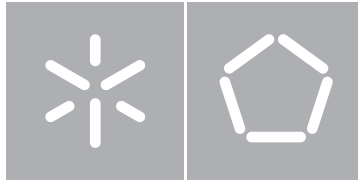




Universidade do Minho
Escola de Engenharia

Rui Miguel Pereira da Costa Barros
Dashboarding - Projeto e Implementação
de Painéis Analíticos

Outubro de 2013



Universidade do Minho

Escola de Engenharia
Departamento de Informática

Rui Miguel Pereira da Costa Barros

Dashboarding - Projeto e Implementação
de Painéis Analíticos

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de
Professor Doutor Orlando Manuel de Oliveira
Belo

Outubro de 2013

Anexo 3

DECLARAÇÃO

Nome

Rui Miguel Pereira da Costa Barros

Endereço electrónico: ruimpc.barros@gmail.com Telefone: 910489070 / 253218539

Número do Bilhete de Identidade: 13801252

Título dissertação /tese

Dashboarding - Projeto e Implementação de Painéis Analíticos

Orientador(es):

Professor Doutor Orlando Manuel de Oliveira Belo

Ano de conclusão: 2013

Designação do Mestrado ou do Ramo de Conhecimento do Doutoramento:

Mestrado em Engenharia Informática

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, 22/10/2013

Assinatura: Rui Miguel Barros

Agradecimentos

Ao professor Orlando Belo, orientador desta dissertação, que sempre se demonstrou disponível para me ajudar a ultrapassar os obstáculos que foram sendo encontrados ao longo do desenvolvimento desta dissertação, e que partilhou comigo todo o seu conhecimento e experiência no assunto abordado.

A todos os professores que me acompanharam ao longo da minha licenciatura e mestrado em engenharia informática pela transmissão de conhecimentos que me permitiram realizar esta dissertação.

Aos meus amigos que me acompanharam ao longo da minha formação e que conseqüentemente foram contribuindo para a mesma. Uma especial menção ao Paulo Rodrigues que acompanhou mais de perto o desenvolvimento desta dissertação.

Aos meus pais pelo apoio que sempre demonstraram ao longo destes anos, e por possibilitarem a minha formação académica.

Por fim, a todos aqueles que não foram aqui mencionados mas que contribuíram direta ou indiretamente para a realização desta dissertação.

Resumo

Dashboarding - Projeto e Implementação de Painéis Analíticos

Na atualidade, graças às elevadas capacidades computacionais e gráficas existentes, é possível dotar os sistemas de processamento analítico com ferramentas de visualização e manipulação de informação muito atrativas e de fácil utilização, em particular quando utilizamos para isso *dashboards*. Os *dashboards* tornam a interação com a informação proveniente de um sistema de processamento analítico mais interativa e eficaz, muito graças à modularidade inerente aos seus componentes gráficos e à sua qualidade inata de representar a informação graficamente. A modularidade também é uma característica importante uma vez que permite modificar o sistema utilizando apenas cliques do rato, enquanto que, por sua vez, a representação gráfica da informação facilita a sua análise e interiorização (Few, 2006a). Estas qualidades, entre outras, fazem com que os *dashboards* sejam uma ferramenta fulcral na análise da informação e no suporte à tomada de decisão no seio de uma empresa, tendo sempre em mente que o sucesso de uma empresa está dependente da capacidade que os seus responsáveis e funcionários têm de tomar decisões acertadas em tempo útil. Em geral, os *dashboards* podem ser utilizados para monitorizar o desempenho de uma empresa, tanto a nível global como a nível individual, definir estratégias de marketing, analisar tendências, entre outros. Nesta dissertação pretendeu-se investigar a utilização de *dashboards* em sistemas de processamento analítico, abordando desde o seu desenho até à sua implementação e exploração prática. Complementarmente, de forma a demonstrar a utilidade e vantagens desse tipo de instrumentos, procedeu-se à implementação de um sistema piloto, incorporando na sua estrutura uma coleção de *dashboards* providos de mecanismos de auto-adaptabilidade aos requisitos dos utilizadores.

Abstract

Dashboarding - Design and Implementation of Analytical Panels

Nowadays, thanks to the existing high computational and graphical capabilities, we can endow systems with analytical processing tools for information visualization and manipulation that are very attractive and easy to use, especially when using dashboards for the purpose. Dashboards make the interaction with the information from an analytical processing system more interactive and effective, mainly due to the modularity inherent to its graphical components and their innate quality of representing the information graphically. Modularity is also an important characteristic since it allows modifying the system using only mouse clicks, whereas the graphical representation of the information facilitates its analysis and internalization (Few, 2006a). These qualities, among others, make dashboards a central tool to the analysis of information and support for decision making within a company, always bearing in mind that the success of a company is dependent on the ability of its managers and employees have to make good decisions in a timely manner. In general, dashboards can be used to monitor the performance of a company, both globally and individually, define marketing strategies, analyze trends, among others. This thesis aimed to investigate the use of dashboards in analytical processing systems, covering from design to implementation and practical exploration. In addition, to demonstrate the utility and advantages of this type of instruments, it was implemented a pilot system incorporating in its structure some dashboards provided with mechanisms of self-adaptability to the requirements of the users.

Índice

Introdução	1
1.1 Informação e <i>Dashboarding</i>	1
1.2 Auto-Adaptabilidade	2
1.3 Motivação e Objetivos	4
1.4 Estrutura da pré-dissertação	6
<i>Dashboarding</i>	9
2.1 Visualização de Dados	9
2.2 O que são <i>Dashboards</i>	10
2.3 Componentes	11
2.4 Categorização	15
2.4.1 Divisão defendida por Stephen Few	15
2.4.2 A abordagem sugerida por Shadan Malik	16
2.5 Vantagens	17
2.6 Funcionalidades	19
2.7 Desenvolvimento de um Dashboard	24
2.7.1 Planeamento	24
2.7.2 Implementação	26
2.8 Personalização	26
2.8.1 Personalização de dados	27
2.8.2 Personalização de interfaces	28
2.9 Auto-adaptabilidade	29
2.9.1 Algoritmos	30
2.9.2 Problemas e especificidades	32
Um Caso Prático de Aplicação	35

3.1	Introdução ao tema	35
3.2	Ferramentas Utilizadas	36
3.3	Composição dos <i>Dashboards</i>	36
3.4	Funcionamento Geral	42
3.5	Recolha dos Perfis de Utilização	47
3.6	Fonte De Dados	51
3.7	Apresentação do Caso de Estudo	53
3.8	Serviço de Reestruturação	55
3.8.1	Top-K Queries	55
3.8.2	Cadeias de Markov	56
3.8.3	Regras de Associação	60
	Conclusão e Trabalho Futuro	67
4.1	Conclusões	67
4.2	Trabalho Futuro	69
	Bibliografia	71

Índice de Figuras

Figura 2.1 – Exemplo de um Geo Map utilizando a ferramenta Google Charts	13
Figura 2.2 - Secção de um <i>dashboard</i> exemplo da galeria de demos da empresa Dundas.....	14
Figura 2.3 - Secção de um <i>dashboard</i> disponibilizado pela Google Finance	14
Figura 2.4 – Representações gráficas do Quarteto de Anscombe (Wikipedia, 2013)	18
Figura 2.5 - Quarteto de Anscombe e respetivas estatísticas (Wikipedia, 2013).....	18
Figura 2.6 - Linha temporal configurável criada utilizando a ferramenta Google Charts.....	21
Figura 2.7 - Gráfico de barras horizontais criado utilizando a ferramenta Google Charts	22
Figura 2.8 - Gráfico de barras verticais criado utilizando a ferramenta Google Charts	23
Figura 3.1 – Organização de um <i>dashboard</i> do sistema	37
Figura 3.2 - Exemplo de um componente utilizando um gráfico de sectores.....	38
Figura 3.3 - Exemplo de um componente utilizando um gráfico de barras.....	39
Figura 3.4 - Exemplo de um componente utilizando um gráfico de colunas.....	39
Figura 3.5 - Exemplo de um componente utilizando um gráfico de linhas	40
Figura 3.6 - Exemplo de um componente utilizando um gráfico de áreas em escada	40
Figura 3.7 - Exemplo de um componente utilizando um gráfico geográfico	41
Figura 3.8 - Exemplo de um componente utilizando uma tabela de dados	41
Figura 3.9 - Exemplo de caixa de informação de um componente	42
Figura 3.10 - Página de <i>login</i>	43
Figura 3.11 – A página principal do sistema de <i>dashboarding</i>	43
Figura 3.12 – Uma lista de configurações de painéis de <i>dashboarding</i>	44
Figura 3.13 – Gravação de uma configuração a pedido do utilizador	45
Figura 3.14 - Alteração da interrogação MDX	46
Figura 3.15 - Componente aumentado	47
Figura 3.16 - Alteração da representação gráfica de um componente.....	47
Figura 3.17 - Esquema lógico da base de dados que contém a informação relativa à utilização do sistema por parte dos utilizadores e as configurações dos diferentes <i>dashboards</i>	48

Figura 3.18 - Exemplo de registos que representam ações do utilizador.	50
Figura 3.19 - Exemplo de uma configuração de um <i>dashboard</i> associada a uma reestruturação. ..	50
Figura 3.20 - <i>Dashboard</i> resultante do processo de reestruturação utilizando o algoritmo <i>Top-K Queries</i>	56
Figura 3.21 - Porção do grafo gerado pelo algoritmo de cadeias de Markov.....	58
Figura 3.22 - Cadeia de Markov	59
Figura 3.23 - Exemplo de um <i>dashboard</i> resultante do processo de reestruturação utilizando o algoritmo de cadeias de Markov.....	60
Figura 3.24 - Exemplo de um <i>dataset</i> utilizado para o cálculo de regras de associação.....	62
Figura 3.25 - Exemplo de interpretação de um <i>dashboard</i> construído utilizando regras de associação.....	64
Figura 3.26 - Exemplo de um <i>dashboard</i> resultante do processo de reestruturação utilizando o algoritmo de regras de associação	65

Índice de Tabelas

Tabela 3.1 - Tecnologias e ferramentas utilizadas no desenvolvimento dos diversos componentes	36
Tabela 3.2 - Descrição das relações usadas para alimentar o cubo "Sales"	51
Tabela 3.3 - Descrição das dimensões do cubo "Sales"	52
Tabela 3.4 - Descrição das métricas do cubo "Sales"	53
Tabela 3.5 - Configurações utilizadas na geração de perfis de utilização	54
Tabela 3.6 - Padrões de alterações introduzidos nos dados gerados.....	54
Tabela 3.7 - Contagem das ocorrências das ações de um utilizador.....	55
Tabela 3.8 - Regras de associação geradas para o caso de estudo	63

Siglas e Acrónimos

API - Application Programming Interface

BA - Business Analytics

BI - Business Intelligence

DW - Data Warehouse

JADE - Java Agent DEvelopment

ML - Machine Learning

OLAP - Online Analytical Processing

Capítulo 1

Introdução

1.1 Informação e *Dashboarding*

Nos dias que correm é crucial que as empresas possuam a maior quantidade de informação possível e que esta seja passível de ser utilizada como suporte à tomada de decisão bem como à monitorização de qualquer outra atividade empresarial. Esta informação costuma ser extraída dos dados recolhidos ao longo dos anos de operacionalidade da empresa, sendo que não é extraordinário que esta informação atinja dimensões de grandeza superior. É portanto pertinente referir que a grande dimensão dos dados, aliada ao facto de estes estarem normalmente distribuídos por sistemas heterogéneos e distribuídos, fazem com que a sua análise sem recurso a uma ferramenta de análise se torne frequentemente impraticável.

Hoje, a necessidade de consolidar a informação extraída a partir de diversas fontes de informação é comum na maioria das empresas. Tal circunstância promoveu os sistemas analíticos a uma posição de relevo no seio de qualquer empresa que se preocupe com a evolução dos seus processos de negócio e com a sua posição no mercado, perante os seus concorrentes. Estes sistemas permitem armazenar a informação de uma forma mais adequada para a satisfação atempada dos vários requisitos que os agentes de decisão usualmente apresentam no desenrolar das suas atividades, disponibilizando serviços e formatos de apresentação, que se adequam de uma forma mais conveniente aos ambientes de negócio e, em particular, aos de tomada de decisão (Slaughter, 2004). As estruturas que os acolhem e mantêm, designadas de forma simplista por cubos, conjugadas com um conjunto de operações que permitem navegar sobre elas, por

exemplo através de operações de sumarização¹ e de desagregação², fornecem uma visão multidimensional sobre os nossos cenários de análise, de forma mais ou menos detalhada, conforme as necessidades do momento. Mas, apesar da sua sofisticação e poder, só por si, estas estruturas e mecanismos não chegam. É necessário, também, que existam meios muito amigáveis para a sua utilização, que propiciem plataformas de exploração de fácil aprendizagem e requerendo esforços mínimos.

Interfaces intuitivos (e produtivos) são claramente ambicionados por qualquer agente de decisão, que não quer despende muito tempo com processos de aprendizagem de ferramentas, nem dispensar muito tempo a realizar tarefas secundárias, como por exemplo a configuração de eixos de análise ou de formatos de apresentação. Isto, quando possível, permitirá ao utilizador focar-se no que realmente importa: a análise da informação e o processo de tomada de decisão. Uma das ferramentas que evoluiu neste sentido e que tem vindo a ganhar cada mais popularidade, principalmente nos últimos anos: o *dashboard*.

A simplicidade e clareza com que um *dashboard* apresenta e transmite a informação que o utilizador necessita para realizar o seu trabalho, no espaço de apenas um monitor, tornou-os uma ferramenta de comunicação de informação muito poderosa, partindo do princípio que estes foram corretamente projetados. Apesar da popularidade dos *dashboards* e da sua enorme utilização nos dias que correm, o máximo potencial da maioria dos *dashboards* nunca é atingido graças ao seu *design* defeituoso ou incompleto (Few, 2006d). Para se criar um *dashboard* de qualidade igual ou superior à esperada é necessário não só perceber como se deve organizar graficamente o mesmo, quais são os componentes gráficos mais indicados para a informação que se pretende representar e porquê, mas também saber definir o seu âmbito e conhecer as necessidades dos utilizadores finais. Só assim é possível desenvolver um *dashboard* capaz de satisfazer as expectativas e necessidades dos utilizadores.

1.2 Auto-Adaptabilidade

A adaptação automática às preferências dos utilizadores é uma característica cada vez mais presente nos sistemas informáticos desenvolvidos na atualidade, facto este que se deve principalmente ao enorme crescimento da informação disponível através de meios de comunicação

¹ Operações de sumarização são conhecidas na literatura inglesa como operações *roll-up*.

² Operações de desagregação são conhecidas na literatura inglesa como operações *drill-down*.

como a *internet*. Para que esta adaptação automática seja possível é necessário que os sistemas tenham um conhecimento prévio das necessidades e das preferências dos seus utilizadores. A construção desses perfis de utilização é normalmente realizada ativamente através da apresentação de questionários (Billsus & Pazzani, 2000), cujo preenchimento usualmente não é obrigatório, ou da recolha de *feedback* solicitando ao utilizador que classifique a utilidade ou o seu grau de satisfação em relação a uma funcionalidade ou ferramenta do sistema (Mezhoudi, 2013).

Apesar destas técnicas funcionarem, elas são relativamente limitadas e requerem que o utilizador esteja disponível para colaborar com o sistema, o que nem sempre é verdade. O tempo que um utilizador comum está disposto a perder na construção de *feedback* ou no preenchimento de questionários não é muito, e a simples solicitação para o fazer pode provocar o seu descontentamento. Uma outra forma de recolher informação sobre os utilizadores é através da captação de informação relativa à utilização do sistema. Desta forma é possível determinar traços dos perfis dos utilizadores, como, por exemplo, quais as funcionalidades que mais utilizam ou quanto tempo despendem em cada área do sistema, sem que seja necessário perguntar diretamente ao utilizador. É, no entanto, necessário ter em conta que a informação recolhida utilizando este tipo de técnicas é em parte ambígua e pode não ser correta, sendo os resultados da sua utilização previsões ou estimativas da realidade. Após a construção dos perfis de utilização dos utilizadores é então necessário desenvolver mecanismos que utilizem essa informação para personalizar o sistema de acordo com as preferências dos utilizadores. Uma vez que o processo de interpretação dos perfis não é determinístico, surge a oportunidade de integrar algoritmos de *Machine Learning* (ML) no mesmo.

A ML pode ser vista como um conjunto de técnicas que permitem a aquisição de informação estruturada a partir de dados brutos de uma fonte dados (Witten et al., 2011b). A informação obtida através da aplicação de algoritmos de ML aos perfis de utilização seria então passível de ser utilizada no processo de personalização do sistema.

As áreas da Informática em que já existem aplicações práticas de adaptações automáticas de sistemas, mais concretamente adaptações ao utilizador, são por exemplo as áreas:

- Da saúde. O elevado grau de complexidade que a informação relacionada com a saúde pode ter, faz com que adaptar a informação transmitida de acordo com as necessidades e capacidades do público-alvo conduza a uma melhor percepção e absorção dessa informação (Hirst et al., 1997).

- Dos transportes. Ajudar um utilizador a determinar o percurso que deve percorrer, para chegar ao seu destino é uma das aplicações práticas que recentemente tem despertado algum interesse. As aplicações *Web* de mapas Google³ e Bing⁴ são dois exemplos de aplicações desse tipo.
- Das vendas e serviços. O sitio Web da Amazon⁵ é um dos casos práticos mais conhecidos de lojas *online*, em que os utilizadores são presenteados com uma lista de produtos que podem ter interesse para os mesmos. Esta lista de produtos é compilada recorrendo à análise das preferências demonstradas pelo utilizador através das suas ações, como por exemplo as compras que este efetuou ao longo dos tempos, bem como através do agrupamento dos utilizadores em grupos com preferências similares, de modo a ser possível inferir as preferências de um utilizador através das preferências de um utilizador com preferências semelhantes. As potencialidades destes mecanismos são enormes não só para a entidade que os disponibiliza, mas também para os utilizadores finais. Analisando o caso apresentado em particular, a capacidade de apresentar produtos de interesse para o utilizador que de outra maneira poderiam passar despercebidos aumenta a probabilidade de realização de uma venda, beneficiando não só a entidade que oferece o produto, mas também o utilizador do sistema que encontrou com mais facilidade o produto desejado.

Muitas outras áreas têm aplicações interessantes destes mecanismos, sendo a área de sistemas de *dashboarding* uma área pouco explorada nesta vertente mas que apresenta enormes possibilidades de sucesso, muito graças à importância do papel que o utilizador tem neste tipo de sistemas.

1.3 Motivação e Objetivos

Os *dashboards* têm um papel fundamental na monitorização de uma empresa e no auxílio à tomada de decisão, já que permitem que um individuo, por exemplo um gestor ou um diretor de vendas, sem grandes conhecimentos relacionados com extração e manipulação de informação proveniente de diversas fontes digitais, como bases de dados ou *data warehouses*, obtenha de forma fácil e rápida a informação que necessita para o suporte das suas ações de tomada de decisão. Essa informação é disponibilizada ao utilizador através, maioritariamente, de componentes gráficos, componentes estes com os quais o utilizador, que se assume ter conhecimentos de

³ Google - <https://maps.google.pt/maps>

⁴ Bing - <http://www.bing.com/maps/>

⁵ Amazon - <http://www.amazon.com>

gestão ou análise de informação, está mais familiarizado, atendendo ao facto de saber lidar com este tipo de representações no dia-a-dia. A pronta disponibilização de qualquer peça de informação disponível na empresa, aliada à sua fácil interpretação e manipulação, tem consequências satisfatórias, como por exemplo:

- A redução do tempo de deteção de situações adversas.
- A redução do tempo gasto a realizar tarefas secundárias e repetitivas, como preparar a informação a ser analisada.
- A informação analisada está atualizada.
- Havendo condições para isso, a informação de qualquer departamento pode ser disponibilizada em tempo real, facilitando a integração interdepartamental (Pauwels et al., 2008).
- A informação está disponível em diferentes níveis de detalhe.

Estes fatores podem ditar o sucesso de uma empresa no presente mercado empresarial, em que cada vez mais é necessário saber, em tempo útil, quando e como agir, sendo que o fator tempo é particularmente importante. Muitas empresas falham não pela falta de informação, mas sim pela falta de informação em tempo útil - a informação acaba por chegar, mas demasiado tarde.

Existe ainda um outro fator de relevo relativo aos sistemas de painéis analíticos, que é o facto de estes necessitarem de uma certa adaptação ao utilizador. Cada utilizador utiliza um diferente subconjunto das funcionalidades oferecidas pelos componentes gráficos do sistema, uma vez que esse subconjunto pode variar consoante o tipo de análise em curso. Apesar da maioria dos sistemas de painéis analíticos permitir uma configuração mais ou menos abrangente dos componentes gráficos, esta configuração é manual, o que faz com que seja necessário que o utilizador perca algum tempo a realizá-la. Existe ainda a possibilidade de existirem artefactos gráficos que o utilizador não utiliza mas que estão presentes nos painéis analíticos. A sua eventual remoção aumentaria o foco do utilizador naquilo que é realmente importante.

Foi seguindo ao longo da linha desta ideia que se desenvolveu um sistema piloto que, para além de demonstrar a utilidade e versatilidade dos diversos tipos de *dashboards* mais típicos e que usualmente estão incorporados em sistemas analíticos, fosse capaz de se adaptar de forma automática aos requisitos de análise que os utilizadores fossem revelando ao longo do tempo. Para isso, no decorrer da elaboração deste trabalho de dissertação estudaram-se os diferentes tipos de painéis que têm sido desenvolvidos ao longo do tempo, bem como a forma como são incorporados

num sistema de processamento analítico, de modo a absorver o conhecimento inerente às implementações que se encontram no mercado. Após esse trabalho realizado, definiu-se uma forma de captar a informação envolvida nos diversos processos de exploração realizados pelos utilizadores numa dada plataforma analítica e, a partir daí, fez-se a criação de um sistema de personalização de *dashboards*, que fosse capaz de refletir as preferências e os hábitos de exploração dos utilizadores em configurações personalizadas para um qualquer painel de monitorização de dados.

1.4 Estrutura da pré-dissertação

Neste primeiro capítulo foi feita uma pequena introdução aos sistemas de *dashboarding* com vista a perceber o porquê do aparecimento destes sistemas e o porquê de serem tão populares. Foi ainda abordada a questão da auto-adaptabilidade dos sistemas às necessidades dos utilizadores, concluindo-se esta introdução com a exposição dos motivos que levaram ao aparecimento desta dissertação, bem como os objetivos que se pretenderam alcançar com a mesma.

No capítulo seguinte, capítulo 2, foi realizado um estado da arte dos sistemas de *dashboarding*. O capítulo começa com uma pequena história da visualização de dados e dos *dashboards*, que nos permite ter uma noção da relevância que estes foram ganhando ao longo do tempo e dos acontecimentos que levaram à sua proliferação. Em seguida foram avaliados os diferentes componentes que podem integrar usualmente um *dashboard*, a fim de repartir a complexidade dos mesmos de modo a realizar uma análise mais detalhada. Tal avaliação fez surgir a necessidade de estudar algumas categorizações de *dashboards* com o objetivo de os dividir em pequenas categorias de forma a nos permitir perceber-los melhor, dentro do contexto em que cada um se insere. São ainda apresentadas as vantagens de utilizar um sistema de *dashboarding* para realizar operações, tanto de tomada de decisão como de monitorização de uma empresa ou organização, e é feita uma exposição e análise das funcionalidades que tornam os *dashboards* tão importantes no processo de visualização de dados. O estudo dos *dashboards* tradicionais é concluído com uma análise das fases de planeamento e de implementação de um *dashboard*, no qual são apresentadas também algumas boas práticas a seguir. Este estudo é então complementado com um avaliação do universo de possibilidades de personalização que podem ser introduzidas num *dashboard* e como é que podemos tornar essas personalizações automáticas, de modo a permitir ao utilizador usufruir das mesmas, sem que para isso tenha que se preocupar muito com elas, através da utilização de mecanismos de *Machine Learning*.

No capítulo 3 é então apresentado o sistema piloto desenvolvido para demonstrar não só as potencialidades gerais de um *dashboard*, mas também algumas das possibilidades relacionadas com a sua auto-adaptabilidade aos seus domínios de exploração. Começa-se, assim, por fazer uma breve introdução ao tema e por apresentar as ferramentas que foram utilizadas durante o processo de desenvolvimento do sistema. Em seguida é explicado como são compostos os *dashboards* do sistema e ainda quais as possibilidades existentes no que diz respeito a componentes gráficos. Para completar a apresentação do sistema é explicado como se deve proceder para iniciar a utilização do mesmo, sendo então vistas todas as suas funcionalidades de modo a explicar o propósito de cada uma, bem como a forma e quando se devem utilizar. Uma vez apresentado o sistema é explicado o mecanismo de recolha de perfis de exploração, mecanismo este fundamental aos seus processos de reestruturação. Depois é apresentado o *data warehouse* e o cubo de dados que foram utilizados no sistema e é explicado como se procedeu à criação do perfil de exploração do utilizador exemplo, que será usado como caso de estudo ao longo da análise dos mecanismos de reestruturação. O capítulo conclui-se então com uma análise individual de cada um dos mecanismos de reestruturação implementados, nomeadamente os mecanismos baseados nos algoritmos Top-K Queries, cadeias de Markov e regras de associação.

Para finalizar, no capítulo 4, foram expressos alguns comentários sobre o estudo realizado, bem como sobre o sistema desenvolvido, e foram deixados ainda algumas ideias e diretrizes sobre o trabalho que poderá ser realizado futuramente com base no trabalho que foi desenvolvido no âmbito desta dissertação.

Capítulo 2

Dashboarding

2.1 Visualização de Dados

A utilização de meios visuais para representar informação é uma prática que está presente há centenas de anos, tanto a nível académico como a nível científico e empresarial (Blewett, n.d.). Foi no fim do século 18, inícios do século 19, que William Playfair criou alguns dos diagramas mais usados na atualidade na representação de informação, nomeadamente o gráfico de barras, o gráfico de linhas e o gráfico de sectores ou circular. Apesar de tudo isso, o seu trabalho não teve grande impacto enquanto vivo (Spence, 2004). Só depois de falecer é que o valor do seu trabalho começou a ser reconhecido. Atualmente William Playfair é conhecido pelo seu enorme contributo para o desenvolvimento de técnicas gráficas, sendo para muitos o pai da visualização de dados.

Nos anos que se seguiram apareceram outros tipos de representações gráficas, como o gráfico de dispersão criado pelo próprio William Playfair. Porém foi o Dr. John Snow que pela primeira vez demonstrou a enormíssima importância da utilização de representações visuais da informação ao criar um dos primeiros mapas de visualização de dados, a fim de determinar a causa da epidemia de cólera de 1854 (Wallis, 2005). Esta técnica permitiu ao Dr. John Snow determinar que a fonte da cólera era o abastecimento de água permitindo a intervenção das entidades responsáveis, salvando um número incalculável de vidas. Tal catástrofe nunca teria sido evitada se o Dr. John Snow não tivesse recorrido à utilização de representações visuais da informação de que dispunha, uma vez que não seria possível interpretar em tempo útil essa informação utilizando as tradicionais técnicas de representação textual da informação.

Poucos anos se seguiram até que outro profissional ligado à saúde, a enfermeira e estatística Florence Nightingale, utilizou os gráficos circulares para mostrar que a principal causa de morte dos soldados Ingleses durante a Guerra da Crimeia era a falta de higiene nos hospitais utilizados para tratar os feridos em combate (Allen, 2010), com o objetivo de persuadir a Rainha Vitória a melhorar as condições hospitalares. Mais tarde, em 1861, Charles Minard desenvolveu um gráfico mapeando a trágica marcha de Napoleão para Moscovo (Corbett, n.d.), gráfico este que foi classificado por Edward Tufte (1983), no seu conceituado livro "*The Visual Display of Quantitative Information*", como sendo possivelmente o melhor gráfico estatístico de sempre. Muitos feitos relacionados com a representação visual da informação se seguiram, mas foi apenas no século 20 que o professor de estatística John Tukey (1977), da universidade de Princeton, mostrou o verdadeiro poder de utilizar a visualização de dados como meio de explorar e fazer sentido dos dados, tendo publicado em 1977 o livro "Exploratory Data Analysis" em que apresentou pela primeira vez o diagrama de caixa (Few, 2007). Nos anos que se seguiram vários indivíduos com interesses na visualização de dados lançaram obras literárias que abordam vários dos seus aspetos, sendo o livro, já acima referido, "*The Visual Display of Quantitative Information*" de Edward Tufte uma das obras mais revolucionárias.

Já nos últimos anos, a visualização de dados tem tomado um lugar fulcral nas áreas de *Business Intelligence* (BI) e de *Business Analytics* (BA), levando ao surgimento de inúmeras empresas especializadas no desenvolvimento de ferramentas de visualização de dados, nomeadamente ferramentas de *dashboarding*. O problema é que muitas destas ferramentas, apesar de serem extremamente agradáveis e de parecerem bastante úteis, eram na realidade difíceis de usar e oferecia pouca utilidade. Só com o estabelecimento da visualização de dados como uma área de estudo nas universidades, é que começaram a surgir ferramentas comerciais com um grau de qualidade aceitável ou até mesmo bom (Few, 2007). Foi então que os *dashboards* se destacaram como sendo uma das técnicas de visualização de dados mais eficiente e popular. A possibilidade de visualizar diversos aspetos da informação em simultâneo abriu as portas a inúmeras oportunidades no mundo empresarial em termos de análise e monitorização de uma empresa.

2.2 O que são *Dashboards*

Os *dashboards* são ferramentas que se distinguem na área de visualização de dados muito graças às suas características gráficas e às possibilidades que oferecem em termos de exploração e de análise de informação. Segundo Stephen Few (2004a):

"Um dashboard é uma representação visual das informações mais importantes necessárias para atingir um ou mais objetivos, consolidada e organizada num único ecrã de modo a que a informação possa ser monitorizada rapidamente".

Por outras palavras, um *dashboard* é um painel que cabe num só ecrã e que permite visualizar de uma só vez toda a informação necessária à monitorização e à análise de um processo ou de uma empresa. Estes permitem ter uma visão da situação de uma empresa ou de um processo ao longo do tempo, tanto a nível global, como a um nível particular, com maior detalhe, facilitando assim processos de tomada de decisão, em termos estratégicos, analíticos ou operacionais.

As primeiras ferramentas semelhantes a *dashboards* foram desenvolvidas nos anos 80 e tinham por nome *Executive Information Systems* - estes sistemas não estavam facilmente disponíveis como atualmente estão os *dashboards*, existiam muito poucos exemplares e a sua utilização era muito restrita uma vez que estes só se encontravam no seio dos escritórios de executivos. Para além disso não eram tão sofisticados nem confiáveis como os *dashboards* atuais, muito graças ao facto de conceitos como *data warehousing* e *business intelligence* ainda não estarem suficientemente desenvolvidos para suportar tal inovação (Few, 2006b). Já nos anos 90, com o surgimento da Internet e com o desenvolvimento de áreas como a *business intelligence*, o *data warehousing* e o *online analytical processing*, surgiram os *dashboards* tal como os conhecemos na atualidade, com bases mais sólidas que os seus predecessores no que diz respeito à manipulação e apresentação dos dados.

Inicialmente as empresas e as organizações começaram a desenvolver as suas próprias soluções de *dashboarding* de modo a facilitar a apresentação e a análise dos dados que foram acumulando ao longo dos tempos. Contudo, pouco tempo depois, começaram a surgir empresas especializadas em desenvolver tais produtos e foi, então, que os *dashboards* começaram a estar disponíveis em grande escala. Atualmente os *dashboards* existem em praticamente todo o lado, sendo possível encontrar versões tanto para *desktop*, como versões para ambientes *Web*. Existem ainda versões *mobile*, nas quais o desafio de colocar a informação necessária num só ecrã, como sabemos, é bastante grande graças ao reduzido tamanho dos ecrãs desses aparelhos.

2.3 Componentes

Os *dashboards* são componentes compostos por elementos maioritariamente gráficos, podendo ser divididos em gráficos, tabelas, manómetros, mapas, indicadores de estado e alertas. Os gráficos

são certamente o tipo de componente mais utilizado nos *dashboards*, muito graças à possibilidade que estes oferecem de representar uma grande quantidade de informação num espaço limitado, o que não seria possível utilizando outro tipo de componente, como por exemplo tabelas. Outra característica dos gráficos que os torna tão populares é serem capazes de representar informação que é usualmente facilmente absorvida pelos utilizadores através da utilização de cores, escalas, dimensões, áreas, direções, ângulos, volumes, curvaturas e sombreados (Cleveland & McGill, 1984). Existem inúmeros tipos de gráficos disponíveis, sendo apresentados em seguida alguns dos mais utilizados:

- Gráfico de barras: gráfico que utiliza barras para representar valores. O comprimento das barras é proporcional aos valores que representam. As barras podem ser verticais ou horizontais.
- Gráfico de sectores: gráfico circular dividido em fatias ou sectores. O comprimento do arco do sector é proporcional à quantia que este representa.
- Gráfico de marcadores (Few, 2004b): gráfico que representa uma única medida quantitativa principal, juntamente com medidas complementares que permitam enriquecer essa medida. Tem portanto uma funcionalidade semelhante à de um manómetro, mas ocupam menos espaço e representam mais informação.
- Histograma: gráfico que representa a distribuição de frequências de diversas classes, comumente representado utilizando barras verticais.
- Gráfico de linhas: gráfico onde os vários valores de uma determinada medida, ou métrica, ao longo do tempo são representados por pontos, sendo traçada uma linha que conecta todos esses pontos. Podem existir várias linhas no mesmo gráfico, possibilitando tanto a análise das variações do valor de uma métrica em particular, como a análise comparativa dos valores de várias métricas ao longo do tempo.

As tabelas, apesar de poderem ser mais ou menos sofisticadas ou personalizadas, seguem sempre o mesmo formato, representando a informação através da sua disposição em linhas e colunas. Estas são um componente essencial num *dashboard* quando é necessário representar a informação a um nível de maior detalhe, uma vez que os outros tipos de representações falham neste aspecto em particular.

Por sua vez, os manómetros representam uma única medida principal através da utilização de um ponteiro e um intervalo de valores, podendo ainda utilizar etiquetas para assinalar valores de

referência. Segundo Stephen Few (Few, 2006b), os manómetros não são muito eficazes pois ocupam demasiado espaço para a informação que representam, e se relembrarmos que o objetivo de um *dashboard* é o de apresentar uma grande quantidade de informação em apenas um ecrã, podemos perceber o porquê da sua opinião. Apesar disso estes continuam a ser muito populares nos *dashboards*, muito porque graças às suas propriedades ilustrativas, que acabam por se sobrepor às suas propriedades funcionais, o que leva ao desaproveitamento do espaço disponível para o *dashboard*.

Os mapas têm um papel importante na representação de informação que só tem significado quando associada a uma zona geográfica. Estes servem-se então de etiquetas e marcadores gráficos para representar essa informação, colocando-os sobre a zona geográfica em causa, havendo também versões onde os marcadores são substituídos por cores e uma escala, tal como é possível verificar na Figura 2.1.

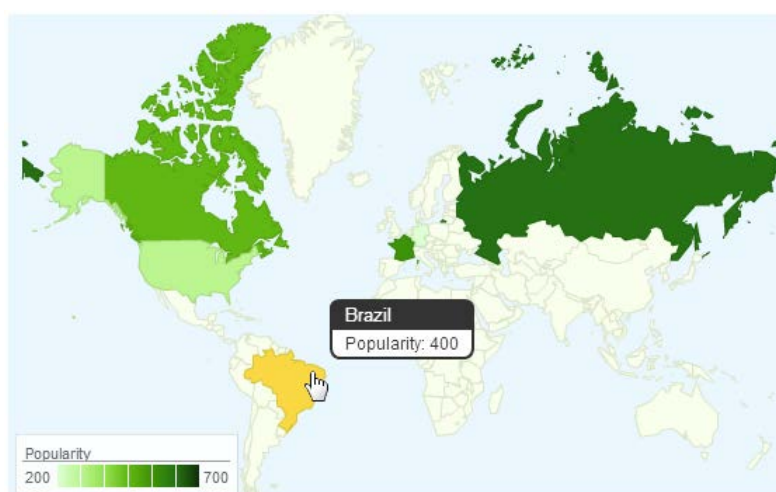


Figura 2.1 – Exemplo de um Geo Map utilizando a ferramenta Google Charts

Os indicadores de estado representam o estado de uma medida em particular comparativamente a um dado objetivo (Microsoft Corporation, 2010?). Estes são muito utilizados para evidenciar tendências, tanto positivas como negativas ou neutras, como é possível observar na Figura 2.2.

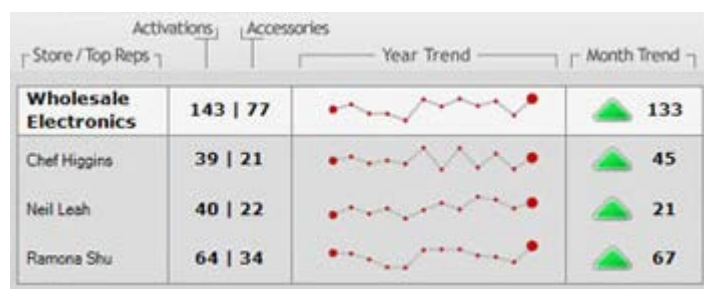


Figura 2.2 - Secção de um *dashboard* exemplo da galeria de demos da empresa Dundas⁶

Os alertas são um mecanismo utilizado nos *dashboards* com o objetivo de alertar, como o seu próprio nome indica, o agente de decisão da ocorrência de uma exceção ou do realce de certos dados (Gemignani, 2008).



Figura 2.3 - Secção de um *dashboard* disponibilizado pela Google Finance⁷

Um alerta é muitas vezes associado a um acontecimento negativo. Porém isso não é necessariamente verdade. Um alerta é tão importante quando é utilizado para sinalizar um acontecimento negativo como quando é utilizado para indicar uma situação favorável. Na Figura 2.3 podemos ver um bom exemplo da utilização de alertas em *dashboards* para assinalar acontecimentos, que tanto podem ser favoráveis como desfavoráveis. Nesta os alertas representam notícias e encontram-se posicionados ao lado de um gráfico de ações, bem como indicados no próprio gráfico, através da utilização de etiquetas com letras, permitindo assim situar facilmente as notícias no tempo.

⁶ Dundas *dashboards* exemplos - <http://www.dundas.com/dashboard/online-examples/>

⁷ Google Finance - <http://www.google.com/finance?q=microsoft>

2.4 Categorização

Graças à miríade de diferentes componentes existentes que podem constituir um *dashboard*, bem como à enorme quantidade de informação distinta, o número de *dashboards* que podem ser produzidos é inimaginavelmente grande, o que obriga a definir algum tipo de agrupamento. Nas seções que se seguem vão ser apresentadas as categorizações propostas por Stephen Few e Shadan Malik.

2.4.1 Divisão defendida por Stephen Few

Stephen Few (2006b) no seu livro *“Information Dashboard Design”* dividiu os *dashboards* em três categorias, mais concretamente estratégicos, operacionais, e analíticos.

Os *dashboards* estratégicos têm como propósito oferecer aos seus utilizadores informação relativa aos objetivos estratégicos de longo prazo de uma empresa. Por outras palavras, este tipo de *dashboards* permite monitorizar o progresso de uma empresa relativamente aos objetivos definidos pela mesma. Estes não requerem que os dados sejam atualizados em tempo real, uma vez que o mais importante é saber o que se tem passado e não o que se está a passar (Dundas Data Visualization Inc., n.d.). Como são relativos a objetivos tão globais e fulcrais no seio da empresa, é de esperar que sejam utilizados por gestores seniores e executivos, isto é, pelos indivíduos no topo da hierarquia de comandos, na tomada de decisão relativa à definição de novos objetivos para a empresa, bem como à monitorização dos objetivos já definidos.

Os *dashboards* operacionais, por sua vez, monitorizam atividades ou variáveis que necessitam de uma monitorização em tempo real. É necessário monitorizar cada momento pois é crucial que seja possível detetar situações anómalas o mais rápido possível de modo a intervir imediatamente, reduzindo ao máximo possíveis prejuízos. A interface deste tipo de *dashboards* deve ser o mais simples e intuitiva possível, e o significado das situações detetadas deve ser claro de modo a permitir uma resposta rápida e correta (Few, 2006c). Estes *dashboards* são utilizados a um nível departamental, em contraste com os *dashboards* estratégicos que são utilizados a um nível executivo (Dundas Data Visualization Inc., n.d.).

Por fim temos os *dashboards* analíticos, que são utilizados na análise de dados. Permitem ao utilizador encontrar padrões, tendências e até mesmo relacionamentos entre os dados. Uma das características mais relevantes destes *dashboards* é o facto de permitirem a visualização dos dados segundo vários níveis de detalhe, tornando assim possível tanto a realização de uma análise geral

de uma empresa ou sistema, bem como uma análise meticulosa e detalhada de uma porção da empresa ou sistema em particular. É portanto de esperar que sejam bastante interativos e que possuam funcionalidades de navegação sobre os dados, como as operações de agregação e desagregação de dados.

2.4.2 A abordagem sugerida por Shadan Malik

Em seguida é apresentada uma classificação sugerida por Shadan Malik (2005), que procura agrupar os *dashboards* consoante o tipo de utilização para que foram criados:

- *Dashboards* de desempenho empresarial, que apresentam informação inerente a diversos departamentos e áreas de negócio, bem como propiciam uma visão global da organização.
- *Dashboards* departamentais, que permitem monitorizar métricas que interessam a um departamento em específico.
- *Dashboards* de monitorização de processos ou de atividades, que permitem monitorizar processos de negócio específicos ou atividades generalizadas de negócio.
- *Dashboards* de aplicações, que fornecem métricas definidas numa determinada aplicação, estando normalmente embebidos nessa mesma aplicação.
- *Dashboards* de clientes, que expõem as métricas relevantes ao cliente de uma organização.
- *Dashboards* de fornecedores, que permitem facilitar e monitorizar a colaboração entre os fornecedores e as organizações cliente.

Cada uma destas categorias de *dashboards* sugere um diferente tipo de utilizador, o que faz com que sejam usados nas suas estruturas diferentes tipos de componentes analíticos e de visualização da informação, uma vez que cada tipo de utilizador tem diferentes necessidades, perícias e conhecimento, analisa os acontecimentos de forma muito particular, está habituado a representações de informação personalizadas e necessita de funcionalidades díspares de acordo com a sua forma de atuação.

Os cenários aplicacionais inerentes a cada uma destas categorias são inúmeros. Por exemplo, relativamente aos *dashboards* de monitorização de processos e de atividades, estes podem ser utilizados para monitorizar situações de fraude, controlo de doenças, linhas de montagem, entre muitos outros. Tomando como segundo exemplo os *dashboards* departamentais, estes podem ser

utilizados para monitorizar e auxiliar processos de tomada de decisão em áreas como a dos recursos humanos, do controlo de qualidade e das vendas.

2.5 Vantagens

Os *dashboards* são ferramentas de visualização de dados criadas com o intuito de colmatar as falhas existentes nos processos de análise de grandes quantidades de dados. Nesses processos eram fundamentalmente utilizados *reports* e folhas de cálculo (ou representações tabulares), para representar a informação. Porém sempre foi evidente a fraca escalabilidade destas formas de visualização de dados, visto que o tempo despendido na análise da informação era diretamente proporcional à quantidade de informação analisada. David Adams, administrador sénior da Accenture, disse uma vez que “Muita da informação chega-nos sob a forma de relatórios tabulares, e nós sentimos que absorver a informação, detetar padrões, identificar anomalias, e ver relacionamentos escondidos rápida e efetivamente é um desafio.” (Adams, n.d.). Nessa comunicação, Adams continua realçando que a evolução das ferramentas de visualização de dados, como é o caso dos *dashboards*, foi fundamental para superar as limitações das representações com as quais estavam acostumados a lidar. A facilidade em absorver a informação rapidamente através da utilização de *dashboards* é facilmente justificada pela sumarização que é aplicada sobre a informação, permitindo realizar análises mais rápidas e mais abrangentes da informação disponível, e a partir daí refiná-la consoante as necessidades através da utilização de operações de desagregação de dados. Em relação à detecção de padrões, anomalias e relacionamentos escondidos, esta tornou-se mais fácil (e em alguns casos possível) porque os gráficos utilizados nos *dashboards*, como por exemplo os gráficos de linhas, representam na perfeição tal informação. Para ilustrar uma situação tomemos como exemplo o caso apresentado na Figura 2.4, na qual estão as representações gráficas de quatro conjuntos de dados idênticos, e cujos valores estão apresentados de seguida, na Figura 2.5. Ao analisarmos a representação tabular dos dados, juntamente com os valores das estatísticas dos mesmos, tudo aponta para que os relacionamentos entre os valores sejam iguais em todos os conjuntos de dados, mas quando nos baseamos nas representações gráficas é evidente que afinal não existe qualquer tipo de semelhança entre os relacionamentos dos vários conjuntos de dados.

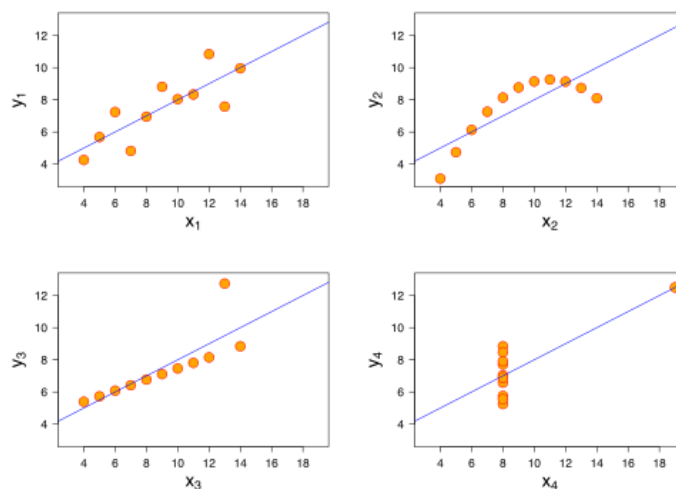


Figura 2.4 – Representações gráficas do Quarteto de Anscombe (Wikipedia, 2013)

I		II		III		IV		Propriedade	Valor
x	y	x	y	x	y	x	y		
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58	Média de x	9 (exato)
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76	Variância de x	11 (exato)
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71	Média de y	7.50 (até duas casas decimais)
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84	Variância de y	4.122 ou 4.127 (até 3 casas decimais)
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47	Correlação entre x e y	0.816 (até 3 casas decimais)
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04	Linha de regressão linear	$y = 3.00 + 0.500x$ (até 2 e 3 casas decimais, respectivamente)
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25		
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50		
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56		
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91		
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89		

Figura 2.5 - Quarteto de Anscombe e respectivas estatísticas (Wikipedia, 2013)

De seguida estão enumeradas as vantagens descritas anteriormente, bem como algumas outras mais, também pertinentes:

- Ter uma visão global e simples de interpretar da situação em análise.
- Navegar nos dados.
- Detetar padrões, tendências, anomalias, e relacionamentos escondidos.
- Ter a informação acessível a todos os empregados da empresa (Malik & Germain, 2009).

- Permitir a uma empresa ou organização manter os seus investidores ou clientes constantemente atualizados (Malik & Germain, 2009).

2.6 Funcionalidades

Graças ao grande poder computacional que hoje em dia está ao alcance de qualquer pessoa, bem como à enorme importância que a informação tem vindo a ganhar - atualmente a informação é dos bens mais valiosos no seio de qualquer empresa -, o investimento por parte das comunidades académicas e científicas, bem como por parte das empresas na área da visualização de dados tem aumentado significativamente. Uma vez que os *dashboards* são uma das ferramentas de visualização de dados mais usadas, se não a mais usada, estes têm vindo a acolher melhorias significativas em relação às suas funcionalidades. Atualmente, a maioria dos *dashboards* são mais do que meras compilações de gráficos, são extremamente interativos e possuem funcionalidades que revolucionaram a análise de informação. Entre as funcionalidades mais importantes que foram desenvolvidas e melhoradas ao longo dos últimos anos, destacam-se as seguintes:

- Os filtros.
- As linhas temporais.
- A possibilidade de alterar em tempo real a representação gráfica da informação.
- As operações de navegação sobre os dados.
- A atualização de dados em tempo real.

Os filtros foram um enorme avanço, em termos de funcionalidades dos *dashboards*. Estes permitem eliminar, ou pelo menos reduzir, a informação desnecessária que se encontra nos elementos de um *dashboard*, tornando-o mais focado no problema em análise e, assim, reduzir as “distrações” inicialmente presentes. Esta simples ação tem, potencialmente, um grande impacto na redução do tempo despendido pelo utilizador na utilização do *dashboard*, bem como na melhoria da análise e interpretação dos dados apresentados uma vez que os dados irrelevantes não são, obviamente, mais uma distração. É importante realçar a necessidade de utilizar os filtros com alguma moderação e cuidado, pois a sua utilização indevida pode levar a que informação importante passe despercebida. Podem haver relacionamentos entre os dados, padrões ou tendências que não são previamente conhecidos, mas que são relevantes para a análise do agente de decisão ou do monitor, que são eliminados do *dashboard* devido à utilização de filtros. Para ilustrar uma dessas situações tomemos por exemplo o caso de uma empresa que vende peças de vestuário. Se o agente de decisão quiser verificar porque é que a venda de camisolas baixou e, se

para isso, filtrar todos os dados presentes no seu *dashboard* de vendas para só incluir dados relativos às camisolas, não irá conseguir verificar que, por exemplo, a quebra na venda de camisolas se deve ao aumento da venda de casacos uma vez que o preço médio de um casaco nas suas lojas, ao contrário do que tinha vindo a acontecer nos últimos anos, era inferior ao preço de uma camisola. Existe uma miríade de outros casos, uns mais problemáticos do que outros. Contudo, este simples exemplo serve para demonstrar a importância de se ser razoável na utilização de filtros que aparentemente podem parecer benéficos numa primeira análise.

Um *dashboard* tem, ou pelo menos devia ter, sempre a noção de tempo a si associado. Quer seja um gráfico de sectores, que representa as vendas do mês de janeiro dos produtos de uma mercearia, quer seja um gráfico de linhas, que apresenta as vendas mensais ao longo do ano de 2012 de uma cadeia de supermercados, a noção de tempo deverá estar sempre presente. Esta noção de tempo pode estar representada no título do gráfico, na sua legenda, ou até mesmo ao longo dos seus eixos. Cada vez mais é dada maior importância aos indicadores temporais, o que é facilmente compreensível se pensarmos que os *dashboards* têm vindo a ser muito utilizados nas áreas de *Business Intelligence* e *Business Analytics*, como interface para sistemas de processamento analítico, nas quais a noção de tempo é crucial. Se pensarmos nos *dashboards* como ferramentas de auxílio à análise de informação, é igualmente perceptível a importância da noção de “quando” é que sucederam as situações em observação.

Em algumas ferramentas de *dashboarding* atuais já existem funcionalidades dedicadas especialmente orientadas para a manipulação da linha temporal dos acontecimentos, como é o caso do exemplo apresentado na Figura 2.6.

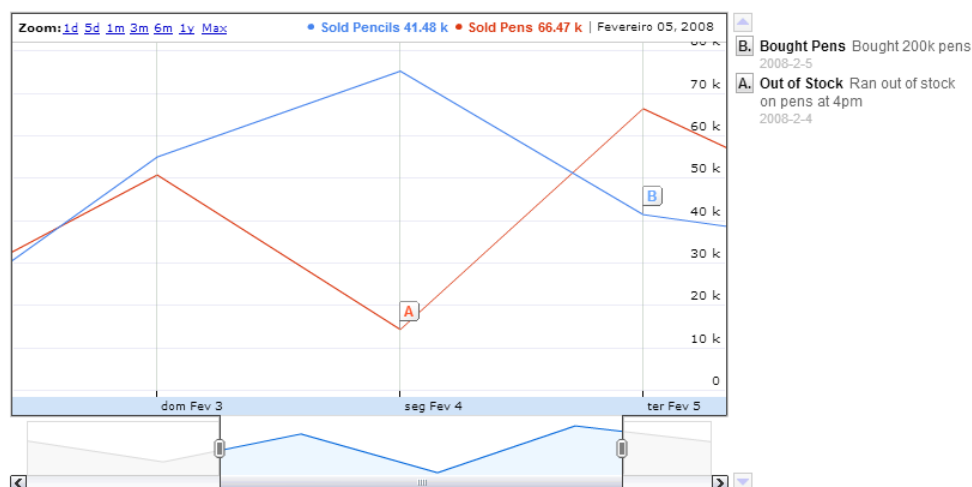


Figura 2.6 - Linha temporal configurável criada utilizando a ferramenta Google Charts

Utilizando esta ferramenta, em concreto, é possível manipular em tempo real a linha temporal apresentada no gráfico, através da utilização da funcionalidade gráfica aí disponível, apresentando no gráfico apenas o intervalo de tempo desejado. Tal funcionalidade podia ser claramente implementada através da utilização de simples filtros sobre as datas, mas tornar-se-ia mais morosa e seria muito mais complicado ajustar frequentemente o intervalo de tempo em visualização. Este é apenas um exemplo da importância que atualmente é dada ao factor tempo no âmbito dos *dashboards*. É, assim, possível perceber o contributo que pequenas funcionalidades como esta têm no suporte à utilização de ferramentas de *dashboarding*.

A possibilidade de alterar em tempo real a representação gráfica da informação que estamos a visualizar permite que o utilizador adapte o *dashboard* às suas necessidades e preferências sem a necessidade de recriar o *dashboard*. O que para muitos pode ser uma funcionalidade supérflua, para outros as suas potencialidades são enormes. Esta característica, desde logo, permite que um *dashboard* seja desenhado e implementado para satisfazer as necessidades de vários utilizadores com diferentes preferências sem que para isso seja necessário negligenciar as preferências de um utilizador em detrimento das preferências de um utilizador com mais influência. Por exemplo, a escolha entre um gráfico de barras (ou gráfico de barras horizontais) e um gráfico de colunas (ou gráfico de barras verticais) é apenas uma questão de gosto uma vez que, como sabemos, a única diferença entre os dois é a orientação das barras, como pode ser verificado através da observação da Figura 2.7 e da Figura 2.8. Ao termos a possibilidade de escolher em tempo real qual dos dois queremos damos a possibilidade a cada utilizador de escolher aquela representação com que se

sente mais confortável, sem que para isso tivéssemos de criar várias versões do mesmo *dashboard*.

Não obstante, é importante referir que a alteração do tipo de representação gráfica de uma peça de informação pode alterar drasticamente o valor dessa informação, podendo mesmo introduzir ou eliminar parte da informação apresentada, uma vez que existe informação inerente à representação gráfica, como é o caso das tendências ou dos relacionamentos entre os dados. Por este motivo é preciso perceber o que cada representação gráfica nos oferece para que seja possível escolher uma representação agradável, mas acima de tudo simples, clara e completa.

A ferramenta Google Charts⁸ é um bom exemplo deste tipo de funcionalidade, graças à separação entre a representação e a definição dos dados que apresenta. Com ela é possível escolher em tempo real a representação gráfica para um conjunto de dados.

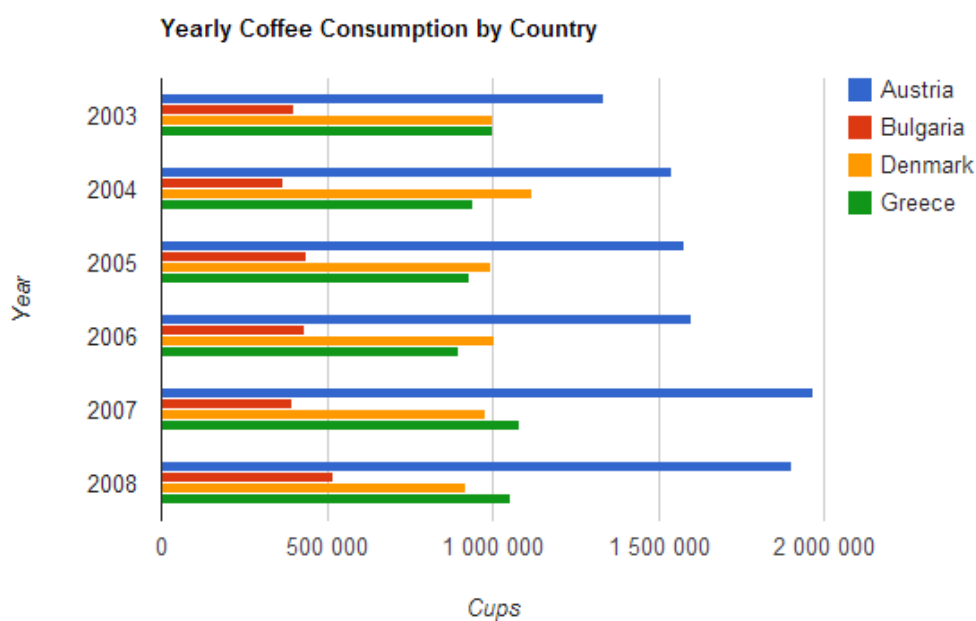


Figura 2.7 - Gráfico de barras horizontais criado utilizando a ferramenta Google Charts

⁸ Google Charts - <https://developers.google.com/chart/>

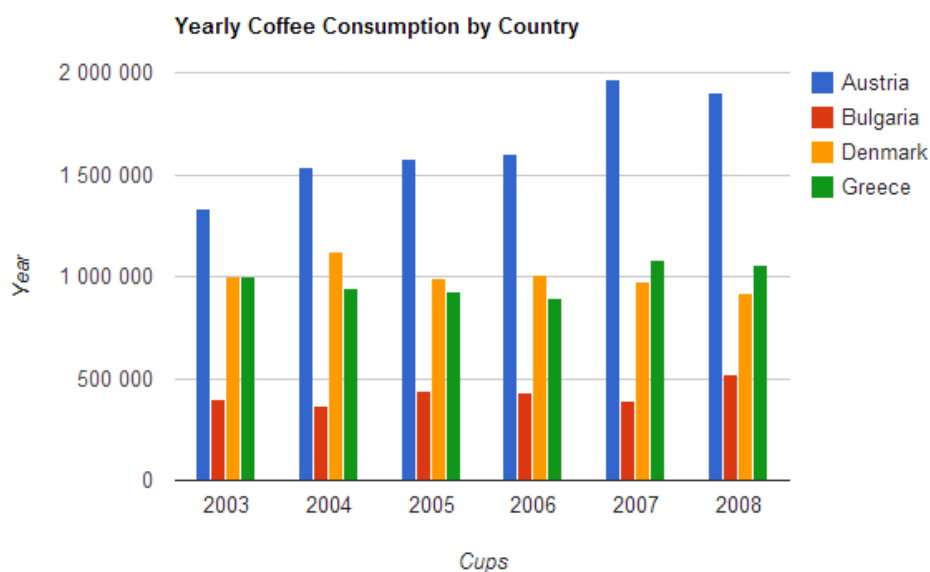


Figura 2.8 - Gráfico de barras verticais criado utilizando a ferramenta Google Charts

As operações de sumarização e de desagregação estão cada vez mais presentes nos *dashboards*, porque são operações fulcrais em qualquer processo de suporte à tomada de decisão. Estes conceitos são fundamentais uma vez que o agente de decisão deve ser capaz de fazer uma análise rápida do panorama geral representado no *dashboard*, bem como de investigar a origem e as causas de possíveis situações adversas ou favoráveis. Seguindo o exemplo da empresa que vende roupa, se o agente de decisão conseguisse detetar que as vendas tinham baixado no último mês, mas não fosse capaz de aplicar operações de desagregação sobre os dados, não seria capaz de verificar se as quebras se registaram em todas as lojas em geral ou se foi uma loja em particular que contribuiu mais para essa quebra, tendo que recorrer a outros meios para determinar a origem do problema. O tempo despendido a tentar encontrar a origem do problema iria ser naturalmente maior do que seria se fosse possível obter essa informação do *dashboard* através da utilização de operações de desagregação de dados. Por outro lado, se apenas a informação das vendas por loja estivesse disponível no *dashboard*, o agente de decisão teria que analisar toda essa informação de modo a saber se existem anomalias, assumindo que nenhuma anomalia é encontrada em grande parte das suas análises, é de esperar um desperdício no tempo despendido a realizá-la uma vez que uma análise do panorama global seria suficiente para tirar as mesmas conclusões.

A atualização dos dados em tempo real é uma operação que consome uma quantidade significativa de recursos e não seria possível a uma grande escala se não fosse o grande poder computacional que os computadores oferecem nos dias de hoje. Esta funcionalidade é essencialmente utilizada

nos *dashboards* de monitorização, em que é fulcral que a informação seja disponibilizada o mais rapidamente possível, de modo a permitir evitar situações de risco ou tirar partido de situações benéficas. Um bom exemplo duma destas situações é um *dashboard* que monitorize a situação de uma bolsa de valores, por exemplo o PSI 20. O mais pequeno atraso na recepção de um valor pode ditar a perda de fortunas – a obtenção dos valores em tempo real oferece uma vantagem enormíssima a um agente de decisão.

2.7 Desenvolvimento de um Dashboard

“Os *dashboards* podem oferecer meios únicos e muito poderosos para apresentar informação, mas raramente provam o seu potencial.” (Few, 2006b). Antes de implementar um *dashboard* é imperativo que se proceda ao seu planeamento. A simplicidade que um *dashboard* sugere leva a que muitas vezes o seu planeamento seja negligenciado, levando ao aparecimento de ferramentas defeituosas e que não satisfazem os objetivos desejados. O maior motivo para se verificarem este tipo de situações é a importância que se dá à aparência em detrimento da utilidade e simplicidade.

Desenvolver um *dashboard* é mais do que colocar sofisticados gráficos e manómetros num ecrã de forma a estes ficarem admiráveis sob um ponto de vista visual, é representar toda a informação necessária ao processo de tomada de decisão ou de monitorização de uma empresa num só ecrã, de maneira a que esta seja facilmente absorvida pelos utilizadores. Para tal é necessário realizar um planeamento cuidadoso. Analisaremos isso na secção que se segue.

2.7.1 Planeamento

Quando se procede ao planeamento de um *dashboard* é fundamental abordar algumas questões chave. Serão essas questões que nos irão ajudar a perceber as necessidades dos futuros utilizadores, bem como as necessidades tecnológicas necessárias para suportar o *dashboard*, permitindo assim que o planeamento do *dashboard* seja realizado o mais corretamente possível.

Quem é a audiência alvo? (Juice Analytics, 2009) Conhecer o público que irá utilizar o *dashboard* é muito importante pois permite determinar informação relevante como:

- O seu grau de instrução, permitindo assim definir com maior precisão o grau de complexidade que o *dashboard* deve apresentar.
- A informação que este espera encontrar.

- Os tipos de representações a que este está habituado e até que ponto está familiarizado com ferramentas de visualização de dados.
- Que informação já sabe, prevenindo a má utilização do espaço do *dashboard*, através da não inclusão da informação irrelevante.

Quais são os dispositivos alvo? Saber qual o tamanho do ecrã dos dispositivos a que o *dashboard* se destina é imperativo para que seja possível selecionar os elementos gráficos adequados, bem como idealizar e projetar a melhor disposição dos mesmos.

Qual o seu propósito? Quando foi abordado o tema da categorização dos *dashboards* foram apresentados vários tipos de *dashboards*, sendo que para cada tipo estava explícito um propósito particular. Após determinar o tipo de *dashboard* torna-se mais simples determinar os seguintes parâmetros chave:

- Nível de detalhe da informação: informação mais ou menos detalhada, com ou sem a possibilidade de aumentar e diminuir o seu detalhe.
- Propósito de utilização: utilização para fins estratégicos, operacionais ou analíticos.
- Janela temporal: informação histórica ou atualizada, com ou sem atualizações em tempo real.
- Âmbito: nível de abrangência da informação.

O planeamento dos componentes do *dashboard* é também uma etapa importante, mas que não exige um planeamento tão intensivo. Desse planeamento fazem parte tarefas como selecionar as cores a utilizar, especificar o posicionamento e orientação das etiquetas num gráfico, ou dos símbolos num indicador de estado, entre outros, podendo este planeamento ser mais ou menos detalhado consoante as necessidades do projeto. Porém, o planeamento de um alerta deve ser relativamente mais cuidado. Segundo Zach Gemignani (2008), a fim de implementar um alerta corretamente, é necessário ter em atenção, entre outros, os seguintes pontos:

- Os parâmetros do alerta devem estar bem definidos. Por exemplo, pode-se definir um alerta para acionar quando as vendas (métrica) mensais (delta) de camisolas nas lojas do norte do país (âmbito) diminuírem 20% (*threshold*), sendo a dimensão em monitorização a dos artigos.
- A granularidade do alerta deve ser bem definida.

- A mudança registada deve ser estatisticamente significativa. Em certas situações um grande aumento ou diminuição percentual não tem muita importância, por exemplo se o valor inicial for demasiado pequeno.
- O alerta deve ser chamativo e agradável.
- A complexidade do alerta deve ser apropriada ao utilizador.
- O alerta deve ajudar o utilizador a decidir o que fazer de seguida.
- O alerta deve ser posto em contexto.

Após realizado o planeamento podemos partir então para a implementação do *dashboard*.

2.7.2 Implementação

A fase de implementação pode ser significativamente simplificada, se para isso for utilizada uma das inúmeras ferramentas existentes no mercado que auxiliam a implementação de *dashboards*, permitindo dessa forma que mesmo indivíduos sem grandes conhecimentos informáticos desenvolvam os seus próprios *dashboards*. Por exemplo, o icCube Web Reporting⁹ é uma ferramenta que permite criar *dashboards* interativos através de um *Web browser*. Utilizando esta aplicação é possível adicionar componentes gráficos aos *dashboards*, bem como ativar operações de aumento de detalhe¹⁰ sobre os dados e permitir a aplicação de filtros sobre os dados. Existem muitas outras ferramentas análogas, como é o caso da Dundas Dashboard¹¹ e InetSoft Style Scope¹², entre outras, que apresentam inúmeras características interessantes e serviços para a criação de sistemas de *dashboarding*. Em geral estas ferramentas oferecem um grau de liberdade na configuração dos componentes gráficos bastante elevado, pelo que servem perfeitamente mesmo quando se pretende efetuar um nível elevado de customização.

2.8 Personalização

Os sistemas computacionais, em particular as suas interfaces, são pensados e desenhados para servir as necessidades de um determinado grupo de utilizadores, composto por indivíduos com diferentes expectativas, preferências e necessidades. Esse grupo de utilizadores pode ser composto pelos funcionários de uma pequena empresa, de uma empresa multinacional, ou até mesmo pelos utilizadores de um determinado sítio *Web*. A dificuldade em desenvolver um sistema

⁹ icCube Web Reporting - <http://www.iccube.com/products/iccube/iccube-web-reporting>

¹⁰ Operações de aumento de detalhe são conhecidas na literatura inglesa por operações *drill-down*

¹¹ Dundas Dashboard - <http://www.dundas.com/dashboard>

¹² InetSoft Style Scope - <http://www.inetsoft.com/products/StyleScope>

que satisfaça todo o grupo de utilizadores, depende não só da quantidade de utilizadores que compõe o grupo, mas também e principalmente da diversidade de elementos do grupo. Dentro de uma pequena empresa é natural que a mentalidade e as necessidades dos diferentes funcionários sejam relativamente similares visto que todos estão inseridos no mesmo meio e trabalham para o mesmo fim, o que facilita o desenvolvimento de um sistema que satisfaça todos os utilizadores. Mas, se pensarmos num sistema desenhado para ser utilizado por utilizadores de diferentes nacionalidades, com diferentes culturas e costumes, como por exemplo um sítio *Web*, esta tarefa torna-se bastante mais complexa. Como tal, salvo em casos excepcionais, não é possível desenvolver um sistema estático que satisfaça a 100% todos os utilizadores finais, surgindo então a necessidade de o tornar flexível, nomeadamente através da inclusão de meios de personalização no mesmo. A ideia de que é possível criar um sistema estático eficazmente perfeito é então posta de parte e surge a necessidade de considerar o utilizador como sendo a chave para o desenvolvimento de um sistema eficaz e capaz de satisfazer as necessidades que levaram à sua criação. No caso particular dos *dashboards* a personalização é uma característica que pode ser introduzida em dois níveis, nomeadamente ao nível dos dados ou da informação apresentada e ao nível da interface.

2.8.1 Personalização de dados

Ao nível dos dados é possível oferecer ao utilizador a possibilidade de seleccionar a informação que este pretenda ver apresentada nos elementos do *dashboard*, sendo que esta seleção pode gerar desde um aumento (ou uma diminuição) do detalhe da informação já presente no *dashboard* até à remodelação completa do *dashboard*. O grau de liberdade sobre os dados que é disponibilizado a um utilizador deve ser devidamente ponderado de modo a assegurar uma correta utilização do *dashboard*, bem como evitar falhas de segurança. É preciso garantir que um utilizador não tem acesso a informação que não lhe compete, sendo para isso necessário limitar o universo de informação a que cada utilizador tem acesso. A implementação destas limitações deve ser estudada de modo não só a evitar o problema de acessos indevidos à informação, como já foi referido, mas também para prevenir uma sobrecarga de informação. A disponibilização de informação irrelevante pode levar à incorreta utilização da mesma (Morris, 2010), podendo originar situações em que informação relevante passe despercebida ou seja acedida com atrasos prejudiciais. É importante ter em mente que por muito capaz que seja o utilizador final, existe sempre a possibilidade de este tomar más decisões na personalização do *dashboard*, pelo que se deve reduzir, sempre que possível, a probabilidade de uma situação destas acontecer.

Para além dos problemas que podem surgir ao nível do utilizador existe ainda a possibilidade de um excessivo grau de personalização levar ao surgimento de problemas a um nível mais global. Quando cada utilizador é capaz de seleccionar os dados que deseja visualizar é possível que a informação retida por cada um dos utilizadores, apesar de se basear na mesma fonte de dados, seja diferente e por vezes até conflituosa.

2.8.2 Personalização de interfaces

Ao nível da interface é possível presentear o utilizador com a possibilidade de personalizar não só a configuração geral do *dashboard*, mas também a configuração de cada um dos seus componentes. As características personalizáveis devem ser definidas consoante a capacidade do utilizador desenhar um *dashboard*, uma vez que uma pequena alteração do mesmo pode reduzir significativamente a sua clareza e eficiência. Em seguida são apresentados alguns exemplos de possíveis personalizações:

- Seleccionar o número de componentes que compõe o *dashboard*.
- Escolher a disposição dos componentes do *dashboard*.
- Definir o tipo de representação gráfica de cada componente do *dashboard*.
- Definir a dimensão dos componentes do *dashboard*.
- Adicionar e remover ferramentas de manipulação dos dados aos componentes do *dashboard*, como por exemplo uma barra de navegação temporal ou uma ferramenta que permita adicionar e remover filtros.

Os erros que podem ser cometidos pelos utilizadores aquando da personalização do *dashboard* são inúmeros e apesar de alguns poderem ser controlados pelo sistema através de notificações ou de regras, é impossível controlar todos os cenários, pelo que a escolha das personalizações a disponibilizar deve ser pensada com cuidado, tendo-se sempre em mente a aptidão do utilizador final. Tais erros são os erros que poderiam ser cometidos durante a criação do *dashboard* uma vez que ao permitir que o utilizador personalize o *dashboard* estamos efetivamente a passar algum do controlo sobre o desenho do *dashboard* para o lado do utilizador. Stephen Few (Few, 2006e), no seu livro "Information Dashboard Design", abordou alguns dos erros mais comumente cometidos no processo de criação de um *dashboard*, como por exemplo:

- Exceder a dimensão máxima recomendada de um ecrã.
- Escolher uma representação gráfica inapropriada para os dados em questão.

- Introduzir uma variedade excessiva de tipos de representações gráficas.
- Dispor os componentes erradamente.

Em suma, quando se pretende tornar um *dashboard* personalizável é necessário ter em consideração inúmeros factores que podem levar à redução da sua clareza e eficácia, mais concretamente aos factores introduzidos pelo utilizador final e derivados da sua mais ou menos baixa aptidão em relação ao desenho de um *dashboard*. É possível introduzir mecanismos que auxiliem o utilizador na personalização do *dashboard*. Por exemplo através da utilização de avisos, recomendações ou tutoriais, o que reduz assim a probabilidade de erro. No entanto, será sempre possível o aparecimento de cenários de erro não contemplados. É portanto imperativo que o nível de personalização introduzido no sistema tenha em consideração as faculdades do utilizador, sempre que possível.

2.9 Auto-adaptabilidade

Em sistemas como os de *dashboarding* em que a quantidade dos dados envolvidos pode ser da ordem de grandeza dos *gigabytes*, por vezes mesmo superior, o utilizador comum incorre com facilidade em erros derivados da falta de capacidade do ser humano de processar com clareza e em tempo útil tamanho universo de dados, o que pode conduzir à incorreta seleção dos dados que devem ser utilizados. Surge então a oportunidade de introduzir no sistema mecanismos de auto-adaptabilidade que permitam auxiliar o utilizador ao longo do processo de preparação do *dashboard*, bem como durante o processo de análise e manipulação dos dados. Estes mecanismos permitem não só reduzir os erros cometidos pelos utilizadores, mas também automatizar o processo de exploração realizado por eles, reduzindo a necessidade de estes realizarem tarefas repetitivas e previsíveis, aumentando a eficiência da utilização do sistema e a redução do tempo desperdiçado a realizar tarefas secundárias. Existe ainda a possibilidade de aplicar mecanismos de auto-adaptabilidade às próprias interfaces dos *dashboards*. Os *dashboards* são componentes que podem ser criados utilizando uma miríade de diferentes elementos gráficos, e caso seja permitido ao utilizador personalizar os mesmos, se for guardado um histórico mais ou menos detalhado dessas personalizações é possível extrair informação relativa às suas preferências sobre as interfaces. É possível detetar quais as representações gráficas preferencialmente utilizadas para representar certos dados, qual a disposição normal dos componentes gráficos, qual o número de componentes gráficos mais indicado, que funcionalidades são ou não utilizadas, entre outras muitas coisas. Com a aplicação destes mecanismos às interfaces pode-se então determinar a

composição mínima e potencialmente mais correta de um *dashboard* para uma determinada situação e utilizador, eliminando assim os artefactos que representam distrações para o utilizador.

Os mecanismos aqui abordados serão baseados nas preferências dos utilizadores, na sua interação com o sistema ao longo do tempo, e como tal é necessário que exista um registo mais ou menos detalhado, consoante a abrangência e complexidade desejada, das ações do utilizador no sistema. Sendo esse pré-requisito satisfeito é possível então passar à seleção de um algoritmo que utilize os perfis de utilizador recolhidos e que melhor se adapte à situação em questão.

2.9.1 Algoritmos

Existem inúmeros algoritmos, nomeadamente na área de *Machine Learning*, passíveis de serem utilizados na análise de perfis de utilização e na consequente detecção de padrões de utilização. Cada um desses algoritmos apresenta diferentes características que o diferencia dos restantes e que o torna mais indicado para cenários aplicativos específicos. Em seguida são apresentadas algumas técnicas que podem ser utilizadas para fazer esse trabalho, nomeadamente:

- *Top-k Queries*.
- *Clustering*.
- Árvores de decisão.
- Cadeias de Markov.
- Regras de associação.
- Séries temporais.

O primeiro algoritmo enunciado, *Top-k Queries*, é o algoritmo mais simples aqui apresentado, e serve para detetar as ações realizadas mais frequentemente por um utilizador em particular. O seu funcionamento é muito simples, as ações do utilizador começam por ser ordenadas consoante a sua frequência e em seguida são seleccionadas as K ações com mais ocorrências. Apesar da simplicidade do algoritmo este não deve ser menosprezado pois a sua utilização pode originar resultados bastante satisfatórios. Para além disso, este pode servir ainda para avaliar comparativamente outros algoritmos que também tenham sido implementados, ou seja, utiliza-lo para definir um patamar mínimo de qualidade, pois se através da utilização de algoritmos mais complexos e pesados computacionalmente não se conseguir obter resultados de maior qualidade então não é recomendável a sua utilização.

Clustering é uma outra abordagem que pode ser utilizada, que se baseia na divisão das ações em grupos. Os grupos criados devem refletir um mecanismo que faz com que as instâncias, neste caso em particular ações, de um grupo tenham uma maior semelhança entre elas do que com as instâncias dos outros grupos (Witten et al., 2011a). Uma das potencialidades desta abordagem é a possibilidade que esta oferece de cruzar as preferências de todos os utilizadores, agrupando-os em grupos de acordo com as semelhanças dos seus padrões de utilização do sistema. Deste modo é possível criar perfis genéricos de utilização que podem ser utilizados não só para complementar o perfil de um utilizador com ações realizadas por utilizadores com perfis semelhantes, mas também para rapidamente traçar um perfil de um novo utilizador.

As árvores de decisão são modelos compostos por um conjunto de regras organizadas hierarquicamente (Motti et al., 2012) e que podem ser utilizados para responder a questões que estejam relacionadas com os dados envolvidos. No caso particular dos *dashboards*, poderiam ser respondidas questões como “Que tipo de representação gráfica deve ser utilizada para representar os dados X?” ou “Qual deve ser o posicionamento do componente A sabendo que o posicionamento do componente B é Y?”. Para além de ser possível trabalhar com a informação relativa a um só utilizador, existe ainda a possibilidade de trabalhar com grupos de utilizadores, possivelmente determinados utilizando técnicas de *clustering*, de modo a realizar previsões mais genéricas. Esta abordagem conjunta das duas técnicas é particularmente útil se a quantidade de informação relativa a um utilizador for diminuta.

As Cadeias de Markov são grafos direcionados onde cada nodo representa um estado e cada ramo representa uma transição entre dois estados, sendo que em cada ramo está representada a probabilidade dessa transição acontecer. A utilização destes modelos estatísticos dá origem a previsões de transições entre estados, sendo as suas potenciais aplicações em sistemas de *dashboarding* em termos de auto-adaptabilidade inúmeras. Podem, por exemplo, ser utilizadas para prever uma sequência de ações de navegação sobre os dados de modo a sugerir um ou mais *dashboards* que incorporem os dados resultantes dessas várias ações, ou então para prever uma sequência de *dashboards* que o utilizador potencialmente irá utilizar.

As regras de associação são uma outra abordagem que permite extrair relacionamentos entre diferentes atributos, que podem representar características do perfil de utilização dos utilizadores. Esta abordagem é bastante promissora pois permite obter muita informação relativa à coexistência de diferentes conjuntos de dados e de diferentes representações gráficas, bem como aos

relacionamentos entre um conjunto de dados e uma representação gráfica em particular. Esta informação permite então adaptar o *dashboard* utilizando os conjuntos de dados mais relevantes, juntamente com as representações gráficas corretas.

Por fim temos as séries temporais que permitem a realização de uma análise temporal dos perfis de utilização dos utilizadores. Utilizando séries temporais é possível detetar padrões temporais de utilização que usualmente fazem parte do quotidiano de um utilizador de um sistema de *dashboarding*, como por exemplo determinar qual o primeiro *dashboard* consultado em cada dia da semana, qual o primeiro *dashboard* consultado depois de almoço, ou qual o *dashboard* que indica o fim de utilização do sistema. Estes são apenas uma pequena parte dos padrões que podem ser obtidos através da utilização de séries temporais, servindo contudo para mostrar a potencialidade deste tipo de abordagem.

2.9.2 Problemas e especificidades

Quando se introduzem mecanismos de auto-adaptabilidade num sistema é necessário ter em mente que o utilizador final pode não estar à vontade com as mesmas (Gena, 2005). Para além disso, é fulcral admitir que esses mecanismos podem também produzir resultados incoerentes e indesejados que podem prejudicar o utilizador. Com base nestas duas premissas há a necessidade de prover estes mecanismos com funcionalidades que permitam ao utilizador tomar o controlo da situação. A primeira medida que deve ser tomada é a introdução de funcionalidades que permitam ao utilizador rejeitar uma adaptação, de adaptar o sistema manualmente e de desligar os mecanismos de auto-adaptabilidade. Deste modo o utilizador sente que tem o controlo total da situação e que não é obrigado a utilizar e a depender destes mecanismos, apesar de os poder utilizar sempre que o desejar fazer. Outra característica importante num sistema que se encontra em constante mudança é a capacidade de manter alguma forma de histórico que permita ao utilizador rever as alterações que foram efetuadas e voltar atrás sempre que necessário, garantindo assim que o utilizador tem toda a informação necessária para reverter alterações indesejadas, mantendo as desejadas. Por fim, sempre que uma adaptação é levada a cabo o utilizador deve ter conhecimento que a mesma sucedeu, e deve ainda ser informado das razões que levaram à adaptação, se assim o desejar (Jameson, 2001), visto que percebendo a origem da adaptação o utilizador terá mais facilidade em aceitá-la.

Existem ainda questões de privacidade relacionadas com o facto de ser necessário registar com um certo detalhe as ações realizadas pelos utilizadores durante a utilização do sistema, o que pode

deixar alguns deles desconfortáveis. É imperativo tomar as devidas medidas de segurança para garantir que essa informação permanece confidencial e ainda que os utilizadores do sistema estão devidamente informados em relação à informação que está a ser recolhida sobre eles, e aos fins para os quais essa informação será utilizada.

Se estes requisitos não puderem ser satisfeitos deve-se reconsiderar a introdução de mecanismos de auto-adaptabilidade em sistemas de *dashboarding*, pois podem prejudicar a utilização do sistema e consequentemente o próprio utilizador. Deve-se, ainda, sublinhar a importância redobrada destes mecanismos de prevenção quando os sistemas de *dashbording* são usados em situações de alto risco, como em situações em que uma má decisão pode prejudicar economicamente uma empresa, a saúde de um paciente, ou até causar um acidente de aviação.

Capítulo 3

Um Caso Prático de Aplicação

3.1 Introdução ao tema

O trabalho prático aqui apresentado foi realizado no decorrer de uma bolsa de iniciação científica financiada pela Portugal Telecom Inovação, SA, de nome **AC2DC – Componentes Analíticos Auto-adaptáveis para *Cloud Based Data Warehouses***. O principal objetivo deste projeto foi o de especificar e implementar componentes analíticos para a exploração de estruturas de dados multidimensionais mantidas em ambientes de data *warehousing* suportados por nuvens computacionais, com especial foco na investigação e implementação de meios que permitam dotar os componentes da capacidade de se adaptarem automaticamente às preferências dos utilizadores.

No decorrer deste capítulo será focada a componente relativa aos *dashboards*, mais concretamente a aplicação *Web* desenvolvida que permite a exploração de dados multidimensionais através da utilização de *dashboards* possuindo a capacidade de se adaptar às preferências dos utilizadores, graças aos mecanismos de auto-adaptabilidade implementados. Desta parte do sistema fazem parte dois grandes componentes, nomeadamente: um servidor OLAP, responsável por satisfazer as interrogações MDX necessárias à criação dos componentes dos *dashboards*, e um sítio *Web* no qual é possível aceder aos *dashboards* e a todas as funcionalidades do sistema. É importante referir que o sistema desenvolvido é composto por um conjunto de agentes de software, pelo que podem ser feitas algumas menções aos mesmos ao longo deste capítulo, em particular. Contudo estes não serão abordados uma vez que se encontram fora do âmbito desta mesma dissertação.

3.2 Ferramentas Utilizadas

Um dos requisitos do projeto era que este fosse realizado recorrendo apenas a ferramentas *open-source*, limitando assim o universo de ferramentas disponíveis, havendo por vezes apenas uma ferramenta indicada para o efeito. Para desenvolver o sistema em geral foi utilizada a linguagem de programação Java¹³, não só por ser uma das linguagens de programação orientada a objetos mais utilizada em todo o mundo (TIOBE Software BV, 2013), mas também graças à disponibilidade apresentada pelas outras ferramentas necessárias à criação do sistema em relação à interação com a linguagem. O sítio *Web* foi desenvolvido utilizando tecnologia Java Servlet¹⁴, a *framework* Jade¹⁵, para implementar os processos de comunicação entre o sistema e a aplicação que tem acesso ao servidor OLAP, uma vez que o sistema é composto por agentes, e ainda a API Google Charts¹⁶ para desenvolver os componentes dos *dashboards*. Já a aplicação responsável por fornecer acesso ao servidor OLAP, bem como tratar da sua manutenção, foi desenvolvida utilizando Java e Jade, sendo que o servidor OLAP utilizado foi o Mondrian¹⁷. Por fim, todas as fontes de dados utilizadas, tanto na aplicação *Web* como no servidor OLAP são PostgreSQL¹⁸.

Componente	Tecnologias e ferramentas utilizadas
Aplicação <i>Web</i>	Java, Java Servlet, Jade Framework, Google Charts API
Aplicação responsável pelo servidor OLAP	Java, Jade Framework
Servidor OLAP	Mondrian
Fontes de dados	PostgreSQL

Tabela 3.1 - Tecnologias e ferramentas utilizadas no desenvolvimento dos diversos componentes

3.3 Composição dos *Dashboards*

Um *dashboard* pode ser composto por um número variado de componentes gráficos, sendo que o número ideal de componentes depende de inúmeros fatores como o tamanho do dispositivo de apresentação, a capacidade do utilizador final de lidar com diferentes quantidades de informação, o tipo de exploração que se pretende realizar, entre outros. No caso prático aqui apresentado

¹³ Java - <http://www.oracle.com/technetwork/java/index.html>

¹⁴ Java Servlet Technology - <http://www.oracle.com/technetwork/java/index-jsp-135475.html>

¹⁵ Jade - <http://jade.tilab.com/>

¹⁶ Google Charts - <https://developers.google.com/chart/>

¹⁷ Mondrian - <http://mondrian.pentaho.com/>

¹⁸ PostgreSQL - <http://www.postgresql.org/>

fixou-se o número de componentes por *dashboard* em seis unidades, de modo a garantir que os *dashboards* consigam ser apresentados em dispositivos com baixas resoluções ou áreas de visualização reduzidas e, ao mesmo tempo, permitir um número significativo de componentes. Os principais dispositivos para os quais o sistema foi pensado são os computadores, onde os dispositivos de visualização possuem maioritariamente resoluções onde a largura é maior do que a altura pelo que os componentes dos *dashboards* foram dispostos horizontalmente, mais concretamente em duas filas de três componentes (Figura 3.1), de modo a otimizar o espaço disponível.

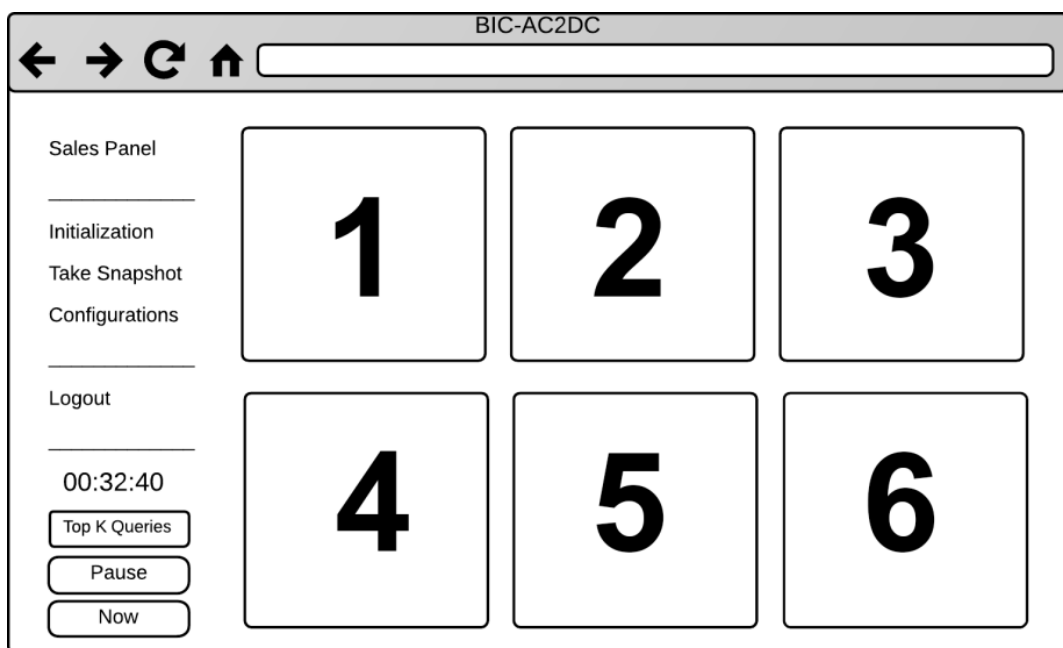


Figura 3.1 – Organização de um *dashboard* do sistema

Cada um destes componentes apresenta informação relativa a um conjunto de dados obtidos através de uma interrogação MDX utilizando uma das representações gráficas disponíveis. Foram implementadas sete representações gráficas comumente utilizadas em *dashboards*, sendo que cada uma delas possui características e casos de utilização particulares. A saber:

- Gráfico de Sectores – utilizado para realçar diferentes proporções entre um conjunto, preferencialmente pequeno, de valores.
- Gráfico de Barras – características similares ao gráfico de sectores, mas em que as proporções são percebidas mais facilmente graças ao seu *design*.

- Gráfico de Colunas – características idênticas ao gráfico de barras, sendo a orientação vertical das barras o único fator diferenciador.
- Gráfico de Linhas – utilizado para representar séries de dados. São ideais para visualizar tendências nos dados, nomeadamente tendências relacionadas com o tempo.
- Gráfico de Áreas em Escada – utilizado para comparar valores que sejam naturalmente representados por áreas.
- Gráfico Geográfico – utilizado para representar valores associando-os a áreas geográficas. Ideal para visualizar distribuições geográficas.
- Tabela de Dados – utilizado para representar valores de uma forma mais textual, de modo a permitir avaliar valores individuais caso seja necessário.

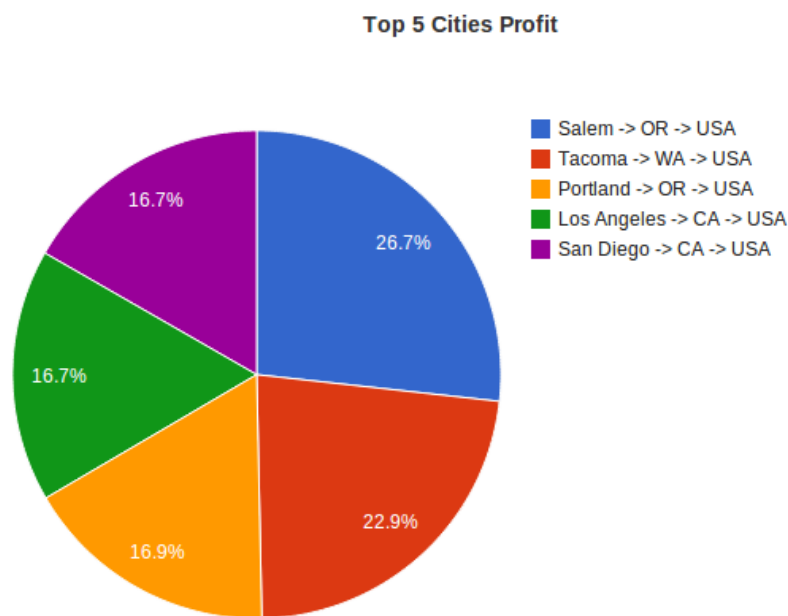


Figura 3.2 - Exemplo de um componente utilizando um gráfico de sectores

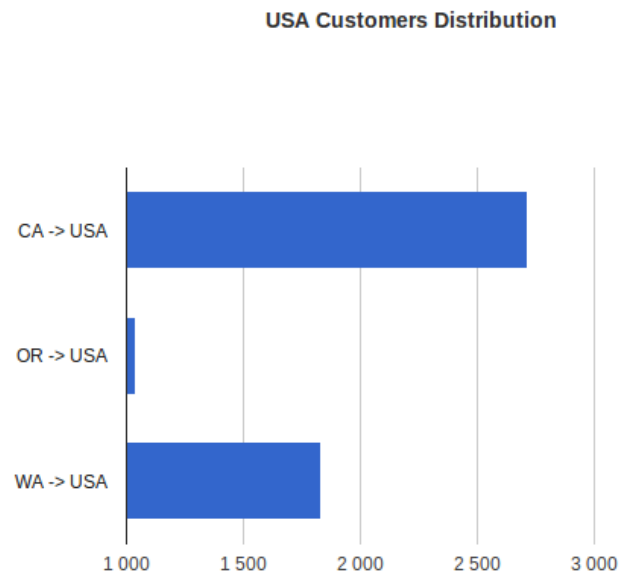


Figura 3.3 - Exemplo de um componente utilizando um gráfico de barras

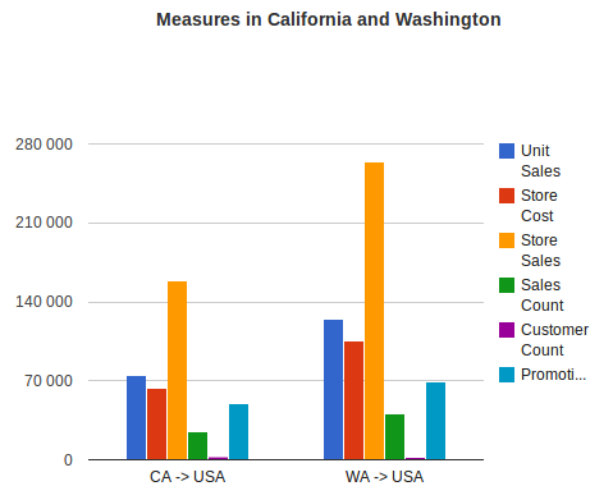


Figura 3.4 - Exemplo de um componente utilizando um gráfico de colunas

Com a diversidade de representações gráficas disponibilizadas pretende-se que cada utilizador tenha à sua disposição aquelas com que se sente mais à vontade e que melhor servem as suas necessidades, evitando assim que tenha que recorrer a representações gráficas não desejadas. Pretende-se ainda que um utilizador menos familiarizado com as diversas representações gráficas existentes seja capaz de experimentar e concluir quais as que melhor servem as suas necessidades

e preferências. Deste modo permite-se que os *dashboards* melhor se ajustem a uma maior diversidade de utilizadores e de casos de utilização.

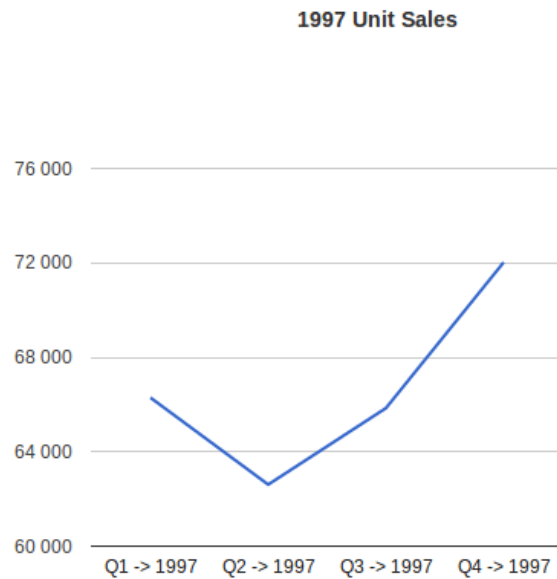


Figura 3.5 - Exemplo de um componente utilizando um gráfico de linhas

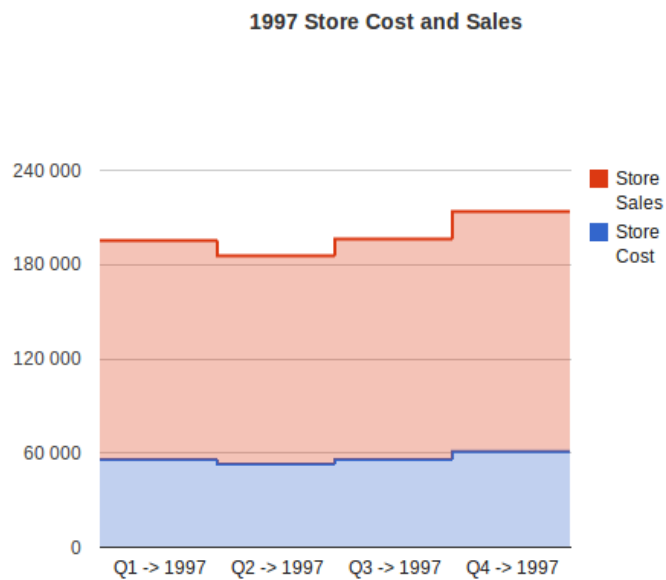


Figura 3.6 - Exemplo de um componente utilizando um gráfico de áreas em escada



Figura 3.7 - Exemplo de um componente utilizando um gráfico geográfico

1997 Store Cost and Sales

	Store Cost	Store Sales
Q1 -> 1997	55752.2405	139628.35
Q2 -> 1997	52964.2248	132666.27
Q3 -> 1997	55904.8694	140271.89
Q4 -> 1997	61005.8989	152671.62

Figura 3.8 - Exemplo de um componente utilizando uma tabela de dados

Em todas as representações gráficas, à exceção das tabelas onde tal funcionalidade não é necessária, quando o utilizador passa o ponteiro do rato por cima é apresentada uma caixa com informação relativa ao ponto, área, barra ou sector pairado. Esta funcionalidade permite que o utilizador tenha acesso aos valores exatos das medidas representadas, uma vez que numa representação visual é por vezes impossível determinar o valor com a precisão desejada.

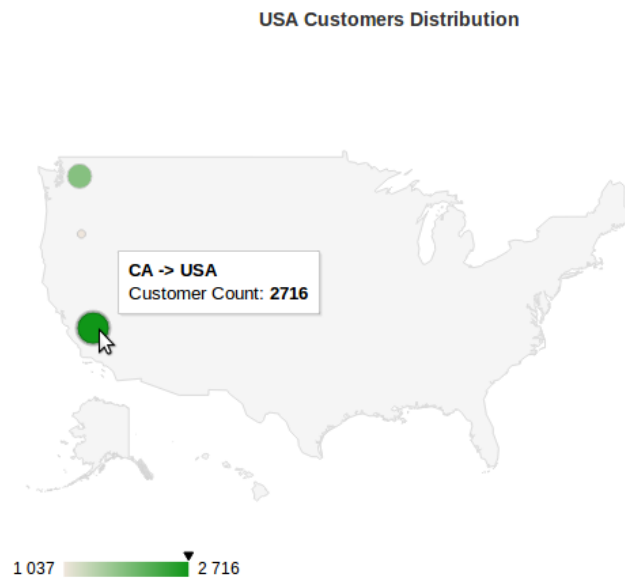
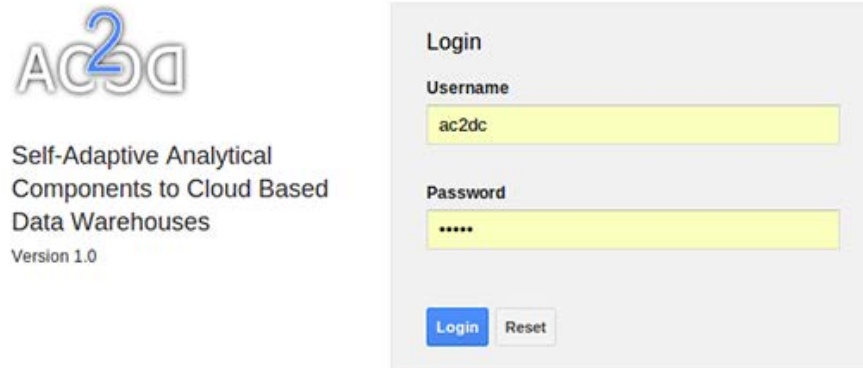


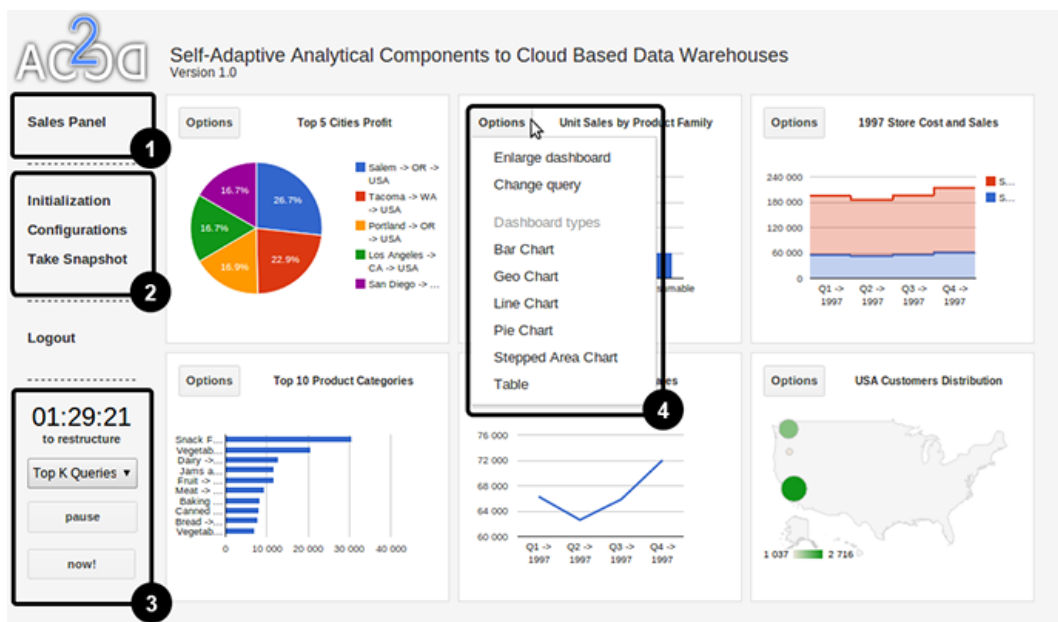
Figura 3.9 - Exemplo de caixa de informação de um componente

3.4 Funcionamento Geral

O acesso ao sistema é feito através de um *browser* convencional. O primeiro passo que um utilizador do sistema deve realizar antes de ter acesso aos seus painéis de exploração é autenticar-se. Para além de evitar acessos indevidos ao sistema, ao requerer que todos os utilizadores se autenticem antes de aceder ao sistema, este tipo de ação possibilita a criação de um conjunto diversificado de *dashboards* para cada utilizador, consoante as suas necessidades, e, acima de tudo, permite a recolha de informação necessária ao processo de reestruturação, disponibilizando as condições necessárias para que o sistema se adapte a cada um dos utilizadores individualmente. Na Figura 3.10 está apresentada a página inicial na qual o utilizador se deve autenticar utilizando as suas credenciais, credenciais estas que devem ser atribuídas por um administrador do sistema, obviamente.

Figura 3.10 - Página de *login*

Uma vez autenticado o utilizador é redirecionado para a página principal na qual tem acesso a todos os seus *dashboards*. Por omissão, nesta página é apresentado o último *dashboard* que o utilizador visualizou antes de fechar a aplicação, sendo que todas as alterações que não tenham sido guardadas serão perdidas, e ainda um menu lateral onde estão presentes múltiplas ações disponíveis para o utilizador. Na Figura 3.11 está um exemplo de uma possível página principal na qual foram assinaladas e numeradas as diversas opções que o utilizador tem disponíveis.

Figura 3.11 – A página principal do sistema de *dashboarding*

Na área assinalada com o número um está apresentada uma lista com todos os *dashboards* disponíveis. É onde o utilizador pode seleccionar o *dashboard* que pretende visualizar. No exemplo apresentado, em particular, o único *dashboard* disponível é o “Sales Panel”. Mas a lista pode-se estender consoante as necessidades do utilizador.

The screenshot shows the AC2D interface. On the left is a sidebar menu with the following items: Sales Panel, Initialization, Configurations, Take Snapshot, Logout, a timer showing 01:25:02 to restructure, Top K Queries, pause, and now!. On the right is a table with the following data:

	Date	Time	Type
Configuration8	2013-09-26	16:25	restructuring
Configuration7	2013-09-26	16:24	restructuring
Configuration6	2013-09-26	16:24	restructuring
Configuration5	2013-09-26	16:24	restructuring
Configuration4	2013-09-26	16:17	restructuring
Configuration3	2013-09-26	16:13	restructuring
Configuration2	2013-09-26	16:09	restructuring
Configuration1	2013-01-01	15:53	restructuring

Below the table are navigation arrows and page numbers 1 and 2. A blue 'Close' button is also visible.

Figura 3.12 – Uma lista de configurações de painéis de *dashboarding*

Em seguida, na área assinalada com o número dois, o utilizador tem ao seu dispor um conjunto de ferramentas que lhe permite gerir as diferentes configurações dos seus *dashboards*. A primeira opção, “Initialization”, faz com que o *dashboard* atual volte à configuração com que inicialmente foi definido, e caso existam alterações não guardadas estas serão perdidas. A opção que se segue, “Configurations”, dá acesso à lista de todas as configurações que foram registadas desde a primeira utilização do *dashboard* que se encontra apresentado. Nesta lista estão presentes todas as configurações que tiveram origem numa reestruturação, bem como todas as configurações que foram explicitamente guardadas pelo utilizador. A cada configuração está associada a data e a hora em que foi esta registada, bem como o seu tipo, isto é, se teve origem numa reestruturação automática ou num *snapshot* realizado pelo utilizador. O principal propósito desta lista é manter um histórico mais ou menos detalhado das alterações que os *dashboards* foram sofrendo ao longo do tempo, oferecendo assim a possibilidade ao utilizador não só de rejeitar uma reestruturação

que não tenha sido a mais correta, mas também de voltar a utilizar uma configuração anterior que ele ache que se adequa melhor à situação de análise de dados atual. Por fim, existe a opção "Take Snapshot" que permite ao utilizador guardar a configuração atual do *dashboard* apresentado, para mais tarde, se desejar, fazer a sua reposição e conseqüente utilização. A configuração guardada ficará disponível na lista de configurações anteriormente apresentada.

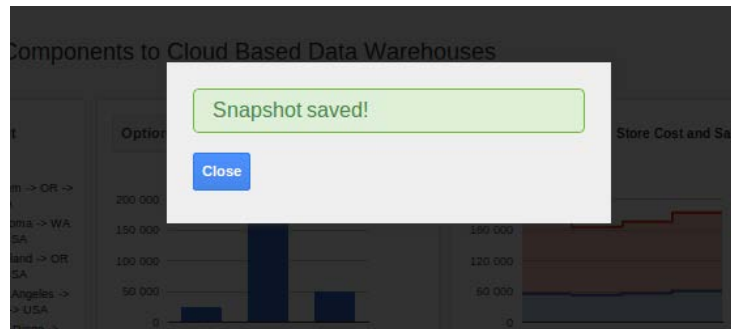


Figura 3.13 – Gravação de uma configuração a pedido do utilizador

Na área número três estão disponíveis as opções relativas à reestruturação dos *dashboards*. Os *dashboards* são reestruturados periodicamente, sendo que o contador apresentado na figura indica o tempo que falta até a próxima reestruturação de painéis. O utilizador pode definir esse tempo de reestruturação, bem como desligar as ações de reestruturação, colocando em modo de "pausa" o contador, e ainda solicitar que o *dashboard* atual seja reestruturado de imediato. Deste modo o utilizador tem controlo sobre o processo de reestruturação, não sendo forçado a usá-lo uma vez que pode optar por desligá-lo, caso não se sinta à vontade com o mesmo ou sinta que este não se encontra com a definição adequada para as suas necessidades de exploração de dados. Quando o contador atinge o valor zero, ou quando uma reestruturação é solicitada de imediato, o *dashboard* atual é automaticamente atualizado de modo a refletir o resultado dessa reestruturação, que será adicionada à lista de configurações disponíveis.

A última área assinalada, área número quatro, é onde se encontram localizadas todas as opções relacionadas com o componente do *dashboard* ao qual estão associadas. Este menu possui essencialmente três funcionalidades:

- "Change Query" – que permite ao utilizador alterar os dados que estão representados no componente de visualização, através da utilização de interrogações MDX, que reflitam um conjunto de dados válido. É ainda possível alterar o título dos dados de modo a melhorar a

representação do novo conjunto de dados, caso seja necessário. Caso seja submetida uma interrogação MDX válida, o componente é atualizado de imediato e a ação é registrada pelo sistema. Mas se esta ação for inválida nada acontecerá, sendo apresentados os mesmos dados no componente em questão.

- “Enlarge Dashboard” – que possibilita ao utilizador aumentar o componente desejado. Esta funcionalidade permite que o utilizador se foque num só componente caso necessário, e com o aumento do tamanho da representação gráfica dos dados, sucede-se um potencial aumento na facilidade com que a informação é absorvida. Esta funcionalidade não afeta o *dashboard* e é possível a qualquer momento voltar à ao modo de visualização normal, onde é apresentado todo o *dashboard*.
- “Chart Types” – que permite que o utilizador altere a representação gráfica utilizada no componente. No fundo do menu em questão estão enumeradas todas as representações gráficas alternativas e para utilizar qualquer uma delas o utilizador apenas terá que seleccionar a que pretende ativar. Ativando uma dessas opções, o componente será atualizado de imediato. Mas caso não seja possível utilizar a representação gráfica seleccionada para representar os dados presentes no componente, uma mensagem de erro será emitida. Tal como acontece quando a interrogação MDX ou o título de um componente é alterado, quando a representação gráfica é alterada, o sistema regista a alteração.

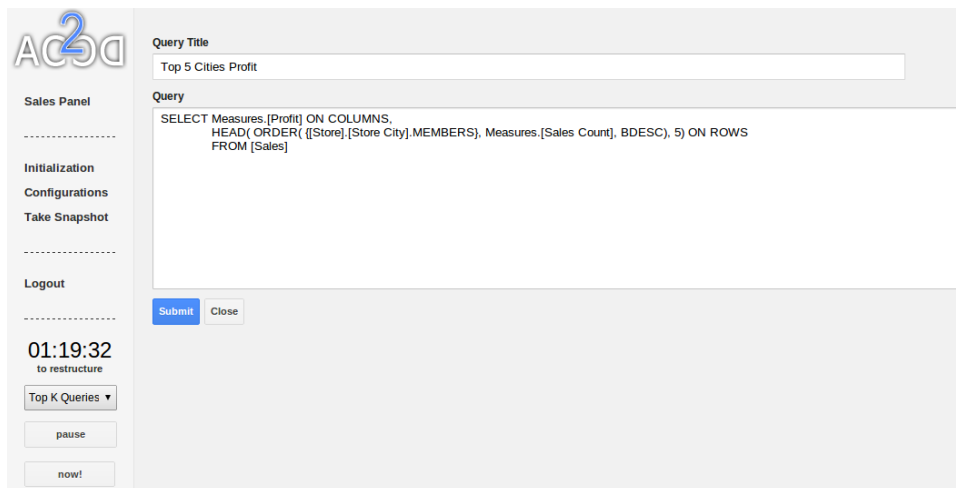


Figura 3.14 - Alteração da interrogação MDX

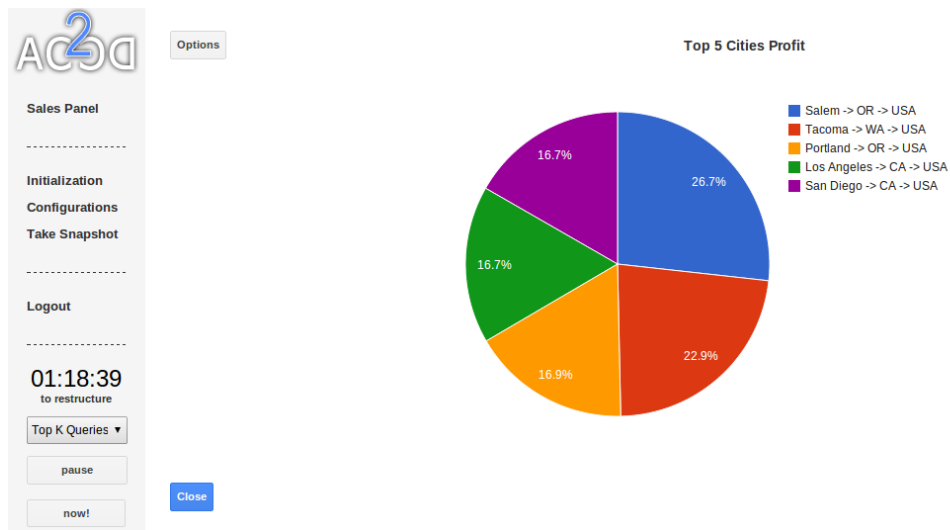


Figura 3.15 - Componente aumentado

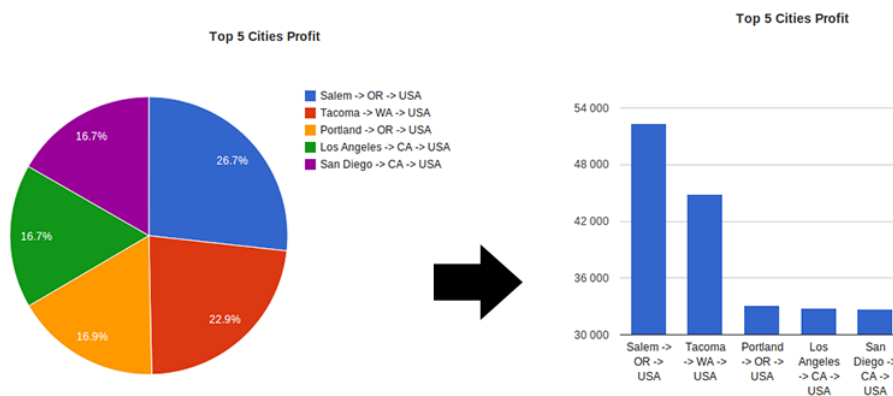


Figura 3.16 - Alteração da representação gráfica de um componente

3.5 Recolha dos Perfis de Utilização

Para criar um sistema auto-adaptável, mais concretamente um sistema que reflita as preferências dos utilizadores e que satisfaça as suas necessidades através da reestruturação automatizada dos *dashboards*, é necessário desenvolver mecanismos que recolham informação relativa às sessões de utilização do sistema por parte dos utilizadores. Desta forma o sistema fica dotado de uma base de conhecimento que lhe permite, de uma forma mais ou menos correta, auxiliar os utilizadores na

análise e exploração dos *dashboards*. Para criar tais mecanismos foi realizada uma análise da informação que potencialmente poderia ser extraída sem a interferência direta do utilizador, ou seja, informação diretamente relacionada com as ações que o utilizador realiza durante um processo de exploração de dados. Deste modo é possível recolher a informação necessária sem interferir com o processo de exploração, processo esse que requiere a total concentração e disponibilidade do utilizador. Com base nessa análise foi compilada uma lista com a informação passível de ser obtida e usada no processo de auto-adaptação do sistema. De referir:

- Data e hora da ação realizada.
- Informação relativa aos componentes do *dashboard* (representação gráfica utilizada, posicionamento absoluto e relativo aos outros componentes do *dashboard*, etc.).
- Informação relativa aos dados (interrogação que dá origem ao conjunto de dados, título atribuído a esse conjunto de dados, etc.)

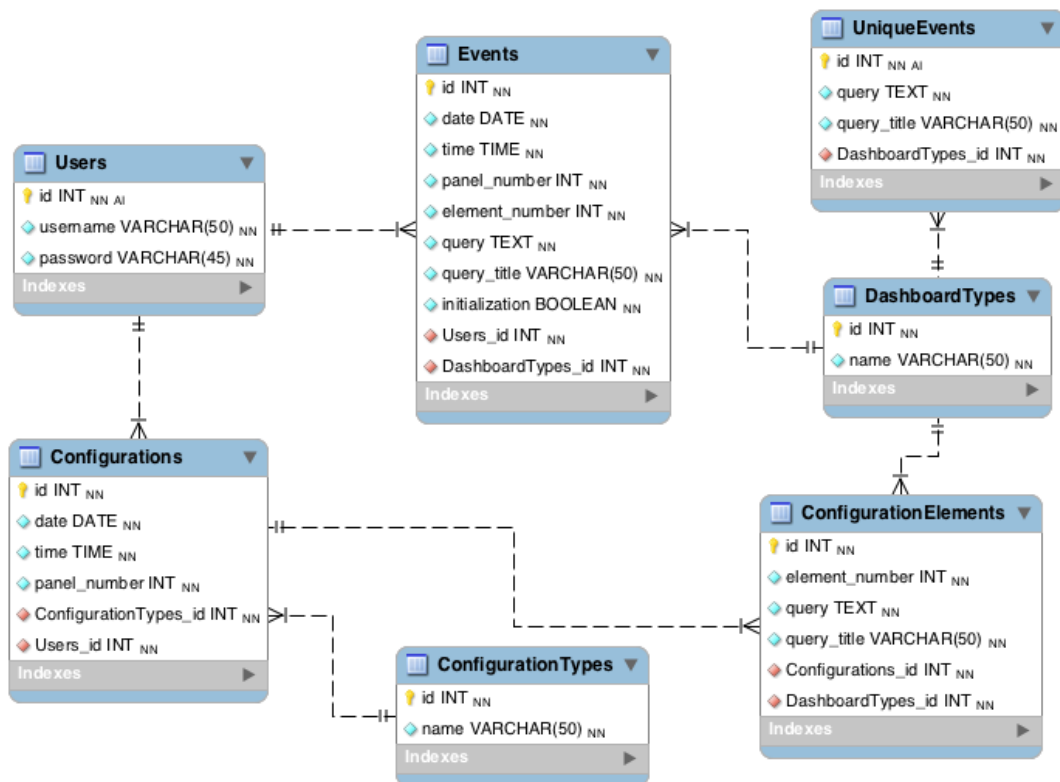


Figura 3.17 - Esquema lógico da base de dados que contém a informação relativa à utilização do sistema por parte dos utilizadores e as configurações dos diferentes *dashboards*.

A partir desta lista foi então pensado e desenvolvido um modelo de dados (Figura 3.17) capaz de albergar toda esta informação. Este modelo foca-se essencialmente nas ações que o utilizador vai realizando ao longo do processo de exploração dos *dashboards*, bem como nas configurações que os *dashboards* vão tomando ao longo da sua existência.

As relações presentes no modelo apresentado podem ser divididas em três grupos distintos, consoante o tipo de informação que albergam. O primeiro grupo é relativo aos utilizadores e contém apenas a relação "Users", na qual é mantida toda a informação relativa às autenticações dos utilizadores. Esta relação é fundamental não só para filtrar o acesso dos utilizadores ao sistema mas também para associar todo o resto da informação obtida ao utilizador que lhe deu origem. Em seguida temos o grupo relativo às ações realizadas pelos utilizadores, às quais chamamos de eventos, e que é composto pelas relações "Events" e "UniqueEvents". Na relação "Events" é guardada toda a informação relativa a cada uma das ações realizadas pelos utilizadores, nomeadamente a data em que esta foi realizada, qual o estado final do componente X do *dashboard* Y depois de realizada mesma, e ainda se esta ação foi despoletada explicitamente pelo utilizador ou através da simples inicialização do *dashboard*. Por outras palavras, cada registo desta relação representa o estado de um elemento de um *dashboard* após a sua inicialização ou modificação. Já a relação "UniqueEvents" é uma tabela auxiliar que serve de suporte ao processo de reestruturação dos *dashboards*, sendo utilizada para atribuir um identificador único a cada evento distinto. Os atributos selecionados para diferenciar os eventos foram a interrogação que dá origem aos dados representados no componente do *dashboard*, o tipo de representação gráfica utilizado para representar esses dados e ainda o título atribuído a esse conjunto de dados. É importante realçar a opção tomada para guardar a informação relativa às ações, ou eventos, desnormalizada. Seria possível criar um relacionamento entre as relações "Events" e "UniqueEvents", eliminando assim a necessidade de ter os campos que se encontram nas duas relações na relação "Events". Porém, tal modelo dos dados limitaria o sistema de reestruturação. Desta forma estaríamos a limitar a diferenciação dos eventos aos três campos selecionados não sendo possível introduzir outras variâncias da tabela "UniqueEvents" caso assim fosse desejado. Assim temos toda a informação necessária na relação "Events", sendo possível a partir daí decidir como devem ser distinguidos os diferentes eventos, consoante as necessidades e preferências dos utilizadores ou os algoritmos de reestruturação utilizados, sendo que no caso particular em estudo será apenas utilizada a relação "UniqueEvents" para o efeito.

	id int	user_id integer	date date	time time without	panel integ	element integer	query text	initialization boolean	query_title character varying(100)	graphic type character varying(20)
1	1	1	2013-01-10	00:00:02	1	1	SELECT Me		Top 5 Cities Pr	PieChart
2	2	1	2013-01-10	00:00:02	1	2	select {f		Unit Sales by P	ColumnChart
3	3	1	2013-01-10	00:00:02	1	3	select {f		1997 Store Cost	SteppedAreaChart
4	4	1	2013-01-10	00:00:02	1	4	select {f		Top 10 Product	BarChart
5	5	1	2013-01-10	00:00:02	1	5	select {f		1997 Unit Sales	LineChart
6	6	1	2013-01-10	00:00:02	1	6	select {f		USA Customers D	GeoChart
7	7	1	2013-01-10	00:00:03	1	4	select {f		USA Unit Sales	GeoChart
8	8	1	2013-01-10	00:00:04	1	1	SELECT Me		Measures in Cal	ColumnChart
9	9	1	2013-01-10	00:00:05	1	3	SELECT Me		Top 10 Stores C	GeoChart
10	10	1	2013-01-10	00:00:06	1	6	with set f		Recent Trends f	BarChart
11	11	1	2013-01-10	00:00:07	1	2	SELECT { f		Drinks by State	LineChart

Figura 3.18 - Exemplo de registos que representam ações do utilizador.

Por fim temos o grupo formado pelas relações relativas às configurações dos *dashboards* e dos seus respetivos componentes, nomeadamente as relações “Configurations”, “ConfigurationTypes”, “ConfigurationElements” e “DashboardTypes”. Na relação “Configurations” são registadas todas as configurações de cada um dos *dashboards* existentes. É registada a data e a hora da criação da configuração, o tipo de configuração, e ainda o utilizador que deu origem a essa mesma configuração. Os tipos disponíveis para uma configuração estão registados na relação “ConfigurationTypes” e são essencialmente dois, nomeadamente o tipo reestruturação e o tipo *snapshot*. Quando uma configuração é gerada automaticamente através do processo de reestruturação o seu tipo é reestruturação, mas se uma configuração tem origem na intervenção direta do utilizador, sendo que é o próprio que requisita que a configuração seja guardada, esta é uma configuração do tipo *snapshot*. A relação “ConfigurationElements” complementa a relação “Configurations” uma vez que contém a informação relativa à configuração de cada uma dos elementos de um *dashboard*. Cada registo desta relação contém informação relativa aos dados representados no componente, à representação gráfica desses dados e ainda um relacionamento que o permite associar a uma configuração em específico. Por fim temos a relação “DashboardTypes” onde estão registados os diversos tipos de representações gráficas disponíveis, como por exemplo representação em tabela ou em gráfico de barras.

	date date	time time without	configuration type character varying(20)	element integer	query text	graphic type character varying(20)	query_title character varying(100)
1	2013-09-26	16:09:29	restructuring	1	select {[Measure	SteppedAreaChart	1997 Store Cost
2	2013-09-26	16:09:29	restructuring	2	select {[Measure	ColumnChart	Unit Sales by Pr
3	2013-09-26	16:09:29	restructuring	3	SELECT Measures.	PieChart	Top 5 Cities Pro
4	2013-09-26	16:09:29	restructuring	4	SELECT Measures.	GeoChart	Top 10 Stores Cu
5	2013-09-26	16:09:29	restructuring	5	SELECT { Except(LineChart	Drinks by State
6	2013-09-26	16:09:29	restructuring	6	SELECT Measures.	ColumnChart	Measures in Cali

Figura 3.19 - Exemplo de uma configuração de um *dashboard* associada a uma reestruturação.

Em suma, a informação guardada na base de dados, que foi criada de acordo com o modelo que acabou de ser apresentado, servirá como base do serviço de reestruturação e ainda para guardar todas as configurações de *dashboards* quer estas sejam criadas pelo próprio sistema durante uma reestruturação, quer sejam criadas manualmente por um utilizador.

3.6 Fonte De Dados

A fonte de dados utilizada para alimentar o servidor OLAP foi uma base de dados bastante conhecida e que acompanha as distribuições do próprio servidor OLAP utilizado, o Mondrian, e que é denominada de Foodmart.

Relação	Descrição
Customer	Dimensão que contém toda a informação relacionada com os clientes como nome, contactos, moradas, data de nascimento, estado civil, rendimento anual, grau de educação, ocupação, número de conta, etc..
Product	Dimensão que contém informação relacionada com os produtos vendidos como marca, nome, pesos, se é ou não reciclável, unidades por embalagem, embalagens por palete, tamanho ocupado na prateleira, entre outros.
Product_class	Dimensão que contém informação relacionada com a classe dos produtos vendidos mais concretamente a sua categoria, subcategoria, família, e o departamento a que pertencem.
Promotion	Dimensão que contém informação relativa às promoções efetuadas durante o processo de venda dos produtos, como o distrito onde a promoção teve lugar, o seu nome, os meios de comunicação utilizados para a promover, o seu custo, e as datas de início e fim da mesma.
Sales_fact_1997	Tabela de factos que representa as vendas realizadas no ano de 1997. Cada registo desta tabela representa uma venda de um produto, e as métricas associadas a cada venda são o número de unidades vendidas, a despesa com o produto vendido e o valor recebido pela venda.
Store	Dimensão que contém informação relacionada com as lojas, como por exemplo o seu tipo, número, morada, gerente, contactos, datas de abertura e de ultima remodelação, entre outras.
Time_by_day	Dimensão que contém informação relativa à data das vendas, como hora, dia, dia da semana, dia do mês, dia do ano, mês, ano, semana do ano, trimestre, e período fiscal.

Tabela 3.2 - Descrição das relações usadas para alimentar o cubo "Sales"

A seleção da Foodmart deveu-se ao facto de esta base de dados ser frequentemente utilizada como caso de estudo, possuir uma quantidade de dados significativa e de se focar numa área de negócio comumente relacionada com a exploração de dados, nomeadamente a área de retalho. Para além desta fonte de dados estão disponíveis vários esquemas de cubos passíveis de serem criados sobre essa mesma fonte. O cubo utilizado para montar o sistema foi o cubo relativo às vendas, ou seja, o cubo "Sales", que é criado a partir de sete das tabelas do Foodmart.

O cubo "Sales" é composto por doze dimensões e nove métricas. Na Tabela 3.3 estão descritas as diferentes dimensões que o compõe e na Tabela 3.4 estão descritas as suas métricas.

Dimensão	Descrição	Tabelas relacionadas
Store	Dimensão que contém informação relacionada com as lojas.	Store
Store Size in SQFT	Dimensão que representa o tamanho das lojas em metros quadrados.	Store
Store Type	Dimensão que representa o tipo das lojas.	Store
Time	Dimensão que contém toda a informação relacionada com a data em que as vendas são efetuadas.	Time_by_day
Product	Dimensão que contém informação relativa aos produtos comercializados.	Product, Product_class
Promotion Media	Dimensão que contém informação relativa aos meios de comunicação utilizados para promover as promoções.	Promotion
Promotions	Dimensão que contém informação relativa às promoções.	Promotion
Customers	Dimensão que contém a informação pessoal dos clientes.	Customer
Education Level	Dimensão que representa o grau de educação dos clientes.	Customer
Gender	Dimensão que representa o género dos clientes.	Customer
Marital Status	Dimensão que representa o estado civil dos clientes.	Customer
Yearly Income	Dimensão que representa os ganhos anuais dos clientes.	Customer

Tabela 3.3 - Descrição das dimensões do cubo "Sales"

Métrica	Descrição
Unit Sales	Unidades vendidas numa venda.
Store Cost	Custos relacionados com o produto vendido.
Store Sales	Ganhos relacionados com uma venda.
Sales Count	Contagem das vendas.
Customer Count	Contagem dos clientes.
Promotion Sales	Valor gasto com as promoções.
Profit	Lucro relacionado com uma venda.
Profit last Period	Lucro relacionado com uma venda registada no último período.
Profit Growth	Percentagem de crescimento dos lucros.

Tabela 3.4 - Descrição das métricas do cubo "Sales"

3.7 Apresentação do Caso de Estudo

Para ser possível utilizar os mecanismos de reestruturação implementados com correção é necessário que existam registos de processos de exploração efetuados pelos utilizadores na base de dados associada a estes mecanismos. Para tal foram gerados registos aleatoriamente de modo a simular vários processos de exploração realizados por um utilizador, sobre um *dashboard*. Durante o processo de geração dos dados foram utilizadas onze configurações diferentes para os componentes do *dashboard*. De modo a simplificar a compreensão do caso de estudo utilizado nesta dissertação decidiu-se utilizar um título diferente para cada configuração. Na Tabela 3.5 estão apresentadas as onze configurações utilizadas. As interrogações utilizadas para gerar os dados relativos a cada configuração foram ocultadas de modo a simplificar a sua percepção.

As primeiras seis configurações foram utilizadas para compor a configuração inicial do *dashboard* tendo sido as restantes utilizadas para simular alterações realizadas por parte do utilizador. Estas alterações são geradas aleatoriamente tendo em conta alguns factos pré-concebidos criados com o objetivo de introduzir padrões de utilização.

	Título	Representação gráfica
1	Top 5 Cities Profit	Gráfico de sectores
2	Unit Sales by Product Family	Gráfico de colunas
3	1997 Store Cost and Sales	Gráfico de áreas em escada
4	Top 10 Product Categories	Gráfico de barras
5	1997 Unit Sales	Gráfico de linhas
6	USA Customers Distribution	Mapa geográfico
7	USA Unit Sales	Gráfico de colunas
8	Measures in California and Washington	Mapa geográfico
9	Top 10 Stores Customers Count	Gráfico de linhas
10	Recent Trends for Best-Selling Brands	Mapa geográfico
11	Drinks by State in 1997	Gráfico de barras

Tabela 3.5 - Configurações utilizadas na geração de perfis de utilização

Na Tabela 3.6 estão apresentados os padrões desejados, sendo que as percentagens apresentadas são as que foram utilizadas pela aplicação que gera os dados como guia podendo não representar de forma exata o resultado final. Todavia, é certo que o resultado final apresenta percentagens aproximadas às apresentadas.

Configuração inicial	Configuração final	Percentagem
Top 5 Cities Profit	USA Unit Sales	70%
Unit Sales by Product Family	Top 10 Stores Customers Count	75%
1997 Store Cost and Sales	Measures in California and Washington	80%
1997 Unit Sales	Recent Trends for Best-Selling Brands	50%
USA Customers Distribution	Drinks by State in 1997	55%

Tabela 3.6 - Padrões de alterações introduzidos nos dados gerados

Tendo em conta as onze configurações e os cinco padrões apresentados foram gerados dados relativos a 300 processos de exploração realizados ao longo do tempo pelo utilizador, tendo sido gerados 1948 registos representativos de alterações feitas pelo utilizador sobre o *dashboard*. Após a geração dos dados obtemos um perfil de utilização de um utilizador sobre um dado *dashboard*, que permite, assim, realizar os processos de reestruturação necessários ao estudo dos mesmos, bem como do sistema em geral.

3.8 Serviço de Reestruturação

3.8.1 Top-K Queries

Apesar da sua simplicidade, o algoritmo *Top-K Queries* tem potencial para originar resultados bastante interessantes quando utilizado no processo de reestruturação de um *dashboard*. O algoritmo *Top-K Queries* implementado foi utilizado para configurar o *dashboard* atual com as seis configurações de componentes mais frequentemente especificadas pelo utilizador no decorrer da exploração do *dashboard*. Para realizar tal operação, o algoritmo percorre todos os registos de ações que compõe o perfil de exploração do utilizador sobre o *dashboard* atual contando a frequência de cada configuração de componente. Ele termina o processo selecionando as seis configurações mais utilizadas. É de notar que numa primeira fase as configurações iniciais são desconsideradas, uma vez que a sua potencial elevada frequência poderia levar a que o algoritmo produzisse um resultado igual à configuração atual, resultando na ineficácia do mesmo. Como tal, apenas são contabilizadas as configurações que o utilizador alterou deliberadamente, sendo que no caso em que estas não sejam suficientes, se utilizam então as configurações iniciais. Na Tabela 3.7 está apresentada a contagem realizada pelo algoritmo utilizando o caso de estudo.

	Título da Interrogação	Ocorrências
1	Drinks by State in 1997	321
2	Top 10 Stores Customers Count	305
3	Measures in California and Washington	271
4	Recent Trends for Best-Selling Brands	229
5	USA Unit Sales	201
6	Unit Sales By Product Family	161
7	Top 10 Product Categories	145
8	Top 5 Cities Profit	130
9	1997 Store Cost and Sales	127
10	USA Customers Distribution	45
11	1997 Unit Sales	13

Tabela 3.7 - Contagem das ocorrências das ações de um utilizador

As seis primeiras configurações são então selecionadas e é alterada a configuração do *dashboard* em apresentação de modo a refletir o processo de reestruturação descrito. Na Figura 3.20 é

possível visualizar o *dashboard* com a configuração que resultou deste processo e no qual podemos ver representadas as seis configurações mais frequentes.

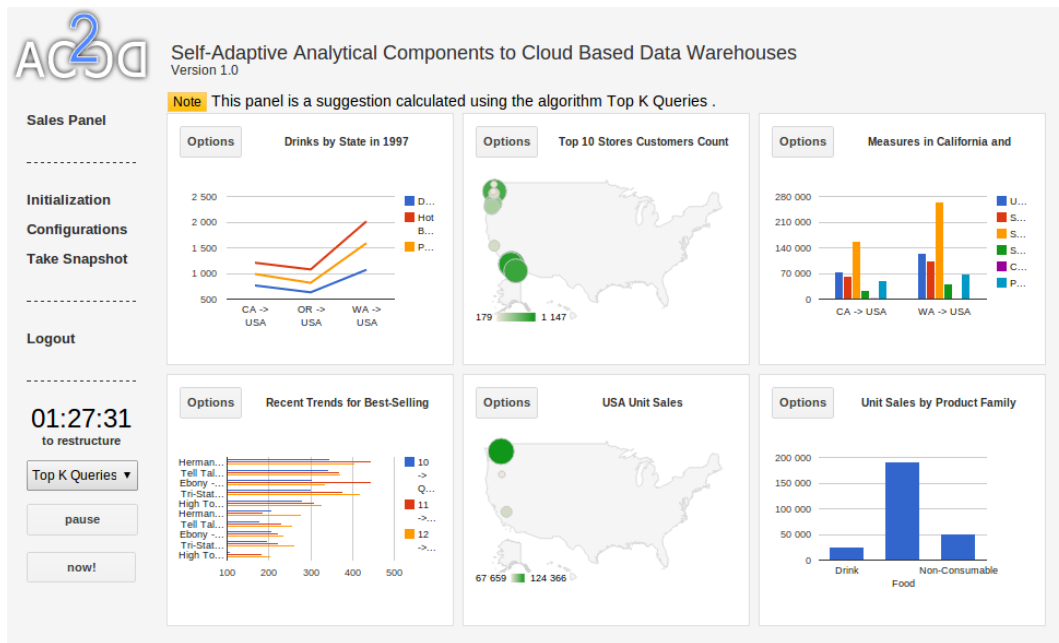


Figura 3.20 - *Dashboard* resultante do processo de reestruturação utilizando o algoritmo *Top-K Queries*

3.8.2 Cadeias de Markov

O segundo mecanismo de reestruturação implementado baseou-se em cadeias de Markov. Através da utilização destas cadeias é possível determinar uma nova configuração para um *dashboard* a partir do estudo das várias configurações que foram sendo utilizadas pelo utilizador, incluindo configurações resultantes de alterações feitas pelo mesmo e que não foram registadas. Estas configurações são extraídas dos dados armazenados na base de dados associada ao processo de reestruturação, mais concretamente dos registos das ações realizadas pelo utilizador que representam alterações dos componentes do *dashboard*. A partir desses dados é então possível extrair informação relacionada com a coexistência dos vários componentes nas diferentes configurações que o *dashboard* foi apresentando ao longo do tempo, possibilitando assim a construção de uma cadeia de Markov que representasse precisamente esses relacionamentos entre os componentes. Ao contrário do método que utiliza o algoritmo de Top-K Queries, neste método todas as configurações são consideradas no processo, mesmo até as configurações iniciais, pelo

que a utilização constante da mesma configuração inicial de um *dashboard* leva ao fortalecimento do relacionamento entre as configurações de componentes utilizadas. A utilização das configurações iniciais pode levar à construção de grafos menos desejáveis caso o utilizador se aperceba que o *dashboard* não é o mais indicado para ele e não utilize a opção de guardar uma configuração, optando assim por alterar o *dashboard* sempre que o abrir, levando à criação de relacionamentos entre configurações que na realidade não estão a ser utilizadas em conjunto, uma vez que são alteradas de imediato. No entanto este tipo de situações não é esperado, uma vez que o utilizador caso veja que a configuração daquele *dashboard* não é a mais indicada a altere e guarde as alterações, definindo assim uma nova configuração inicial para o *dashboard*. Deste modo a configuração inicial irá conter apenas relacionamentos válidos e desejados, originando melhores resultados na utilização das cadeias.

O primeiro passo para que seja possível aplicar este mecanismo é a criação de um grafo que represente os relacionamentos entre as várias configurações de componentes e que permita por fim extrair um conjunto de seis configurações de modo a criar um *dashboard* útil para o utilizador. Um nodo do grafo representa uma configuração de um componente, configuração esta que pode ter pertencido a um qualquer componente do *dashboard* em causa, e uma aresta entre dois nodos, que representa o grau de coexistência das configurações presentes nesses dois nodos, isto é, a frequência com que essas duas configurações foram visualizadas em simultâneo. Para representar as configurações nos nodos serão utilizados os identificadores atribuídos na tabela "UniqueEvents", criada para o efeito, em que cada identificador representa um conjunto de dados composto por um título, uma interrogação e um método de representação gráfica, tal como foi explicado anteriormente. As arestas do grafo são direcionadas e pesadas, e, numa primeira fase, o seu peso representa o número de vezes que as configurações contidas nos nodos inicial e final foram utilizadas em simultâneo no *dashboard*. Numa fase posterior esses pesos são convertidos em probabilidades, passando a representar a probabilidade da configuração contida no nodo inicial ser apresentada em simultâneo com a configuração presente no nodo final. Essas probabilidades são calculadas de acordo com as frequências que inicialmente estavam presentes nas arestas, dividindo cada uma delas pelo total da soma de todas as arestas que saem do nodo inicial associado a cada aresta. O grafo associado ao exemplo utilizado é visualmente muito pesado graças ao enorme número de arestas presentes pelo que será apresentado na Figura 3.21 apenas uma porção do mesmo, onde podem ser observados os relacionamentos entre algumas das configurações.

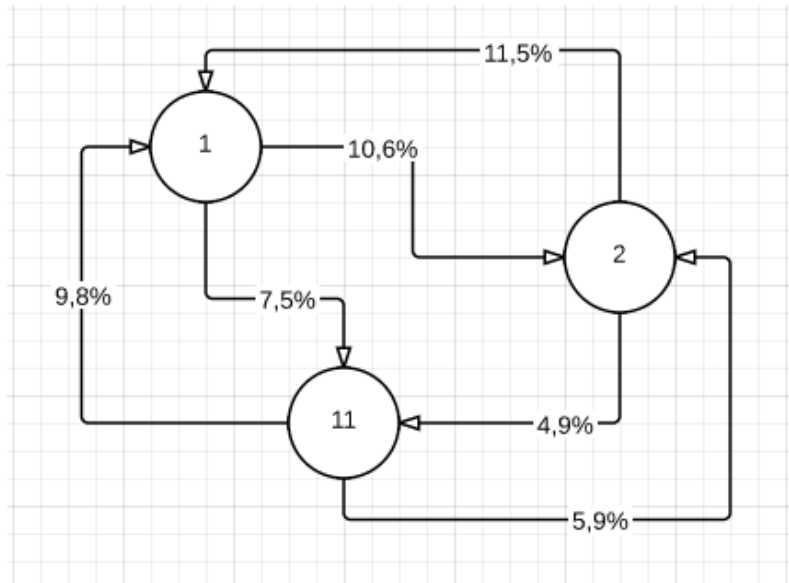


Figura 3.21 - Porção do grafo gerado pelo algoritmo de cadeias de Markov

É importante notar que como uma configuração de componente é sempre utilizada juntamente com outras cinco configurações, uma vez que cada *dashboard* tem sempre seis componentes, a probabilidade máxima que uma aresta pode conter é de 20%. Assim sendo, se numa aresta estiver uma probabilidade de 10%, esta representa na realidade uma ocorrência simultânea das configurações presentes nos nodos inicial e final em 50% das configurações do *dashboard* em questão.

Estando criado o grafo, é então altura de o utilizar para gerar uma cadeia de seis configurações para reestruturar o *dashboard*. Para gerar a cadeia é necessário indicar o nodo de partida, e como não existe um nodo que seja mais indicado para tal do que os outros, optou-se por seleccionar um nodo aleatoriamente. Tomou-se contudo a decisão de não deixar esta escolha completamente ao acaso, associando uma probabilidade de seleção a cada nodo. Esta probabilidade é calculada de acordo com o número de ocorrências de cada nodo, mais concretamente da configuração presente no mesmo, levando a que configurações com mais utilizações no *dashboard* sejam utilizadas com mais frequência como ponto de partida para o algoritmo. A partir deste nodo é então criada uma sequência composta por seis nodos, em que cada nodo é escolhido aleatoriamente de entre o conjunto de nodos relacionados com o último nodo seleccionado, e tal como no caso da seleção do nodo de partida, as probabilidades são tidas em consideração. É importante que o processo não seja completamente aleatório, daí a utilização das probabilidades presentes nas arestas. Todavia,

este pequeno grau de aleatoriedade introduzido é importante, pois permite, não só apresentar o caminho à partida mais óbvio, mas também caminhos tão ou mais úteis para o utilizador e que sejam menos evidentes. Tomemos o caso de ser necessário escolher entre um “nodo X”, que tenha uma frequência de 100, e um “”, que tenha frequência de 99. Sem a aleatoriedade o “nodo X” seria sempre selecionado. Mas com a introdução da aleatoriedade pesada é possível obter tanto o “nodo X” como o “nodo Y” na sequência sensivelmente o mesmo número de vezes não descartando assim uma opção potencialmente viável. Com o passar do tempo, se o “nodo Y” não for assim tão relevante como à partida parecia ser, a frequência do mesmo em relação ao “nodo X” irá diminuindo, fazendo com que o “nodo X” se torne cada vez mais frequente na sequência, adaptando assim a sequência à medida do utilizador.

Graças ao grau de aleatoriedade introduzido não é possível apresentar o *dashboard* resultante da aplicação do algoritmo ao caso de estudo mas apenas um possível resultado. Na Figura 3.23 está representado então um possível *dashboard* criado a partir da cadeia de Markov resultante da aplicação deste algoritmo, cadeia essa que está apresentada na Figura 3.22.

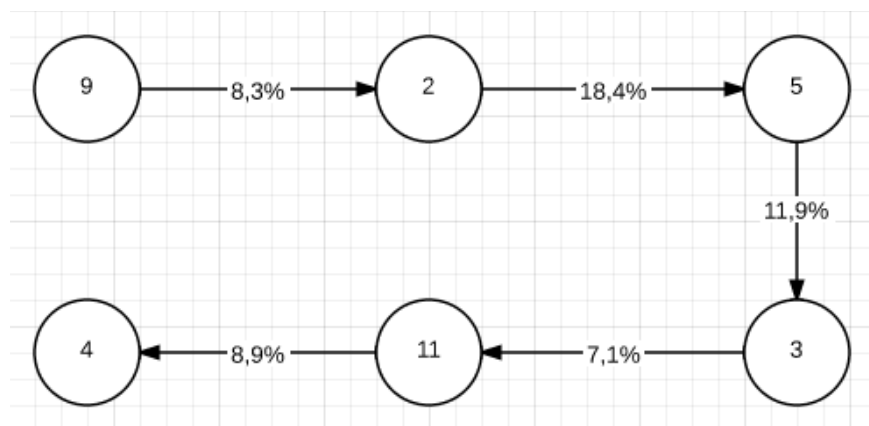


Figura 3.22 - Cadeia de Markov

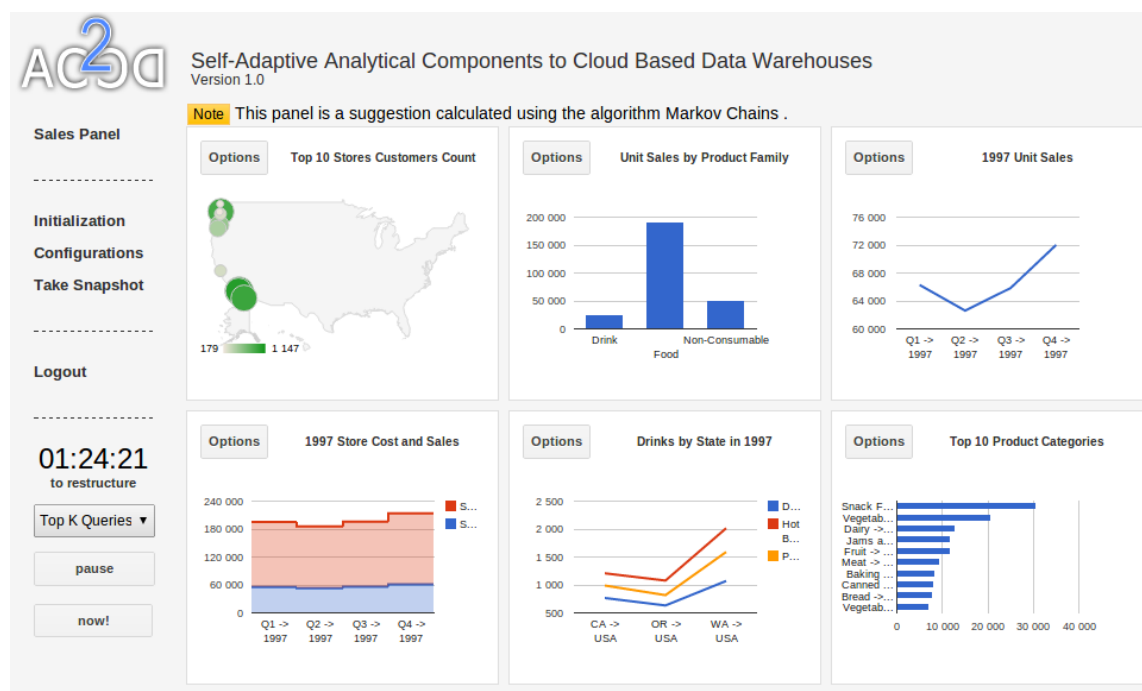


Figura 3.23 - Exemplo de um *dashboard* resultante do processo de reestruturação utilizando o algoritmo de cadeias de Markov

3.8.3 Regras de Associação

O último mecanismo de reestruturação implementado neste sistema baseou-se em regras de associação. Para tal recorreu-se às bibliotecas disponibilizadas pelo WEKA¹⁹, nas quais existe uma implementação de um algoritmo de regras de associação, mais concretamente, uma implementação do algoritmo Apriori²⁰. Apesar de o objetivo que se pretendeu alcançar com a utilização deste método ser o mesmo dos outros métodos já aqui apresentados, nomeadamente reestruturar os *dashboards* de modo a melhor satisfazer as necessidades dos seus utilizadores, a abordagem que foi usada é relativamente diferente. No primeiro método apresentado, Top-K Queries, não foi tido em consideração nenhum tipo de relacionamento entre as configurações dos componentes, apenas foi tida em conta a sua frequência. Nas cadeias de Markov a coexistência das configurações num mesmo *dashboard* foi a chave para a criação das mesmas. Já no presente mecanismo em que são utilizadas as regras de associação pretende-se observar o relacionamento entre duas configurações que são utilizadas em sequência, ou seja, situações em que a

¹⁹ WEKA - <http://www.cs.waikato.ac.nz/ml/weka/>

²⁰ WEKA Apriori - <http://weka.sourceforge.net/doc.dev/weka/associations/Apriori.html>

“configuração A” é utilizada dando depois lugar à “configuração B”. Tal como acontecia no mecanismo baseado em cadeias de Markov, este método está sujeito a erros causados pela má utilização do sistema por parte do utilizador. Quando o utilizador ignora a possibilidade de guardar uma configuração que ache mais adequada e que pretende visualizar sempre que abre um *dashboard*, levando a que este altere o *dashboard* sempre que o abre, faz com que sejam introduzidos relacionamentos entre configurações que não são necessariamente pretendidos. Contudo, se assumirmos que os utilizadores do sistema estão minimamente aptos à sua utilização, este tipo de erros serão evitados, não influenciando, a longo prazo, o correto funcionamento do algoritmo. À partida os relacionamentos que são extraídos e utilizados no algoritmo podem parecer pouco relevantes uma vez que se o utilizador alterou a configuração de um componente seria porque não pretende voltar a visualizar a antiga configuração mas isso não é necessariamente verdade. Tomemos agora como exemplo uma das ações que mais frequentemente é realizada durante a exploração de um *dashboard*, isto é, a alteração do grau de detalhe de um conjunto de dados, quer seja através de operações de agregação ou de desagregação, quer seja através da utilização de um filtro para reduzir a abrangência dos dados. Se, por exemplo, um utilizador altera frequentemente um componente que apresenta as “vendas por cidade do país X”, de modo a apresentar as “vendas por país do continente Y”, demonstra uma tendência do utilizador visualizar os conjuntos de dados em sequência, sendo possível que este tenha que voltar a visualizar as “vendas por cidade do país X” após a alteração ter ocorrido, tendo para isso que voltar a alterar o componente. É seguindo esta filosofia que se decidiu aplicar o algoritmo de regras de associação a um conjunto de dados no qual se encontram representadas as transições entre configurações. Deste modo o *dashboard*, após ser reestruturado, irá conter tendências apresentadas pelo utilizador ao longo do processo de exploração do mesmo.

O primeiro passo para aplicar o algoritmo é então criar um conjunto de dados que contenha as transições entre configurações que foram sendo registadas ao longo do processo de exploração do *dashboard* que se pretende reestruturar. Como o algoritmo que se vai utilizar faz parte das bibliotecas do WEKA o formato em que os dados se devem encontrar é o mesmo que é utilizado no próprio WEKA. Na Figura 3.24 está representado uma porção do *dataset* utilizado no caso estudo.

```
@relation queries

@attribute q1 {1,2,3,4,5,6,7,8,10,11}
@attribute q2 {1,2,3,4,5,6,7,8,10,11}

@data
1, 8
1, 8
8, 4
1, 9
1, 8
8, 3
1, 8
8, 1
```

Figura 3.24 - Exemplo de um *dataset* utilizado para o cálculo de regras de associação

O formato aqui utilizado é relativamente fácil de compreender. O ficheiro deve ser iniciado com a palavra-chave “@relation” seguida do nome da relação, neste caso “queries”. Em seguida devem vir os atributos que se pretendem utilizar, bem como o conjunto de valores válidos para cada um deles - neste caso temos a configuração antecedente e a configuração precedente como atributos. A palavra-chave “@attribute” sinaliza o atributo e deve ser seguido pelo nome do atributo e pela lista de valores possíveis, que neste caso são os identificadores de todas as configurações existentes. Por fim devem ser enunciados os dados que se pretendem avaliar, utilizando a palavra-chave “@data” para sinalizar o começo dos mesmos. Os dados devem estar em conformidade com os atributos definidos, sendo que os atributos em cada registo devem ser separados por vírgulas. Depois de ter o *dataset* pronto pode-se então proceder à utilização do algoritmo de regras de associação, sendo para isso apenas necessário definir algumas configurações, nomeadamente o valor mínimo de confiança que se pretende, neste caso foi definido um valor mínimo de confiança de 30%, e ainda o valor para o mínimo suporte, que neste caso foi definido em 1%. Na Tabela 3.8 encontram-se as regras geradas pelo algoritmo para o caso de estudo.

	Regra	Confiança
1	q1=3 268 ==> q2=9 181	68%
2	q1=2 298 ==> q2=11 188	63%
3	q2=9 305 ==> q1=3 181	59%
4	q2=11 321 ==> q1=2 188	59%
5	q1=1 264 ==> q2=8 152	58%
6	q2=8 271 ==> q1=1 152	56%
7	q1=6 113 ==> q2=10 45	40%

Tabela 3.8 - Regras de associação geradas para o caso de estudo

Através da análise da Tabela 3.8 podemos ver que foram encontradas sete regras que satisfazem os parâmetros mínimos utilizados. Porém, também é possível observar que algumas delas representam o mesmo par de configurações, o que faz com que seja necessário proceder à sua seleção. Para construir o *dashboard* vão então ser utilizadas as três regras de maior confiança que permitam a construção de um *dashboard* com seis configurações distintas. No caso de estudo podemos ver que as regras a utilizar são as regras número 1, 2 e 5, uma vez que as regras número 3 e 4 representam as mesmas configurações que as regras 1 e 2 respetivamente, sendo as configurações daí resultantes, e que serão utilizadas na reestruturação do *dashboard*, as configurações com os identificadores 3, 9, 2, 11, 1 e 8. Para representar as regras de associação selecionadas no *dashboard* optou-se por utilizar a primeira fila de componentes para albergar as configurações presentes nos antecedentes das regras, e a segunda fila de componentes para albergar as configurações presentes nos consequentes das regras, mantendo desta forma as configurações mais fortemente relacionadas mais próximas. A Figura 3.25 ilustra essa disposição utilizada, onde a primeira regra é representada por A -> B, a segunda regra por C -> D e a última regra por E -> F.

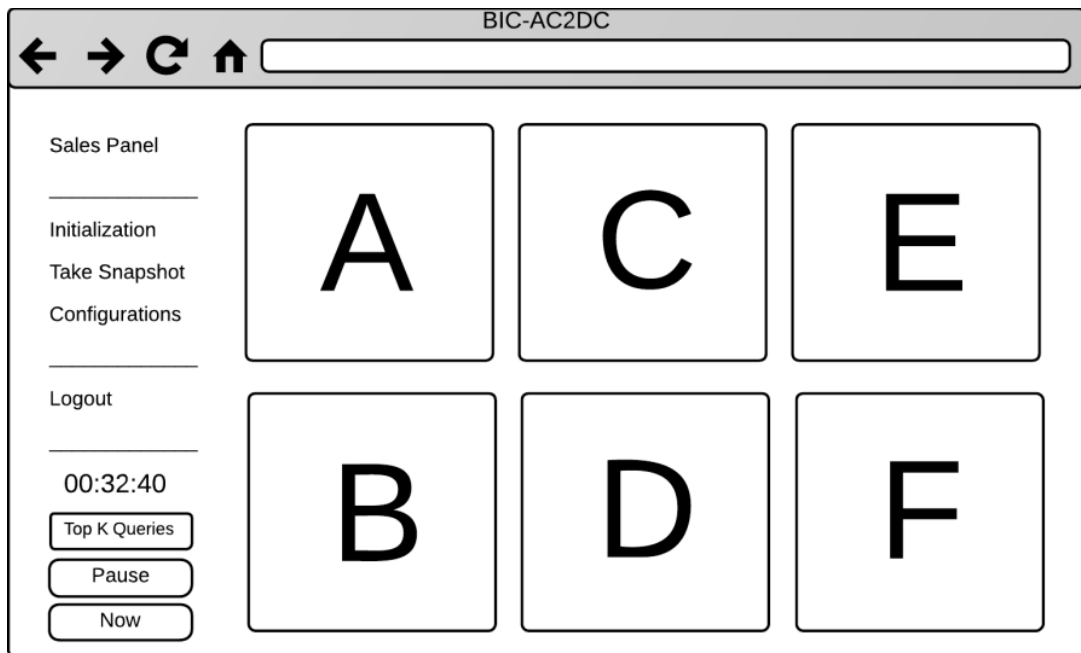


Figura 3.25 - Exemplo de interpretação de um *dashboard* construído utilizando regras de associação

Desta forma temos então um *dashboard* no qual estão representadas as três transições entre configurações de componentes mais relevantes. À partida pode parecer que os três pares de configurações utilizados não estão relacionados, uma vez que são escolhidos de três regras de associação distintas, mas uma vez que todas as configurações envolvidas, mais concretamente os conjuntos de dados utilizados, fazem parte do mesmo *dashboard* todas elas estão intrinsecamente relacionadas. Não fazia sentido dois conjuntos de dados que não tivessem nenhum tipo de relacionamento fazerem parte do mesmo *dashboard*. É seguindo esse princípio, que se afirma que todas as configurações selecionadas constituem um *dashboard* coerente. O *dashboard* resultante da aplicação deste método ao caso de estudo está apresentado na Figura 3.26, na qual podem ser observadas todas as particularidades acima descritas.

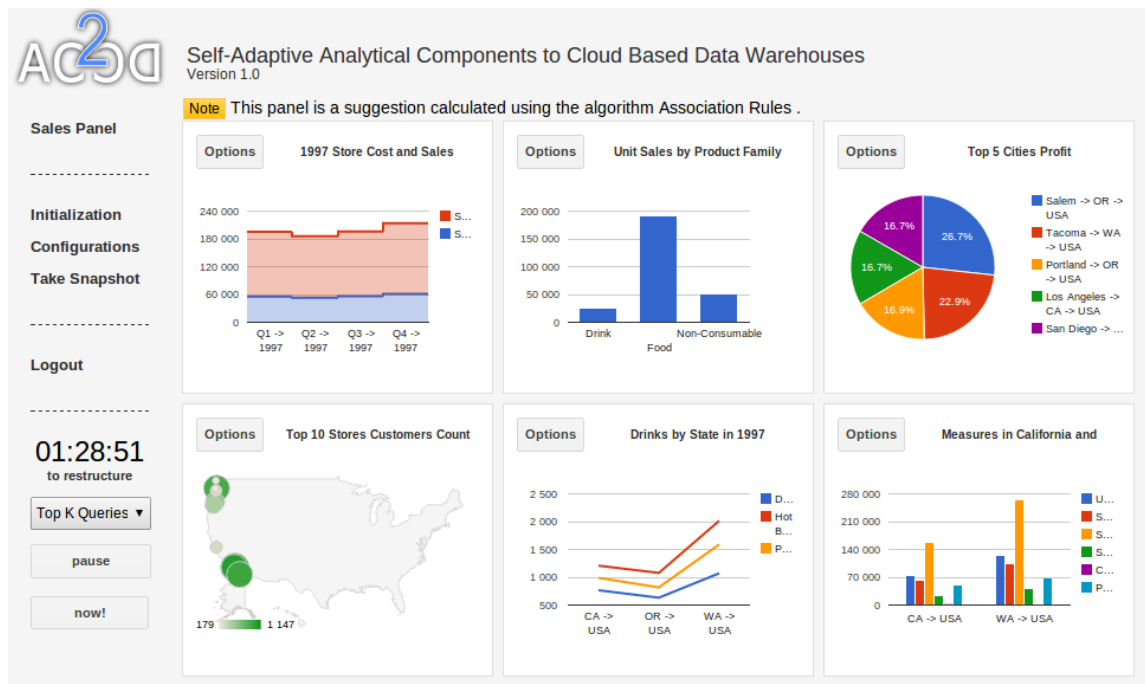


Figura 3.26 - Exemplo de um *dashboard* resultante do processo de reestruturação utilizando o algoritmo de regras de associação

Capítulo 4

Conclusão e Trabalho Futuro

4.1 Conclusões

A importância que a visualização de dados tem nas áreas de *BI* e de *BA* nos dias que correm levou à proliferação dos *dashboards*, graças às enormíssimas quantidades de dados existentes no seio das empresas, contendo informação fundamental ao bom funcionamento e evolução das mesmas. Inúmeras empresas, tanto grandes como pequenas, assim como muitos particulares, começaram a produzir os seus próprios *dashboards* sem que para isso fossem aplicados os conhecimentos necessários, o que levou ao aparecimento de muitas ferramentas de *dashboarding* “defeituosas” sob o ponto de vista funcional. Foi portanto necessário iniciar um processo de definição de boas práticas e *standards*, de modo a garantir que, pelo menos, os *dashboards* que sejam desenvolvidos de hoje em diante sejam mais funcionais e úteis. Alguns especialistas em *dashboards* têm contribuído para a definição e ensinamento dessas boas práticas, quer dando aulas e seminários sobre o assunto, quer escrevendo livros ou artigos científicos nos quais presenteiam o leitor com inúmeros conselhos e guias, que quando usados corretamente levam à criação de *dashboards* de maior qualidade. Stephen Few é uma das referências nesta área, tendo já escrito diversos artigos sobre o assunto, e ainda um dos livros de maior referência no que concerne à criação de *dashboards*, de seu título “Dashboard Design Insight” (2006b).

Nesta dissertação foi realizado um estudo sobre o estado atual da visualização de dados, em particular dos *dashboards*, bem como sobre alguns dos mecanismos de ML que melhor poderiam auxiliar processos automatizados de reestruturação de *dashboards*. Neste estudo começou por ser avaliada a evolução da visualização de dados e dos *dashboards* a fim de perceber quando este começaram a ter relevância e quais as marcas históricas mais significantes à compreensão do

assunto. Em seguida foi feita uma análise dos diversos componentes que podem constituir um *dashboard*, de forma a perceber a complexidade que um *dashboard* pode ter, bem como a miríade de diferentes *dashboards* que podem ser criados. Conhecer as potencialidades destes componentes, perceber em que situações cada um deles deve ser utilizado e como estes se relacionam quando utilizados para um fim comum, é o primeiro passo para conseguir distinguir um bom de um mau *dashboard*. Nunca é demais realçar a importância e atenção que devem ser dadas a cada um dos componentes em particular, bem como ao *dashboard* no seu todo, e ainda ao utilizador final do mesmo, se o objetivo é criar um *dashboard* funcional e adequado às necessidades do utilizador.

Criar um *dashboard* é muito mais do que escolher os gráficos mais apelativos, que dão origem a um *dashboard* visualmente estimulante, é todo um processo de recolha e estudo de informação relacionada com o utilizador que irá conduzir à criação de um sistema capaz de adaptar o conceito de *dashboarding* às suas necessidades. Após perceber a complexidade por de trás dos vários componentes e a potencialidade oferecida por cada um deles, foi necessário estudar algumas das categorizações de *dashboards* mais relevantes de modo a conhecer os diferentes tipos de *dashboards* que podem ser criados. Após este estudo foram avaliadas as vantagens e as funcionalidades que um *dashboard* tem para oferecer, com o objetivo de perceber o que torna estes sistemas de visualização de dados tão importantes e fundamentais para qualquer indivíduo que tenha que lidar com grandes quantidades de dados. Para concluir este estudo foram apresentadas algumas linhas de orientação que devem ser seguidas aquando do desenvolvimento de um *dashboard*, de modo a evitar erros comumente cometidos por pessoas menos conhecedoras desta área e das suas matérias.

Numa segunda fase foi desenvolvido um sistema piloto de suporte à decisão baseado em *dashboards* capaz de adaptar esses mesmos *dashboards* às preferências dos utilizadores. O sistema desenvolvido vai de encontro aos objetivos inicialmente traçados e foi utilizado com sucesso na análise dos três mecanismos de reestruturação implementados. Através da utilização do sistema é possível realizar uma análise real e detalhada do *DW* que atualmente é suportado pelo mesmo, estando o sistema completamente operacional nesse sentido. É ainda possível ao utilizador adaptar os *dashboards* às suas necessidades, podendo este alterar tanto as características gráficas do mesmo como os dados em apresentação, alterações estas que vão sendo registadas pelo sistema de modo a permitir o funcionamento dos mecanismos de reestruturação disponíveis. Uma vez que não foi possível recorrer a utilizadores reais, com

necessidades reais, para testar o sistema de reestruturação, foi necessário proceder à geração de perfis de utilização que apresentassem certos padrões de funcionamento de modo a efetuar alguns testes. Tal contratempo limitou a análise que pode ser feita a cada um dos algoritmos usados no processo de reestruturação uma vez que seria necessário receber *feedback* dos utilizadores de modo a saber se os *dashboards* resultantes deste processo eram úteis para os utilizadores e se consequentemente seriam usados pelos próprios.

4.2 Trabalho Futuro

O sistema piloto que foi desenvolvido é apenas um ponto de partida para um sistema que acreditamos ter enormes potencialidades, nas áreas da visualização e exploração de dados. Podem ser adicionadas inúmeras funcionalidades e ferramentas que tornem a exploração de dados mais rápida e eficaz, como por exemplo introduzir mecanismos de navegação sobre os dados mais evoluídos, ou proporcionar uma melhor adaptação dos componentes ao espaço disponível de modo a facilitar a leitura dos mesmos.

Em relação aos mecanismos de reestruturação é possível, ainda, introduzir novos métodos utilizando alguns dos algoritmos aqui apresentados ou até mesmo outros, e que não foram implementados, uma vez que qualquer um deles tem potencial para atingir resultados bastante interessantes em contextos de visualização de dados em que uma característica como a auto-adaptação seja, obviamente, um elemento claro de diferenciação. Seria ainda interessante realizar uma análise num ambiente real do funcionamento destes mecanismos de reestruturação, de modo a avaliar quais os que apresentam melhores resultados, e em que situações, permitindo assim afiná-los e até mesmo combinar a utilização de vários algoritmos, a fim de tornar estes mecanismos numa realidade no mundo dos *dashboards*.

Bibliografia

Adams, D., n.d. *Data Visualization*. [Online] Disponível em: http://www.officeadvisor.com/media/Accenture_White_Paper.pdf [Acedido em 26 de Janeiro de 2013].

Allen, S., 2010. *A Brief History of Visualization*. [Online] Disponível em: <http://interactiondesign.sva.edu/classes/datavisualization/2010/07/08/introduction/> [Acedido em 2 de Fevereiro de 2013].

Billsus, D. & Pazzani, M.J., 2000. *User Modeling for Adaptive News Access*. Irvine, CA, USA: Kluwer Academic Publishers Dept. of Information and Computer Science, University of California.

Blewett, R., n.d. *The Importance of Data Visualization to Business Decision Making*. [Online] Disponível em: <http://www.dundas.com/dashboard/resources/articles/the-importance-of-data-visualization-1.aspx> [Acedido em 4 de Fevereiro de 2013].

Cleveland, W. & McGill, R., 1984. Journal of the American Statistical Association. *Journal of the American Statistical Association*, 79(387), pp.531-32.

Corbett, J., n.d. *Charles Joseph Minard: Mapping Napoleon's March, 1861*. [Online] Disponível em: <http://www.csiss.org/classics/content/58> [Acedido em 2 de Fevereiro de 2013].

Dundas Data Visualization Inc., n.d. *The Dashboard Demystified*. [Online] Disponível em: http://www.dundas.com/dashboard/resources/articles/dashboard_demystified.aspx [Acedido em 27 de Janeiro de 2013].

Few, S., 2004a. *Dashboard Confusion*. [Online] Disponível em: http://www.perceptualedge.com/articles/ie/dashboard_confusion.pdf [Acedido em 25 de Outubro de 2012].

Few, S., 2004b. *Bullet Graph Design Specification*. [Online] Disponível em: http://www.perceptualedge.com/articles/misc/Bullet_Graph_Design_Spec.pdf [Acedido em 26 de Outubro de 2012].

Few, S., 2006a. *Common PITFALLS in Dashboard Design*. [Online] Disponível em: http://www.perceptualedge.com/articles/Whitepapers/Common_Pitfalls.pdf [Acedido em 25 de Outubro de 2012].

Few, S., 2006b. *Information Dashboard Design*. Sebastapol, California: O'Reilly Media Inc.

Few, S., 2006c. In *Information Dashboard Design*. Sebastapol, California: O'Reilly Media, Inc. pp.40-46.

Few, S., 2006d. Introduction. In *Information Dashboard Design*. Sebastapol, California: O'Reilly Media, Inc.

Few, S., 2006e. Thirteen Common Mistakes in Dashboard Design. In *Information Dashboard Design*. Sebastapol, California: O'Reilly Media, Inc.

Few, S., 2007. *Data Visualization Past, Present and Future*. [Online] Cognos Disponível em: http://www.perceptualedge.com/articles/Whitepapers/Data_Visualization.pdf [Acedido em 1 de Fevereiro de 2013].

Gemignani, Z., 2008. *A Dashboard Alerts Checklist*. [Online] Disponível em: <http://www.juiceanalytics.com/writing/dashboard-alerts-checklist/> [Acedido em 30 de Janeiro de 2013].

Gena, C., 2005. Methods and techniques for the evaluation of user-adaptive systems. *Cambridge University Press*.

Hirst, G., DiMarco, C., Hovy, & Parsons, , 1997. Authoring and Generating Health-Education Documents That Are Tailored to the Needs of the Individual Patient. In *User Modeling: Proceedings of the Sixth International Conference, UM97.*, 1997.

Jameson, A., 2001. User-Adaptive and Other Smart Adaptive Systems: Possible Synergies. In *First EUNITE Symposium*. Tenerife, 2001.

-
- Juice Analytics, 2009. *A Guide to Creating Dashboards People Love to Use*. [Online] Disponível em: <http://pt.scribd.com/doc/24699115/Guide-to-Dashboard-Design> [Acedido em 5 de Fevereiro de 2013].
- Malik, S., 2005. In *Enterprise Dashboards: Design and Best Practices for IT*. Hoboken, New Jersey: John Wiley & Sons, Inc. pp.95-144.
- Malik, S. & Germain, D., 2009. *Dashboards: Data Visualization for all Elements of the Business*. [Online] Disponível em: http://www.information-management.com/specialreports/2009_136/dashboard_data_visualization-10015173-1.html [Acedido em 1 de Fevereiro de 2013].
- Mezhoudi, N., 2013. *User Interface Adaptation based on User Feedback and Machine Learning*. Louvain: Université catholique de Louvain.
- Microsoft Corporation, 2010? *Getting started with SharePoint status indicators*. [Online] Disponível em: <http://office.microsoft.com/en-us/sharepoint-server-help/getting-started-with-sharepoint-status-indicators-HA010380634.aspx?CTT=5&origin=HA010380601> [Acedido em 28 de Janeiro de 2013].
- Morris, W., 2010. *Consistency versus Personalization*. [Online] Disponível em: <http://www.dashboardinsight.com/articles/digital-dashboards/fundamentals/consistency-versus-personalization.aspx> [Acedido em 1 de Outubro de 2013].
- Motti, V.G., Mezhoudi, N. & Vander, J., 2012. Machine Learning in the Support of Context-Aware Adaptation. In *Context-Aware Adaptation of Service Front-Ends 2012.*, 2012.
- Pauwels, K. et al., 2008. *Dashboards & Marketing: Why, What, How and What Research is Needed?* [Online] Disponível em: <http://www.marketing.uni-frankfurt.de/fileadmin/Publikationen/MSI-Marketing-Dashboard-Paper.pdf> [Acedido em 24 de Outubro de 2012].
- Slaughter, A., 2004. *OLAP Introduction*. [Online] Disponível em: <http://web.mit.edu/profit/PDFS/SlaughterA.pdf> [Acedido em 25 de Outubro de 2012].
- Spence, I., 2004. *Playfair, William (1759–1823), inventor of statistical graphs and writer on political economy*. [Online] Disponível em: <http://www.oxforddnb.com/view/printable/22370> [Acedido em 4 de Fevereiro de 2013].
-

TIOBE Software BV, 2013. *TIOBE Programming Community Index for September 2013*. [Online] Disponível em: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> [Acedido em 4 de Outubro de 2013].

Tufte, E., 1983. *The Visual Display of Quantitative Information*. 2nd ed. Cheshire: Graphics Press.

Tukey, J., 1977. *Exploratory Data Analysis*. Addison-Wesley.

Wallis, J.C., 2005. *CENS Edu/DM Compendium*. [Online] Disponível em: <http://www.moebiusstrip.org/CENS/DataVisTechRep.doc> [Acedido em 4 de Fevereiro de 2013].

Wikipedia, 2013. *Anscombe's quartet*. [Online] Disponível em: http://en.wikipedia.org/wiki/Anscombe's_quartet [Acedido em 5 de Fevereiro de 2013].

Witten, I.H., Frank, E. & Hall, M.A., 2011a. Algorithms: The Basic Methods. In *Data Mining Practical Machine Learning Tools and Techniques*. 3rd ed. Burlington: Morgan Kaufmann Publishers. p.138.

Witten, I.H., Frank, E. & Hall, M.A., 2011b. Preface. In *Data Mining Practical Machine Learning Tools and Techniques*. 3rd ed. Burlington: Morgan Kaufmann Publishers.