttons, with annotations explaining their functions:

- a) Especificar o arquivo CSV:** Points to the 'Arquivo CSV' button, which is next to the text 'C:\Users\glaubergad\Desktop\teste_sala'.
- b) selecionar pasta de destino:** Points to the 'Selecionar Destino' button, which is next to the text 'C:\Users\glaubergad\Desktop\Dashboar'.
- c) Especificar título do dashboard:** Points to the 'Titulo do Dashboard' input field, which contains the text 'Meu Dashboard'.
- d) Especificar título, tipo de gráfico, Tipo de redução e , atributos de dimensão E medição:** This annotation points to four fields: 'Titulo do Grafico' (containing 'Sexo'), 'Tipo de Gráfico' (a dropdown menu showing 'dc.pieChart'), 'Atributo Dimensão' (a dropdown menu showing 'SEXO'), and 'Atributo Somatória' (a dropdown menu showing 'Atributo do CSV').
- e) adicionar gráfico à lista:** Points to the 'Adicionar Gráfico' button.
- f) finalizar dashboard:** Points to the 'Finalizar Dashboard' button.

Other visible elements include a 'Redução' section with radio buttons for 'Contagem' (selected) and 'Somatória', a 'Graficos Adicionados' list containing 'IDADE|IDADE|Idade|dc.lineChart', and buttons for 'Limpar Gráficos' and 'Finalizar Dashboard'.

Desenvolvimento da Aplicação

Classe Controller

- Responsável pela mediação entre as ações e entradas de dados do usuário e os modelos;
- Merecem destaque os métodos `setDataset()`, `addGrafico()` e `endDashboard()`.

Desenvolvimento da Aplicação

Classe Controller – setDataset()

```
01. //Método instancia um objeto Dataset com o conjunto de dados contido no arquivo CSV selecionado.  
02. private void setDataset() throws Exception {  
03.     dataset = new Dataset(csvFile);  
04.     populateCbAtributoX(dataset.getAtributos());  
05.     if (dataset.getNumFields() > 0) {  
06.         populateCbAtributoY(dataset.getAtributos());  
07.         lblSomatoria.setVisible(true);  
08.         cbAtributoY.setVisible(true);  
09.         rbReduceSum.setVisible(true);  
10.     } else {  
11.         cbAtributoY.setVisible(false);  
12.         rbReduceSum.setVisible(false);  
13.     }  
14.     populateCbTipoGrafico();  
15. }
```

Desenvolvimento da Aplicação

Classe Controller – addGrafico()

```
01. public void addGrafico(ActionEvent actionEvent) {
02.     try {
03.         String atributoX = cbAtributoX.getSelectionModel().getSelectedItem().toString();
04.         String atributoY;
05.         if (this.grouping == 1) {
06.             atributoY = cbAtributoY.getSelectionModel().getSelectedItem().toString();
07.         } else {
08.             atributoY = cbAtributoX.getSelectionModel().getSelectedItem().toString();
09.         }
10.         String tipoGraf = cbTipoGrafico.getSelectionModel().getSelectedItem().toString();
11.         String titGraf = tfTituloGrafico.getText();
12.         graficos.add(new Grafico(atributoX, atributoY, titGraf, tipoGraf, this.grouping));
13.         lvGraficos.getItems().setAll(graficos);
14.         cbAtributoX.getSelectionModel().clearSelection();
15.         cbAtributoY.getSelectionModel().clearSelection();
16.         btnEndDashboard.setDisable(false);
17.     } catch (Exception e) {
18.         System.out.println("AddGrafico:" + e.toString());
19.         new Alert(Alert.AlertType.INFORMATION, "Voce nao selecionou os dados para inserir o grafico!").showAndWait();
20.     }
21. }
```

Desenvolvimento da Aplicação

Classe Controller – endDashboard()

```
01. //Metodo finaliza a geracao do Dashboard, executando o motor de templates FreeMarker, copiando os arquivos para a pasta de destino e gerando o zip de saida.
02. public void endDashboard(ActionEvent actionEvent) {
03.     setaDashboard();
04.     try {
05.         setaGerador();
06.         setaPackSaida();
07.     } catch (Exception e) {
08.         new Alert(Alert.AlertType.INFORMATION, "Nao foi possivel gerar o Dashboard. Verifique as opcoes escolhidas!").showAndWait();
09.     } finally {
10.         new Alert(Alert.AlertType.INFORMATION, "Dashboard gerado com sucesso!").showAndWait();
11.         resetForm();
12.     }
13. }
14.
15. //Metodo instancia objeto Dashboard juntando todas as informacoes definidas pelo usuario na UI
16. private void setaDashboard() {
17.     tituloDashboard = tfTituloDashboard.getText();
18.     dashboard = new Dashboard(dataset, tituloDashboard, graficos);
19. }
20.
21. //Metodo instancia a classe gerador, que processa o template FreeMarker e gera em disco o arquivo Dashboard.html.
22. private void setaGerador() throws IOException, TemplateException {
23.     new Gerador(dashboard, destFolder);
24. }
25.
26. //Metodo instancia novo pack de saida para gerar em disco o Dashboard completo.
27. private void setaPackSaida() throws IOException {
28.     new PackSaida(destFolder, csvFile);
29. }
```


Desenvolvimento da Aplicação

Classe Dataset

- Objeto concreto para representação do Arquivo CSV para o DashGen
- Uso da Apache Commons CSV para manipulação dos atributos e dados contidos no arquivo CSV
- Possui lógica de teste por força bruta para identificação dos atributos numéricos

Desenvolvimento da Aplicação

Classe Dataset

Figura 23 – Código fonte dos métodos isNumeric() e columnDiscoverType()

```
01. //Metodo para teste se uma String contem um valor numerico
02. private boolean isNumeric(String string) {
03.     try {
04.         Double.parseDouble(string);
05.         return true;
06.     } catch (NumberFormatException e) {
07.         return false;
08.     }
09. }
10.
11. //metodo que testa as 100 primeiras linhas do CSV a procura de valores numericos
12. private String columnDiscoverType(String col) {
13.     int numeric = 0, string = 0;
14.     //Necessario resetar o Record para ler novamente as linhas do CSV a partir do inicio
15.     this.setRecord();
16.     for (CSVRecord reg : this.record) {
17.         if (this.isNumeric(reg.get(col))) {
18.             numeric++;
19.         } else {
20.             string++;
21.         }
22.         if (reg.getRecordNumber() == 100)
23.             break;
24.     }
25.     if (string > 0) {
26.         return "T";
27.     } else {
28.         this.numFields++;
29.         return "N";
30.     }
31. }
```

Fonte: Elaborada pelo Autor

Desenvolvimento da Aplicação

Classe Gerador

- responsável por operar o Motor de Gabaritos Apache Freemarker.
- Deve receber como parâmetro de construtor o objeto Dashboard e o descritor File do diretório de destino especificado pelo usuário.

Desenvolvimento da Aplicação

Classe Gerador

Figura 24 – Código fonte dos métodos da classe Gerador

```
01. //Configurador dos dados necessarios para processamento do template
02. private void configurar() throws IOException {
03.     cfg.setDirectoryForTemplateLoading(HOME);
04.     cfg.setIncompatibleImprovements(new Version(2, 3, 28));
05.     cfg.setDefaultEncoding("UTF-8");
06.     cfg.setLocale(new Locale("pt", "BR"));
07.     cfg.setTemplateExceptionHandler(TemplateExceptionHandler.RETHROW_HANDLER);
08.     this.template = cfg.getTemplate("dashboard.ftl");
09. }
10.
11.
12. //Metodo responsavel por criar um HashMap com todos os dados e objetos a serem preenchidos sobre o template
13. private void carregaInput(){
14.     input.put("arquivo", dashboard.getDataset().getArquivo());
15.     input.put("titulo", dashboard.getNomeDashboard());
16.     input.put("graficos", dashboard.getListaGraficos());
17. }
18.
19. //Executa o processamento do template e a geracao do arquivo final, gravando no caminho especificado sob o nome
20. // dashboard.html
21. private void processar() throws IOException, TemplateException {
22.     BufferedWriter fileWriter = new BufferedWriter(new OutputStreamWriter(
23.         FileOutputStream(new File(destDir.getAbsolutePath() +sep+"dashboard.html")), "UTF-8"));
24.     template.process(input, fileWriter);
25.     fileWriter.close();
26. }
```

Fonte: Elaborada pelo Autor

Desenvolvimento da Aplicação

Classe PackSaida

- responsável por copiar as dependências do Dashboard para o diretório de destino e compactar esse conteúdo em um arquivo .ZIP

Desenvolvimento da Aplicação

Classe PackSaida

```
01. //Usa FileUtils para facilitar o processo de copia recursiva do diretorio e subdiretorios
02. private void processarCopia() throws IOException {
03.     FileUtils.copyFile(this.csvFile, new File(this.destDir.getAbsolutePath() + sep + "data" + sep + csvFile.getName()));
04.     FileUtils.copyDirectory(this.boilerplateDir, this.destDir);
05. }
06. //processa a compactacao dos arquivos contidos no diretorio de origem da aplicacao em um arquivo ZIP no diretorio de destino
07. private void processarZip() {
08.     String FILE_NAME = "dashgen.zip";
09.     ZipUtil.pack(this.destDir, new File(this.destDir.getAbsolutePath() + sep + FILE_NAME));
10. }
```

Desenvolvimento da Aplicação

Gabarito FTL

- Arquivo “modelo” de como deve ficar o artefato final.
- Texto fixo entremeado pelas marcações FTL.

Desenvolvimento da Aplicação

Gabarito FTL – Partes mais importantes

```
22.     <div class="row">
23.         <#list graficos as grafico>
24.             <div id="grafico${grafico_index +1}" class='col-xs-12 col-md-12 col-lg-4'>
25.                 <strong>${grafico.nome}</strong>
26.                 <span class="reset" style="display: none;">Selecionado:<span class="filter">
27.                     </span></span>
28.                 <a class="reset" href="javascript:grafico${grafico_index +1}.filterAll();dc.redrawAll();" style="display: none;">Limpar</a>
29.                 <div class="clearfix"></div>
30.             </div>
31.         </#list>
32.     </div>
```

```
40.     //instanciacao e identificacao do local de exibicao no DOM
41.     <#list graficos as grafico>
42.         var grafico${grafico_index+1} = ${grafico.tipo}("#grafico${grafico_index+1}");
43.     </#list>
```


Desenvolvimento da Aplicação

Gabarito FTL – Partes mais importantes

```
52.      //Estabelecimento da dimensao principal do grafico
53.      <#list graficos as grafico>
54.      var grafico${grafico_index+1}Dim = ndx.dimension(d => d.${grafico.atributoX});
55.      </#list>
56.
57.      //Redutores
58.      <#list graficos as grafico>
59.      <#if grafico.grouping == 1>
60.      var grafico${grafico_index+1}Group = grafico${grafico_index+1}Dim.group().reduceSum(d =
    > d.${grafico.atributoY});
61.      var grafico${grafico_index+1}minX = grafico${grafico_index+1}Dim.bottom(1)[0]["${grafic
    o.atributoY}"];
62.      var grafico${grafico_index+1}maxX = grafico${grafico_index+1}Group.top(1).value;
63.      <#else>
64.      var grafico${grafico_index+1}Group = grafico${grafico_index+1}Dim.group();
65.      var grafico${grafico_index+1}minX = grafico${grafico_index+1}Dim.bottom(1)[0]["${grafic
    o.atributoX}"];
66.      var grafico${grafico_index+1}maxX = grafico${grafico_index+1}Dim.top(1)[0]["${grafico.a
    tributoX}"];
67.      </#if>
68.      </#list>
```

Desenvolvimento da Aplicação

Gabarito FTL – Partes mais importantes

```
70.      //Parametros especificos de cada grafico
71.      <#list graficos as grafico>
72.      grafico${grafico_index+1}
73.      <#if grafico.tipo != "dc.lineChart">
74.          //Exibe os cinco atributos com valores mais altos, agrupando os demais em Others
75.          .cap(5)
76.      </#if>
77.          .dimension(grafico${grafico_index+1}Dim)
78.          .group(grafico${grafico_index+1}Group)
79.      <#if grafico.tipo == "dc.lineChart">
80.          //Define a escala do eixo X baseado nos valores minimo e maximo do agrupamento de r
educacao.
81.          .x(d3.scale.linear().domain([grafico${grafico_index+1}minX, grafico${grafico_index+
1}maxX]))
82.      </#if>
83.      <#if grafico.tipo == "dc.rowChart">
84.          //Garante o reajuste automatico do eixo X
85.          .elasticX(true)
86.      </#if>
87.      ;
88.  </#list>
```

Desenvolvimento da Aplicação

DashGen – Demonstração para Validação

- Arquivo CSV:
“c:\tccrepo\support\teste_salario.csv”.
- Pasta de destino:
”%USERPROFILE%\desktop\dashboard”
- Título do dashboard: “Renda Bruta Brasil”.

DashGen – Considerações Finais

- Aplicativo se provou eficaz para gerar painéis gráficos.
- Traz uma noção clara de aplicação usando motor Apache Freemarker
- Com algumas adaptações pode atender outros domínios e contextos.
- DC.js é uma ferramenta poderosa para gerar visualizações de dados.

DashGen – Trabalhos Futuros

- a) Refatoração do código fonte para melhorar o desempenho da aplicação;
- b) Transformação do mecanismo de composição em uma biblioteca reutilizável em outros contextos além da geração de *dashboards*;
- c) Usando o DashGen como base, criar um artefato nos padrões do projeto Maven;

DashGen – Trabalhos Futuros

- d) Aprimoramento do projeto, embarcando inteligência artificial capaz de propor gráficos;
- e) Permitir maior personalização do Dashboard, como seleção de cores e modelos de tela pré-definidos;
- f) Otimização da interface gráfica do usuário, trazendo maior interatividade e tornando-a mais intuitiva.

DASHGEN

Referencias

ACADEMIAIN. **QLIK SENSE: O QUE É, COMO FUNCIONA E QUAIS AS VANTAGENS?** Disponível em: <<https://blog.academaiain1.com.br/qlik-sense-o-que-e-como-funciona-e-quais-as-vantagens/>>. Acesso em: 10 jan. 2020.

APACHE.ORG. **The Apache Velocity Project**. Disponível em: <<https://velocity.apache.org/>>. Acesso em: 7 jan. 2019a.

APACHE.ORG. **Changes Report - Apache Velocity**. Disponível em: <<https://velocity.apache.org/engine/devel/changes.html>>. Acesso em: 22 ago. 2019b.

APACHE.ORG. **Version History - Apache Freemarker**. Disponível em: <https://freemarker.apache.org/docs/app_versions.html>. Acesso em: 25 ago. 2019c.

APACHE.ORG. **What is Maven**. Disponível em: <<https://maven.apache.org/what-is-maven.html>>. Acesso em: 1 ago. 2019d.

APACHE.ORG. **Apache Commons CSV**. Disponível em: <<https://commons.apache.org/proper/commons-csv/index.html>>. Acesso em: 2 nov. 2019e.

DASHGEN

Referencias

APACHE.ORG. **Apache Commons IO**. Disponível em:
<<https://commons.apache.org/proper/commons-io/>>. Acesso em: 10 nov. 2019f.

APACHE.ORG. **Apache Netbeans**. Disponível em: <<https://netbeans.org/>>.
Acesso em: 12 jan. 2019g.

APACHE.ORG. **FreeMarker Java Template Engine**. Disponível em:
<<https://freemarker.apache.org/index.html>>. Acesso em: 1 jun. 2019.

ASTAH. **Astah UML**. Disponível em: <<http://astah.net/editions/uml-new>>. Acesso em: 10 jan. 2020.

BAELDUNG.COM. **Introduction to Apache Velocity**. Disponível em:
<<https://www.baeldung.com/apache-velocity>>. Acesso em: 7 jan. 2019.

BERGEN, J. VAN. **Velocity or FreeMarker? Two open source Java-based template engines compared**. Disponível em:
<<https://www.javaworld.com/article/2077797/open-source-tools/velocity-or-freemarker.html>>. Acesso em: 29 ago. 2018.

BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. 3. ed. Rio de Janeiro: Elsevier, 2015.

DASHGEN

Referencias

BOSTOCK, M. et al. **D3.js - Data Driven Documents**. Disponível em: <<https://d3js.org/>>. Acesso em: 10 nov. 2018.

COSTA, H. M. K. et al. Grandes Massas de Dados na Nuvem: Desafios e Tecnicas para Inovação. **Sbrc 2012**, n. Ouro Preto, MG, Brasil, 2012.

CROSSFILTER ORGANIZATION. **Crossfilter js library**. Disponível em: <<https://github.com/crossfilter/crossfilter>>. Acesso em: 2 nov. 2019.

CRUZ, S. A. B.; MOURA, M. F. **Formatação de Dados Usando a Ferramenta Velocity**Campinas, 2002.

JETBRAINS. **IntelliJ IDEA**. Disponível em: <<https://www.jetbrains.com/idea/>>. Acesso em: 12 jan. 2020.

KRUEGER, C. W. Software reuse. **ACM Computing Surveys**, v. 24, n. 2, p. 131–183, 1992.

LUCRÉDIO, D. Uma Abordagem Orientada a Modelos para Reutilização de Software. p. 277, 2009.

DASHGEN

Referencias

LUIZ, A. **Visualização dos dados estatísticos da UERJ : proposta de dashboards baseados no trabalho de Jacques Bertin.** [s.l.] UERJ, 2013.

MAVENREPOSITORY. **Mavenrepository Apache Velocity.** Disponível em: <<https://mvnrepository.com/artifact/org.apache.velocity/velocity>>. Acesso em: 10 fev. 2019a.

MAVENREPOSITORY. **Mavenrepository Apache Freemarker.** Disponível em: <<https://mvnrepository.com/artifact/org.freemarker/freemarker>>. Acesso em: 10 jan. 2019b.

MICROSOFT. **O que é o PowerBI desktop.** Disponível em: <<https://docs.microsoft.com/pt-br/power-bi/desktop-what-is-desktop>>. Acesso em: 12 out. 2019.

MOURA, M. F. et al. **Comunicado Técnico Uma Análise Comparativa das Soluções Tecnológicas Utilizadas nas Apresentações de Dados da Agência de Informação Embrapa.** [s.l.] Embrapa, 2004.

ORACLE. **Oracle - JavaFX Overview(Release 8).** Disponível em: <<https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm>>. Acesso em: 2 dez. 2019.

POTENCIER, F. **The flexible, fast, and secure template engine for PHP.** Disponível em: <<https://twig.symfony.com/>>. Acesso em: 10 jan. 2020.

DASHGEN

Referencias

QLIK. **Qlik Sense - Plataforma de análise de dados**. Disponível em: <<https://www.qlik.com/pt-br/products/qlik-sense>>. Acesso em: 10 jan. 2020.

RAUDJÄRV, R. **ZT-ZIP**. Disponível em: <<https://github.com/zeroturnaround/zt-zip>>. Acesso em: 10 nov. 2019.

SHIMABUKURO JUNIOR, E. K. Um Gerador de aplicações configurável. 2006.
SOMATIVA. **Tableau Desktop**. Disponível em: <<http://www.somativa.com.br/tableau-desktop>>. Acesso em: 12 out. 2019.

SOMMERVILLE, I. **Engenharia de Software**. 3. ed. Sao Paulo: Pearson, 2013.

SYRIANI, E.; LUHUNU, L.; SAHRAOUI, H. Systematic mapping study of template-based code generation. **Computer Languages, Systems and Structures**, v. 52, p. 43–62, 2018.

TABLEAU. **Tableau Desktop**. Disponível em: <<https://www.tableau.com/pt-br/products/desktop>>. Acesso em: 12 dez. 2019.

TEAM DC.JS. **dc.js - Dimensional Charting Javascript Library**. Disponível em: <<https://dc-js.github.io/dc.js/>>. Acesso em: 10 nov. 2019.

TUTORIALSPPOINT. **Tutorialspoint - DC.js tutorials**. Disponível em: <<https://www.tutorialspoint.com/dcjs/>>. Acesso em: 11 nov. 2018.

DASHGEN

Referencias

WIKIPEDIA.ORG. **Apache FreeMarker**. Disponível em:
<https://en.wikipedia.org/wiki/Apache_FreeMarker>. Acesso em: 25 nov. 2018.

WIKIPEDIA.ORG. **Apache Velocity**. Disponível em:
<https://en.wikipedia.org/wiki/Apache_Velocity>. Acesso em: 10 fev. 2019a.

WIKIPEDIA.ORG. **Apache Maven**. Disponível em:
<https://pt.wikipedia.org/wiki/Apache_Maven>. Acesso em: 22 ago. 2019b.

WIKIPEDIA.ORG. **JavaFX**. Disponível em: <<https://en.wikipedia.org/wiki/JavaFX>>. Acesso em: 11 nov. 2019c

WIKIPEDIA.ORG. **MVC**. Disponível em: <<https://pt.wikipedia.org/wiki/MVC>>. Acesso em: 11 dez. 2019d.

WIKIPEDIA.ORG. **JAR (file format)**. Disponível em:
<[https://en.wikipedia.org/wiki/JAR_\(file_format\)](https://en.wikipedia.org/wiki/JAR_(file_format))>. Acesso em: 10 jan. 2020e.

WIKIPEDIA.ORG. **Template Processor**. Disponível em:
<https://en.wikipedia.org/wiki/Template_processor>. Acesso em: 6 fev. 2020.

XNAT.ORG. **Apache Velocity Cheatsheet**. Disponível em: <<https://wiki.xnat.org/docs16/4-developer-documentation/xnat-codex/velocity-cheat-sheet>>. Acesso em: 7 ago. 2019.