



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Eduardo Chagas de Oliveira

**Uma Proposta de Metamodelo de Visualização da Informação,
Baseada em Treemaps**

UBERLÂNDIA – MINAS GERAIS
2015

Eduardo Chagas de Oliveira

**Uma Proposta de Metamodelo de Visualização da Informação,
Baseada em Treemaps**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como requisito parcial para obtenção do título de Mestre em Ciências.

Área de Concentração: Visualização da Informação

Orientador: Prof. Dr. Alexandre Cardoso

Coorientador: Prof. Dr. Edgard Lamounier

UBERLÂNDIA – MINAS GERAIS

2015

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

O48p Oliveira, Eduardo Chagas de, 1982-
2015 Uma proposta de metamodelo de visualização da informação,
baseada em Treemaps / Eduardo Chagas de Oliveira. - 2015.
 102 f. : il.

Orientador: Alexandre Cardoso.
Coorientador: Edgard Afonso Lamounier Júnior.
Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.

1. Engenharia elétrica - Teses. 2. Visualização da informação -
Teses. I. Cardoso, Alexandre, 1964- II. Lamounier Júnior, Edgard
Afonso, 1964- III. Universidade Federal de Uberlândia. Programa de
Pós-Graduação em Engenharia Elétrica. IV. Título.

CDU: 621.3

Eduardo Chagas de Oliveira

Uma Proposta de Metamodelo de Visualização da Informação, Baseada em Treemaps

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciências e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia.

Área de concentração: Visualização da Informação

Banca Examinadora:

Prof. Dr. Alexandre Cardoso

Orientador

Prof. Dr. Edgard Afonso Lamounier Júnior

Co-orientador

Prof. Dr. Keiji Yamanaka

Universidade Federal de Uberlândia

Profa. Dra. Veronica Teichrieb

Universidade Federal de Pernambuco

Uberlândia, abril de 2015

Resumo

Interpretar e entender uma grande quantidade de dados é um desafio dos modelos de Visualização de Informação. A Visualização da Informação é importante para possibilitar uma apropriada obtenção da informação através da visualização. A escolha do modelo de visualização de informação mais apropriado antes de iniciar a resolução de um determinado problema visual é primordial para obtenção de uma solução eficaz. A implementação do modelo envolve escolher também componentes e a arquitetura que suporte as suas funcionalidades necessárias, de maneira eficaz. Esta dissertação descreve uma abordagem de classificação das visualizações da informação baseado em *treemap*, capaz de buscar e identificar o melhor modelo de visualização da informação para um determinado problema. Isso compreende na criação de um metamodelo de Visualização da Informação adequado. Primeiramente, o atual estado do campo de visualização foi descrito e então, são mostradas as regras e critérios utilizados na pesquisa dessa dissertação, a fim de apresentar uma proposta de metamodelo baseado em *treemap* para modelos de visualização. Além disso, é apresentado um estudo de caso com as informações contidas no metamodelo de visualização da tabela periódica e a análise da complexidade de tempo da busca da informação em cada um dos dois metamodelos. Finalmente, uma avaliação dos resultados é apresentada, através de experimentos com usuários e análise comparativa entre os modelos baseados em *treemap* e na tabela periódica.

Palavras-chaves: visualização da informação, modelos da visualização da informação, sistemas de visualização, tipos de visualização, resolução de problemas visuais, *treemap*.

Abstract

The interpretation and understanding of large quantities of data is a challenge for current information visualization methods. The visualization of information is important as it makes the appropriate acquisition of the information through the visualization possible. The choice of the most appropriate information visualization method before commencing with the resolution of a given visual problem is primordial to obtaining an efficient solution. This article has as its objective to describe an information visualization classification approach based on Treemap, which is able to identify the best information visualization model for a given problem. The implementation of the model involves choosing components and also the architecture that stands its necessary functions effectively. This is understood through the construction of an adequate information visualization meta-model. Firstly, the actual state of the visualization field is described, and then the rules and criteria used in our research are shown, with the aim of presenting a meta-model proposal based on treemap visualization methods. Besides this, the author present a case study with the information contained in the periodic table visualization meta-model along with an analysis of the information search time complexity in each of the two meta-models. Finally, an evaluation of the results is presented through the experiments conducted with users and a comparative analysis of the methods based on Treemap and the Periodic Table.

Keywords: information visualization, information visualization models, visualization systems, visualization types, visualization problem solving, treemap.

Aos meus amados pais, Terezinha das Chagas
Pereira e Jair João de Oliveira.

Agradecimentos

Primeiramente, agradeço a Deus, por tudo de bom que eu vivenciei e pelo percurso traçado em minha vida, junto à sua presença.

Ao meu orientador, Prof. Dr. Alexandre Cardoso e coorientador Prof. Dr. Edgard Afonso Lamounier Júnior, pela disponibilidade e acompanhamento exercido durante a construção da minha dissertação e pelos esclarecimentos necessários para a obtenção de novos conhecimentos. Agradeço pela constante orientação e apontamentos que contribuíram no meu crescimento profissional e pessoal. Seus ensinamentos foram fundamentais para a realização dessa dissertação.

Agradeço também a todos meus mestres, que ajudaram neste período de aprendizagem.

Agradeço ao Prof. Dr. Keiji Yamanaka e Veronica Teichrieb pela participação na banca examinadora, pelas contribuições e correções realizadas nesta dissertação.

Aos funcionários do departamento de Engenharia Elétrica da Universidade Federal de Uberlândia, que sempre me recepcionaram muito bem.

Ao engenheiro Leonardo Borges, presidente da empresa Datawin, pela oportunidade de aplicar a teoria em atividades práticas em um ambiente de trabalho profissional, pelos desafios repassados e confiança na solução dos mesmos.

À Freeman, à Cemig, à Smartscan e à Fundação Fisk por criar novos desafios práticos e pela oportunidade de participação em seus projetos.

Ao CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e a CGW (Computer Graphics Works) pelo apoio financeiro no projeto GT411.

À minha irmã Luciene Chagas de Oliveira, pelo incentivo, apoio e companheirismo. Agradeço à sua ajuda e dedicação nos dias de estudo e trabalho que passamos juntos.

Aos meus pais Terezinha das Chagas Pereira e Jair João de Oliveira, e a minha irmã, Liliane Chagas de Oliveira, que mesmo à distância, mostraram-me apoio e solidariedade.

À minha namorada Lara de Oliveira Gonçalves pelo apoio e carinho.

Aos meus colegas de trabalho por todo esforço, ajuda e companheirismo neste momento da minha vida.

A todas as pessoas queridas, amigos e parentes, que fazem parte da minha vida e a tornam completa;

A todos os meus colegas do laboratório de Engenharia Biomédica da Universidade Federal de Uberlândia, por me proporcionarem um maravilhoso ambiente de trabalho e por todo o apoio durante este período de convivência. Guardo com carinho ótimas recordações de todos vocês.

A todos que colaboraram de forma direta ou indireta na elaboração e construção desta dissertação.

Publicações Obtidas

Artigo Aceito em Congresso e Ainda Não Apresentado

- ALEXANDRE, C.; LAMOUNIER, E.A.J.; OLIVEIRA, E.C.; OLIVEIRA, L.C.; MATTIOLI, L. **Meta-model of Information Visualization Based on Treemap.** In: The 3rd World Conference on Information Systems and Technologies (WorldCIST'15), Portugal, 2015.

Trabalho Completo Publicado em Anais de Congresso

- ALEXANDRE, C.; LAMOUNIER, E.A.J.; OLIVEIRA, E.C.; OLIVEIRA, L.C. **A proposal for a Meta-Information Visualization using Treemap.** In: The International Conference on Computational Science and Computational Intelligence (CSCI), 2014, v. 1, p. 247-252. ISBN 978-1-4799-3009-8, doi 10.1109/CSCI.2014.49.
- ALEXANDRE, C.; LAMOUNIER, E.A.J.; OLIVEIRA, E.C.; OLIVEIRA, L.C. **Performance in Search Data Methods and Components for Information Visualization.** In: VI Congresso Tecnológico InfoBrasil, Fortaleza, 2014.

Apresentação de Trabalho

- ALEXANDRE, C.; LAMOUNIER, E.A.J.; OLIVEIRA, E.C.; OLIVEIRA, L.C. **Measure Performance of Information Visualization.** In: X Workshop de Visão Computacional, Uberlândia, 2014.

Sistemas Publicados na WEB

- Visualization Information. Version 9.0. Disponível em: <<http://www.visualization.information.cgwusa.com>>. Acesso em: 25 set. 2013.
- Projeto SGF, Sistema de Gestão Fisk. Versão 1.0. Disponível em: <<http://www.sgf.datawin.com.br>> (Ambiente de produção da Datawin), <<http://www.fisk.com.br/sgf>> (Ambiente de produção do Fisk) e

<<http://www.fisk.com.br/sgfhomolog>> (Ambiente de homologação do Fisk). Acesso em: 27 out. 2014.

- Projeto CGW da Freeman. Versão 1.0. Disponível em: <<http://www.cgwusa.com>>. Acesso em: 01 mar 2014.
- Projeto GT411 da Cemig. Versão 1.0. Disponível em: <<http://www.gt411.cgwusa.com>>. Acesso em: 01 mar 2014.

“A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.”

Arthur Schopenhauer

Sumário

RESUMO	I
ABSTRACT	II
<i>Artigo Aceito em Congresso e Ainda Não Apresentado</i>	<i>vi</i>
<i>Trabalho Completo Publicado em Anais de Congresso</i>	<i>vi</i>
<i>Apresentação de Trabalho</i>	<i>vi</i>
<i>Sistemas Publicados na WEB</i>	<i>vi</i>
SUMÁRIO.....	1
LISTA DE FIGURAS	3
LISTA DE TABELAS	6
LISTA DE SIGLAS	7
INTRODUÇÃO E JUSTIFICATIVA.....	8
1.1 OBJETIVO PRINCIPAL.....	13
1.2 OBJETIVOS ESPECÍFICOS.....	14
1.3 ORGANIZAÇÃO DESTA DISSERTAÇÃO	17
FUNDAMENTAÇÃO TEÓRICA	19
2.1 CRITÉRIOS PARA CLASSIFICAÇÃO DA VISUALIZAÇÃO DA INFORMAÇÃO	19
2.1.2 Outros Critérios para Taxonomia da Visualização da Informação	20
2.2 TEORIA DA ANÁLISE DE ALGORITMOS	26
2.3 CONCEPÇÃO DA TREEMAP.....	29
TRABALHOS CORRELATOS.....	32
3.1 PESQUISA DE EFICÁCIA NA VISUALIZAÇÃO DA INFORMAÇÃO.....	32
3.2 PROPOSTAS DE CLASSIFICAÇÃO DE VI	34
3.3 TABELA PERIÓDICA DA VISUALIZAÇÃO DA INFORMAÇÃO.....	37
PROPOSTA DE METAMODELO PARA VISUALIZAÇÃO DA INFORMAÇÃO	41
4.1 TREEMAP DA VISUALIZAÇÃO DA INFORMAÇÃO	41
4.2 METODOLOGIA E DETALHES DA IMPLEMENTAÇÃO	45
4.3 VISÃO GERAL DA ANÁLISE DE ALGORITMOS.....	46
4.3.1 Análise de Algoritmos de Visualização	46

4.3.2 <i>Análise de Algoritmos de Visualização dos Metamodelos</i>	50
4.3.3 <i>Análise de Resultados de Testes</i>	51
COMPONENTES PARA VISUALIZAÇÃO DA INFORMAÇÃO EM SISTEMAS WEB	57
5.1 MODELO DE PESQUISA EM LISTA DE DADOS	58
5.1.1 <i>Componente de Multiselectão</i>	59
5.1.2 <i>Componente de Paginação</i>	59
5.1.3 <i>Componente Gráfico em Hierarquia</i>	63
5.2 MODELO TABLE	64
5.2.1 <i>Componente Gráfico de Horários</i>	64
5.2.2 <i>Componente Gráfico de Relatórios</i>	66
5.3 MODELO TREEMAP	70
5.3.1 <i>Componente Gráfico da Treemap</i>	70
5.4 ANÁLISE DE DESEMPENHO DOS COMPONENTES	70
ARQUITETURA DO SISTEMA WEB	75
6.1 PROPOSTA ARQUITETURAL.....	75
6.2 ELABORAÇÃO DA ARQUITETURA PROPOSTA.....	77
DISCUSSÃO E CONCLUSÕES	91
7.1 DESEMPENHO NA VISUALIZAÇÃO DA INFORMAÇÃO.....	91
7.2 CONTRIBUIÇÕES.....	93
7.3 TRABALHOS FUTUROS	93
REFERÊNCIAS	94
CONCEITOS DE ARQUITETURA DE SISTEMA.....	100

Listas de Figuras

FIGURA 1. DIAGRAMA ESQUEMÁTICO DO PROCESSO DE VISUALIZAÇÃO. (WARE, 2004, P. 4) ⁹	9
FIGURA 2. OBJETIVOS PRINCIPAIS DA DISSERTAÇÃO EM FORMA DE MAPA MENTAL.....	16
FIGURA 3. AS SETE VARIÁVEIS VISUAIS DE JACQUES BERTIN (BERTIN, 1983 APOUD BRADLEY, 2013).....	20
FIGURA 4. CRITÉRIO GRÁFICO SEGUNDO CLEVELAND E MCGILL'S. (CAILLETEAU, 1999, P. 43).....	22
FIGURA 5. ACUIDADE DE PERCEPÇÃO DOS ATRIBUTOS VISUAIS, SEGUNDO MACKINLAY.	24
FIGURA 6. DRILLDOWNUP DE “CONCEPT VISUALIZATION”.....	29
FIGURA 7. EXEMPLO DE ESTRUTURA DE TREEMAP.	30
FIGURA 8. DRILLDOWNUP DE “CONCEPT VISUALIZATION – DETAIL AND OVERVIEW”.....	30
FIGURA 9. EXEMPLO DE DECOMPOSIÇÃO DE MODELO POR CRITÉRIOS DE VISUALIZAÇÃO. (WINCKLER; PALANQUE; FREITAS, 2004).....	34
FIGURA 10. PROTÓTIPO PROPOSTO POR MATUSI PARA ESCOLHA DOS CRITÉRIOS NA CRIAÇÃO DOS MODELOS. (MATUSI; YAMANOUCHI; SUNAHARA, 2011)	37
FIGURA 11. TABELA PERIÓDICA DOS MODELOS DE VISUALIZAÇÃO (LENGLER; EPPLER, 2007).....	39
FIGURA 12. TREEMAP PARA VISUALIZAÇÃO DA INFORMAÇÃO.....	44
FIGURA 13. BUSCA BINÁRIA VISUAL ATRAVÉS DE DADOS EM LISTA ORDENADA E SEM ORDENAÇÃO.....	47
FIGURA 14. COMPARAÇÃO REPRESENTATIVA DE INFORMAÇÕES DOS RESULTADOS DAS NOTAS DE ALUNOS (CARDOSO, 2014).....	48
FIGURA 15. ANALISE DE ALGORITMO DA “STRATEGY VISUALIZATION” DA “PERIODIC TABLE” (LENGLER; EPPLER, 2007).....	49
FIGURA 16. ANÁLISE DE ALGORITMO DA STRATEGY VISUALIZATION DA TREEMAP.	50
FIGURA 17. RESULTADO DOS TESTES – RESPOSTA DO QUESTIONÁRIO.....	52
FIGURA 18. COMPARAÇÃO ENTRE OS METAMODELOS TREEMAP E PERIODIC TABLE – PERCENTUAL DE ERROS POR PERGUNTA.....	53

FIGURA 19. COMPARAÇÃO ENTRE OS METAMODELOS TREEMAP E <i>PERIODIC TABLE</i> – PERCENTUAL DE ERROS POR CRITÉRIOS DE VISUALIZAÇÃO.....	54
FIGURA 20. COMPARAÇÃO ENTRE OS METAMODELOS TREEMAP E <i>PERIODIC TABLE</i> – TEMPO DE RESPOSTA POR PREGUNTA.....	55
FIGURA 21. COMPARAÇÃO ENTRE OS METAMODELOS TREEMAP E <i>PERIODIC TABLE</i> – TEMPO DE RESPOSTA POR CRITÉRIO.....	56
FIGURA 22. COMPONENTE DE MULTISELEÇÃO CUSTOMIZADO DO DOJO.	59
FIGURA 23. COMPONENTE DE PAGINAÇÃO SOB DEMANDA DO DOJO.	60
FIGURA 24. DIAGRAMA DE SEQUÊNCIA DO MODELO DE BUSCA PAGINADA DE DADOS SOB DEMANDA.....	62
FIGURA 25. COMPONENTE GRÁFICO EM HIERARQUIA CUSTOMIZADO DO DOJO.....	64
FIGURA 26. COMPONENTE GRÁFICO DE HORÁRIOS CUSTOMIZADO DO DOJO.....	65
FIGURA 27. COMPONENTE GRÁFICO DE RELATÓRIO DO DOJO.....	67
FIGURA 28. DIAGRAMA DE SEQUÊNCIA PARA O MODELO DE APRESENTAÇÃO DE RELATÓRIOS.	68
FIGURA 29. TELA DO RELATÓRIO DE POSIÇÃO FINANCEIRA DO SISTEMA WEB SGF.	69
FIGURA 30. RELATÓRIO DE POSIÇÃO FINANCEIRA DO SISTEMA WEB SGF.	69
FIGURA 31. RESULTADO DE TESTE DE DESEMPENHO DO COMPONENTE DE MULTISELEÇÃO. ..	71
FIGURA 32. RESULTADO DE TESTE DE DESEMPENHO DO COMPONENTE DE PAGINAÇÃO.	72
FIGURA 33. COMPONENTE DE TAG DO DOJO PARA REDUÇÃO DO ESPAÇO USADO DE FILTROS DE PESQUISA.	74
FIGURA 34. CAMADAS DE USO ESPECÍFICO DA APLICAÇÃO E USO ESPECÍFICO DE NEGÓCIO DO SISTEMA SGF.....	78
FIGURA 35. CAMADAS DE MIDDLEWARE, INFRA-ESTRUTURA E SISTEMA DO SISTEMA SGF.	79
FIGURA 36. CAMADAS DE USO ESPECÍFICO DA APLICAÇÃO E USO ESPECÍFICO DE NEGÓCIO DO SISTEMA CEMIG.....	80
FIGURA 37. CAMADAS DE MIDDLEWARE, INFRA-ESTRUTURA E SISTEMA DO SISTEMA CEMIG.	81
FIGURA 38. CAMADAS DE USO ESPECÍFICO DA APLICAÇÃO E USO ESPECÍFICO DE NEGÓCIO DO SISTEMA FREEMAN.....	82

FIGURA 39. CAMADAS DE MIDDLEWARE, INFRA-ESTRUTURA E SISTEMA DO SISTEMA FREEMAN.....	83
FIGURA 40. EXEMPLO DE DER DO SISTEMA WEB DA CEMIG.	89
FIGURA 41. EXEMPLO DE DIAGRAMA DE CLASSES DO SISTEMA WEB DA CEMIG.	90

Listas de Tabelas

TABELA 1. TABELA COM AS COMPLEXIDADES DE TEMPO DE ALGORITMOS (PUNTAMBEKAR, 2010, P. 1-27).....	27
TABELA 2. CRITÉRIOS QUE DEFINEM AS FORMAS DE VISUALIZAÇÃO NO METAMODELO TREEMAP.	42

Listas de Siglas

API	<i>Application Programming Interface</i>
BI	<i>Business Intelligence</i>
CNAB	Centro Nacional de Automação Bancária
CSS	<i>Cascading Style Sheets</i>
DER	Diagrama de Entidade e Relacionamento
FEBRABAN	Federação Brasileira de Bancos
HTTP	<i>Hypertext Transfer Protocol</i>
IIS	<i>Internet Information Service</i>
MVC	<i>Model View Controller</i>
RV	Realidade Virtual
SGF	Sistema de Gestão Fisk
TO	<i>Transfer Objects</i>
TTT	<i>Task by Data Type Taxonomy</i>
URL	<i>Universal Resource Locator</i>
VI	Visualização da Informação
WEB	<i>World Wide Web</i> – Rede de abrangência mundial

Capítulo 1

Introdução e Justificativa

Atualmente, a quantidade de dados disponíveis nos sistemas de informação é incrementada de forma contínua e considerável, sendo uma elevada quantidade de elementos de dados, podendo cada elemento ter os mais variados atributos. O mesmo ocorre em muitos domínios do conhecimento, fazendo com que as aplicações dos modelos tradicionais para a visualização e análise de dados se tornem insuficientes, complexos e inefficientes.

Scientific datasets are often difficult to analyse or visualise, due to their large size and high dimensionality.(...) These datasets are becoming more and more common; examples include scientific simulation results, geographic information systems, satellite images, and biomedical scans. The overwhelming amount of information contained in these datasets makes them difficult to analyse using traditional mathematical or statistical techniques. It also makes them difficult to visualise in an efficient or useful manner. (HEALEY, 2000)

A área da Visualização de Informação (VI) é uma disciplina que possui um crescente campo de pesquisa, que faz uso de um meio para converter a representação de alguma informação para que o usuário possa utilizá-la de uma maneira mais acessível.

Lengler e Eppler (2007) definem um método ou modelo de visualização como um processo sistemático, baseado em regras, representação externa, permanente e gráfica que retrata a informação de uma forma que é propícia à aquisição de conhecimentos, o desenvolvimento de uma compreensão elaborada, ou comunicar experiências.

A visualization method is a systematic, rule-based, external, permanent, and graphic representation that depicts information in a way that is conducive to acquiring insights, developing an elaborate understanding, or communicating experiences. (LENGLER; EPPLER, 2007)

A visualização fornece uma habilidade para compreender grande quantidade de informações e para resolver problemas com dados. Isso permite a percepção de padrões tais como características locais.

Segundo Colin Ware (2004), o processo de visualização inclui quatro estágios básicos, ilustrados na Figura 1:

- A coleção e armazenamento dos dados;
- O pré-processamento para transformar os dados em alguma informação que possa ser entendida;
- A visualização de informação através de algoritmos gráficos que produzem imagens na tela;
- O sistema cognitivo de percepção humana;

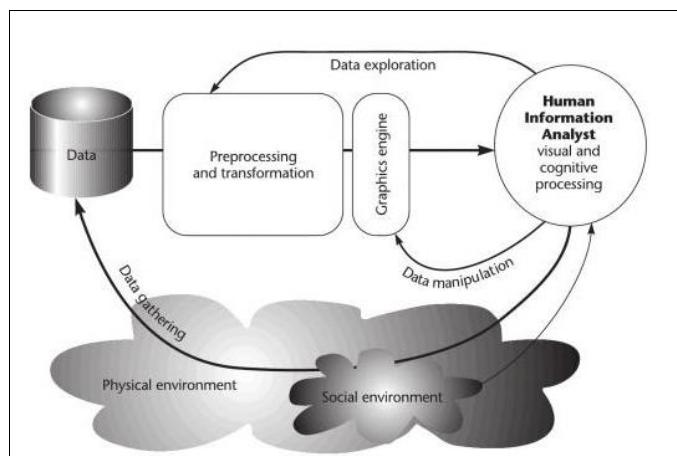


Figura 1. Diagrama esquemático do processo de visualização. (WARE, 2004, p. 4)

Segundo Gupta (2000), o cérebro humano, particularmente a sua percepção e cognição, é um dos modelos mais interessantes para o desenvolvimento de sistemas inteligentes.

The human brain, particularly its faculty for perception and cognition, is the most intriguing models for developing intelligent systems. (GUPTA, A. M.; GUPTA, M. M., 2000)

Um processo de VI deve buscar a melhor representação da informação para possibilitar a interpretação e o entendimento de uma grande quantidade de dados digitais que estão se tornando fáceis de adquirir com as novas tecnologias, por exemplo a API. Assim, em uma boa representação, a informação pode ser convertida em um conjunto de representações gráficas e deve ser aproveitada no entendimento da percepção humana.

Nowadays, any kinds of real-time and a large amount of digital information are becoming easy to acquire and utilize by the Internet revolution comparing with the past. In particular, API system that the users send a HTTP request to specific URL and get a HTTP response with information data sets, is used by many web developers because of it can be controlled by just programming scripts. These data sets are reflected to web application's user interface instantly then the users can get real-time information via the Internet. (MATUSI; YAMANOUCHI; SUNAHARA, 2011)

O objetivo de um sistema de VI é propiciar condições para interpretar e compreender as informações, através de uma representação gráfica dos dados, possibilitando a aquisição de novos conhecimentos.

A quantidade de modelos para Visualização da Informação aumentou nos últimos tempos devido à busca de novas soluções para problemas de complexa visualização das informações e a grande quantidade de informações disponíveis, conforme mencionado por Matusi (2011).

Com isso, há um crescimento da exigência pela qualidade dos modelos e da escolha mais adequada do modelo de visualização da informação para solucionar o dado problema.

O processo de busca pelo modelo é de suma importância e consiste também em um processo sistematizado de Visualização da Informação dos modelos de Visualização da Informação. Desse modo, para se buscar um modelo apropriado para solução de um problema qualquer, é necessário entender os critérios de busca que possibilitem a visualização da informação mais adequada para os possíveis modelos de visualização, já então existentes.

Um processo de VI tem como objetivo principal buscar a melhor forma de mapear as informações e limitar a quantidade de dados que serão apresentados para garantir a melhor percepção e cognição visual destas informações pelos usuários. A melhor forma de mapeamento está relacionada ao objetivo esperado pela representação da informação, seja pela facilidade de busca da informação, seja através da ênfase de determinadas informações em relação a outras ou através da organização das informações. Esta representação da informação é realizada através de critérios e técnicas de visualização.

Os critérios da visualização, por exemplo, cor, tamanho e posição, auxiliam o usuário na separação e identificação dos dados possibilitando uma busca rápida, mais interativa e de fácil manipulação da informação. Os modelos se distinguem, uns dos outros, através do seu uso de critérios, técnicas e estruturas de visualização.

Assim como existem modelos para Visualização da Informação, existem modelos para categorização dos modelos de VI, ou seja, modelos para Visualização da Informação da Visualização da Informação. Estes últimos são denominados nesta dissertação como metamodelos de Visualização da Informação.

Existem algumas tentativas de criação desses metamodelos, como por exemplo, a Tabela Periódica da Visualização da Informação. Entretanto, atualmente existe uma carência de metamodelos que possuem uma estrutura de busca rápida, interativa e de fácil manipulação para busca dos modelos mais adequados para solução dos problemas de VI. Alguns autores como Hans-Jörg Schulz, Thomas Nocke, Maguns Heitzler e Heidrun Schumann consideram o problema em aberto e que pode demandar um esforço útil.

The literature contains a wealth of classifications, taxonomies, and frameworks that describe these tasks: lists of verbal task descriptions, mathematical task models, domain-specific task collections, and procedural task combinations into workflows. All of these serve the respective purpose well for which they have been developed. However, the research question of how to consolidate them under the hood of one common design space is still open, even though it has been shown on a smaller scale that such a combination into a common framework can be a useful endeavor. (SCHULZ; NOCKE; HEITLER; SCHUMANN apud BECKS; SEELING e FUJISHIRO; FURUHATA; ICHIKAWA; TAKESHIMA, 2013).

Neste sentido, esta dissertação propõe um *framework* para visualização de informações baseado na lógica de Treemap, inspirada no modelo proposto pela Tabela Periódica. Este *framework* pode ser visto como um metamodelo que permite representar a visualização, interpretar e compreender as informações e possui uma estrutura de busca da informação simples e rápida.

A implementação deste metamodelo exige o uso de componentes WEB para fazer a busca da informação e apresentá-la ao usuário. Neste sentido, se faz necessário a escolha dos componentes WEB que podem compor este metamodelo, a fim de torná-lo um parâmetro de eficiência e eficácia para qualquer busca de modelos de visualização. Da mesma forma, pode-se levantar a hipótese de que existem modelos e componentes WEB que se distinguem dos demais pelo desempenho na busca da informação.

Assim, constitui a motivação para realização desta dissertação o desafio de obter um novo metamodelo mais adequado que os existentes na literatura, facilitando na busca de modelos que podem ser úteis no desenvolvimento de componentes e sistemas para visualização da informação. A motivação secundária é também contribuir, de certa maneira, com a pesquisa visual de usuários da WEB, possibilitando uma visualização mais rápida, usual e adequada a suas necessidades de negócio.

Outras motivações desta dissertação foram os estudos realizados durante a elaboração e construção de componentes de busca na WEB para a biblioteca online do sistema de realidade virtual (RV) da CEMIG e da Freeman e também nos componentes de busca na WEB do sistema SGF da Fundação Fisk, que contribuíram como prova de conceito para análise comparativa do desempenho de alguns modelos do metamodelo proposto.

Sendo assim, o público-alvo dessa dissertação consiste em desenvolvedores de sistemas WEB, pesquisadores de modelos de VI e outros que já conhecem os conceitos básicos de gráficos visuais e possam manipular a busca de modelos em um determinado metamodelo de VI.

1.1 Objetivo Principal

Esta dissertação apresenta como finalidade criar uma proposta de metamodelo de visualização da informação mais adequada que os metamodelos propostos atualmente, principalmente no que diz respeito ao quesito de desempenho de busca dos modelos no metamodelo.

Assim, é também objetivo analisar, desenvolver e testar a hipótese de que existem modelos de visualização da informação melhores ou mais adequados que outros com relação ao desempenho na busca pela informação. Como resultado final, esta dissertação propõe um *framework*, na condição de um metamodelo de VI, que possibilite o usuário, escolher o modelo mais apropriado ao desenvolvimento do seu sistema, com maior rapidez.

Não é objetivo desta dissertação recriar o conjunto de modelos do metamodelo da tabela periódica da visualização da informação, apesar de que foram selecionados alguns modelos para se realizar uma prova de conceito, com a finalidade de analisá-los para verificar a possibilidade de conceber, elaborar e construir outros modelos, mais adequados que eles.

1.2 Objetivos Específicos

A seguir, foram listados os objetivos específicos, estipulados com o propósito de alcançar o objetivo principal desta dissertação:

- Definir uma arquitetura de software capaz de utilizar e integrar os componentes dos principais modelos de visualização da informação. Estes componentes devem ser suficientes ao desenvolvimento dos sistemas WEB:
 - **SGF:** sistema de gestão das franquias pelas Fundação Fisk
 - **Biblioteca RVCemig:** biblioteca de modelos do sistema de realidade virtual RVCemig, responsável pela navegação nas 57 usinas e subestações da CEMIG
 - **Biblioteca Freeman:** biblioteca responsável por mostrar e controlar as versões dos projetos (*unity3d*) de RV da Freeman.
- Escolher os componentes necessários para implementar os modelos mais usados na visualização da informação.
- Implementar instâncias dos componentes dos modelos de VI mais usados.
- Testar as instâncias dos componentes dos modelos de VI mais usados.
- Comparar as instâncias dos componentes com relação ao desempenho e qualidade na visualização da informação.
- Verificar os principais problemas de cada componente implementado, suas vantagens e desvantagens.
- Implementar um software com a arquitetura para instanciar todos os componentes dos modelos de visualização da informação mais usados.
- Construir um metamodelo de visualização da informação.
- Implementar o software do metamodelo.
- Testar o metamodelo criado e comparar os resultados com os metamodelos já existentes.
- Analisar os resultados finais de testes dos metamodelos.
- Verificar o motivo de algum metamodelo de VI ser melhor que outro.

Com a finalidade de atingir os objetivos específicos, algumas metas foram traçadas:

- Pesquisar os modelos mais usados na visualização da informação.
- Eleger os modelos mais adequados para visualização da informação segundo as necessidades do usuário.
- Definir a lista de modelos mais usados na visualização da informação.
- Buscar os frameworks que possibilitam a implementação dos componentes dos modelos de VI mais usados.
- Estudar as características que podem garantir que um determinado modelo de visualização da informação possa ser mais adequado a outro.
- Buscar técnicas que possibilitam melhorar os componentes com relação ao desempenho e qualidade na visualização da informação.
- Eleger os metamodelos mais adequados para a visualização da informação.
- Pesquisar os metamodelos de visualização da informação tradicionais.
- Verificar se os metamodelos já existentes são mais adequados para busca rápida da informação.
- Escolher os modelos candidatos ao metamodelo.

Um diagrama do tipo mapa mental com os principais objetivos e metas desta dissertação é apresentado na Figura 2.

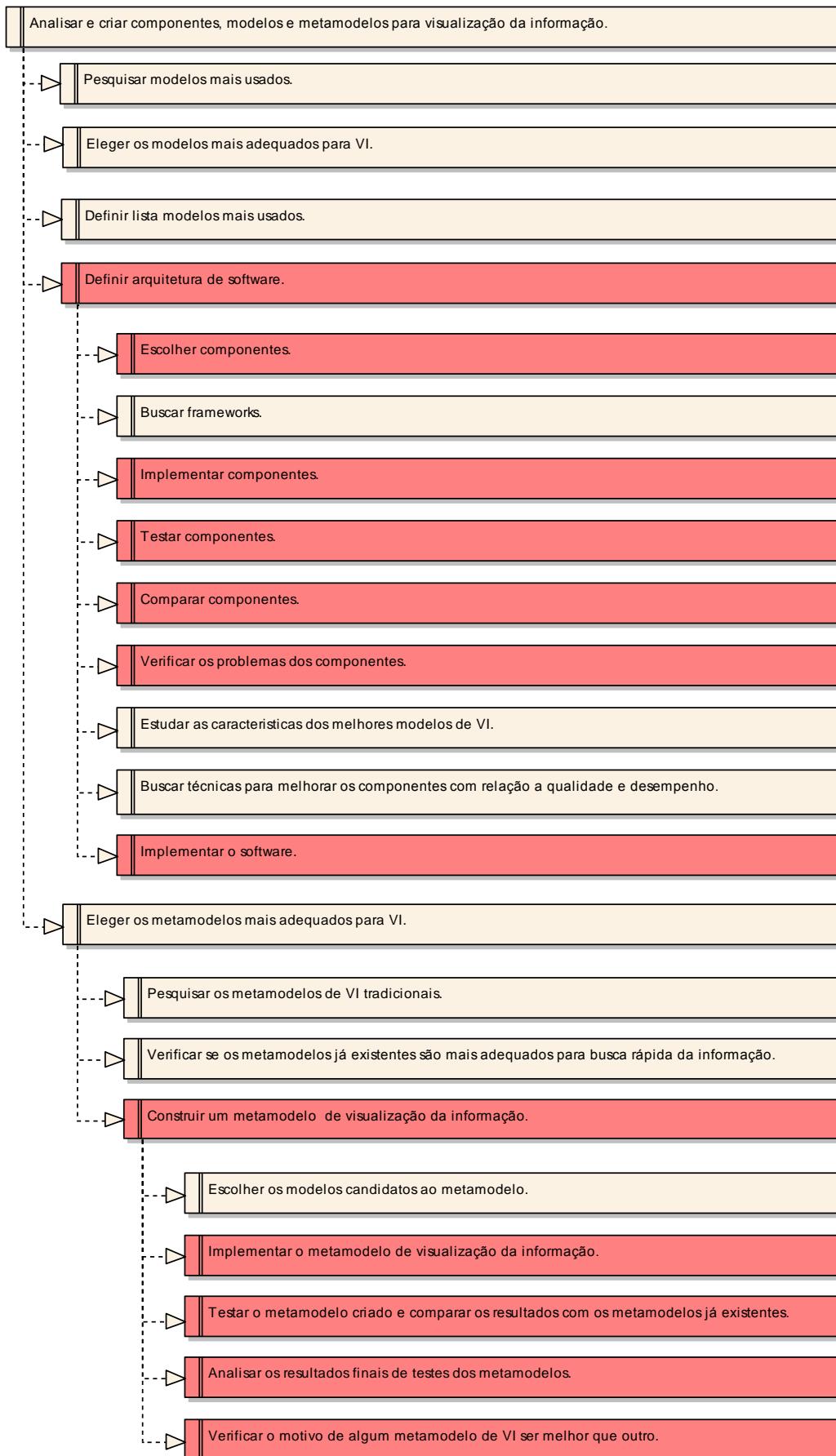


Figura 2. Objetivos principais da dissertação em forma de mapa mental.

1.3 Organização desta Dissertação

O Capítulo 1 apresenta uma introdução e as justificativas que conceberam a criação desta dissertação. São apresentadas também, as principais motivações que levaram a realização da dissertação.

O Capítulo 2 contém a fundamentação teórica da dissertação englobando, inicialmente, os conceitos e definições suficientes para a classificação da visualização da informação. Uma análise histórica e evolutiva destes conceitos, em sua ordem cronológica, mostra o motivo das suas mudanças e também pode indicar alguma necessidade futura de mudança na teoria, conforme a permanência da busca por novas taxonomias para separação e classificação dos modelos de visualização da informação que ainda surgem. Ainda neste capítulo, são apresentados os modelos e metamodelos de visualização da informação existentes na literatura e os conceitos básicos que possibilitam a criação de um novo metamodelo. Dois outros conceitos fundamentais para a dissertação foram contemplados, a teoria da análise de algoritmos, muito usada na área computacional. A análise de algoritmos permite medir o desempenho dos algoritmos computacionais.

O Capítulo 3 apresenta os trabalhos correlatos e linhas de pesquisas próximas à proposta de metamodelos de visualização da informação, como é a proposta da tabela periódica da visualização da informação.

O Capítulo 4 descreve a proposta de criação de um metamodelo de visualização da informação. A análise comparativa entre o metamodelo existente e o novo metamodelo é realizada neste capítulo, juntamente com a execução de testes de desempenho de ambos e, finalmente, são apresentados os seus resultados. Para estudo e análise de desempenho do metamodelo, alguns modelos do metamodelo foram concebidos, elaborados e implementados como prova de conceito de que alguns modelos são mais adequados que outros para determinada finalidade.

O Capítulo 5 contém a apresentação dos componentes de busca de informação que foram implementados e uma análise comparativa dos desempenhos entre eles. A implementação deles deve fazer integração com outros componentes do sistema e sua organização deve seguir um projeto arquitetural. A arquitetura permite a organização dos pacotes e componentes necessários para o desenvolvimento de software. O conceito de arquitetura utilizado nesta dissertação encontra-se no Anexo 1 e a arquitetura proposta para implementação dos componentes WEB analisados nesta dissertação encontra-se no Capítulo 6.

Por fim, o Capítulo 7 apresenta as conclusões da dissertação, suas contribuições e, ainda, sugestões de possíveis trabalhos futuros que podem continuar o trabalho realizado.

Capítulo 2

Fundamentação Teórica

O presente capítulo aborda os conceitos básicos que suportam o caminho percorrido para atingir o objetivo desta dissertação. Dessa forma, este capítulo apresenta como finalidade o levantamento histórico e teórico acerca dos critérios utilizados para categorização dos modelos e componentes da VI, bem como novos critérios ou técnicas que podem auxiliar na escolha de melhores componentes e modelos. Com os conceitos de análise de algoritmos e Treemap é possível elaborar, analisar e criar metamodelos mais eficientes.

2.1 Critérios para Classificação da Visualização da Informação

Os diversos modelos de visualização da informação se distinguem uns dos outros pelo uso de diferentes formas ou padrões de representação gráfica da informação, denominadas nesta dissertação como “critérios” para classificação e organização gráfica da visualização da informação. Um exemplo de critério é a organização visual de dados por cores, símbolos, tamanhos, etc. Esses critérios são usados na taxonomia da informação e também para classificação dos modelos nos metamodelos de visualização da informação.

2.1.2 Outros Critérios para Taxonomia da Visualização da Informação

A visualização explora principalmente o sentido da visão, sendo que na percepção visual devem existir representações adequadas e considerar muitas das propriedades do sistema visual humano.

Visualizations have a small but crucial and expanding role in cognitive systems. Visual displays provide the highest bandwidth channel from the computer to the human. We acquire more information through vision than through all of the other senses combined. (WARE , 2004, p.2)

Na visualização da informação, existem vários critérios de estruturas visuais para separação das diversas informações existentes, tais como a forma, tamanho, cor, posição e símbolos. Existem critérios mais perceptivos que outros. Além disso, os dados podem estar organizados segundo esses critérios, em uma estrutura que possibilitem uma busca mais rápida, como por exemplo, estruturas hierárquicas.

Jacques Bertin (1983) classificou sistematicamente o uso de sete critérios visuais básicos fundamentais para representar qualquer informação gráfica e definir as relações entre os dados virtuais. A Figura 3 mostra os sete critérios definidos por ele.

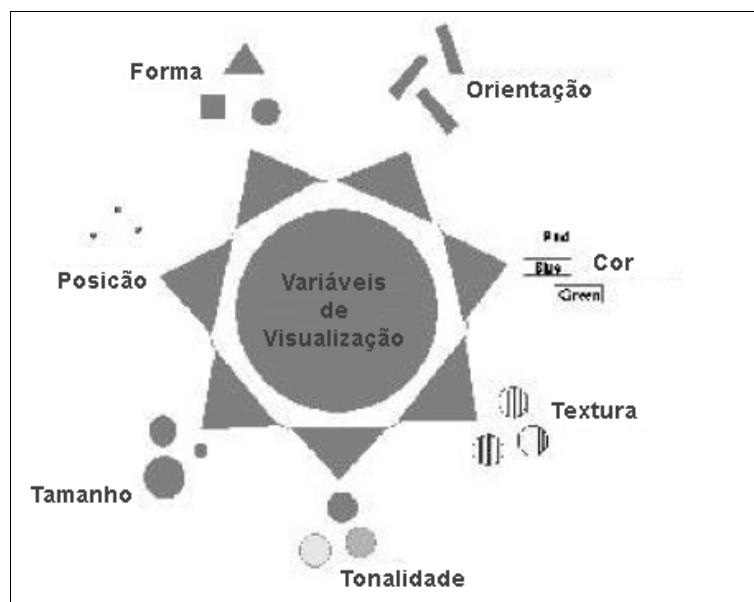


Figura 3. As sete variáveis visuais de Jacques Bertin (BERTIN, 1983 apoud BRADLEY, 2013).

Jacques Bertin (1983), identified seven important visual variables: position, form, orientation, colour, texture, value, and size. An essential feature of any visual display is that the audience's attention should be focused on one main point. Any of these seven variables can achieve this end; ideally, they should all work together. (BRADLEY, 2013, p. 349)

A Figura 4 mostra que é possível também classificar os critérios em mais ou menos perceptíveis da forma quantitativa. Segundo Cleveland e McGill's (1984), a posição e o tamanho são critérios mais perceptíveis que os critérios de cor e a opacidade.

(...) 10 elementary perceptual tasks that people use to extract quantitative information from graphs. (Color saturation is not illustrated, to avoid the nuisance and expense of color reproduction.) The pictorial symbol used for each task is meant to be suggestive and might not necessarily invoke only that task if shown to a viewer. For example, a circle has an area associated with it, but it also has a length, and a person shown circles might well judge diameters or circumferences rather than areas, particularly if told to do so. (CLEVELAND; MCGILL, 1984)

Entretanto, para este autor, o grau de percepção entre os critérios é mostrado de acordo com a comparação quantitativa da informação.

A Figura 5 mostra os critérios de visualização, segundo as perspectivas de graus de percepção de dados quantitativos e não quantitativos (quando são nominais ou ordinais). As informações cujos dados não são quantitativos podem ser disponibilizados de uma melhor forma, quando usados os critérios mais adequados para esse tipo de visualização. Assim, a falta ou falha na disponibilização da informação pode estar relacionada na escolha ou construção de um modelo de visualização, mais precisamente na utilização dos critérios que organizam os dados em tal técnica.

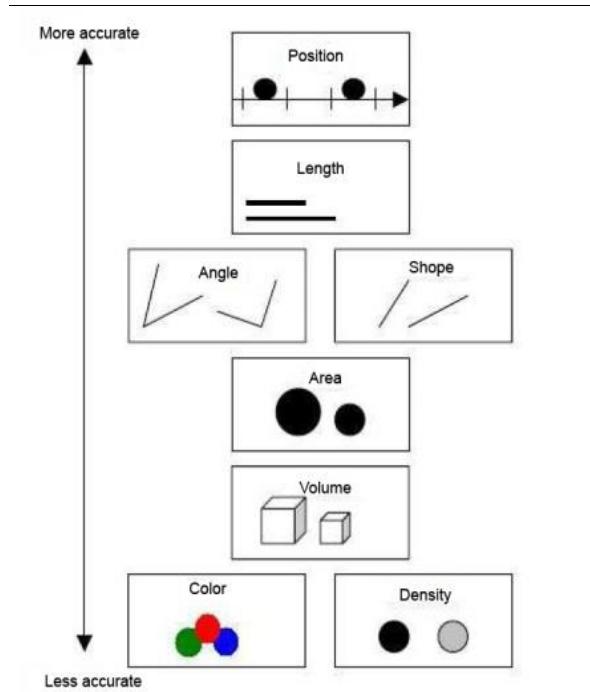


Figura 4. Critério gráfico segundo Cleveland e McGill's. (CAILLETEAU, 1999, p. 43)

Segundo Mackinlay (1999), a escolha da técnica de representação visual adequada para um determinado conjunto de informações deve se basear em critérios de expressividade e eficácia. O critério de expressividade está relacionado às representações gráficas que traduzem exatamente a informação com foco no interesse do utilizador. O critério de eficácia está relacionado com a facilidade de compreender as representações e as informações que elas expressam. O uso inadequado do critério de visualização da informação diminui a percepção dos dados pelo usuário e determina uma complexidade de tempo requerida ao usuário, para interpretar tais informações.

Just as data can be distinguished by whether it is nominal, ordinal, or quantitative (we even distinguished subcategories such as spatial), so the spatial axes can be similarly distinguished according to whether they are unstructured (unlabeled), nominal (labeled), ordinal (have an ordinal quantity associated with them), or quantitative (e.g. weighted links). The graphic space is defined by composing together spatial axes to form a substrate into which the rest of the visual representation is poured. Which data variables are associated with the axes of the space is the most potent representational decision. Only two or at most three variables can be encoded into graphical axes; additional variables must be encoded into

graphical axes; additional variables can be encoded into graphical axes; additional variables must be encoded using other graphical properties. The graphic properties of connection and containment allow additional ways to use space in coding. In addition, variables can be associated with what type of mark is placed upon the graphical substrate and these marks can be modified by retinal properties, such as size, orientation, grayscale, color, texture, and shape. (CARD; MACKINLAY; SHNEIDERMAN; CARD, 1999, p. 637)

É possível calcular o grau de eficácia de um modelo de visualização da informação realizando a análise da complexidade de tempo do conjunto de critérios utilizados nessa técnica. Como visto no tópico 2.2 *Teoria da Análise de Algoritmos* desta dissertação, a complexidade pode ser constante, logarítmica, linear ou exponencial quando se trata de análise de desempenho de busca computacional.

Segundo Alexandre e Tavares (2010), além de manter a integridade dos dados geradores, a visualização deve ser caracterizada tanto pela facilidade de leitura, quanto pela aptidão em fornecer o que se deseja a partir da interpretação da imagem gerada. Para ser efetiva, uma visualização deve transmitir rapidamente as informações e não induzir a erros; para tal, deve atender às capacidades de percepção da visão humana. Segundo Dastani (2002), tanto a expressividade como a eficácia são dependentes da percepção humana, pois não há efetividade sem uma representação expressiva que esteja de acordo com os requisitos da percepção humana.

We believe that the expressiveness criterion makes only sense if visual structures are perceivable by human viewers. This implies that both expressiveness and effectiveness criteria depend on the human visual system and that the expressiveness / effectiveness distinction is not a distinction between perceptually depending and perceptually not depending conditions. (DASTANI, 2002)

Assim, para Alexandre e Tavares, outros fatores, além do processo cognitivo podem também influenciar no processo de visualização da informação.

Considering the visualization of data such as cognitive process, other factors of human perception can also be used to contribute to the information visualization process. In fact, these additional factors can help the cognitive human processes in the recuperation of information contained in the data involved. (DULCLERCI; TAVARES, 2010)

Alexandre e Tavares (2010) relacionam os atributos visuais com a tabela de Mackinlay, elaborada com as prioridades de acordo com o seu grau de acuidade em termos perceptivos e a natureza dos dados. Na Figura 5, são considerados três tipos de dados aos quais vão sendo apresentados por ordem decrescente os atributos visuais perceptíveis.

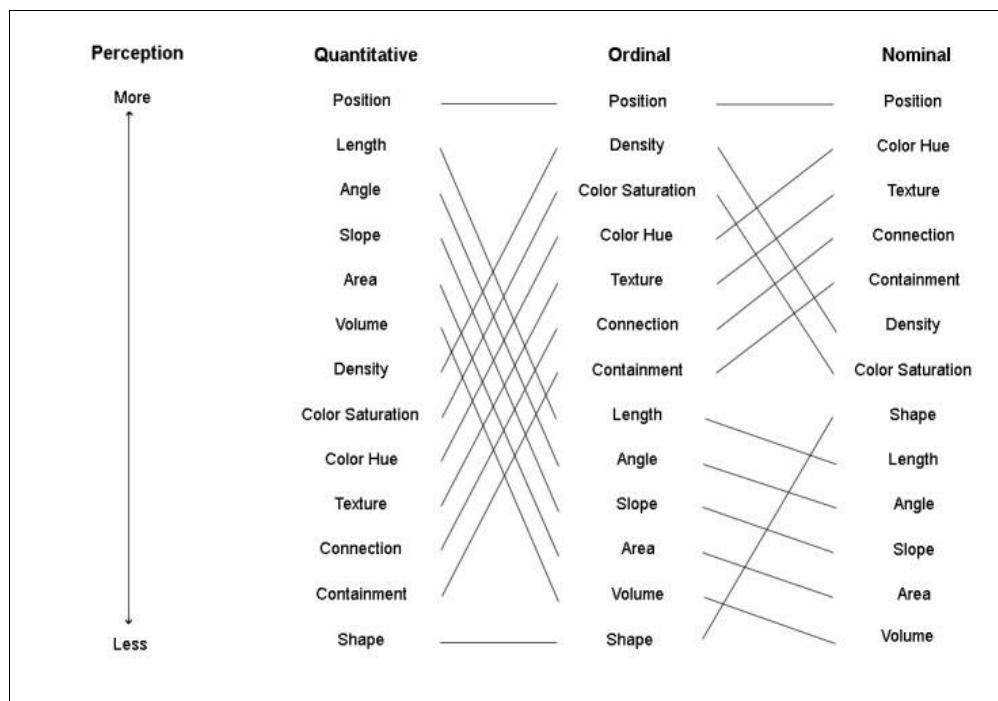


Figura 5. Acuidade de percepção dos atributos visuais, segundo Mackinlay.

O valor de um determinado sistema computacional de visualização de dados pode ser determinado pelas capacidades perceptíveis e do conhecimento obtido pelo seu utilizador a partir das imagens de visualização geradas pelo mesmo sistema. A percepção, por sua vez, depende do uso adequado dos critérios para a forma quantitativa, ordinal ou nominal utilizada na representação.

Segundo Shneiderman (1996), é possível otimizar ainda mais a utilização isoladamente de um critério de visualização da informação, por exemplo, a cor.

Designers are just discovering how to use the rapid and high resolution color displays to present large amounts of information in orderly and user-controlled ways. (SHNEIDERMAN, 1996)

Shneiderman compara imagens, que possuem maior quantidade de dados abstraídos pelos símbolos numa apresentação visual, com os textos, que possuem menor quantidade de informações visuais. Mas um critério de cor durante a análise de algoritmos de visualização pode apresentar menor complexidade de tempo, comparados ao critério de símbolos, principalmente se eles são parecidos e possuem o mesmo tamanho. No seu artigo, a abordagem da Treemap foi aplicada com sucesso aos diretórios de computador, dados de vendas, tomada de decisões de negócio e navegação na WEB, mas os usuários tomam 10-20 minutos para acomodar a mapas de árvores mais complexas. Isso se deve ao processo de aprendizagem cognitivo com a técnica de visualização, consistindo em um pré-requisito para sua utilização.

Shneiderman propõe uma taxonomia com sete tipos de dados que caracterizam os gráficos visuais, chamada *Task by Data Type Taxonomy* - TTT. Os tipos de dados da taxonomia são:

- Unidimensional
- Bidimensional
- Tridimensional
- Temporal
- Multidimensional
- Árvores
- Grafos

Shneiderman separa as organizações estruturais de árvores e grafos dos demais tipos e enfoca que existem variações e combinações destes tipos de dados. A TTT também define sete tarefas de usuário, em alto nível de abstração:

- Visão geral (*Overview*) – obter uma visão geral de toda a coleção.
- Aumento de tamanho (*Zoom in*) – aumento de tamanho sobre itens de interesse.

- Filtrar (*Filter*) – filtrar itens desinteressantes.
- Detalhes por demanda (*Details on demand*) – eleger um item ou grupo e obter detalhes quando necessário.
- Identificar relações (*Relate*) – encontrar e exibir as relações entre itens.
- Histórico (*History*) – manter um histórico de ações para desfazer e refazer ações e refinamento progressivo, recuperando aquelas que forneceram o resultado desejado.
- Extração (*Extract*) – permitir a extração de subconjuntos e de parâmetros de consulta.

Shneiderman sugere a busca por mais tarefas ou refinar a TTT para expandir a sua taxonomia (SHNEIDERMAN, 1996).

Observa-se que algumas taxonomias propostas pelos autores preocupam com o desempenho que o modelo oferece ao usuário na busca das informações. Alguns autores diferenciam e classificam os critérios através do seu desempenho no processo de cognição. Eles procuram mensurar o tempo gasto em cada modelo, através dos seus critérios utilizados, mas não conseguem uma forma objetiva de medir os desempenhos dos modelos, a fim de compará-los.

2.2 Teoria da Análise de Algoritmos

A análise de algoritmos é um ramo da ciência da computação que estuda as técnicas de análise e projeto de algoritmos. A análise é um mecanismo capaz de orientar a melhor forma de programar um algoritmo. Sendo assim, não é necessário que o algoritmo esteja implementado em uma linguagem de programação para realizar a sua análise.

A análise de algoritmos computacionais estipula a complexidade de tempo e de espaço através da complexidade do algoritmo. A complexidade do algoritmo reflete o esforço computacional requerido para executá-lo e complexidade de tempo e espaço refletem respectivamente a velocidade e quantidade de memória requerida ao executar o algoritmo. Outra regra para determinar a complexidade de um

algoritmo é levar em consideração os piores e os médios esforços dos subcálculos ou subprocedimentos.

Geralmente, o cálculo da complexidade concentra-se em determinar a ordem de magnitude do número operações fundamentais na execução do algoritmo e é representado por uma notação, por exemplo, a notação "*the Big oh*" é representada por 'O' e é usado junto com o "*order of growth*" para classificar a eficiência do algoritmo. A Tabela 1 mostra os possíveis "*order of growth*" utilizados na representação da complexidade de tempo de algoritmo. Ela também apresenta as classes de eficiências de algoritmos e suas características.

Tabela 1. Tabela com as complexidades de tempo de algoritmos (Puntambekar, 2010, p. 1-27).

Nome da Classe de Eficiência	Ordem de Crescimento	Descrição	Exemplo
Constante	1	Unidade base de complexidade.	Obter tamanho de uma lista.
Logarítmico	$\log n$	Se a complexidade é logarítmica, então o algoritmo nunca percorre todas as suas entradas, pois o problema é dividido em partes menores que serão iteradas.	Operação de busca binária
Linear	n	O tempo de execução do algoritmo depende do tamanho da entrada n .	Operação de busca sequencial.
$n \log n$	$n \log n$	Alguns exemplos de entrada são considerados para a lista de tamanho n .	Ordenando os elementos usando <i>merge sort</i> ou <i>quick sort</i> .
Quadrático	n^2	Essa complexidade existe quando é necessário percorrer a entrada n de forma duplicada para resolver o problema.	Percorrer uma matriz de elementos.
Cúbico	n^3	Quando o algoritmo precise de três iterações, ele possui este tipo de eficiência.	Performance na multiplicação de matrizes.
Exponencial	2^n	Essa complexidade existe quando uma taxa muito grande de crescimento de validações existe no algoritmo.	Geração de todos os subconjuntos de n elementos
Fatorial	$n!$	Esse tipo de complexidade existe quando é necessário permutar todas as combinações para resolver o problema.	Geração de todas as permutações de um problema combinatório.

Quando um mesmo problema pode ser resolvido por algoritmos de diferentes complexidades de tempo, é dito que o algoritmo é resolvido na menor delas. Mas caso um algoritmo tenha partes com diferentes eficiências, a complexidade deve considerar os piores e médios casos, conforme já posicionado anteriormente. Assim, para um determinado problema como ordenação de uma lista de números inteiros, é possível desenvolver algoritmos com complexidade $O(n^2)$ que requerem maior

complexidade de tempo, mas também é possível desenvolver algoritmos para o mesmo problema com complexidades $O(\log n)$ que requerem menor complexidade de tempo. Logo, este problema é de complexidade $O(\log n)$.

Suponha agora outro problema, mas que para solucioná-lo, seja necessário o uso de dois algoritmos, um $O(\log n)$ e outro $O(n^2)$, o pior caso é considerado e a complexidade de tempo passa a ser $O(n^2)$.

Os algoritmos com complexidade de tempo com constante, por exemplo, $O(\log n + c)$, tem as suas constantes desprezadas. Neste caso, a complexidade passa a ser $O(\log n)$. A constante significa que é preciso percorrer a lista de informações em um número fixo de vezes.

Um modelo de visualização da informação pode ser mais adequado que outro, dependendo dos objetivos a serem alcançados. Os objetivos podem estar relacionados com a disponibilização de determinados dados de visualização. Por exemplo, quando é preciso organizar as informações de maneira em que seja fácil e rápido encontrar os dados, pode-se utilizar uma visualização da informação que possua estrutura hierárquica ou em grafos. Isto pode possibilitar uma análise particular desta visualização.

A análise do desempenho dos modelos de visualização é feita no tópico **4.3.1 Análise de Algoritmos de Visualização** dessa dissertação. Dessa forma, testes foram realizados para verificar se a Treemap garante maior agilidade e facilidade na busca da informação com relação ao modelo utilizado pela visualização da informação da tabela periódica.

Isso implica que determinados modelos de visualização da informação podem ser mais adequados que outros, no que diz respeito à complexidade de tempo para abstrair os seus dados.

2.3 Concepção da Treemap

Treemap consiste em um modelo de visualização da informação criada por Shneiderman em 2006, capaz de organizar as informações em uma estrutura hierárquica, através dos princípios dessas próprias informações e apresentá-las num gráfico, a partir dessa estrutura (SHNEIDERMAN, 1996).

Isso permite o acesso rápido da informação, de forma visual, separando as informações mais importantes ou mais desejadas após a sua busca, em um processo fortemente interativo.

O usuário pode separar as informações segundo vários critérios visuais, como por exemplo: posição, área, cor e textura. A Figura 6 mostra um exemplo de Treemap cuja estrutura está orientada pelos critérios de posição, cor e área da “Concept Visualization”. É possível determinar uma estrutura na forma de árvore para cada critério de busca. No caso da imagem da Figura 7, a estrutura da árvore mostra uma organização orientada pelo critério de busca através de grupo da Treemap.

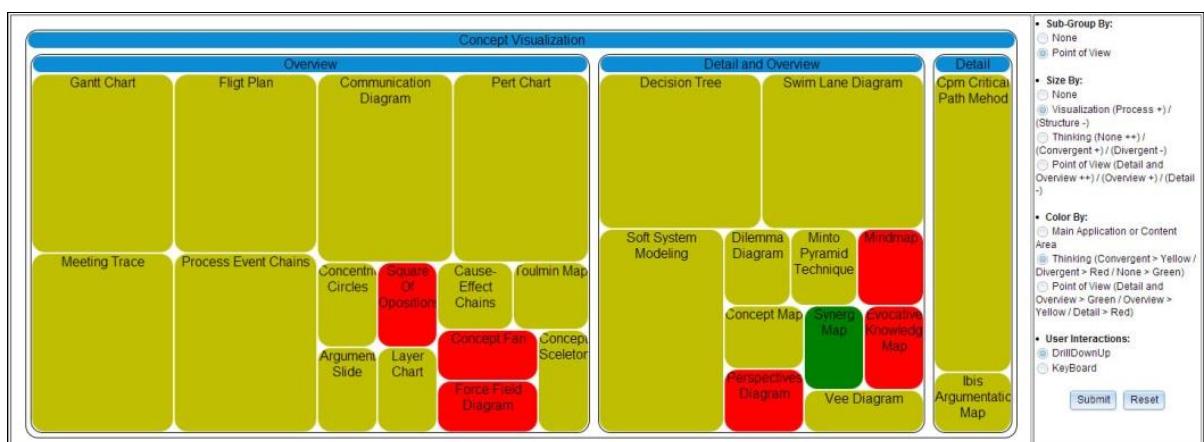


Figura 6. DrillDownUp de “Concept Visualization”.

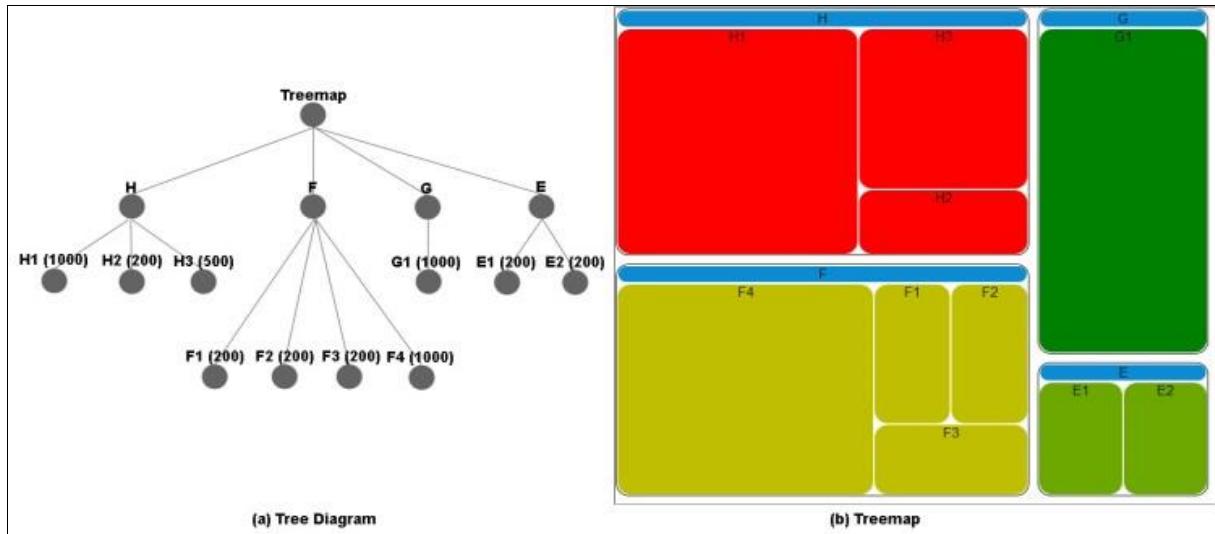


Figura 7. Exemplo de estrutura de Treemap.



Figura 8. DrillDownUp de “Concept Visualization – Detail and Overview”.

A estrutura da Treemap possibilita disponibilizar diversas formas de visualização para separá-las segundo critérios de busca dos dados. O usuário pesquisa a informação, associando-a a um princípio desejado (por exemplo, “Point of View” do tipo “Detail”) segundo um critério de busca também desejado (por exemplo, pela cor vermelha) e a Treemap apresenta as informações conforme a preferência do usuário.

Com tal arranjo, a mesma facilita a interação de acordo com as necessidades visuais do usuário, no que diz respeito à diversidade de escolha de formas de visualização. Por exemplo, é possível destacar determinadas informações segundo cores ou, as mesmas informações segundo área, para usuários daltônicos.

Assim, a Treemap possui diversas formas de visualizações através de percursos otimizados, possibilitando maiores percepções e interações com os usuários.

Capítulo 3

Trabalhos Correlatos

Algumas linhas de pesquisa, trabalhos e até mesmo propostas de metamodelos foram desenvolvidos com o intuito de alcançar conceitos fundamentais para elaboração de uma estrutura universal que represente os modelos de visualização de informação. Este capítulo mostra algumas destas tentativas e outros trabalhos correlatos que possuem este mesmo direcionamento.

3.1 Pesquisa de Eficácia na Visualização da Informação

Herman, Melançon e Marshall (2002) fazem uma pesquisa pelos modelos de visualização da informação para navegação e visualização gráfica. Para eles, a estrutura do modelo de visualização da informação é importante para aplicabilidade do gráfico.

A simple way to determine the applicability of graph visualization is to consider the following question: is there an inherent relation among the data elements to be visualized? If the answer to the question is "no", then data elements are "unstructured" and the goal of the information visualization system might be to help discover relations among data through visual means. If, however, the answer to the question is "yes", then the data can be represented by the nodes of a graph, with the edges representing the relations. (HERMAN; MELANÇON; MARSHALL, 2000)

Em seu trabalho, os autores analisam os modelos com relação ao desempenho e usabilidade abordados para a navegação em grandes gráficos e métodos de redução de complexidade visual através de reorganização de dados. Dois fatores importantes para a usabilidade consistem na navegação e interação. Para isso, várias técnicas podem ser utilizadas como “*zoom and plan*”, “*Focus+Context Techniques*”, “*Incremental Exploration and Navigation*” e “*Clustering*”.

Para Winckler, Palanque e Freitas (2004), cada aplicação requer um estudo específico para a escolha dos modelos de visualização a serem utilizados, a fim de determinar se a técnica é útil.

Em seu trabalho, os modelos podem ser decompostos através de critérios de visualização, como por exemplo, cor, *zoom* ou foco. Dois modelos são utilizados para análise e comparação, a Treemap e o Hiperbolic Browser.

It consists of a set of optional and concurrent rendering functions (Focus, Isolate and reinforce). Figure 5b show how a particular strategy (Reinforce) has been further decomposed to include specific implementation constraints. For example, the rendering functions could be achieved either by highlighting the colors or zooming over graphic elements. (WINCKLER; PALANQUE; FREITAS, 2004)

Embora o trabalho mostre que os critérios podem mensurar a usabilidade e eficácia do modelo, não é demonstrado como eles podem ser comparados para atingir este objetivo ou como o desenvolvedor mensura qual o critério mais útil e eficaz para a sua elaboração e construção. A Figura 9 ilustrada os a decomposição do modelo em critérios de visualização, feito por Winckler, Palanque e Freitas.

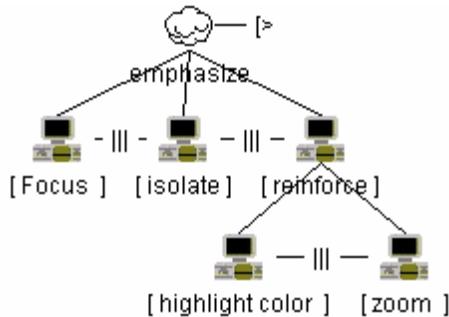


Figura 9. Exemplo de decomposição de modelo por critérios de visualização.
(WINCKLER; PALANQUE; FREITAS, 2004)

Segundo os autores, a avaliação de usabilidade de um modelo de visualização da informação somente pode ser realizada através da representação visual e das interações técnicas. Os autores concluem que os modelos que possuem estruturas formais são mais eficazes que os modelos não estruturados.

Usability evaluation of an information visualization technique can only be done by the joint evaluation of both the visual representation and the interaction techniques. (...) We conclude that the use of task models allows generating test scenarios which are more effective than informal and unstructured evaluations. (WINCKLER; PALANQUE; FREITAS, 2004)

Embora a conclusão direcione no uso de modelos formalmente estruturados, não existe uma proposta nestes trabalhos correlatos para unificação dos modelos numa representação de metamodelo formalmente estruturado.

3.2 Propostas de Classificação de VI

Alguns trabalhos correlatos buscam a criação de novos modelos de visualização da informação em áreas específicas.

Maria Cristina de Oliveira e Haim Levkowitz (2003) perceberam a necessidade de criação de modelos de visualização para ajudar na compreensão dos resultados de algoritmos de mineração de dados, pois são capazes de criar um modelo mental de como este tipo de sistema em particular funciona.

Visual mapping techniques are now being used both to convey results of mining algorithms in a manner more understandable to end users and to

help them understand how an algorithm works. In fact, the ability to create a good mental model of how a particular DM algorithm works is essential if end users, usually the domain experts, are ever to exercise a greater control over the DM/KDD process. (OLIVEIRA; LEVKOWITZ, 2003)

Keim (2002) verifica a necessidade de uma classificação dos modelos de VI para facilitar e agilizar a escolha do modelo mais adequado para representar as informações dos resultados de mineração de dados. A mineração envolve a busca de um volume de dados e, segundo o autor, uma boa visualização da informação pode ajudar a interpretar a grande quantidade de informações:

Exploring and analyzing the vast volumes of data becomes increasingly difficult. Information visualization and visual data mining can help to deal with the flood of information. The advantage of visual data exploration is that the user is directly involved in the data mining process. There is a large number of information visualization techniques which have been developed over the last decade to support the exploration of large data sets. In this paper, we propose a classification of information visualization and visual data mining techniques which is based on the data type to be visualized, the visualization technique and the interaction and distortion technique. (KEIM, 2002)

Na sua análise, os modelos de visualização são úteis para mostrar uma visão geral dos dados, mas também permitir ao usuário identificar subconjuntos de interesse, as vezes concentrando os dados deste subconjunto em outro modelo ou técnica de visualização. Esta dissertação utiliza a classificação de visualização da informação de Shneiderman (1996).

Focando na dimensionalidade e estrutura organizacional do modelo, este trabalho correlato descreve e analisa alguns modelos de visualização limitados à área de interesse da mineração.

Stolte (2003) também busca a criação de novos modelos que envolvem especificamente a possibilidade de variação da multi-escala gráfica. O objetivo destes modelos é buscar maior eficácia na visualização dos dados. Para os autores, a técnica

destes modelos consegue abstrair grande quantidade de dados nas representações gráficas.

Multiscale visualizations are an effective technique for facilitating this process because they change the visual representation to present the data at different levels of abstraction as the user pans and zooms. At a high level, because a large amount of data needs to be displayed, it is highly abstracted. As the user zooms, the data density decreases and thus more detailed representations of individual data points can be shown. (STOLTE, 2003)

Matsui, Yamanouchi e Sunahara (2011) propõe uma classificação e um framework para VI no desenvolvimento de aplicações WEB. Os desenvolvedores WEB têm que lidar com dados para estabelecer a forma de mostrar seu significado para usuários em geral. A combinação de dados pode ser volumosa e impor um fardo para os programadores. Isto ocorre, segundo os autores, pois desenvolvedores e *web designers* possuem diferentes conhecimentos e habilidades sobre como obter e mostrar informações significativas.

In ordinal method, web developers have to deal with data to establish how to show its meaning to the users usually. For example, to combine data sets, e.g. cartographic information and weather information and to use the method of information visualization for the user interface to make them more understandable. But these procedures could impose a burden on web developers. This is caused by that web programmers and web designers as web developers have a different knowledge and skills about how to get and show meaningful information. Therefore, sometimes each effort to tell information to the users through web application tends not to correspond. (MATUSI; YAMANOUCHI; SUNAHARA, 2011)

Sendo assim, os autores criaram um protótipo, ilustrado pela Figura 10, na qual o usuário pode escolher os tipos de informações e também os critérios de visualização que eles necessitam na criação do modelo.

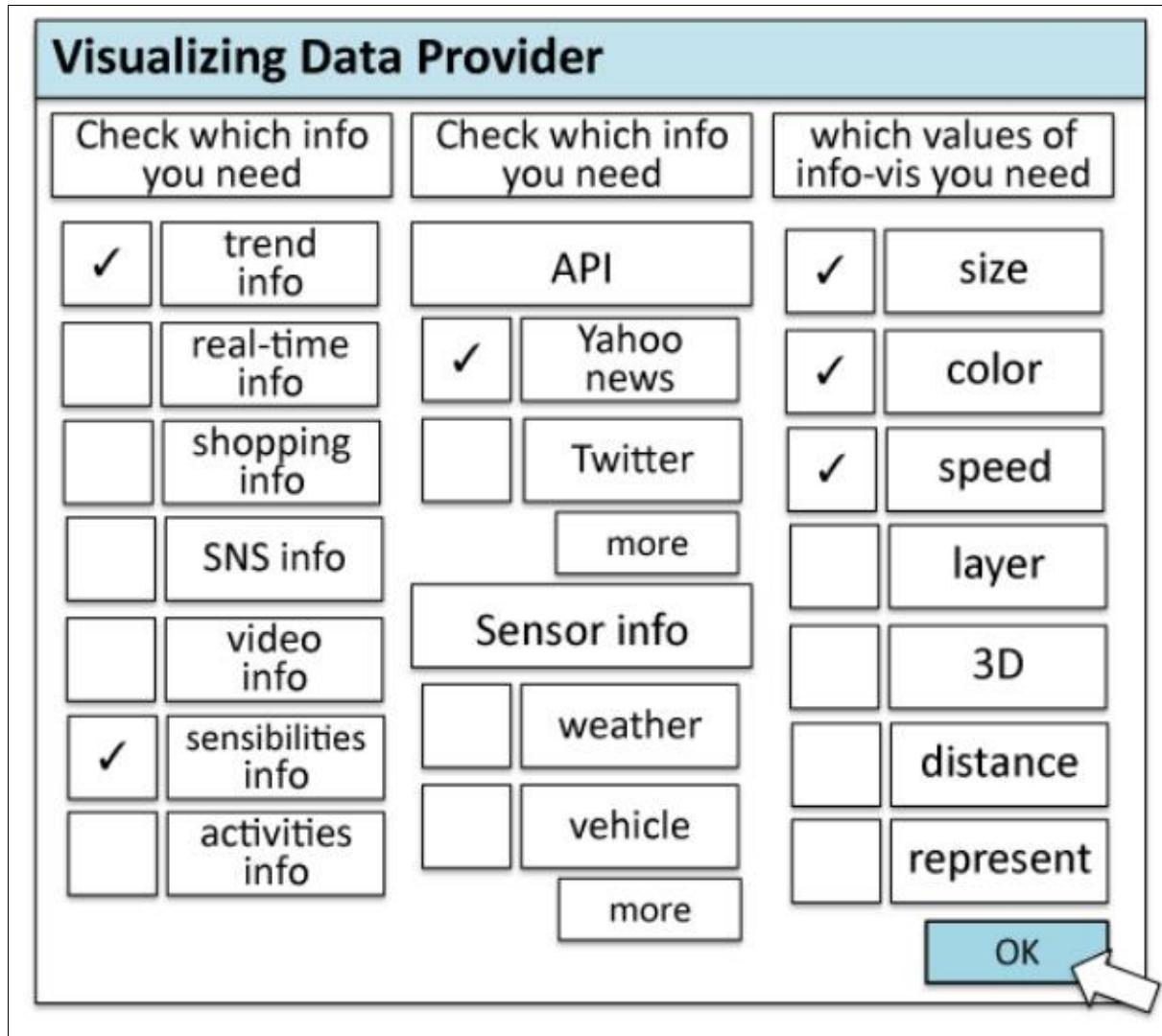


Figura 10. Protótipo proposto por Matusi para escolha dos critérios na criação dos modelos. (MATUSI; YAMANOUCHI; SUNAHARA, 2011)

Verifica-se uma necessidade da classificação da VI nos trabalhos de Keim, Matsui e Stolte, mas não se verifica a iniciativa de se criar a unificação dos modelos em um metamodelo, mas se limita apenas nas técnicas para se atingir ou melhorar os modelos para objetivos específicos como aumentar a eficácia ou criar modelos para resolver melhor determinados problemas específicos, como melhor representar as informações resultantes da mineração de dados numa estrutura gráfica.

3.3 Tabela Periódica da Visualização da Informação

O metamodelo constituído e apresentado por Lengler e Eppler (2007) classifica as técnicas de visualização e as organizaram em uma tabela periódica da informação. O modelo utilizado consiste em uma metáfora da tabela periódica da química.

Os princípios da estrutura organizacional que os autores utilizaram são:

- *Complexity of Visualization*: de baixo para cima, refere-se ao número de regras aplicadas para a utilização e / ou número de interdependências dos elementos a serem visualizadas.
- *Main Application or Content Area [como?, o que?]*: Dados, Informação, Conceito, Metáfora, Estratégia, Conhecimento Misturado. Além disso, os membros desse grupo também podem ser classificados de acordo com a intensidade do conhecimento, iniciando das explícitas, cujo conhecimento é objetivo (como *Data Visualization*), para as mais implícitas, visualizações subjetivas do conhecimento (como *Compound Knowledge Visualization*).
- *Point of View [quando?]*: *Detail* (com destaque aos itens individuais), *Overview* (figuras grandes), *Detail and Overview* (os dois ao mesmo tempo).
- *Type of Thinking Aid [por quê?]*: *Convergent* (reduzindo a complexidade) versus *Divergent* (adicionando complexidade).
- *Type of Representation [what?]*: *Process* (passo a passo cíclico no tempo e/ou sequência contínua), *Structure* (por exemplo, hierarquia ou redes causais).

A Figura 11 mostra o modelo de visualização da tabela periódica da informação proposto por Lengler e Eppler. Selecionando os modelos de visualização mais importantes segundo os autores.

In these sources we have found approximately 160 visual methods. We have reduced these to a set of a hundred methods (...) The resulting hundred visual methods that have met these criteria were then analyzed with regard to the following properties: *graphic format* employed (i.e., quantitative chart, qualitative diagram, cartographic map, visual metaphor, tables) , typical content type (e.g., concepts, problems, people), *application context* (e.g., management, engineering, counseling etc.) and *scope* (narrow vs. wide), *difficulty* of their application, *originating discipline*, vicinity over overlaps to other visual methods. (Lengler e Eppler, 2007).

A PERIODIC TABLE OF VISUALIZATION METHODS																		
>○< C conneum				Data Visualization Visual representations of quantitative data in schematic form (either with or without axes)				Strategy Visualization The systematic use of complementary visual representations in the analysis, development, formulation, communication, and implementation of strategies in organizations.										G graphic facilitation
>○< Tb table	>○< Ca cartesian coordinates			>○< Information Visualization The use of interactive visual representations of data to optimize cognition. This means that the data is transformed into an image, it is mapped to screen space. The image can be changed by users as they proceed working with it.				>○< Metaphor Visualization Visual Metaphors position information graphically to organize and structure information. They also convey an insight about the represented information through the key characteristics of the metaphor that it employed										
>○< Pi pie chart	>○< L line chart			>○< Concept Visualization Methods to elaborate (mostly) qualitative concepts, ideas, plants, and analyses.				>○< Compound Visualization The complementary use of different graphic representation formats in one single schema or frame										
>○< B bar chart	>○< Ac area chart	>○< R radar chart colweb	>○< Pa parallel coordinates	>○< Hy hyperbolic tree	>○< Cy cycle diagram	>○< T timeline	>○< We venn diagram	>○< Mi mindmap	>○< Sq square of oppositions	>○< Cc concentric circles	>○< Ar argument slide	>○< Sw swim lane diagram	>○< Gc gantt chart	>○< Pm perspectives diagram	>○< D dilemma diagram	<○> Pr parameter ruler	<○> Ri rich picture	
>○< Hi histogram	>○< Sc scatterplot	>○< Sa sankey diagram	>○< In information lens	>○< E entity relationship diagram	>○< Pt petri net	>○< Fl flow chart	>○< Cl clustering	>○< Lc layer chart	>○< Py moto pyramid technique	>○< Ge cause-effect chains	>○< Tl toulmin map	>○< Dt decision tree	>○< Cp con critical path method	<○> Cf concept fan	<○> Co concept map	<○> Lc iceberg	<○> Lm learning map	
>○< Tk tukey box plot	>○< Sp spectrogram	>○< Da data map	>○< Tp treemap	>○< Cn cone tree	>○< Sy system dyn./ simulation	>○< Df data flow diagram	>○< Se semantic network	>○< So soft system modeling	>○< Sn synergy map	>○< Fo force field diagram	>○< Ib irri argumentation map	>○< Pr process event chairs	>○< Pe pert chart	>○< Ev evocative knowledge map	>○< V fee diagram	<○> Hh heaven 'n' hell chart	<○> I infomural	
Cy Process Visualization		Note: Depending on your location and connection speed it can take some time to load a pop-up picture. © Ralph Lengler & Martin J. Eppler, www.visual-literacy.org																
Hy Structure Visualization		version 1.5																
Hy Overview		>○< Su supply demand curve	>○< Pc performance charting	>○< St strategy map	>○< Oc organisation chart	>○< Ho house of quality	>○< Fd feedback diagram	>○< Ft failure tree	>○< Mq magic quadrant	>○< Ld life-cycle diagram	>○< Po porter's five forces	>○< S s-cycle	>○< Sm stakeholder map	>○< Is ishikawa diagram	>○< Tc technology roadmap			
Hy Detail		>○< Ed edgeworth box	>○< Pf portfolio diagram	>○< Sg strategic game board	>○< Mz münzberg's organigram	>○< Z zwickly's morphological box	>○< Ad affinity diagram	>○< De decision discovery diagram	>○< Bm bcg matrix	>○< Stc strategy canvas	>○< Vc value chain	>○< Hy hyper-cycle	>○< Sr stakeholder rating map	>○< Ta tags	>○< Sd spray diagram			

Figura 11. Tabela periódica dos modelos de visualização (LENGLER; EPPLER, 2007).

A tabela periódica disponibiliza 100 modelos de visualização da informação em uma única figura, posicionando uma sigla para cada modelo e o seu nome logo abaixo da sigla. Ao realizar o apontamento no modelo (posicionando o mouse na imagem), a tabela periódica de visualização da informação mostra um correspondente exemplo desse modelo. A tabela possui duas legendas para os critérios organizacionais dos modelos de visualização da informação utilizados na estrutura deste metamodelo.

Entretanto, a tabela não possibilita a pesquisa direta pelo nome de um determinado modelo. Ela também só comporta um número reduzido de modelos, não sendo possível abranger novos modelos que podem ser criados. Por se tratar de uma única imagem para toda a tabela, a busca por um determinado modelo de visualização através do seu nome não é simples, pois o usuário deverá visualizar

modelo por modelo e buscar o nome desejado. Não há interação para mostrar somente os modelos de visualização correspondentes a uma determinada categoria ou modelos que atendam um dado critério.

A organização da tabela periódica posiciona os modelos de visualização, que possuem características e finalidades semelhantes, por proximidade. Tal estratégia facilita a categorização dos modelos de visualização de forma estática, principalmente no que diz respeito à classificação através de "Aplicação principal ou área de conteúdo" ("Main Application or Content Area"). Entretanto, a partir desse metamodelo, há dificuldades na busca da informação quando se deseja pesquisar por vários critérios simultaneamente, como por exemplo: obter todos os modelos de visualização que sejam "Concept Visualization", "Detail and Overview", "Visualization Structure" e "Thinking Divergent".

Assim, além de consistir em uma busca lenta, o metamodelo não disponibiliza uma forma interativa para obtenção do resultado, de forma a separá-lo visualmente das demais informações disponíveis no metamodelo.

A tabela periódica da visualização da informação se utiliza do critério de cor da área e posição para diferenciar as categorias de "Aplicação principal ou área de conteúdo" ("Main Application or Content Area"), cor da sigla para distinguir os princípios de "Tipo de Representação" ("Type of Representation") e símbolos para distinguir os princípios de "Ponto de vista" ("Point of View") e "Tipo de Pensamento de Ajuda" ("Type of Thinking Aid"). Várias técnicas de visualização foram utilizadas para classificar os modelos de visualização da informação segundo os critérios utilizados na metáfora da tabela periódica, mas o processo se torna complexo, uma vez que não facilita a busca dos modelos durante a visualização da informação.

Ao realizar consecutivas pesquisas de modelos através desses símbolos, o usuário pode se confundir nos critérios de pesquisa dos modelos, principalmente quando suas imagens são similares, embora os critérios destes símbolos sejam distintos.

Capítulo 4

Proposta de Metamodelo para Visualização da Informação

Este capítulo possui como finalidade apresentar uma proposta de metamodelo para visualização da informação que possua uma estrutura ideal para busca do melhor modelo de visualização a ser utilizado em determinada manipulação de informação, baseada nos conceitos da Treemap. Além disso, a análise de algoritmo, os detalhes de sua implementação, testes dos metamodelos e seus resultados também são apresentados para a análise comparativa com outras propostas.

4.1 Treemap da Visualização da Informação

O metamodelo da Tabela Periódica da Visualização da Informação, proposto por Lengler e Eppler foi convertido nesta dissertação em um metamodelo de Treemap para Visualização da Informação, mantendo os princípios da estrutura organizacional, descritos no tópico 3.3 *Tabela Periódica da Visualização da Informação*, com os novos critérios relacionados na Tabela 2. É mantido o processo interativo de apontamento e seleção em um modelo para mostrar um exemplo de visualização da informação correspondente a um dado modelo.

Tabela 2. Critérios que definem as formas de visualização no metamodelo Treemap.

	<i>Main Application or Content Area</i>	<i>Point of View</i>	<i>Type of Representation</i>	<i>Type of Thinking Aid</i>
Posição (ou Tamanho)	X	X		
Área		X	X	X
Cor	X	X		X

As legendas foram substituídas por uma busca interativa com o usuário, que permite a escolha do critério, o qual deve gerar a forma de visualização da Treemap. A busca interativa considera os critérios através dos princípios correlacionados na Tabela 2. Observe que o critério de símbolo usado no metamodelo da *Periodic Table*, não é mais utilizado.

A escolha dos critérios que definem as formas de visualização das informações é importante para definição da busca mais adequada e com a percepção esperada pelo usuário. Neste metamodelo, alguns critérios podem ser utilizados para definir a busca da informação de mais de um princípio para categorizar os modelos de visualização da informação. Por exemplo, é possível buscar pelo princípio de “Point of View” através de qualquer um dos critérios de busca, mas o critério de posição é suficiente para organizar o princípio de “Main Application or Content Area”.

Para os princípios que possuem vários tipos, é necessário associar a variação no critério. Por exemplo, critério de cor para o princípio “Type of Thinking Aid” possui as variações de amarelo para “Convergent”, vermelho para “Divergent” e verde para “None”. O critério de área para o princípio “Type of Representation” possui as variações de maior área para “Process” e menor área para “Structure”.

Além disso, outra possível interação que facilita a visualização da informação consiste no mecanismo de *DrillDownUp* ou *Keyboard*. Em casos de Treemaps que contenham muitas informações (dezenas, centenas ou milhares), o recurso possibilita

mostrar somente o grupo de informações de interesse, conforme ilustrado nas Figura 6 e Figura 8.

A Figura 12 apresenta o modelo, proposto nesta dissertação, da Treemap de Visualização da Informação. O modelo agrupa, por default, os modelos por "Main Application or Content" e possibilita a interação com o usuário. No exemplo dessa figura, o usuário solicitou a separação por sub-grupos por "Detail e Overview", tamanho por "Visualization", na qual os modelos caracterizados com visualizações por processos possuem tamanhos maiores que os classificados por estruturas. O critério de cor foi utilizado para separar as informações de "Thinking", em que os modelos de cores amarelas correspondem ao "Convergent", vermelho os "Divergent" e verde os "None".

Assim, para obter todos os modelos de visualização que sejam "Concept Visualization", "Detail and Overview", "Visualization Structure" e "Thinking Divergent", basta buscar no grupo de "Concept Visualization" e subgrupo "Detail and Overview", os modelos que possuem área e tamanho pequenos e cor vermelha. Em uma única visualização apresentam-se todos os modelos por meio dos critérios selecionados pela interação do usuário. Entretanto, para facilitar a visualização de um conjunto específico de modelos, ainda é possível utilizar o mecanismo facilitador de interação *DrillDownUp*, clicando duas vezes no grupo de interesse "Concept Visualization" ou no subgrupo de interesse "Detail and Overview" do mesmo.

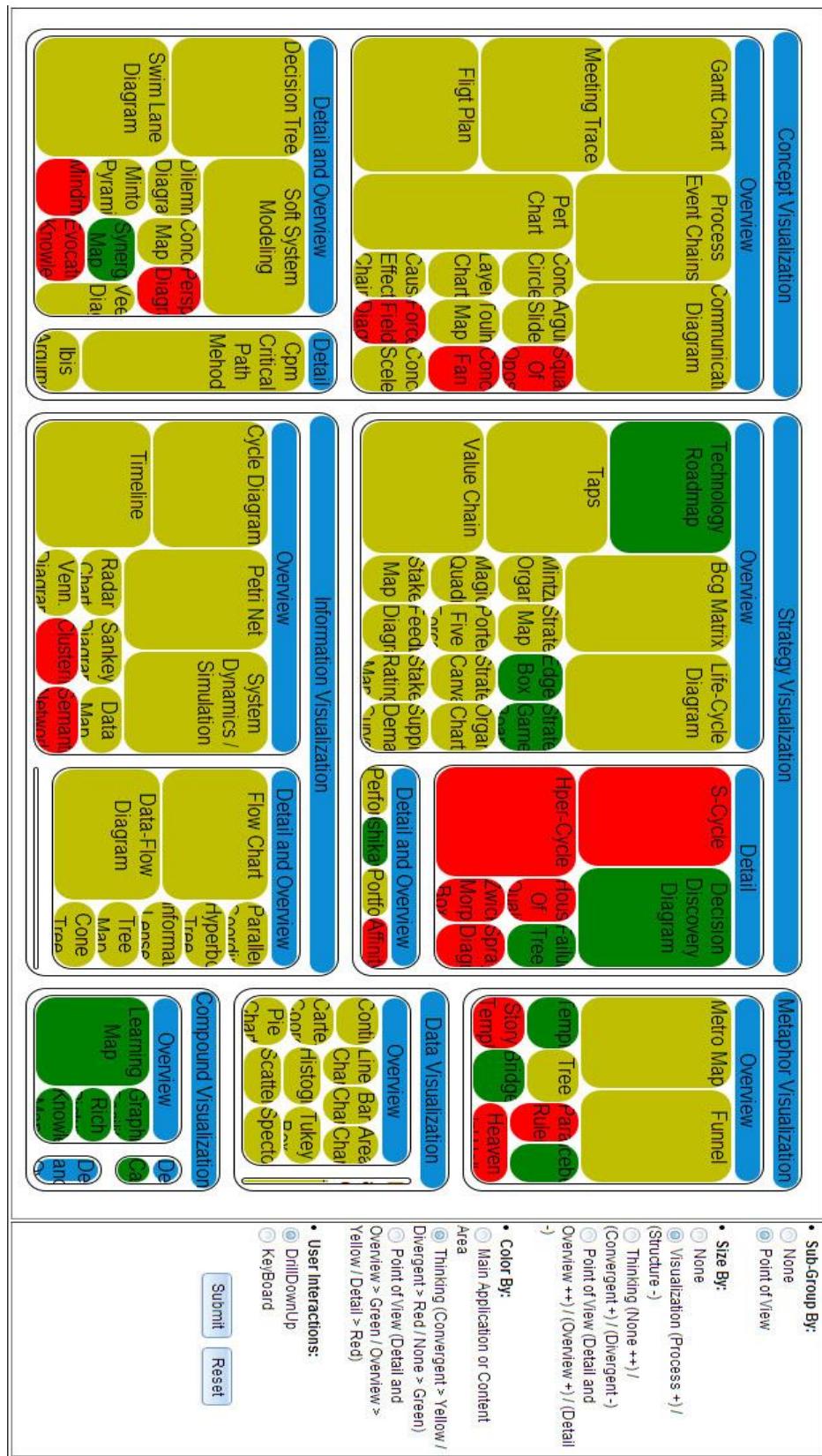


Figura 12. Treemap para Visualização da Informação.

4.2 Metodologia e Detalhes da Implementação

Várias ferramentas de desenvolvimento disponibilizam componentes que possibilitam a criação de *Treemap*, como por exemplo, Java, .NET e Flex. Para o desenvolvimento da ferramenta do metamodelo de visualização baseado em Treemap foram utilizadas as tecnologias Javascript, HTML5 e Dojo Toolkit.

Foram implementados dois *Tab Container*, o primeiro para disponibilizar todas as funcionalidades da solução do metamodelo Treemap (descritas no tópico **4.1 Treemap da Visualização da Informação**) e outro para disponibilizar doze casos de testes intercalando buscas de informações nos metamodelos Treemap e *Periodic Table*.

O desenvolvimento utiliza a biblioteca “dojox/treemap/TreeMap” para a criação da Treemap e a biblioteca “dojox/treemap/DrillDownUp” para o desenvolvimento da interação com o usuário para minimizar ou maximizar grupos e subgrupos da Treemap.

O segundo *Tab Container* contém o desenvolvimento de um procedimento sequencial de doze testes intercalando perguntas para que sejam respondidas com base no modelo Treemap e perguntas a serem respondidas usando a *Periodic Table*. As perguntas são intercaladas e possuem o mesmo nível de dificuldade de busca para ambos os modelos. O processo de teste armazena o tempo gasto pelo usuário para responder cada uma das perguntas e o resultado (correto ou incorreto) de cada uma das questões.

A implementação dos testes buscou selecionar um conjunto de perguntas que envolvam uma maior cobertura dos possíveis caminhos de busca da informação em ambos metamodelos e o uso isolado dos critérios em cada caminho. Isso possibilita realizar conclusões mais assertivas a partir dos resultados de testes.

O sistema também armazena o tempo gasto e o resultado até o final do processo. A Figura 17 mostra o resultado final processado pelo sistema e apresentado ao usuário, quando ele responde a última das doze questões. A partir dos resultados

e do tempo gasto em milissegundos, para cada uma das questões, é possível fazer um comparativo de performance de uso de cada um dos metamodelos.

4.3 Visão Geral da Análise de Algoritmos

Esta seção possui a finalidade de analisar e comparar os processos de busca da informação nas visualizações dos metamodelos apresentados nesta dissertação. O fundamento a ser utilizado consiste na análise de algoritmos, com a hipótese que, assim como esses algoritmos de computadores, que podem ser percorridos logarítmicamente, linearmente e exponencialmente, a percepção humana de uma visualização da informação também pode demorar tempos logarítmicos, lineares e exponenciais dependendo da estrutura e dos critérios de busca visuais disponibilizados para o usuário no modelo de visualização da informação, bem como da ordem de magnitude do número de operações fundamentais que a percepção humana precisa realizar para perceber, entender e abstrair os dados da forma visual.

O número de modelos de visualização da informação é pequeno e a técnica de análise de algoritmos geralmente é feita quando envolve grandes quantidades de informações. Entretanto, o processo cognitivo de um usuário comum pode ser lento para a leitura de determinado modelo, assim como ocorre nos metamodelos da Treemap e Periodic Table, o que viabiliza a análise de algoritmos para quantidades pequenas de informações.

4.3.1 Análise de Algoritmos de Visualização

Este tópico apresenta como finalidade a análise dos algoritmos de visualização da informação de ambos os metamodelos, Periodic Table e Treemap. Hipoteticamente, a análise de algoritmos de visualização da informação se fundamenta nos mesmos princípios da análise de algoritmos computacionais.

A Figura 13 mostra duas listas de números. Não existe ordem na segunda lista, enquanto que a primeira lista está ordenada. A lista ordenada possui coloração de tons de cinza, que possibilita a sua divisão em duas partes iguais e também em quatro partes iguais. Executando visualmente a busca binária do número 211 na

primeira lista, divide a lista em duas outras, do número 1 ao 83 e do número 95 ao 237. Como está ordenada, verifica-se que o número 211 se encontra na segunda lista, resultante da divisão. Ao dividir esta segunda lista em outras duas, do número 95 ao 155 e do número 159 ao número 237, verifica-se que o número ainda se encontra na segunda lista resultante. Ao dividi-la novamente do número 159 ao 206 e do número 211 ao número 237, encontramos o número 211 desejado, na lista resultante.

1	3	7	8	10	11	13	15	18	19	21	23	25	27	31	32	53	61	72	83	95	101	111	121	131	132	141	145	152	155	159	164	169	201	206	211	220	225	236	237
27	159	225	13	1	23	61	132	211	111	101	201	8	19	155	237	10	3	131	145	220	95	7	25	169	18	32	121	11	236	31	15	21	83	152	206	53	164	72	141

Figura 13. Busca binária visual através de dados em lista ordenada e sem ordenação.

Ao buscar visualmente na lista de baixo, a tendência é buscar da direita pela esquerda ou da esquerda pela direita, número a número, o que exige mais comparações da mente humana para encontrar o mesmo número 211. Nesta busca, não se elimina conjuntos de dados indesejáveis, como é feito na busca visual binária. Pode-se dizer que a busca binária tem um desempenho maior e de nível logarítmico, considerando que se eliminam blocos de dados de forma gradual.

É possível fazer análises visuais análogas, com dados de diferentes medidas ou agrupamentos. Veja na Figura 14 as notas dos alunos matriculados em disciplinas e que possuem vinculação (regular ou especial). O gráfico da esquerda mostra o produto cartesiano de cada aluno com a respectiva disciplina, nota e vinculação, enquanto que o gráfico da direita mostra as informações correspondentes, a partir de legendas. As legendas possibilitam ao usuário, visualizar as informações de forma agrupada através de critérios de visualização. No exemplo desta figura, o critério de "cor" é usado para diferenciar as notas dos alunos e o critério de "símbolo" é usado para distinguir a vinculação do aluno na disciplina.



Figura 14. Comparação representativa de informações dos resultados das notas de alunos (CARDOSO, 2014).

Observe que é bem mais rápido responder a pergunta sobre quais alunos especiais obtiveram a nota abaixo de 7.6 a partir do gráfico à direita, pois as informações estão agrupadas e o usuário consegue eliminar grupos de dados irrelevantes ao raciocínio lógico envolvido. O mecanismo de distinção gráfica da direita é necessário para possibilitar agilidade cognitiva e ao mesmo tempo suficiente para resolver o problema. Enfim, estes critérios possibilitam a busca logarítmica em contrapartida da busca combinatória do gráfico à esquerda.

Caso o usuário confunda os símbolos (por exemplo, quando os símbolos são bem parecidos) ou caso o usuário confunda as cores (por exemplo, quando ele é dautônico), o gráfico da direita perde a condição de busca logarítmica, aproximando do gráfico da esquerda com relação ao quesito de desempenho.

A mesma análise pode ser realizada nos modelos da treemap e na tabela periódica da informação. Considere os $n=28$ modelos de VI existentes na categoria "Strategy Visualization", segundo o metamodelo da *Periodic Table*, mostrados na Figura 15. Para buscar todos os modelos com o princípio "Visualization" do tipo "Process" é suficiente contar os $m=6$ modelos segundo o critério de cor azul correspondente a esse princípio ("life-cycle diagram", "s-cycle", "techmology roadmap", "decision discovery diagram", "value chain" e "hype-cycle"). Assim, a complexidade de tempo para extrair essa informação é $O(m)$, pois deve-se percorrer visualmente esses " m " elementos. Esse critério de busca através de cor é bem usado pela percepção humana.

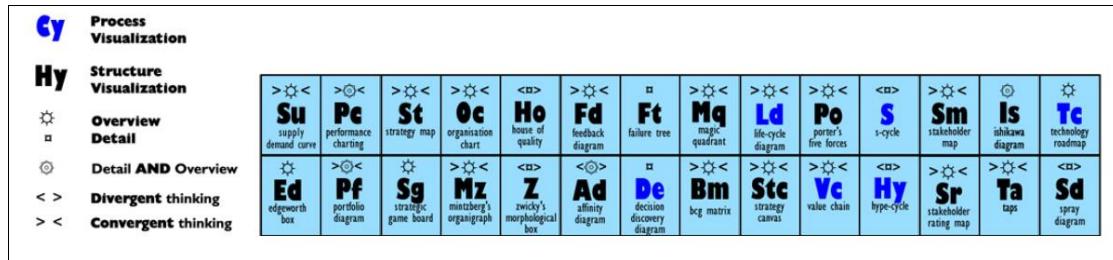


Figura 15. Analise de algoritmo da “Strategy Visualization” da “Periodic Table” (LENGLER; EPPLER, 2007).

Entretanto, para buscar todos os modelos com o princípio “Point of View” do tipo “Overview” e ao mesmo tempo com o princípio “Thinking” do tipo “Convergent” é necessário percorrer todos os modelos da “Strategy Visualization” para verificar se o símbolo de cada modelo corresponde ao da legenda dos critérios a serem buscados. Após verificar para o primeiro critério, deve-se percorrer todos os modelos do conjunto desse critério e delimitar quais também atendem ao segundo critério. Assim, é necessário percorrer todos os $n=28$ modelos, delimitando $o=17$ modelos do tipo “Overview” e, logo em seguida, percorrer os $o=17$ modelos para verificar $l=14$ contém também o princípio “Thinking” do tipo “Convergent”. A complexidade de tempo desse algoritmo é de $O(m) + O(n)$. Como $m > n$, ou seja, m é o pior caso, a complexidade pode ser simplificada para $O(m)$.

O mesmo estudo de caso utilizado na análise do metamodelo da *Periodic Table* será utilizado na análise do metamodelo da Treemap, a seguir. Entretanto, os critérios de visualização dos metamodelos são diferentes, pois enquanto a *Periodic Table* utiliza o critério de símbolo para buscar por determinados princípios, a Treemap utiliza somente os critérios de posição, área e cor para qualquer princípio.

Neste metamodelo da Treemap, utilizando o critério de cor para “Point of View” e posição ou área para “Thinking” como mostrado na Figura 16, basta contar o número de áreas amarelas com tamanhos intermediários correspondente a $l=14$. A complexidade de tempo desse algoritmo é de $O(l)$. Como cada critério está organizado em árvore, “ l ” corresponde a uma parte logarítmica de “ n ” e a complexidade de tempo de algoritmo de visualização é menor no metamodelo Treemap comparado ao metamodelo Periodic Table, para a busca desse exemplo proposto, pois $l < m$.



Figura 16. Análise de algoritmo da Strategy Visualization da Treemap.

4.3.2 Análise de Algoritmos de Visualização dos Metamodelos

Calcular a complexidade de tempo de algoritmo de visualização de cada metamodelo é o mesmo que calcular a complexidade de tempo correspondente a cada critério utilizado nas possíveis buscas pelos modelos existentes.

Como o metamodelo da Tabela Periódica utiliza os critérios de grupo ou posição (para "Main Application or Content Area"), cor (para "Type of Representation") e símbolo (para "Type of Thinking Aid" e "Point of View"), logo a complexidade de tempo para os dois primeiros critérios é logarítmica, mas para o critério de símbolo é linear. Seja "x" o número médio de itens de cada conjunto de elementos que contém o critério de grupo, "y" o número médio de itens de cada conjunto de elementos que contém o critério de cor e "z1" e "z2" os números médios de itens de cada conjunto de elementos que contém os critérios de símbolo, então a complexidade de tempo para busca neste metamodelo é $O(z1 + z2 + \log(x) + \log(y))$.

Já o metamodelo da Treemap utiliza os critérios de grupo ou posição (para "Main Application or Content Area" e "Point of View"), cor (para "Main Application or Content Area", para "Point of View" e para "Type of Thinking Aid") e área ou tamanho (para "Point of View", para "Type of Representation" e para "Type of

Thinking Aid"), mostrado na Tabela 2. Seja "x" o número médio de itens de cada conjunto de elementos que contém o critério de grupo, "y" o número médio de itens de cada conjunto de elementos que contém o critério de cor e "z" o número médio de itens do conjunto de elementos que contém o critério de área, então a complexidade de tempo para busca neste metamodelo é $O(\log(x) + \log(y) + \log(z))$. A complexidade de tempo do metamodelo da Treemap é menor que a do metamodelo da *Periodic Table* e, portanto, apresenta melhor desempenho de qualquer busca de dados na visualização da informação.

4.3.3 Análise de Resultados de Testes

A avaliação de técnicas de VI é necessária para verificar a eficiência destas técnicas em tarefas de apoio dos usuários, bem como verificar as suas capacidades para mostrar informações importantes que facilitam a análise, interpretação e compreensão do conjunto de dados. Tal avaliação é, acima de tudo, importante para levar a desenvolvimento de novas técnicas.

In the last few years, the evaluation of visualization techniques has been addressed explicitly by the information visualization community [2][20]. Such evaluation is necessary in order to verify (1) the efficiency of these techniques as support for users' tasks and (2) if they are able to show relevant information, which make easy to analyze, interpret, and comprehend the datasets. Such evaluation is important, over all, to drive the development of new techniques. (Pillat; Valiati; Freitas, 2005)

Nesta dissertação, construiu-se uma ferramenta WEB de testes a fim de realizar experimentos e coletar dados para análise dos resultados.

O teste foi aplicado a 20 usuários, dos quais 75% destas pessoas realizaram o teste presencial e o restante realizou-o online.

No teste, o usuário deve responder a 12 perguntas relacionadas na Figura 17. Cada pergunta avalia um ou mais critérios de visualização. Após responder todas as perguntas, o sistema mostra o resultado do teste informando a quantidade de erros, o

tempo gasto para responder a pergunta, o metamodelo de visualização e os critérios envolvidos em cada pergunta.

Number	Question	Answer	Expected Answer	Result	Answer Time	Visualization Information	Criteria
1	Quantas visualizações da informação possui na 'Information Visualization'?	9	18	Incorrect	5357	Treemap	group
2	Quantas visualizações da informação possui a 'Metaphor Visualization'?	9	9	Correct	2179	Periodic Table	group
3	Quantas visualizações do tipo 'Visualization - Process' existem na 'Strategy Visualization'?	9	8	Incorrect	1988	Treemap	group, (size - position)
4	Quantas visualizações do tipo 'Visualization - Process' existem na 'Concept Visualization'?	9	10	Incorrect	1664	Periodic Table	group, cor
5	Quantas visualizações do tipo 'Thinking - Divergent' existem na 'Concept Visualization'?	9	5	Incorrect	2663	Treemap	group, (size - position)
6	Quantas visualizações do tipo 'Thinking - Divergent' existem na 'Strategy Visualization'?	9	6	Incorrect	1472	Periodic Table	group, symbol
7	Quantas visualizações do tipo 'Point of View - Detail' existem na 'Concept Visualization'?	9	2	Incorrect	1648	Treemap	group, color
8	Quantas visualizações do tipo 'Point of View - Detail' existem na 'Information Visualization'?	9	1	Incorrect	1527	Periodic Table	group, symbol
9	Quantas visualizações com os tipos 'Point of View - Detail and Overview', 'Visualization - Structure' e 'Thinking Divergent' existem na 'Concept Visualization'?	9	3	Incorrect	1776	Treemap	group, (size - position), color
10	Quantas visualizações com os tipos 'Point of View - Detail and Overview', 'Visualization - Structure' e 'Thinking Convergent' existem na 'Concept Visualization'?	9	4	Incorrect	1327	Periodic Table	group, symbol, color
11	Quantas visualizações com os tipos 'Point of View - Overview' e 'Thinking None' existem no modelo inteiro?	9	10	Incorrect	2688	Treemap	size, color
12	Quantas visualizações com os tipos 'Point of View - Overview' e 'Thinking Divergent' existem no modelo inteiro?	9	8	Incorrect	1824	Periodic Table	symbol, symbol

Thank you for cooperate.

Figura 17. Resultado dos Testes – Resposta do Questionário.

Além disso, o teste foi dividido intercalando perguntas entre o metamodelo Treemap e o metamodelo *Periodic Table*, ou seja, as perguntas impares são referentes à Treemap e as perguntas pares são referentes à *Periodic Table*. Cada par de perguntas representa a mesma tarefa, usando diferentes meios (metamodelo da Treemap e *Periodic Table*), mas que evita a influência da resposta da primeira pergunta com relação à segunda. Ou seja, o usuário, ao resolver uma pergunta do par através de um modelo, não implica que ele resolva a segunda questão do par mais rapidamente, pois não se trata do mesmo problema, mas apenas diferentes situações com mesma dificuldade de solução.

A Figura 18 apresenta o percentual de erro de cada pergunta. Assim, é possível realizar a comparação entre os erros gerados para os metamodelos Treemap e *Periodic Table*. As duas primeiras perguntas são simples e, por isso, não resultou em erros nos testes dos usuários. Pode ser observado que em todas as perguntas, o metamodelo TreeMap obteve melhor resultado em comparação à *Periodic Table*.

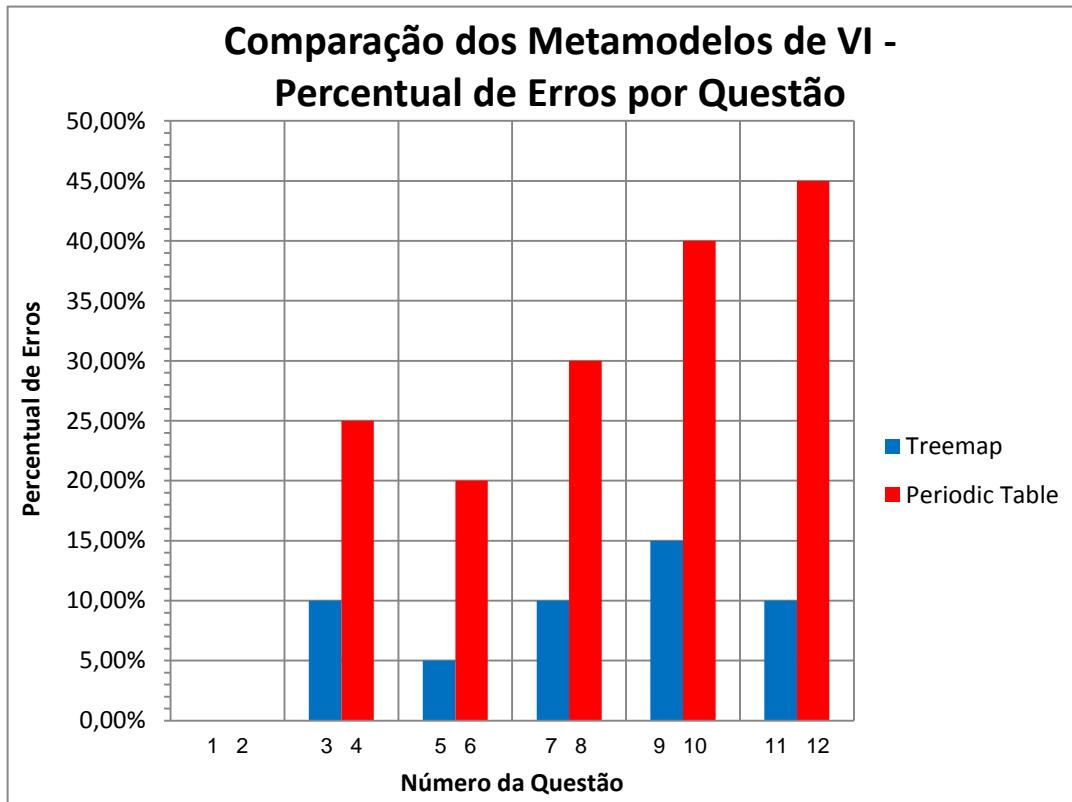


Figura 18. Comparação entre os Metamodelos TreeMap e *Periodic Table* – Percentual de Erros por Pergunta.

A Figura 19 apresenta os resultados do percentual de erros para cada um dos dois metamodelos segundo os critérios utilizados nos testes. Os critérios avaliados foram: *group*, *color*, *symbol* e *size* ou *position*, sendo que o metamodelo Treemap não usa o critério de *symbol* e a *Periodic Table* não usa o critério de *size* ou *position*.

Através dos resultados, pode ser verificado que o metamodelo da TreeMap, em todos os critérios, teve menor percentual de erros do que a *Periodic Table*. Além disso, pode ser feita uma análise por critério e observa-se que aqueles relacionados a *color*, *size* e *position* são melhores para a percepção humana que os critérios de *symbol*.

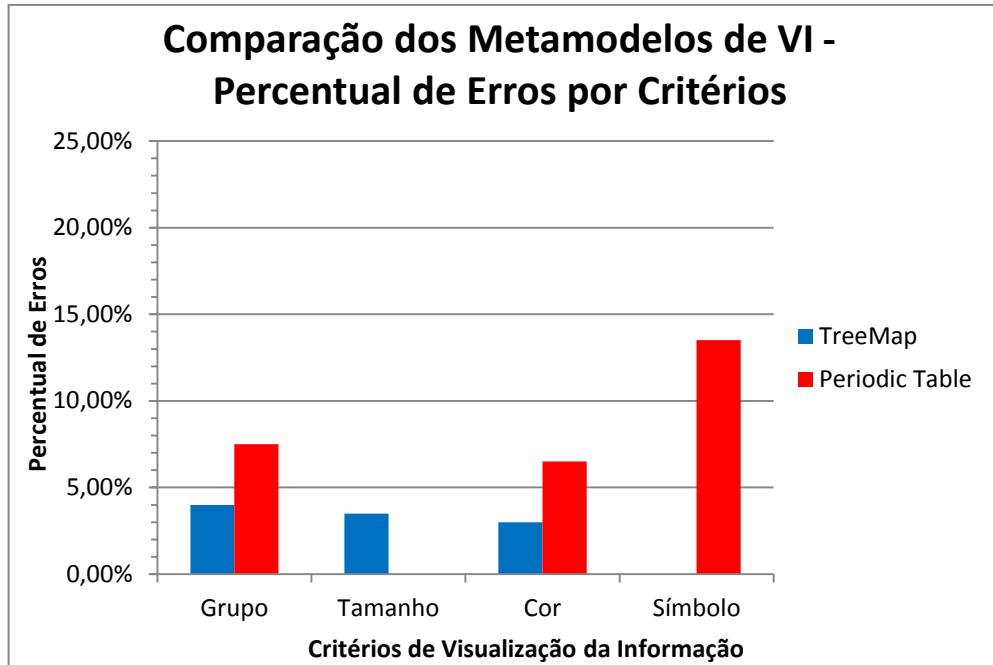


Figura 19. Comparação entre os Metamodelos TreeMap e *Periodic Table* – Percentual de Erros por Critérios de Visualização.

Através do cálculo do tempo gasto para responder cada pergunta é possível avaliar também o tempo da percepção humana para cada modelo de visualização.

A Figura 20 mostra o tempo gasto da percepção humana para responder cada pergunta, com base nos testes realizados. Observa-se que gasta-se mais tempo para responder a *Periodic Table* em todas as perguntas correspondentes do que para as questões da TreeMap, e que para as perguntas 11 e 12 que comparam critérios de *symbol* da *Periodic Table*, o tempo gasto é logarítmico para TreeMap e linear para a *Periodic Table*, pois para encontrar a resposta da pergunta 12 referente a *Periodic Table*, o usuário deverá percorrer cada elemento do modelo.

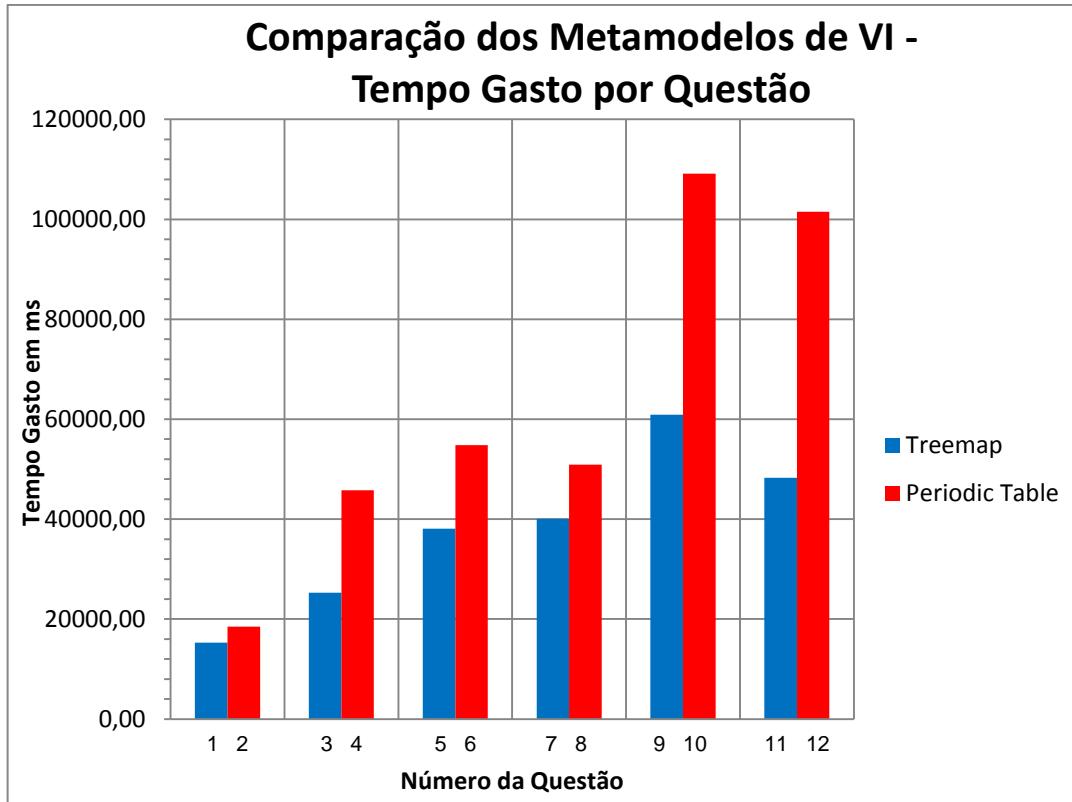


Figura 20. Comparação entre os Metamodelos TreeMap e *Periodic Table* – Tempo de Resposta por Pergunta.

Outra análise possível de ser verificada, conforme apresentada no gráfico da Figura 21, é com relação ao tempo gasto para responder as perguntas por critério de visualização de informação. Podemos observar que os tempos de respostas maiores de percepção foram gastos para os critérios de *color* e *symbol*, devido ao usuário confundir a cor ou o símbolo a ser verificado. O menor tempo de resposta foi gasto para o critério de *group*, devido à facilidade no processo de busca da informação através desse critério. Os critérios de *size* e *position* tiverem resultados semelhantes. Nesta outra visão de gráfico também se verifica que o tempo gasto de resposta para o metamodelo TreeMap em todos os critérios apresentaram-se menores que o metamodelo *Periodic Table*.

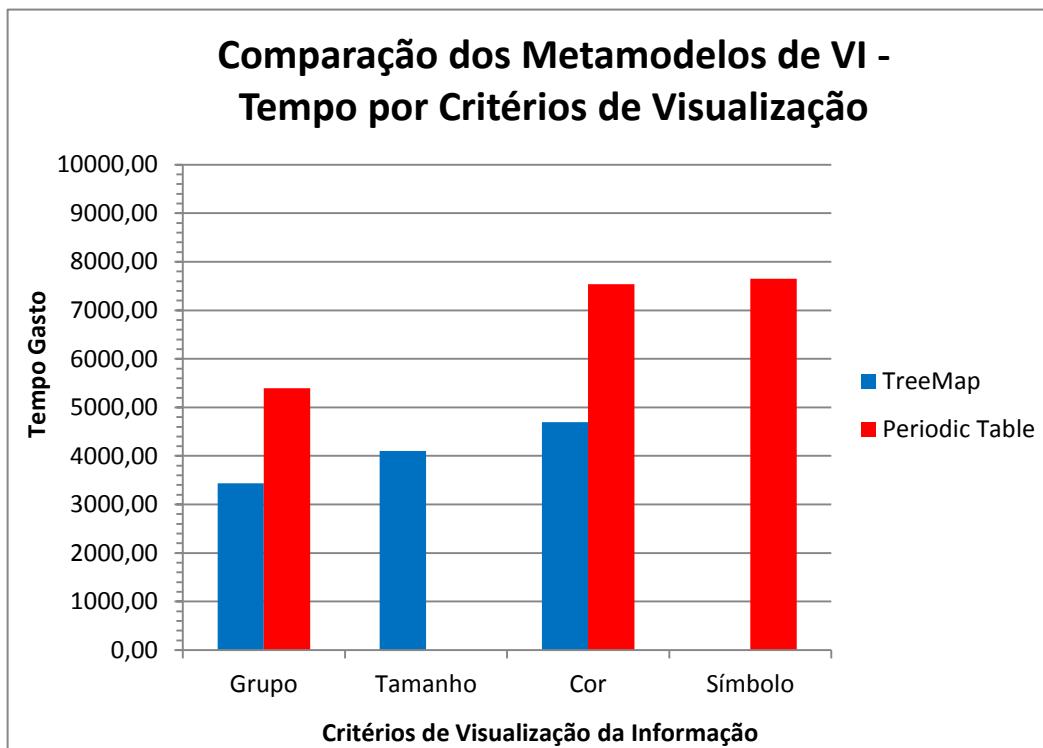


Figura 21. Comparação entre os Metamodelos TreeMap e *Periodic Table* – Tempo de Resposta por Critério.

Capítulo 5

Componentes para Visualização da Informação em Sistemas WEB

Construção de sistemas WEB requer criação de componentes WEB para a visualização da informação, para interagir e possibilitar a navegação do usuário.

Os sistemas WEB são implementados com a utilização de várias linguagens de desenvolvimento, seguem padrões de protocolos de comunicação e outros padrões utilizados para a navegação na internet. Assim, é necessário definir a arquitetura dos componentes e subsistemas para garantir a sua organização e integração, os quais são importantes para conceber o sistema WEB e consistirem no fundamento básico no desempenho do projeto. A definição de arquitetura utilizado nesta dissertação baseia-se nos conceitos apresentados no Anexo 1.

Não é intuito desta dissertação, revisar todos os modelos do metamodelo proposto, mas é conveniente analisar e desenvolver alguns componentes de determinados modelos, em busca de um raciocínio que demostre que alguns modelos podem ser mais adequados que outros, em determinadas situações e objetivos do usuário.

Este capítulo mostra os componentes e uma arquitetura suficientes para a concepção de um sistema WEB com a visualização da informação voltada para pesquisa de dados.

5.1 Modelo de Pesquisa em Lista de Dados

O modelo de pesquisa em lista de dados é importante nas buscas em sistemas WEB e deve ter um desempenho aceitável para qualquer componente que realize tal busca.

Nada é mais desgastante do que tentar entrar em uma WebApp e receber uma mensagem do tipo "servidor ocupado", com a sugestão para você tentar mais tarde. Nada é mais desconcertante do que uma WebApp que responde instantaneamente em algumas situações e depois parece entrar em um estado de espera infinita em outras. Todas essas ocorrências acontecem diariamente na Web e todas estão relacionadas a desempenho. (PRESSMAN, 2011, p. 485)

Existem vários componentes gráficos para o modelo de pesquisa de lista de dados. Outros modelos de visualização da informação de dados muito usados consistem nos relatórios gráficos. Este tópico descreve alguns componentes gráficos do modelo de pesquisa de lista de dados e de relatórios gráficos especializados para solucionar determinados problemas que os envolvem.

Existem também vários frameworks em *javascript* para desenvolvimento de componentes WEB. Alguns contêm código aberto, como por exemplo, JQuery e Dojo. Nesta dissertação, o framework escolhido consiste no Dojo, pois ele oferece diversos componentes já implementados e possibilita a atualização ou criação de novos outros. Além disso, a sua filosofia de programação é bastante simples e ao mesmo tempo robusta.

Although the breadth and depth of Dojo produces a lot of complexity, the infrastructure is painstakingly reviewed on a continual basis by some of the best web hackers in the world for high quality coding standards, consistent naming conventions, performance, maintainability, and overall

ease of use for the application developer. (MATHEW RUSSELL, 2008, p. XVIII)

5.1.1 Compartente de Multiseleção

Componentes de multiseleção consistem em mecanismos que possibilitam a busca e também a seleção de vários itens em um conjunto de informações. Um exemplo é o componente *picklist*.



Figura 22. Compartente de multiseleção customizado do Dojo.

Um dos maiores desafios da visualização da informação consiste na criação de componentes que ocupam pouco espaço na tela. Isso ocorre principalmente quando se trata de sistemas móveis, cuja resolução da tela é menor que dos monitores convencionais.

Existem funcionalidades cuja seleção é múltipla, mas a busca necessita somente de uma propriedade ou nome do negócio a ser selecionado. A Figura 22 mostra um exemplo de compartente de multiseleção que ocupa pouco espaço na tela. Ele é bastante eficiente para pesquisas simples, que não envolvem filtros de pesquisa e que necessitam de tela de menor resolução.

5.1.2 Compartente de Paginação

Compartente de paginação é um mecanismo que possibilita a busca em demanda e a seleção de vários itens em um conjunto de informações. A busca por demanda é bastante eficiente, pois busca em um conjunto de informações e retorna os dados pesquisados por solicitação de demanda.

A Figura 23 apresenta o gráfico do componente de paginação sob demanda do Dojo para pesquisar dados do *prospect* através dos filtros de “Nome” e “Email”. Ele mostra que foi ordenado pelo resultado de pesquisa “Nome” de forma descendente.

O maior problema do componente de paginação consiste na busca por demanda de determinados dados combinados e a manutenção desses dados, controlando a sua inclusão, deleção e edição sem perder a busca por demanda.

The screenshot shows a web-based application titled "Prospect". At the top, there are search fields for "Nome" and "Email", and a "Novo" button. Below the search area is a table displaying two rows of prospect data. The columns are labeled "Nome", "E-mail", and "Data cadastro". The first row has "Nome" as "teste 2", "E-mail" as "teste2@yahoo.com.br", and "Data cadastro" as "14/04/2014". The second row has "Nome" as "teste", "E-mail" as "teste@yahoo.com.br", and "Data cadastro" as "03/04/2014". At the bottom of the page, there is a navigation bar with links for "1 - 2 de 2 ítems", "17 | 34 | 68 | 100 | Todos", and a set of icons for navigation.

Figura 23. Componente de paginação sob demanda do Dojo.

Para solucionar esse problema é necessário projetar duas visões e criar um ponto de integração entre elas. Utilizando o framework MVC da Microsoft .net, a divisão deste componente consiste em visão gráfica do componente e visão arquitetural do componente.

5.1.2.1 Visão Gráfica do Componente

Os requisitos visuais do componente e de interação com o usuário são projetados pela visão gráfica do componente. Por exemplo, em um banco de dados com milhões de pessoas, o usuário pode filtrar a pesquisa desejada e o sistema envia o resultado em demandas paginadas de 50 a 50 de pessoas, possibilitando a ele

modificar a quantidade da demanda, solicitar o número da página a ser demandada, selecionar itens em várias páginas, ordenar por determinada coluna da grade de paginação, separar as linhas pares das linhas ímpares para melhor visualização, etc.

Como o resultado de pesquisa é por demanda, a pesquisa no banco de dados se trata somente dos 50 registros paginados. Separar as linhas pares das linhas ímpares não tem integração com o banco de dados, ou com as camadas *model* e *controller* do ASP.NET MVC, mas os requisitos que mudam a configuração da quantidade de páginas ou da ordenação na paginação fazem integração com o banco de dados. Para esses requisitos, é necessário que a visão gráfica do componente se comunique, através de requisições, com o banco de dados, estabelecendo uma visão arquitetural do componente.

5.1.2.2 Visão Arquitetural do Componente

O framework gráfico Dojo é fácil de integrar com o ASP.NET MVC e também como Web API (Interface de Programação de Aplicativos). O framework do Dojo é composto basicamente por arquivos javascript, html e arquivos *Cascading Style Sheets* (CSS) de configuração de estilos. Ele se encontra na camada *view* da arquitetura ASP.NET MVC e depende da comunicação com os controles .net para fazer a busca paginada no banco de dados. Nesta arquitetura .NET existem dois tipos de controles, o controle do ASP.NET MVC e o controle do Web API. Tais controles são diferentes e as informações de coluna de ordenação da grade, direção da ordenação (crescente ou decrescente) e intervalo de pesquisa provenientes do Dojo, embora estejam na requisição (variável *Request*) do protocolo HTTP, se posicionam em locais distintos.

No Web API, as variáveis de paginação do Dojo encontram-se no grupo do cabeçalho da requisição *Request.Headers.Range* e nos parâmetros do identificador universal de recurso *Request.RequestUri.ParseQueryString()*. Já no ASP.NET MVC encontram-se no próprio cabeçalho da requisição *Request.Headers* e nos parâmetros do modelo GET *Request.Params*. Kurtz (2013) indica como pode ser feito a integração das variáveis de paginação, descrevendo o funcionamento das mensagens no cabeçalho de requisição.

You have access to the request headers, the body, the URL used to invoke the call, client certificates, and many other valuable properties. You can also use the `HttpRequestMessage` object to create a response that is pre-wired to the given (...) (KURTZ, 2013, p. 65)

Leslie Orchard (2009) mostra mais detalhado que o cabeçalho de requisição consiste em um *array* de informações que pode ser manipulado. Estes conceitos são suficientes para manipular esta estrutura e fazer a integração de buscas *JsonRest* do Dojo com o .NET.

Either an object, "`{header: value}`" or an array (even number indexes hold the name of the header and odd indexes hold the value) representing headers to be sent along with the request. Prototype provides 4 defaults headers that you can override to meet your needs. (LESLIE ORCHARD; PEHLIVANIAN; KOON; JONES, 2009, p. 35)

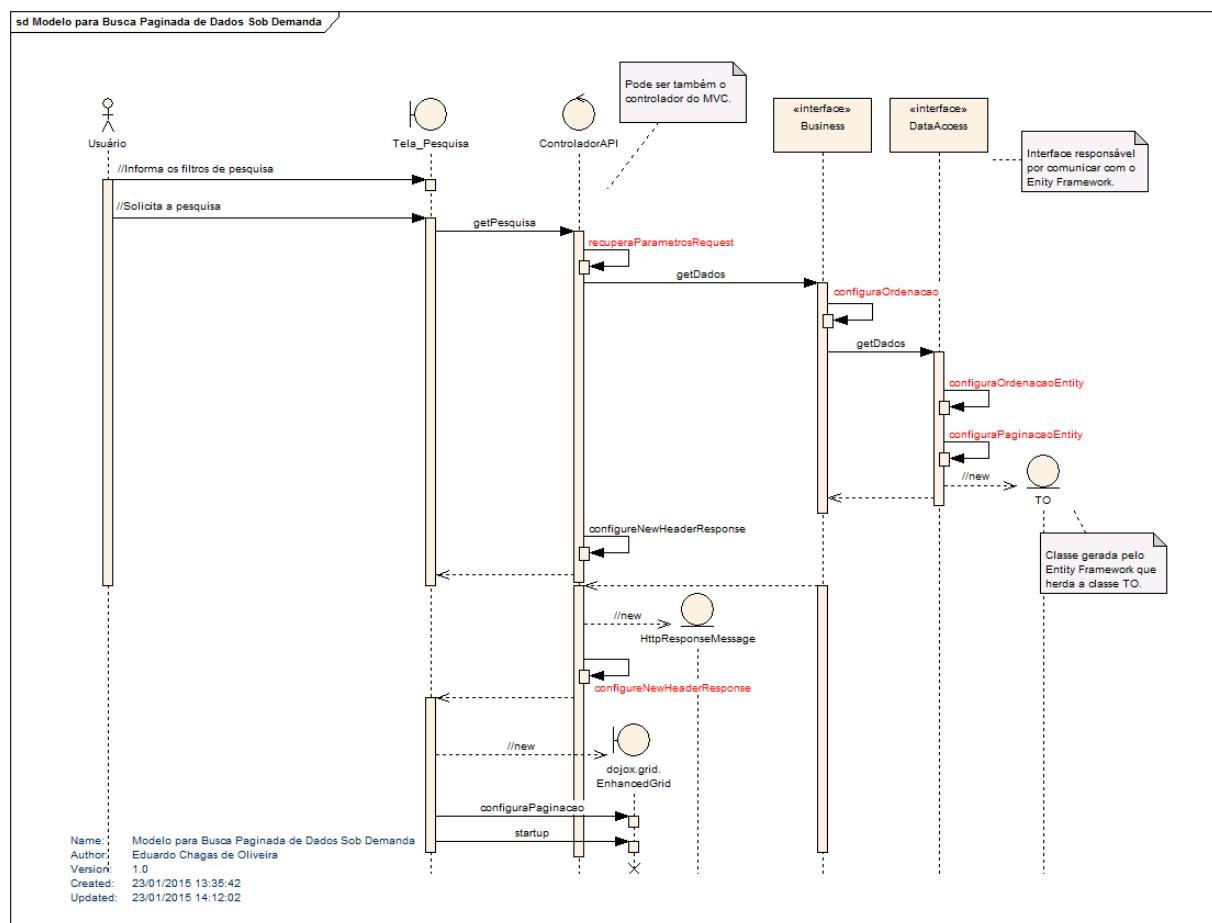


Figura 24. Diagrama de sequência do modelo de busca paginada de dados sob demanda.

Obtendo essas variáveis na camada de controle, é possível tratar na camada de negócio por qual variável será ordenado, caso o usuário não solicitou ordenação na grade gráfica ou quando é a primeira renderização da grade. Na camada de acesso aos dados, usando o Entity Framework, é possível usar as variáveis de paginação nos seus modelos *Skip* e *Take* de pesquisa no banco de dados, para delimitar os registros que serão pesquisados e retornados para o corpo da requisição. A Figura 24 mostra o diagrama de sequência do modelo de busca paginada de dados sob demanda. Os métodos em vermelho foram tratados para que a pesquisa seja sob demanda no banco de dados.

5.1.3 Componente Gráfico em Hierarquia

Muitos dados podem ser organizados em hierarquia. No seu processo de estruturação para disponibilizá-lo visualmente, o usuário aprende, conhecendo as relações das informações em um nível mental e cognitivo de entendimento dessas informações. O componente gráfico *dojox.grid.LazyTreeGrid* no framework do Dojo é um exemplo desse recurso que possibilita a manipulação das informações gráficas disponibilizadas em hierarquia.

A Figura 25 mostra a disponibilização das informações do desempenho dos alunos por avaliação e formas de avaliar, para que o usuário, neste caso, o professor, possa informar a data da atividade avaliativa, observação e nota para cada aluno.

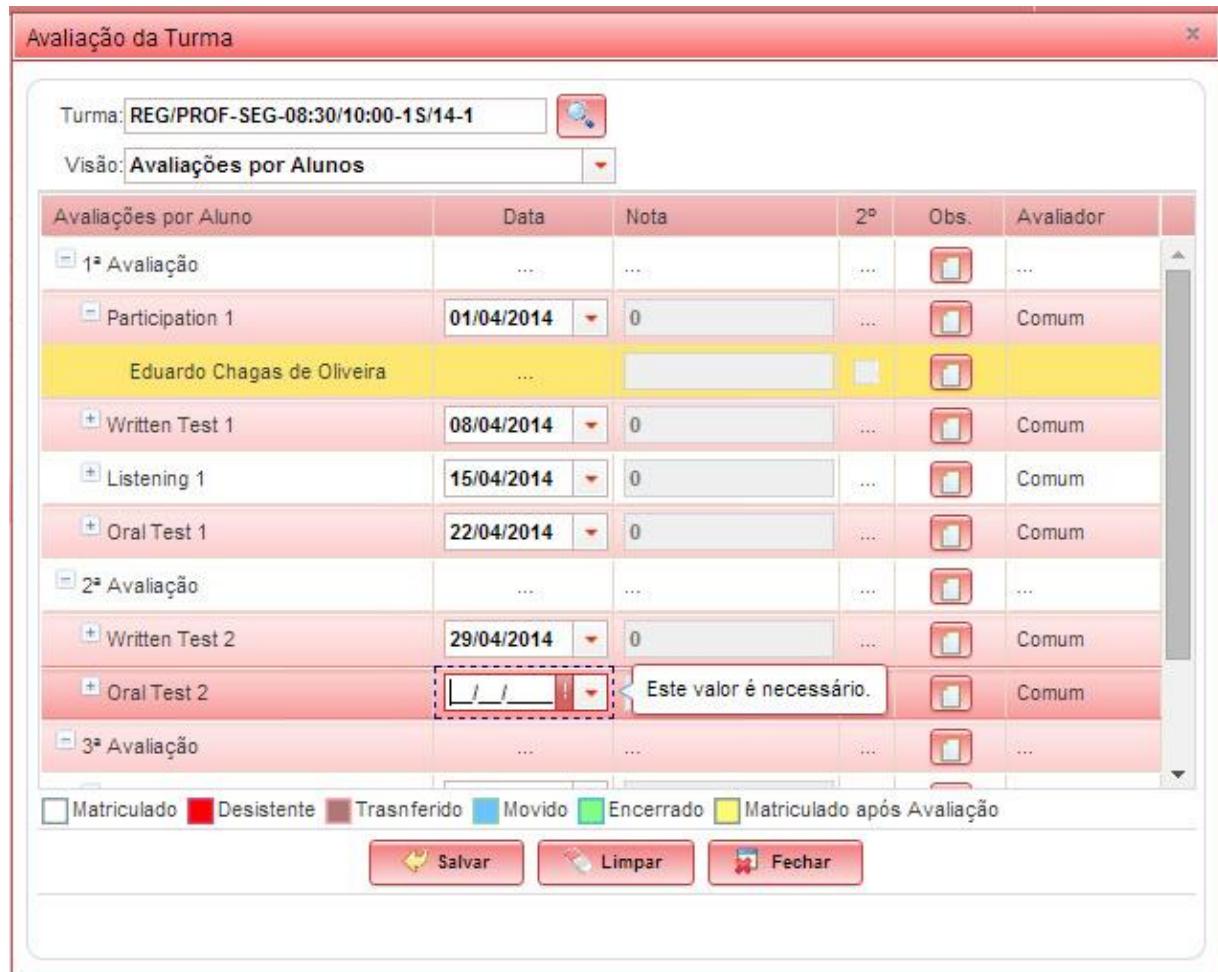


Figura 25. Componente gráfico em hierarquia customizado do Dojo.

O componente gráfico de hierarquia auxilia no lançamento das notas dos alunos em cada avaliação, além de mostrar a situação desse aluno em cada atividade avaliativa pela legenda.

5.2 Modelo *Table*

O modelo “*Table*” consiste em uma forma de visualização que permite ao usuário buscar a informação de forma tabular. O componente gráfico de horários consiste em um exemplo deste modelo que foi desenvolvido no sistema WEB SGF.

5.2.1 Componente Gráfico de Horários

O planejamento de tarefas diárias, semanais e mensais é muito comum em sistemas WEB. A visualização da informação das tarefas de forma gráfica, na qual se possa observar e manipular os horários entre as tarefas pode facilitar o trabalho de

planejamento do usuário, principalmente quando existe paralelismo concorrente entre as tarefas.

O componente gráfico de horários mostra as informações buscadas e obtidas na forma de lista, próximo ao modelo de tabela, mas ordenado por horários em uma grade horária.

A Figura 26 mostra diversas tarefas que são executadas em uma sala de aula, a ocupação de horários por professores e alunos e também alunos de outras turmas na mesma sala. O sistema facilita ao usuário o controle do remanejamento de horários das turmas na mesma sala ao usuário, através das possíveis interseções ocasionadas no visual gráfico do componente de grade de horários do Dojo. O componente utilizado do Dojo consiste no *dojox.calendar* e foi customizado para não apresentar o cabeçalho especificando determinada semana do mês, podendo ser qualquer uma.

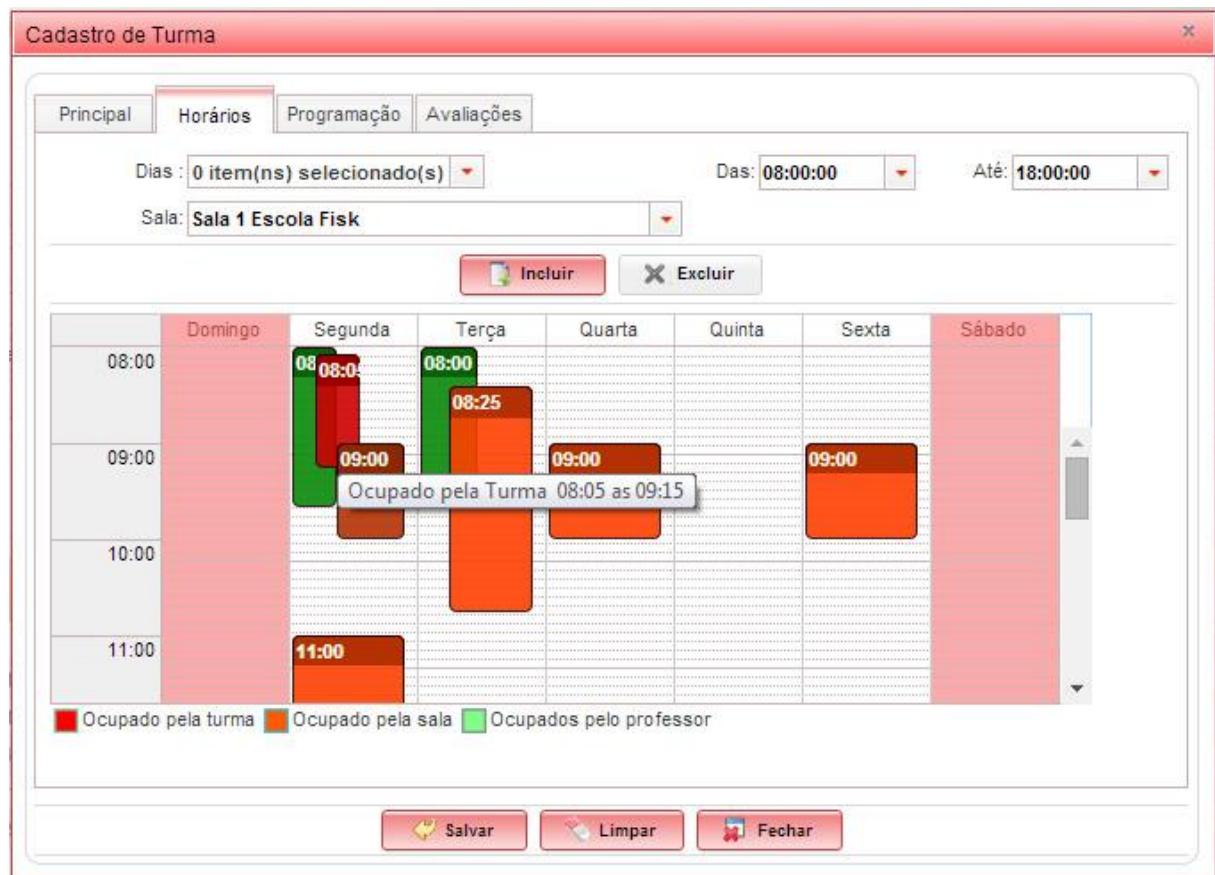


Figura 26. Componente gráfico de horários customizado do Dojo.

Este componente tem uma forma gráfica que ajuda o usuário abstrair e entender os processos cognitivos de inclusão e interseção de horários com muita facilidade, a qual é possível por causa da sua forma e organização gráfica, mostrados lado a lado na grade horária, mantendo o início e fim de cada um. Isso, junto à legenda do horário, possibilita o processo cognitivo de identidade e comparação entre eles.

5.2.2 Componente Gráfico de Relatórios

Outra forma de desenvolver o modelo “Table” pode ser através de emissão de relatórios nos sistemas WEB. Eles podem também mostrar modelos como “Pie Chart”, “Bar Chart”, “Line Chart”, “Area Chart”, entre outros.

Os resultados de pesquisa de Business Intelligence (BI) e também resultados de pesquisa de busca em lista podem ser convertidos para o componente gráfico de relatório. Alguns componentes de relatório como o Dojo suportam atualização dos dados por demanda. Isso torna possível a visualização da informação de forma progressiva, ou seja, na medida em que ela é processada.

No Dojo isso é possível, graças à interpolação de diferentes conjuntos de dados gráficos. A Figura 27 mostra a visualização gráfica em quatro momentos sequenciais da evolução do algoritmo genético que busca a minimização da função

$$f(x) = -((\text{sen}(x - 256))^2 - 0.5)/(1 + 0.001 * (x - 256)^2), \text{ na qual } x \text{ está no intervalo de } [0, 512].$$

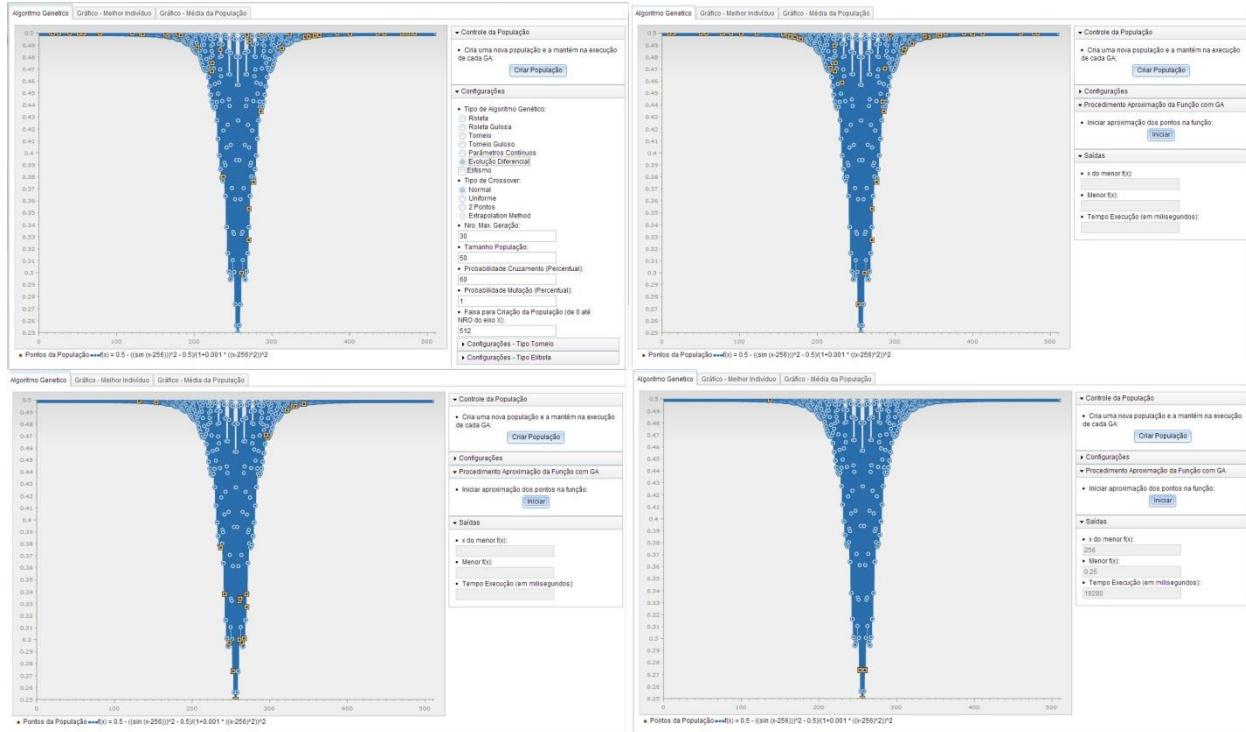


Figura 27. Componente gráfico de relatório do Dojo.

Esse relatório possibilita uma visualização analítica da evolução do algoritmo ao buscar o mínimo da função. É possível visualizar analiticamente a conversão dos pontos de mínimos encontrados durante a evolução do algoritmo genético para o mínimo da função e verificar o desempenho desse algoritmo na resolução desse problema.

A evolução do algoritmo genético é feita de forma interativa e o resultado é colocado no gráfico no final de cada interação, através da função assíncrona "setTimeout" do Javascript. De forma integrada, as bibliotecas "dojox/charting/plot2d/Lines" e "dojox/charting/plot2d/MarkersOnly" do Dojo possibilitam incluir e interpolar gráficos em tempo de execução, com os resultados de cada interação do algoritmo genético.

Outra ferramenta que pode ser utilizada junto ao Dojo, consiste no Report Service. Esta ferramenta possui fácil integração com as bibliotecas da Microsoft, em especial com o Entity Framework, responsável pelas consultas no banco de dados.

Numa arquitetura de sistemas, os componentes de relatórios devem estar empacotados separadamente em um subsistema de apresentação. No paradigma de

programação .NET, este subsistema constitui um projeto WEB e pode ser colocado em um *pool* de aplicação diferente do *pool* de aplicações do sistema, no servidor WEB IIS. O pool de aplicações é responsável pelo gerenciamento de aplicações e quando um *pool* de aplicações está comprometido, não influencia nos demais. Isto constitui em uma vantagem funcional da arquitetura.

Nos sistemas WEB SGF, Cemig e Freeman, a arquitetura projeta os subsistemas de relatório separados dos demais subsistemas e, além disso, foi criado um subsistema Web, para renderização dos relatórios. Isso foi feito para que seja possível separar as aplicações de relatórios em uma máquina dedicada, ou seja, separada do servidor de aplicações e do banco de dados. Assim, quando o usuário solicita algum relatório, o servidor de aplicações recupera os filtros do relatório, criptografa e faz uma requisição ao servidor de relatórios, responsável por criar e emitir os relatórios. A Figura 28 mostra a sequência completa do padrão de emissão de relatórios dos sistemas WEB SGF, Cemig e Freeman, contemplando a arquitetura MVC.

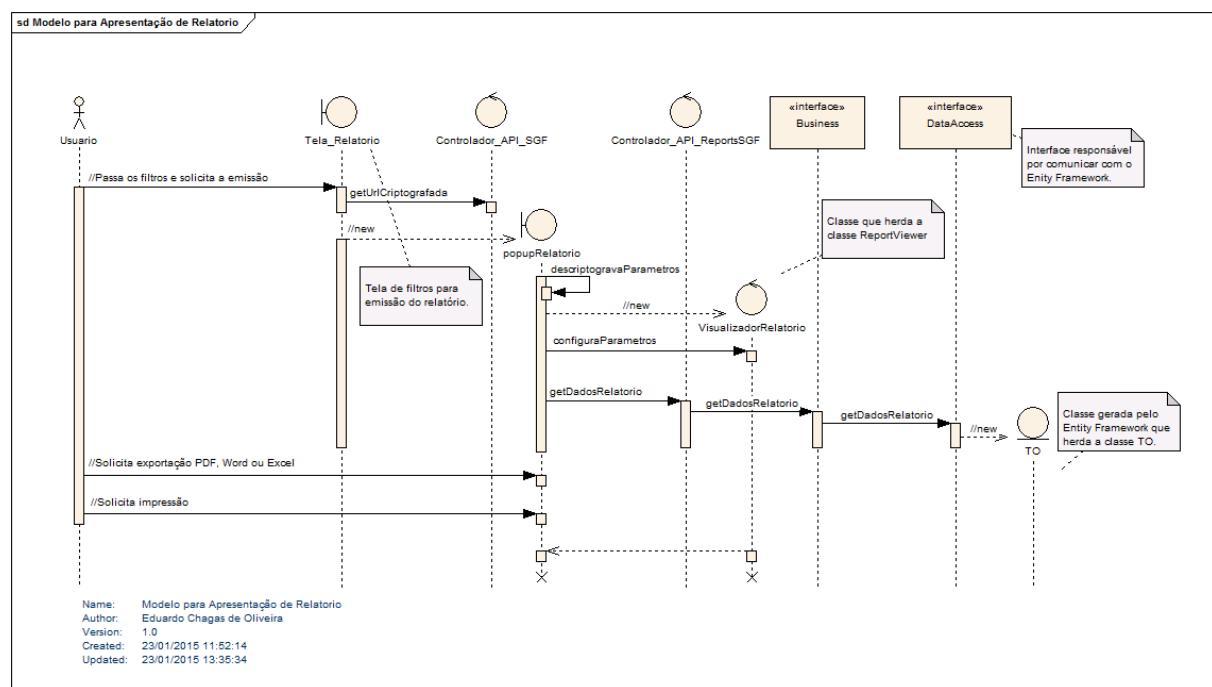


Figura 28. Diagrama de sequência para o modelo de apresentação de relatórios.

The screenshot shows the SGF web application's reporting interface. At the top, there's a navigation bar with links for 'Secretaria', 'Coordenação', 'Financeiro', and 'Relatórios'. On the right side of the header, there's a 'FISK CENTRO DE ENSINO' logo and some system icons. Below the header, the main title is 'Posição Financeira'. A message 'Sua sessão expira em 19 min 35 seg' is displayed. The central area contains a form with several sections: 'Filtros' (Movement: Contas a Pagar, Order: Vencimento, Supplier: [empty]), 'Mostrar Detalhes' (Show Responsible: checked, Personal Data: checked, Address: checked, Description of Discounts: checked, Account Type: checked), and 'Período da Data Inicial' (Initial Date: 04/01/2012, Final Date: 09/02/2015, Base Date: 09/02/2015). At the bottom right of this form is a red button labeled 'Relatório'.

Figura 29. Tela do relatório de posição financeira do sistema WEB SGF.

A Figura 29 mostra um exemplo de tela para requisição do relatório de posição financeira de uma franquia do Fisk. Nesta tela é possível configurar parâmetros para emissão do relatório e buscar as informações financeiras em um determinado período. Quando o usuário clica no botão “Relatório”, o sistema emite o relatório de posições financeiras solicitado, conforme visto na Figura 30.

The screenshot shows the generated report titled 'Contas a Pagar por Data de Vencimento' for the period from 04/01/2012 to 09/02/2015, based on the base date of 09/02/2015. The report displays a table of transactions grouped by due date. The columns include Vencimento (Due Date), Fornecedor (Supplier), Telefone (Phone), Emissão (Issuance), Título (Title), Documento (Document), Valor (Value), Juros (Interest), Multa (Fine), Desconto (Discount), and Saldo (Balance). The table shows multiple entries for different suppliers (Amarildo de Souzas, Antonio) across various due dates (29/11/2014, 30/11/2014, 01/12/2014, 10/12/2014, 29/12/2014, 10/01/2015, 15/01/2015, 29/01/2015). A summary table at the bottom provides totals for each column.

Vencimento	Fornecedor	Telefone	Emissão	Título	Documento	Valor	Juros	Multa	Desconto	Saldo
29/11/2014	Amarildo de Souzas		30/11/2014	8-1	Titulo	125,00	0,00	0,00	0,00	125,00
				1		125,00	0,00	0,00	0,00	125,00
30/11/2014	Amarildo de Souzas		16/10/2014	5-2	Titulo	80,00	0,00	0,00	0,00	80,00
						80,00				
				1		80,00	0,00	0,00	0,00	80,00
01/12/2014	Amarildo de Souzas		01/12/2014	7-1	Titulo	125,00	0,00	0,00	0,00	125,00
				1		125,00	0,00	0,00	0,00	125,00
10/12/2014	Antonio		10/12/2014	1017-1	Titulo	480,00	0,00	0,00	0,00	480,00
				1		480,00	0,00	0,00	0,00	480,00
29/12/2014	Amarildo de Souzas		30/11/2014	8-2	Titulo	125,00	0,00	0,00	0,00	125,00
				1		125,00	0,00	0,00	0,00	125,00
10/01/2015	Antonio		10/12/2014	1017-2	Titulo	480,00	0,00	0,00	0,00	480,00
				1		480,00	0,00	0,00	0,00	480,00
15/01/2015	Antonio		14/01/2015	1016-1	Titulo	300,00	0,00	0,00	0,00	300,00
				1		300,00	0,00	0,00	0,00	300,00
29/01/2015	Amarildo de Souzas		16/10/2014	5-3	Titulo	30,00	0,00	0,00	0,00	30,00
						30,00				
				1		30,00	0,00	0,00	0,00	30,00
Totais				8		1.745,00	0,00	0,00	0,00	1.745,00

Página: 1 de 1 RptTitulosData.rdlc Usuário: master Impresso em: 09/02/2015 11:35:56

Figura 30. Relatório de posição financeira do sistema WEB SGF.

5.3 Modelo Treemap

O modelo Treemap consiste em uma forma de visualização mais dinâmica que as anteriores e possui estrutura organizacional em árvore.

5.3.1 Componente Gráfico da Treemap

Quando os dados a serem visualizados estão relacionados entre si a partir de critérios que podem categorizá-los segundo algum parâmetro, é possível organizar estes dados em uma estrutura de árvore e mostrar através do componente gráfico de Treemap.

A biblioteca do Dojo que implementa a treemap é “`dojox/treemap/TreeMap`”. As informações que constituem este componente podem vir da base de dados ou através de arquivos provenientes de qualquer camada do sistema, mas não é feita sob demanda. O ganho da treemap está na disponibilização dinâmica dos dados pelo componente, durante a interação e visualização das informações.

A Figura 7 mostra um exemplo de estrutura de Treemap. Depois que o componente de treemap disponibiliza as informações ao usuário, a busca dos dados é bastante efetiva. A análise deste modelo e seu componente são discutidos no Capítulo 5 desta dissertação.

5.4 Análise de Desempenho dos Componentes

Os componentes gráficos de multiseleção oferecem ganhos em espaço na página WEB, mas o usuário terá que pesquisar o item desejado em toda a lista. É possível amenizar esse problema disponibilizando os itens em ordem alfabética. A pesquisa no banco de dados busca toda a lista de itens, ao menos que o componente utilize *autocomplete* e faça a pesquisa por demanda.



Figura 31. Resultado de teste de desempenho do componente de multiselectão.

A Figura 31 mostra um gráfico com os resultados de teste de desempenho do componente de multiselectão para verificar a quantidade de registros que pode ser disponibilizada pelo componente sem comprometer o tempo de resposta. O eixo das abscissas contém a quantidade de registros solicitados na requisição e o eixo das ordenadas contém o tempo gasto para buscar os registros e apresentá-los. O tempo de busca corresponde ao somatório dos tempos de renderização do componente WEB, de rede, de processamento de negócio e de banco de dados. Verifica-se que o componente de multiselectão é viável para buscar até dois mil registros.

Por outro lado, os componentes de grade de paginação não renderizam toda a informação na tela, processando os dados em parcelas. Na análise destes componentes de grade, é possível categorizar e comparar três formas de buscas: sem paginação, paginação sem demanda e paginação sob demanda.

O componente sem paginação busca e mostra todas as informações na grade sem dividi-las em parcelas. O componente de paginação sem demanda busca todas as informações, mas divide as informações em páginas para renderizar. Os componentes gráficos de paginação por demanda não buscam toda a lista de itens no

banco de dados, mas somente a parcela referente à demanda. Essa parcela pode ser referente também a um conjunto de filtros de pesquisa e ordenação.

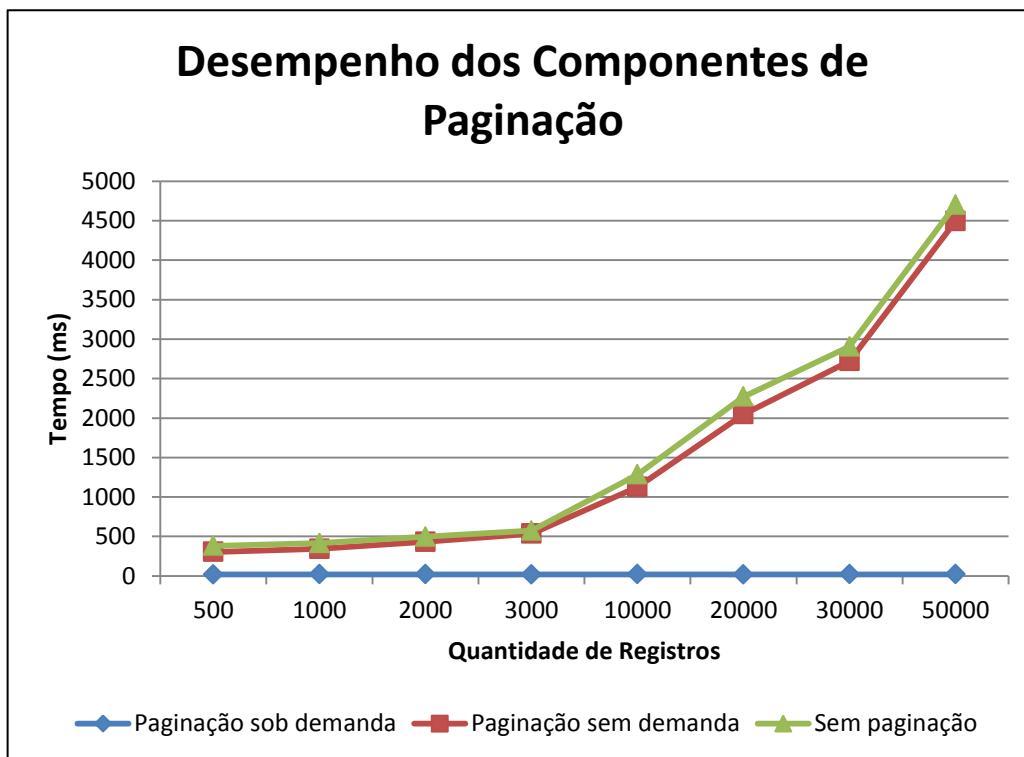


Figura 32. Resultado de teste de desempenho do componente de paginação.

A Figura 32 mostra o resultado comparativo dos testes de desempenho dos componentes gráficos sem paginação, com paginação sem demanda e com paginação sob demanda. O eixo das abscissas contém a quantidade de registros solicitada na requisição e o eixo das ordenadas contém o tempo gasto para buscar os registros e apresentá-los em cada tipo de componente. O tempo de busca corresponde ao somatório dos tempos de renderização do componente WEB, de rede, de processamento de negócio e de banco de dados. Verifica-se que os componentes sem paginação e com paginação sem demanda são viáveis para buscar até dezenove mil registros.

Os componentes gráficos de paginação por demanda oferecem uma melhor solução quando se trata de desempenho, mas possui a desvantagem de maior dificuldade de implementação. Além disso, realizam um maior número de requisições de rede e de banco quando o usuário pagina a grade. Assim como os

outros componentes gráficos de paginação, requerem maior espaço na tela para disponibilização de campos de pesquisa e ordenação dos resultados de pesquisa.

Ainda é possível reduzir os espaços de campos de pesquisa dos componentes de paginação. A Figura 33 mostra que é possível reduzir o espaço usado pelos filtros de pesquisa com o componente de tags *dijit.TitlePane*. No exemplo desta figura, os filtros de pesquisa da tela de feriados foram reduzidos para dar espaço para a grade de resultado da pesquisa.

O componente gráfico de hierarquia oferece uma busca logarítmicas das informações disponibilizadas ao usuário. A busca é bastante eficiente, pois reduz, em partes, a busca pela informação a cada decisão na árvore ou, no caso do componente visual do Dojo, a cada expansão na escolha do usuário. Entretanto, o componente do Dojo para gráfico em hierarquia não faz a pesquisa por demanda e também não possui *plug-in* para paginação, assim como o componente de gráfico por paginação. Isso torna esse componente inviável para grandes quantidades de informações, como pesquisa e disponibilização acima de duzentos mil registros do banco de dados.

Sala | Produto | Estágio | Duração | Modalidade | Regime | Conceito | **Feriado** | Evento | Motivo da Falta | Motivo da Desistência | Tipo de Atividade Extra

Pesquisa

Descrição: Início: Tipo: **Todos** ▾
Somente com Ano:

▼ Período - Data Inicial
Ano: Mês: Dia:

▼ Período - Data Final
Ano: Mês: Dia:

Ações Relacionadas ▾ **Novo** **Relatório** **Todos Itens** ▾

<input type="checkbox"/>	Feriado	Dia Início	Mês Início	Ano Início	Dia Final	Mês Final	Ano Final	Feriado Escola	Financeiro
<input type="checkbox"/>	Carnaval	1	4	2014	4	4	2014	Não	Sim
<input type="checkbox"/>	Corpus Christie	19	6	2014	19	6	2014	Não	Não
<input type="checkbox"/>	Paixão de Cristo	18	4	2014	18	4	2014	Não	Sim

Sala | Produto | Estágio | Duração | Modalidade | Regime | Conceito | **Feriado** | Evento | Motivo da Falta | Motivo da Desistência | Tipo de Atividade Extra

Pesquisa

Ações Relacionadas ▾ **Novo** **Relatório** **Todos Itens** ▾

<input type="checkbox"/>	Feriado	Dia Início	Mês Início	Ano Início	Dia Final	Mês Final	Ano Final	Feriado Escola	Financeiro
<input type="checkbox"/>	Carnaval	1	4	2014	4	4	2014	Não	Sim
<input type="checkbox"/>	Corpus Christie	19	6	2014	19	6	2014	Não	Não
<input type="checkbox"/>	Paixão de Cristo	18	4	2014	18	4	2014	Não	Sim

1 - 3 de 3 itens **17** | 34 | 68 | 100 | Todos ▾

Figura 33. Componente de *tag* do Dojo para redução do espaço usado de filtros de pesquisa.

Capítulo 6

Arquitetura do Sistema WEB

Componentes WEB de alta performance exigem suporte de uma arquitetura capaz de acoplá-los sem perda do seu desempenho proposto. A integração destes componentes, que podem ser de diferentes frameworks também exige uma arquitetura que os organize de forma a permitir a sua fácil integração. Este capítulo propõe e mostra o desenvolvimento de uma arquitetura capaz de integrar componentes WEB de alto desempenho.

A definição de arquitetura utilizado nesta dissertação baseia-se nos conceitos apresentados no Anexo 1.

6.1 Proposta Arquitetural

A arquitetura candidata para atender as necessidades dos componentes WEB com alta performance deve possuir suportabilidade de paginação sob demanda. Além disso, deve apresentar uma facilidade de desenvolvimento da paginação sob demanda para recebimento dos parâmetros de número da página, de ordenação da coluna e quantidade de itens por página do protocolo HTTP e aplica-los na consulta da busca da informação no banco de dados.

A arquitetura candidata também deve suportar o acoplamento de componentes e subsistemas para que seja fácil substituir um framework por outro

nas mudanças evolutivas dos sistemas que usam esta arquitetura. Por exemplo, deve ser possível mudar, de maneira fácil, um framework Dojo da versão 1.8.0 para 2.0.0 ou mudar o framework Dojo para o framework JQuery. Isso garante facilidade na evolução dos componentes e também a evolução arquitetural. O acoplamento também possibilita o uso de componentes de frameworks diferentes no mesmo sistema.

A arquitetura em camadas usando ASP.NET API e MVC possibilita o fácil acoplamento de componentes e subsistemas e, ainda, possui integração bastante funcional com o protocolo HTTP, provendo funcionalidades em forma de serviços WEB. O Entity Framework da Microsoft possibilita a pesquisa dos parâmetros de pesquisa sob demanda, com fácil implementação e garante a performance da busca no banco de dados. Além disto, este framework garante a fácil integração do paradigma orientado a objetos com o paradigma relacional. Outro framework candidato para acesso ao banco de dados poderia ser o Hibernate do Java. A arquitetura proposta privilegia a integração dos frameworks da Microsoft .NET para garantir o fácil desenvolvimento do sistema como um todo.

Componentes WEB de alta performance podem ser utilizados em sistemas de pequeno a grande portes, mas são mais requeridos em sistemas que possuem grandes quantidades de informações a serem buscadas para visualização. A arquitetura em camadas ajuda na organização dos subsistemas, as dependências entre eles e, portanto, dos seus acessos. A arquitetura proposta nesta dissertação divide os subsistemas nas camadas:

- **Uso Específico da Aplicação:** Nesta camada se encontram os subsistemas que fazem uso específico da aplicação, que possuem particularidades da aplicação para acesso às regras de negócio da camada de uso específico do negócio.
- **Uso Específico do Negócio:** Esta camada contém os subsistemas que tratam as regras de negócio do sistema.

- **Middleware:** Camada mediadora da arquitetura para melhor distribuir os subsistemas.
- **Infra-Estrutura:** Camada composta pelos subsistemas de infra-estrutura e utilidades genéricas do sistema.
- **Sistema:** Camada composta pelos frameworks e bibliotecas utilizados pelo sistema.

6.2 Elaboração da Arquitetura Proposta

Deve ser possível reutilizar a mesma arquitetura para centralizar diferentes funcionalidades, de diferentes requisitos. Os modelos arquiteturais destes sistemas se distinguem pelo uso de diferentes subsistemas e suas dependências. As Figura 34 e Figura 35 retratam o modelo arquitetural em camadas do sistema SGF do Fisk. As Figura 36 e Figura 37 retratam o modelo arquitetural em camadas do sistema de realidade virtual da Cemig. As Figura 38 e Figura 39 retratam o modelo arquitetural em camadas do sistema de realidade virtual da Freeman.

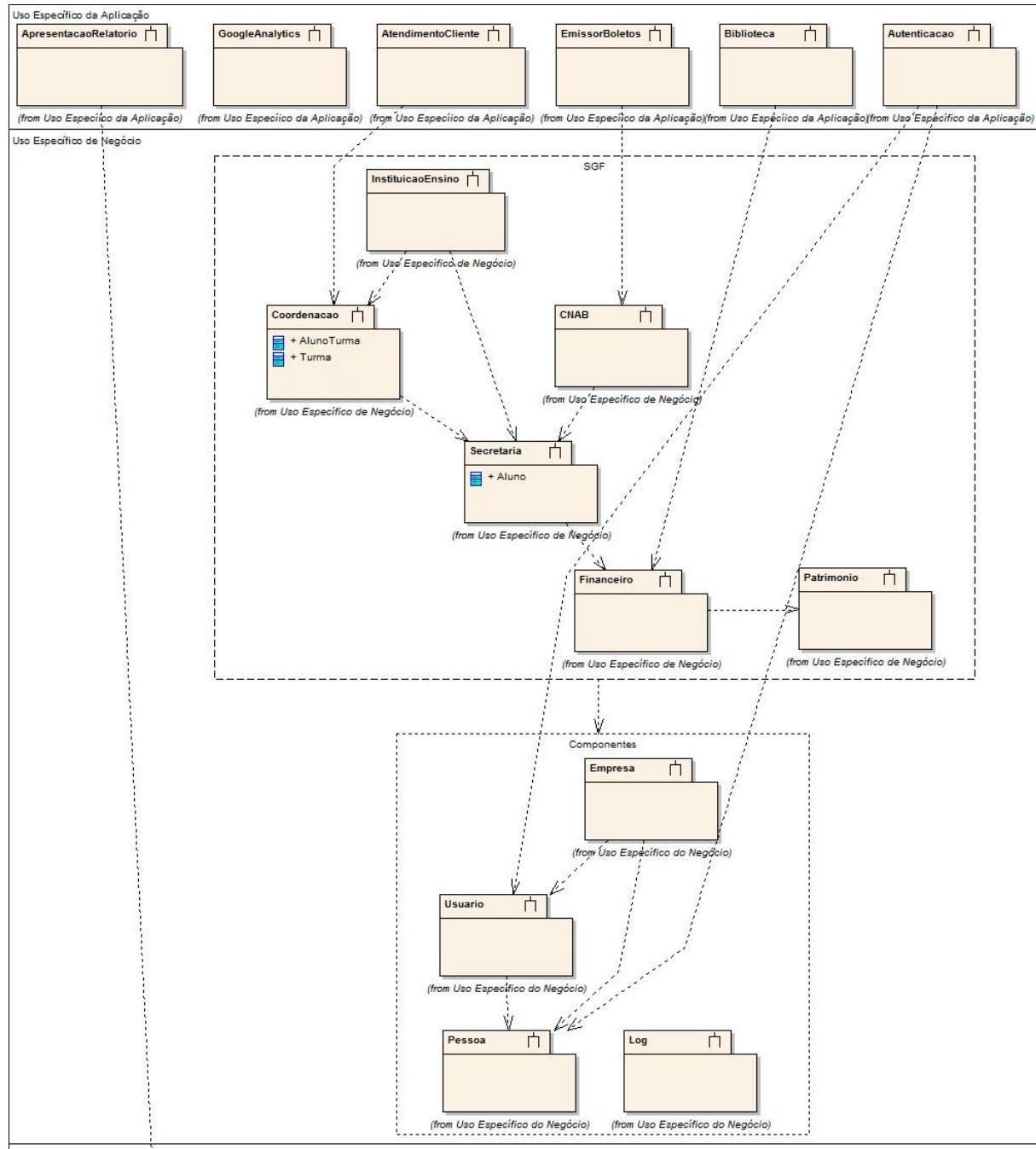


Figura 34. Camadas de Uso Específico da Aplicação e Uso Específico de Negócio do sistema SGF.

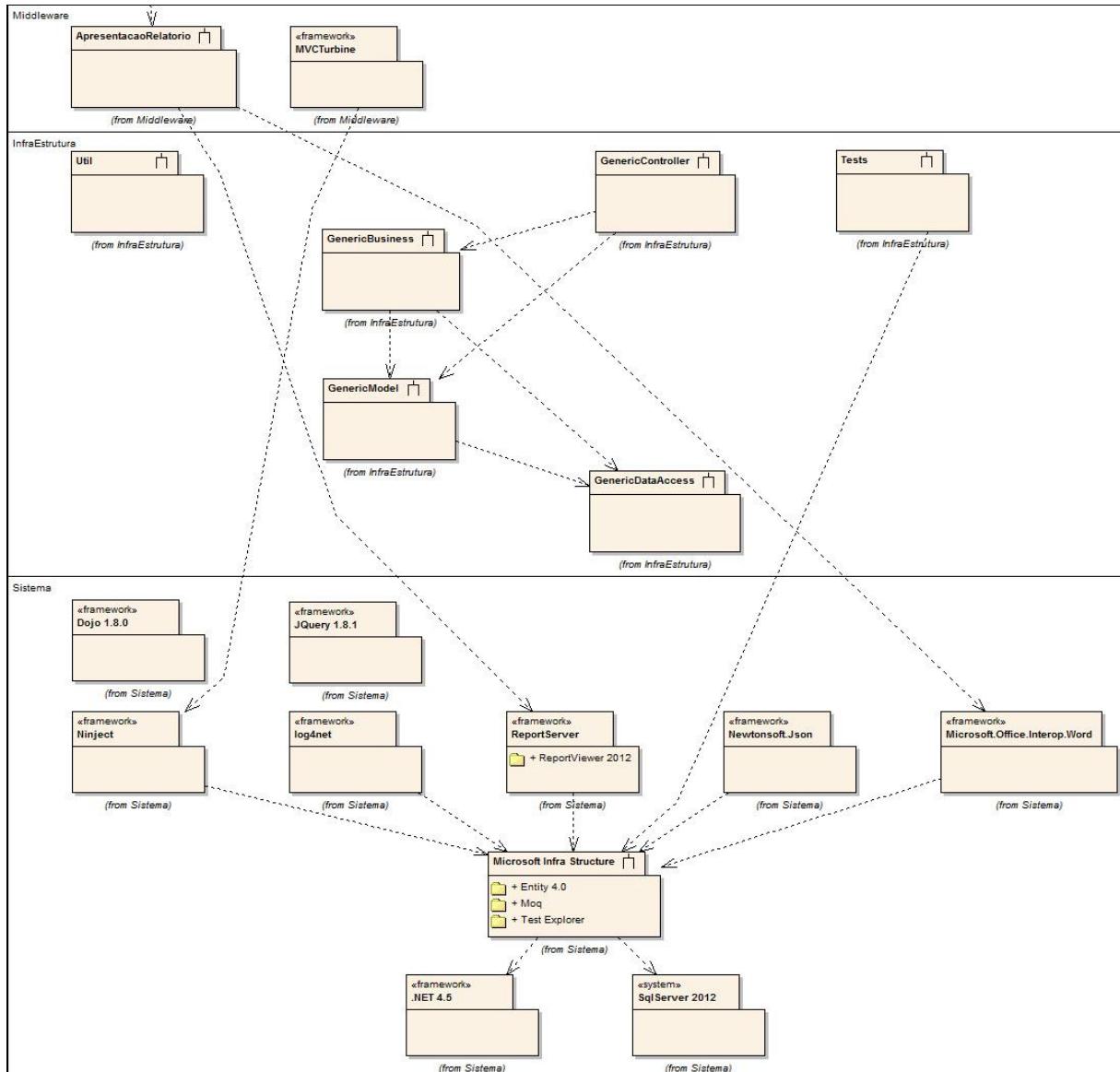


Figura 35. Camadas de Middleware, Infra-Estrutura e Sistema do sistema SGF.

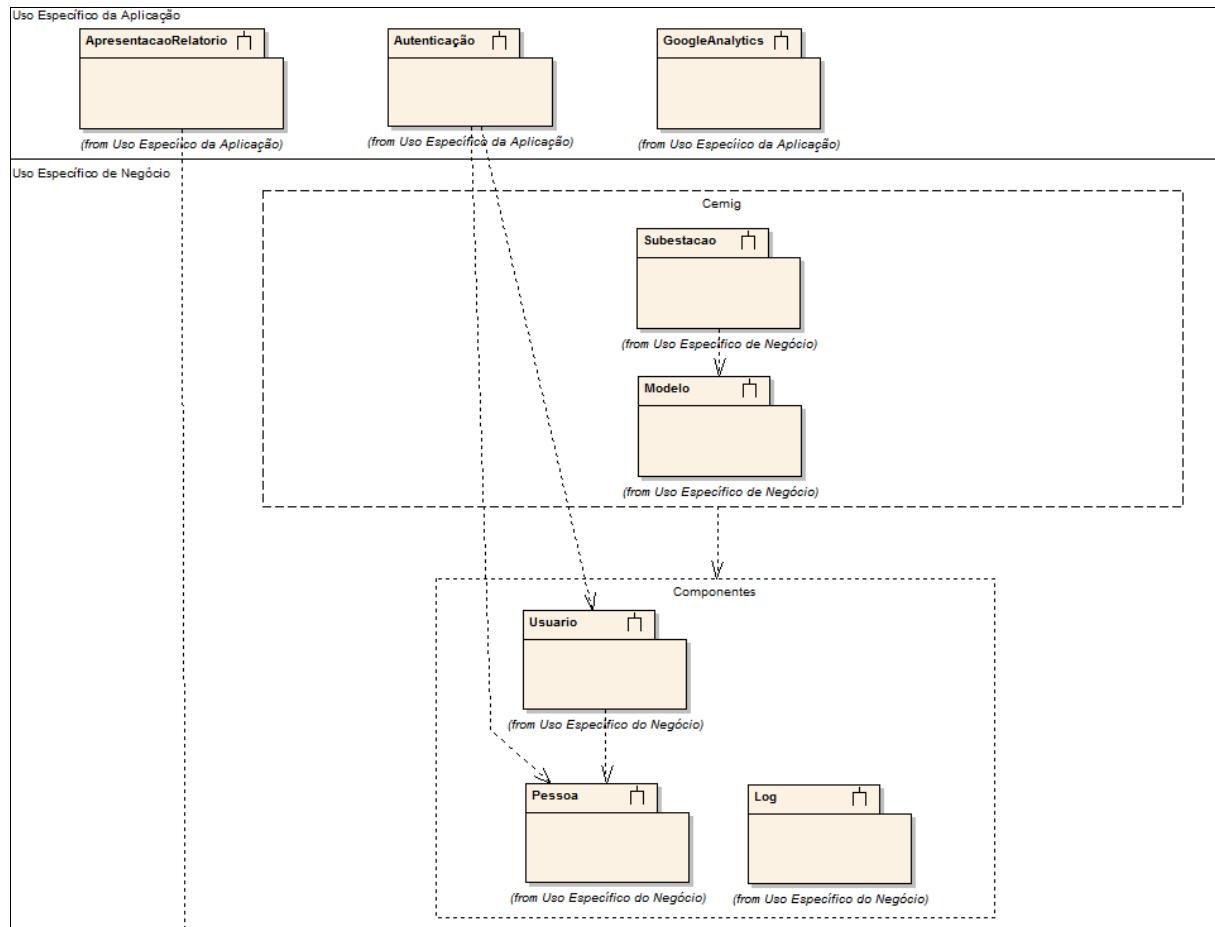


Figura 36. Camadas de Uso Específico da Aplicação e Uso Específico de Negócio do sistema Cemig.

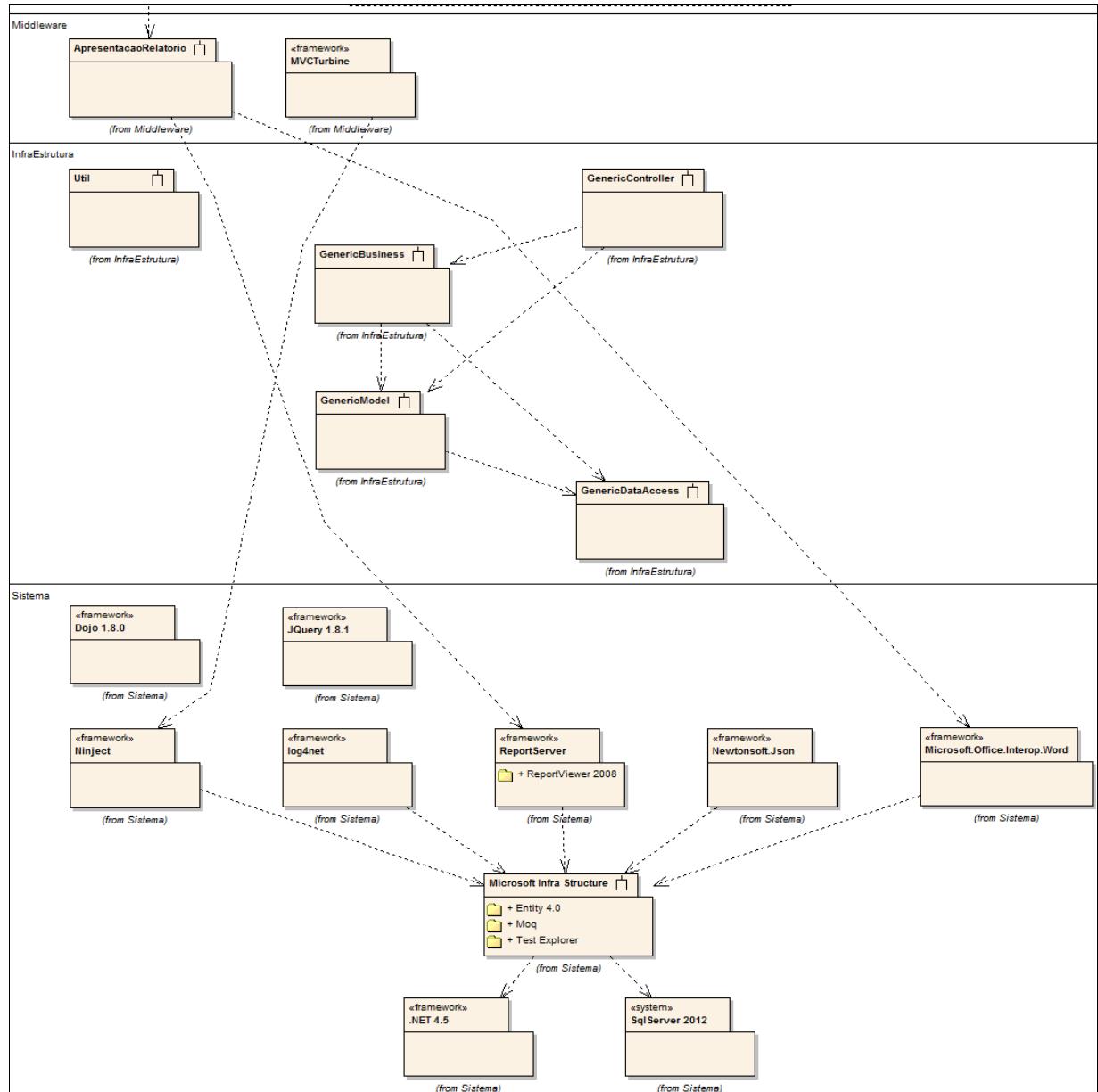


Figura 37. Camadas de Middleware, Infra-Estrutura e Sistema do sistema Cemig.

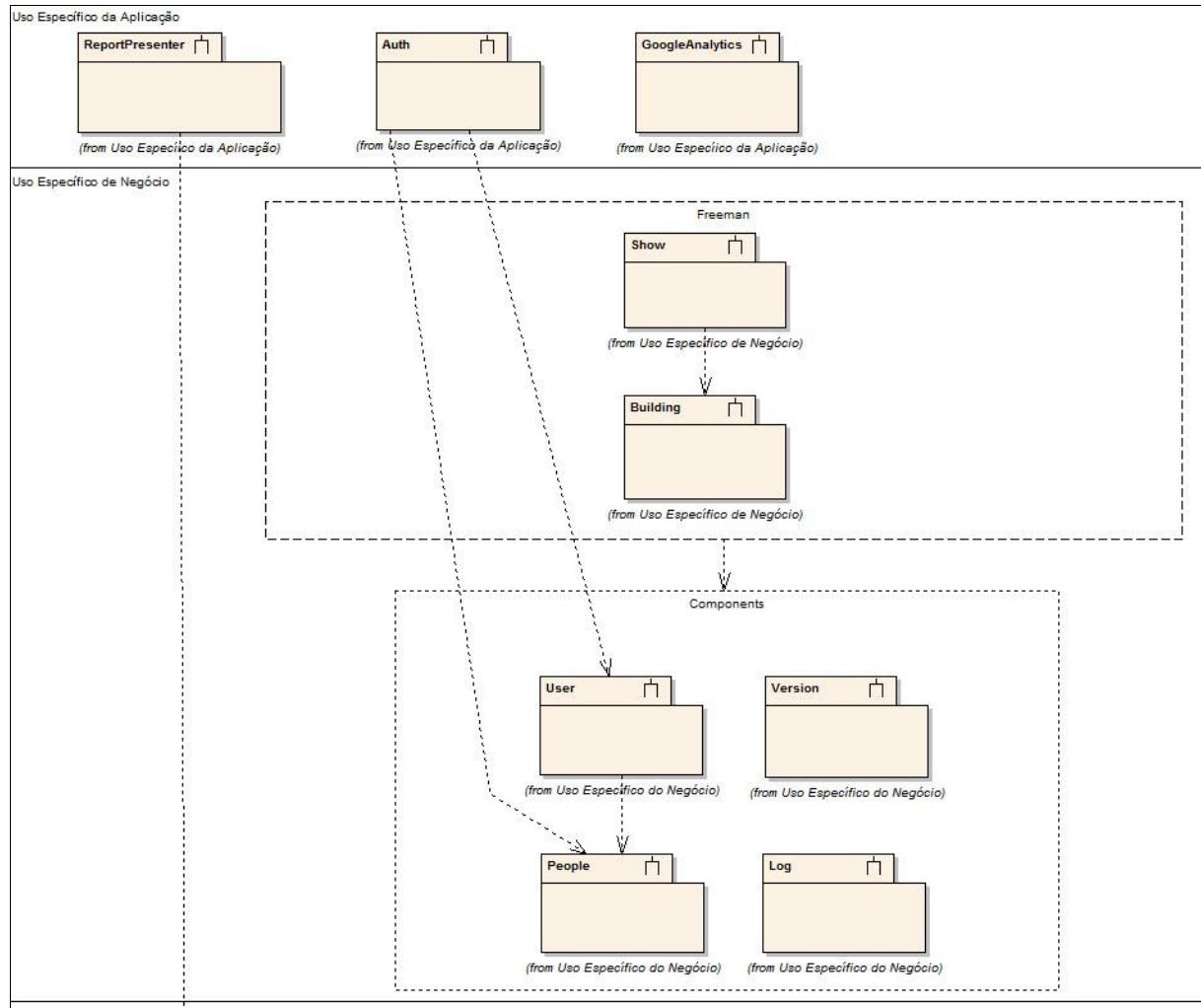


Figura 38. Camadas de Uso Específico da Aplicação e Uso Específico de Negócio do sistema Freeman.

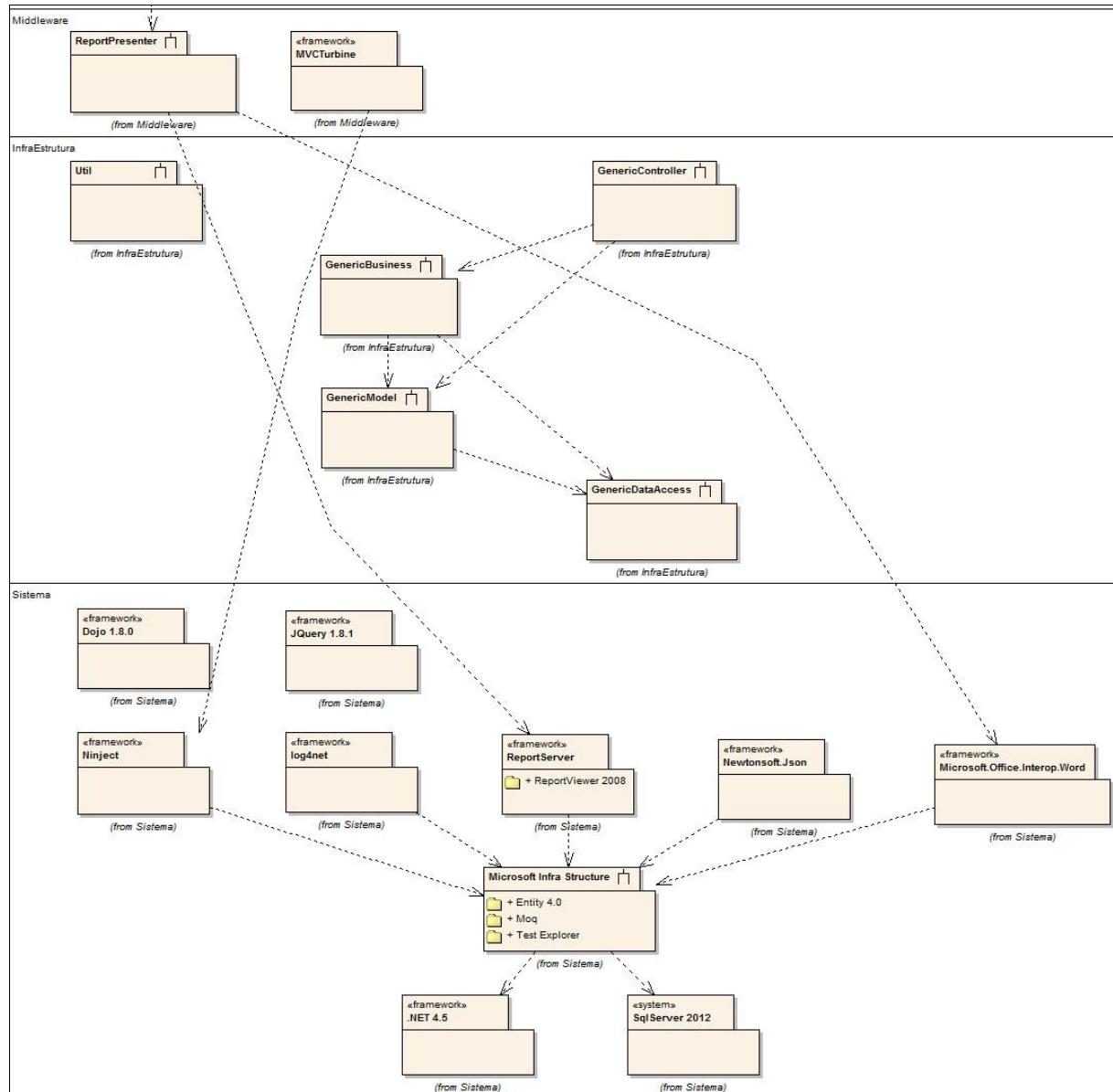


Figura 39. Camadas de Middleware, Infra-Estrutura e Sistema do sistema Freeman.

A mesma arquitetura foi utilizada para os sistemas SGF, Cemig e Freeman, que necessitam dos mesmos componentes WEB para o modelo de busca de informação. Isto é possível, embora esses sistemas possuem requisitos, regras e processos de negócio distintos, regras de sistema e processos de sistemas específicos.

Utilizar a mesma arquitetura de software não implica necessariamente que o processo ou metodologia de desenvolvimento de software seja o mesmo. Por exemplo, o SGF foi desenvolvido com o processo ágil SCRUM, enquanto que os sistemas da Freeman e Cemig não foram desenvolvidos com um processo formal. Entretanto, a organização arquitetural do sistema auxilia e facilita o desenvolvimento

e manutenção do software. A aprendizagem do time de desenvolvimento é mais rápida quando o sistema está centralizado em uma arquitetura.

A camada Uso Específico da Aplicação do sistema SGF contém os subsistemas:

- **ApresentacaoRelatorio:** responsável por apresentar os relatórios gerencias do sistema.
- **GoogleAnalytics:** *framework* e BI da empresa Google para emissão de relatórios com a finalidade de acompanhamento e análise dos acessos dos usuários no sistema WEB. Os relatórios contêm diversos indicadores e métricas que ajudam a compreender quais os módulos do sistema estão funcionando conforme o esperado. É possível conhecer as ferramentas que os usuários utilizam para acessar o sistema, verificar a frequência de acesso, quantidade de usuários, acompanhar o tempo médio de resposta do servidor, por requisição, o tempo médio de download das páginas, o tempo de carregamento do conteúdo do documento e tempo de carregamento da página, entre outras métricas.
- **AtendimentoCliente:** responsável pelos atendimentos gerais de clientes.
- **Biblioteca:** responsável pelo controle geral bibliotecário.
- **Autenticacao:** responsável por autenticar usuários no sistema.
- **EmissorBoletos:** responsável pela emissão de boletos para as escolas franquiadas receberem as faturas dos alunos.

A camada Uso Específico de Negócio está dividida em subsistemas do SGF e componentes reutilizáveis. Ela contém os subsistemas:

- **InstituicaoEnsino:** responsável por gerenciar de forma centralizada todas as instituições de ensino da fundação Fisk.
- **Coordenacao:** responsável por controlar as regras de negócio referente às instituições.

- **CNAB:** responsável pela gestão e controle dos recebimentos, envio e retorno das remessas das faturas em forma de arquivos para os bancos que utilizam o padrão da FEBRABAN. O SGF faz integração com o Banco do Brasil, Basa, Santander, BRB, Caixa, Bradesco, Itau, Sudameris, Real , Unibanco, Safra, Sicredi e Sicoob.
- **Secretaria:** responsável pelas regras referentes às secretarias das coordenações.
- **Financeiro:** responsável pelas regras financeiras da instituição de ensino.
- **Patrimonio:** responsável pelo controle de todo o estoque e patrimônio físico e financeiro da instituição.

As regras de negócio referentes aos requisitos de multiempresa, multiusuário e auditoria do sistema são geridas pelos subsistemas componentizáveis “Empresa”, “Usuario”, “Pessoa” e “Log”, respectivamente.

- **Empresa:** responsável pelas regras de negócio gerais para qualquer sistema WEB, que utiliza requisitos de empresa ou multi-empresa.
- **Usuario:** responsável pelas regras de negócio gerais para qualquer sistema WEB, que utiliza requisitos de usuário ou multi-usuário.
- **Pessoa:** responsável pelas regras de negócio relativas á herança de pessoas físicas e jurídicas para qualquer sistema WEB.
- **Log:** responsável pelo controle e gestão de log e auditoria de qualquer sistema WEB.

A camada “Middleware” é uma camada intermediária, composta pelos subsistemas:

- **ApresentadorRelatorio:** capaz de mediar as regras específicas da aplicação referente aos relatórios e o framework ReportService da camada Sistema.
- **MVCTurbine:** biblioteca para configurações de outras bibliotecas de injeção de dependências como o Ninject da camada Sistema.

A camada InfraEstrutura possui os subsistemas:

- **Util:** responsável por fornecer funções de utilidade às demais camadas, como por exemplo, criptografia ou envio de email.
- **GenericController:** responsável por abstrair as funcionalidades dos componentes de controle.
- **GenericBusiness:** responsável por abstrair as funcionalidades dos componentes de negócio.
- **GenericModel:** responsável por abstrair as funcionalidades das classes e interfaces referentes aos objetos de transferências de informações entre as camadas do sistema.
- **GenericDataAccess:** responsável por abstrair as funcionalidades dos componentes de banco de dados.
- **Tests:** responsável pela implementação e execução dos testes de unidade de qualquer subsistema.

A camada “Sistema” possui:

- **Dojo 1.8.0:** biblioteca javascript na versão 1.8.0, para suporte ao desenvolvimento de interfaces gráficas.
- **JQuery 1.8.1:** biblioteca javascript na versão 1.8.1, para suporte ao desenvolvimento de interfaces gráficas.
- **Ninject:** biblioteca responsável pela injeção de dependências do sistema. Esta biblioteca facilita o desenvolvimento das comunicações entre interfaces dos subsistemas e classes do sistema.
- **log4Net:** biblioteca responsável por gerenciar os logs da aplicação em arquivos.
- **ReportServer:** framework responsável por emitir relatórios padrões gerenciais.
- **Microsoft.Office.Interop.Word:** biblioteca responsável por emitir relatórios dinâmicos e com personalização dos usuários administradores do sistema.

- **Newtonsoft.Json:** biblioteca responsável pela serialização de dados que são enviados do protocolo HTTP para a aplicação.
- **Entity Framework:** framework responsável pelo controle e transmissão de dados ao banco de dados.
- **Mok:** biblioteca responsável pela criação de esboços de comportamentos de modelos para simulação em testes
- **Text Explorer:** plugin responsável pela realização de testes de unidade no Microsoft Visual Studio.
- **.NET 4.5:** atualização do framework Microsoft .NET 4.0, que consiste em uma plataforma de desenvolvimento e execução de sistemas e aplicações.
- **SQL Server 2012:** banco de dados da Microsoft.

De forma similar, os modelos arquiteturais dos sistemas da Freeman e da Cemig possuem a maior parte dos subsistemas nas camadas “Sistema”, “InfraEstrutura” e “Middleware” em comum com os do sistema do SGF. Entretanto, somente foram realizados testes de sistema e não sendo realizados testes de unidade nos mesmos. Assim, o subsistema “Tests” não existe nos seus modelos arquiteturais na camada “InfraEstrutura”. Já nas camadas “Uso Específico da Aplicação” e “Uso Específico de Negócio” os subsistemas são diferentes entre si e com relação aos do SGF. As regras de aplicações dos subsistemas “Biblioteca” e “AtendimentoCliente” são exclusivas ao SGF. As regras de negócio dos subsistemas “Subestacao”, para administrar as subestações de energia elétrica, e “Modelo”, para gerir o cadastro dos modelos das subestações são específicos da Cemig, enquanto que as regras de negócio dos subsistemas “Show”, para gerenciar as demonstrações e projetos das construções dos prédios e “Building”, o qual controla estas construções, são específicas do negócio da Freeman.

O Entity Framework cria automaticamente as classes a partir das tabelas do banco de dados (*database-first*) ou cria automaticamente a base de dados a partir do modelo de classes e diagrama de entidade e relacionamento - DER do sistema (*code-first*). Nesta ferramenta, é possível desenhar e criar as classes a partir do DER. Na

arquitetura proposta, todas estas classes são *transfer objects* - TO, como padrão de projeto e estas classes percorrem todas as camadas do MVC, desde o subsistema “DataAccess” até a camada de apresentação. Os componentes WEB utilizam estes objetos TO para popular as informações da tela.

A Figura 41 ilustra um exemplo de diagrama de classes do sistema WEB da Cemig, com as classes geradas a partir das tabelas do banco de dados, ilustradas no DER da Figura 40. Cada subsistema pode ter o seu próprio diagrama de classes e comunicação com o banco de dados. Entretanto, foi observado nesta dissertação que a criação de um único diagrama de classes facilita a manutenção do projeto, pois evita a repetição das classes.



Figura 40. Exemplo de DER do sistema WEB da Cemig.

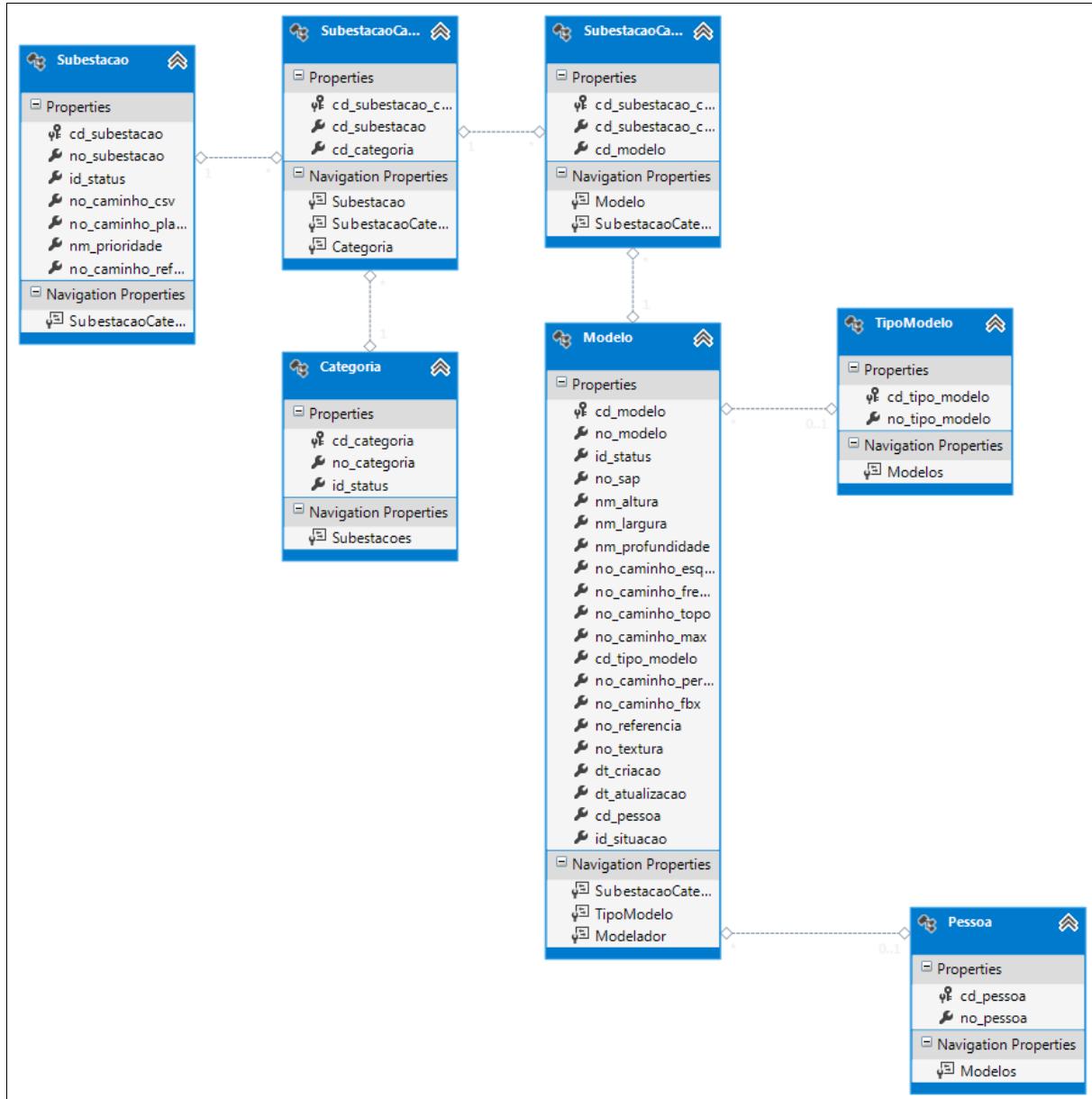


Figura 41. Exemplo de diagrama de classes do sistema WEB da Cemig.

Outra ferramenta que colaborou muito para o desenvolvimento do sistema foi a escolha do repositório *TortoiseSVN*, para armazenamento dos arquivos fontes e executáveis. Embora a arquitetura seja toda baseada em bibliotecas e ferramentas Microsoft, optou-se pelo *TortoiseSVN*, pois o mesmo controla as versões dos arquivos através de revisões, o que facilita a rastreabilidade do código fonte, com relação ao requisito ou a um possível erro e também possibilita a separação das *sprints* do SCRUM em linhas de desenvolvimento (*trunk, branches e tags*).

Capítulo 7

Discussão e Conclusões

Com base na proposta de metamodelo, sua implementação e testes comparativos com o metamodelo já existente, este capítulo apresenta como finalidade realizar as discussões finais e conclusões a respeito dos resultados obtidos. Conclusões da implementação, análise e testes dos componentes de busca também serão aqui focados.

7.1 Desempenho na Visualização da Informação

Os critérios da visualização da informação possibilitam a identificação e a busca de informações através da estrutura organizacional dos modelos de visualização. Uma análise ingênuas das necessidades da visualização da informação pode criar arquitetura, ou disponibilizar componentes, com desempenho não adequado, degradação da performance desejada pela aplicação ou navegação não usual na interface gráfica.

Para encontrar o componente mais adequado à necessidade do negócio e para encontrar a melhor eficiência na navegação do usuário, é necessário conhecer as diferenças entre os diversos componentes gráficos disponíveis para desenvolvimento e, se possível, evoluí-los e até criar novos, quando não existentes. Mas para isso é

preciso realizar uma análise correta do desempenho e qualidade gráfica do componente perante as reais necessidades da visualização da informação.

As facilidades nas interações do sistema com o usuário devem ser elaboradas a partir da usabilidade dos componentes gráficos para agilizar os processos que garantem as suas necessidades de negócio. Estes devem possuir uma forma de organização da informação capaz de fornecer atalhos cognitivos e mecânicos durante a leitura das informações feita pelos usuários.

O metamodelo proposto neste trabalho de visualização de informações, baseado na estrutura lógica de Treemap, foi inspirada no modelo proposto da Tabela Periódica de modelos de visualização. Os resultados dos teste comparativos dois dois metamodelos apontam que a estrutura da inovadora proposta permitiu uma busca mais rápida, interativa e de fácil manipulação, para realizar a pesquisa dos modelos mais adequados na solução dos problemas de visualização da informação.

Alguns critérios de visualização são mais perceptíveis que outros. Assim, é possível determinar a complexidade de tempo de algoritmo tanto dos metamodelos quanto dos modelos de visualização de informação, a partir dos critérios de visualização utilizados por eles. Isso é medido através de um método criado nesta dissertação, denominado análise de algoritmos de visualização, análogo ao método de análise de algoritmos computacionais. Este método possibilita antecipar problemas de desempenho e expressividade na escolha do modelo mais adequado para visualização da informação antes da sua implementação, ou seja, durante a sua análise. Isso reduz transtornos de implementação indesejáveis e mudanças drásticas de interfaces gráficas.

De acordo com os experimentos realizados, podemos observar que o metamodelo de visualização baseado em TreeMap possui maior eficácia e expressividade que o metamodelo baseado em *Periodic Table*, considerando os critérios de visualização de grupo e cor. Além disso, os resultados dos experimentos mostram que o modelo de análise de algoritmos pode ser aplicado na visualização da informação.

O metamodelo Treemap possui os critérios de posição e tamanho que foram avaliados com bons resultados de capacidade de abstração e distinção de percepção humana. O critério de visualização através de símbolo que o metamodelo *Periodic Table* propõe, teve piores resultados em relação aos outros critérios, pois os usuários confundem muito os símbolos na busca pela informação.

Conclui-se que a estrutura proposta permite que os usuários tenham facilidade de compreender e buscar as representações e informações, expressadas pelo metamodelo proposto de visualização baseado em TreeMap.

7.2 Contribuições

As principais contribuições desta dissertação são:

- (1) Apresentação de uma análise comparativa de desempenhos de diferentes componentes gráficos para busca da informação.
- (2) Criação de uma arquitetura para sistemas WEB que suporte a implementação dos componentes gráficos.
- (3) Criação de um metamodelo de visualização da informação inovador, com melhor desempenho na busca da informação que os metamodelos existentes.

7.3 Trabalhos Futuros

Trabalhos futuros poderão ser desenvolvidos, com o intuito de melhorar e expandir os resultados encontrados, destacando-se:

- (1) Seleção de mais modelos para compor o metamodelo de visualização da informação, já que a busca suporta a organização de um maior número de modelos sem degradar o seu desempenho.
- (2) Reavaliar a estrutura organizacional da busca da informação utilizada no metamodelo.
- (3) Verificar se é possível atualizar os *frameworks* utilizados na arquitetura para as versões mais atuais.

Referências

BECKS, A.; SEELING, C. **A task-model for text corpus analysis in knowledge management.** In Proc. of UM-2001, 2001.

BRADLEY, N. **Marketing research. Tools & techniques.** Oxford: Oxford University Press, 2013.

CAILLETEAU, L. **Interfaces for Visualizing Multi-Valued Attributes: Design and Implementation Using Starfield Displays.** Diplome d'Etudes Approfondies, Human-Computer Interactions Laboratory, University of Maryland and Ecole des Mines de Nantes, Nantes Cedex, France, 1999.

CARD, S. K.; J.D. MACKINLAY; SHNEIDERMAN, B.; CARD, M. **Readings in Information Visualization: Using Vision to Think.** Morgan Kaufman Series in Interactive Technologies, Academic Press, 1999. ISBN: 1-55860-533-9.

CARDOSO, A. Notas de aula. Disponível em: <<http://www.alexandre.eletrica.ufu.br/vi/notas.htm>>. Acessado em: 01 abr 2014.

CLEVELAND, W.S.; MCGILL, R. **Graphical perception: Theory, experimentation and application to the development of graphical methods.** In Journal of the American Statistical Association, 1984, v. 79, p. 531-554. doi: 10.1080/01621459.1984.10478080.

DASTANI, M. **The role of visual perception in data visualization.** In Journal of Visual Languages and Computing, 2002, v. 13, n. 6, p. 601-622.

Dojo Toolkit. Disponível em: <<http://www.dojotoolkit.org/>>. Acessado em: 01 jan 2014.

DULCLERCI, A. S.; TAVARES, J. M. R. S. **Introduction of Human Perception in Visualization.** In International Journal of Imaging, CESER Publications, v. 4, p. 60-70, 2010. ISSN: 0974-0627.

FUJISHIRO, I.; FURUHATA, R.; ICHIKAWA, Y.; TAKESHIMA, Y. **A taxonomic approach to semi-automatic design of information visualization applications using modular visualization environment.** In J. D.Mackinlay, S. F. Roth, and D. A. Keim, editors, Proc. of IEEE InfoVis'00, pages 77–83. IEEE Computer Society, 2000.

GUPTA SOLO, A.M.; GUPTA, M.M. **Perspectives on Computational Perception and Cognition under Uncertainty.** In Proceedings of IEEE International Conference on Industrial Technology. v. 1, p. 221-224, 2000.

JQuery. Disponível em: <<http://jquery.com/>>. Acessado em: 01 abr 2014.

KEIM, D. A. **Information Visualization and Visual Data Mining.** In IEEE Transactions on Visualization and Computer Graphics. v. 8, p. 1-8, 2002. doi: 10.1109/2945.981847.

KURTZ, J. **ASP.NET MVC 4 and the Web API, Building a REST Service from Start to Finish.** New York: Apress, 2013. ISBN 9781430249771.

LENGLER, Ralph; EPPLER, Martin. **Towards A Periodic Table of Visualization Methods for Management.** GVE 2007, Proceedings of the IASTED International Conference on Graphics and Visualization in Engineering, 2007.

LESLIE ORCHARD, M.; PEHLIVANIAN, A.; KOON, S.; JONES H. **Professional JavaScript Frameworks Prototype, YUI, Ext JS, Dojo and MooTools.** Indianapolis: Wiley Publishing, 2009. ISBN 978-0470452028.

MATHEW RUSSELL, A. **Dojo: The Definitive Guide,** Cambridge: O'Reilly Media, 2008. ISBN 978-0596516482.

HEALEY, C.G. **Building a Perceptual Visualisation Architecture.** Behaviour & Information Technology. v. 19, n. 5, p. 349-366, 2000.

HERMAN, I.; MELANÇON, G.; MARSHALL, M. S. **Graph Visualization and Navigation in Information Visualization: A Survey.** In IEEE Transactions on Visualization and Computer Graphics. v. 6., p. 24-43, 2000.

MARTINS, J. C. C. **Técnicas para Gerenciamento de Projetos de Software.** Rio de Janeiro: Brasport, 2007.

MATSUI, K.; YAMANOUCHI, M.; SUNAHARA, H. **A Proposal of Framework for Information Visualization in Developing of Web Application.** In IEEE/IPSJ 11th International Symposium on Applications and the Internet (SAINT), 2011. p 457-462, 2011. doi: 10.1109/SAINT.2011.85.

OLIVEIRA, M. C. F; LEVKOWITZ, H. **From Visual Data Exploration to Visual Data Mining: A Survey.** In IEEE Transactions on Visualization and Computer Graphics. v. 9, n. 3., p. 378-394, 2003 doi: 10.1109/TVCG.2003.1207445.

PILLAT, R. M.; VALIATI, E. R.; FREITAS, C. M. D. S. **Experimental study on evaluation of multidimensional information visualization techniques.** In Proceedings of the 2005 Latin American conference on Human-computer interaction. ACM New York, NY, USA. p 20-30, 2005 doi: 10.1145/1111360.1111363.

PUNTAMBEKAR, A. **Design And Analysis Of Algorithms.** Technical Publications, Índia: Technical Publications Pune, p. 1-27, 2010. ISBN: 9788184317787.

PRESSMAN, R. S. **Engenharia de Software. Uma abordagem profissional.** Porto Alegre:AMGH Editora Ltda, 2011.

SCHULZ, H.-J.; NOCKE, T.; HEITLER, M.; SCHUMANN, H. **A Design Space of Visualization Tasks.** Visualization and Computer Graphics, IEEE Transactions on, vol.19, no.12, pp.2366,2375, 2013.

Sencha. Disponível em: <<http://www.sencha.com/>>. Acessado em: 01 abr 2014.

SHNEIDERMAN, B. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In IEEE Symposium on Visual Languages. p. 336-343, 1996.

SILVEIRA, P.; SILVEIRA, G.; LOPES, S.; MORERIRA, G.; STEPPAT, N.; KUNG, F.
Introdução à arquitetura e design de software: uma visão sobre a plataforma java.
Rio de Janeiro: Elsevier, 2012.

STOLTE, C.; TANG, D.; HANRAHAN, P. **Multiscale Visualization Using Data Cubes.** In IEEE Transactions on Visualization and Computer Graphics. IEEE Educational Activities Department Piscataway, NJ, USA. v. 9, p. 176-187, 2003 doi: 10.1109/TVCG.2003.1196005.

Treemap Flex. Disponível em:
<<http://flextoolbox.com/documentation/treemap/2/com/flextoolbox/controls/TeeMap.html>>. Acessado em: ago 2013.

Treemap Java Api. Disponível em:
<<http://docs.oracle.com/javase/7/docs/api/java/util/TreeMap.html>>. Acessado em: ago 2013.

Treemap .Net Api. Disponível em: <<http://msdn.microsoft.com/en-us/library/aa990118%28v=VS.80%29.aspx>>. Acessado em: ago 2013.

WARE, Colin. **Information Visualization – Perception for Design**. San Francisco: Morgan Kaufmann. 2º Ed. Rio de Janeiro: Elsevier, 2004.

WAZLAWICK, R. S. **Engenharia de Software. Conceitos e Práticas**. Rio de Janeiro: Elsevier, 2013.

WINCKLER, M. A.; PALANQUE, P.; FREITAS, C. M. D. S. **Tasks and scenario-based evaluation of information visualization techniques**. In Proceedings of the 3rd annual conference on Task models and diagrams. p. 165-172, 2004. doi: 10.1145/1045446.1045475

Anexo 1

Conceitos de Arquitetura de Sistema

Antes mesmo de escolher os modelos gráficos, é preciso elaborar uma base que suporte a implementação, integração e organização destes modelos no mesmo sistema. Durante a elaboração de um sistema, através de sínteses e esboços conceituais dos requisitos funcionais e não funcionais do sistema é possível eleger uma arquitetura candidata. Segundo Wazlawick (2013), entende-se por arquitetura candidata o esboço inicial da arquitetura que será refinada durante a construção do sistema.

Na fase de concepção poderá ser feita a *síntese arquitetural*, que é uma prova de conceito de que a arquitetura é viável. No início da elaboração será, então, definida uma arquitetura candidata inicial, possivelmente baseada nesta prova de conceito. Nas iterações subsequentes, a ênfase mudará para a adição ou o refinamento de funcionalidades sobre a arquitetura, que vai se estabilizando até a fase de construção. (WAZLAWICK, 2013).

O objetivo da arquitetura de um software é fornecer uma estrutura onde estão inseridos todos os componentes do sistema. Ao estabelecer uma arquitetura, são identificados os vários elementos que compõe o software, sendo que estes geralmente são organizados em pacotes ou subsistemas, cada um com um ou mais componentes. A arquitetura em camadas presume que o software é dividido em camadas com funções específicas entre elas. Estas funções são empacotadas em subsistemas e encapsuladas na arquitetura para se comunicar com outros subsistemas.

O empacotamento e encapsulamento dos subsistemas permite alta coesão entre as classes contidas num mesmo subsistema e alto acoplamento entre os subsistemas. Além disso, previne problemas com referências circulares entre eles e facilita o reuso na arquitetura, na inclusão ou remoção de subsistemas, uma vez que

as dependências entre eles são feitas pelas suas interfaces públicas. Martins (2007) define coesão como o grau de relação entre os componentes.

A coesão é uma das diretrizes mais importantes na definição dos componentes e implica que um componente ou classe deve encapsular somente os atributos e operações mais relacionados entre si, e com a classe ou componente propriamente dito. (MARTINS, 2007, p. 24)

O acoplamento consiste no grau de conectividade entre os componentes. Os dois conceitos são suficientes para criação de padrões arquiteturais, como encapsulamento e acoplamento de subsistemas.

A medida do grau de conectividade entre as classes e componentes é chamada de acoplamento. Reduzir o acoplamento entre classes e componentes é outro importante objetivo do projeto, que ajuda a facilitar o desenvolvimento e manutenção do software. (MARTINS, 2007, p. 24)

Os padrões arquiteturais são essenciais para garantir uma série de benefícios como portabilidade, interoperabilidade, reusabilidade e manutenibilidade. Eles buscam oferecer uma estrutura ideal para o desenvolvimento de software. Os sistemas atuais são distribuídos em computadores remotos e ao desenvolver este tipo de sistema é preciso manter seu acoplamento e funcionamento, buscando sempre a qualidade e segurança destes elementos computacionais.

Um exemplo de um subsistema que pode ser encapsulado na arquitetura consiste nos testes de unidade. Entende-se por teste de unidade, o teste automatizado numa unidade de código desenvolvido, de forma atômica e independente do restante do código.

Um teste deve garantir que a execução daquele trecho mínimo de código esteja correto. (SILVEIRA, P.; SILVEIRA, G.; LOPES; MOREIRA; STEPPAT; KUNG, 2012, p. 120)

Na arquitetura, os testes de unidade podem testar as funcionalidades de cada um dos componentes, bem como realizar um teste de desempenho, stress ou

integração com outro componente. Além disso, ele pode indicar a falta de acoplamento e coesão nos subsistemas da arquitetura.

Uma vantagem dos testes de unidade sobre outros tipos é que, além de garantir que antigos bugs não se repitam no futuro, eles dão feedback sobre o acoplamento e coesão do código. (SILVEIRA, P.; SILVEIRA, G.; LOPES; MOREIRA; STEPPAT; KUNG, 2012, p. 120).

Dos testes de unidade, se destacam os testes de desempenho, para verificar qual o tempo que o componente suporta diante de uma carga de informações do mundo real. Quando se trata de uma aplicação WEB, o teste de desempenho deve considerar um conjunto de variáveis como tempo de rede, tempo para processar regras de negócio, tempo de busca no banco de dados, tempo de integração com o protocolo HTTP e demais protocolos, etc.

Teste de desempenho são projetados para simular situações de carga no mundo real. Na medida em que cresce o número de usuários simultâneos da WebApp, ou o número de transações online aumenta, ou a quantidade de dados (download ou upload) aumenta, o teste de desempenho ajudará (...) que componentes do sistema são responsáveis pela degradação do desempenho. (...) (PRESSMAN, 2011, p. 485)

Segundo Martins, a arquitetura chamada “3 Camadas” é mais utilizada atualmente no desenvolvimento de software. Ela representa uma evolução da arquitetura cliente-servidor, pois nela a camada de negócio fica separada da camada cliente e da camada servidora. A arquitetura 3 Camadas é a mais viável quando se trata de sistemas atuais e que usam Internet. As regras de sistemas são regras funcionais típicas da camada cliente e da camada de banco de dados, enquanto as regras de negócio são provenientes das funcionalidades sintetizadas a partir dos processos de negócio mapeados no sistema.