

Luis Gustavo Neves da Silva

# Visualização de Informação em Sistemas Científicos

**Rio de Janeiro  
2015**



Luis Gustavo Neves da Silva

# Visualização de Informação em Sistemas Científicos

Dissertação apresentada à Escola de Matemática Aplicada da Fundação Getúlio Vargas, para a obtenção de Título de Mestre em Ciências.

Orientador: Asla Sá

**Rio de Janeiro  
2015**

Silva, Luis Gustavo Neves da  
Visualização de informação em sistemas científicos / Luis Gustavo  
Neves da Silva. - 2015.  
75 f.

Dissertação (mestrado) – Fundação Getulio Vargas, Escola de  
Matemática Aplicada.

Orientadora: Asla Medeiros e Sá.

Inclui bibliografia.

1. Visualização da informação. 2. Ciência – Processamento de dados.  
3.  
Software - Desenvolvimento. 4. Computação gráfica. I. Sá, Asla  
Medeiros. II.  
Fundação Getulio Vargas. Escola de Matemática Aplicada. III. Título.

CDD – 005.118


**LUIS GUSTAVO NEVES DA SILVA**

**VISUALIZAÇÃO DE INFORMAÇÃO EM SISTEMAS CIENTÍFICOS.**

Dissertação apresentada ao Curso de Mestrado em Modelagem Matemática da Informação da Escola de Matemática Aplicada da Fundação Getulio Vargas para obtenção do grau de Mestre em Modelagem Matemática da Informação.

Data da defesa: 08/12/2014.

**ASSINATURA DOS MEMBROS DA BANCA EXAMINADORA**

A handwritten signature in blue ink, appearing to read 'Asla Medeiros e Sá', is written over a horizontal line.

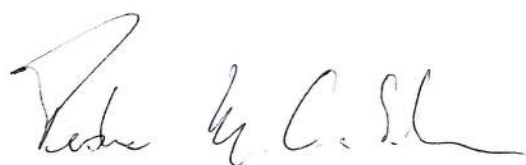
**Asla Medeiros e Sá**  
Orientador (a)

A handwritten signature in blue ink, appearing to read 'Renato Rocha Souza', is written over a horizontal line.

**Renato Rocha Souza**

A handwritten signature in blue ink, appearing to read 'Flávio Coelho Codeço', is written over a horizontal line.

**Flávio Coelho Codeço**

A handwritten signature in blue ink, appearing to read 'Pedro Mário Cruz e Silva', is written over a horizontal line.

**Pedro Mário Cruz e Silva**

## Agradecimentos

Agradeço ao consultor da Petrobras Flávio de Lemos Gondin, por ter acreditado e defendido na empresa a importância da realização deste mestrado, o qual não poderia ter sido realizado sem seu apoio.

Gostaria de agradecer aos professores Jonh Edson e Flávio Coelho por terem me motivado a fazer o mestrado.

Agradeço também a minha orientadora, Asla Sá, pela sua colaboração e constante entusiasmo na busca pelo conhecimento.

Gostaria de agradecer a todos os professores do mestrado, pela sua dedicação ao ensino. Em especial, gostaria de agradecer aos professores Renato Rocha, Moacyr Silva e Antonio Carlos Saraiva Branco pela motivação e apoio ao longo do curso.

Agradeço aos clientes dos projetos Abelardo Borges, Adriana de Souza, Carlos Eduardo, Fabio Liporace, Leticia Forrer, Maira Andrade, Sérgio Gregório e Vivian Passos pela colaboração, confiança e amizade sempre presentes ao longo da condução de todos esses projetos.

Agradeço aos colegas de equipe Luiz Eduardo Borges, Ricardo Duarte, Rodrigo Rosa e Viviane Nascimento pelo espírito de equipe, bom humor e dedicação aos projetos.

Agradeço aos colegas Daniel Fialho, Elisabete Thomaselli e Ronaldo Rufino pelo apoio e motivação.

Agradeço a minha esposa pelo seu amor, carinho e compreensão sempre.

### *Resumo*

Este trabalho é um estudo sobre a aplicação de técnicas de visualização de informação em sistemas de software científico, i.e., sistemas de software voltados para matemática, ciências e engenharias. Enquanto sistemas dessa natureza normalmente fazem uso da visualização científica e figuram como caso de sucesso nessa área, nem sempre são projetados considerando os princípios de visualização de informação. Esse trabalho tem por objetivo avaliar a aplicação desses conceitos em alguns sistemas reais, desenvolvidos com diferentes níveis de entendimento nessa área de conhecimento e com base nessas experiências propor o desenvolvimento de componentes de software capazes de facilitar a criação de sistemas semelhantes e ao mesmo tempo promover a aplicação destes conceitos.

**Palavras-chave:** Visualização de Informação, Sistemas de software científico, Visualização Científica, Análise de Dados

### *Abstract*

This work is a study on the application of techniques of information visualization in scientific software, ie, software systems focused on math, science and engineering systems. While such systems typically make use of scientific visualization and are listed as a success story in this area they are not always designed considering the principles of information visualization. This study aims to evaluate the application of concepts in some real systems, developed with different levels of understanding in this area of knowledge and based on these experiences suggest the development of software components that can facilitate the creation of similar systems while promoting the application of these concepts.

**Keywords:** Information Visualization, Scientific Software, Scientific Visualization, Data Analysis



# Lista de Figuras

2.5	Quarteto de Anscombe . . . . .	10
2.6	Combinação de tabelas e gráficos . . . . .	12
2.7	Quadro de temperaturas . . . . .	12
2.8	Os mesmos dados apresentados em duas tabelas, com a primeira não fazendo um bom uso dos elementos de separação e realce. . . . .	13
2.9	Exemplo de Infográfico . . . . .	15
2.10	Gráfico de barras apresentando melhor os mesmos dados que um gráfico de pizza. . . . .	17
2.11	Ilusão de ótica - A torre da direita aparenta ser mais inclinada para maioria das pessoas, mas as fotos são idênticas. . . . .	17
2.12	Ilusão de ótica - A maioria das pessoas não percebe que as fotos são idênticas. . . . .	17
2.13	As marcas de escala menores nos eixos Y facilitam a compreensão da escala logarítmica. . . . .	18
2.14	Exemplo diagrama de Gantt. . . . .	19
2.15	EventFlow . . . . .	20
2.16	Groove . . . . .	21
2.17	ThemeRiver . . . . .	21
3.1	Argus - Zoom . . . . .	26

3.2	Argus - Visualizando múltiplos gráficos . . . . .	27
3.3	Argus - Uso do "crosshair" . . . . .	28
3.4	Argus - Curva de tendência e exclusão de pontos . . . . .	28
3.5	Argus - Curva de tendência e exclusão de pontos . . . . .	29
3.6	Argus - Criando o Relatório . . . . .	29
3.7	Argus - Relatório Criado . . . . .	30
3.8	SAGAS - Gráficos com diferentes escalas . . . . .	32
3.9	Fouling - Variáveis e seus apelidos . . . . .	34
3.10	Fouling - configurando o eixo Y . . . . .	34
3.11	Fouling - Sem múltiplos eixos . . . . .	35
3.12	Fouling - Com múltiplos eixos . . . . .	35
3.13	Fouling - Sem média móvel . . . . .	36
3.14	Fouling - Com média móvel . . . . .	36
3.15	Fouling - diagrama de unidade . . . . .	37
3.16	Fouling - Árvore de navegação . . . . .	38
3.17	Estrutura de abas do sistema Revestir . . . . .	39
3.19	Strita - Pontos selecionados em duas camadas . . . . .	43
4.1	Bret1 . . . . .	51
4.2	Bret2 . . . . .	51
4.3	Bret3 . . . . .	51
4.4	Bret4 . . . . .	52

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	1
1.2	Visualização de Informação . . . . .	1
1.3	Estrutura . . . . .	4
<b>2</b>	<b>Conceitos e Elementos em Sistemas de Informação</b>	<b>5</b>
2.1	Conceitos Básicos . . . . .	5
2.1.1	Percepção . . . . .	5
2.1.2	Cores . . . . .	7
2.1.3	Quarteto de Anscombe . . . . .	9
2.2	Elementos Clássicos . . . . .	11
2.2.1	Texto, Tabelas e Gráficos . . . . .	11
2.2.2	Tabelas . . . . .	12
2.2.3	Infográficos . . . . .	14
2.2.4	Gráficos . . . . .	16
2.2.5	Gráficos de Dados Temporais . . . . .	19
2.2.6	Animações . . . . .	22
2.2.7	Mapas Georreferenciados . . . . .	23
<b>3</b>	<b>Sistemas de Software Científico</b>	<b>25</b>

3.1	Casos de Estudo . . . . .	25
3.1.1	ARGUS . . . . .	25
3.1.2	SAGAS . . . . .	31
3.1.3	Fouling . . . . .	33
3.1.4	Revestir . . . . .	38
3.1.5	STRITA . . . . .	42
<b>4</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>45</b>
4.1	Conclusões . . . . .	45
4.2	Novos Componentes . . . . .	46
4.2.1	Componente para Combinação de Tabela e Gráfico . . . . .	46
4.2.2	Componente para Criação de Infográficos . . . . .	47
4.2.3	Componentes para Análise em Animações de Gráficos e Vídeos .	48
	<b>Referências Bibliográficas</b>	<b>53</b>
<b>A</b>	<b>Tecnologias</b>	<b>57</b>
A.1	Python e JavaScript . . . . .	57
A.2	Plotagens - Flot . . . . .	58
A.3	Mapas e Infográficos - SVG . . . . .	59
A.4	Tabelas - Slickgrid e jqGrid . . . . .	61
A.5	Documentação - Ipython Notebook . . . . .	62
A.6	Relatórios . . . . .	62



# Capítulo 1

## Introdução

### 1.1 Objetivos

Este trabalho tem como objetivo realizar uma breve revisão de literatura de visualização de informação e destacar a aplicação destes conceitos em alguns sistemas desenvolvidos em parceria com a Gerência de Tecnologia de Informação para o Centro de Pesquisas da Petrobras. Para este estudo, foram analisados cinco sistemas elogiados e reconhecidos como destaque pela empresa. Nas conclusões, no final deste trabalho, apresentaremos uma breve reflexão sobre os benefícios da aplicação de conceitos de visualização de informação nestes sistemas.

### 1.2 Visualização de Informação

Em seu livro Ward et al. (2010) o professor Daniel Keim define a visualização de informação como a comunicação de representações gráficas para criação de um modelo ou imagem mental. O potencial oferecido pela visualização de informação passou por um enorme crescimento nos últimos vinte anos, em função do aperfeiçoamento dos computadores pessoais. Ainda assim, sua essência pode ser observada em exemplos

muito anteriores à invenção da computação e considerados clássicos. A visualização de informação está incorporada à vida diária das pessoas em revistas, jornais, mapas, sinalização, análises de investimentos, diagnósticos, entre outras atividades.

Por sua ubiquidade, a visualização de informação se tornou uma ferramenta eficaz para facilitar a análise e comunicação de praticamente todos os domínios do conhecimento. O estudo da visualização de informação é crítico para correta veiculação da mensagem a ser comunicada, uma vez que a visualização tem a capacidade de melhorar o processo decisório, possui potencial didático e pode até minimizar a consolidação de conceitos falsos, como o Hans Rosling costuma demonstrar em suas apresentações com sua ferramenta GapMinder (Rosling (2014)).

O uso adequado da visualização de informação no projeto de sistemas científicos, i.e, sistemas voltados para matemática, ciências e engenharias, possui peculiaridades que podem se tornar um desafio para o desenvolvedor. Sistemas científicos são sistemas complexos que envolvem estruturas de dados específicas e geralmente possuem como objetivo serem utilizados por várias pessoas ao longo de todas as etapas do processo de visualização, descritas por Ben Fry em seu livro Fry (2008), ao longo de um ciclo de vida que tende a ser muito maior que o de uma ferramenta de escritório. Tais projetos possuem em sua previsão mais de dez anos de duração apenas para implementação e com tempo de manutenção muitas vezes ilimitados.

Essa abrangência demanda o projeto de interfaces gráficas que agrupam em uma mesma tela diversos componentes como gráficos, planilhas, mapas e animações cuja aplicação eficiente é extremamente influenciada pela visualização de informação. Cabe ao desenvolvedor projetar adequadamente cada uma dessas funcionalidades, mantendo em mente que os pesquisadores serão os executores da visualização.

Dessa forma, o desenvolvedor não possui pleno controle de como a aplicação será utilizada, gerando considerações adicionais na implementação da funcionalidade de visualização. A possibilidade da visualização não ser feita em sua totalidade pela ferra-

menta sendo desenvolvida precisa ser considerada. Como veremos em alguns exemplos, o projeto de interfaces capazes de disponibilizar os dados, fornecendo meios para exportação dos mesmos em diversos formatos é muito importante. Esta flexibilidade ajuda ao pesquisador, que pode preferir desenvolver suas próprias soluções de visualização e este também pode fornecer concepções de novas formas de visualização que poderão ser incorporadas aos sistemas em versões futuras.

Como a visualização adequada depende do contexto e o desenvolvedor não é o ator final em sua execução, surgem questões difíceis de se responder em relação ao projeto destes sistemas: O pesquisador vai conseguir usar adequadamente essas funcionalidades? Este conjunto é eficiente ou na verdade uma fonte de distração? Ele vai usar a funcionalidade correta para cada etapa de seu trabalho? A disposição do conjunto favorece isso? São alguns dos questionamentos lançados como desafio no desenvolvimento destes sistemas.

A base para desenvolver um projeto que atenda razoavelmente bem a esses questionamentos demanda um conhecimento mais aprofundado do processo de trabalho dos pesquisadores aos quais a ferramenta se destina. Para projetar uma ferramenta capaz de transmitir a mensagem desejada, é preciso ter alguma noção do conteúdo dessa mensagem. Além disso, também é necessário um conhecimento dos princípios básicos de visualização de informação, que serão aplicados conjuntamente ao longo do processo de desenvolvimento.

Entretanto, encontramos usualmente em literatura de visualização de informação a descrição isolada da aplicação de um conceito de visualização de informação em um componente visual em particular, como por exemplo, um mapa, tabela, gráfico e não a aplicação integrada de vários conceitos em sistemas reais.

Este trabalho almeja contribuir mostrando a aplicação integrada desses conceitos em sistemas científicos desenvolvidos para o Centro de Pesquisas da Petrobras. Adotaremos a abordagem de observar e destacar a utilização dos princípios de visualização nestes



sistemas.

Esses sistemas foram reconhecidos como casos de destaque pela empresa e são elogiados por seus clientes, porém o presente trabalho não inclui estudos formais sobre a usabilidade com os usuários.

## 1.3 Estrutura

No segundo capítulo desta dissertação apresentamos uma revisão de literatura de visualização de informação, destacando alguns conceitos utilizados na melhoria destes sistemas, focando em conceitos básicos. A literatura recente de visualização de informação tem se ampliado diariamente e não pretendemos fazer uma revisão completa da mesma e sim esboçar os conceitos estruturantes.

O terceiro capítulo apresenta a maior contribuição deste trabalho fornecendo uma breve descrição dos sistemas desenvolvidos e das soluções de visualização neles implementadas de forma integrada. Elementos diferentes normalmente apresentados em oposição como por exemplo gráficos e tabelas são apresentados em conjunto. Algumas dessas soluções são aperfeiçoamentos e combinações de soluções tradicionais, enquanto outras são abordagens mais inovadoras.

Por último, No quarto capítulo é feita uma reflexão sobre os casos estudados, sobre padrões que se repetiram e apresentamos propostas de componentes que podem ser desenvolvidos a partir das soluções descritas para facilitar a produção de sistemas de software científico e promover a aplicação dos conceitos de visualização de informação abordados.

O apêndice é composto por uma descrição técnica da implementação das funcionalidades de visualização e das tecnologias utilizadas nos sistemas abordados.

## Capítulo 2

# Conceitos e Elementos em Sistemas de Informação

### 2.1 Conceitos Básicos

#### 2.1.1 Percepção

De acordo com Few (2012), Ward et al. (2010), Healey et al. (2008) e Ware (2004) o canal de comunicação utilizado no contexto de visualização de informação é o meio visual, canal de recepção de informação ao qual são dedicados 70% dos receptores de sentido do corpo humano. O processamento dessa informação é feito em diversas etapas que sumarizam e abstraem a mensagem a ser entendida. Estima-se que apenas 10% da informação percebida pelos olhos seja transmitida da retina ao cérebro pelo nervo ótico e no cérebro ela é primeiramente processada rapidamente em níveis automáticos e inconscientes.

As informações processadas pelos olhos são enviadas para memória icônica do cérebro pelo nervo óptico e ficam ali por menos de um segundo enquanto uma interpretação extremamente rápida acontece. Esse processamento, automático, inconsciente e limitado

apenas a reconhecimento, é chamado de processamento de pré-atenção (pre-attentive). Ele detecta um conjunto limitado de atributos, como cor, posição e forma e tem um papel importante na produção visual - para se destacar um grupo de objetos, é recomendado usar atributos de pré-atenção.

Porém, muitos desses atributos não permitem ou são limitados em termos de percepção quantitativa, sendo a posição e o comprimento mais adequados. Atributos visuais como cor e tamanho não são compreendidos pelo cérebro como valores absolutos mas como diferenciais dependentes de contexto, podendo causar confusões e até mesmo ilusões de ótica como vemos nas figuras 2.1, 2.11 e 2.12, efeitos que devem ser evitados na produção de comunicação visual.

A informação visual segue da memória icônica para memória de curto prazo, onde o cérebro combina o que considera útil em agrupamentos visuais, em um processo consciente de atenção. O cérebro só consegue interpretar um número pequeno de agrupamentos por vez e que para processar mais grupos, outros precisam ser esquecidos ou movidos para memória de longo prazo.

Essa limitação deve ser considerada na prática durante o projeto de interfaces. Por exemplo, se um gráfico apresentar dez séries de dados utilizando cores ou símbolos diferentes para cada uma delas, é preciso considerar que o leitor vai precisar de várias consultas a legenda do gráfico para compreensão do mesmo.

Baumeister e Tierney (2011) ressaltam a importância de tentar evitar essa situação e manter a percepção da informação no nível inconsciente, pois isso facilita a leitura e aumenta a chance de que a transmissão da mensagem seja concretizada corretamente. Níveis mais elevados de consciência são recursos limitados que podem não estar disponíveis no momento da leitura da informação. A resposta a essa indisponibilidade pode variar da mera recusa consciente à leitura ou no pior caso, à substituição inconsciente da mensagem por estereótipos e pré-concepções.

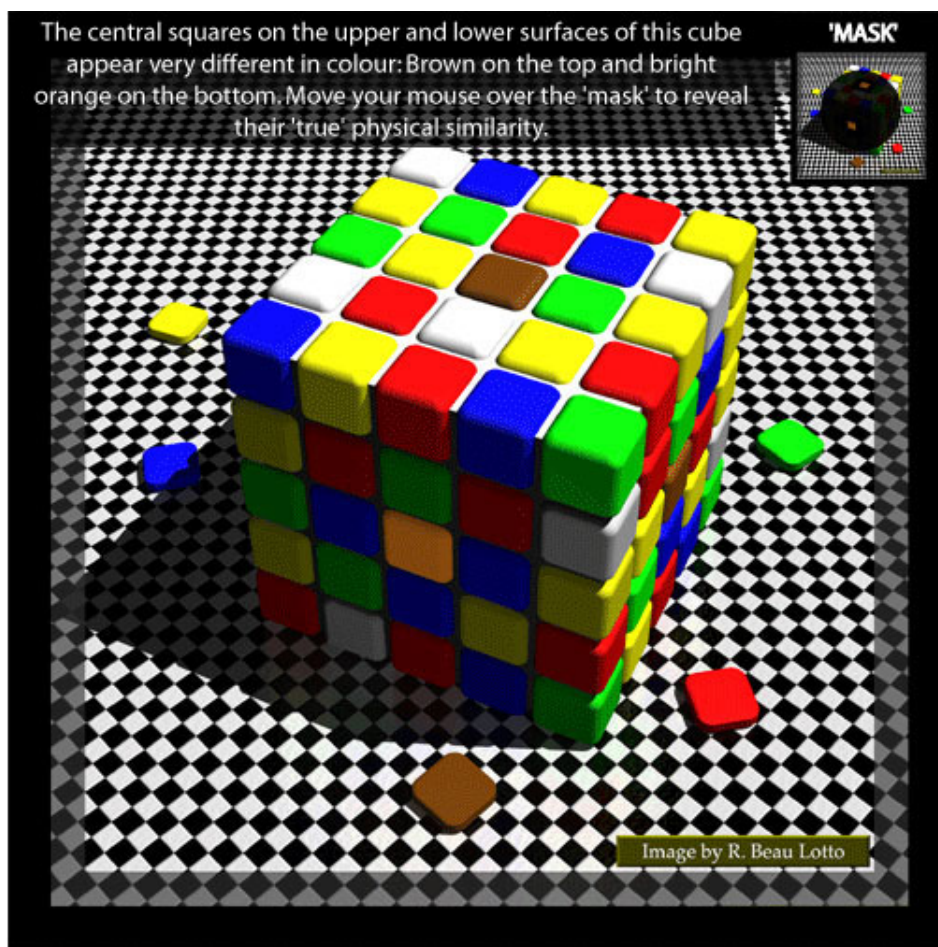


Figura 2.1: Ilusão de cor causada pelo contraste

### 2.1.2 Cores

Lynn (2014) descreve como a combinação de cores utilizadas na produção de uma visualização pode passar despercebida quando é agradável, mas pode ser bastante incômoda quando mal selecionada. As cores utilizadas na visualização devem ter um relacionamento entre si que promova equilíbrio e harmonia visual. As combinações de cores mais utilizadas em geral se relacionam entre si na forma de variações de relacionamentos monocromáticos, complementares ou triádicos em relação ao círculo cromático. No relacionamento monocromático (ver figura 2.2), as cores apresentam

variações de intensidade de luminância do mesmo tom.



Figura 2.2: Relacionamento Monocromático

No relacionamento complementar (ver figura 2.3), as cores utilizadas se posicionam em extremos opostos no círculo cromático. Essa disposição aumenta o contraste entre as cores, a luminância das mesmas deve ser baixa.



Figura 2.3: Relacionamento Complementar

No relacionamento triádico 2.4, cores de três tonalidades equidistantes no círculo cromático são selecionadas.

Como afirma Few (2012), qualquer que seja o relacionamento entre as cores escolhidas para o projeto, deve ser evitado o uso de cores totalmente saturadas, pois elas aumentam muito o contraste total da visualização.

O contraste deve ser reservado para facilitar a leitura de pontos críticos na visualização, como por exemplo, nas informações em formato de texto, que deve ter pelo menos 80 por cento de contraste em relação ao fundo.

Existem diversas aplicações na Web para auxiliar a escolha de uma combinação de cores para projetos de interface, como por exemplo a Kuler da Adobe Adobe (2014),



Figura 2.4: Relacionamento Triádico

o ColorScheme (2014) e o COLRD (2014). Algumas delas como a Kuler disponibilizam paletas pré-definidas, que podem ser muito úteis. A COLRD mostra as paletas que foram aplicadas em fotografias, o que pode ser muito útil para melhorar a compreensão desses conceitos.

### 2.1.3 Quarteto de Anscombe

Quarteto de Anscombe é o nome dado a quatro conjuntos de dados que aparentam ser idênticos quando descritos por certas técnicas de estatística descritiva (como a média e a variância), mas que são muito distintos quando exibidos graficamente. Ele leva o nome do estatístico F.J. Anscombe que o publicou pela primeira vez em 1973, com o objetivo de demonstrar a importância de se visualizar os dados graficamente antes de analisá-los e o efeito de valores muito discrepantes nas propriedades estatísticas. Anscombe (1973)

Na linguagem R (ver RProject (2014)), basta digitar "anscombe" para reproduzir o quarteto original:

```
> anscombe
      x1 x2 x3 x4      y1      y2      y3      y4
1   10 10 10  8   8.04  9.14   7.46   6.58
2    8  8  8  8   6.95  8.14   6.77   5.76
3   13 13 13  8   7.58  8.74  12.74   7.71
```

4	9	9	9	8	8.81	8.77	7.11	8.84
5	11	11	11	8	8.33	9.26	7.81	8.47
6	14	14	14	8	9.96	8.10	8.84	7.04
7	6	6	6	8	7.24	6.13	6.08	5.25
8	4	4	4	19	4.26	3.10	5.39	12.50
9	12	12	12	8	10.84	9.13	8.15	5.56
10	7	7	7	8	4.82	7.26	6.42	7.91
11	5	5	5	8	5.68	4.74	5.73	6.89

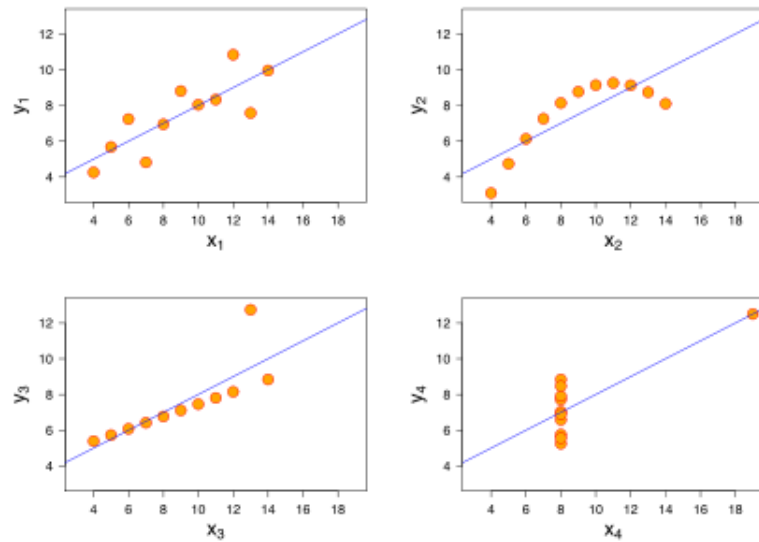


Figura 2.5: Quarteto de Anscombe

As séries 1, 2, 3 e 4 mostradas graficamente na figura 2.5 possuem média 9, variância 11. A média dos  $y$ 's nas séries é 7.5 e a variância é 4.125. Em todas elas o coeficiente de correlação igual a 0.816 e a curva de regressão linear  $y = 3 + 0.5x$ .

O gráfico da série  $X_1$  mostra uma relação linear normal, o da série  $X_2$  mostra uma clara relação, mas não linear. O gráfico da série  $X_3$  mostra uma clara relação linear, mas ela é deslocada pelo ponto discrepante. O gráfico da série  $X_4$  mostra uma relação clara, mas um dado discrepante desloca a curva de regressão linear em 45 graus apesar

das 10 outras observações.

## 2.2 Elementos Clássicos

### 2.2.1 Texto, Tabelas e Gráficos

De acordo com Few (2012) o uso de tabelas e gráficos é comum atualmente, mas poucos aprenderam como torná-los eficazes, inclusive porque esses recursos só passaram a ser produzidos por um maior número de pessoas após a criação do computador pessoal no final do século 20.

Quando a informação consiste apenas poucos números a linguagem escrita é mais eficaz em termos de comunicação. Uma tabela simples bem projetada, pode fornecer melhor entendimento e mais informação que um gráfico mal feito, apesar de todo apelo e facilidades perceptivas do meio visual. Existe elegância na simplicidade. Por exemplo, na figura 2.6 temos uma combinação de tabela e gráficos das últimas 10 temperaturas medidas em quatro setores de um centro de processamento de dados, sendo apenas a última medição relevante. Compare com a figura 2.7. Além de refletir a disposição geográfica dos setores no centro de processamento de dados, as temperaturas fora dos limites críticos passaram a ser marcadas em vermelho.

Em informações quantitativas a mensagem sempre está ligada a alguma coisa relevante sendo medida e esse relacionamento deve ser expresso claramente. Tabelas precisam ter cabeçalhos, gráficos precisam ter legendas. Informações quantitativas podem expressar alguma agregação ou sumarização de um conjunto grande de números. Nesses casos, gráficos são recursos indicados, facilitando bastante a comunicação.

Segundo Tufte (1983), as práticas gerais de projeto orientado para comunicação buscam dois objetivos fundamentais: destacar e organizar os dados. O destaque adequado aos dados pode ser obtido utilizando o conceito de remover elementos desnecessários, evitar o destaque aos elementos visuais estruturais alheios aos dados e se concentrar



Horários	1	2	3	4	5	6	7	8	9	10	Gráfico
Temperatura Área 1	29.5	24.7	30.4	30.4	22.5	22.5	22.5	22.5	22.5	22.5	
Temperatura Área 2	24.2	30.7	21.3	30.7	24.8	24.8	24.8	24.8	24.8	24.8	
Temperatura Área 3	24.1	28.7	24.1	24.8	27.3	27.3	27.3	27.3	27.3	27.3	
Temperatura Área 4	25.7	21.7	28.4	28.3	20.2	20.2	20.2	20.2	20.2	20.2	

Figura 2.6: Combinação de tabelas e gráficos

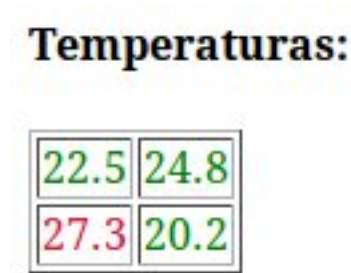


Figura 2.7: Quadro de temperaturas

nos dados. Esse conceito é aplicável tanto a gráficos e tabelas.

Tabelas podem ser usadas sempre que uma relação existir entre valores de conjuntos separados, pois permitem alinhar valores relacionados, se estes forem posicionados na mesma linha ou coluna, facilitando a comparação. Também facilitam a busca de valores individuais e são mais precisas que gráficos.

Gráficos, por sua vez, são mais eficazes quando a mensagem está relacionada ao que é expresso pelo conjunto completo dos dados, como por exemplo, sua tendência no tempo. Ressaltam as exceções e facilitam a visualização de similaridades e diferenças entre conjuntos de dados, pois dão forma a eles.

### 2.2.2 Tabelas

No projeto de tabelas, é uma boa prática de visualização usar o mínimo de separadores possível e deixar que a própria estrutura do texto e o espaçamento (espaço em branco) moldem a tabela. Quando o espaço da tabela é muito pequeno, isso pode se tornar inviável e então grades e separadores devem ser usados, mas com cautela. Ver Tufte

(1983) e Few (2012)

Esses elementos, além das cores e fontes do texto, podem ser usados para enfatizar as partes importantes dos dados. Ao lidar com dados numéricos em tabelas, é importante levar em conta o alinhamento do texto e a formação numérica (truncagem, notação científica, número de casas decimais, percentuais). Destacar agregações como totais e médias facilita a leitura.

A figura 2.8 mostra na parte superior uma tabela que apresenta alguns problemas de visualização. Neste caso, o uso de separadores é visualmente agressivo e a formatação dos campos numéricos à esquerda torna a leitura mais difícil. Saldos negativos não são marcados por cor ou negrito. Além disso, a totalização é apresentada da mesma maneira que as outras linhas, sem distinção. Abaixo, temos uma sua versão melhorada da mesma informação, que se aproveita da disponibilidade de mais espaço para abrir mão dos separadores e usa cores para dar maior ênfase na agregação e nos saldos negativos.

Mês	Despesas	Ganhos	Caixa
Janeiro	201,21	100,21	-101,0
Fevereiro	80,32	100,21	19,98
Março	92,21	100,21	8
Abril	44,33	100,21	55,88
Total	418,07	400,84	-17,23

Mês	Despesas	Ganhos	Saldo
Janeiro	201,21	100,21	-101,00
Fevereiro	80,32	100,21	19,98
Março	92,21	100,21	8,00
Abril	44,33	100,21	55,88
<b>Total</b>	418,07	400,84	-17,23

Figura 2.8: Os mesmos dados apresentados em duas tabelas, com a primeira não fazendo um bom uso dos elementos de separação e realce.

Tabelas são amplamente utilizadas em sistemas científicos. A partir da definição

básica de tabela do padrão HTML, muitas adaptações podem ser feitas, desde recursos básicos de visualização, como alternância de cores nas linhas a implementações mais elaboradas, como o uso de fontes coloridas para destaque de erros e pontos de atenção nos dados.

Dentre os recursos disponíveis estão a possibilidade de reordenação das tabelas a partir de alguma coluna selecionada ou para tabelas maiores uma barra de rolagem mantendo o cabeçalho fixo, a edição de dados e o autocompletar de campos.

### 2.2.3 Infográficos

Infográficos são representações visuais de informações criados com o objetivo de apresentar informações complexas de forma mais rápida e clara. Recentemente, com a proliferação de ferramentas gratuitas e fáceis de usar tornaram a sua criação também acessível para um grande segmento da população. Nos jornais, infográficos são geralmente usados para mostrar o clima e ilustram as notícias de maior destaque. Alguns livros são quase inteiramente compostos por infográficos como Macaulay (1988). A figura 2.9 mostra um exemplo de infográfico.

Edward Tufte define ' Graphical Displays ' que devem :

- mostrar os dados;
- induzir o espectador a pensar sobre a substância, em vez da metodologia, design gráfico, tecnologia de produção gráfica, etc;
- evitar distorcer o que os dados têm a dizer;
- apresentam muitos números em um pequeno espaço;
- tornar grandes conjuntos de dados coerentes;
- incentivar o olho a comparar diferentes partes dos dados;

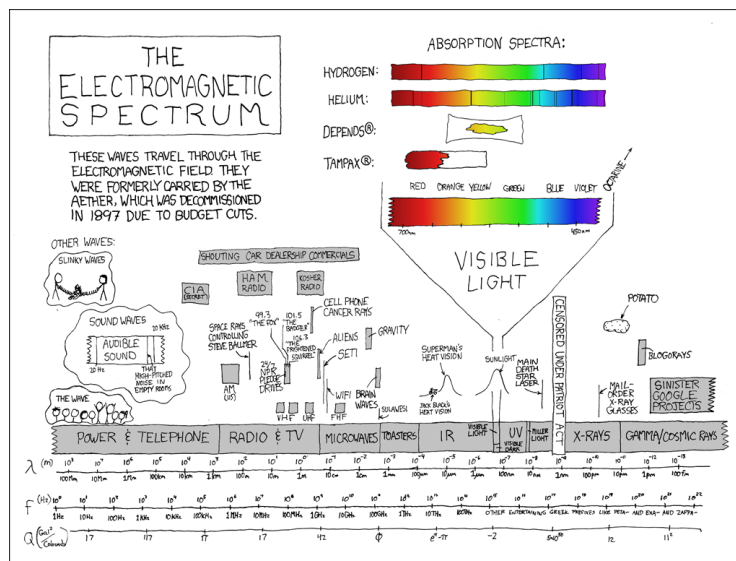


Figura 2.9: Exemplo de Infográfico

- revelar os dados em vários níveis de detalhe, a partir de uma visão mais ampla da estrutura.
- servir a um propósito razoavelmente claro: descrição , exploração , tabulação ou decoração.
- estar estreitamente integrado com as descrições estatísticas e verbais de um conjunto de dados.

Gráficos revelam dados. Na verdade os gráficos podem ser mais precisos e reveladores que os cálculos estatísticos convencionais.

Os infográficos contemporâneos muitas vezes tratam de assuntos qualitativos ou mais abstratos, mas em geral a definição de Tufte ainda se aplica, em um sentido amplo, ao que os infográficos são e ao que eles fazem, que é condensar grandes quantidades de informações em uma forma que possam ser mais facilmente absorvidos.

## 2.2.4 Gráficos

Stephen Few analisa em seu livro "Show Me The Numbers"(ver Few (2012)) a utilidade e a adequação do uso de gráficos clássicos como o gráfico de pizza, barras, etc. Algumas das conclusões de suas análises serão resumidas aqui.

O autor argumenta que diferentes tipos de relacionamentos quantitativos demandam diferentes formas de gráficos. Dentre as formas mais comuns temos os gráficos de pontos, linhas, barras e de área.

O gráfico de área mais conhecido é o gráfico de pizza. Esse tipo de gráfico causa problemas de percepção e tem seu uso não recomendado. Mesmo quando for crucial manter em foco a visão das partes em relação ao todo, gráficos de pilha ("stacked bar") são uma melhor alternativa. A figura 2.10 mostra os mesmos dados de um gráfico de pizza melhor apresentados e com um uso melhor das cores em um gráfico de barras. Ele foi produzido de forma que leva as pessoas a não perceberem corretamente, quando comparadas, as quantidades de Maçã e Morangos, e as quantidades de Pêras e Bananas.

Gráficos devem atender o objetivo de corretamente fazer a correspondência das quantidades com a escala visual. Este objetivo, por si mesmo conflita com o uso de efeitos 3D estéticos em gráficos. Os atributos de comprimento de linha e posicionamento 2D fornecem boa percepção visual das quantidades e devem ser usados nos gráficos. Outros atributos, como cor e área não permitem a percepção adequada de quantidades.

O site "Atirei o Pau no Gráfico"apresenta vários exemplos onde esses recursos foram mal utilizados e apresenta alternativas melhores. Ver ANOVA (2014)

Para evitar erros de interpretação, os gráficos devem preferencialmente usar a mesma escala entre os eixos e essas escalas devem começar do zero. Quando isso não for adequado para visualização dos dados, deve-se alertar sobre a diferença. As marcas nos eixos (marcas de escala) devem ser consistentes, sem ausências mesmo em intervalos que não contenham dados.

Os valores podem ser representados nos gráficos através de pontos, barras ou linhas.

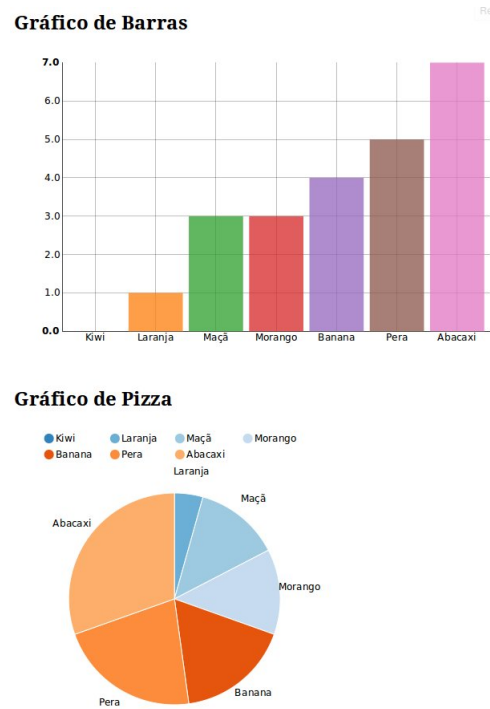


Figura 2.10: Gráfico de barras apresentando melhor os mesmos dados que um gráfico de pizza.

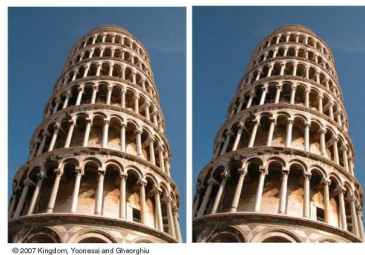


Figura 2.11: Ilusão de ótica - A torre da direita aparenta ser mais inclinada para maioria das pessoas, mas as fotos são idênticas.



Figura 2.12: Ilusão de ótica - A maioria das pessoas não percebe que as fotos são idênticas.

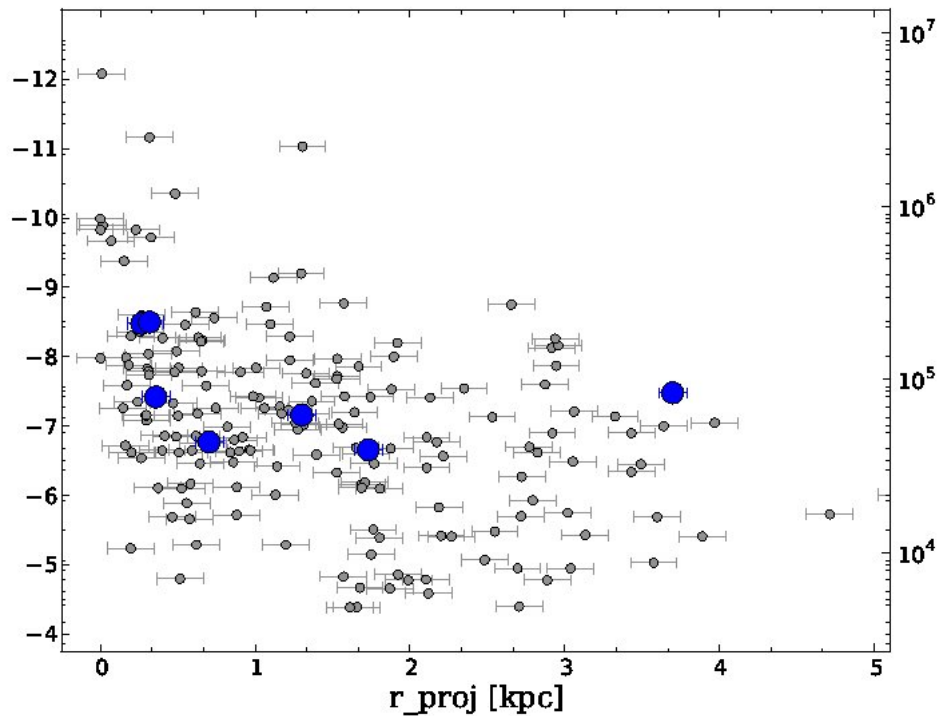


Figura 2.13: As marcas de escala menores nos eixos Y facilitam a compreensão da escala logarítmica.

Quando pontos são usados para representar as quantidades, eles devem ser visualmente distintos. Símbolos e cores diferentes ajudam a manter essa distinção.

Rótulos, legendas e marcas de escala facilitam a interpretação correta dos dados e por isso fornecem informação crucial, mas devem ser visualmente menos destacados que os valores.

A quantidade de marcas de escala deve ser suficiente para determinar os valores dos dados dispostos entre elas. Marcas de escala menores normalmente sugerem um nível de precisão que o gráfico não possui e só são úteis em escalas logarítmicas. A figura 2.13 mostra um exemplo de marcas de escala menores bem utilizadas.

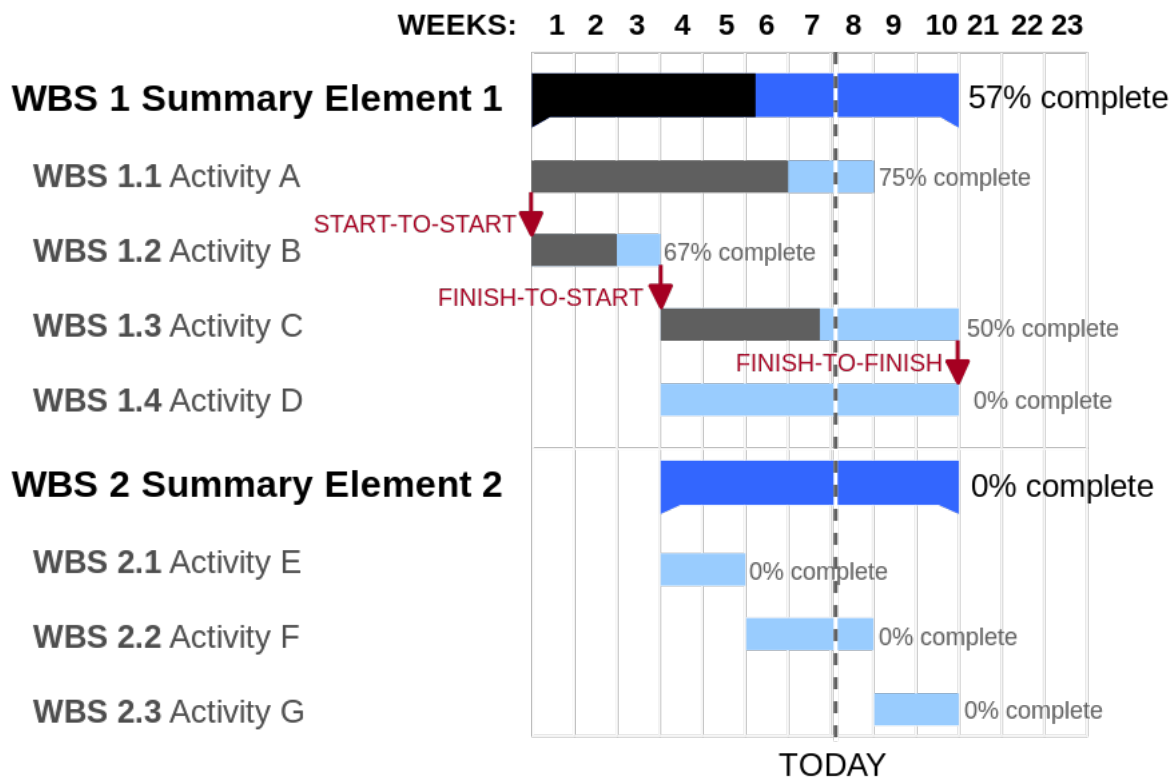


Figura 2.14: Exemplo diagrama de Gantt.

### 2.2.5 Gráficos de Dados Temporais

A visualização de dados temporais é feita tradicionalmente de acordo com a natureza dos mesmos. Dados categóricos, como início e fim de fases são em geral tratados com ferramentas como os diagramas de Gantt (ver figura 2.14 ) Wilson (2003). Sequências de dados temporais com várias medições são em geral visualizados como séries temporais.

Os diagramas de Gantt foram desenvolvido em 1917 pelo engenheiro mecânico Henry Gantt como uma ferramenta de controle de produção. Nos diagramas de Gantt dados categóricos como os intervalos de tempo representando o início e fim de etapas de um projeto aparecem como barras sobre o eixo horizontal. O EventFlow (ver figura 2.15) Megan Monroe e Wongsuphasawat (2014) é outra uma ferramenta de visualização para dados temporais categóricos que implementa técnicas como o diagram de Gantt.



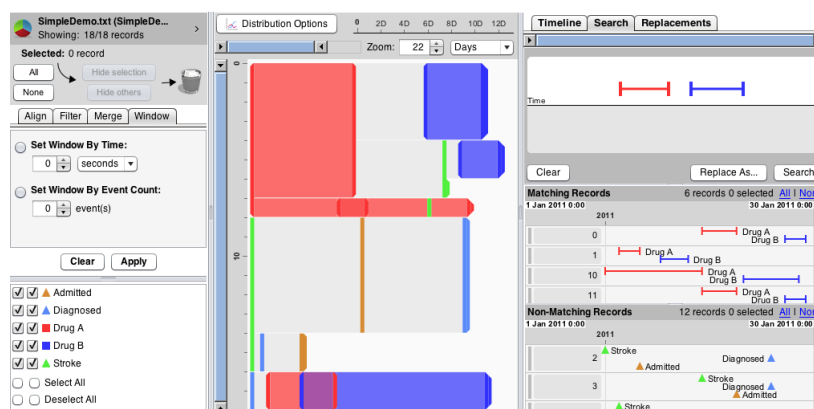


Figura 2.15: EventFlow

Séries temporais consistem em uma sequência de pontos de dados, geralmente medidos em intervalos de tempos uniformes. São usados principalmente para fazer previsões. Segundo Tufte (1983) são o tipo de gráfico mais antigo e um dos mais utilizados mas detalhes como diferenças de formato de hora e data, a internacionalização e a variabilidade intervalo de tempo a ser mostrado tornam complexa a implementação de programas computacionais dedicados a produção desses gráficos.

Além dessa abordagem tradicional, existem outras ferramentas para visualização de dados de séries temporais como a ThemeRiver de Susan Havre (2014) e a GROOVE de T. Lammarsch (2014) (ver figuras 2.16 e 2.17).

Na ThemeRiver os dados são visualizados como variações temáticas ao longo do tempo. O "rio" flui a partir da esquerda para a direita através do tempo, onde a mudança de largura de cada série, apresentada como uma "corrente colorida", é usada para descrever as alterações na intensidade dos temas ao longo do tempo. A GROOVE é uma ferramenta de visualização que se beneficia das estruturas hierárquicas do tempo, como horas, dias, meses, estações, etc.

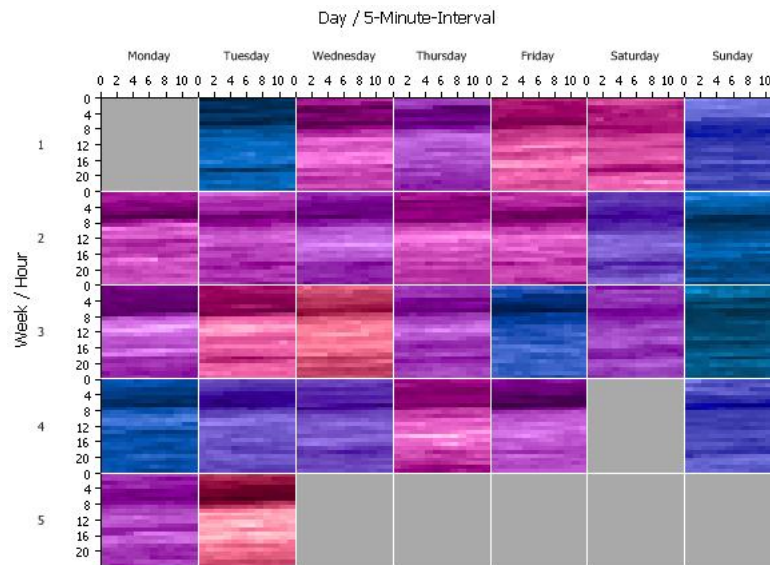


Figura 2.16: Groove

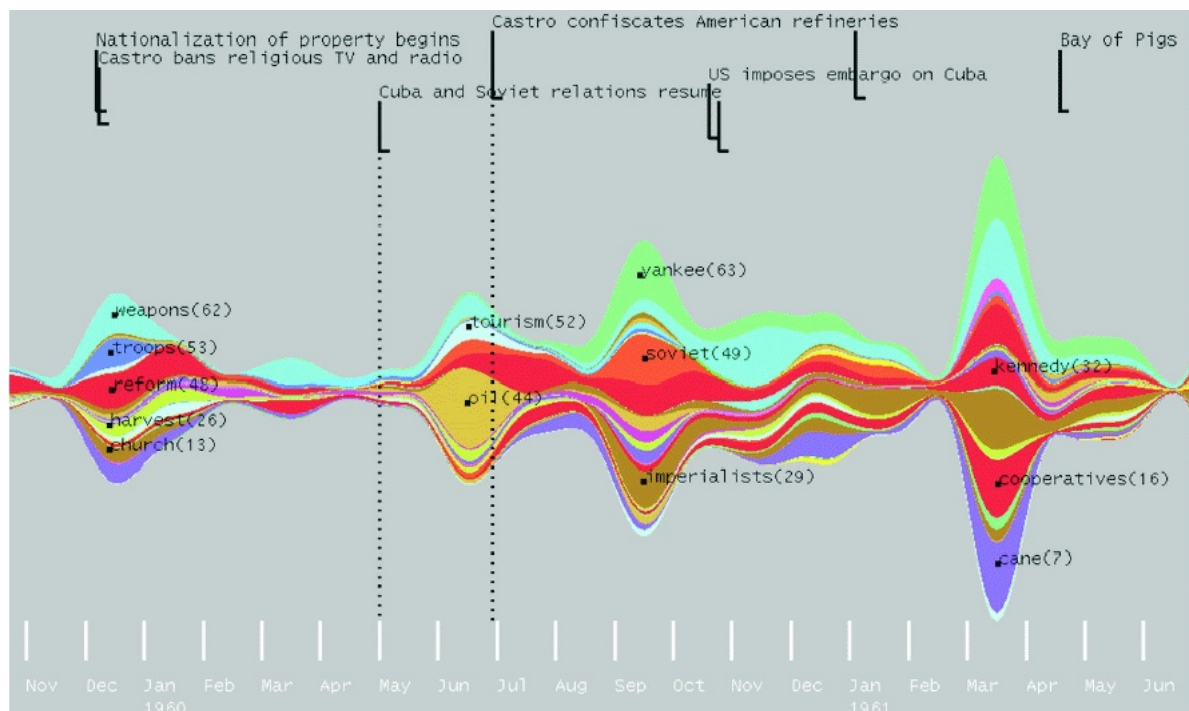


Figura 2.17: ThemeRiver

### 2.2.6 Animações

De acordo com Coltheart (1980), a exibição rápida de sequências de imagens estáticas com diferenças mínimas entre si gera a ilusão de movimento através da persistência da visão. Em sistemas interativos, as transformações que podem ser feitas podem criar esse efeito.

A criação de animações atualmente é bastante facilitada tanto pelas tecnologias quanto pela vasta literatura conceitual disponível sobre como produzi-las.

Animações podem ser uma técnica poderosa quando usadas apropriadamente ou muito ruim quando mal utilizadas. Podem aumentar o apelo visual, mas podem dificultar a exploração visual.

Podemos entender a animação como uma série de mudanças visuais entre os quadros. É fácil entender o que aconteceu quando o número de mudanças é pequeno e a animação fica mais complexa conforme as mudanças aumentam.

Cavanagh e Alvarez (2005) afirmam que objetos se movendo na mesma direção e velocidade serão percebidos como um único grupo. Objetos com suas próprias trajetórias se destacarão visualmente como separados. É difícil acompanhar mais de quatro ou cinco objetos separadamente. Os olhos passam a seguir apenas alguns e tratam os outros como ruído.

O livro de Steele e Iliinsky (2010) argumenta que a animação é bastante utilizada em visualização científica por partir de dados tridimensionais e ser muito difícil mostrar os processos ao longo do tempo de outra forma.

Em contraste, a visualização de informação normalmente lida com espaços de dados abstratos onde as dimensões não correspondem às daquelas do mundo real. Por isso, existe comparativamente muito menos exemplos de animações publicadas na comunidade de visualização de informação.

Em mais de cem estudos de visualização e animação, o uso de animação não apresentou melhor resultado quando comparado ao uso de diagramas estáticos. Em um

estudo de visualização de algoritmos, as animações se mostraram eficientes quando permitiam que o funcionamento dos algoritmos fosse manipulado e refletiam essas alterações. Quando assistidas passivamente as animações não se mostraram mais eficientes que outros métodos.

A visualização GapMinder do Hans Rosling é um caso de sucesso, mas apresenta muito mais informação que os estudos de percepção demonstram ser possível processar. A apresentação guiada de Rosling parece ser crucial para o sucesso da visualização.

Estudos no Microsoft Research (MSR) indicam que a animação é uma forma mais lenta e menos precisa de fornecer informação. Para explorar os dados por conta própria, os usuários precisam reexibir a animação várias vezes para verificar algum fato. Os usuários claramente desejam ter a capacidade de controlar a animação no tempo e isso pode torná-la menos eficiente que uma série de imagens estáticas, pois estas últimas permitem que se vá diretamente para a parte que se quer ver.

No entanto, estes estudos admitem que animações são emocionalmente poderosas e mais envolventes. No contexto de apresentação, os dados e a mensagem já são conhecidas pelo apresentador e o público somente assiste. A animação é apresentada diretamente, sem muitas voltas, ao contrário do que faria alguém explorando os dados. Nesse contexto, a animação é recomendada.

### 2.2.7 Mapas Georreferenciados

No contexto de visualização de informação, são diagrama esquemático que não mostram a posição geográfica exata, mas posições relativas dos elementos que precisam ser mostrados e suas ligações.

Segundo Meunier (2014), inspirados no mapa do Metropolitano de Londres (Museum (2014)), a concepção básica tem sido amplamente escolhida na representação de outras redes de todo o mundo, especialmente no mapeamento topológico e não geográfico. Ferramentas de software foram desenvolvidas para auxiliar a produção desse tipo de

mapa.

Steele e Iliinsky (2010) afirmam que o tamanho do mapa e como ele acomoda as informações que devem ser mostradas é um fator importante, mas não tão relevante quanto a forma como os dados são apresentados.

A geografia é o ponto de partida para o projeto de mapas e parte da experiência de quem vai usá-los. Ela precisa ser estilizada e simplificada, mas os relacionamentos dos pontos de referência devem ser preservados.

Além da geografia, as linhas devem denotar os caminhos a serem seguidos. Mais do que meramente serem usadas separadamente para facilitar a distinção das linhas, as cores podem ser organizadas em espectro para fornecer mais semântica ao mapa. O uso de cores é importante para tornar o mapa mais vasculhável, além de apenas mais legível.

Mapas podem envolver questões emocionais. Algumas cores devem ser preservadas, como o azul representando água. Da mesma forma, representar elementos marcantes por desenhos de suas formas tem mais apelo que rótulos em texto com seus nomes.

O texto também pode ser separado em tons de cinza, guardando o preto para as informações mais relevantes. Bem organizado, o mapa fornece camadas de informação sem comprometer sua clareza e funcionalidade.

## Capítulo 3

# Sistemas de Software Científico

### 3.1 Casos de Estudo

Neste capítulo descreveremos os sistemas que serviram como caso de estudo para esta dissertação. Serão abordadas as interfaces de cinco sistemas científicos. Em particular, nestes sistemas, suas interfaces foram produzidas em formato "WEB", apesar do modelo "desktop" ser o mais utilizado para esse tipo de sistema.

O formato WEB tende a evidenciar rapidamente, por comparação com os outros sites WEB, a inobservância de princípios básicos de visualização, ainda muito comuns em sistemas científicos em uso, como por exemplo o uso de cores com luminância alta sob fundo preto.

Os sistemas abordados nesse estudo foram desenvolvidos em momentos diferentes e apresentam diferentes níveis de maturidade no uso dos conceitos de visualização de informação, embora sempre com alguma noção a respeito do assunto.

#### 3.1.1 ARGUS

O sistema ARGUS é uma ferramenta padronizada e personalizada para a análise, diagnóstico e acompanhamento contínuo das unidades de refino.

Os dados manipulados por este sistema são fornecidos por sensores nas instalações industriais e podem ser visualizados como tabelas e gráficos interativos.

Um dos objetivos do sistema ARGUS é fornecer resultados das análises e previsões, conforme simulado pelo núcleo de cálculo, de dados derivados de medições de centenas de sensores de diversas instalações industriais.

Dessa forma, o sistema é utilizado para promover consultas e comparações dessas análises e medidas. Para isso, o sistema foi desenvolvido para permitir a plotagem de diversos tipos de gráficos e a manipulação interativa dos mesmos.

As soluções utilizadas para criação e manuseio desses gráficos e tabelas serão abordadas neste estudo.

O ARGUS não trabalha com a definição dos intervalos dos eixos, mas disponibiliza recursos de zoom para permitir o ajuste de escalas. Para orientar o usuário nesse processo, uma figura pequena com a imagem do gráfico completo, com a seção que está sendo exibida em destaque é mostrada. Um botão para restaurar a configuração original também foi criado. Ver figura 3.1

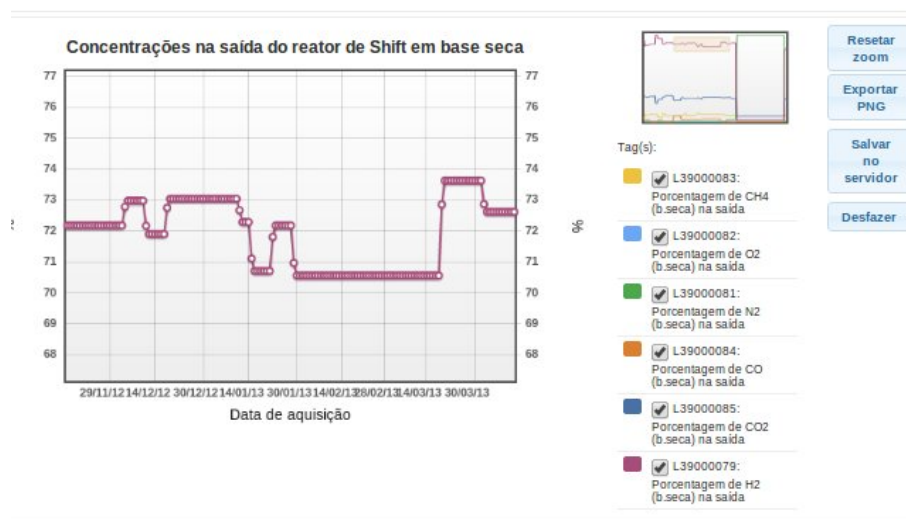


Figura 3.1: Argus - Zoom

A funcionalidade de exibição de gráficos com múltiplas escalas nos eixos também

não foi implementada e o sistema permite apenas o uso de no máximo duas escalas. Para promover a comparação de variáveis com grandezas muito diferentes o projeto do sistema optou pela exibição na mesma tela de vários gráficos, integrados por um recurso interativo de "crosshair", onde uma linha vermelha cruzando os gráficos acompanha o movimento do mouse. Ver figuras 3.2 e 3.3.

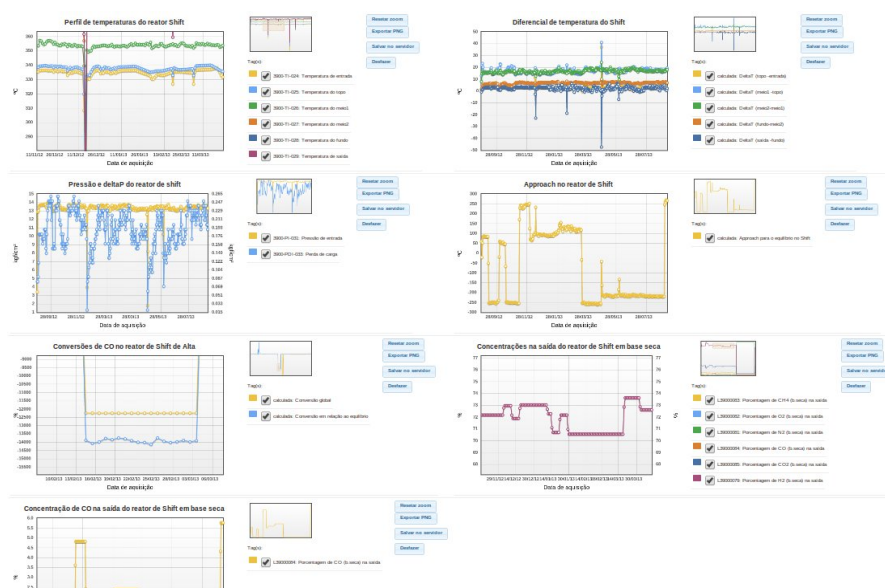


Figura 3.2: Argus - Visualizando múltiplos gráficos

Neste sistema também não foi implementada nenhuma suavização dos dados. Entretanto, foi desenvolvida uma funcionalidade para exibir curvas de tendências nos gráficos e um recurso interativo que permite a eliminação de pontos atípicos com um clique do mouse. Ver figuras 3.4 e 3.5.

Atendendo a filosofia de disponibilizar os dados, os gráficos e consultas podem ser exportados em formatos de imagem e planilha.

Além disso, o sistema permite a criação de relatórios a partir de uma convenção de códigos. O relatório é editado normalmente pelo usuário em um editor WYSIWYG, no caso o TineMCE (ver AB (2014)). No local onde se deseja um gráfico ou uma tabela, o cliente utiliza um código convencionado para determinar como os dados devem ser



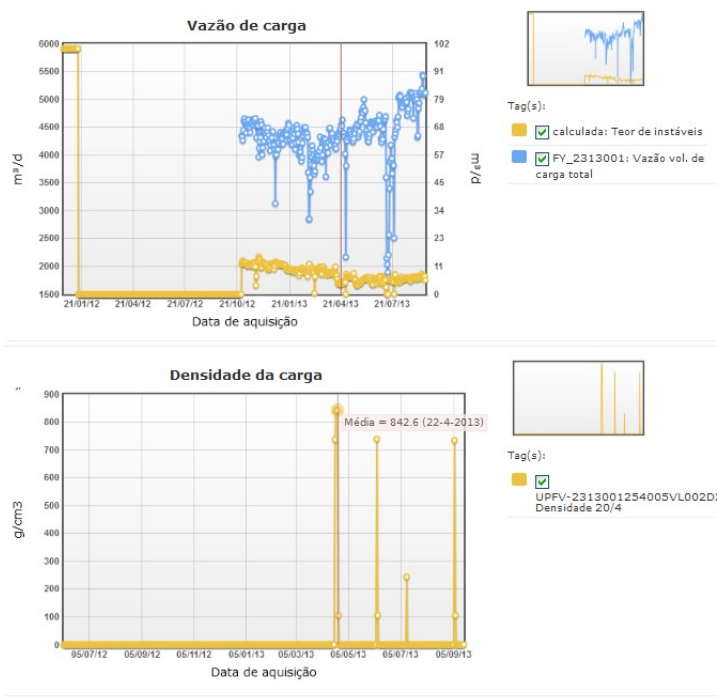


Figura 3.3: Argus - Uso do "crosshair"

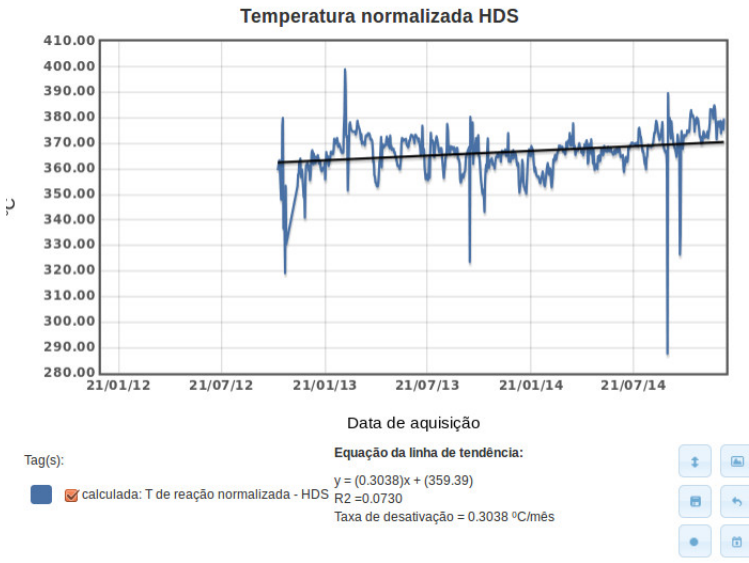


Figura 3.4: Argus - Curva de tendência e exclusão de pontos

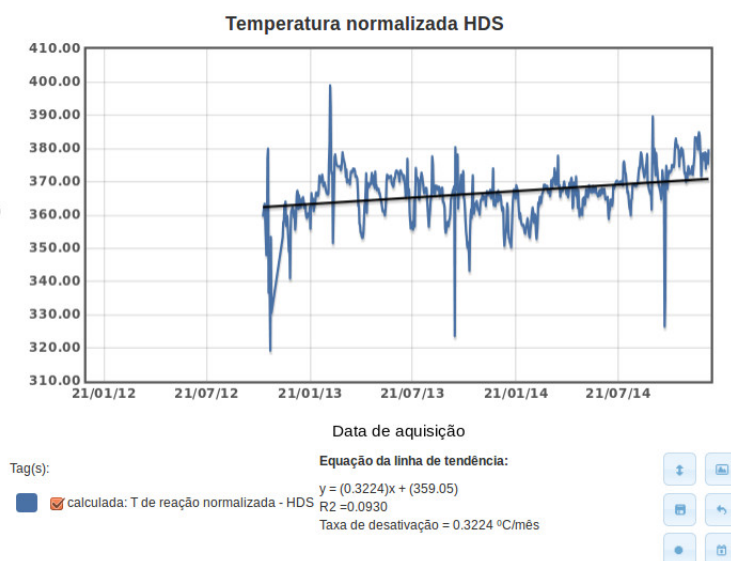


Figura 3.5: Argus - Curva de tendência e exclusão de pontos

apresentados.

A interface de criação de relatório no Argus. No topo, há campos para "Título:" (Teste com Gráficos e Tabelas) e "Salvo em:" (Data: 14/09/2013, Hora: 05:26:59). Abaixo, há uma barra de ferramentas com ícones para formatação de texto e inserção de elementos. O corpo principal do relatório contém:

**Este é um teste com Gráficos e tabela**

RELATÓRIO MENSAL DE PROCESSO		Logo do ARGUS
refinaria:		
GERÊNCIA:		
página:	XX de XX (colocar funcionalidade para automatizar)	

Abaixo da tabela, há campos para inserir gráficos e tabelas:

[[ form.graf1 ]] E [[ form.graf2 ]]

Na base, há um campo para inserir dados:

[[ form.dados1 ]]

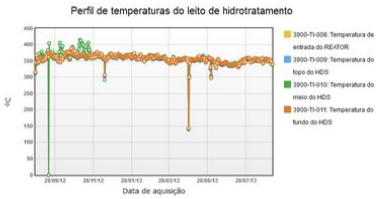
Figura 3.6: Argus - Criando o Relatório

Esse código é posteriormente substituído por caixas de lista, onde o cliente pode então determinar qual gráfico ou resultado deve aparecer ali. Na etapa final, de pré-visualização do relatório que será impresso, o cliente pode então ver seu relatório

completo. Todas as etapas de edição estão reunidas na mesma tela do sistema. Ver figuras 3.6 e 3.7

Este é um teste com Gráficos e tabela

RELATORIO MENSAL DE PROCESSO		Logo do ARGUS
refinaria:		
GERENCIA:		
pagina:	XX de XX (colocar funcionalidade para automatizar)	



E

Nome Seção	Teg	Descrição	Aquisição	Média	Mínimo	Máximo	Unidade Média	Decimais
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3912-12-30	324.1803552	192.4513548	611.6379580	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3912-12-31	509.4951155	293.5009080	827.2415436	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-01	445.5853960	271.3893428	627.7834444	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-02	487.2884685	274.8061170	659.5747979	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-03	571.0894081	341.0666040	808.7572015	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-04	561.4862847	389.9244689	719.2389570	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-05	571.1631183	371.2980240	859.3229077	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-06	585.5742368	426.8711247	764.5757877	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-07	617.4835730	388.7434387	790.1213378	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-08	581.3621879	398.9169315	864.4357930	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-09	585.9765248	412.8618969	858.8383389	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-10	563.1084865	384.3213500	789.9472870	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-11	635.2960248	487.4021303	845.3301198	m3/d	2
Sistema de Carga	4100-FI-054PV	Vazão vol. de NK	3913-01-12	650.7850245	509.8911175	853.677186	m3/d	2

Figura 3.7: Argus - Relatório Criado

### 3.1.2 SAGAS

O sistema SAGAS (Solução Analítica do Gás) é um simulador voltado para compreensão do comportamento de poços e reservatórios. Ele modela o fluxo de fluidos compressíveis em meio poroso, utilizando uma solução derivada do método de Green. Ver Jr. et al. (2013).

Os resultados desse simulador são fornecidos em formato texto e precisavam ser transportados para planilhas para serem analisados. Esse processo era trabalhoso e por particularidades dos dados nem sempre produzia um entendimento adequado dos resultados, o que acabou demandando a criação de uma interface para o mesmo.

Um ponto de destaque nesse estudo é a implementação da interface deste sistema. O uso de gráficos em escalas diferentes apresentando de maneiras variadas o mesmo conjunto de dados para se obter a visualização adequada serão abordados.

O resultado do simulador do sistema SAGAS, fornecido em formato texto são dados que correspondem a funções e suas derivadas. Para correta visualização destes dados, os gráficos precisam ter seus eixos em escalas lineares, logarítmicas e mistas, conforme necessário. Infelizmente a biblioteca utilizada não disponibiliza escalas menores que evidenciam melhor o uso de escala logarítmica. Ver figura 3.8.

O Quarteto de Anscombe Anscombe (1973), nos mostra a importância da visualização dos dados, mesmo quando os dados são bem agrupados em escalas lineares. Identificar pontos discrepantes em dados com variações de escala é ainda mais difícil, tornando a visualização crucial para correta interpretação dos dados.

Os três gráficos, com escalas distintas, apresentados em conjunto tem o objetivo de promover a interpretação correta dos dados. Como existe uma expectativa de como as curvas devem se apresentar, é fácil detectar visualmente se houve algum problema com a análise.

Na interface do sistema SAGAS os dados mostrados na tabela, ainda apresentada na parte superior da página, podem ser analisados através de três gráficos iterativos.

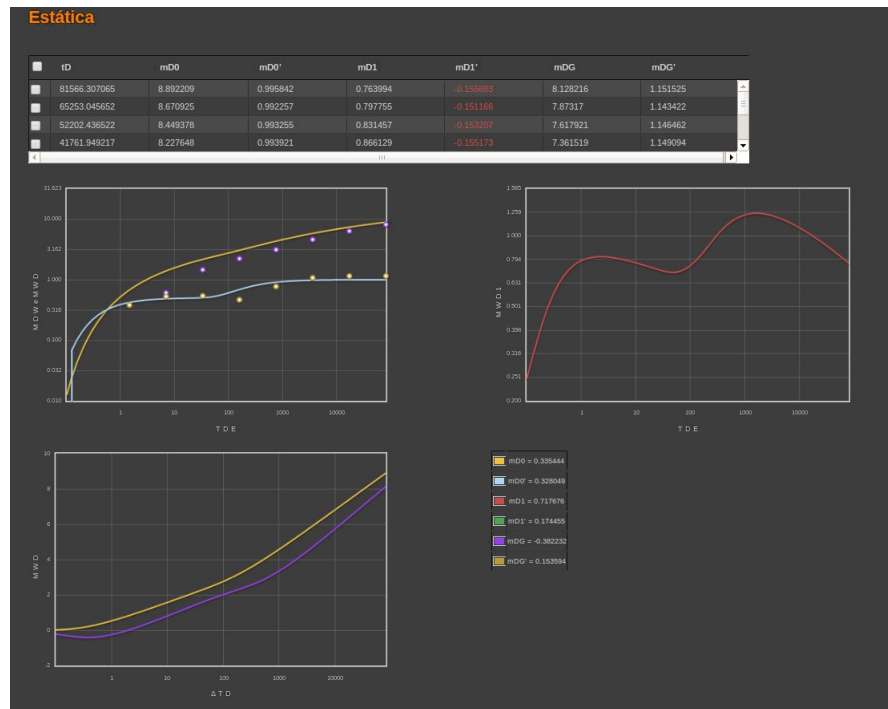


Figura 3.8: SAGAS - Gráficos com diferentes escalas

O primeiro apresenta uma parte dos dados da tabela em uma interpolação de pontos sobreposta por um gráfico de dispersão, com ambos os eixos em escala logarítmica.

No segundo, apenas o eixo y tem escala logarítmica e no terceiro os eixos são mostrados em escala linear. A legenda unificada dos três gráficos apresenta os valores de forma iterativa, atualizados conforme o passar do mouse nos pontos plotados. O conjunto desses valores em cada ponto equivale a uma linha da tabela.

A implementação desses gráficos pode se tornar complicada. No caso do sistema SAGAS, programação adicional teve que ser realizada para exibir os eixos com escala não linear e para unificar a legenda. Além disso, para plotagem dos gráficos de dispersão, um algoritmo de amostragem (sampling) teve que ser aplicado. Para as curvas interpoladas, ao contrário foram fornecidas as opções de interpolação linear a cada ponto ou o uso de splines.

### 3.1.3 Fouling

O simulador FoulingTR é um sistema desenvolvido pela Petrobras para fazer o acompanhamento da eficiência e da deposição em permutadores de calor e baterias de pré-aquecimento (BPA) das unidades operação. O objetivo do sistema é monitorar em tempo real as BPAs, de tal forma que o engenheiro responsável pela unidade possa utilizar as informações disponibilizadas para fazer diagnósticos da performance dos permutadores de calor. O sistema utiliza as informações para gerar modelos de deposição, que serão usados para montar a programação das paradas para limpeza dos permutadores de calor, a qual é obtida através da utilização de um algoritmo de otimização. Atualmente, o sistema está instalado em unidades de destilação atmosférica, coqueamento retardado e plataformas de produção de petróleo.

Os resultados em texto desse simulador eram armazenados em planilhas muito extensas, que chegavam atingir os limites do software de escritório e tornavam o trabalho muito difícil. A melhor manutenção desses dados, além da demanda de outras funcionalidades para exploração e análise dos dados motivaram a criação de uma interface para o sistema.

Como resposta a essa demanda, foi desenvolvida uma solução onde os dados passaram a ser armazenados em um servidor e uma interface Web para o sistema foi desenvolvida. Este projeto foi escolhido em 2011 como Destaque pela gerência da área de Serviços de Tecnologia de Informação e Telecomunicação da Petrobras.

O uso de gráficos, mapas georreferenciados e infográficos navegáveis, permitindo um acompanhamento virtual na unidade industrial (refinaria ou plataforma) nesta interface será abordado neste trabalho.

Uma execução do simulador FoulingTR tem como resultados os valores esperados ao longo do tempo relativos a uma série de condições físicas medidas por sensores nas instalações industriais. Em geral, o valor esperado para algumas centenas desses sensores são produzidos, para cada unidade simulada.

Lista de Variáveis			
Posição	Descrição	Unidade	Apelido
486	Corrente RV4: temperatura	°C	
487	Corrente RV4: vazão mássica	kg/h	
488	Corrente RV4: vaz vol	m3/h	
489	Corrente RV5: vaz vol	m3/h	
490	Corrente RV6: vaz vol	m3/h	
491	Corrente RV7: temperatura	°C	
492	Corrente RV8: temperatura	°C	
493	Corrente RV9: temperatura	°C	
494	Carqa Térmica do P-90AB	MMkcal/h	

Figura 3.9: Fouling - Variáveis e seus apelidos

Para fins de análise, muitas vezes é preciso comparar valores correlacionados, mas de ordens de grandeza muito diferentes, como por exemplo, a temperatura em um equipamento e o seu impacto econômico na unidade. Análises desse tipo são feitas através de gráficos, mas para estes serem úteis é preciso bastante cuidado nos ajustes do número de casas decimais dos dados e nos intervalos dos eixos dos gráficos.

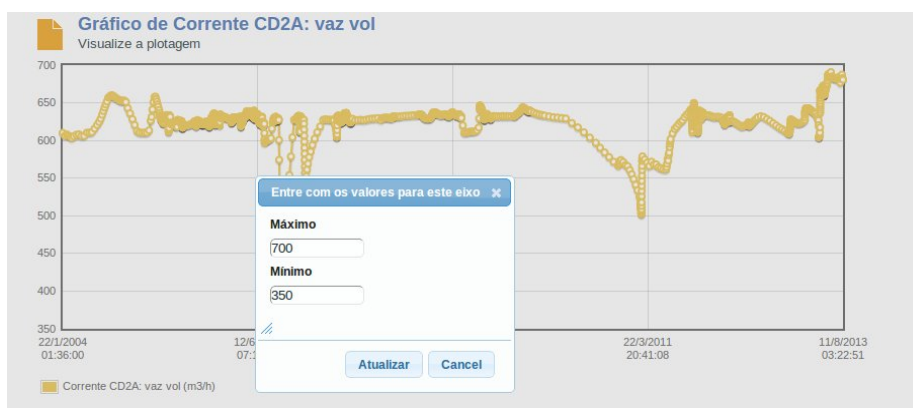


Figura 3.10: Fouling - configurando o eixo Y

Para atender a esta necessidade, na interface do sistema, esses valores são configuráveis. Além disso, existe também a opção de exibir as séries em eixos sobrepostos

o que permite avaliar medidas com ordem de grandeza diferentes no mesmo gráfico. Ver figuras 3.10, 3.11 e 3.12. Existe também um esperado ruído nos dados e as curvas precisam ser suavizadas. Na interface elas são apresentadas com o recurso de média móvel configurável para suavização. Ver figuras 3.13 e 3.14.

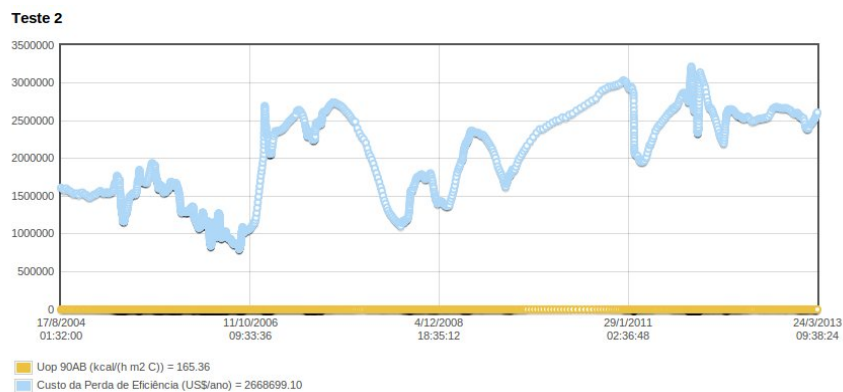


Figura 3.11: Fouling - Sem múltiplos eixos

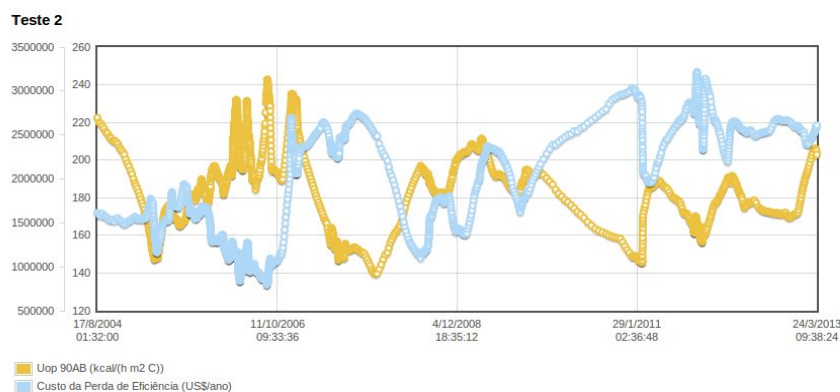


Figura 3.12: Fouling - Com múltiplos eixos

Apesar do sistema ser todo voltado para visualização de gráficos, seguindo a filosofia de disponibilizar os dados, a interface tem opções para fornecer arquivos textos com os resultados para todas as medidas em um dado momento e também, em formato de planilha, o conjunto dos dados exibidos em um gráfico.

Os gráficos são iterativos e os valores exatos de cada ponto são exibidos em caixas



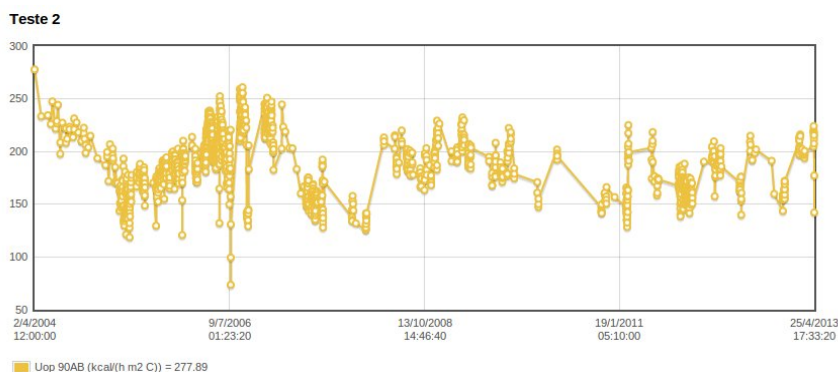


Figura 3.13: Fouling - Sem média móvel

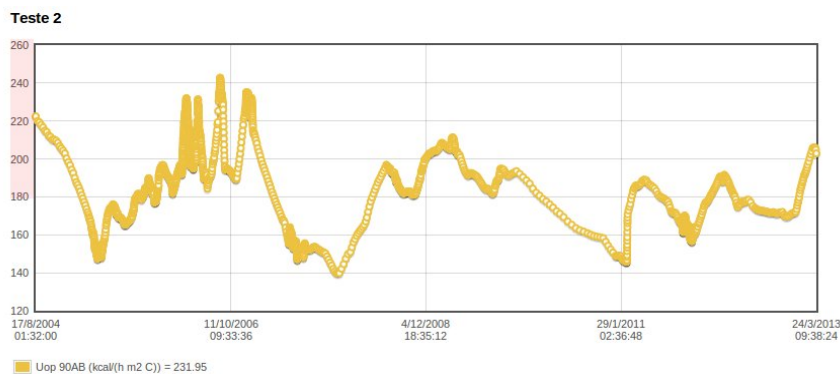


Figura 3.14: Fouling - Com média móvel

de referência ("tooltips") quando se passa com o mouse por cima deles. Isso proporciona a capacidade de fazer rapidamente conferências mais detalhadas, onde o normal seria ter que consultar tabelas.

Apesar da interface dispor dessas capacidades, a visualização de valores relativos a centenas de medidas, identificadas por nomes nem sempre muito significativos tornava difícil a utilização do sistema, em especial para pessoas de outras unidades ou da administração central.

A primeira versão da interface tentava mitigar isso com a possibilidade de se configurar nomes alternativos para cada medida, mas isso só resolve parte do problema. Ver

figura 3.9.

A solução que adotamos posteriormente para o problema foi a configuração de mapas navegáveis para cada unidade industrial onde o sistema é utilizado. O número de medidas para cada unidade e a demanda de popularizar o sistema nos desencorajou a utilizar qualquer solução que gerasse os mapas automaticamente. Ver figura 3.15.

Como se trata de mapas interativos exibidos no computador, usuário é apresentado a uma visão geral da unidade. Cada seção da unidade (ramal) é representada nessa visão geral com um símbolo clicável. Cliques nesses símbolos de ramais ou de outros equipamentos mais complexos conduzem a visões expandidas, que apresentam as medidas correspondentes. Cliques nas medidas levam a visualização de gráficos da mesma e "tooltips" fornecem o nome de cada uma. Além disso, a visualização apresenta um calendário, que também permite a visualização dos dados apresentados no tempo.

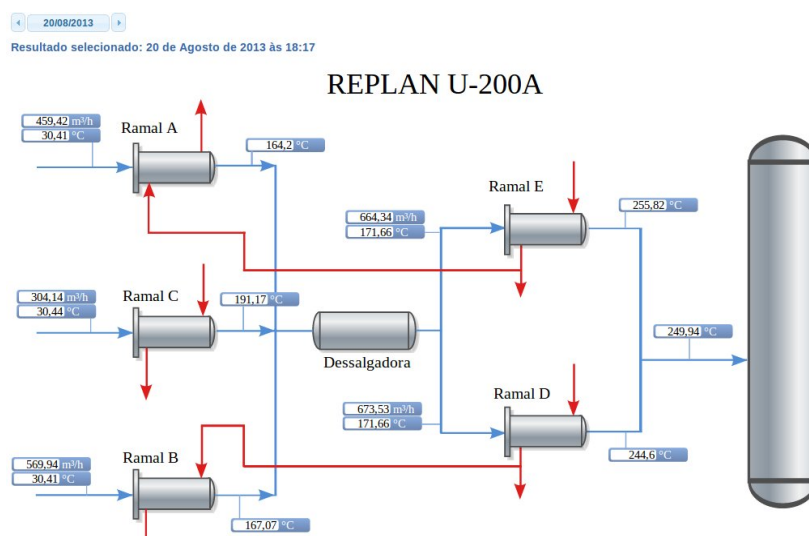


Figura 3.15: Fouling - diagrama de unidade

Esses mapas seguem vagamente a localização dos equipamentos nas unidades. Além disso, para facilitar a localização dos dados no mapa pelos clientes do sistema, ícones de apelo emocional como o flare e os trocadores foram confeccionados e cuidadosamente posicionados nos mapas.

Por último, para orientar os clientes que possuem acesso a dados de mais de uma unidade ou mais de uma instalação industrial uma estrutura de árvore permite ao cliente navegar para a unidade que quiser em qualquer uma das instalações. Atualmente mais de vinte refinarias e plataformas estão representadas no sistema, com diversas unidades cada. Ver figura 3.16.



Figura 3.16: Foulings - Árvore de navegação

### 3.1.4 Revestir

Esta aplicação tem como objetivo projetar o dimensionamento do revestimento de um poço para que ele consiga suportar todas as cargas de pressão a que ele venha a ser submetido, em cada segmento de revestimento usado na perfuração, denominados fase de perfuração, ao longo de sua vida útil, normalmente em um mínimo de dez anos até trinta ou quarenta anos. Essas cargas podem ocorrer pela pressão marinha ou terrestre, pela mudança do meio físico nas diferentes formações geológicas ao longo da perfuração ou por acidentes de operação que causam fluxos abruptos de fluidos ou gás.

As fases de perfuração que devem garantir a segurança do poço devem resistir a

mudanças de temperaturas que variam das centenas de graus centígrados quilômetros sob o solo a abaixo de zero no fundo do oceano. Condições do terreno, como ondulações e terremotos precisam ser considerados. Se não dimensionadas corretamente, elas podem causar a explosão ou colapso do poço, causando sérios danos patrimoniais e ambientais, além de ameaçar a segurança dos trabalhadores presentes no entorno.

O sistema orienta o projetista na aplicação em seu projeto do conjunto de cargas obrigatórias conforme estabelecido por normas da companhia, além de permitir a configuração de outras cargas específicas, de acordo com as particularidades do projeto.

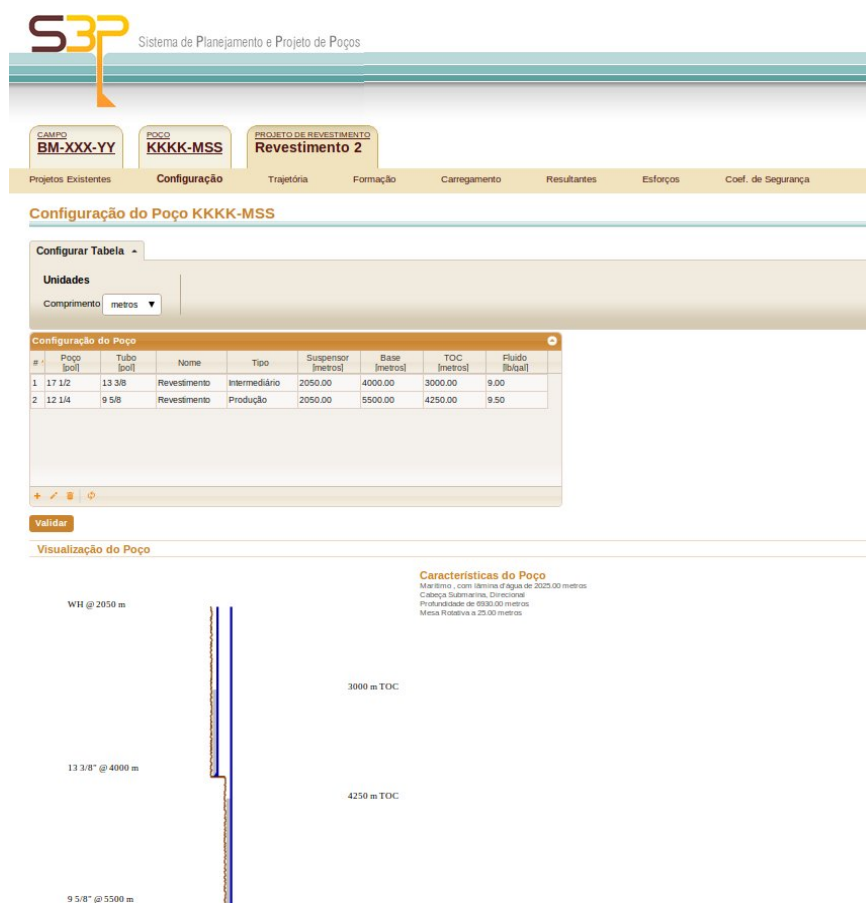


Figura 3.17: Estrutura de abas do sistema Revestir

O sistema Revestir foi criado, desde o início, com uma interface projetada para

automatizar processos de trabalho que eram antes executadas em papel ou com o auxílio de planilhas. No contexto da visualização de informação, o sistema apresenta como pontos de interesse a criação dinâmica de um infográfico do projeto de poço e o uso de tabelas e gráficos associados. Além disso, abordaremos como o sistema foi feito para orientar o pesquisador ao longo do projeto.

O sistema Revestir automatizou cálculos e etapas de projeto de poços que eram feitas em planilhas e outros softwares específicos. Esse processo de trabalho não unificado, com regras e metodologias variadas para cada tipo de poço, envolvendo sistemas de unidade diferentes e profissionais que trabalham em unidades distantes gerava o risco de falhas de integração, de esquecimento de etapas e a falhas de comunicação.

A interface do sistema foi produzida utilizando uma estrutura de abas como metáfora para sistemas como os de "carrinho de compras" dos sites de venda pela internet. Ao final do projeto de um poço, o usuário é avisado pelo sistema sobre qualquer problema e encaminhado para o local onde está o problema, de forma semelhante aos programas de cobrança de imposto de renda. Ver figura 3.17.

Fortemente influenciado pela rotina de trabalho, o Revestir foi projetado com as tabelas acima dos gráficos, prática que preferimos evitar em projetos que envolvem menos edição dos dados. As planilhas, apesar de serem usadas durante boa parte do tempo, são utilizadas em atividades mais demoradas como aferição de dados e podem aparecer abaixo. Como recomenda Few (2012) os gráficos são mais usados para análises rápidas de projetos e devem em geral ser mostrados primeiros, aumentando a usabilidade do sistema.

Em um projeto típico, mais de 200 regras de negócio são verificadas pelo sistema e apontadas para o usuário em caso de problemas, através variações de cores nas planilhas e mensagens de alerta.

As tabelas foram automatizadas para apontar as falhas mais comuns ao longo do preenchimento, mas durante o desenvolvimento do projeto se mostrou impossível em

alguns casos e computacionalmente inviável em outros apontar todas as falhas no mesmo momento. O planejamento de quais regras deveriam ser validadas logo no preenchimento das planilhas e de qual preenchimento poderia ser feito de forma automática consumiu grande parte do tempo de projeto do sistema.

Associada a um destes formulários de tabela temos uma funcionalidade do módulo que faz a apresentação de um infográfico, permitindo ao engenheiro uma rápida visualização estrutural do projeto que está sendo criado. Essa figura é atualizada em tempo real enquanto o módulo está sendo utilizado.

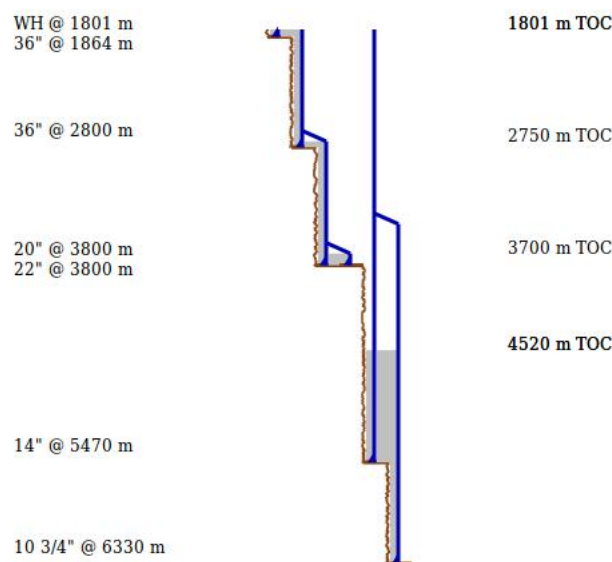


Figura 3.18: Revestir - Infográfico

Nela, cada fase de perfuração é desenhada, na medida do possível, em escala aproximada aos dados reais do projeto, com um tamanho mínimo sendo utilizado no caso de objetos muito pequenos. Como não existe um limite pré-determinado para o tamanho do poço, as dimensões das fases são normalizadas e desenhadas de acordo com a área disponível para a figura. Ver figura 3.18.

Além das fases de revestimento, também é desenhada uma representação da cimentação utilizada em cada fase, a partir do dado "Topo de Cimento", inserido no projeto. A posição dessas fases e cimentação é marcada na figura e detalhes como o assentamento e a ligação de revestimentos suspensos (liners) são representados no infográfico.

Apesar da figura apresentada nesse infográfico ser de grande simplicidade, sendo produzida apenas com segmentos de reta, ela tem papel importante para melhorar o entendimento do que está sendo projetado pelo engenheiro. Mesmo um projeto com poucas fases rapidamente combina várias distâncias com muita variação que precisam ser levadas em conta, sendo bem difícil manter uma imagem mental atualizada de uma estrutura como a mostrada na figura 3.18 ao longo do projeto.

Para o sistema Revestir também foi desenvolvida a possibilidade de criação de relatórios para impressão editáveis com a informação dos projetos.

Como os mesmos princípios de visualização aplicados na interface se aplicam para a informação impressa, não houve grandes mudanças, do ponto de vista da visualização de informação, na passagem desses dados para o formato texto.

Entretanto, a ausência de interatividade, as necessidades de formatação para impressão em papel diferentes das para apresentação em tela e a necessidade de fornecer edição gerou alguns desafios de implementação. As proporções do infográfico precisaram ser distorcidas em alguns momentos para se obter uma melhor apresentação.

### **3.1.5 STRITA**

O sistema STRITA (Análise de testes de injetividade em Reservatórios Estratificados) é um simulador de um processo de completação de poços, que modela testes de injeção de água e analisa o fluxo bifásico em um reservatório de óleo com múltiplas zonas de produção, denominados multicamadas. A. Barreto Jr. (2011)

Os resultados desse simulador são textos extensos e de difícil interpretação, mesmo depois da exportação dos mesmos para planilha e com a plotagem de gráficos. Essa

difficuldade motivou a criação de uma interface WEB para o sistema.

O uso de uma animação feita a partir de gráficos nessa interface e como ela permite a anotação de pontos específicos para exibição estática, será abordado neste trabalho.

O simulador do STRITA fornece uma saída de texto muito grande na forma de diversas tabelas. O sistema possui um recurso de visualização que permite a comparação de dados reais com dados simulados, na mesma plotagem.

Os resultados do simulador podem ser analisados através da plotagem de segmentos desses dados, que representam um momento no tempo. Entretanto, o conjunto completo dos resultados na realidade modela uma animação do avanço do fluido injetado nas camadas do reservatório.

Esse avanço varia muito ao longo do tempo, geralmente sendo muito lento no início do teste e aumentando muito no final. Identificar pontos variações nesse avanço é justamente o objetivo do simulador, mas selecionar esses momentos para análise via plotagem a partir dos dados em texto era um problema.

A solução desenvolvida de visualização foi a plotagem interativa de todos os momentos, através de uma barra de rolagem, criando o efeito de uma animação.

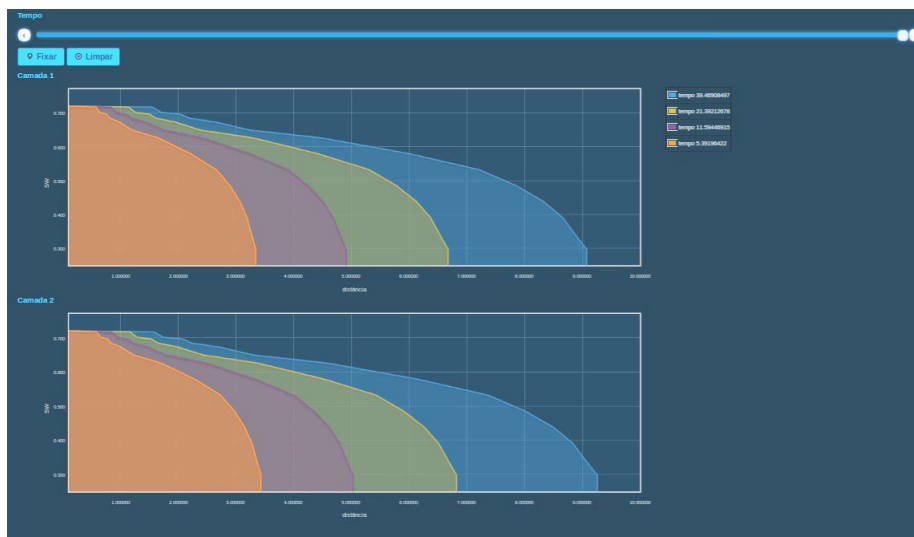


Figura 3.19: Strita - Pontos selecionados em duas camadas



Para que essa animação não atrapalhasse a análise, o código foi bastante otimizado para que a plotagem fosse rápida e que o avanço e recuo no tempo fossem precisos.

Além disso, foi criada uma funcionalidade permitindo a fixação na animação de momentos no tempo, onde o gráfico do instante selecionado é mantido com uma cor diferente do tom de azul utilizado na animação escolhida a partir de uma paleta pré-definida. Ver figura 3.19.

## Capítulo 4

# Conclusões e Trabalhos Futuros

### 4.1 Conclusões

A aplicação dos conceitos de visualização de informação nas interfaces dos sistemas científicos estudados foi realizada em estágios diferentes do desenvolvimento, algumas vezes na criação de interfaces novas para sistemas já em uso e em outros casos no desenvolvimento em conjunto com o pesquisador desde o início da concepção do projeto. Em todos os casos, a aplicação destes conceitos se mostrou um grande benefício do ponto de vista dos pesquisadores que utilizam esses sistemas.

As idéias expressas nesses sistemas são a concretização de anos de pesquisa e experimentação que permitiram o desenvolvimento de algoritmos e processos que produzem uma grande massa de dados para cada caso ou ensaio em que são aplicados. Esses resultados precisavam ser analisados manualmente ou com auxílio de ferramentas rudimentares como planilhas e gráficos estáticos.

As interfaces produzidas com desenvolvimento atento aos conceitos de visualização de informação promovem a correta visualização dessas massas de dados, facilitando a análise, confirmando o resultado teórico esperado pelo pesquisador ou chamando a sua atenção para qualquer resultado inesperado.

## 4.2 Novos Componentes

A repetição de alguns padrões ao longo do desenvolvimento dos sistemas estudados neste trabalho nos motiva a propor o desenvolvimento de novos componentes, capazes de combinar características de componentes que foram utilizados no desenvolvimento e automatizar as adaptações que foram feitas. Esses novos componentes poderão reduzir o trabalho necessário para produzir novos sistemas e promover boas práticas de visualização de informação.

Existem várias questões ainda não resolvidas no que diz respeito a implementação desses componentes. Eles devem ser escritos apenas em Javascript ou devem ser criados com uma combinação dessa linguagem com a linguagem Python? Que componentes podem servir como ponto de partida e quais funcionalidades devem ser completamente reimplementadas? Qual a complexidade dessas implementações? Como devem ser as interfaces de programação desses novos componentes?

Embora ainda não tenham sido respondidas muitas das questões referentes a implementação, estamos planejando como esses novos componentes podem ser especificados.

Apresentaremos a seguir uma descrição desses componentes onde a tabela é usada como base para abrigar dados e especificar a formatação. As descrições são baseadas numa implementação utilizando componentes Web com elementos customizados (ver WebComponents (2014)), mas resultados semelhantes poderiam ser obtidos utilizando a biblioteca Angular (2014).

### 4.2.1 Componente para Combinação de Tabela e Gráfico

Gráficos interativos e tabelas com seus dados foram usados em praticamente todos os projetos apresentados nesse estudo. Esses componentes visuais são na verdade utilizados em conjunto, formando uma ferramenta de visualização e análise de dados.

Boa parte do trabalho de programação envolvendo a ligação entre esses componentes

poderia ser evitada com a criação de um componente único, que partisse da especificação de uma tabela em HTML e um único comando se encarregando na criação do gráfico ou gráficos especificados na tabela.

Esse componente poderia contar com informações adicionais que a tabela poderia conter em sua estrutura, especificando quais colunas deveriam ser especificadas em escalas logarítmicas e quais unidades deveriam ser utilizadas. A ligação entre os componentes, consistindo na atualização dos valores na legenda do gráfico, possibilidade de exclusão de pontos e redimensionamento de eixos e traço de curvas a partir de pontos selecionados seriam fornecidos automaticamente pelo componente.

Além disso, os botões para exportar o gráfico em formato de imagem, exportar os dados da planilha em formato de planilha também poderiam ser fornecidos automaticamente. O componente promoveria, dessa forma, o compartilhamento das informações no momento em que elas já foram pré-selecionadas pelo pesquisador que está utilizando o sistema.

Apresentamos abaixo um exemplo de especificação de tabela com dois gráficos. A segunda coluna é usada para fornecer o valor do eixo x nos dois gráficos. O eixo y é fornecido pela terceira e quarta coluna. O primeiro gráfico é um gráfico de linha azul e o segundo é um gráfico de pontos vermelho com eixo y em escala logarítmica.

```
<graph name="gr1" color="blue" type="line"\>
<graph name="gr2" color="red" type="dot"\>
<table id="mytable">
<tr>
<th>Tipo</th>
<th graph="gr1 gr2">Hora</th>
<th graph="gr1" axis="yaxis">Valor</th>
<th graph="gr2" axis="yaxis" log="true">Temperatura</th>
</tr>
```

#### 4.2.2 Componente para Criação de Infográficos

Infográficos e mapas referenciados foram utilizados nos sistemas S3P e Fouling e devem vir a ser utilizados no Argus. Uma abordagem semelhante também foi utilizada para

confeção de relatórios no sistema Argus. A base nesses trabalhos foi a utilização de arquivos no formato SVG com referências ("tags") que foram substituídas com expressões regulares.

O maior trabalho nessas implementações foi o realizado no sistema Fouling, onde um componente de calendário com setas para deslocamento no tempo e a transição de um arquivo SVG para outro, feita via links dentro desses arquivos foram implementadas.

A proposta nesse caso é a criação de um componente que receba como argumentos os arquivos SVG, os dados em uma tabela com uma coluna marcando o tempo, se houver e que se encarregue de realizar essa transição entre arquivos e a substituição das referências pelos dados. Havendo a coluna marcada como contendo as datas ou horas no tempo, o componente também deve se encarregar da navegação dos dados no tempo.

Apresentamos abaixo um exemplo de especificação de tabela para infográfico. Os arquivos `svg_file1`, `svg_file2` e `svg_file3` serão usados, com a navegação começando do primeiro. As tags `tipo`, `valor1`, `valor2` e `valor3` nos arquivos `svg` serão substituídas pelos valores das colunas respectivas na tabela. A coluna com nome "hora" será usada para fornecer as datas para o calendário

```
<graph name="infogr1" type="info" files="svg_file1:svg_file2:svg_file3">
<table id="mytable">
<tr>
<th name="tipo">Tipo</th>
<th name="hora" type="time">Hora</th>
<th name="valor1">Valor</th>
<th name="valor2">Temperatura</th>
<th name="valor3">Custo</th>
</tr>
```

### 4.2.3 Componentes para Análise em Animações de Gráficos e Vídeos

No sistema STRITA foram combinados componente de plotagem de gráfico, um componente de barra deslizante e botões foram combinados para formar uma ferramenta de seleção de momentos específicos ao longo de uma animação.

Apesar de algumas bibliotecas de plotagem apresentarem o recurso de utilizar uma

barra deslizante para representar o tempo em uma animação feita a partir de transições em um gráfico, dentre elas a biblioteca Matplotlib (ver Hunter (2014)), em geral não é possível selecionar e sobrepor momentos no tempo, como ocorre no sistema STRITA.

Propomos a criação de um único componente que forneça a barra deslizante para navegação dos dados no tempo e os botões para selecionar e desselecionar momentos, como foi feito no STRITA com a combinação de vários componentes. O ponto de partida pode ser uma tabela, da mesma forma como propusemos para nosso componente que combina plotagem e tabela. Nesse caso, uma das colunas deve ter uma marcação especial para indicar que seus dados se referem ao tempo para que o componente possa usá-la na barra deslizante.

No sistema STRITA o gráfico é apresentado inicialmente sem nenhum momento pré-selecionado. O componente não precisa ter essa limitação e as células da tabela na coluna referente ao tempo podem ter uma marcação especial para indicar que o momento a que se referem deve ser selecionado.

Apresentamos abaixo um exemplo de especificação para tabela com a coluna referente ao tempo demarcada.

```
<table id="mytable" type="gap">
<tr>
<th type="color">País</th>
<th type="time">Data</th>
<th type="yaxis">Valor</th>
<th type="xaxis">Temperatura</th>
<th type="size">Custo</th>
</tr>
```

Segue abaixo uma especificação de gráfico no estilo "gapminder". Uma cor é atribuída pelo componente automaticamente para cada tipo especificado na primeira coluna. A segunda coluna é utilizada para barra deslizante, a terceira e a quarta coluna controlam a posição dos pontos no gráfico. A classe especificada para última coluna faz o componente usar o Custo para dimensionar a variação do tamanho dos pontos, ao longo do tempo.

```
<table id="mytable" class="anim dot">
```

```

<tr>
<th class="time">Data</th>
<th class="yaxis">Valor</th>
<th class="xaxis red">Japão</th>
<th class="size red">Custo no Japão</th>
<th class="xaxis green">Brasil</th>
<th class="size green">Custo no Brasil</th>
<th class="xaxis blue">Argentina</th>
<th class="size blue">Custo na Argentina</th>
</tr>

```

Apresentamos a seguir, outra especificação no estilo "gapminder", dessa vez as cores são atribuídas aos países pelo usuário e o eixo y e as datas são comuns.

Especificação para gráfico com barra deslizante. A primeira coluna armazena as datas a serem utilizadas na barra deslizante. A segunda deve armazenar uma tabela especificando o gráfico como ele deve ser apresentada naquele momento.

```

<table id="mytable" type="anim">
<tr>
<th type="time">data</th>
<th type="table"> ... </th>
</tr>

```

Um componente semelhante a este também pode ser útil para análise de vídeos. Nesse caso, imagens de momentos selecionados poderiam ser sobrepostos em uma imagem composta. Tal componente pode ter aplicação na análise de vídeos de práticas desportivas, em estudos de acidentes e de movimento em geral.

As imagens dos momentos selecionados podem ser combinadas utilizando a mesma técnica empregada no programa faceblend.py, do pacote facesworkshop. Nele fotos de rostos são combinadas utilizando o comando blend da biblioteca Python PIL.

Nas figuras 4.1-3 apresentamos como seria o resultado do componente, aplicado a imagens de um vídeo de uma palestra do desenvolvedor Bret Victor.



Figura 4.1: Imagem do vídeo de palestra de Bret Victor

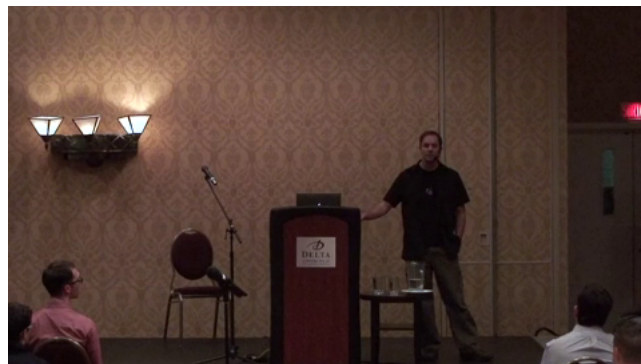


Figura 4.2: Imagem do vídeo de palestra de Bret Victor

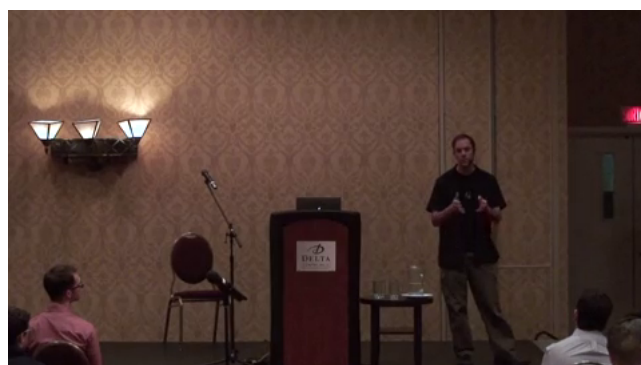


Figura 4.3: Imagem do vídeo de palestra de Bret Victor



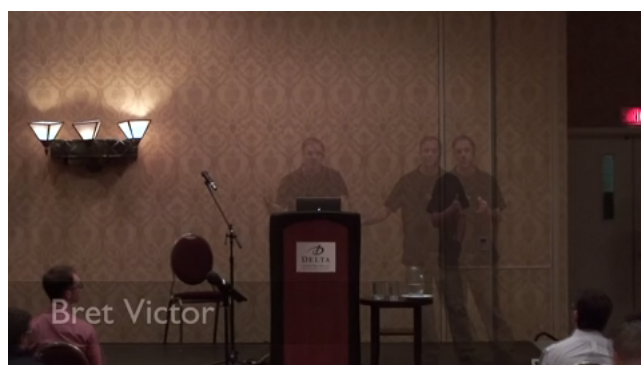


Figura 4.4: Colagem de imagens do vídeo de palestra de Bret Victor

# Referências Bibliográficas

- A. Barreto Jr., A. Peres, . A. P. (2011). Water injectivity tests on multilayered oil reservoirs. *SPE Journal*.
- AB, M. S. (2014). <http://www.tinymce.com/>.
- Adobe (2014). <http://kuler.adobe.com/>.
- Angular (2014). <http://angularjs.org>.
- ANOVA (2014). <http://www.atireiopaunografico.com.br/2010/11/resposta-para-o-quiz-grafico-de-barras.html>.
- Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician*, 27(1):pp. 17–21.
- Baumeister, R. e Tierney, J. (2011). *Willpower: Rediscovering the Greatest Human Strength*. Penguin Group US.
- Borges, L. E. (2014). <http://ricardoduarte.github.io/python-para-desenvolvedores/>.
- Cavanagh, P. e Alvarez, G. A. (2005). Tracking multiple targets with multifocal attention. *Trends in Cognitive Sciences*, 9(7):349 – 354.
- ColorScheme (2014). <http://colorschemedesigner.com/>.
- COLRD (2014). <http://colrd.com>.

- Coltheart, M. (1980). Iconic memory and visible persistence. *Perception Psychophysics*, 27(3):183–228.
- Few, S. (2012). *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. ANALYTICS Press.
- Fry, B. (2008). *Visualizing Data*. 1st edition.
- GraphViz (2014). <http://www.graphviz.org/>.
- Greenfeld, D. e Roy, A. (2013). *Two Scoops of Django: Best Practices for Django 1.5*. CreateSpace Independent Publishing Platform.
- Healey, C. G., Kocherlakota, S., Rao, V., Mehta, R., e Amant, R. S. (2008). Visual perception and mixed-initiative interaction for assisted visualization design. *IEEE Trans. Vis. Comput. Graph.*, 14(2):396–411.
- Hunter, J. (2014). <http://www.matplotlib.org/>.
- Jr., A. B. B., Peres;, A. M., e Pires, A. P. (2013). A variable-rate solution to the nonlinear diffusivity gas equation by use of green’s-function method. *SPE Journal*, 18(1):57—68.
- Kerim e Jens (2014). <http://codeboje.de/pysvg/>.
- Leibman, M. (2014). <https://github.com/mleibman/slickgrid/wiki>.
- Lynn, J. (2014). <http://www.worqx.com/>.
- Macaulay, D. (1988). *The way things work*. Houghton Mifflin.
- Megan Monroe, R. L. e Wongsuphasawat, K. (2014). <http://www.cs.umd.edu/hcil/eventflow/manual/index.html>.

- Meunier, S. (2014). <http://www.smokycogs.com/blog/subway-map-generator-plugin-for-jquery/>.
- Moitzi, M. (2014). <http://readthedocs.org/docs/svgwrite/en/latest/>.
- Museum, L. T. (2014). <http://homepage.ntlworld.com/clivebillson/tube/tube.html>.
- perez, F. (2014). <http://ipython.org/notebook.html>.
- Rauschmayer, A. (2014). *Speaking JavaScript*. O'Reilly Media.
- Rosling, H. (2014). <http://www.gapminder.org/>.
- RProject (2014). <http://www.r-project.org/>.
- Schnur, L. (2014). <http://www.flotcharts.org/>.
- Steele, J. e Iliinsky, N. (2010). *Beautiful Visualization*. Theory in practice series. O'Reilly Media.
- Susan Havre, Beth Hetzler, L. N. (2014). <http://www.ifs.tuwien.ac.at/silvia/wien/vu-infovis/references/havre-ieeeinfovis00.pdf>.
- SVG (2014). <http://www.w3.org/graphics/svg/>.
- T. Lammarsch, W. A. (2014). <http://publik.tuwien.ac.at/files/pubdat217972.pdf>.
- Tomov, T. (2014). <http://www.trirand.com/jqgridwiki/doku.php>.
- Tufte, E. (1983). *The visual display of quantitative information*. Graphics Press.
- Ward, M., Grinstein, G., e Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. A. K. Peters, Ltd., Natick, MA, USA.
- Ware, C. (2004). *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

WebComponents (2014). <http://w3c.github.io/webcomponents/explainer>.

Wilson, J. M. (2003). Gantt charts: A centenary appreciation. *European Journal of Operational Research*, 149(2):430 – 437. Sequencing and Scheduling.

# Apêndice A

## Tecnologias

### A.1 Python e JavaScript

Para o desenvolvimento dos sistemas descritos neste trabalho, foi utilizada a linguagem de programação Python nos servidores Web, usando o framework Django (ver Greenfeld e Roy (2013)). Nesta arquitetura, o servidor é responsável por armazenar dos dados e fornecer os mesmos em páginas HTML com o código em javascript, que pode ser executado nos navegadores dos clientes.

A linguagem Python é uma linguagem bastante utilizada e apreciada no meio científico. É uma linguagem clara e concisa, com forte separação de escopo de nomes, tipagem forte, política de evitar erros silenciosos e formas únicas e bem definidas de executar cada tarefa. Essas características tornam a linguagem bastante produtiva e popular. Borges (2014)

Por meio dos templates do framework Django, as as páginas HTML são geradas com o código em linguagem JavaScript adaptado com os dados que devem ser exibidos nos navegadores nos clientes. Essa adaptação do código é feita em um processo semelhante uso de macros nos ambientes C e C++.

A linguagem JavaScript (ver Rauschmayer (2014)) tem como vantagens o fato de

ser muito rápida e poder controlar o ambiente do navegador no cliente, onde o código escrito nela é executado. Isso permite a produção de componentes gráficos com bastante interatividade. Outra vantagem de executar o código no navegador é que isso distribui o tempo de processamento nos computadores onde será feita a visualização, evitando a sobrecarga dos servidores.

Apesar de ser uma linguagem interpretada, a competição entre os fabricantes de navegadores por mais usuários levou a criação de interpretadores muito otimizados, o que tornou a linguagem JavaScript uma linguagem muito rápida. Por outro lado, essa tecnologia foi desenvolvida por grupos diferentes e nem sempre cooperativos. Devido a sua importância a linguagem conseguiu ser padronizada, mas ao contrário da linguagem Python ela não é tão clara, possui várias formas de executar a mesma tarefa e frequentemente produz erros silenciosos. Tais características fazem a linguagem demandar longos períodos de depuração de erros e testes, o que acaba comprometendo sua produtividade.

Fazendo o uso dessas duas linguagens foi possível combinar componentes de diversas bibliotecas para conseguir fazer com que eles tivessem o comportamento desejado nos sistemas. O uso dessas bibliotecas será descrito nas seções a seguir.

## **A.2 Plotagens - Flot**

Nos sistemas descritos nesse trabalho, as soluções de plotagem foram construídas com a biblioteca javascript Flot. Ver Schnur (2014).

Essa biblioteca possui limitações em comparação com a biblioteca python Matplotlib (ver Hunter (2014)), que seria nossa opção caso os sistemas tivessem como requisito estarem disponíveis na intranet da empresa. Essas limitações geraram algumas tarefas extras, como por exemplo, as plotagens em escala logarítmica usadas nos sistemas que tiveram que ser desenvolvidas internamente. Outras funcionalidades da biblioteca precisaram ser extendidas para se adequar ao sistema Fouling. O algoritmo que controla

a marcação nos eixos ("tick marks") teve que ser feito.

Funcionalidades como a exibição de legendas e marcação de valores de acordo com o movimento do mouse, bem como o destaque de qual eixo estava controlando cada ponto selecionado também tiveram que ser programados.

Como a API da biblioteca está longe de ser completa, em muitas vezes tivemos que recorrer ao código fonte para fazer essas adaptações. No entanto, a biblioteca se mostrou bastante adaptável, fornecendo acesso aos eventos e permitindo a configuração de tudo foi necessário.

Essa flexibilidade foi fundamental para criação da plotagem animada feita para o sistema Strita. A alternativa considerada antes seria gerar filmes no servidor, o que comprometeria totalmente a usabilidade da ferramenta. O resultado, deixando o servidor encarregado somente da seleção dos dados e produzindo toda a exibição da animação no cliente WEB foi muito rápido, o que viabilizou a solução como ferramenta de análise.

### **A.3 Mapas e Infográficos - SVG**

Scalable Vector Graphics SVG (2014) é um formato de imagem vetorial 2D baseado em XML com especificação aberta desenvolvida pelo World Wide WEB Consortium (W3C). Esse formato foi apresenta diversas características que o tornam desejável para o contexto de visualização de informação. Dentre elas:

- É um formato de imagem vetorial, podendo se adequar a qualquer necessidade de tela e de resolução no momento da exibição.
- É um formato de texto, podendo ser processado programaticamente após sua produção.
- Permite interatividade e suporta animações.
- Pode ser exibido em navegadores WEB.
- Suporta navegação por hyperlinks.



- É programável em linguagem JavaScript.
- Possui ferramentas de edição poderosas e livres, como o software Inkscape ou o SVG Edit.
- Pode ser totalmente processado no servidor WEB e ainda prover interatividade no navegador.

As imagens em arquivos SVG podem ser geradas a partir de softwares de edição manual como o Inkscape ou a partir de bibliotecas de software como o GraphViz, Matplotlib ou Pygal. O ajuste e extensão de funcionalidades desses arquivos é bastante facilitado pelo fato do formato SVG ser meramente um formato de texto. As transformações podem ser realizadas a partir de substituições de string e do uso de expressões regulares.

A experiência com o Fouling nos serviu de referência para recomendar que o formato SVG para criação de mapas georeferenciados.

Entretanto, a criação manual de mapas e infográficos é trabalhosa e pode demandar mais tempo e recursos que o disponível para o projeto. Nesses casos, a criação de mapas e diagramas automaticamente é mais indicada e para isso podemos utilizar o software Graphviz (ver GraphViz (2014)). Com ele é possível gerar diagramas em formato SVG que podem ser perfeitamente integrados ao processo.

Quando mapas mais elaborados são necessários, como no sistema Fouling, também utilizamos o formato SVG, mas nesse caso os arquivos com os mapas das unidades de refinaria foram desenhados no Inkscape e nos lugares adequados, foram colocadas palavras chaves predefinidas para indicar qual variável de medição deveria ser apresentada nessas posições.

Com o uso de expressões regulares, foi possível determinar rapidamente quais variáveis de medição deveriam ser usadas. A posição correta foi obtida automaticamente, meramente substituindo a palavra chave (já posicionada corretamente) pela variável de medição correspondente. O mesmo processo foi usado para inserção de gráficos e

tabelas no sistema ARGUS.

No sistema Fouling, o software Inkscape foi utilizado para edição das figuras em SVG e uma extensão foi desenvolvida internamente para ele.

No sistema Revestir A biblioteca originalmente usada para gerar a imagem no formato SVG foi a pySVG de Kerim e Jens (2014). Ela teve sua última atualização em 2010 e se mostrou limitada, ao não permitir especificar a espessura das linhas a serem plotadas.

Após uma pesquisa, foi escolhida a biblioteca svgwrite (ver Moitzi (2014)), mais atualizada e bem documentada. Mudar de biblioteca não foi muito difícil, bastou a alteração de alguns comandos. Com a nova biblioteca, foi possível especificar o tamanho da espessura da linha, o que simplificou um pouco o código.

## A.4 Tabelas - Slickgrid e jqGrid

As tabelas utilizadas no sistemas Revestir, SAGAS, STRITA, Fouling e SAUR foram criadas com as tecnologias Slickgrid e jqGrid. Ver Leibman (2014) e Tomov (2014)

Nos sistemas, essas tabelas foram configuradas para facilitar o trabalho dos clientes, fornecer informações relevantes e evitar erros.

As tabelas foram configuradas para discretamente alternar a cor de fundo a cada linha, para facilitar a visualização. Além disso, as cores das fontes são utilizadas para destacar erros e incoerências.

Alguns campos foram configurados para exibir caixas de escolha, para evitar erros de digitação e orientar o cliente nos valores possíveis. Em alguns casos, uma caixa de diálogo é exibida para facilitar ao cliente a escolha dos valores de uma célula.

Nos sistemas SAGAS e STRITA, as tabelas possuem alguns recursos de integração com os gráficos que exibem seus valores. Pontos escolhidos a partir da tabela podem ser excluídos dos gráficos e a linha na tabela referente a cada ponto no gráfico é mostrada

conforme o posicionamento do mouse.

## A.5 Documentação - Ipython Notebook

Atualmente a documentação do sistema STRITA está sendo produzida utilizando-se o ipython notebook. Ver perez (2014)

Sua capacidade de integrar a execução de código python com a criação de documentos com formatações diversas como markdown, html, LaTeX e a sua capacidade de exibir imagens e vídeos tem se mostrado muito útil.

Recentemente, nosso grupo reeditou o livro "Python Para Desenvolvedores" utilizando essa ferramenta. Ver Borges (2014)

## A.6 Relatórios

No sistema Revestir, a necessidade de criar relatórios editáveis fora da interface WEB nos fez optar pela criação dos mesmos no formato RTF, que pode ser editado em diversos editores de texto como o Notepad, Word e BrOffice Writer.

Para gerar os gráficos, nas configurações especificadas interativamente pelos clientes, foi feita a simulação do acesso do navegador pelo servidor, utilizando as tecnologias CasperJS e PhantomJS.

No sistema Argus a edição ocorre somente na interface WEB, o que tornou a criação do documento, neste aspecto, mais simplificada. Foi utilizado o formato HTML e o editor TineMCE. Um processo de substituição semelhante ao utilizado nos diagramas do sistema Fouling foi aplicado para realizar a troca dos códigos convencionados de gráfico e tabela no documento.

Terminada a edição, o documento em formato HTML é formatado para impressão e exportado em formato PDF. Essa funcionalidade foi desenvolvida com o uso da biblioteca WeasyPrint.