INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PARÁ

CURSO DE GRADUAÇÃO TECNOLÓGICA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GLAUBER MATTEIS GADELHA

**Gerador de quadros de apresentação de dados em formato de Dashboards**

BELÉM

2019

GLAUBER MATTEIS GADELHA

**Gerador de quadros de apresentação de dados em formato de Dashboards**

Trabalho de Conclusão de Curso apresentado à coordenação do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Pará para obtenção de Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Claudio Roberto de Lima Martins

BELÉM

2019

**RESUMO**

No cenário mundial, a demanda por sistemas digitais cresce constantemente, exigindo alinhamento com as tecnologias mais atuais e uso de ferramentas que acelerem a entrega de aplicações no prazo mais curto possível. No desenvolvimento de softwares, independente da linguagem de programação utilizada, a necessidade de implementação de grandes e complexos blocos de códigos de forma recorrente é presente no dia a dia dos profissionais, portanto, qualquer ferramenta que facilite a aplicação destes blocos padronizados, com as modificações necessárias para o domínio de um problema determinado, pode vir a evitar o desperdício de horas de programação. Em face destas premissas, este trabalho é um estudo de reuso de software e geração de código fonte, mais especificamente focado no desenvolvimento de camada de apresentação de dados em formato gráfico, a que chamamos comumente de Dashboards.

Palavras - chave: Desenvolvimento de aplicações, apresentação de dados, dashboards, geradores de código.

**ABSTRACT**

In actual world scenary, the digital systems demand grows constantly, needing compliance with the most efficient and new technologies and the use of software development tools that speeds up the applications deployment in the shortest schedule. In software development, language independently, the need of coding large and complex blocks recurrently it’s usual to IT professionals day by day, so any tool that reduces the workload of coding this standarized blocks can save several coding hours. On the face of this premisses, this research work aims to study software reuse and source code generating, most specifically focused on Data Presentation Layer on graphic format, usually called Deshboards.

Keywords: Software development, data presentation, dashboards, code gerenation.

Sumário

[1 INTRODUÇÃO 6](#_Toc535170427)

[1.2 Objetivo 7](#_Toc535170428)

[1.3 Procedimentos metodológicos 7](#_Toc535170429)

[1.4 Organização do trabalho 8](#_Toc535170430)

[2 Fundamentação Teórica 8](#_Toc535170431)

[2.1 Reuso de Software 8](#_Toc535170432)

# INTRODUÇÃO

A evolução tecnológica, além de muitas vantagens e confortos, nos trouxe algo que ao é mesmo tempo valioso e assustador: o acúmulo minuto a minuto de massas de dados cada vez maiores. Cientistas da Tecnologia da Informação, Matemáticos e estatísticos vem trabalhando em sofisticadas técnicas voltadas a encontrar padrões nessas massas de dados, transformando caos em informação útil.

Ainda assim, os volumes são tamanhos que inviabilizam o entendimento de certos relatórios por um executivo, um investidor ou um cliente. Para minimizar o problema, tem se empregado formas de apresentação de informações sintetizadas em quadros gráficos, que entregam um resumo visual de centenas, muitas vezes milhares de linhas de uma tabela estruturada. Isso facilita o entendimento e suporta de forma mais simples e direta a tomada de decisão.

Mesmo com as linguagens de programação, como Java, C#, entre outras terem evoluído e expandido sua biblioteca de Interfaces de Aplicações para Programação, comumente chamadas APIs, ainda é trabalhoso desenvolver a camada de apresentação de dados de forma gráfica e dinâmica. Uma página JSP ou ASP contendo um dashboard que sintetiza em uma tela um conjunto de informações de um relatório de pesquisa em banco de dados requer algumas centenas de linhas de código.

A demanda de mercado por entregas no mais curto prazo possível, bem como a necessidade de se desenvolver artefatos que permitam fácil entendimento, documentação, manutenção e evolução, além da importância de uma apresentação clara e simplificada dos dados, por si só, sustentam a necessidade de pesquisa no campo da geração rápida e eficiente de relatórios gráficos. Isto, se feito apropriadamente, reduz a quantidade de falhas de projeto e deixa a equipe de programadores e analistas mais livres para tratar regras de negócio e complexidades típicas de cada domínio de problema. (LUCRÉDIO, 2009).

Poder utilizar-se de bibliotecas de classes ou até mesmo programas standalone que gerem código a partir de um conjunto de definições feitas previamente pelo usuário, o que pode ser chamado de um assistente ou wizard, reduz uma etapa trabalhosa e significante do esforço total de desenvolvimento de um sistema completo ou de módulos de um sistema maior. (SHIMABUKURO JUNIOR, 2006).

## 1.2 Objetivo

O objetivo deste trabalho é desenvolver uma aplicação em linguagem Java para desktop usando técnicas de programação generativa orientada a gabaritos (no inglês, *templates*) que facilite a qualquer usuário comum com um arquivo CSV contendo um conjunto de dados, a geração de painéis de informações gerenciais (*Dashboards*) para representação destes dados. Os gabaritos são codificados usando linguagem de template Apache FreeMarker e os painéis gerados são codificados em HMTL5 usando CSS3 e as bibliotecas Javascript DC.js, Crossfilter.js e D3.js.

## 1.3 Procedimentos metodológicos

O trabalho foi conduzido em três etapas. Em primeiro lugar, foi feita investigação de literatura e trabalhos acadêmicos a fim fundamentar teoricamente a aplicação, abordando a geração automática de código fonte e suas vantagens, aplicação de linguagens de template, além de técnicas e ferramentas para apresentação de dados em formato dashboard em aplicações web.

Na etapa seguinte, foi feita a análise e modelagem da aplicação, utilizando-se de algumas técnicas já consolidadas de engenharia de software.

Na etapa final, a aplicação foi desenvolvida aplicando as tecnologias e métodos investigados nas etapas anteriores, bem como testada e validada.

## 1.4 Organização do trabalho

Este trabalho de conclusão de curso foi organizado como segue descrito abaixo:

No capítulo 2 é descrita toda a fundamentação teórica e técnica para o trabalho. Compreende as vantagens do reuso de software, técnicas de apresentação de dados, a descrição resumida do histórico e vantagens da linguagem de template Apache FreeMarker e as bibliotecas DC.js, Crossfilter.js e D3.js, responsáveis por apresentar os dados de forma gráfica e dinâmica em páginas HTML5.

No capítulo 3 descreve-se o projeto da aplicação geradora de dashboards e seu desenvolvimento. Por último, no capítulo 4 são feitas as considerações finais.

# Fundamentação Teórica

Este capítulo apresenta a base teórica sobre reuso de software e apresentação de dados em formato gráfico dashboard. Além disso, fundamenta tecnicamente as bibliotecas e linguagens utilizadas no desenvolvimento da aplicação final.

## 2.1 Reuso de Software

Uma das definições encontradas na literatura pesquisada para desenvolvimento deste trabalho para o reuso de software é o processo de se criar software a partir de software existente, ao invés de simplesmente construí-lo do início (LUCRÉDIO, 2009). Qualquer dos artefatos de software de um projeto anterior e bem sucedido, como código fonte, classes e bibliotecas compiladas, planos, estratégias, diagramas, entre outros, podem e devem ser reutilizados para agilizar o desenvolvimento de outros artefatos ou sistemas. (SHIMABUKURO JUNIOR, 2006).

Apesar do reuso de software parecer a solução ideal e mais óbvia para entrega rápida de soluções para as demandas de mercado, até os dias atuais são poucos os times de Análise e Desenvolvimento de Sistemas que aplicam o conceito de forma sistemática.

Lucrédio (2009, p. 30) ressalta em sua tese que reuso de software remonta de 1947, no início da programação armazenada, quando Wheeler e Wilkes desenvolveram o conceito de *jump*, um precursor do comando *goto*, que possibilitava reaproveitar blocos de código dentro do mesmo programa. Desta data adiante, programadores reaproveitam blocos de código em arquivos pessoais, programas antigos, repositórios públicos e até mesmo em sua memória.

Para Sommerville ( 2013, p.297), apesar do potencial naturalmente reusável de componentes e sistemas de software, pode se tornar muito cara sua readaptação para aplicação em um novo domínio. Reaproveitar artefatos devidamente testados e validados certamente pode trazer benefícios na redução do custo total do desenvolvimento de um sistema, além de outras vantagens. Porém algumas dificuldades inerentes também a este reaproveitamento podem mostrar que essa redução de custo pode não ser tão grande quanto se espera.

## 2.2 Geração de Código

## 2.3 Motores de template

## 2.3.1 Java Server Pages

## 2.3.3 Apache Velocity

## 2.3.4 Apache Freemarker

# 3 - Desenvolvimento do Protótipo

Neste capítulo é apresentado o desenvolvimento da ferramenta de geração de dashboards, utilizando práticas de análise orientada a objetos, além de demonstrar exemplos de aplicação e os resultados obtidos.

## 3.1 - Descrição Geral

Partindo de uma especificação de um arquivo de intercambio de dados CSV, a aplicação IFPA DashGen permite gerar um Dashboard contendo uma tabela e 3 tipos de gráficos, sendo 1 gráfico de área, 1 gráfico de barras e 1 gráfico pizza, com filtros dinâmicos que permitem seleção das dimensões pelo usuário, assim como redefinição para a visão inicial.

O funcionamento da aplicação se dá como segue descrito abaixo:

1. O usuário indica ao DashGen qual o arquivo fonte de dados;
2. O DashGen captura a primeira linha do arquivo fonte de dados, considerando que esta traga os nomes dos atributos contidos;
3. O usuário especifica o domínio (eixo X) e a imagem(Eixo Y) de cada tipo de gráfico, bem como o título e as etiquetas dos atributos de domínio e imagem;
4. O usuário especifica o caminho onde o arquivo contendo a estrutura do Dashboard deve ser armazenado;
5. O DashGen instancia os Templates incluindo os dados especificados, gerando um Dashboard completo.
6. O DashGen salva um arquivo compactado em formato ZIP na pasta especificada pelo usuário, que pode ser descompactado no caminho da aplicação que este desejar.

(*fazer figura ilustrando os passos descritos)*

## 3.2 Descrição das Etapas de Desenvolvimento

O trabalho de desenvolvimento da aplicação IFPA DashGen foi dividido nos seguintes passos:

1. Levantamento dos requisitos;
2. Especificação do uso do motor de Templates, baseado nos diagramas UML de caso de uso, de classes e de sequencia;
3. Implementação do motor de Templates;
4. Implementação do gerador de Dashboard.

## 3.3 – Levantamento dos Requisitos

A aplicação produzida neste trabalho, a partir das especificações de entrada, será capaz de gerar um Dashboard completo. As entradas são: o arquivo de dados, os arquivos de template, os atributos de dimensão, o título do arquivo e o caminho para armazenamento do Dashboard gerado.

Baseado nisso, foi feito o levantamento de requisitos a serem atendidos, fechando o escopo da aplicação. No quadro 1, podem ser observados os requisitos funcionais e no quadro 2, podem ser observados os requisitos não-funcionais.

|  |  |  |
| --- | --- | --- |
| **Requisitos Funcionais** | | |
| **Requisito** | **Descrição** | **Caso de Uso** |
| RF01 | O sistema deve acessar o arquivo de dados CSV | UC01 |
| RF02 | O sistema deve identificar os metadados do arquivo de dados | UC01 |
| RF03 | O sistema deve permitir que o usuário especifique o arquivo de dados | UC01 |
| RF04 | O sistema deve permitir que o usuário especifique o caminho onde o Dashboard deve ser armazenado | UC01 |
| RF05 | O sistema deve gerar o Dashboard, compactando toda a estrutura em um arquivo ZIP, no caminho especificado pelo usuário. | UC01 |

Quadro 1 – Requisitos Funcionais

|  |  |
| --- | --- |
| **Requisitos não funcionais** | |
| **Requisito** | **Descrição** |
| RNF01 | Sistema deve ser desenvolvido em Java 8. |
| RNF02 | Sistema deve ter interface gráfica simples e intuitiva, com uso de mouse, tendo todas as informações exibidas com clareza e ter funcionalidades acessíveis por botões e menus. |
| RNF03 | O desempenho da geração do Dashboard deve ser alto. O tempo de geração do arquivo destino não deve exceder 30 segundos. |
| RNF04 | O padrão do arquivo de dados deve ser CSV, com a primeira linha contendo a descrição dos atributos. |
| RNF04 | A linguagem de templates a ser aplicada será a Apache FreeMarker. |
| RNF05 | O formato de compactação do arquivo de saída será ZIP. |

Quadro 2 – Requisitos Não Funcionais

## 3.4 – Casos de Uso

Na interação com o usuário, a aplicação terá um único caso de uso, como pode ser verificado no diagrama de casos de uso.



Figura 1 – Caso de uso UC01

A descrição do caso de uso UC01 – Gerar\_Dashboard foi detalhada considerando o processo completo de geração do Dashboard do ponto de vista do usuário da aplicação. Este detalhamento pode ser observado no Quadro 3.

|  |  |
| --- | --- |
| **UC01 – Caso de uso Gerar\_Dashboard:** Permite ao usuário especificar o arquivo de dados, os atributos a serem usados como dimensões nos gráficos e o caminho para a geração do arquivo compactado com o Dashboard completo. | |
| Pré-condições: | Usuário executa a aplicação Dashgen. |
| Cenário Principal: | 1. Ao acessar o sistema, o usuário informa o caminho para o arquivo CSV; 2. O sistema apresenta o primeiro gráfico e solicita que sejam indicadas as dimensões, apresentando os atributos obtidos da primeira linha do CSV; 3. O sistema segue para a apresentação do segundo gráfico e solicita que sejam indicadas as dimensões, apresentando atributos obtidos do CSV; 4. O sistema então apresenta o terceiro gráfico, solicitando novamente as dimensões a partir dos atributos obtidos do CSV; 5. O caso de uso termina no momento em que o usuário faz o download do arquivo ZIP gerado pelo sistema. |
|  |  |
|  |  |