

UNIVERSIDADE FEDERAL DO PARÁ
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ROBERTO YURI DA SILVA FRANCO

**PRISMA-MDE - Ambiente Distribuído e
Escalável para múltiplas visões de dados
coordenados**

Prof. Dr. Bianchi Serique Meiguins
Orientador

UNIVERSIDADE FEDERAL DO PARÁ
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Roberto Yuri da Silva Franco

**PRISMA-MDE – Ambiente Distribuído e
Escalável para múltiplas visões de dados
coordenados**

Dissertação de Mestrado apresentada
para obtenção do grau de Mestre em
Ciência da Computação.
Programa de Pós Graduação em
Ciência da Computação.
Instituto de Ciências Exatas e Naturais.
Universidade Federal do Pará.
Orientador Prof. Dr. Bianchi Serique
Meiguins

Roberto Yuri da Silva Franco

**PRISMA-MDE – Ambiente Distribuído e
Escalável para múltiplas visões de dados
coordenados.**

Dissertação de Mestrado apresentada
para obtenção do grau de Mestre em
Ciência da Computação.
Programa de Pós Graduação em
Ciência da Computação.
Instituto de Ciências Exatas e Naturais.
Universidade Federal do Pará.

Data da aprovação: Belém-PA. 25/02/2013

Banca Examinadora:

Prof. Dr. Bianchi Serique Meiguins
Instituto de Computação – UFPA – Orientador

Prof. Dr. Mario Massakuni kubo
Centro Universitário Alvorada – DF – Membro

Prof. Dr. Josivaldo de Souza Araujo
Instituto de Computação – UFPA – Membro

Visto:
Prof. Nelson Cruz Sampaio Neto, Dr. (UFPA)
Coordenador do PPGCC – UFPA

Belém
2013

“A ciência de hoje é a tecnologia de amanhã.”

Edward Teller

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter dado a possibilidade de viver, pensar e aprender.

A toda minha família, pela confiança e incentivo que recebo todos os dias e por aturar minhas “rabugices” diárias.

A todos os professores, mestres e doutores na vida, que contribuíram para minha formação e conclusão desse trabalho.

Agradecimentos em especial...

À minha mãe Margarete Franco, por ter me colocado no mundo, pelos seus ensinamentos e por estar ao meu lado sempre me apoiando e incentivando em todos os momentos.

Ao meu pai Roberto Tocantins, que durante a sua caminhada, deixou um dos mais importantes ensinamentos da minha vida: “O importante dessa vida é ser feliz”.

Aos meus avós, Francisca e Walter da Silva, pelo carinho, incentivo, orações e cuidados que recebo desde a infância.

Aos meus irmãos Allen e Karen Franco, pela convivência e apoio de todos os dias.

Aos meus sobrinhos afilhados Samia e Salim Franco, por me mostrarem a simplicidade da vida e deixar os dias mais alegres.

Ao Waldonio Brito pelas consultas médicas e por fazer a minha irmã ser menos “rabugenta”.

Aos amigos Sarah Lobato e Jonathas Pinheiro pela parceria e amizade de todas as horas e principalmente por aceitarem participar das minhas “barcas furadas”.

Ao Professor Bianchi Serique pela orientação, apoio e por me guiar em todos os momentos, sem ele este trabalho não seria possível.

Aos amigos Nikolas, Anderson, Diovanni, Mauro, aos novos amigos do projeto Motorola/CIN-UFPE/FADE, aos amigos mais antigos do mestrado (UFPA), do IDEFLOR e da época faculdade pelas conversas, ajudas e principalmente por participarem dos testes de usabilidade em alguma fase deste trabalho.

Ao Rafael e Ranieri que construíram bastante neste trabalho.

À todos que não foram diretamente citados, mas contribuíram na minha vida e na conclusão deste trabalho, o meu muito obrigado.

Roberto Yuri da Silva Franco

SUMÁRIO

AGRADECIMENTOS	VI
LISTA DE SIGLAS	X
LISTA DE FIGURAS	XI
LISTA DE TABELAS	XIII
PUBLICAÇÕES	XIV
RESUMO	XV
ABSTRACT	XVI
1 INTRODUÇÃO	1
1.1 Objetivos	3
1.1.1 Objetivos Específicos	3
1.2 Metodologia	3
1.3 Contribuição	4
1.4 Organização	4
2 VISUALIZAÇÃO DA INFORMAÇÃO	6
2.1 Processo de Visualização	8
2.2 Percepção e as Oito Variáveis Visuais	9
2.3 Tipos de Dados para Visualização da Informação	12
2.3.1 Unidimensionais (1D):.....	13
2.3.2 Bidimensionais (2D):	13
2.3.3 Tridimensionais (3D):.....	14
2.3.4 Multidimensional:	14
2.3.5 Temporal:.....	15
2.3.6 Hierárquico:	16
2.3.7 Rede de dados:	16
2.4 Técnicas de visualização da informação	17
2.4.1 Dispersão de Dados	17
2.4.2 Coordenadas Paralelas	17
2.4.3 Treemap	18
2.4.4 Características de uma boa Ferramenta de Visualização	19
2.4.5 Múltiplas Visões Coordenadas de Dados	19
2.5 Trabalhos Relacionados	22
2.5.1 <i>Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata.</i>	22
2.5.2 <i>Coordinated Views to Assist Exploration of Spatio-Temporal Data.</i>	22

2.5.3	<i>PRISMA - A Multidimensional Information Visualization Tool using Multiple Coordinated Views.</i>	23
3	AMBIENTES MULTIPLAS TELAS	25
3.1	Definição	25
3.1.1	Ambiente projetivo de múltiplas telas para visualização de informação	26
3.2	Trabalhos Relacionados	27
3.2.1	<i>Design Considerations for Collaborative Information Workspaces in Multi-Display Environments</i>	28
3.2.2	<i>Three Modes of Multi-Surface Interaction and Visualization</i>	29
3.2.3	<i>PointRight: Experience with Flexible Input Redirection in Interactive Workspaces</i>	30
3.2.4	<i>Adapting a Single-User, Single-Display Molecular Visualization Application for Use in a Multi-User, Multi-Display Environment</i>	31
3.2.5	<i>Code Space: Touch + air Gesture Hybrid interactions for Supporting Developer Meetings.</i>	31
3.2.6	<i>LibGlass</i>	32
4	PRISMA-MDE	35
4.1	Características Principais	35
4.2	Arquitetura	36
4.2.1	Cliente	37
4.2.2	Servidor	42
4.3	Protótipo e Funcionalidades	46
5	RESULTADOS	50
5.1	Avaliação de Usabilidade	50
5.1.1	Hipóteses	50
5.1.2	Base de dados	53
5.1.3	Tarefas	53
5.1.4	Metodologia de Aplicação	54
5.1.5	Resultados	56
5.2	Avaliação de Desempenho	59
5.2.1	Hipóteses	60
5.2.2	Resultados	61
6	CONSIDERAÇÕES FINAIS	65
6.1	Desafios Encontrados	67
6.2	Trabalhos Futuros	67
	REFERÊNCIAS	68

LISTA DE SIGLAS

IV	<i>Information Visualization</i>
VI	Visualização de Informação
MDE	<i>Multi-Display Environment</i>
MDEs	Ambientes de Múltiplas telas.
CMV	<i>Coordinated Multiple Views</i>
MVC	Múltiplas Visões Coordenadas
API	Application Program Interface

LISTA DE FIGURAS

Figura 1. Exemplo de visualização da informação (Opazo, 2011).....	7
Figura 2. Estudo do fluxo de ar na decolagem de aeronaves (James & Leon, 2011).....	8
Figura 3. Pipeline de visualização (Ward, Grinstein , & Keim, 2010).....	8
Figura 4. Exemplo de propriedades de pré-atenção.	10
Figura 5. Recomendações para utilização de atributos visuais bidimensionais (Bertin, 1983).....	11
Figura 6. Exemplo de Word Tree (Lent, 2007).	13
Figura 7. Exemplo de Heatmap com cluster (Bare, 2013).	14
Figura 8. Imagem 3d sobre as estatísticas de prostituição na cidade de São Francisco–EUA (McCune, 2013).	14
Figura 9. Exemplo de Scatter Plot Matrix (SPLOM) (Heer, Bostock, & Ogievetsky, 2008).....	15
Figura 10. Exemplo de Visualização para dados temporais. (Kraak, 2003)	15
Figura 11. Exemplo de diagrama node-link. (Kenttälä, 2011)	16
Figura 12. Exemplo de Visualização em dados do tipo rede. (Bertini & Stefaner, 2012)	16
Figura 13. Exemplo de Técnica de Dispersão de Dados multidimensional.	17
Figura 14. Exemplo de aplicação da técnica de Coordenadas Paralelas	18
Figura 15. Organização hierárquica dos dados no Treemap	18
Figura 16. Exemplo do Hyperwall 2. (Dunbar, Hardman, & Thigpen, 2013)	20
Figura 17. Imagem da ferramenta Snap-Together (North & Shneiderman, 2000).....	22
Figura 18. Imagem da ferramenta proposta por (Shimabukuro, Flores, Oliveira, & Levkowitz, 2004).....	23
Figura 19. Imagem da ferramenta PRISMA.....	24
Figura 20. Ambiente projetivo de visualização da Informação múltiplas telas (Ferramenta PRISMA).	27
Figura 21. Fotos do ambiente proposto por Waldner (Waldner, 2009).....	29
Figura 22. Fotos do ambiente proposto por Shen, Esenther, et al.	29
Figura 23. Fotos do ambiente proposto por Johanson, Hutchins, et al.....	30
Figura 24. Fotos do ambiente proposto por Forlines e Lillien, et al.....	31
Figura 25. Fotos do ambiente proposto por Bragdoni et al.	32
Figura 26. Arquitetura do framework libGlass (Guimarães P. , 2004).	34
Figura 27: Exemplo do ambiente de múltiplas telas que simula um <i>desktop</i> estendido. 35	
Figura 28: Principais módulos da arquitetura do PRISMA.....	37
Figura 29. Diagrama de classes do módulo controle de eventos.....	40
Figura 30. Diagrama de classe do módulo gerenciador de múltiplas telas.....	43
Figura 31. Diagrama de classes do módulo de gerenciamento de evento.	45
Figura 32. Exemplo de seleção e coordenação de dados.....	46
Figura 33. Execução do PRISMA-MDE no Ambiente de múltiplas telas.	47

Figura 34. Ambiente PRISMA-MDE com duas máquinas estendido uma técnica de visualização.	48
Figura 35. Ambiente PRISMA-MDE com quatro máquinas estendido uma técnica de visualização.	49
Figura 36 Foto foi ambiente de teste utilizado pelos usuários.	55
Figura 37. Média de acertos por tarefa.	56
Figura 38. Média de Tempo para resolução da tarefa.	57
Figura 39. Carga de trabalho por área de avaliação.	58
Figura 40. Classificação das ações de configurar e filtrar.	58
Figura 41. Usuários que se sentiram confortáveis ao utilizar as telas e mouse.	59
Figura 42. Consumo de tempo no processo de renderização da visão geral por tamanho da base de dados em registros.	61
Figura 43. Consumo de tempo no processo de carga inicial dos dados por tamanho da base de dados em registros.	61
Figura 44. Consumo de CPU no processo de renderização da visão geral por tamanho da base dados em registros.	62
Figura 45. Consumo de CPU no processo de carga inicial da base de dados por tamanho da base dados em registros.	62
Figura 46. Consumo de CPU nas ações de filtrar por tamanho da base dados em registros.	63
Figura 47. Consumo de memória RAM no processo de renderização da visão geral por tamanho da base dados em registros.	63
Figura 48. Consumo de memória RAM no processo de carga inicial das bases de dados por tamanho da base dados em registros.	64
Figura 49. Consumo de memória RAM na ação de filtrar por tamanho da base dados em registros.	64

LISTA DE TABELAS

Tabela 1: Classificação das tarefas por tipo e complexidade	56
---	----

PUBLICAÇÕES

FRANCO, Roberto Yuri da_Silva; Teixeira, Ranieri Barros; Guimaraes, Rafael Veras; Carneiro, Nikolas Jorge Santiago; Meiguins, Aruanda Simoes; Meiguins, Bianchi Serique. PRISMA-MDE - Distributed and Scalable Environment for Multiple Views of Data Coordinates. In: 2012 16th International Conference on Information Visualisation (IV), 2012, Montpellier – France. pg 646-651

Lourenco, Rodrigo Augusto de_Moraes; Guimaraes, Rafael Veras; Carneiro, Nikolas Jorge S; **FRANCO, Roberto Yuri da_Silva**; Meiguins, Aruanda Simoes Goncalves; Meiguins, Bianchi Serique. Exploring, Comparing and Coordinating Multiple Datasets in an Information Visualization Tool. In: 2012 16th International Conference on Information Visualisation (IV), 2012, Montpellier - France. pg 613-618.

Silva Junior, Jairo de Jesus Nascimento da; Meiguins; Bianchi Serique; Carneiro, Nikolas Jorge S.; Meiguins, Aruanda Simões Goncalves ; **FRANCO, Roberto Yuri da Silva**; Soares, Anderson Gregório Marques: PRISMA Mobile: An Information Visualization Tool for Tablets. In 16th International Conference on Information Visualisation (IV), 2012, Montpellier - France. pg 182-187

RESUMO

Os sistemas de informação atuais têm produzido um grande volume de dados. O processo de análise visual dessas informações tem tomado mais tempo, tem apresentado dificuldades de interação com os dados e há informações não perceptíveis ao usuário em função da oclusão e agregação. A área de visualização da informação, e áreas relacionadas, tem buscado ao longo do tempo superar essas dificuldades, seja por evolução no hardware como aumento de espaço visual e poder de processamento, evoluções em software como melhoramento de técnicas e algoritmos, ou evolução de software mais hardware que garanta a escalabilidade de crescimento dos dados. Nesse contexto, essa dissertação objetiva minimizar os problemas encontrados na manipulação visual de grande quantidade de dados, propondo uma arquitetura escalável e distribuída para aplicações de visualização de informação com múltiplas visões coordenadas por meio do PRISMA-MDE. Os benefícios da arquitetura são: melhorar o poder de processamento para ferramentas de visualização, diminuir o tempo de renderização das técnicas de visualização, diminuir o problema de oclusão e agregação, entre outros. Serão apresentados testes de usabilidade, e testes comparativos de desempenho (tempo de renderização, uso da memória, carga de processador) com uma aplicação de visualização de informação em um único desktop.

Palavra-chave: Visualização da Informação, Visões Coordenadas, Coordenadas Paralelas, Treemap, Ambiente de Múltiplas Telas, PRISMA, Análise de dados.

ABSTRACT

The current information systems have produced a large volume of data. The process of visual analysis of this information has taken longer, it has presented difficulties to interact with data, and there are information that is not perceived by the user because of occlusion and aggregation. The area of information visualization, and related areas, has sought over time to overcome these difficulties, either by hardware evolution as increased visual space and processing power, as developments in software improvement techniques and algorithms, and software evolution plus hardware that ensures scalability of data growth. In this context, this dissertation aims to minimize the problems encountered in the handling of large amounts of visual data by proposing architecture for scalable and distributed visualization applications with multiple views of information coordinates. The benefits of this architecture are improving the processing power for visualization tools; reduce the rendering time of visualization techniques, reducing the problem of occlusion and aggregation, among others. We will be presented usability testing and comparative testing of performance (rendering time, memory usage, processor load) with an application of information visualization on a single desktop.

Keywords: Information Visualization, Multiple Coordinated Views, Parallel Coordinates, Treemap, Multi-Display Environment, PRISMA, Big Data, Big Analysis.

1 INTRODUÇÃO

O processo de tomada de decisão em qualquer área do conhecimento humano pode ser complexo, dependendo de variáveis como: contexto do problema, quantidade de dados para análise, o tempo para realizar a tarefa especificada, diversidade de conhecimento necessário para análise dos dados, precisão da análise, entre outras.

O problema torna-se exponencialmente maior em função da quantidade de dados eletrônicos que são guardados diariamente em todo mundo, em várias áreas e setores: universidades, governo, e iniciativa privada. Estimativas apontam que ao final de 2013 o mundo terá aproximadamente quinhentos exabytes de dados armazenados eletronicamente (Hilbert & López, 2011).

Para apoiar esse processo, uma diversidade de técnicas e ferramentas computacionais para diferentes áreas foram desenvolvidas, dentre essas se destaca a área de visualização de informação. A visualização de informação é uma área de pesquisa que estuda a transformação de dados abstratos em imagens, onde os dados são facilmente visualizados e compreendidos pelos seres humanos (Spence R. , 2001).

Desta forma, o uso de técnicas e ferramentas de visualização possibilita aos usuários melhor percepção sobre os dados e seus relacionamentos, com isso o processo de análise se torna mais rápido, fácil e intuitivo, e preciso uma vez que padrões, grupos, e *outliers* dos dados saltam aos olhos através de representações gráficas.

Nos processos cotidianos de estudo, trabalho e pesquisa, é comum e necessário que um grupo de pessoas esteja reunido para debater algum assunto e enriquecer o processo de análise de dados. Então o desafio passa a ser: como apresentar uma diversidade de dados de forma que a percepção de um grupo de usuários localizados em uma mesma sala seja rápida, fácil e intuitiva?

O uso de diversas áreas e técnicas de visualização é recomendado quando há uma grande quantidade ou diversidade de dados para análise. Com isso, os analistas lidam

com uma quantidade massiva de dados multidimensionais, oriundos de múltiplas fontes e temporalmente dependentes para tomar decisões efetivas em situações onde o fator tempo é decisivo. Embora vários problemas de análise possam ser resolvidos utilizando-se apenas métodos puramente analíticos, outros requerem a flexibilidade e poder do sistema cognitivo humano (Ware, 2004). Em face desta complexidade, é necessário utilizar ferramentas que permitam aos analistas aproveitarem suas capacidades inatas de percepção de modo a potencializar o processo de descoberta de informação, e em última instância melhorar a precisão da tomada de decisão (Thomas & Cook, 2005).

Em virtude do problema da crescente escala dos dados, uma das grandes limitações para as ferramentas de visualização de informação é a “falta de espaço” para representação dos dados na tela. Muito dessa limitação é impulsionada pela possibilidade de utilização de múltiplas técnicas de representação (diferentes) de dados simultaneamente. Dessa forma, o espaço antes dedicado a uma visão dos dados, é dividido entre duas ou mais visões. Este problema resulta em consequências desagradáveis para o usuário. A maior delas está relacionada à oclusão de dados em técnicas que representam os itens individualmente, como os gráficos de dispersão de dados e gráficos de coordenadas paralelas, dentre outras técnicas de visualização da informação. A oclusão ocorre quando dois ou mais elementos visuais se sobrepõem, obstruindo a percepção de alguns deles, e ocasionando má interpretação da visão ou perda de detalhes e relações importantes (Novotny, 2004).

Este trabalho apresenta uma ferramenta de visualização de informação interativa com múltiplas visões de dados coordenadas em um ambiente distribuído e escalável de múltiplas telas. Cada computador é responsável por renderizar e responder às interações de uma técnica de visualização (*treemap*, coordenadas paralelas ou dispersão de dados). Dessa forma, cada tela é dedicada a apenas um gráfico e a carga imposta pela resposta às interações coordenadas é distribuída entre os nós.

Assim, o desenvolvimento desta dissertação quer proporcionar além do ambiente distribuído e escalável de múltiplas visões coordenadas para múltiplas telas, diretrizes para o desenvolvimento de ambientes MDEs e um estudo comparativo entre uma ferramenta tradicional de visualização da informação e um ambiente de múltiplas telas para visualização.

1.1 Objetivos

Esse trabalho pretende conceber e desenvolver um método de que busque minimizar o problema da oclusão e agregação de dados e problemas de processamento de grandes bases de dados.

1.1.1 Objetivos Específicos

Os objetivos específicos são apresentados a seguir:

- Propor diretrizes para o desenvolvimento de ambientes de múltiplas telas para ferramentas de visualização de informação.
- Melhorar o tempo de processamento para grande quantidade de dados.
- Possibilitar o carregamento de base de dados maiores.
- Facilitar as tarefas de comparação, identificação e a análise dos dados.
- Possibilitar uma área maior para o renderização de técnicas de visualização de informação.

1.2 Metodologia

A metodologia empregada para o desenvolvimento do trabalho foi dividido em duas partes. A seguir serão apresentadas as partes, bem como uma descrição de como essas serão executadas.

O primeiro passo do trabalho é o desenvolvimento de um ambiente distribuído e escalável para ferramentas de visualização da informação em múltiplas telas, que será desenvolvida a partir da ferramenta PRISMA (Godinho, et al., 2007). Esta ferramenta foi escolhida por ser desenvolvida na plataforma Java, devido à familiaridade com o seu código e por atender todas as características que uma boa ferramenta de visualização de informação deve possuir, de acordo com as recomendações da comunidade científica (Shneiderman B. , 1996).

O PRISMA é uma ferramenta desktop baseada em múltiplas visões coordenadas, que conta com três técnicas principais: dispersão de dados (Spence R. , 2001), coordenadas paralelas (Inselberg, 2009) e treemap (Shneiderman B. , 1992). Essa ferramenta será utilizada para compor o ambiente de múltiplas telas, surgindo o ambiente PRISMA – MDE (*Multi-Display Environment*), que estende funcionalidade da ferramenta PRISMA para um ambiente distribuído e escalável.

A segunda parte é efetuar uma avaliação do ambiente através de testes de usabilidade e testes de desempenho. Os testes de usabilidade padronizados são

utilizados para provar algumas hipóteses desse trabalho. Nessa fase será executado um questionário com tarefas que usuário deverá realizar no ambiente, com isso será possível medir a eficiência, acurácia, exploração e percepção dos dados, logo após será feito o teste NASA TLX (*Task Load Index*) que é projetado para obter estimativas de carga de trabalho de um ou mais usuários, enquanto eles estão realizando uma tarefa ou imediatamente depois, buscando aferir o custo de realizar exigências da tarefa para usuário (Hart & Stavenland, 1988).

Algumas métricas de desempenho, tais como: carga de processador, uso da memória e tempo de execução foram coletadas em diversos contextos (base de dados diferentes) tanto para PRISMA Desktop quanto para PRISMA-MDE. O objetivo principal é realizar comparação entre a ferramenta PRISMA desenvolvida por (Godinho, et al., 2007) e o ambiente PRISMA-MDE projetado neste trabalho para confirmar ou não hipóteses formuladas sobre o ambiente e que serão apresentadas na seção 5.1.1.

1.3 Contribuição

De maneira geral, as principais contribuições deste trabalho são as análises e considerações sobre o desenvolvimento de um ambiente distribuído de visualização de informação. Além disso, é disponibilizar uma ferramenta que facilite a comparação de grandes bases de dados, melhorar o tempo de interação e processamento dessas bases, através da construção de um ambiente distribuído e escalável de múltiplas telas e processadores para ferramentas de visualização de informação. A principal contribuição deixada à ferramenta PRISMA é desenvolvimento do ambiente PRISMA-MDE, pois passou a permitir que a ferramenta processasse mais dados e possuísse um desempenho maior para a renderização das técnicas de visualização de informação.

1.4 Organização

A organização desta dissertação se dá da seguinte forma, na segunda seção serão apresentados os principais conceitos, um breve histórico sobre visualização de informação, um detalhamento das três técnicas principais de visualização de informação utilizadas nesse trabalho e os trabalhos relacionados na área de visualização.

Na seção três será apresentada a definição de ambiente de múltiplas telas, assim como uma breve definição para ambientes projetivos de múltiplas telas para Visualização de Informação.

A seção quatro inicia com a apresentação das características principais do ambiente de múltiplas telas PRISMA-MDE, bem como a arquitetura proposta e o detalhamento de cada módulo que compõem a arquitetura do ambiente MDE.

A quinta seção, apresenta as formas de avaliação do ambiente proposto, através de uma avaliação de usabilidade e uma avaliação de desempenho. Nesta seção também são apresentados os resultados dessas duas avaliações.

Por último, na sexta seção são apresentadas as considerações finais, os desafios encontrados, e também os trabalhos futuros. Ao final são listadas as referências e como anexo 01 é apresentado o questionário utilizado na avaliação de usabilidade.

2 VISUALIZAÇÃO DA INFORMAÇÃO

De acordo com Plaisant (Plaisant, 2001), a área de visualização da informação objetiva a concepção de representações gráficas compactas e interfaces de software que permitam ao usuário a manipulação de um grande número de itens de dados visuais, possivelmente extraídos de grandes bases de dados.

Uma ferramenta de visualização criada com essas concepções permite ao usuário:

- Descobertas;
- Decisões;
- Apresentações.

Sobre:

- Padrões (tendências, agrupamentos, intervalos, exceções, etc);
- Grupos de itens;
- Itens individuais.

Card (Card, 2009) afirma que permitir ao usuário suporte computacional para visualizar e interagir com dados abstratos amplifica, ou reforça a cognição humana, habilitando o usuário a ganhar conhecimento sobre os dados e o relacionamento entre eles.

De maneira geral, aplicações de visualização de informação devem disponibilizar meios para transformar (através de técnicas de pré-processamento de dados), representar e apresentar dados (através de técnicas de visualização de informação), interagir com os dados (implementação e mecanismos como filtros, por exemplo, através de componentes de interface) para criar uma visualização interativa sobre os dados que permita e incentive a interação do usuário. A Figura 1 apresenta o resultado de aplicação de visualização de informação. Aplicações de Visualização da Informação

com essas características possibilitam aos usuários analisar os dados por exploração (navegando nos dados), em vez de raciocínio puro, e desenvolver a compreensão de estruturas e relacionamento sobre os mesmos, auxiliado pela imediata atualização da visualização de dados após a interação do usuário. Essas etapas descrevem de forma resumida o processo de visualização (Ward, Grinstein, & Keim, 2010).

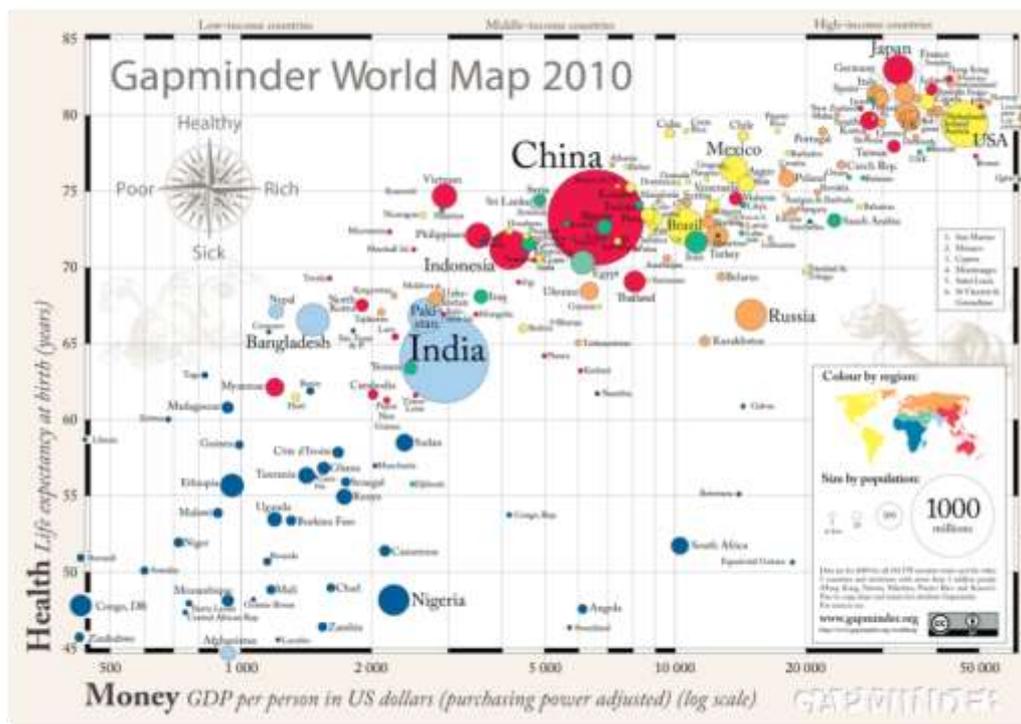


Figura 1. Exemplo de visualização da informação (Opazo, 2011)

Há um campo relacionado, e algumas vezes sobreposto, à visualização de informação chamada de “visualização científica”. A visualização científica se preocupa em representar visualmente uma simulação tridimensional de algo real (Spence R., 2001). Por exemplo, nuvens fluindo através de uma cadeia de montanhas, dada certa condição do vento, ou estudar o comportamento do ar e sua temperatura na decolagem e pouso de aeronaves (Figura 2). Por exemplo, uma imagem de ressonância magnética e uma imagem de raio-x são considerados Visualização Científica porque eles apresentam dados que possuem forma física, buscando representar fielmente esta forma de uma maneira que seja fácil de visualizar, reconhecer e compreender. Este texto não trata de visualização científica, entretanto muitas das técnicas que serão apresentadas são pertinentes às duas áreas.

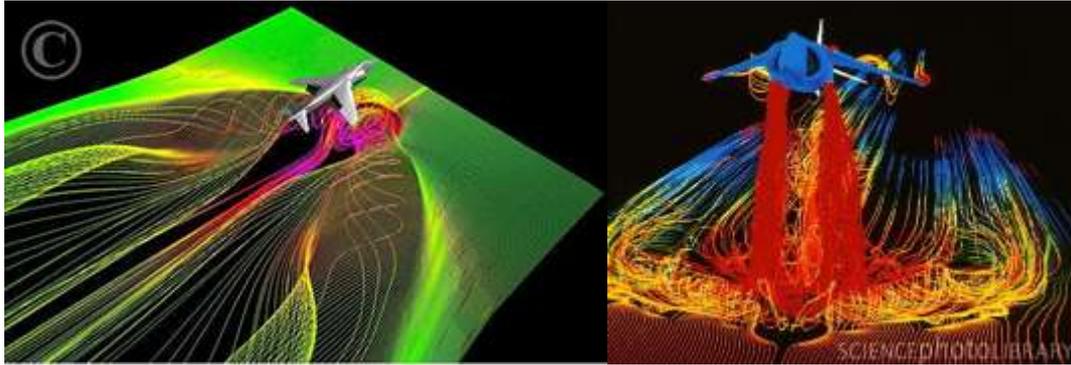


Figura 2. Estudo do fluxo de ar na decolagem de aeronaves (James & Leon, 2011)

2.1 Processo de Visualização

Normalmente, o processo de visualização da informação faz parte de um processo de análise exploratória de dados, descoberta de conhecimento, ou análise visual. A qualidade dessas análises é dependente da qualidade dos dados que são disponibilizados para interação, sendo necessário haver uma fase prévia para eliminação de erros ou ruídos dos dados.

Não há fórmula para uma visualização bem sucedida, e uma visualização de informação útil é dependente do usuário que a está utilizando, em função da sua experiência tanto com a ferramenta quanto no domínio do negócio. Por esse aspecto, é interessante que uma ferramenta de visualização seja flexível o suficiente para permitir adaptar-se aos diversos perfis de usuário: experientes, iniciantes, com mais ou menos habilidades no domínio do problema, até os mesmos se sentirem confortáveis a atingir seus objetos em relação à visualização proposta.

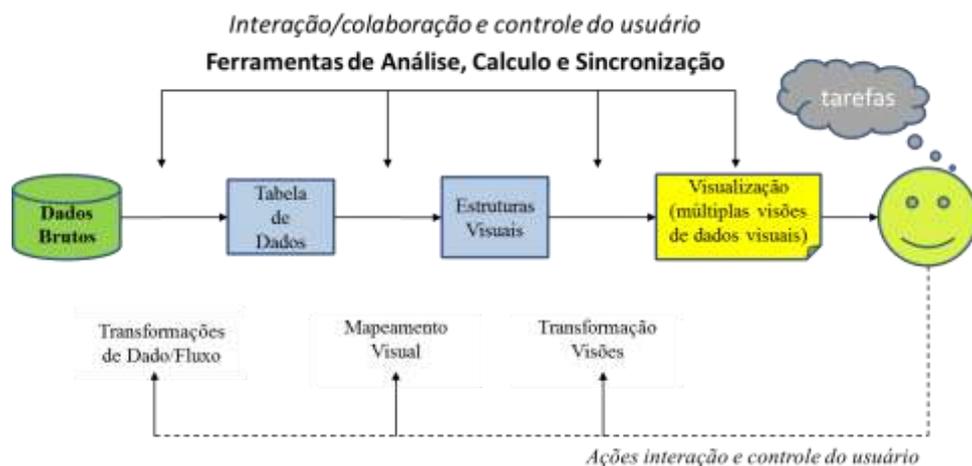


Figura 3. Pipeline de visualização (Ward, Grinstein, & Keim, 2010).

A interação com uma ferramenta de visualização ocorre em diversos níveis e gera diversos produtos intermediários (Figura 3). As principais etapas do processo/pipeline de visualização da informação são descritas a seguir:

- **Modelagem de dados:** necessidade de estruturar os dados que serão visualizados, em padrão a ser definido para possibilitar o uso de diversas fontes com layouts diferentes.
- **Seleção de Dados:** identificar um subconjunto dos dados potências a serem visualizados, que pode realizado manualmente pelo usuário ou de forma automatizada através de métodos algorítmicos;
- **Mapeamento Visual dos dados:** é a etapa em que os dados abstratos são vinculados a uma possibilidade de característica visuais, por exemplo, podem ser mapeados para característica de tamanho, posição, cor, forma, etc;
- **Configuração de parâmetros da Cena (Transformação da Visão):** a ferramenta deve permitir que o usuário pudesse especificar vários atributos da visualização que são relativamente independentes dos dados, por exemplo, cor, som, luz, etc;
- **Renderização ou geração da visualização:** renderização da visualização é a representação da imagem no dispositivo de saída, como um monitor LCD, e que varia de acordo com o mapeamento visual que esta sendo utilizado para dados. Além disso, a maioria das visualizações incluem informações complementares para facilitar a interpretação dos dados, tais como: eixos, anotações, imagens, etc;

2.2 Percepção e as Oito Variáveis Visuais

Por muitos anos os pesquisadores da visão humana tem investigado como o sistema visual humano analisa imagens. E um resultado inicial importante foi à descoberta de um conjunto limitado de propriedades visuais que são detectadas rapidamente e precisamente pelo sistema visual de baixo nível. Essas propriedades foram inicialmente denominadas de propriedades de pré-atenção, onde a detecção dessas propriedades ou características de não necessidade como passo preliminar da atenção ou foco do usuário (Ware, 2004) (Ward, Grinstein , & Keim, 2010). Exemplos de propriedades de pré-atenção podem ser observadas nos exemplos da Figura 4, respectivamente, propriedade

cor e forma. Uma propriedade de pré-atenção é caracterizada por permitir ao usuário identificar em uma grande quantidade de elementos visuais características diferentes com tempo menor dos 250 milissegundos.

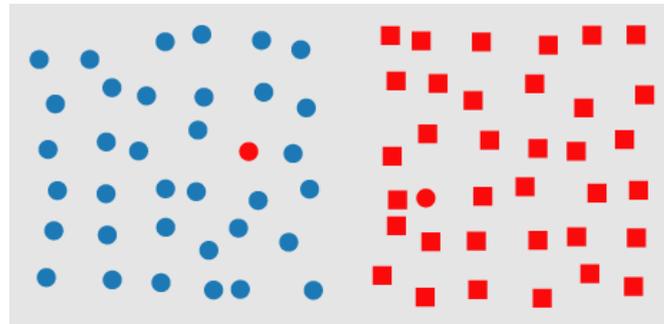


Figura 4. Exemplo de propriedades de pré-atenção.

As seguintes características visuais são classificadas como propriedades de pré-atenção, são elas: tamanho, largura, curvatura, quantidade, terminadores, interseções, grupos, tonalidades, forma, movimento, direção de movimento, etc. (Ward, Grinstein, & Keim, 2010).

As principais tarefas visuais que os usuários podem realizar com as propriedades de pré-atenção são:

- **Detecção de alvo:** usuário rapidamente e precisamente detectam a presença ou ausência de um elemento alvo com características visuais únicas dentro de um conjunto de elementos visuais;
- **Detecção de fronteira:** usuários rapidamente e precisamente identificam regiões de textura entre dois grupos de elementos, onde todos os elementos em cada grupo têm propriedades visuais comuns;
- **Rastreamento de região:** usuários podem rastrear um ou mais elementos com uma única característica visual quando eles se movem tanto no tempo quanto no espaço.
- **Contagem e estimativa:** usuários conseguem contar ou estimar o número de elementos com uma única característica visual.

Quando um usuário inicia o processo de análise de uma visualização, ou gráfico, primeiramente o usuário percebe grupos de objetos com características visuais semelhantes. Posteriormente, o usuário tenta caracterizar esses grupos.

Um trabalho pioneiro no relacionamento do uso de atributos visuais para utilizar o sistema de percepção humana de forma mais eficiente para representações gráficas 2D foi realizado por Jacques Bertin (Bertin, 1983). A Figura 5 apresenta um resumo deste trabalho.

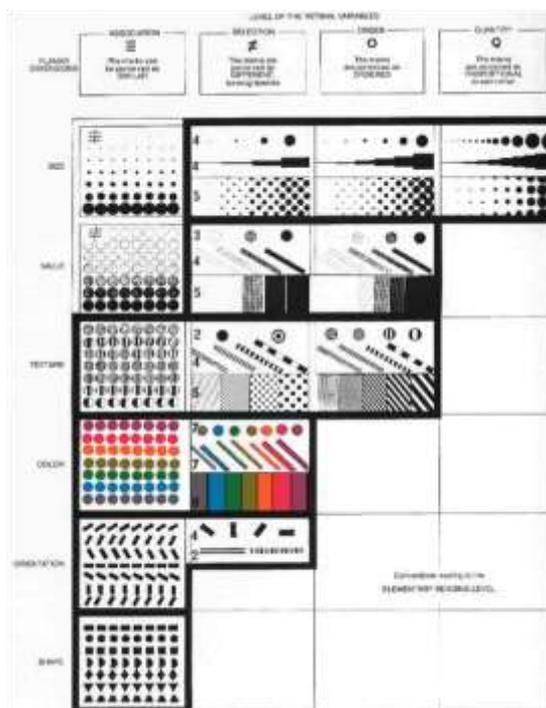


Figura 5. Recomendações para utilização de atributos visuais bidimensionais (Bertin, 1983).

Atualmente podem-se destacar oito variáveis visuais: posição, forma, tamanho, brilho, cor, orientação, textura, e animação (Ward, Grinstein, & Keim, 2010).

Posição: é a primeira e mais importante variável visual, uma vez que possibilita a organização dos elementos visuais em um espaço de visualização, que pode ser em uma, duas ou três dimensões.

Forma: basicamente formas são primitivas gráficas que representam dados, tais como: pontos, linhas, áreas, volumes e suas combinações. Qualquer objeto gráfico pode ser utilizado como forma, incluindo símbolos, letras e palavras.

Tamanho (comprimento, área e volume): as primeiras duas variáveis, posição e forma, são necessárias para definir a visualização, sem essas duas variáveis não é possível ter uma visualização. As próximas variáveis visuais afetam de maneira

individual as representações que serão visualizadas. Tamanho determina quanto pequeno ou grande a forma deverá ser desenhada.

Brilho: ou luminância, é a segunda variável visual utilizada para modificar formas para codificar dados adicionais. Para codificação de número, o brilho não permite distinguir dois valores exatos, mas é bem mais indicado seu uso para fornecer diferença relativa para variáveis contínuas e intervalares. Para uma identificação mais precisa é interessante aplicar poucas escalas de brilho.

Cor: o brilho afeta o quanto de branco ou escuro são visualizadas nas cores. As cores podem ser definidas por dois parâmetros: matiz e saturação, e são utilizadas para mapear valores individuais de dados.

Orientação: esta propriedade descreve o quanto a forma deve sofrer de rotação, e esta relacionada a um dado de valores variáveis. Não pode ser aplicada a todas as formas, círculos, por exemplo.

Textura: é considerada como uma combinação de muitas outras variáveis visuais, incluindo formas, cor e orientação. Texturas normalmente são associadas com polígonos, regiões, ou superfícies.

Animação: a animação pode ser associada com qualquer outra variável visual, desde que a variável mude ao longo do tempo apresentado mais informações.

Diferentes variáveis visuais podem servir a diferentes propósitos, uma das possíveis categorizações de uso e propósito é apresentada a seguir (Ward, Grinstein , & Keim, 2010):

- **Variáveis Visuais Seletivas:** tem objetivo de identificar todos os elementos de uma determinada categoria, são elas: tamanho, brilho, textura, cor, orientação;
- **Variáveis Visuais Associativas:** identificar a relação entre dois elementos em uma dimensão particular, ou todos os elementos do grupo com características semelhantes, são elas: textura, cor, orientação, forma;
- **Variáveis Visuais de Ordenação:** são úteis para identificar a ordenação de uma dimensão particular.

2.3 Tipos de Dados para Visualização da Informação

Shneiderman classificou os dados em sete tipos diferentes, o tipo de dado tem grande influência na escolha da codificação visual, e, por conseguinte da técnica de

visualização a ser utilizada (Shneiderman B. , 1996), elas estão descritas nas subseções a seguir:

2.3.1 Unidimensionais (1D):

Tipo de dados linear incluem listas sequenciais, documentos de texto, código fonte de programas, etc. e são apresentados em alguma ordem sequencial (Figura 6).



Figura 6. Exemplo de Word Tree (Lent, 2007).

2.3.2 Bidimensionais (2D)

Os dados do tipo bidimensional apresentam um número de atributos, que é normalmente utilizado diretamente para sua representação visual em 2D. Exemplos são a longitude e latitude, largura e altura, etc., utilizados para representar dados mapas geográficos, plantas de engenharia, plano cartesiano, etc. A Figura 7 apresenta um heatmap com informações de cluster, considerando-se apenas uma coluna, como gear, tem-se o posicionamento de cada carro e cor que representa um valor referente a cada cor e carro, sendo possível comparar a variação de cores.

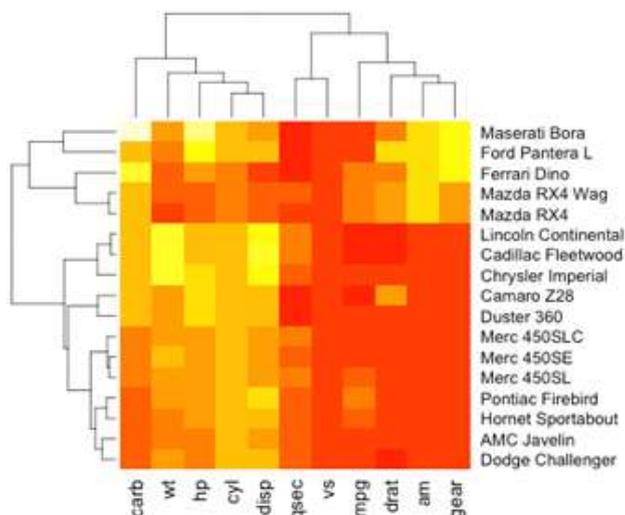


Figura 7. Exemplo de Heatmap com cluster (Bare, 2013).

2.3.3 Tridimensionais (3D)

Este tipo de dado possui as características dos dados bidimensionais incorporando a informação de volume. A Figura 8 apresenta uma representa gráfica 3D de crimes em São Francisco.

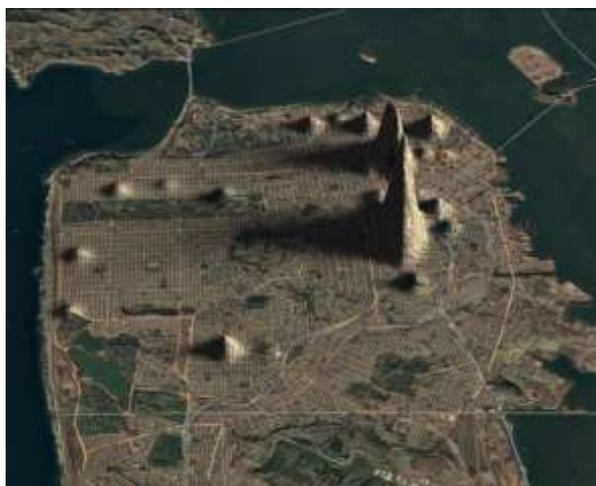


Figura 8. Imagem 3D sobre as estatísticas de prostituição na cidade de São Francisco–EUA (McCune, 2013).

2.3.4 Multidimensional

Este tipo de dados representa dados que primariamente não são espaciais. O número de atributos associado a cada item é sempre mais que três. Os dados utilizados podem ser abstratos e de natureza estatística, como por exemplo: dados de produção fabril, base de dados cinematográfica, distribuição de livros numa biblioteca, etc. A

Figura 9 apresenta um conjunto de matrizes correlacionadas caracterizando a multidimensionalidade de dados.

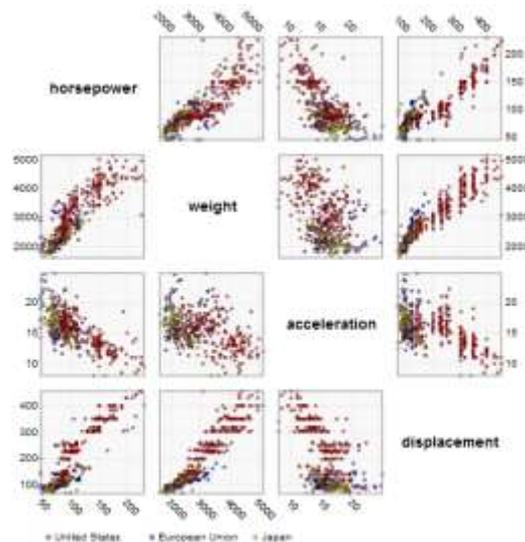


Figura 9. Exemplo de Scatter Plot Matrix (SPLOM) (Heer, Bostock, & Ogievetsky, 2008)

2.3.5 Temporal

Tipo de dado onde o atributo tempo é incorporado aos demais tipos descrito. Deve-se observar que dados temporais podem relatar eventos que ocorreram simultaneamente ou de forma sobreposta, e a escala de tempo pode ser diferente nos dois eventos (por exemplo, um em milissegundos e outro em minutos), e a representação visual escolhida deve representar essas necessidades. A Figura 10 é uma representação 3D da visualização feita por Minard, originalmente em 2D, sobre a marcha de Napoleão a Moscou.

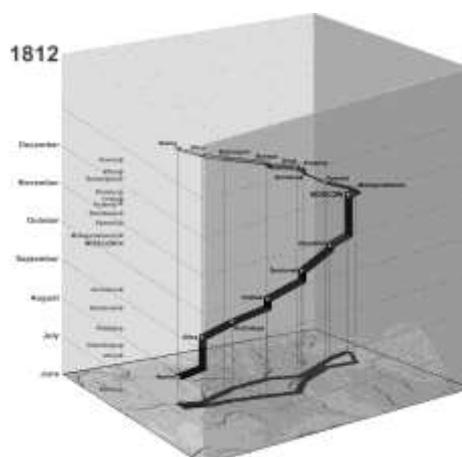


Figura 10. Exemplo de Visualização para dados temporais. (Kraak, 2003)

2.3.6 Hierárquico

Este tipo de dados também é conhecido como estrutura em árvore. Eles são conjuntos de dados dispostos em nós, onde cada nó possui apenas um nó pai (situado hierarquicamente acima deste), e pode ter vários nós filhos (situados hierarquicamente abaixo deste). Os nós e ligações podem ter múltiplos atributos. A própria estrutura hierárquica traduz informação relevante. A análise de hierarquias pode focar um único nó, ligação, coleção de nós ou até mesmo toda a estrutura (Figura 11).

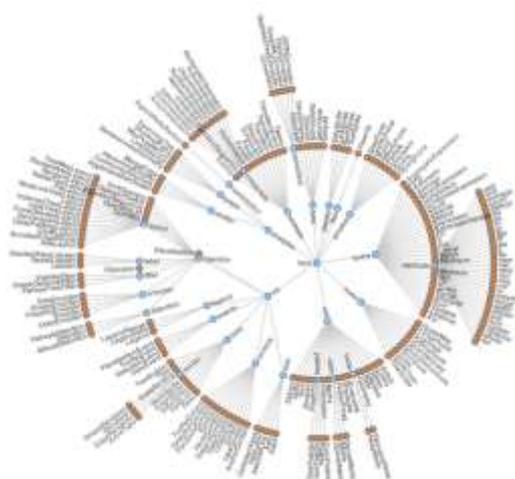


Figura 11. Exemplo de diagrama node-link. (Kenttälä, 2011)

2.3.7 Rede de dados

São nós conectados por links previamente definidos. Esses links podem ser organizados em árvores ou em hierarquias, e a melhor maneira de manipulação é permitindo mudar o foco sobre os nós (Figura 12).

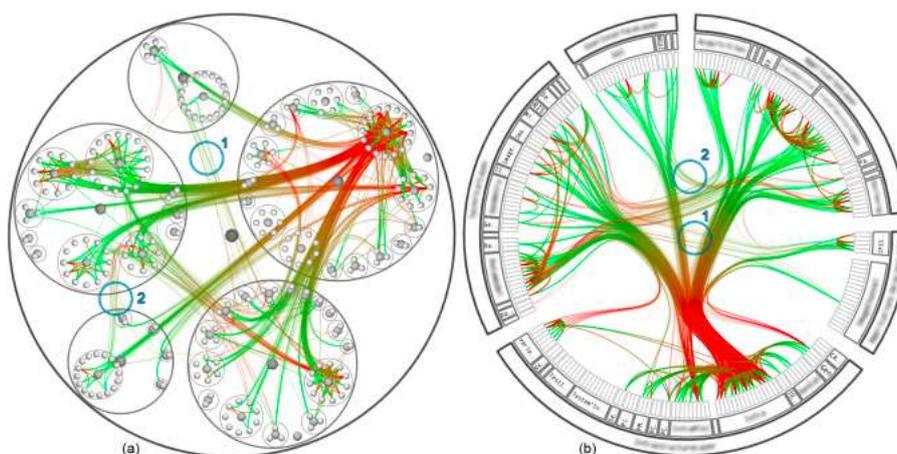


Figura 12. Exemplo de Visualização em dados do tipo rede. (Bertini & Stefaner, 2012)

2.4 Técnicas de visualização da informação

Algumas técnicas serão mencionadas ao longo do texto e aparecem com maior frequência, entre elas destacam-se: Dispersão de Dados (gráfico X-Y), Coordenadas Paralelas e Treemap.

2.4.1 Dispersão de Dados

A técnica de dispersão apresentada, também chamada de Gráfico de Dispersão 2D, possibilita visualizar uma relação (correlação) entre duas variáveis X e Y (por exemplo, peso e altura). São representados pontos de dados individuais no espaço bidimensional onde os eixos representam as variáveis (X no eixo horizontal e Y no eixo vertical). A técnica de Dispersão de Dados possibilita uma análise multidimensional à medida que são disponibilizadas em uma ferramenta de visualização de informação configurações de eixos, forma, tamanho, cor, etc. (Figura 13).

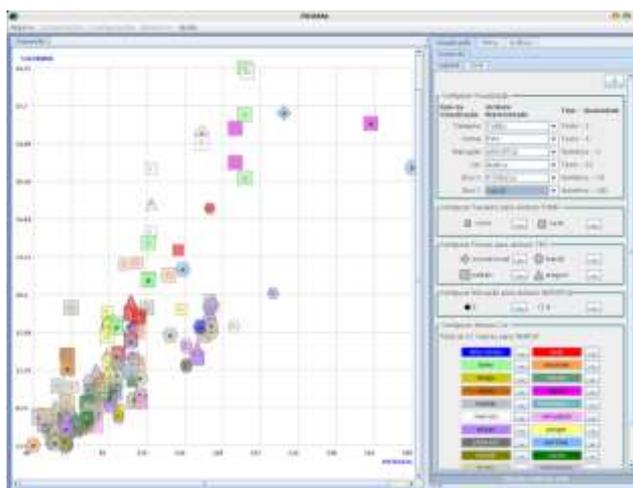


Figura 13. Exemplo de Técnica de Dispersão de Dados multidimensional.

2.4.2 Coordenadas Paralelas

A técnica consiste em interligar os eixos aos seus adjacentes através de linhas retas (Figura 14). Essas linhas são traçadas de acordo com os registros da base de dados. Cada eixo representa uma escala que contém todos os valores possíveis para um determinado atributo (Inselberg, 2009).

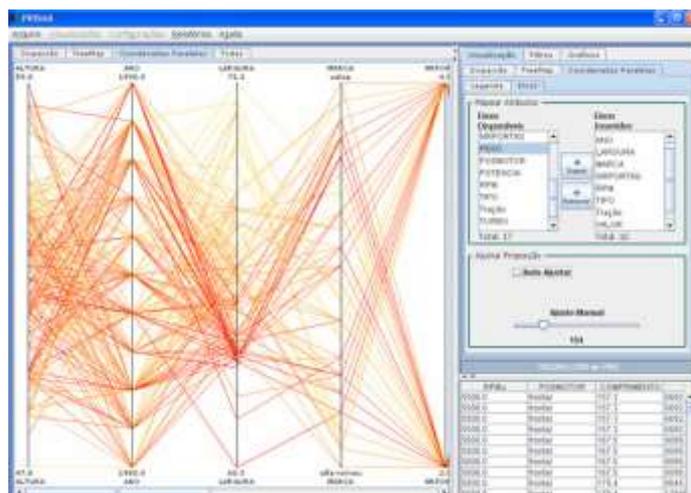


Figura 14. Exemplo de aplicação da técnica de Coordenadas Paralelas

2.4.3 Treemap

Treemaps são métodos de visualização de preenchimento de espaço, capazes de representar grandes coleções hierárquicas de dados quantitativos (Shneiderman & Wattenberg, 2001) (Shneiderman B. , 2009). A função de um Treemap basicamente é dividir um espaço em vários retângulos, onde a área de cada retângulo se relaciona ao valor de um atributo em um registro específico de um conjunto de dados. Portanto, quanto maior o valor deste atributo maior será a área do retângulo que representa um item de dado ou um conjunto de dados. Os algoritmos utilizados para a criação das sequências de retângulos no espaço de preenchimento são cluster e squarified. Além de ser uma técnica de ocupação de espaços, o Treemap possibilita a organização dos dados hierarquicamente (Figura 15), e desta forma, permitindo realizar comparações entre grupos de itens.

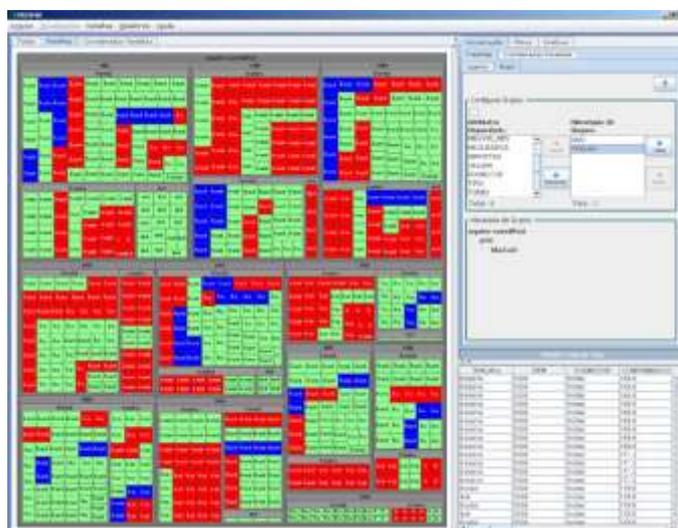


Figura 15. Organização hierárquica dos dados no Treemap

2.4.4 Características de uma boa Ferramenta de Visualização

(Shneiderman B. , 1996), (Carr, 1999) e (Card, 2009) sugeriram características mínimas que uma boa ferramenta de visualização deveria ter, com base nas tarefas que usuário poderia executar, são elas:

- **Visão geral:** o usuário precisa ganhar uma noção sobre todos os dados que serão analisados, para depois focar em um subconjunto de dados de interesse. Deve-se considerar tipo de técnica de visualização escolhida, limitações do hardware de visualização, percepção do usuário, etc.
- **Zoom Semântico:** é possível focar a análise visual em subconjunto de dados, sem perder a noção do todo, com a possibilidade de informações adicionais na visualização.
- **Filtro:** uma tarefa importante no processo de análise visual é a redução de tamanho do conjunto de dados de análise, denominado filtro, que se dá eliminando itens da visualização com base em seus atributos. O filtro de dados pode ser realizado através de componentes de interface ou pela interação na própria visualização dos dados.
- **Detalhes sob demanda:** quando os usuários estão explorando um conjunto de dados, eles necessitarão ver detalhes adicionais sobre um item em particular. Assim, a ferramenta precisa disponibilizar mecanismos de interação que possibilitem recuperar essas informações adicionais sobre o item ou grupo de análise.
- **Relacionamento:** se o usuário descobre um item de interesse, a ferramenta deveria proporcionar uns mecanismos para encontrar itens semelhantes para análise.
- **Histórico:** o usuário precisa de suporte para desfazer uma ação, mostrar os passos percorridos até aquele ponto.

2.4.5 Múltiplas Visões Coordenadas de Dados

O processo de exploração e análise de dados requer que a ferramenta de visualização da informação que permita ao usuário fazer a seleção de itens ou grupos de dados, relacionamento de itens, navegação em estruturas, visão geral, mapeamento de atributos visuais, etc. (Heijs, 2007).

Para grandes conjuntos de dados às ferramentas de visualização possibilitam a utilização de várias técnicas de visualização para suporte ao processo de exploração e análise. Há vantagens e desvantagens nessa abordagem, a principal vantagem é minimizar as deficiências de cada técnica de visualização da informação em função do conjunto de técnicas que podem ser utilizadas, a principal desvantagem está na possível sobrecarga cognitiva para relacionar as várias visões e o entendimento das visões em si, e na possível falta de espaço no dispositivo de visualização para as diferentes técnicas. (Convertino, Chen, Yost, Ryu, & North, 2003).

Para facilitar a associação que o usuário tem de fazer entre as diferentes visões, técnicas de interação e coordenação são utilizadas junto às visualizações para destacar o relacionamento entre as visões, esta área é denominada de Múltiplas Visões Coordenadas (CMV - *Coordinated and Multiple Views*) (Carpendale, 2007). Geralmente, sistemas que utilizam CMV possuem mais de uma técnica de visualização para explorar um mesmo conjunto de dados, conjuntos de dados diferentes ou diferentes subconjuntos de um único conjunto de dados.

Toda aplicação que utiliza CMV necessita de espaço em tela para exibir simultaneamente as diferentes visões. Dependendo do número de visões é difícil para o usuário conseguir observar em uma única tela todas as visões.

Atualmente a NASA apresenta uma arquitetura que incorpora um conjunto de 128 monitores, os quais podem exibir uma cena em conjunto ou cenas separadas. Essa arquitetura, chamada de Hyperwall 2, é uma evolução da Hyperwall (Ellsworth, Green, Henze, Moran, & Sandstrom, 2006), e foi proposta com o intuito de visualizar dados multidimensionais e simulações de forma coordenada.

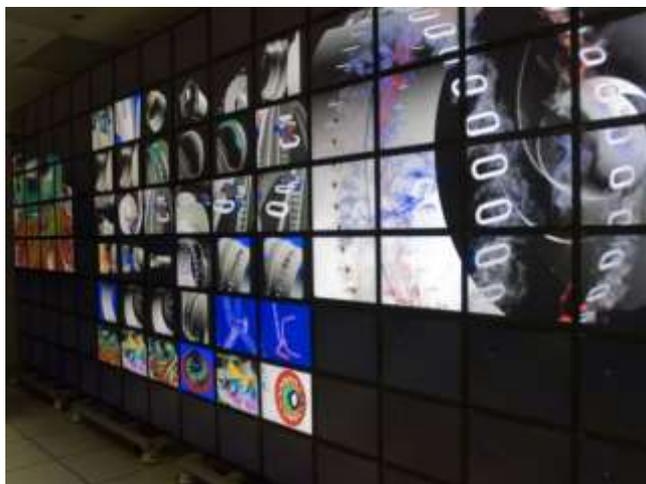


Figura 16. Exemplo do Hyperwall 2. (Dunbar, Hardman, & Thigpen, 2013)

Infelizmente o sistema da NASA, Hyperwall 2 não está disponível em larga escala para uso. Mas a ideia é válida e deve ser implementada em menor escala, o que deve abranger a grande maioria das demandas de visualização da informação. Para o desenvolvimento de sistemas de visualização de informação com múltiplas visões coordenadas, as recomendações mais frequentes são (Baldonado, Kuchinsky, & Woodruff, 2000):

- Quando há uma diversidade de atributos, modelos, perfis de usuário, níveis de abstração ou gênero.
- Quando as visões diferentes destacam correlações ou disparidades.
- Quando há necessidade de diminuir a complexidade do conjunto de dados, utilizando múltiplas visões mais simples.
- Usar múltiplas visões minimamente, justificar o uso de múltiplas visões versus custo de aprendizado do usuário e espaço de visualização.

Em relação ao aspecto de coordenação entre as visões, uma boa ferramenta de visualização deveria possibilitar: relacionamento entre itens visuais de visões diferentes, filtro, ações de configuração de características visuais como cor, transparência e tamanho, ordenação de itens visuais nas visões, permite ao usuário adicionar / remover atributos das visões de dados (Pillat & Freitas, 2006).

Os principais desafios no desenvolvimento de sistemas MVC (Pillat, Valiati, & Freitas, 2005), (Baldonado, Kuchinsky, & Woodruff, 2000), (North & Shneiderman, 2000):

- Os mecanismos de coordenação.
- Requisitos computacionais para renderização das visões.
- Disposição da interface – layout, com espaço normalmente muito reduzido.
- Interação do usuário entre as diversas formas de visualização.
- Aspectos cognitivos relacionados ao uso de sistemas de múltiplas visões coordenadas:
 - Tempo e esforço necessário para o aprendizado do sistema.
 - Sobrecarga de informações na memória de trabalho do usuário.
 - Esforço necessário para comparação.

- Esforço necessário para troca de contexto.

2.5 Trabalhos Relacionados

Nesta seção são apresentadas ferramentas com funcionalidade semelhantes à ferramenta proposta nesse trabalho. Serão apresentados três trabalhos sobre visualização de informação. O terceiro trabalho apresentado nesta foi o ponto de partida para o desenvolvimento do ambiente PRISMA-MDE.

2.5.1 *Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata.*

Snap-Together é uma ferramenta de visualização de informação que permite aos usuários de forma rápida e dinâmica misturar e combinar visualizações e coordenações para construir interfaces de exploração personalizadas sem programação. O modelo conceitual da ferramenta é baseado no modelo de base de dados relacional. Os usuários carregam as relações de carga em visualizações e em seguida coordenam com base relacional. Os usuários podem criar diferentes tipos de coordenações, tais como: *brushing*, *drill down*, visão geral, detalhes sob demanda, e rolagem sincronizada. Desenvolvedores podem fazer suas visualizações independentes com uma API simples. A avaliação do *snap-together* revela benefícios, problemas cognitivos, e preocupações de usabilidade (North & Shneiderman, 2000)

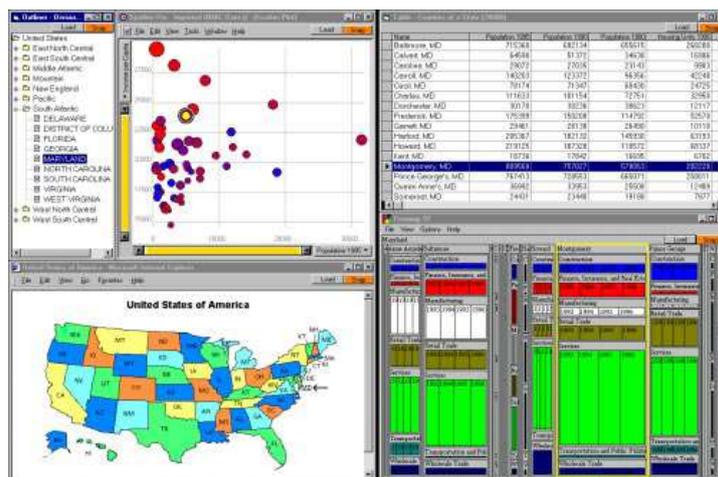


Figura 17. Imagem da ferramenta *Snap-Together* (North & Shneiderman, 2000).

2.5.2 *Coordinated Views to Assist Exploration of Spatio-Temporal Data.*

Este trabalho é uma ferramenta de visualização de informação que fornece estratégias de visualizações coordenadas para ajudar os usuários na realização de tarefas

impulsionados pela presença de atributos temporais e espaciais. Essa ferramenta possui técnicas de visualização temporais destinados a essas tarefas, para ilustrar a sua utilização como uma aplicação que envolve um processo de classificação climática. A classificação climática requer um processamento extenso de um banco de dados contendo os valores diários de precipitação de chuva coletados ao longo de mais 50 anos em vários locais no Estado de São Paulo, Brasil (Shimabukuro, Flores, Oliveira, & Levkowitz, 2004).

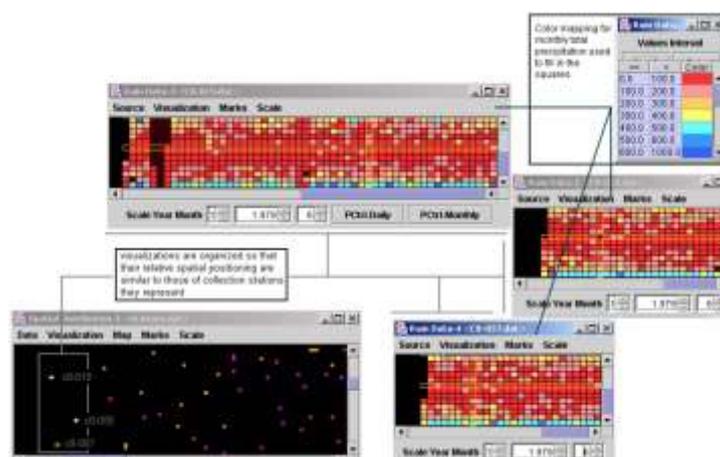


Figura 18. Imagem da ferramenta proposta por (Shimabukuro, Flores, Oliveira, & Levkowitz, 2004).

2.5.3 PRISMA - A Multidimensional Information Visualization Tool using Multiple Coordinated Views.

A ferramenta PRISMA é uma ferramenta desktop baseada em múltiplas visões coordenadas, que conta com três técnicas principais: dispersão de dados, coordenadas paralelas e treemap.

PRISMA suporta os principais recursos considerados imprescindíveis pela comunidade de visualização em uma ferramenta visualização de informação: visão geral (representação gráfica de todos os dados, e apresenta a dimensão completa do conjunto a ser analisado), zoom (Um foco gráfico e específico em subconjunto de dados apresentados visualmente ao usuário), filtros (Redução do conjunto de dados, diferentemente do zoom os filtros reduzem a base de dados e não apenas aplicam um foco em uma região específica), detalhes sob demanda (informações adicionais de um item do conjunto de dados), apresenta relacionamento entre os dados e histórico de ações (mantém as temporariamente as ações do usuário armazenadas, dando opção de

desfazer). Além disso, o PRISMA oferece portabilidade, usabilidade e extensibilidade. (Godinho, et al., 2007).

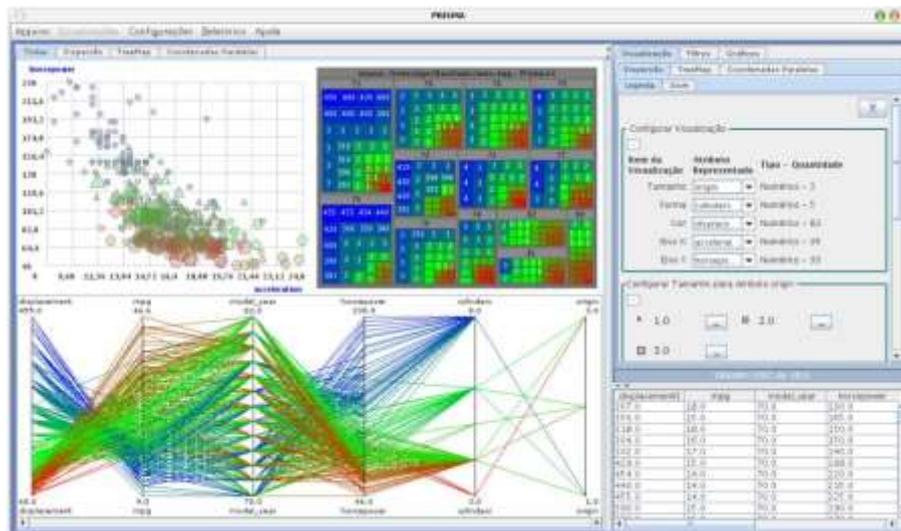


Figura 19. Imagem da ferramenta PRISMA

3 AMBIENTES MULTIPLAS TELAS

Ambientes múltiplas telas (*MDEs*) estão se tornando cada vez mais comum. Com a introdução de técnicas de multimodais de interação, como o olhar e gestos, pode estabelecer uma interface sofisticada e intuitiva para *MDEs* onde as telas são utilizadas facilmente e dinamicamente de acordo com os pontos de vista dos usuários. Cada usuário pode interagir com os vários monitores como se ele estivesse na frente de um simples ambiente desktop. Esses ambientes compõem telas que podem estar em diferentes locais e de diferentes ângulos para o usuário.

Como resultado, pode se tornar muito difícil gerir as janelas, ler o texto, e manipular objetos. Para evitar isto, deve-se primeiro identificar os fatores básicos de exibição e controle que são afetados pela perspectiva, como visibilidade e compartilhamento. Alguns estudos mostram que a correção da perspectiva representa uma melhora significativamente substancial no desempenho do usuário em todas essas tarefas.

3.1 Definição

Ambientes de Múltiplas Telas (*MDE - Multi-Display Environments*) são sistemas de computadores onde vários monitores são empregados para aumentar a área visível do usuário, geralmente é empregado, quando é necessário utilizar muitos recursos visuais ao mesmo tempo. Na área de visualização de informação esse recurso é muito utilizado para superar o problema da falta de espaço para as visualizações da interface gráfica (Johanson, 2002.), outro problema que este tipo de ambiente ajuda a minimizar é o já citado problema de oclusão de dados.

Os monitores podem ser de diferentes tipos, tamanhos e marca (Forlines & Lllien, 2008.) e estarem conectados a uma ou várias máquinas (Johanson, 2002.). Ocorre ganho de desempenho, quando é utilizado um monitor por máquina, onde as várias máquinas têm o seu poder de processamento concatenado para o processamento das necessidades

do ambiente, fazendo assim, um sistema distribuído. No caso de *MDEs* para ferramentas de visualização é recomendado que as tarefas de renderização das técnicas e processamento das bases de dados sejam distribuídas entre as máquinas.

Existem variadas técnicas de interação para os ambientes de múltiplas telas, de modo a explorar suas possibilidades específicas. Uma questão importante é o modo como o sistema lida com dispositivos de entrada e como o conteúdo é apresentado através das várias telas. Vários projetos baseados nesse tipo de ambiente aplicam redirecionamento de entrada ou redirecionamento de conteúdo para interação com os dispositivos (Wallace, Mandryk, & Inpen, 2008.).

No redirecionamento de entrada, o usuário controla um único dispositivo de entrada para interagir com objetos nas diversas telas, ou seja, para alcançar um item em uma tela, o usuário precisa deslocar o mouse até lá.

Há duas abordagens principais para permitir esse deslocamento. No modo *Extended Desktop*, o usuário movimenta o cursor do mouse por entre as telas, como se estivesse interagindo com um único *desktop*. No modo *Multi-Monitor Mouse*, o usuário movimenta o cursor de uma tela para outra pressionando uma tecla do teclado.

O redirecionamento de conteúdo permite que o conteúdo de uma tela seja visualizado em outra. Ou seja, o usuário utiliza uma única tela para interagir com o conteúdo das diversas outras. Ao contrário do redirecionamento de entrada, não é o cursor que se movimenta, mas sim o conteúdo.

3.1.1 Ambiente projetivo de múltiplas telas para visualização de informação

O grande número de pixels disponíveis em ambientes projetivos de múltiplas telas permite que múltiplas visões de conjuntos de dados sejam apresentadas simultaneamente, o que pode ajudar não só a compreensão do usuário de um conjunto de dados, mas também na coordenação de um grupo de trabalho conjunto (Forlines & Lillian, 2008.). Um ambiente de projeção pode ser uma solução para o compartilhamento de visualizações com muitas pessoas.



Figura 20. Ambiente projetivo de visualização da Informação múltiplas telas (Ferramenta PRISMA)

A Figura 20 ilustra como funciona o ambiente projetivo de múltiplas telas do ambiente PRISMA-MDE. Este ambiente funciona com duas ou mais telas projetadas e para cada tela um computador que processa e gera as imagens.

Um problema presente em ambientes de projeção para visualização da informação é forma de interação do usuário com o ambiente. Outros métodos de interação como: teclado, mouse, comando por voz, comando por gestos, notebook e tablete, pode ser empregados nesse tipo de ambiente.

3.2 Trabalhos Relacionados

Foram utilizados seis trabalhos sobre ambientes de múltiplas telas. O primeiro trabalho utilizado foi *Design Considerations for Collaborative Information Workspaces in Multi-Display Environments* e foram empregadas as considerações de design para a aplicação de visualização com múltiplas visões proposto por (Waldner, 2009), como o foco do trabalho não é ambientes colaborativos, então, não foram utilizadas as considerações de design para ambientes colaborativos de múltiplas telas. Outro trabalho utilizado foi o *Three Modes of Multi-Surface Interaction and Visualization* de onde foi utilizada a taxonomia de estilos de visualização com múltiplas visões em ambientes de múltiplas telas propostas no trabalho (Shen, 2006).

Além desses trabalhos foram utilizados os conceitos de redirecionamento de entrada proposto no trabalho: “*PointRight: Experience with Flexible Input Redirection in Interactive Workspaces*” de (Johanson, 2002.). Também foi utilizado outro trabalho que discute a adaptação de sistemas de única tela e ambientes múltiplas telas intitulado:

“*Adapting a Single-User, Single-Display Molecular Visualization Application for Use in a Multi-User, Multi-Display Environment*”, proposto por (Forlines & Lillian, 2008.).

Também foi utilizado um trabalho que discute considerações de design e interação para ambientes múltiplos dispositivos, intitulado: “*Code Space: Touch + air Gesture Hybrid interactions for Supporting Developer Meetings*” proposto por (Bragdoni, 2011). Por último, será apresentado o *framework libGlass* que é uma API para computação distribuída, muito utilizada em realidade aumentada e virtual (Guimarães, Gnecco, Bressan, & Zuffo, 2003), esse *framework* foi usado para efetuar a computação distribuída deste trabalho e para isso foi feita algumas melhorias na parte de dispositivos de entrada do ambiente. Os trabalhos relacionados utilizados nesse projeto serão descritos abaixo.

3.2.1 *Design Considerations for Collaborative Information Workspaces in Multi-Display Environments*

Esse trabalho apresenta uma série de considerações de projeto para empregar múltiplas visões em ambiente colaborativo de múltiplas telas que são: adaptações da visualização dependendo de fatores de exibição e preferências do usuário, técnicas de interação para facilitar o compartilhamento de informações e para guiar a atenção dos usuários para relevantes itens no ambiente, bem como a concepção de um ambiente de trabalho flexível, ajustável para diferentes tamanhos de grupo e tarefas específicas.

A partir dessas considerações os autores propõem um sistema para ambientes colaborativos de múltiplos usuários e múltiplas telas para ferramentas de visualização de informação (Figura 21). Os autores defendem que o projeto do sistema deve ser separado em funcionalidade básica do ambiente de múltiplas telas, tais como manipulação de múltipla entrada e a gestão das exposições físicas e funcionalidade de mais alto nível proporcionado pelo sistema de visualização (Waldner, 2009).



Figura 21. Fotos do ambiente proposto por Waldner (Waldner, 2009)

3.2.2 *Three Modes of Multi-Surface Interaction and Visualization*

Esse trabalho descreve uma taxonomia de estilos de visualização para ambientes múltiplas telas de múltiplas visões como: compartilhamento de conteúdo, visualização de dados, manipulação da interface do usuário entre múltiplas superfícies interativas (mesa, quadros, monitores, etc.). Além dessa taxonomia de estilos, os autores propõem três modos de múltiplas superfícies de visualização e interação: independente, reflexivo, e coordenado (Figura 22).



Figura 22. Fotos do ambiente proposto por Shen, Esenther, et al

3.2.3 *PointRight: Experience with Flexible Input Redirection in Interactive Workspaces*

Os autores descrevem a concepção e experiência com *PointRight*, um ponteiro *peer-to-peer* e sistema de redirecionamento de teclado que funciona em ambiente de múltiplas máquinas e múltiplos usuários (Figura 23). *PointRight* emprega um modelo geométrico para redirecionamento de entrada nas telas impulsionado por várias máquinas independentes e sistemas operacionais. Ela foi criada para *desktops* interativos que incluem grandes, displays compartilhados e laptops individuais, mas é uma ferramenta geral que suporta várias configurações diferentes e modos de utilização. Embora os sistemas anteriores previssem reencaminhamento de ponteiro e controle de teclado, neste trabalho, os autores apresentam um sistema mais geral e flexível, juntamente com uma análise dos tipos de religação que devem ser manuseados por qualquer sistema de redirecionamento de ponteiro. Este artigo descreve o sistema, as formas em que foi usado, e as lições que foram aprendidas a partir de seu uso ao longo dos últimos dois anos. (Johanson, 2002.)



Figura 23. Fotos do ambiente proposto por Johanson, Hutchins, et al

3.2.4 *Adapting a Single-User, Single-Display Molecular Visualization Application for Use in a Multi-User, Multi-Display Environment*

Este trabalho discutiu a adaptação de um aplicativo de visualização molecular *open-source* para um único usuário e única tela, para um sistema de ambiente de múltiplas telas e múltiplos usuários (Figura 24). *Jmol* uma aplicação *open-source* em *Java applet* para visualização de arquivos PDB que permite visões coordenadas da mesma molécula a serem exibidas em um espaço de múltiplas telas. Cada exibição no espaço de trabalho é conduzido por um PC separado, e visões coordenadas são realizados através da passagem de comandos de script *RasMol* através da rede. O ambiente de mesa inclui um visor capaz de detecção de toque de entrada, duas grandes exposições verticais, e um *TablePC*. A apresentação de moléculas de grandes dimensões está adaptada para tirar o melhor das diferentes qualidades de cada monitor, também é apresentado um conjunto de técnicas que permitem a interação de grupos de trabalho nesse ambiente para melhor colaborar. (Forlines & Lilien, 2008.)

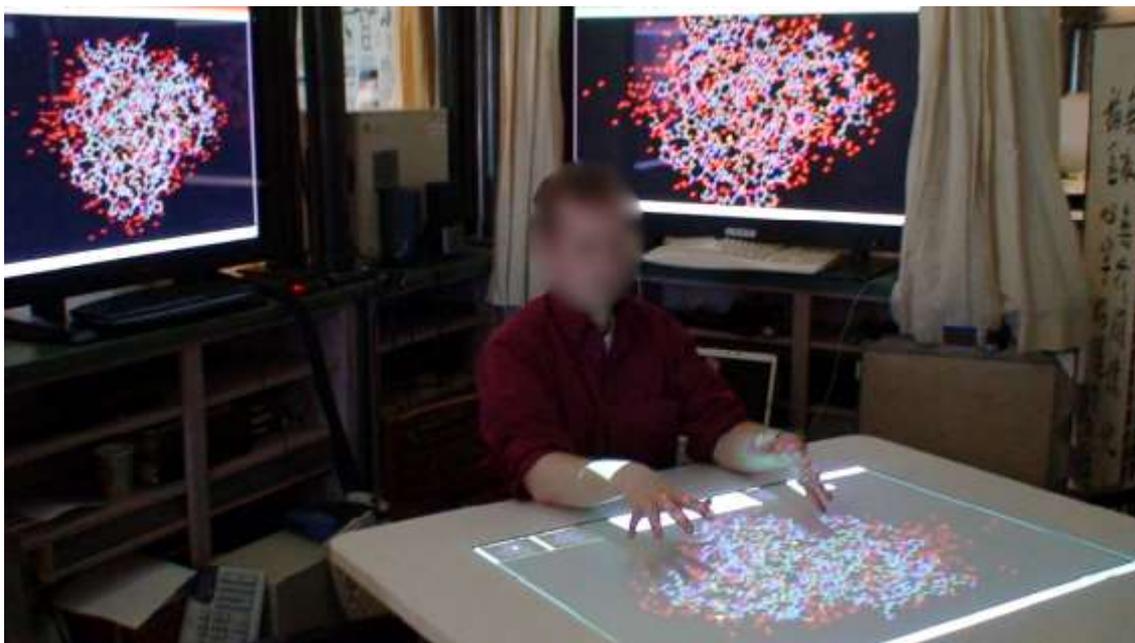


Figura 24. Fotos do ambiente proposto por Forlines e Lillien, et al

3.2.5 *Code Space: Touch + air Gesture Hybrid interactions for Supporting Developer Meetings.*

Nesse trabalho, os autores apresentam o “*Code Space*”, um sistema híbrido de toque e interações no ar (gesto) para apoiar reuniões de pequenos grupos de desenvolvedores, facilitando o acesso, controle e partilha de informação entre vários

dispositivos pessoais e exposições públicas (Figura 25). Esse sistema usa uma combinação de uma tela *multi-touch* compartilhada, dispositivos de toque móvel e sensores Microsoft Kinect. São apresentadas várias técnicas de interações, que usam uma combinação de gestos e toques. No estudo apresentado os desenvolvedores profissionais foram positivos sobre o design de interação.



Figura 25. Fotos do ambiente proposto por Bragdoni et al

3.2.6 *LibGlass*

A *libGlass* é uma biblioteca para computação distribuída e computação em nuvem, com foco em alta desempenho, aplicações de baixo *lag*, particularmente realidade virtual e aumentada. (Guimarães, Gnecco, Bressan, & Zuffo, 2003)

O *framework libGlass* é um conjunto escalável de componentes que podem ser usados por aplicativos para realizar computação distribuída. Aplicações são construídas reutilizando os componentes disponíveis conforme necessário. Um dos objetivos principais da *libGlass* é ser um *framework* de fácil utilização, não só adequado para novas aplicações, mas também para o código legado. Esta é uma característica importante, já que a maioria das soluções disponíveis para a computação distribuída

requer uma quantidade substancial de reescrita de código legado, e alguns deles exigem uma mudança completa do design do aplicativo. (Guimarães, Gnecco, Bressan, & Zuffo, 2003)

Os principais objetivos do *framework libGlass* serão citados abaixo:

- **Transparência:** a biblioteca deve ser o mais transparente possível. Todas as tarefas que são repetitivas ou que pode ser feito automaticamente devem ser feita pela biblioteca, sem intervenção do usuário. A API (*Application Programmer's Interface*) deve ser simples e intuitiva, com uma curva de aprendizado suave e fornece primitivas de alto nível que podem ser facilmente usados para resolver vários problemas.
- **Extensibilidade:** a biblioteca deve ser facilmente extensível, sem necessidade de recompilação ou qualquer outra modificação para suportar novos recursos, eles devem trabalhar como *plug-ins*.
- **Desempenho e eficiência:** uma vez que a biblioteca é para computação distribuída, tem que ser eficiente, consumindo pouco tempo de processamento possível.
- **Rede independência de protocolo:** uma camada de abstração torna possível alterar os protocolos de rede subjacentes facilmente. O aplicativo pode usar o protocolo de rede mais eficiente para as suas necessidades.
- **Escalabilidade:** não há uso de uma solução distribuída que não escala bem. O *framework libGlass* tem de trabalhar bem em todos os tipos de ambientes, desde pequenos grupos até grandes redes, e ser capaz de se adaptar para alcançar melhores resultados.
- **Arquitetura de rede reconfigurável:** soluções mais atuais são baseadas em uma arquitetura de rede fixa, normalmente mestre / escravo ou cliente / servidor. Dados os requisitos de escalabilidade e desempenho, e o fato de que *peer-to-peer* aplicações que estão cada dia mais comum, é razoável fixar a arquitetura de rede: o desenvolvedor deve ser livre para definir como irá se conectar com o outro.
- **Confiabilidade e tolerância a falhas:** computação distribuída geralmente exige confiabilidade. Não é aceitável que o mau comportamento ou falha de um único nó trava o sistema inteiro. O *framework libGlass* deve ser capaz de lidar com falhas dos nós, mantendo a aplicação em execução e evitar bloqueios e outros

problemas que podem surgir. Os nós devem ser autorizados a entrar ou sair a qualquer momento.

- **Suporte a thread:** a biblioteca deve estar completamente thread-safe. Com processadores multicore, é inaceitável para uma estrutura de computação distribuída a ter problemas de threads.

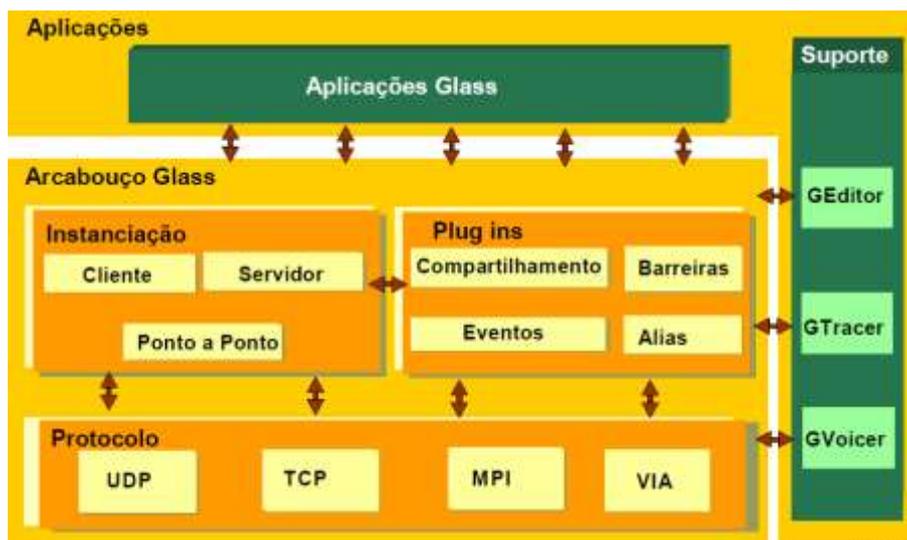


Figura 26. Arquitetura do framework libGlass (Guimarães P. , 2004)

A Figura 26 descreve a arquitetura do *framework libGlass* que é composto pelo arcabouço Glass e seus componentes instanciação, protocolo e plug-ins. O componente de instanciação permite inicializar as aplicações conforme a arquitetura da Glass (cliente ou servidor). O protocolo encapsula as bibliotecas escondendo as diferenças entre os protocolos de comunicação (TCP, UDP e outros). Os plug-ins são responsáveis por fornecer serviços de transmissão de eventos, compartilhamento de dados, de barreiras de dados e associação de funções (Guimarães P. , 2004).

4 PRISMA-MDE

O PRISMA - MDE (*multi-display environment*) é um ambiente distribuído de múltiplas telas combinado com a ferramenta de visualização da informação PRISMA, que melhora o problema da oclusão de dados e permite que o usuário interaja com varias instâncias da aplicação em paralelo. Cada instância é executada numa máquina própria, exibindo os resultados da técnica de visualização diferente.

Ao utilizar máquinas completas e não apenas conectar mais monitores no sistema é possível contar com os recursos da máquina para carregar e manipular bases de dados cada vez maiores, sem prejudicar a interação do usuário. O usuário ainda pode adicionar mais técnicas para cada instância, ou mais máquinas e mais técnicas.

4.1 Características Principais

A característica principal desse projeto é passar a impressão de um desktop estendido contínuo para usuário, com objetivo de facilitar a interação com mouse entre as diferentes telas passando a impressão de visão contínua. Assim, a organização dos monitores foi disposta lado a lado e o modo de interação escolhido foi redirecionamento de entrada (Figura 27). A Figura 27 exemplifica um ambiente de múltiplas telas com redirecionamento de entrada (mouse e teclado) e simulação de um desktop estendido contínuo.

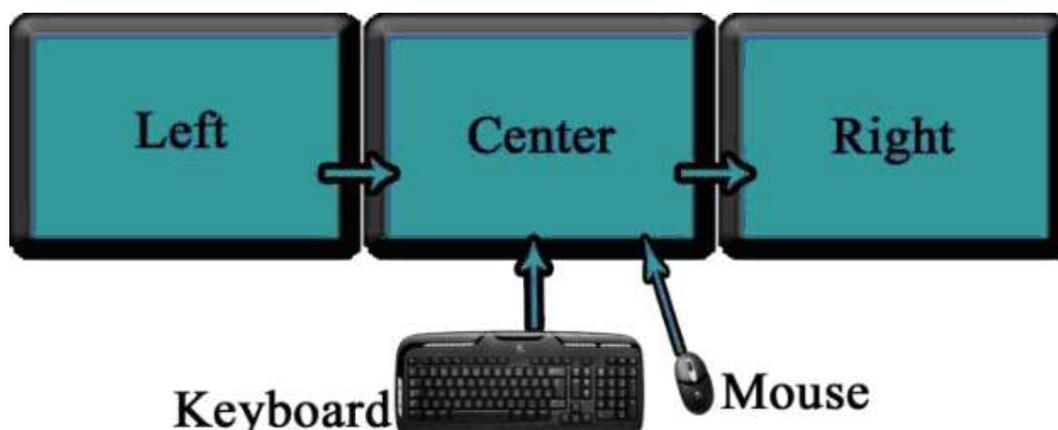


Figura 27: Exemplo do ambiente de múltiplas telas que simula um *desktop* estendido

Uma desvantagem dessa abordagem ocorre em ambiente com muitas telas, por exemplo, se for utilizado dez monitores e o usuário for acessar a última tela, sendo que ele está na primeira tela, o deslocamento do mouse será demorado, pois será necessário percorrer tela por tela até chegar à última tela.

Tanenbaum define sistema distribuído como um conjunto de computadores independentes que na visão do usuário é um sistema único e consistente. (Tanenbaum, 2007). Então, outra característica é um ambiente de processamento distribuído, onde cada máquina conectada ao ambiente processa cada instância da ferramenta PRISMA-MDE, apesar de possuir características de arquitetura cliente-servidor, onde o servidor concentra a lógica de negócios para gerenciamento do ambiente MDE e os clientes distribuem o processamento da ferramenta PRISMA-MDE, mantendo o servidor como um histórico de ações, utilizado caso algum cliente fique *off-line*.

4.2 Arquitetura

A arquitetura do PRISMA-MDE foi projetada baseada no padrão de projeto MVC (*model view control*), possuindo uma distribuição de responsabilidades entre os módulos bem definida. A representação da arquitetura foi dividida em duas partes: cliente e servidor, porém este último pode estar presente dentro de um dos clientes, sem prejuízos de qualidade para o sistema. Neste trabalho é chamado de “nó” qualquer cliente e de “nó mestre” o cliente que possui o servidor.

A arquitetura do cliente é composta por seis módulos principais: *GlassClient*, PRISMA e quatro módulos de controle (Figura 28), que tratam as ações do usuário no ambiente. O servidor possui cinco módulos principais: *GlassServer* e quatro módulos de gerenciamento (Figura 28), que mantem o histórico das ações do usuário e quando necessário aplicam regras de negócios de acordo com o tipo de ação que o módulo troca.

Os submódulos de gerenciamento e controle de comunicação, presente dentro do *GlassServer* e *GlassClient* respectivamente, possuem estruturas de controle de comunicação baseando no framework *libGlass*. Nas próximas subseções será apresentado cada módulo e descrição detalhada.

A Figura 28 apresenta uma representação gráfica da arquitetura para três nós (clientes) e um servidor, que é considerado como uma quarta máquina no sistema. Apresenta também, todos os submódulos presente dentro dos módulos. Esta

representação da arquitetura faz jus ao modelo adotado para o ambiente de teste que é apresentado na seção 5 de resultados.

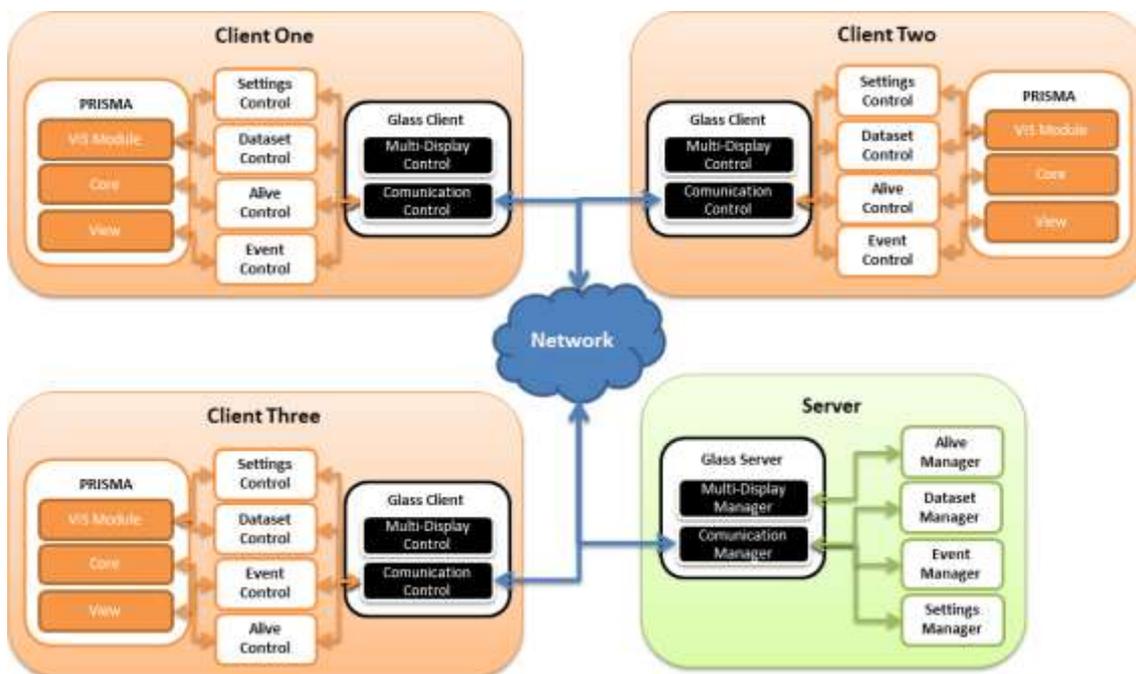


Figura 28: Principais módulos da arquitetura do PRISMA-MDE

4.2.1 Cliente

A denominação cliente ou “nó” é dado a uma máquina que executa uma instância completa da ferramenta PRISMA e que compõem o ambiente PRISMA-MDE. A seguir serão apresentados os módulos e cada submódulo presente nos clientes.

4.2.1.1 PRISMA

A base de desenvolvimento desse trabalho é a ferramenta de visualização de informação PRISMA (Godinho, et al., 2007). Esta ferramenta foi escolhida porque é uma ferramenta de *desktop* baseado em múltiplas visualizações coordenadas, que tem três técnicas principais: dispersão de dados (Spence R. , 2001), coordenadas paralelas e Treemap, a ferramenta foi desenvolvida totalmente em Java, dando total portabilidade aos sistemas que executam Java, além disso, possui recursos importantes para uma boa ferramenta de IV, como visão geral, detalhes sobre a demanda, seleção, filtros dinâmicos, zoom, configurações dos atributos, gráficos estatísticos, relatórios customizados, acesso a diversas fontes de dados, relacionamento entre dados, e histórico

de ações (Carr, 1999). A seguir serão apresentados os três principais módulos da ferramenta.

4.2.1.2 *MóduloVIS*

A principal funcionalidade deste módulo é manter informações de configuração e gerar as visualizações com base nestas configurações. Este módulo também faz o gerenciamento de uma fonte de dados (XML, CSV ou Banco de Dados SQL) para carga, seleção e organização dos registros, gerenciamentos dos filtros, cores, detalhes sob demanda e o gerenciamento das técnicas de visualização para geração de visões gráficas. Essas informações de configuração também são compartilhadas com o módulo de controle de configurações.

4.2.1.3 *Núcleo*

O núcleo é o módulo principal da ferramenta PRISMA-MDE, possui uma ligação com o módulo de apresentação, um gerador de MóduloVis, e um MóduloVis. É responsável por diversas atividades de controle, inicialização, encerramento da aplicação, gerenciamento do MóduloVis, inicialização do módulo de apresentação, execução de ações provenientes da interface com o usuário e entrelaçamento entre os componentes gráficos do módulo de Apresentação com os modelos contidos no MóduloVis. Outra função é trocar informações de status com o módulo de controle de status.

4.2.1.4 *Apresentação*

A Apresentação cria e gerencia os componentes de interface gráfica do usuário. Estes componentes são criados de acordo com a necessidade do usuário, por exemplo, quando o usuário adiciona uma nova fonte de dados ou técnica para a aplicação, esta requer a criação de novos painéis de configuração por exemplo. O módulo de Apresentação tem uma forte ligação com o Núcleo, sempre que a Apresentação cria uma nova visão (seja esta um painel de configuração ou um painel de desenho) é feita uma requisição ao Núcleo para entrelaçar a visão recém-criada com seu respectivo módulo.

Esta abordagem foi escolhida durante o desenvolvimento da ferramenta PRISMA (Godinho, et al., 2007) e segundo os autores visa trabalhos futuros com interfaces web, em que o módulo de Apresentação poderá se tornar um *WebService* provendo

informações e um caminho de comunicação para qualquer plataforma compatível, no caso do ambiente MDE, essa abordagem poderá ser aproveitada futuramente para desenvolvimento de ambientes colaborativos.

4.2.1.5 *Módulos de Controle*

Os módulos de controle tem a função genérica de identificar uma ação do usuário, transformar em mensagem e enviar para o módulo de comunicação. Assim como o processo inverso, recebem a mensagem do módulo de comunicação e transformam em uma ação. A seguir, será mostrado cada tipo de ações tratadas por cada módulo.

4.2.1.6 *Controle de Configurações*

O módulo controle de configuração é responsável por empacotar e desempacotar as mensagens e converter em ações de configurações das técnicas, como: configuração das cores, tamanho e rotulo dos itens, configuração dos eixos X e Y das técnicas coordenadas paralelas e dispersão de dados.

Um exemplo do uso deste módulo é quando um cliente X faz a mudança de cores de um conjunto de atributos, este módulo identifica a ação, interpreta e transforma em mensagem, que é encaminhada aos outros clientes pelos módulos de comunicação do servidor, quando a mensagem chega ao outro cliente esse módulo, que também está presente no outro cliente, recebe a mensagem e converte na ação de mudança de cores, executando automaticamente dentro do módulo do PRISMA.

4.2.1.7 *Controle de Banco de Dados*

O controle de banco de dados tem uma função principal de manter a base de dados atualizada, este módulo comunica com o gerenciador de banco de dados do servidor para receber a carga inicial da base, e caso a base tenha sido alterada fazer as atualizações, por exemplo, sempre que um usuário faz um filtro, este módulo converte a ação em uma *string* de consulta e envia para os módulos de comunicação, e quando este módulo recebe uma *string* de consulta ele converte em uma ação de filtro.

4.2.1.8 *Controle de Eventos*

Esse módulo tem a responsabilidade de sincronizar os eventos de entrada entre os dispositivos, este é responsável por receber as ações de entrada do servidor e aplicar na sua máquina. Somente o servidor poderá ter o mouse sendo redirecionado para os todos

os cliente, porém, com já comentado anteriormente, o servidor poderá ser hospedado dentro de um cliente fazendo deste um nó mestre.

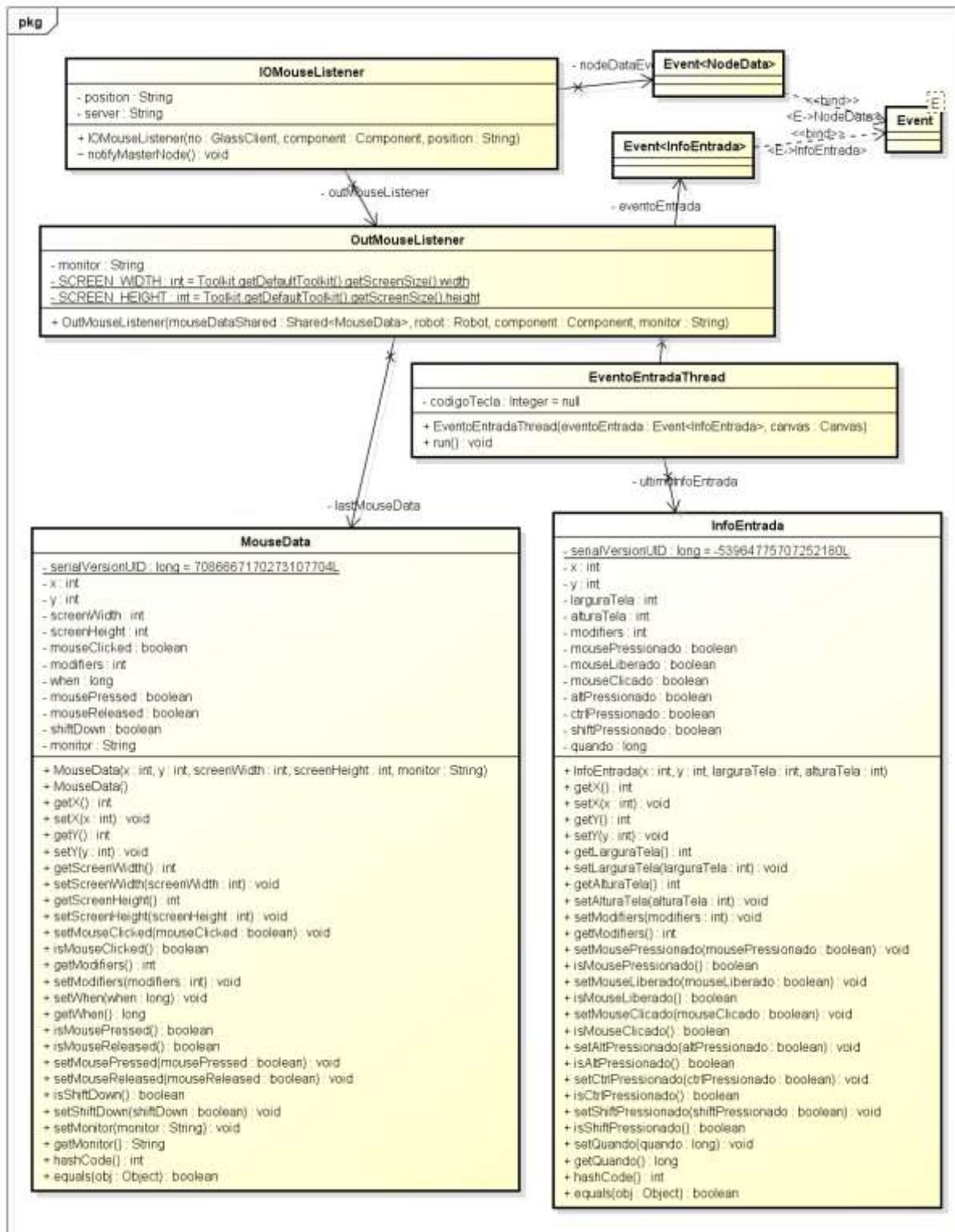


Figura 29. Diagrama de classes do módulo controle de eventos.

Atualmente esse módulo trabalha apenas com eventos de entrada de mouse e teclado, mas pode ser facilmente estendido para outros controles, como: dispositivos de interação por voz, gestos, *tablets*, e outros dispositivos de entrada não convencionais como controles de videogames e outros. Na página anterior foi apresentado um diagrama de classes que exemplifica o funcionamento do módulo de controle de eventos. A seguir será apresentada a descrição das principais classes.

A Figura 29 apresenta o diagrama de classe das principais classes do módulo de controle de eventos. O módulo de controle de eventos faz o sincronismo das ações de entrada do mouse. A classe *OutMouseListener* define *ActionLisneners* para processar informações de eventos de mouse enviados pelo módulo de comunicação. A classe *EventoEntradaThread* é responsável por aguardar e processa um *InfoEntrada* encaminhado pelos módulos de comunicação. Nessa classe *InfoEntrada* é definido e gerencia o domínio de informações das entradas do mouse.

4.2.1.9 Controle de Status

O módulo de controle de status é responsável informar ao servidor toda a vez que um cliente se conecta ou desconecta, também recebe do servidor as informações dos outros clientes e mantém uma tabela com o nome dos clientes ativos, para ser usado pelo módulo de comunicação para troca de informações. Este também, responde as mensagens do servidor de verificação de status, estas mensagens servem para saber se o servidor ficou off-line inesperadamente.

4.2.1.10 GlassClient

GlassClient é um conjunto de módulos responsável pelo controle do cliente e fazer a comunicação com o servidor. Esse conjunto foi desenvolvido a partir do *framework libGlass*, que uma API (*Application Programming Interface*) que auxilia o desenvolvimento de aplicações distribuídas, com foco em alta performance, lag baixos, e aplicações de realidade virtual e aumentada.

O framework Glass é um conjunto escalável de componentes que podem ser usados por aplicativos para realizar computação distribuída. Aplicações são construídas reutilizando os componentes disponíveis conforme necessário. Um dos objetivos principais da Glass é ser um framework de fácil utilização, não só adequado para novas aplicações, mas também para o código legado. Esta é uma característica importante, já que a maioria das soluções disponíveis para a computação distribuída requer uma

quantidade substancial de reescrita de código legado, e alguns deles exigem uma mudança completa do design do aplicativo.

Os módulos de controle de múltiplas telas e controle de comunicação fazem parte desse conjunto de módulos. A seguir serão mostradas as principais funções de cada um desses módulos.

4.2.1.11 Controle de Múltiplas Telas

Esse módulo é responsável por consultar os registros de todos os clientes do servidor e saber a posição das telas, por exemplo, vamos utilizar três máquinas, sendo uma na esquerda, uma na direita e uma no centro, neste caso a máquina central também irá hospedar o servidor. Nesse exemplo, o módulo de controle de múltiplas telas atua organizando as telas, ele registra qual a posição da máquina que está presente e faz as configurações de extensão da área de trabalhos para os clientes.

Na Figura 27, o servidor presente na tela central tem a informação que existem mais dois clientes um do lado direito e outro do lado esquerdo, quando o mouse é deslocado para direita, as ações de entrada devem ser enviados para a máquina do lado direito. Essa é a função desse módulo, saber qual a posição das máquinas vizinhas e a qual máquina se destina aquela ação.

4.2.1.12 Controle de Comunicação

A principal responsabilidade desse módulo é estabelecer a comunicação com o servidor e efetuar a troca de mensagens. Para isso utiliza-se uma rede de computadores com a pilha de protocolos TCP/IP, onde a comunicação é estabelecida a partir do protocolo TCP, este foi escolhido, pois é um protocolo para conexões confiáveis, que permite a entrega sem erro de fluxo de bytes.

4.2.2 Servidor

O servidor é uma máquina que armazena, processa e encaminha todas as mensagens trocadas entre os módulos do sistema, exceto o módulo de controle de evento, neste é mantido somente a última mensagem. Também pode ser chamado de “nó mestre”, porém, neste caso, além do servidor será executada uma instância do cliente.

4.2.2.1 GlassServer

Este, assim como o *GlassClient* também é um conjunto de módulos desenvolvidos baseados no *framework libGlass*. Os módulos: gerenciador de múltiplas telas e

gerenciador de comunicação compõe esse conjunto. Sua principal função é receber e enviar as mensagens enviadas pelos clientes e aplicar algumas regras específicas de cada módulo do servidor caso necessário.

4.2.2.2 Gerenciador de Múltiplas Telas

O gerenciador de múltiplas telas recebe as informações dos dispositivos que estão se conectados ao servidor e mantém essas informações para que os clientes possam atualizar no módulo de controle de múltiplas telas, para isso, este se comunica com módulo gerenciador de status para saber quais os clientes ainda estão ativos. Outra função importante desse módulo é fornecer a carga inicial das configurações, por exemplo, o usuário está interagindo com apenas duas telas, e já possui diversas configurações realizadas como: cores, filtros, e outros, quando o usuário incluir a terceira máquina esta já iniciará com as configurações feitas.

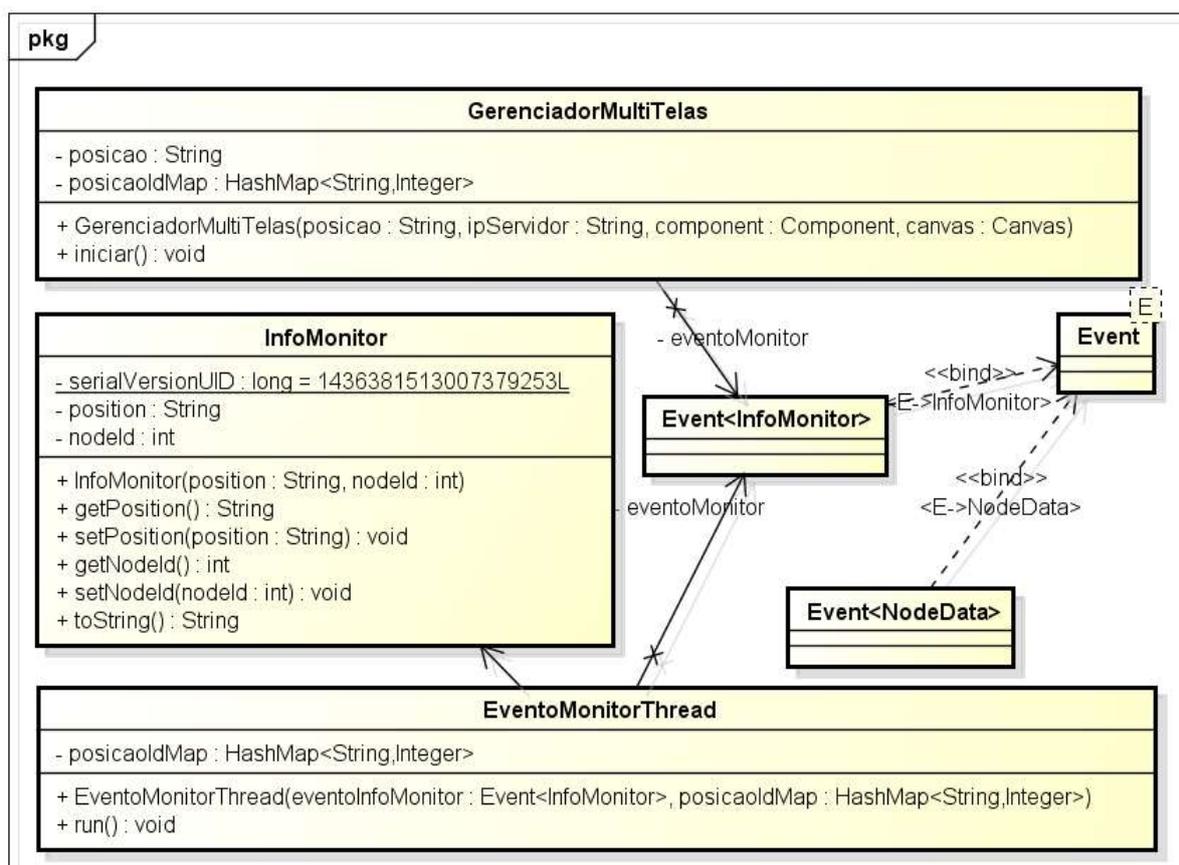


Figura 30. Diagrama de classe do módulo gerenciador de múltiplas telas

A Figura 30 é um diagrama das principais classes utilizadas no módulo gerenciador de múltiplas telas, que tem como função principal administrar as telas do ambiente.

A classe *GerenciadorMultiplasTelas* coordena a organização das máquinas que executam as visualizações, através do recebimento dos *InfoMonitor* através da classe *EventoThreadMonitor*, que por sua vez a classe *EventoMonitorThread* aguarda e processa um *InfoMonitor*, que é uma classe responsável por definir e gerencia o domínio de informações das telas utilizadas no ambiente.

4.2.2.3 *Gerenciador de Comunicação*

O módulo de gerenciamento de comunicação, assim como o controle tem comunicação tem a função de efetuar a troca de mensagem entre o servidor e os clientes, através do protocolo TCP. Outra função é encaminhar para algum dos módulos de gerenciamento antes de encaminhar a mensagem ao destinatário, ou seja, quando este recebe uma mensagem de configuração, essa é encaminhada para o módulo de gerenciamento de configurações que aplica algumas regras se necessário e encaminha de volta para este módulo, que por sua vez, reenvia para os clientes.

4.2.2.4 *Módulos de Gerenciamento*

O conjunto de módulos de gerenciamento tem como principal função manter a última mensagem, ou um histórico das mensagens que foram trocadas. Também, este aplica algumas regras caso seja necessário.

4.2.2.5 *Gerenciador de Configurações*

O módulo de gerenciamento mantém o histórico das mensagens recebidas e aplica regras para sincronização entre as configurações. Para reduzir o tráfego na rede, os filtros são tratados como configurações, pois a mensagem não inclui trecho da base de dados, apenas a consulta que é gerada a redução.

4.2.2.6 *Gerenciador de Banco de Dados*

Este é responsável por dar a carga inicial da base de dados, e replicar para os clientes, sendo feita atualizações na base dos clientes, quando a base desse módulo sofrer alterações. Este só mantém a última mensagem armazenada, devido a serem transferido como consultas.

4.2.2.7 *Gerenciador de eventos*

Nessa camada, também só é armazenada a última mensagem, pois não há necessidade de desfazer as ações de entrada. Está é responsável por controlar os eventos de entradas e saída. Atualmente esse módulo interpreta apenas ações de entrada e saídas

de mouse e teclado, mas poderia ser facilmente estendido para outros tipos de interações. A captura de eventos é sempre feito do servidor ou “nó mestre” e encaminhada para o módulo de comunicação que envia para o cliente destino.

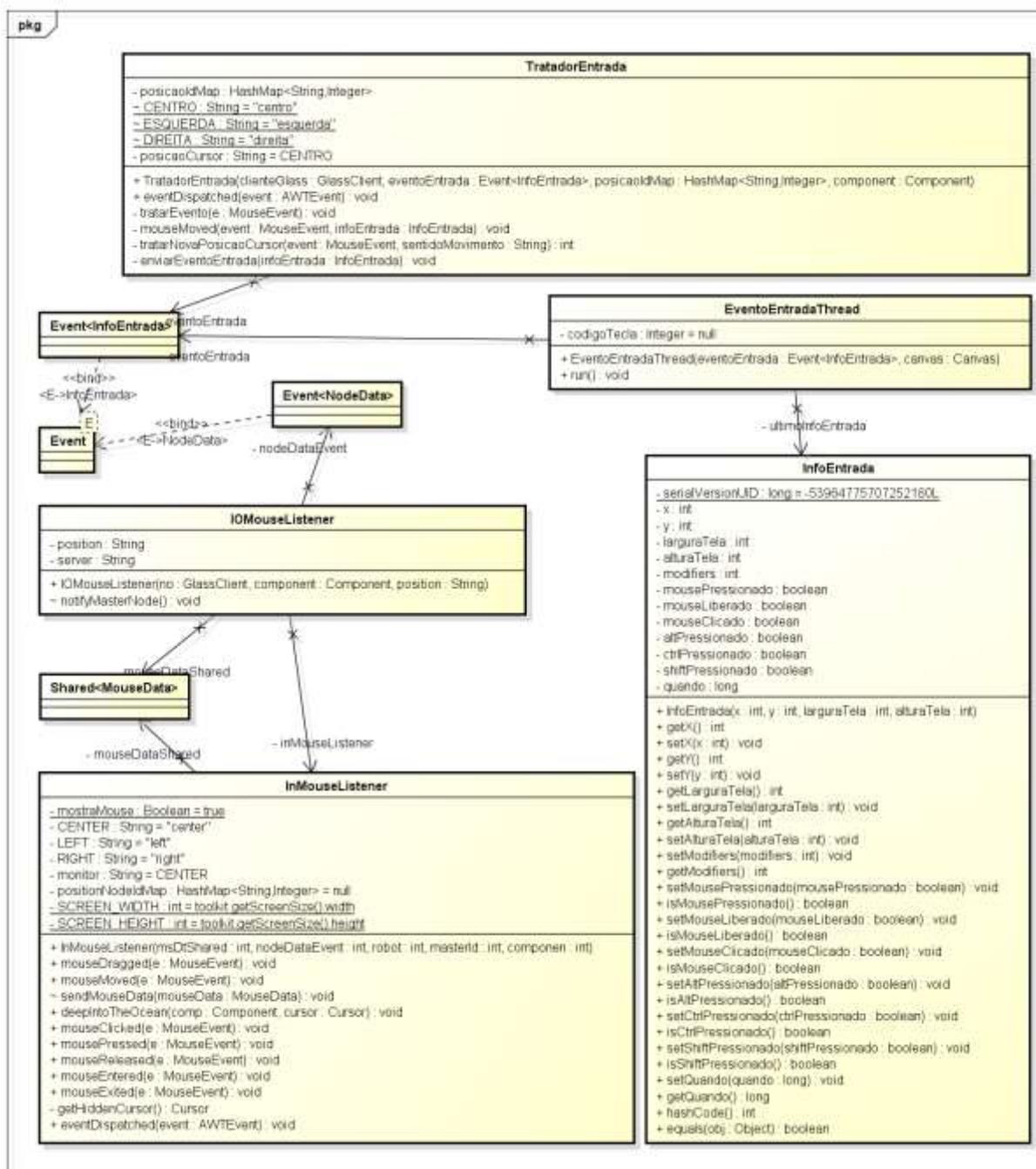


Figura 31. Diagrama de classes do módulo de gerenciamento de evento.

A Figura 31, apresenta um diagrama de classes das principais classes utilizadas pelo módulo de gerenciamento de eventos. As três principais classes deste módulo são *inMouseListener*, que é responsável por processar eventos de mouse gerados na máquina de origem (a que recebe entradas de mouse), depois a classe *tratadorEntrada*

aplicar umas regras para definir a tela e encaminhar para o módulo de comunicação que fica responsável por fazer a entrega da mensagem ao cliente.

Outra classe importante é a classe *eventoEntradaThread* que é responsável por aguardar e processar um *InfoEntrada*, que é uma classe que armazena as informações da interação do usuário, no caso do exemplo do diagrama do mouse.

4.2.2.8 Gerenciador de Status

O gerenciador de status é responsável por receber e armazenar as mensagens do módulo de controle de status, que informa quando o cliente conecta e desconecta, além disso, este módulo testa a comunicação em um período de 60 segundos para verificar se a máquina cliente está disponível.

4.3 Protótipo e Funcionalidades

O ambiente PRISMA-MDE conta com as três principais técnicas de visualização de informações: Treemap, Coordenadas paralelas e Dispersão de dados, outras técnicas podem ser facilmente programadas e integradas à ferramenta PRISMA a partir das interfaces pré-definidas (Godinho, et al., 2007).

O ambiente possui seleção e coordenação de dados entre as múltiplas telas como exemplificado na Figura 32 abaixo, essas funções facilitam a cognição a partir do destaque da mesma informação em diferentes telas e técnicas de visualização.

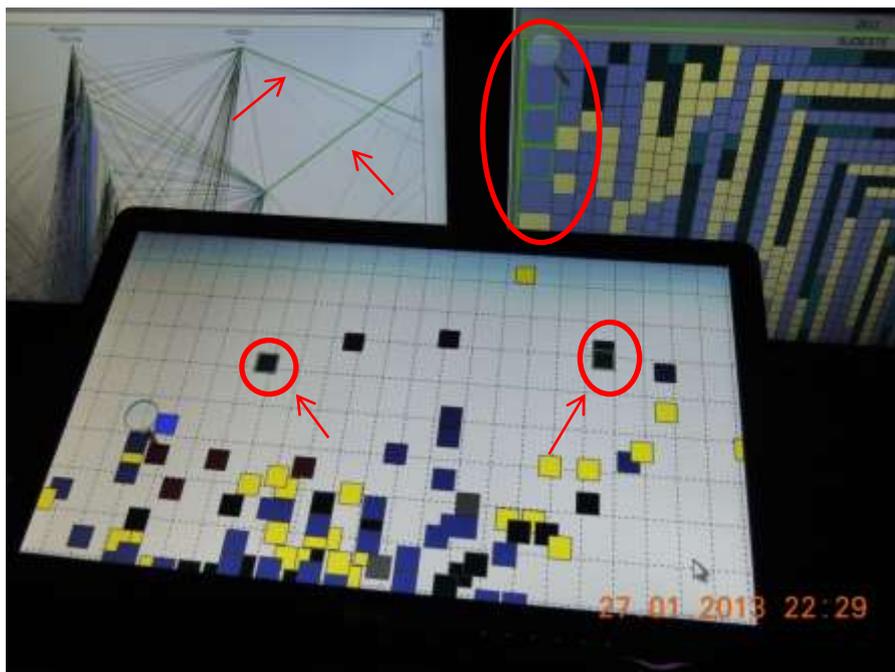


Figura 32. Exemplo de seleção e coordenação de dados

O ambiente PRISMA-MDE, também, suporta os principais recursos considerados imprescindíveis pela comunidade de visualização em uma ferramenta visualização de informação:

- Visão geral: Representação gráfica de todos os dados, e apresenta a dimensão completa do conjunto a ser analisado.
- Zoom: Um foco gráfico e específico em subconjunto de dados apresentados visualmente ao usuário.
- Filtros: Redução do conjunto de dados, diferentemente do zoom os filtros reduzem a base de dados e não apenas aplicam um foco em uma região específica.
- Detalhes sob demanda: informações adicionais de um item do conjunto de dados.
- Histórico de ações: mantém as temporariamente as ações do usuário armazenadas, dando opção de desfazer (Carr, 1999).
- Ordenação: classificação do conjunto de dados em função de um atributo específico.
- Mapeamento visual de atributos (Pillat & Freitas, 2006).

A Figura 33, mostra o protótipo do ambiente em execução, com três monitores e três computadores.



Figura 33. Execução do PRISMA-MDE no Ambiente de múltiplas telas

Outro conceito empregado no ambiente PRISMA-MDE é o conceito da escalabilidade. Bondi (Bondi, 2000) define escalabilidade como uma característica desejável em todos os sistemas que demonstra a habilidade de manipular uma porção crescente de trabalho de forma uniforme, ou seja, estar preparado para crescer. Diante disso, o ambiente PRISMA-MDE foi projetado para suportar a entrada de mais máquinas. Não houve um estudo mais aprofundado de quantas máquinas o PRISMA-MDE poderia utilizar, mas pode-se observar nas figuras 32, 33, 34 e 35 que é possível fazer esse crescimento.

A Figura 34 mostra o ambiente PRISMA-MDE com apenas duas máquinas e dois monitores, estendendo uma única técnica para os dois monitores, já a Figura 33 mostra o ambiente com três máquinas e três monitores, ambiente este usado para execução de todos os testes de usabilidade e desempenho, e por último Figura 35 mostra o PRISMA-MDE com quatro máquinas e quatro monitores, em todos os ambientes só foi utilizado um mouse e um teclado ligado ao nó mestre.

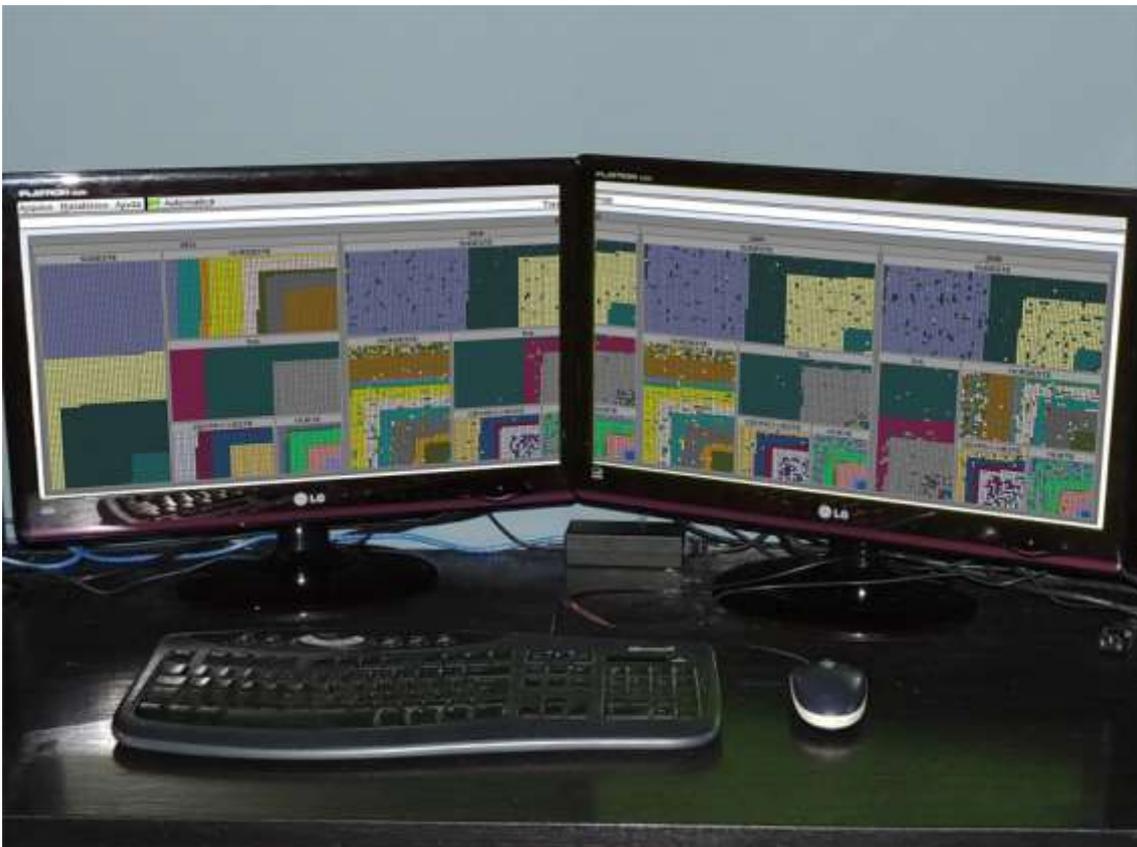


Figura 34. Ambiente PRISMA-MDE com duas máquinas estendido uma técnica de visualização



Figura 35. Ambiente PRISMA-MDE com quatro máquinas estendido uma técnica de visualização

5 RESULTADOS

Este capítulo tem como objetivo descrever o processo de avaliação do ambiente de múltiplas telas PRISMA-MDE, para isso foi proposto duas formas de avaliação: a primeira por testes de usabilidade fazendo um estudo de design do ambiente, e por se tratar de um ambiente distribuído foi proposta uma avaliação de desempenho do ambiente como segunda forma.

A seguir serão apresentadas as avaliações e a metodologia aplicada, as bases utilizadas e os resultados encontrados em cada avaliação.

5.1 Avaliação de Usabilidade

Uma das metas principais do ambiente PRISMA-MDE é melhorar o problema da oclusão de dados, através da utilização de um ambiente distribuído e escalável de múltiplas telas, para isso, acredita-se que o ambiente possa obter resultados melhores quando comparado apenas com a ferramenta PRISMA. A avaliação de usabilidade é utilizada com um método de gerar dados quantitativos para análise, além de avaliar outras informações, que serão demonstradas mais a frente.

Os testes com usuários têm por objetivo avaliar a usabilidade através da experiência de uso dos usuários que utilizarão o sistema (Rubin, 1994), outro objetivo é verificar a facilidade que o software possui de ser claramente compreendido. A metodologia dessa avaliação é baseada na coleta de dados quantitativos, e também na observação do comportamento dos usuários, observando suas reações quanto à aspectos do sistema, especialmente problemas e dificuldades que ocorrer em um cenário real (Barbosa & Silva, 2010).

5.1.1 Hipóteses

No trabalho escrito por (Rocha & Baranauskas, 2003) é destacada a importância de definir bem os objetivos antes da definição do teste, uma vez que estes devem ser

realizados de forma a ressaltar os critérios de usabilidade a serem medidos, para isso neste trabalho vamos aplicar a metodologia proposta por (Goodall, 2009), que sugere a criação de hipóteses sobre a acurácia (precisão, quantidade de acertos das tarefas propostas), eficiência (ganho de tempo nas soluções das tarefas, a partir da redução do esforço físico, mental e cognitivo), exploração (melhor desempenho, medido a partir do número de insights descobertos) e percepção do usuário (classificação de satisfação do usuário). A seguir serão apresentadas as hipóteses geradas para cada um dos cinco itens apresentados.

5.1.1.1 Acurácia:

A acurácia afere a precisão de solução das tarefas propostas para o usuário. Essa precisão é medida a partir da quantidade de acertos e erros que os usuários têm nas tarefas dos testes de usabilidade. A vantagem de se aferir a acurácia é que as ferramentas que propiciam um maior número de acertos possuem grandes chances de serem mais fáceis de interagir.

- **Hipótese 1:** O ambiente PRISMA-MDE resultara em menos erros em relação à ferramenta PRISMA.
- **Hipótese 1a:** A vantagem de menos erros usando o ambiente PRISMA-MDE será mais frequente em tarefas de comparação entre as técnicas em relação à ferramenta PRISMA.
- **Hipótese 1b:** Não haverá uma significativa vantagem de menos erros usando o ambiente PRISMA-MDE em tarefas de identificação, em comparação com a ferramenta PRISMA.

É esperado que os participantes fossem mais precisos e terem menos erros em todas as tarefas usando o ambiente PRISMA-MDE do que a ferramenta PRISMA. A hipótese 1 avalia o ambiente de maneira geral, enquanto as hipóteses 1a e 1b estão avaliando as tarefas de interação entre a ferramenta e o ambiente. A acurácia deverá ser mais acentuada para as tarefas de comparação, devido à existência de uma tela para cada técnica, a competição por espaço será menor.

5.1.1.2 Eficiência:

A eficiência mede o ganho de tempo nas resoluções das tarefas propostas aos participantes do teste. Para que ocorra o ganho de tempo à ferramenta tem que ser capaz

de fazer uma redução do esforço físico, mental e cognitivo do usuário. A partir disto, foram propostas as hipóteses a seguir:

- **Hipótese 2:** No ambiente PRISMA-MDE o usuário irá gastar menos tempo para a conclusão das tarefas, em comparação com a ferramenta PRISMA.
- **Hipótese 2a:** A vantagem de gastar menos tempo na conclusão das tarefas será observando no ambiente PRISMA-MDE para as tarefas de comparação em relação à ferramenta PRISMA.
- **Hipótese 2b:** Não haverá significativa vantagem de gastar menos tempo usando o ambiente PRISMA-MDE em tarefas de identificação e conclusão em relação à ferramenta PRISMA.

De maneira geral é esperado que os participantes gastassem menos tempo para a resolução de todas as tarefas propostas usando o ambiente PRISMA-MDE do que a ferramenta PRISMA, como foi descrito pela hipótese 2. As hipóteses 2a e 2b estão avaliando as tarefas de interação entre a ferramenta e o ambiente. A eficiência assim como a acurácia deveram ser mais acentuadas para as tarefas de comparação, devido à competição por espaço menor, ou seja, possui mais espaço para a renderização das técnicas de visualização.

5.1.1.3 Exploração:

A exploração mede o melhor desempenho, aferido a partir do número de insights descobertos, ou seja, o maior número de ideias, percepções que usuário poderá ter durante a execução das tarefas propostas. A seguir será descrita a hipótese criada baseada nas tarefas de exploração.

- **Hipótese 3:** O ambiente PRISMA-MDE resultará em um maior número de insights durante a exploração de dados, em comparação com a ferramenta PRISMA.

Durante as tarefas exploratórias é esperado que os participantes tivessem um melhor desempenho, utilizando o ambiente PRISMA-MDE quando comparado à ferramenta PRISMA. Isto se deve ao fato do ambiente possuir melhores condições para exploração visual de dados.

5.1.1.4 *Percepção do usuário:*

A percepção do usuário é uma classificação de satisfação do usuário, ou seja, ao final das tarefas o usuário foi convidado a responder um questionário onde ele iria expressar a sua satisfação ou insatisfação ao utilizar o ambiente PRISMA-MDE.

- **Hipótese 4:** A percepção do ambiente PRISMA-MDE será mais positiva quando comparada a ferramenta PRISMA.

De maneira geral é esperado que o ambiente PRISMA-MDE produza um grau de satisfação maior para o usuário, devido à existência de mais recursos e ser um ambiente que propicia a exploração visual de grandes bases de dados, com uma melhor eficiência e acurácia, quando comparado à ferramenta PRISMA.

5.1.2 **Base de dados**

Uma das principais dificuldades desse trabalho foi encontrar uma base de dados de domínio público com características sólidas e com um tamanho considerável para que fosse possível fazer as análises. A base de dados escolhida foi uma base multidimensional com dados reais, os dados apresentados são de todas as bolsas de pós-graduação pagas no Brasil nos anos de 2008 a 2011. Esta base foi escolhida por ser uma base extensa com dados reais e consistentes, sendo de fácil compreensão para todos os usuários que irão executar os testes e não necessitando de nenhum conhecimento adicional para compreendê-la.

Esta base possui 27.552 registros e dezoito atributos, sendo treze qualitativos e cinco quantitativos que são: ano, unidade de federação (UF), nome do município, região, código do programa, programa de fomento, nome da instituição de ensino superior, status jurídico, nome do programa, área de avaliação, área de conhecimento, grande área de conhecimento, quantidade de bolsas de mestrado, doutorado, mestrado profissional, pós-doutorado, professor visitante nacional sênior e total de bolsas.

5.1.3 **Tarefas**

As tarefas foram agrupadas em três grupos: tarefas de baixa complexidade (requer uma resposta simples, como um “sim” ou “não”), tarefa de média complexidade (não requer a análise de vários atributos, no máximo dois) e tarefas de alta complexidade (requer uma resposta mais elaborada e a análise de três ou mais atributos) (Maciel, Meiguins, Loureno, Meiguins, & Godinho, 2008).

- **Tarefas de identificação e baixa complexidade:**

1. Tarefa 01: O maior número de programas de pós-graduação que receberam bolsas no Brasil foi no ano de 2010?

Respostas: Não, 2011.

- **Tarefas de comparação e média complexidade:**

2. Tarefa 01: Houve um aumento na quantidade de programas de pós-graduação que receberam bolsas da CAPES do ano de 2010 para 2011 no estado do Rio de Janeiro na grande área de estudo exatas e ciências da terra? Se sim qual foi o aumento?

Respostas: Sim, três programas.

3. Tarefa 02: Comparando as grande áreas de estudos ciências exatas e da terra com ciências da saúde, qual foi o estado que teve mais programas de pós-graduação recebendo bolsas da CAPES no período de 2008 à 2011.

Respostas: São Paulo

- **Tarefas de identificação e alta complexidade:**

4. Tarefa 01: Qual a grande área de estudos e qual o estado que tiveram mais programas de pós-graduação que receberam bolsas da CAPES no ano 2011.

Respostas: Estado: São Paulo, Grande área de estudos: Ciências da saúde.

5.1.4 Metodologia de Aplicação

A aplicação dos testes de usabilidade ocorreu em um ambiente com três computadores formato torre, com oito giga bytes de memória RAM, processador Intel core i7 e três monitores de 20 polegadas, cada monitor ligado a uma torre. O servidor da aplicação foi colocado no cliente central, fazendo assim, “um nó mestre”, sendo apenas utilizado o mouse e o teclado do cliente mestre. A Figura 36 apresenta esse ambiente descrito. Na ferramenta PRISMA foi utilizada apenas uma das máquinas que compunham o ambiente PRISMA-MDE.

Foi escolhido um grupo de dez pessoas com experiência em computação e visualização de informação. Esse grupo foi dividido aleatoriamente em dois grupos de cinco pessoas. Um grupo para executar as tarefas na ferramenta PRISMA e outro grupo para executar as tarefas no ambiente PRISMA-MDE.

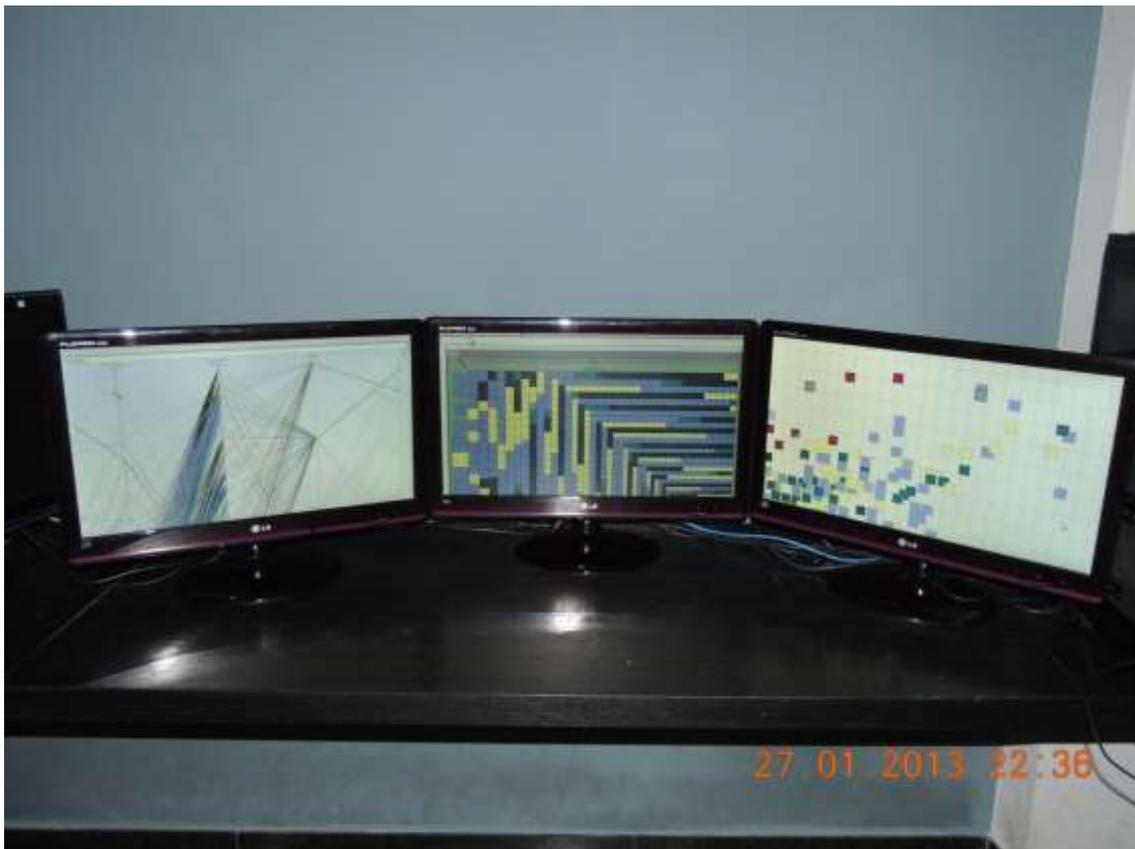


Figura 36 Foto foi ambiente de teste utilizado pelos usuários

O início dos testes ocorreu com uma breve apresentação da aplicação para os usuários com duração média de cinco minutos, após a apresentação inicial, os avaliadores convidaram os usuários a interagir com a ferramenta por mais cinco minutos, onde poderiam tirar dúvidas e fazer algumas tarefas que não compoñham o teste, mas possibilitava uma noção maior das funcionalidades do ambiente.

Depois da apresentação geral das funcionalidades do ambiente PRISMA-MDE, os usuários passaram pelo primeiro questionário, que visava comprovar ou não as hipóteses apresentadas na seção 5.1.1. Esse questionário possui as tarefas apresentadas na seção 5.1.3 e está presente no anexo 01 com o tempo máximo de solução de cinco minutos para cada. Após a resolução das tarefas no primeiro questionário foi passado um formulário adicional com 10 perguntas objetivas para avaliar o grau de satisfação do usuário ao interagir com a ferramenta.

Por último, os usuários foram convidados a responder a avaliação NASA Task Load Index (NASA TLX) que é uma escala multidimensional projetado para obtenção de estimativas de carga de trabalho dos testes de usabilidade de um ou mais operadores,

enquanto eles estão realizando uma tarefa ou imediatamente depois (Hart & Stavenland, 1988).

Carga de trabalho é um termo que representa o custo de realizar exigências da tarefa para o operador humano. Se as pessoas pudessem realizar tudo o que se espera que façam com rapidez, precisão e confiabilidade utilizando os recursos disponíveis, o conceito teria pouca importância prática (Hart & Stavenland, 1988).

5.1.5 Resultados

Para facilitar a análise dos gráficos será apresentada uma tabela com o número da tarefa, o tipo e o grau de complexidade (Tabela 1).

Tabela 1: Classificação das tarefas por tipo e complexidade

Tarefa	Tipo	Complexidade
01	Identificação	Baixa
02	Comparação	Média
03	Comparação	Média
04	Identificação	Alta

Fonte: Elaboração própria. Apresenta a classificação das tarefas por tipo e complexidade de acordo como definido no item 5.1.3.

Percebe-se que o grupo que resolveu as tarefas utilizando o ambiente PRISMA-MDE teve uma quantidade de acertos maior do que o grupo que utilizou a ferramenta PRISMA. De maneira geral, houve um acerto de 18% a mais no ambiente PRISMA-MDE, esse acerto foi ainda mais acentuado nas tarefas de comparação. Isso confirma as hipóteses 1, 1a e 1b, que supõem que o PRISMA-MDE propiciaria menos erros, principalmente em tarefas de comparação, o que pode ser observado na Figura 37 com o gráfico das médias de acerto por tarefa.



Figura 37. Média de acertos por tarefa

Em geral os usuários que utilizaram o ambiente PRISMA-MDE foram 22% mais rápido que os usuários que utilizaram a ferramenta PRISMA, comprovando assim as hipóteses 2, 2a e 2b, que trata da eficiência, ou seja, os usuários levariam menos tempo para a resolução das tarefas, principalmente para as tarefas de comparação. A ferramenta PRISMA se mostrou mais ágil em uma tarefa de identificação de complexidade fácil, pois para resolução da tarefa não seria necessária a utilização de mais de uma técnica e inclui a análise de um único atributo.



Figura 38. Média de Tempo para resolução da tarefa

De maneira geral a carga de trabalho para a ferramenta PRISMA foi mais alta do que a carga de trabalho para o ambiente PRISMA-MDE. Esses valores são calculados a partir de uma escala marcada pelo usuário imediatamente após a execução do teste, proposta no teste NASA TLX (Hart & Staveland, 1988). Os usuários do ambiente PRISMA-MDE se sentiram menos frustrados e tiveram uma demanda física e mental menor do que os usuários da ferramenta PRISMA, como pode ser observado na Figura 39. Outro item importante que vale destacar é que os usuários do PRISMA acharam que seu desempenho foi superior ao que realmente obtiveram. Isso se deve em função da resolução de algumas tarefas com algumas técnicas de visualização que propiciam oclusão de dados, passando uma ideia errada ao usuário quanto o seu desempenho, porém essa questão necessita de um estudo mais aprofundado.

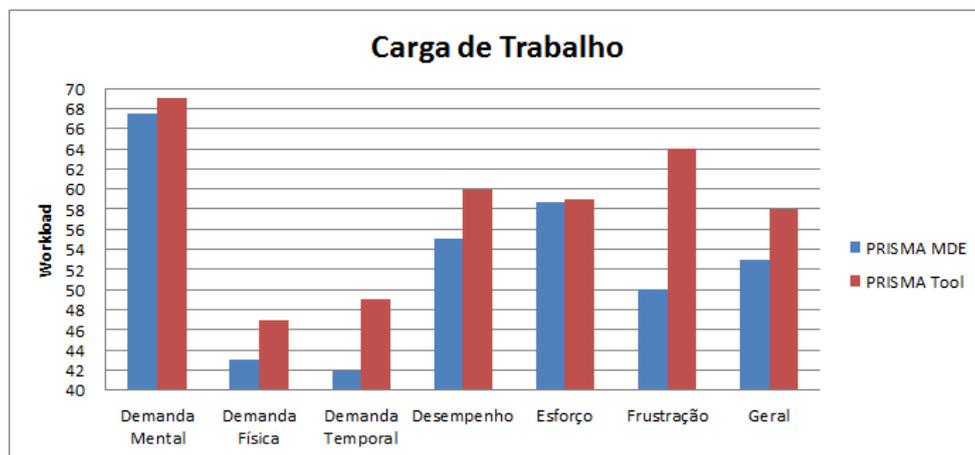


Figura 39. Carga de trabalho por área de avaliação

A metade dos usuários do ambiente PRISMA-MDE classificou como difícil à ação de configuração, por outro lado, mais 60% dos usuários da ferramenta PRISMA classificaram como difícil às ações de configuração. A ação de filtrar foi considerada difícil por 60% dos usuários do PRISMA-MDE e 35% dos usuários da ferramenta PRISMA, esses dados podem ser observado na Figura 40. Percebe-se que tanto a ferramenta PRISMA quanto o ambiente PRISMA-MDE precisam de uma revisão na sua interface a fim de melhorar a usabilidade.

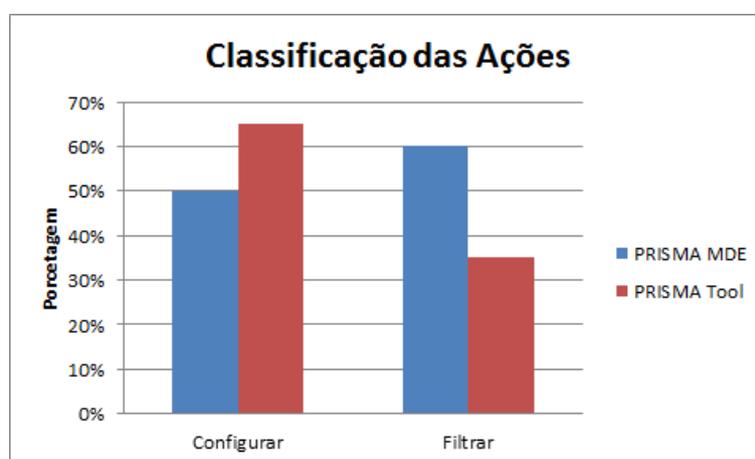


Figura 40. Classificação das ações de configurar e filtrar

Outra informação observada é que somente 60% dos usuários do ambiente PRISMA-MDE e 40% dos usuários da ferramenta PRISMA utilizaram mais de uma técnica de visualização de informação para resolver a mesma tarefa.

Outra questão observada foi como os usuários se sentiram utilizando três técnicas em uma única tela e três telas cada uma com uma técnica. No ambiente PRISMA-MDE

55% dos usuários se sentiram confortáveis em utilizar as três telas com uma técnica cada e 45% dos usuários da ferramenta PRISMA se sentiram confortáveis em utilizar três técnicas em três telas, estes dados estão representados graficamente na Figura 41.

O mouse também foi avaliado pelos usuários, onde 55% dos usuários do PRISMA-MDE se sentiram confortáveis em utilizar o mouse simulando um desktop estendido. Na ferramenta PRISMA só 20% dos usuários se sentiram confortáveis em utilizar o mouse para deslocar a telas através de barras de rolagem e visualizar todas as técnicas de visualização.

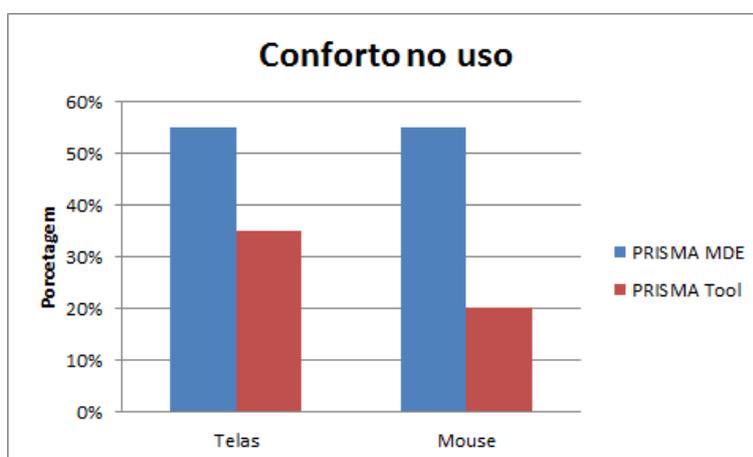


Figura 41. Usuários que se sentiram confortáveis ao utilizar as telas e mouse

5.2 Avaliação de Desempenho

Normalmente as avaliações de desempenho fazem ou buscam fazer um teste de carga que encontre os limites de processamento de dados do software testado, sendo avaliado em determinados cenários e configurações. Esse trabalho não busca encontrar o limite de processamento da ferramenta PRISMA ou do ambiente PRISMA-MDE, mas sim, efetuar uma comparação entre as aplicações de visualização de informação.

Para isso, foi empregada a ideia da metodologia dos testes de usabilidade adaptada, que busca comprovar hipóteses levantadas durante o desenvolvimento do ambiente, e que é esperado que o ambiente PRISMA-MDE seja capaz de executar com qualidade.

A avaliação de desempenho proposta nesse trabalho tem como objetivo mensurar o desempenho do ambiente PRISMA-MDE e da ferramenta PRISMA através de três variáveis o tempo de processamento, o consumo de memória e ocupação do

processador. Nesse trabalho, o tempo de processamento é considerado a partir do momento que usuário termina de ordenar a ação para o computador até o momento em que o computador exhibe o resultado da ação, por exemplo, o usuário deseja carregar três técnicas de visualização da informação em três monitores diferentes com o ambiente PRISMA-MDE. O tempo irá iniciar a contagem após o usuário ter selecionado as técnicas e quando ele soltar o botão do mouse (*MouseButtonRelease*) e só irá parar de contar quando o computador concluir a renderização das técnicas em todas as máquinas.

5.2.1 Hipóteses

Ao longo do planejamento e desenvolvimento do ambiente PRISMA-MDE, foram criadas algumas expectativas que o ambiente poderia ter características de processamento superiores as já encontradas na ferramenta PRISMA. Todas essas possibilidades foram convertidas em três hipóteses que descreve o desempenho do ambiente PRISMA-MDE. Estas serão descritas abaixo:

- **Hipótese 1:** O tempo de renderização da visão geral será menor no ambiente PRISMA-MDE quando comparada a ferramenta PRISMA.
- **Hipótese 2:** O consumo de CPU da aplicação e renderização de uma consulta de filtro será menor no ambiente PRISMA-MDE comparada a ferramenta PRISMA.
- **Hipótese 3:** A diferença do consumo de memória da carga inicial dos dados não terá relevância significativa entre o ambiente PRISMA-MDE e a ferramenta PRISMA.

A Hipótese 1 faz referência a carga inicial dos dados, desde o momento em que o usuário termina a escolha das técnicas de visualização até o momento que a máquina termina de renderizar as imagens na tela.

A Hipótese 2 faz referência a aplicação da *string* de consulta e renderização de uma consulta de filtro, desde o momento em que o usuário termina de configurar o filtro até o momento em que a máquina termina de renderizar a nova imagem na tela.

A Hipótese 3 faz referência apenas a carga inicial dos dados, ou seja, desde o momento em que o usuário termina de conectar a base de dados, ou abrir o arquivo até o momento em que o sistema disponibiliza a escolha das visualizações.

5.2.2 Resultados

A Figura 42 mostra o tempo gasto para renderizar a visão geral para bases de cinco mil, dez mil, vinte e sete mil e cinquenta mil registros. Em todas as bases o ambiente PRISMA-MDE foi mais ágil, observando que na base de cinquenta mil a ferramenta PRISMA levou mais de trinta e seis segundos para renderizar a visão geral das três técnicas de visualização utilizadas nos testes, confirmando assim a primeira parte da Hipótese 1.

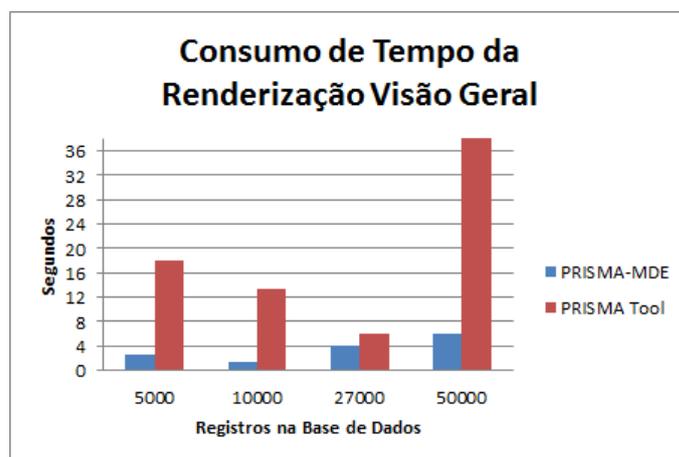


Figura 42. Consumo de tempo no processo de renderização da visão geral por tamanho da base de dados em registros

A segunda parte da hipótese um faz referência ao tempo de carga das bases de dados, que é apresentado na Figura 43, onde o PRISMA-MDE também atendeu as expectativas sendo mais ágil que a ferramenta PRISMA. De maneira geral o ambiente levou 18% de tempo a menos que a outra ferramenta.

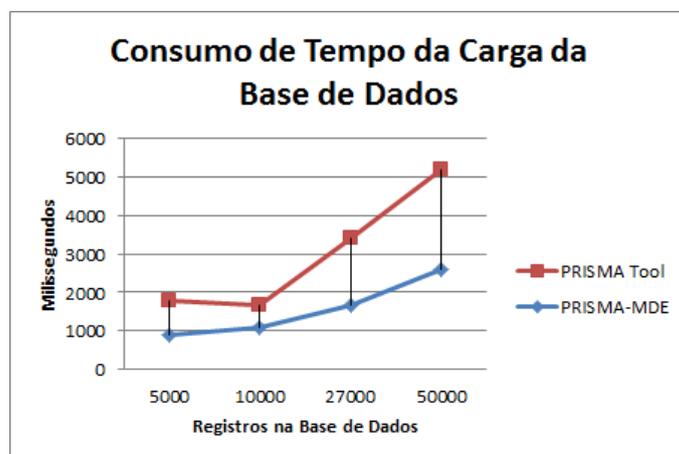


Figura 43. Consumo de tempo no processo de carga inicial dos dados por tamanho da base de dados em registros

Outro ponto avaliado foi o consumo de CPU na renderização da visão geral, de maneira geral, o ambiente PRISMA-MDE consumiu 33% a menos que a ferramenta PRISMA, conforme observado na Figura 44.

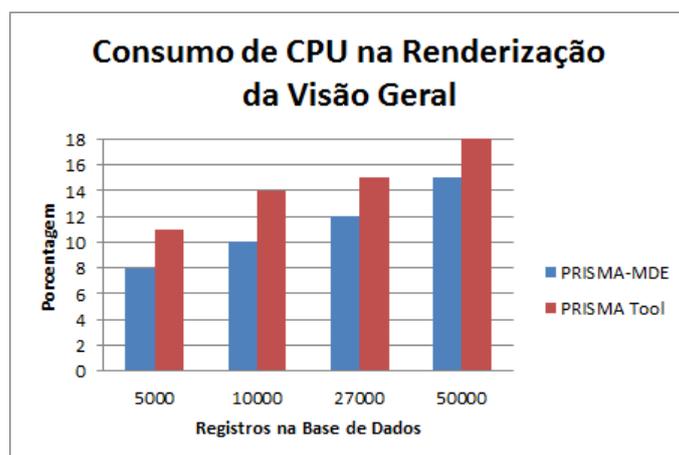


Figura 44. Consumo de CPU no processo de renderização da visão geral por tamanho da base dados em registros

Também foi avaliado o consumo de CPU durante a carga inicial dos dados, de maneira geral o PRISMA-MDE usou 10% menos CPU do que a ferramenta PRISMA. Em bases com vinte e sete mil registros o consumo CPU chegou a ser 20% menor, conforme observado na Figura 45.

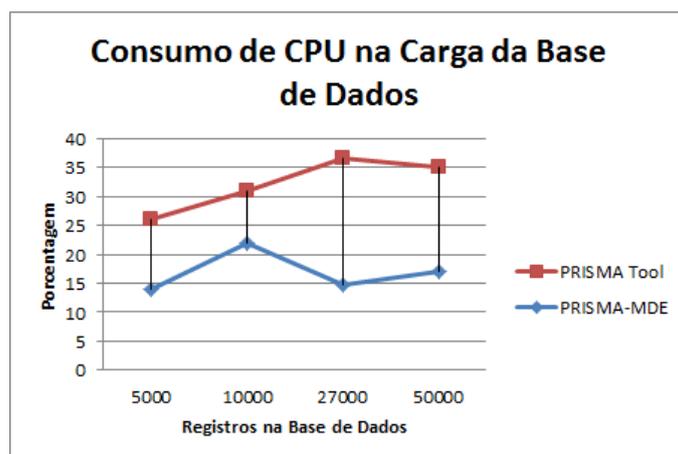


Figura 45. Consumo de CPU no processo de carga inicial da base de dados por tamanho da base dados em registros

A Figura 46 trata da Hipótese 2, referente ao consumo de CPU nas ações de filtrar, onde mais uma vez o PRISMA-MDE demonstrou um consumo menor. Em base de

dado com cinco mil registros essa diferença chegou a 16%, confirmando a hipótese dois.

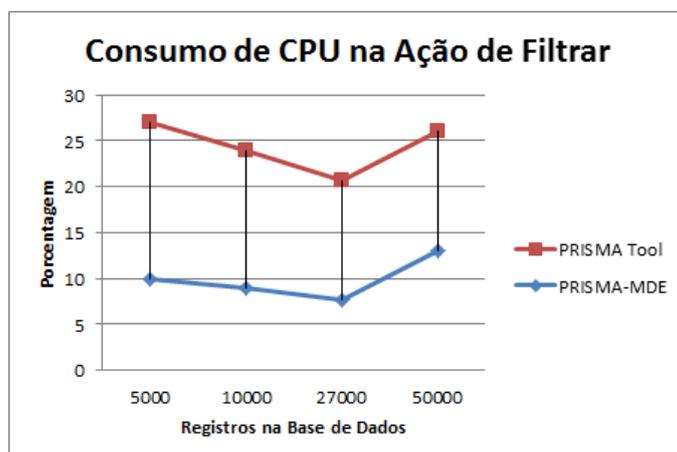


Figura 46. Consumo de CPU nas ações de filtrar por tamanho da base dados em registros

O consumo de memória da renderização da visão geral foi 9% menor para ambiente PRISMA-MDE, quando comparado à ferramenta PRISMA. Chegando a picos de até 40% a menos nas bases de vinte e sete mil registros. Esses dados estão graficamente demonstrados na Figura 47.

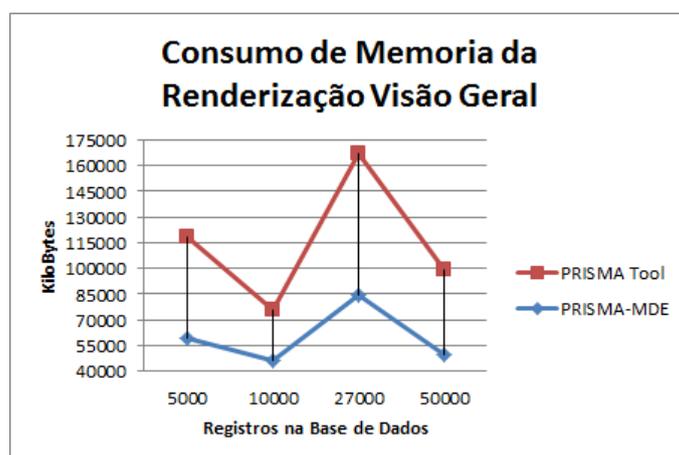


Figura 47. Consumo de memória RAM no processo de renderização da visão geral por tamanho da base dados em registros

A Hipótese 3 faz referência ao consumo de memória RAM da carga inicial dos dados. Em alguns tamanhos de base de dados existe uma diferença de consumo não muito significativa entre o ambiente PRISMA-MDE e a ferramenta PRISMA, isto se deve, pois a carga inicial das bases de dados é replicada para as diversas máquinas no

ambiente, fazendo assim o consumo de memória ser próximo ao consumo de memória da carga inicial da base de dados da ferramenta PRISMA, mesmo assim ainda é possível perceber uma diferença de 3% a menos no consumo de memória do PRISMA-MDE, como pode ser observado na Figura 48.

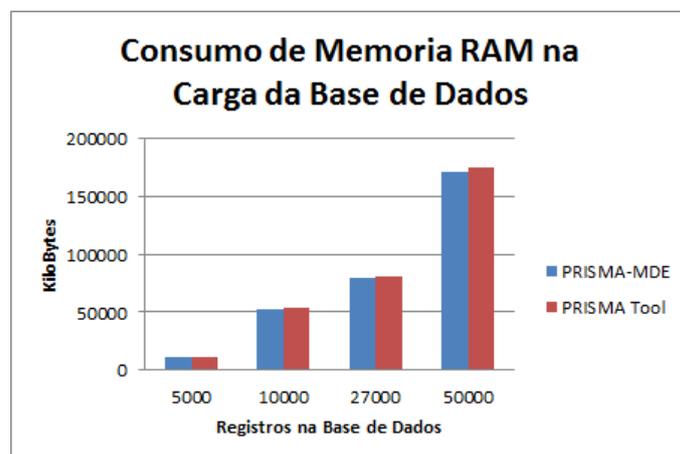


Figura 48. Consumo de memória RAM no processo de carga inicial das bases de dados por tamanho da base dados em registros

A Figura 49 demonstra o consumo de memória RAM na ação de filtrar, onde o ambiente MDE consumiu menos memória principalmente em bases maiores. Esse consumo foi 52% menor que a ferramenta PRISMA.

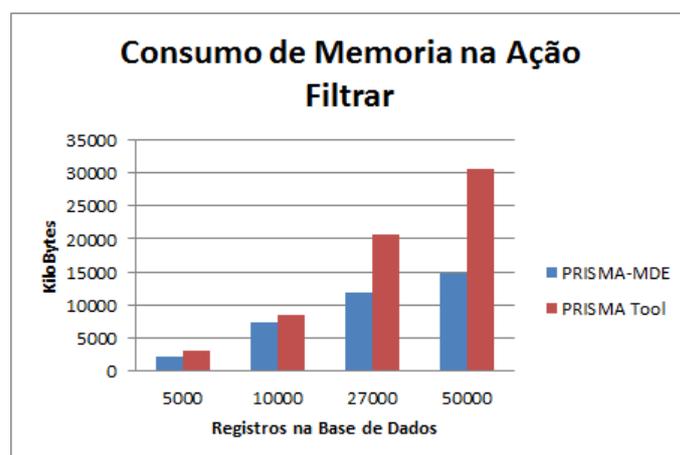


Figura 49. Consumo de memória RAM na ação de filtrar por tamanho da base dados em registros

6 CONSIDERAÇÕES FINAIS

Na literatura, há inúmeras aplicações que utilizam CMV (*Coordinated Multiple Views*), com objetivo de melhorar a percepção do usuário sobre os dados e seus relacionamentos, através de uma diversidade de técnicas de visualização sobre o mesmo conjunto de dados, ou vários conjuntos de dados relacionados. No entanto, poucas são as aplicações que focam a exploração dos dados utilizando múltiplas visões e múltiplas telas, em função do custo, complexidade de implementação, etc. A aplicação de múltiplas telas objetiva minimizar o problema de falta de espaço visual para representar a crescente quantidade de dados eletrônicos armazenados a cada dia. A falta de espaço visual pode prejudicar o processo de análise visual do usuário em função do fenômeno da oclusão. A oclusão ocorre quando dois ou mais elementos visuais se sobrepõem, obstruindo a percepção de alguns deles, e ocasionando má interpretação da visão ou perda de detalhes e relações importantes.

Pelo contexto apresentado, este trabalho se propôs a evoluir uma ferramenta de visualização de informação CMV de uma única tela para um sistema distribuído de múltiplas telas e processadores. A ferramenta de visualização de informação utilizada foi PRISMA. Como suporte de comunicação para o ambiente distribuído foi utilizada biblioteca libGlass, o núcleo do PRISMA, que gera as visualizações foi mantido, tendo suas interfaces de comunicação reconstruídas, e outros módulos de controle incorporados. A coordenação entre as visualizações apresenta agora dois níveis, a coordenação local, entre as técnicas de visualização no mesmo nó-processador, e a coordenação global, que é a coordenação entre todas as técnicas de visualização nos diversos nós-processadores que permitem a coordenação para uma determinada propriedade visual, por exemplo, cor, tamanho, transparência, etc.

Algumas diretrizes foram definidas para evolução do PRISMA, denominado de PRISMA-MDE, enquanto ambiente distribuído e escalável de múltiplas visões coordenadas de dados:

- **Interação:** o ambiente deve possuir uma interação fácil e usável, e deve permitir suporte a outras formas de interação não convencionais (kinect, wii remote,).
- **Desempenho:** PRISMA MDE deve apresentar melhor performance que o PRISMA desktop;
- **Percepção do Usuário:** a percepção do usuário deve ser melhor no PRISMA MDE quando comparado ao PRISMA desktop;
- **Configuração / Reconfiguração:** o usuário precisa ser capaz de configurar ou reconfigurar a posição, tamanho, resolução e o próprio dispositivo;
- **Sincronismo / Atualização:** os dados precisam estar sincronizados em todos os dispositivos e serem mantidos atualizados durante toda a execução do sistema;
- **Escalabilidade:** o ambiente precisa ter a capacidade de aumentar quando for necessário para o usuário, ou seja, o usuário poderá adicionar mais dispositivos durante a execução do sistema;
- **Tolerância a falhas:** o ambiente precisa suportar falhas de comunicação, ou seja, quando um dispositivo se tornar *off-line*, quando o mesmo retornar ele deverá voltar sincronizado e atualizado com os demais dispositivos;
- **Múltiplas visões coordenadas:** o ambiente deve possuir recursos para a coordenação dos dados nas múltiplas telas apresentada.

Outros objetivos desse trabalho eram conceber um método para minimizar o problema da oclusão de dados em ferramentas de visualização, possibilitar uma área maior para renderização de técnicas de visualização de informação e propor um método para o desenvolvimento de ambientes múltiplas telas para ferramentas de visualização de informação de baixo custo.

Como forma de avaliar todo o processo, foram propostos dois testes uma para aferir o desempenho do software gerado e comparar ao já existente e outro teste para avaliar a usabilidade do ambiente proposto.

O desempenho do ambiente PRISMA-MDE se mostrou melhor do que a ferramenta PRISMA. Nos testes de usabilidade, o ambiente PRISMA-MDE também se mostrou melhor do que a ferramenta PRISMA, porém o mesmo teste mostrou que o ambiente

carece de um aperfeiçoamento na interface com o usuário, a fim de buscar uma melhor interação com o mesmo.

6.1 Desafios Encontrados

O principal desafio encontrado durante o desenvolvimento deste trabalho foi o desafio técnico de entender as estruturas da ferramenta PRISMA e descobrir aonde aplicar as modificações para as novas funcionalidades, sem perda dos recursos já existentes, assim como a correção de novos bugs. Outro desafio importante foi encontrar uma base de dados grande e de domínio público que pudesse ser utilizada para execução dos testes.

6.2 Trabalhos Futuros

Os trabalhos futuros sugeridos são:

- Estender o ambiente de múltiplas telas para um ambiente colaborativo de múltiplos usuários.
- Fazer novos estudos de usabilidade a fim de melhorar a interface da do ambiente PRISMA-MDE.
- Fazer um estudo mais aprofundado sobre dispositivos de interação para ambientes de múltiplas telas.
- Adaptar o ambiente para a utilização de dispositivos como *tablets*, celulares e outros.
- Avaliar o comportamento do ambiente em uma rede geograficamente distribuída (WAN).
- Promover um estudo mais aprofundado para avaliar qual o poder escalabilidade do ambiente.

REFERÊNCIAS

- Baldonado, M., Kuchinsky, A., & Woodruff, A. (2000). Guidelines for Using Multiple Views in Information Visualization. *Advanced Visual Interfaces*, (pp. 110-119).
- Barbosa, S. D., & Silva, B. S. (2010). *Interação Humano Computador*. Rio de Janeiro: Campus.
- Bare, C. (01 de 2013). *Drawing heatmaps in R*. Fonte: R-bloggers: <http://www.r-bloggers.com/drawing-heatmaps-in-r/>
- Bertin, J. (1983). *Semiology of Graphics*.
- Bertini, E., & Stefaner, M. (15 de 03 de 2012). *Data Stories*. Acesso em 17 de 02 de 2013, disponível em How do you evaluate visualization?: <http://datastori.es/episode-3-vis-evaluation/>
- Bondi, A. (2000). Characteristics of scalability and their impact on performance. *The 2nd international workshop on Software and performance*, (pp. 195-203). Ontário, Canadá.
- Bragdoni, A. e. (2011). Code Space: Touch + air Gesture Hybrid interactions for Supporting Developer Meetings. *ACM ITS 2011*. Kobe, Japan.: ACM.
- Card, K. S. (2009). Information Visualization, using vision to think. *Morgan Kaufmann*. Cal. USA.
- Carpendale, S. (2007). Vislink: Revealing relationships amongst visualizations. . *IEEE Transactions on Visualization and Computer Graphics*, v. 13, pp. 1192–1199.
- Carr, D. A. (1999). Guidelines for Designing Information Visualization Applications. *Proceedings of Ericsson Conference in Usability Engineering*, (pp. 1-7).
- Convertino, G., Chen, J., Yost, B., Ryu, Y. S., & North, C. (2003). Exploring context switching and cognition in dual-view coordinated visualizations. *Conference on Coordinated and Multiple Views In Exploratory Visualization (CMV '03)* (p. 55). Washington: IEEE.
- Ellsworth, D., Green, G., Henze, C., Moran, P., & Sandstrom, T. (09 de 2006). Concurrent Visualization in a Production Supercomputing Environment. *IEEE Transactions on Visualization and Computer Graphics*, pp. 997-1004 .
- Forlines, C., & Lillian, R. (2008.). Adapting a single-user, single-display molecular visualization application for use in a multi-user, multi-display environment. *working conference on Advanced visual interfaces*. (pp. 367-371.). Napoli, Italy.: ACM Press.
- Godinho, P., Meiguins, B., Carmo, C., Garcia, M., Almeida, L., & Lourenço, R. (2007). PRISMA - A Multidimensional Information Visualization Tool using Multiple

- Coordinated Views. *11th International Conference on Information Visualization*, (pp. 23-32). Zurich.
- Goodall, J. R. (2009). Visualization is Better! *6th International Workshop on Visualization for Cyber Security 2009* (pp. 57-68). Atlantic City, New Jersey, USA: IEEE.
- Guimarães, M. P., Gnecco, B., Bressan, P., & Zuffo, M. (2003). Programação Distribuída Aplicada à Realidade Virtual. *CSBC*, (pp. 114-134). Porto Alegre.
- Guimarães, P. (2004). Um ambiente para desenvolvimento de aplicações para realidade virtual baseado em aglomerados gráficos. *Um ambiente para desenvolvimento de aplicações para realidade virtual baseado em aglomerados gráficos*. São Paulo, São Paulo, Brasil.
- Hart, S. G., & Stavenland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In: P. Hancock, & N. Meshkati, *Human mental workload* (pp. 183-139). Amsterdam: North Holland: Elsevier.
- Heer, J., Bostock, M., & Ogievetsky, V. (2008). A Tour Through the Visualization Zoo.
- Heijs, A. (2007). Requirements for coordinated multiple view visualization systems for industrial applications. *Fifth International Conference on Coordinated and Multiple Views in Exploratory* (pp. 76-79). DC: IEEE.
- Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *SciencExpress*, 60-65.
- Inselberg, A. (2009). *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer.
- Johanson, B. e. (2002.). PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. *15th Annual ACM Symposium on User Interface Software and Technology*. (pp. 227-234.). Paris, France.: ACM Press.
- Kenttälä, J. (11 de 06 de 2011). *CSIRT*. Acesso em 17 de 02 de 2013, disponível em What a good data visualisation should do: <http://csirtfoundry.com/blog/vis-what/>
- Kraak, M. (05 de 04 de 2003). Challenges in Geospatial Analysis and Visualization. *ISPRS Journal of Photogrammetry and Remote Sensing*, pp. 390–399, Volume 57.
- Lent, C. (04 de 09 de 2007). *Visualization: Character Relationships across Religions*. Acesso em 17 de 02 de 2013, disponível em Visualization: Genesis Word Trees: <http://www.openbible.info/blog/2007/09/visualization-genesis-word-trees/>
- Maciel, M., Meiguins, B., Loureno, R., Meiguins, A., & Godinho, P. (2008). The Impact of Multiple Coordinated Views on the Visual Data Exploration and Analysis . *12th International Conference Information Visualisation* (pp. 113 - 119). London: IEEE.
- McCune, D. (01 de 2013). *If San Francisco Crime were Elevation*. Fonte: dougmccune.com: <http://dougmccune.com/blog/2010/06/05/if-san-francisco-crime-was-elevation/>

- North, C., & Shneiderman, B. (2000). Snap-Together Visualization: A User Interface for coordinating Visualizations via Relational Schemata. *Advanced visual interfaces*, (pp. 23-26).
- Novotny, M. (2004). Visually Effective Information Visualization of Large Data. *Eighth Central European Seminar on Computer Graphics (CESCG)* (pp. 41-48). Budmerice, Slovakia: CRC Press.
- Opazo, C. (21 de 06 de 2011). *Techademia - Technology and Teaching at Vassar College*. Acesso em 16 de 02 de 2013, disponível em Open Data & Tools for Information Visualization: <http://pages.vassar.edu/acs/open-data-tools-for-information-visualization/>
- Pillat, R. M., & Freitas, C. (2006). Coordinated Views in the InfoVis Toolkit. *AVI '06 Proceedings of Advanced Visual Interfaces*. (pp. 496-499). Venezia, Italy: ACM Press.
- Pillat, R. M., Valiati, E. R., & Freitas, C. M. (2005). Experimental Study on Evaluation of Multidimensional Information Visualization Techniques. *CLIH'05*, (pp. 20-30). Cuernavaca - Mexico.
- Plaisant, C. (2001). *Information Visualization - Lecture Notes*.
- Rocha, H. V., & Baranauskas, M. C. (2003). *Design e Avaliação de Interfaces Humano-Computador*. Campinas: Instituto de Computação - Universidade Estadual de Campinas.
- Rubin, J. (1994). *Handbook of usability testing*. New York: John Wiley & Sons.
- Shen, C. e. (2006). Three Modes of Multi-Surface Interaction and Visualization. *Information Visualization and Interaction Techniques for Collaboration across Multiple Displays Workshop associated with CHI'06 International Conference*, (pp. 1-6). Montreal, CANADA.
- Shimabukuro, M., Flores, E., Oliveira, M., & Levkowitz, H. (2004). Coordinated Views to Assist Exploration of Spatio-Temporal Data: a Case Study. *the Second International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV'04)*. IEEE.
- Shneiderman, B. (1992). Tree visualization with tree-maps: A 2-D space-filling approach. *ACM Transactions on Graphics*, 92-99.
- Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings of the IEEE Symposium on Visual Languages*, (pp. 336 -343).
- Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Proceedings of the IEEE Symposium on Visual Languages* (pp. 336-343). Washington: IEEE Computer Society Press.
- Shneiderman, B. (01 de 01 de 2009). *Treemaps for Space-Constrained Visualization of Hierarchies*. Acesso em 01 de 05 de 2012, disponível em HCIL - Department of Computer Science - University of Maryland: <http://www.cs.umd.edu/hcil/treemap-history>
- Shneiderman, B., & Wattenberg, M. (2001). Ordered Treemap Layouts. *IEEE Symposium on Information Visualization*., (pp. 73-78).

- Spence, R. (2001). Information Visualization. *In Addison Wesley – ACM Press.*, p. 459.
- Tanenbaum, A. S. (2007). *Distributed Systems: Principles and Paradigms*. PEARSON.
- Thomas, J. J., & Cook, K. A. (2005). *Illuminating the Path: The research and development agenda for visual analytics* (I ed.). (J. J. Thomas, & K. A. Cook, Eds.) USA: IEEE CS Press.
- Waldner, M. L. (2009). Design Considerations for Collaborative Information in Multiple Display Environment. *CoVIS - Workshop on Collaborative Visualization on Interactive Surfaces*. USA.
- Wallace, J. R., Mandryk, R. L., & Inpen, K. M. (2008.). Comparing content and input redirection in MDEs. *the 2008 ACM conference on Computer supported cooperative work*. (pp. 157-166.). San Diego, CA, USA.: ACM Press.
- Ward, M. O., Grinstein , G., & Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*.
- Ware, C. (2004). *Information Visualization: Perception for Design* (2ª edição ed.). San Francisco, CA, USA: Elsevier.

Anexo 01 – Formulário de Avaliação de usabilidade

Entrevistado: _____

1. O maior número de programas de pós-graduação que receberam bolsas no Brasil foi no ano de 2010? **Tempo:** _____

Resposta: _____

Qual a Complexidade da tarefa:

- a) Muito fácil
- b) Fácil
- c) Normal
- d) Difícil
- e) Muito difícil

2. Tarefa 01: Houve um aumento na quantidade de programas de pós-graduação que receberam bolsas da CAPES do ano de 2010 para 2011 no estado do Rio de Janeiro na grande área de estudo exatas e ciências da terra? Se sim qual foi o aumento? **Tempo:** _____

Resposta: _____

Qual a Complexidade da tarefa:

- a) Muito fácil
- b) Fácil
- c) Normal
- d) Difícil
- e) Muito difícil

3. Tarefa 02: Comparando as grandes áreas de estudos ciências exatas e da terra com ciências da saúde, qual foi o estado que teve mais programas de pós-graduação recebendo bolsas da CAPES no período de 2008 a 2011. **Tempo:** _____

Resposta: _____

Qual a Complexidade da tarefa:

- a) Muito fácil
- b) Fácil
- c) Normal
- d) Difícil
- e) Muito difícil

4. Tarefa 01: Qual a grande área de estudos e qual o estado que tiveram mais programas de pós-graduação que receberam bolsas da CAPES no ano 2011. **Tempo:** _____

Resposta: _____

Qual a Complexidade da tarefa:

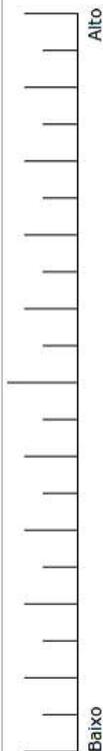
- a) Muito fácil
- b) Fácil
- c) Normal
- d) Difícil
- e) Muito difícil

Anexo 01 – Formulário de Avaliação de usabilidade – Continuação

- 1. Complexidade de entendimento da base de dados:**
 - a) Muito fácil
 - b) Fácil
 - c) Normal
 - d) Difícil
 - e) Muito difícil
- 2. Como o usuário se sentiu ao utilizar três monitores:**
 - a) Muito confortável
 - b) Confortável
 - c) Normal
 - d) Desconfortável
 - e) Muito desconfortável
- 3. Como usuário se sentiu ao utilizar o mouse nos três monitores:**
 - a) Muito confortável
 - b) Confortável
 - c) Normal
 - d) Desconfortável
 - e) Muito desconfortável
- 4. Alguma vez configurou mais de uma técnica para responder a mesma tarefa?**
 - a) Sim
 - b) Não
- 5. Classifique a ação configurar:**
 - a) Muito fácil
 - b) Fácil
 - c) Normal
 - d) Difícil
 - e) Muito difícil
- 6. Classifique a ação filtrar:**
 - a) Muito fácil
 - b) Fácil
 - c) Normal
 - d) Difícil
 - e) Muito difícil
- 7. Classifique a função de coordenação entre as visões de dados:**
 - a) Não ajuda
 - b) Pouco ajuda
 - c) Normal
 - d) Ajuda
 - e) Ajuda muito
- 8. Fez uso do brushing (destacar dados) para resolver algumas das tarefas?**
 - a) Sim
 - b) Não
- 9. O uso de múltiplas telas é útil?**
 - a) Sim
 - b) Não
- 10. É melhor ter técnicas ou visões diferentes dos mesmos dados?**
 - a) Técnica
 - b) Visão
 - c) Não se aplica

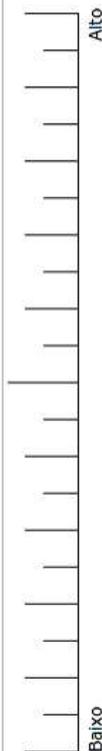
Avaliação de Carga de Trabalho NASA-TLX:

Demanda Mental:



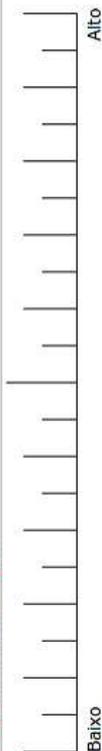
Quanto de atividade mental e perceptiva foi necessário (raciocínio, decisão, cálculo, lembrança, busca, identificação)?

Demanda Física:



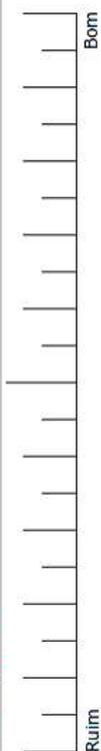
Quanto de esforço físico foi necessário (pressionamento de botões, movimentos com o mouse...)?

Demanda Temporal:



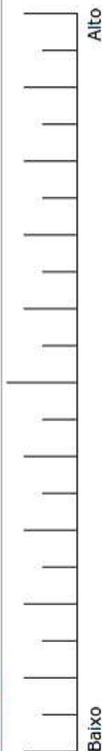
O quão pressionado pelo tempo você se sentiu?

Desempenho:



O quão bem sucedido você acha que foi no alcance dos objetivos das tarefas?

Esforço:



O quão arduamente você teve que trabalhar para atingir o seu nível de desempenho?

Frustração:



O quão inseguro, desencorajado, irritado, estressado e chateado você se sentiu?

Enviar