

# Uma Ferramenta Web para Análise Visual de Dados

Leandro Almeida<sup>1</sup>, Bianchi Meiguins<sup>1</sup>, Aruanda Gonçalves<sup>2</sup>, Rodrigo Lourenço<sup>1</sup>,  
Nikolas Carneiro<sup>3</sup>, Rafael Guimarães<sup>3</sup>

<sup>1</sup> Programa de Pós-Graduação em Ciências da Computação, Universidade Federal do Pará

<sup>2</sup> Área de Ciências Exatas e Tecnologias – Centro Universitário do Pará

<sup>3</sup> Faculdade de Computação – Universidade Federal do Pará

almeida85@gmail.com, bianchi.serique@terra.com.br, aruanda@redeinformatica.com.br,  
{rodrigo1503, nikolas.carneiro, rafaveguim}@gmail.com

**Abstract.** The goal of this paper is to present a prototype of a web tool for visual data analysis. The application supports multiple coordinated views including three visualization techniques - scatter plot, treemap and parallel coordinates – and statistics charts. Coordination among views is supported by color, filter, selection and brushing mechanisms. Coordination allows the user to intuitively identify existing relationships on the analyze dataset. Standard web technologies and design patterns were applied in the development of the presented prototype.

**Keywords:** information visualization, multiple coordinated views, web

## 1 Introdução

A diversidade e a frequência de uso de sistemas de informações nos diversos setores da sociedade proporcionaram um enorme acúmulo de dados eletrônicos, e que cresce a cada dia. É consenso comum, que nesses depósitos eletrônicos de dados encontram-se informações valiosas, ainda pouco ou não exploradas, que poderiam melhorar a eficiência de um processo na indústria, gerar novos rumos de uma pesquisa científica, possibilitar ao governo a aplicação melhor de recursos escassos, tornar uma empresa mais competitiva no mercado de serviços e vendas de produtos, entre muitos outros exemplos. Utilizar esses dados para balizar tomadas de decisão no presente e antecipar ações no futuro é uma grande necessidade e um grande desafio.

Muitas das tomadas de decisão ultrapassam as fronteiras geográficas, uma das características do que se conhece por mundo globalizado, e, para facilitar esse processo, os meios de comunicação são necessários, destacando-se dentre esses a Internet. Com a popularidade da Internet como meio de comunicação e troca de informações é natural pensar em direcionar esforços para desenvolver aplicações computacionais multiusuários para esta plataforma que possam auxiliar o processo de tomada de decisões.

Dentre as aplicações que possibilitam analisar grande quantidade de dados de forma rápida, interativa e intuitiva destacam-se as aplicações de Análise Visual de Dados, ou Visualização de Informação, que criam “imagens” de um conjunto de dados compostos por atributos visuais, tais como cor, tamanho e formas geométricas,

para que o usuário tenha uma melhor percepção dos dados e seus relacionamentos, através da criação de cenários para testar e confirmar hipóteses [8]. Assim, o sentido humano da visão, quando explorado corretamente torna-se um recurso muito eficiente para essa tarefa, uma vez que este sentido apresenta extrema facilidade e rapidez na detecção de transformações nas imagens [4].

Atualmente, a maioria das aplicações de Análise Visual de Dados são voltadas para ambientes desktops utilizando múltiplas visões coordenadas [3], quando essas ferramentas se apresentam em ambiente Web, elas se mostram com poucos recursos, com interface pobre, e normalmente com apenas uma técnica de visualização de informação para representação dos dados. O uso de múltiplas visões permite que o usuário tenha visões diferenciadas do mesmo conjunto de dados e a coordenação mantém a consistência entre as visões e permite a descoberta de relacionamentos entre os dados de visões diferentes [1].

Nesse contexto, a proposta desse trabalho é apresentar um protótipo que contempla as principais demandas de uma boa ferramenta de visualização de informação para plataforma web, utilizando múltiplas técnicas de visualização de informação coordenadas, e que seja de fácil interação e disponibilização. O artigo está organizado da seguinte forma: a seção 2 apresenta os trabalhos relacionados; a seção 3 descreve o protótipo, detalhando arquitetura, recursos e cenários de uso; e, por fim, a seção 4 apresenta as considerações finais e trabalhos futuros.

## 2 Trabalhos Relacionados

Dentre os trabalhos relacionados, destacam-se:

- Snap-Together [5] é uma ferramenta de visualização que permite ao usuário “misturar e agregar” dinamicamente diferentes visualizações e coordenações, para construir interfaces de exploração customizadas sem precisar programar. A Figura 1 mostra um ambiente de visualização coordenada no Snap-Together para exploração de dados de censos dos estados e regiões dos Estados Unidos. Os usuários podem explorar estados de perspectivas nominais, geográficas e numéricas usando a árvore (outliner), o mapa e o gráfico de dispersão (scatterplot). Ao selecionar um estado são exibidas as informações detalhadas do mesmo através da tabela e o mapa de árvores (treemap).

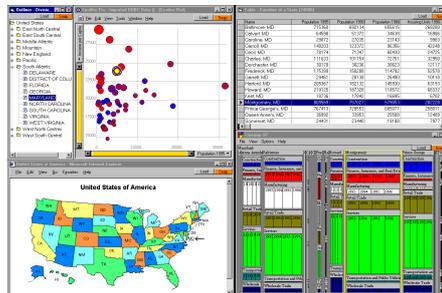


Fig. 1. Ferramenta Snap-Together utilizando visões e seleção coordenadas [5].

- VisGets [2] são widgets (ou pequenas aplicações) de visualização de informação cuja manipulação gera uma consulta web. O objetivo geral é oferecer suporte ao usuário de modo a obter uma visão geral sobre os recursos da web e oferecer um meio de filtrar visualmente os dados, através de consultas de busca dinâmicas, que combinam filtros de mais de um atributo. Uma das formas de coordenação é através do mouse. Parando o mouse sobre um elemento visual, todos os demais elementos relacionados são destacados em todas as visões dos VisGets (Figura 5).



Fig. 2. Seleção em três widgets relacionados [2].

### 3 Protótipo

Essa seção descreve as principais características e funcionalidades do protótipo.

#### 3.1 Principais Características

O protótipo foi desenvolvido segundo o modelo cliente-servidor. O cliente é o próprio navegador web do usuário e o servidor constitui-se de um conjunto de classes Java implantadas em um servidor de aplicação (Apache Tomcat 6). Uma das principais preocupações iniciais no projeto foi o lado cliente. Que tecnologias utilizar? As visões seriam renderizadas no cliente ou no servidor?

Decidiu-se por clientes mais leves, pois todo o processamento é centralizado no servidor (consulta, cálculos e renderizações). Essa opção permite que uma máquina cliente não necessite ter grande poder de processamento para extrair informações de bases de dados maiores.

As tecnologias utilizadas no desenvolvimento incluem para o cliente a combinação de HTML com CSS para construção da interface do protótipo, associado com Javascript para a manipulação e geração automática dessa interface. Para a comunicação com o servidor utilizou-se a tecnologia AJAX (Asynchronous Javascript and XML) e notações JSON (Javascript Object Notation). As visualizações são geradas no servidor a partir das requisições enviadas pelo cliente, sendo então retornadas como imagens para serem exibidas.

A tela principal apresenta uma área para visualização e uma para configuração, composta por abas. Na área de visualização, a primeira aba apresenta todas as visões disponíveis, que são as técnicas de visualização e os gráficos auxiliares. As demais abas apresentam cada visão de forma separada. Na área de configuração, os controles

são organizados em abas e sub-abas, que são: visões, filtros, cores e detalhes (Fig. 3). Na região inferior do protótipo existe uma barra de status e uma barra de progresso, para indicar o percentual de itens que estão sendo visualizados do total.

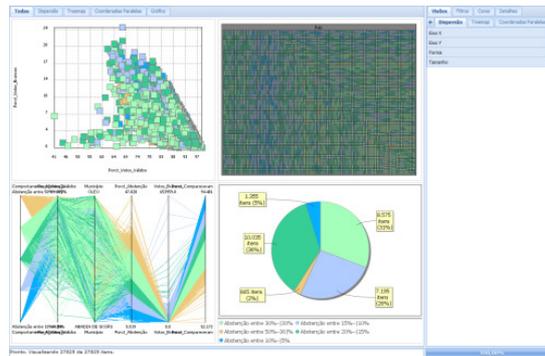


Fig. 3. Tela principal do protótipo.

Três técnicas de visualização foram implementadas no Prisma [3], sendo reutilizadas nesse protótipo, que são: dispersão, treemap e coordenadas paralelas. Além das técnicas, também estão disponíveis ao usuário gráficos auxiliares, que são descritos na seção 4.5. Como no Prisma, as coordenações entre as visões são possíveis através da cor, filtros, seleção, brushing e detalhes sob-demanda.

### 3.2 Arquitetura

A arquitetura do protótipo é ilustrada na Fig. 4.

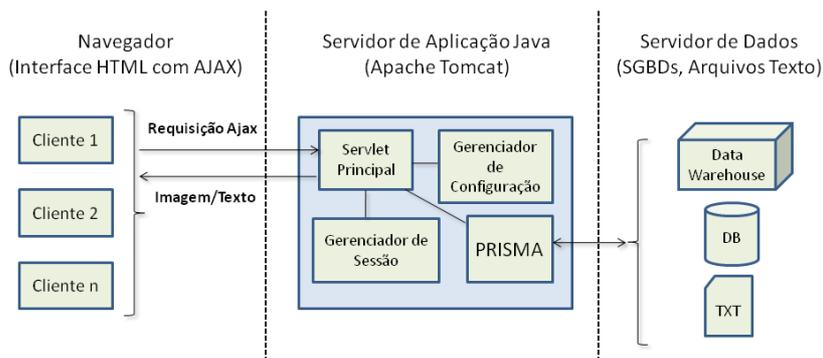


Fig. 4. Visão geral da arquitetura.

Como já citado, o cliente é o navegador Web do usuário e consiste de uma interface HTML para interação. O servidor é composto por um Servlet Principal, um Gerenciador de Sessão, um Gerenciador de Configuração e a aplicação Prisma. Define-se cada um:

- Servlet Principal: classe responsável por receber e tratar as requisições HTTP enviadas a partir dos clientes. Também é responsável por encaminhar as respostas dessas requisições ao cliente solicitante. Possui uma instância do Gerenciador de Sessão, do Gerenciador de Configuração e do Prisma.
- Gerenciador de Sessão: responsável por gerenciar as sessões de cada usuário, verificando a validade de uma sessão a cada requisição.
- Gerenciador de Configuração: responsável por criar e recuperar os contextos de cada usuário, ou seja, o conjunto de configurações das visões e filtros realizados pelo usuário.
- PRISMA: conjunto de classes responsáveis pela construção da visualização, contendo o modelo de dados, o mapeamento visual dos atributos, a gerência de conexões com as fontes de dados, a realização de filtros, a renderização e coordenação das visões, dentre outros.

A visualização e interação são realizadas no navegador. Utilizou-se duas bibliotecas Javascript para a construção da interface gráfica, – jQuery e ExtJS – pois abstraem as diferenças existentes entre diversos navegadores, além de auxiliarem na composição da interface através de componentes disponíveis nessas bibliotecas, como as abas. As requisições são enviadas ao servidor através do objeto XMLHttpRequest, com os parâmetros encapsulados em JSON.

No servidor, os dados são carregados, processados e filtrados. Também é no servidor que as visões são geradas e transformadas em imagens, que são enviadas como resposta ao cliente. Eventualmente, textos podem ser enviados como resposta, contendo informações de detalhe sobre um determinado item selecionado do conjunto de dados, bem como dados importantes para a geração automática da interface, que será explicada na seção seguinte.

Nessa primeira versão do protótipo algumas limitações são impostas:

- as fontes de dados, incluindo textos pré-formatados e bases de dados relacionais, são previamente cadastradas no servidor por um administrador;
- as visões no lado cliente são restritas a esses dados pré-especificados;
- a disposição das visões na primeira aba do protótipo são estáticas, ou seja, o usuário não escolher quais técnicas serão visualizadas e como essas técnicas ficarão dispostas na tela (reorganizar o *layout*).

### 3.3 Geração Automática da Interface

A interface é constituída de HTML e CSS criados dinamicamente através dos frameworks Javascript utilizados. Na primeira vez em que o protótipo é carregado, ou seja, quando é solicitada uma visualização de uma base de dados pelo cliente, são enviadas informações sobre a mesma, como nomes, tipos e valores distintos para cada coluna, configuração inicial da visualização, modelo de cores, formas e tamanhos (ver Fig. 5).

A partir dessas informações, o cliente gera a interface automaticamente via Javascript, criando controles para: configuração de atributos visuais (comboboxes); seleção de cores, formas e tamanhos; e filtros. A Fig. 6 ilustra uma interface gerada dinamicamente.

Qualquer interação com algum desses controles gera uma requisição HTTP via AJAX, que é enviada ao servidor, solicitando a atualização das visões. Por exemplo, mudar um atributo visual da cor implicaria no reenvio ao cliente por parte do servidor tanto das visões modificadas (imagens), quanto do modelo de cores (metadados) para alterar o controle na interface.

```

1 {
2   "colunas": [
3     {
4       "nome": "REGIAO",
5       "tipo": "STRING",
6       "descricao": "DISCRETO",
7       "valores": ["NORTE", "NORDESTE", "CENTRO-OESTE", "SUL", "SUDESTE"]
8     },
9     {
10      "nome": "Votos_Branco",
11      "tipo": "INTEGER",
12      "descricao": "CONTINUO",
13      "min": 0,
14      "max": 653,959
15    },
16    ...
17  ]
18 }

```

Fig. 5. Exemplo de JSON com metadados.



Fig. 6. Interface de configuração das cores.

### 3.4 Técnicas de Interação

Nesta seção são detalhados os mecanismos de interação do protótipo, que atendem aos requisitos mínimos de uma boa ferramenta de visualização.

**Detalhes sob demanda.** O recurso de detalhes sob-demanda é ativado através do clique do mouse em um item qualquer da visão. Ao clicar num item, o servidor envia os valores dos atributos configurados na aba “Detalhes”, e então o cliente as exibe sobre o item. A Fig. 7 mostra a utilização desse recurso, bem como a interface de configuração dos atributos.

**Brushing.** A técnica de brushing consiste em destacar um conjunto de dados em uma visão, destacando os mesmos dados nas demais visões [7]. Para realizar brushing entre as técnicas de visualização, basta clicar e/ou arrastar segurando a tecla CTRL em um item/área no dispersão ou coordenadas paralelas (ver Fig. 8). Além disso, é possível realizar brushing cumulativo, ou seja, destacar diferentes regiões ou grupos em uma visão. Para desfazer um brushing, basta clicar com o botão direito do mouse sobre a visão.

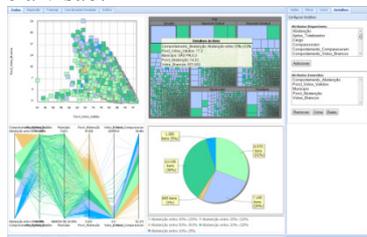


Fig. 7. Detalhes sobre um item.

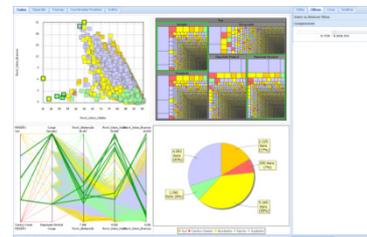


Fig. 8. Brushing coordenado.

**Zoom.** A técnica de zoom está entre os itens de interesse, pois permite focar em um subconjunto dos dados para análise. Para realizar o zoom o usuário deve clicar e arrastar uma área no dispersão ou coordenadas paralelas. No treemap, basta clicar duas vezes em um grupo. Além disso, o zoom é coordenado, fazendo com que os itens selecionados em uma técnica sejam também selecionados nas demais visões [4].

**Consultas Dinâmicas.** O protótipo utiliza consultas dinâmicas [6] para a realização de filtros. Dois tipos de controles são gerados, de acordo com o tipo de dados: caixas de marcação (checkbox) e selecionadores de intervalos (rangeslider). Dados categóricos são mapeados para checkboxes, e dados numéricos para rangesliders. O usuário pode selecionar quais atributos deseja filtrar. Para mais detalhes, ver o trabalho anterior [3].

### 3.5 Gráficos Auxiliares

Existem quatro tipos de gráficos disponíveis: pizza, barra (histograma), linha e medidor. O usuário pode incluir vários gráficos para análise simultânea. Cada gráfico pode ser configurado de acordo com as necessidades do usuário, e podem ser coordenados pelo uso do filtro.

### 3.6 Cenários de Uso

Nesta seção é descrito um exemplo de uso do protótipo, realizado na primeira versão do protótipo, visando mostrar a utilização dos recursos para a extração de informações. A base de dados selecionada para o exemplo foi a base de eleições no ano de 2006 no Brasil. A base contém 20 colunas, sendo 8 categóricas e 12 numéricas, e 27.825 registros. A base contém informações sobre o nome do município, nome da região, nome do cargo político, número de votos válidos, brancos e nulos, número de abstenções, dentre outros.

Como primeiro cenário, analisou-se quais os estados que tiveram maiores índices de votos nulos para deputado estadual. Para alcançar esse objetivo, primeiro foi realizado um filtro removendo os demais cargos políticos que não eram de interesse. Em seguida, configurou-se o eixo Y no dispersão para exibir a porcentagem de votos nulos. No treemap foi realizado um agrupamento por região e estado. Por fim, foi realizado um brushing na área superior do dispersão de modo o correlacionar com o treemap. Assim, os itens que estão com valores mais altos no eixo Y foram identificados no treemap (Fig. 9). Percebeu-se rapidamente que os estados da Bahia, Maranhão e Goiás estiveram entre os estados com as maiores porcentagens de votos nulos.



Fig. 9. Brushing realizado no dispersão destacou três grupos no treemap.

Como segundo cenário exemplo, analisou-se a porcentagem de eleitores que votaram por região. Adicionou-se cinco medidores para representar a porcentagem de comparecimento às urnas para cada uma das regiões. Como ilustrado na Fig. 10, a região que teve maior índice de comparecimento foi a região sul, seguida da região sudeste, nordeste, centro-oeste e norte.

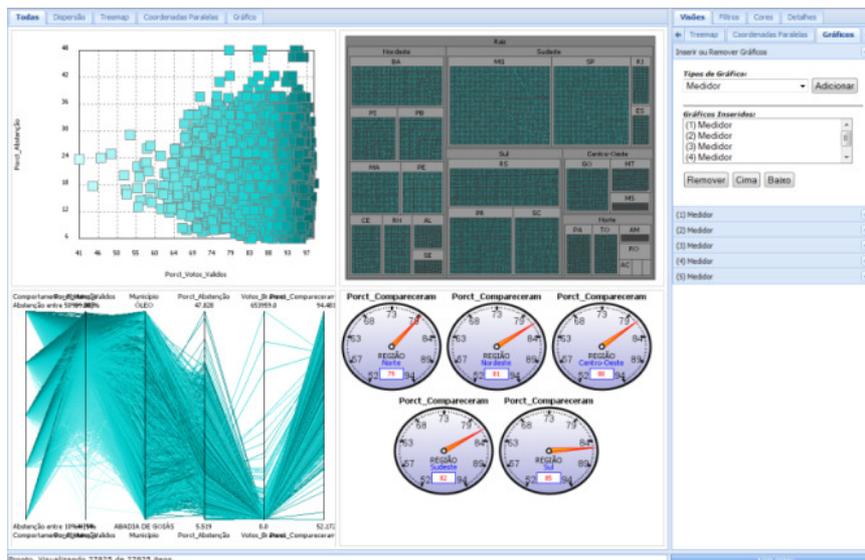


Fig. 10. Cinco medidores exibindo porcentagem de comparecimento por região.

## 4 Considerações Finais

Este artigo apresentou um protótipo de ferramenta de visualização de informação para plataforma Web. Utilizou-se tecnologias padrão Internet/Web, tais como: Javascript, Ajax, Java, entre outros. Três técnicas de visualização de informação foram implementadas: dispersão, treemap, coordenadas paralelas, além de gráficos estatísticos auxiliares. A coordenação é previamente definida entre as visões de dados, incluindo-se os gráficos auxiliares, e estão organizadas como múltiplas visões coordenadas na mesma tela ou múltiplas visões coordenadas em telas separadas, ou seja, visões individuais.

O benefício de se utilizar a plataforma web é em razão da popularidade da Internet, que tem estabelecido a plataforma como ideal para o desenvolvimento de novas aplicações em função de atingir um número maior de pessoas, máquinas clientes sem a necessidade de grande poder de processamento, facilidade de manutenção, distribuição das aplicações desenvolvidas e ser um ambiente propício a troca de informações e colaborações. Um dos problemas dessa plataforma é a limitação dos recursos disponíveis no navegador do usuário se comparado com o desktop, que disponibiliza interação e gerência dos recursos da máquina mais avançados, enriquecendo a experiência do usuário. Outro problema importante ressaltar, é que o tempo de resposta das interações do usuário com a aplicação depende em grande parte da velocidade de conexão com a internet, uma vez que as respostas são apresentadas como imagens.

O desafio principal que norteou o desenvolvimento inicial deste trabalho foi prover o mesmo nível de qualidade de interação e visualização que o Prisma (desktop), ou o mais próximo possível desse nível de qualidade, em função dos recursos limitadores que são impostos a aplicações web interativas.

No contexto apresentado pode-se afirmar que o protótipo atende os requisitos mínimos para uma boa ferramenta de visualização de informação e de uma ferramenta com múltiplas visões coordenadas. Apesar de ainda estar em pleno desenvolvimento, já apresentou desafios interessantes para uma ferramenta de visualização para Web, alguns deles são:

- Criar/gerenciar contexto de cada usuário;
- Responder as requisições de usuários diferentes de acordo com cada contexto, gerando visões diferentes para cada usuário;
- Mapear a interação do usuário no navegador para o correto processamento no servidor (exemplo: selecionar um item ou uma área pra fazer zoom/brushing);
- Gerar automaticamente a interface a partir dos metadados;
- Construir a interface de customização para a os itens visuais: formas, cores e tamanhos, ou seja, ao escolher um determinado atributo para forma é necessário exibir uma legenda e entender a correlação do valor de atributo e uma determinada forma geométrica;

Destacam-se os principais desenvolvimentos futuros:

- Avaliações ergonômicas da interface;
- Aplicação de ensaios de usabilidade;
- Suporte ao *upload* para o servidor de bases de dados em texto pré-formatado para visualização imediata;

- Desenvolvimento de novos mecanismos de filtros, como: filtros através de consultas por palavras, múltiplos filtros em um mesmo atributo, dentre outros;
- Configuração de novos layouts pelo usuário para as múltiplas visões coordenadas na mesma tela;
- Visualização simultânea de bases de dados diferentes, com controles diferentes de interação, com propósitos de comparação das bases de dados;
- Customização do tipo (técnica de visualização de informação) e quantidade de visões a serem utilizadas, como serão coordenadas entre si e o conjunto de dados que devem representar;
- Desenvolvimento de algoritmos otimizados para consultas a grandes bases de dados voltados para contexto do protótipo;
- Extensão da ferramenta para se tornar uma ambiente colaborativo;

## Referências

1. Baldonado, M. Q. W.; Woodruff, A.; Kuchinsky, A. Guidelines for using multiple views in information visualization. Proceedings of the working conference on Advanced Visual Interfaces, pp. 110 – 119. Palermo. Italy. 2000.
2. Dörk M., Carpendale S., Collins C. and Williamson C. VisGets: Coordinated Visualizations for Web-based Information Exploration and Discovery. Proceedings of IEEE Symposium on Information Visualization, 2008.
3. Godinho, P. I., Meiguins, B. S., Meiguins, A. S., Casseb do Carmo, R. M., de Brito Garcia, M., Almeida, L. H., and Lourenco, R. PRISMA - A Multidimensional Information Visualization Tool Using Multiple Coordinated Views. Proceedings of IEEE Symposium on Information Visualization, 2007.
4. Kerren, A.; Ebert, A.; Meyer, J. Human-Centered Visualization Environments. Germany: Springer, 2007. 403 p.
5. North, Chris; Shneiderman, Ben. Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata. Proceedings of the working conference on Advanced visual interfaces, pp 128-135. Palermo, Italy. 2000.
6. Shneiderman, B. Dynamic queries for visual information seeking. IEEE Software, 11(6):70–77, 1994.
7. Shneiderman, B.; Card, S. K.; Mackinlay, J. Readings in Information Visualization: Using Vision to Think. 1 ed. Morgan Kaufmann, 1999.
8. Spence, R. Information Visualization: Design for Interaction. Barcelona: Acn Press. Second Edition, 2007.