

**REGRAS DE ASSOCIAÇÃO APLICADAS A UM MÉTODO DE
APOIO AO PLANEAMENTO DE RECURSOS HUMANOS**

por

Miguel José Pires da Silva Almeida Veloso

Tese de Mestrado em Análise de Dados e Sistemas de Apoio à Decisão

Orientada por

Prof. Doutor Alípio Jorge

Faculdade de Economia

Universidade do Porto

2003

“The secret of success is to know something nobody else knows”.

- Aristotle Onassis

Nota Biográfica

Miguel José Pires da Silva Almeida Veloso nasceu em Maputo (ex-Lourenço Marques) – Moçambique - a 1 de Julho de 1971. Em 1996 concluiu a licenciatura em Engenharia de Sistemas e Informática pela Universidade do Minho. Iniciou a sua actividade profissional na *Lusodata – Sistemas Informáticos* onde ganhou larga experiência no desenvolvimento de *software* através de ferramentas *CASE* orientadas aos objectos. Em 1998 aceitou o desafio de se deslocar para a cidade de Lisboa para integrar a equipa de *Management Consulting* - grupo de *Information Technologies* - da *Ernst & Young* Portugal. Actualmente, e desde o ano 2000, exerce funções de gestão de projecto na *Enabler – Solutions for Retailing*, onde tem desenvolvido competências na área de *Data Wharehouse / Business Intelligence*.

Agradecimentos

Em primeiro lugar quero endereçar um agradecimento muito especial ao Prof. Doutor Alípio Jorge pela forma dedicada como conduziu a orientação desta tese.

Ao Jorge Brás pela autorização que me concedeu para poder utilizar dados da *Enabler* na elaboração do modelo apresentado neste documento.

Ao Jorge Santos e ao José Ribas pelo apoio concedido.

Ao Prof. Doutor Paulo Azevedo pelos conselhos e pela ajuda na utilização do *CAREN* e ao projecto de investigação POSI/2001 Class.

A todos os meus colegas de trabalho que amavelmente acederam ao meu pedido para o preenchimento do inquérito que será apresentado.

À Alina, ao João e aos meus pais por todo apoio e ajuda.

Resumo

Nesta dissertação foi explorada a utilização de regras de associação para tarefas de recomendação, tendo como aplicação prática um caso de apoio à escolha de recursos humanos para a constituição de equipas em projectos. Foi proposta uma metodologia de apoio à decisão com base em modelos derivados a partir de dados históricos de utilização dos recursos humanos da empresa. Para sustentar esta metodologia foi desenvolvido um sistema de recomendação que utiliza um modelo de filtragem colaborativa baseado em regras de associação. As recomendações são efectuadas de duas formas: primeiro, recomendando um único elemento da equipa (recurso), dada uma equipa de projecto parcialmente constituída; segundo, recomendando alterações a uma equipa completa, previamente constituída. A avaliação foi feita a vários níveis: estimação das características preditivas dos modelos; adequação dos resultados aos objectivos da empresa – através da análise de um inquérito elaborado para medir a percepção dos potenciais utilizadores deste sistema, face à adequação das recomendações produzidas por este. O caso foi abordado seguindo a metodologia *CRISP-DM*.

Abstract

The subject of this thesis is the use of association rules for recommendation tasks, applied to a case of supporting human resources selection in building project-teams. A methodology of decision support is proposed, grounded on the basis of models built on historical data related to company's human resources policy. In order to sustain this methodology, it was developed a recommendation system that uses a collaborative filtering model based on association rules. Recommendation is made at two levels: first by recommending a single team element given a partially built team; and second by recommending changes to a complete team. Assessment is made at several levels: estimation of the models' predictive characteristics; appropriateness of the results to the company's goals – through a users' perception survey. The case was developed following the *CRISP-DM* methodology.

Índice

1	INTRODUÇÃO	1
2	REGRAS DE ASSOCIAÇÃO	7
2.1	INTRODUÇÃO	7
2.2	DESCOBERTA DE REGRAS DE ASSOCIAÇÃO	9
2.3	SELECÇÃO DE REGRAS	12
2.4	PÓS PROCESSAMENTO E EXPLORAÇÃO DE REGRAS DE ASSOCIAÇÃO	18
2.5	RESUMO DO CAPÍTULO	19
3	APLICAÇÃO DE REGRAS DE ASSOCIAÇÃO.....	20
3.1	CLUSTERING DE REGRAS DE ASSOCIAÇÃO	20
3.2	REGRAS DE ASSOCIAÇÃO PARA CLASSIFICAÇÃO	22
3.3	SISTEMAS DE RECOMENDAÇÃO	24
3.4	AValiação de SISTEMAS DE RECOMENDAÇÃO	29
3.5	SISTEMAS DE RECOMENDAÇÃO E REGRAS DE ASSOCIAÇÃO	32
4	RECOMENDAÇÃO DE RECURSOS HUMANOS PARA EQUIPAS DE PROJECTOS.....	35
4.1	METODOLOGIA	36
4.2	COMPREENSÃO DO NEGÓCIO	40
4.2.1	Caracterização da Enabler.....	40
4.2.2	Definição do Problema	48
4.3	COMPREENSÃO DOS DADOS	53
4.3.1	Análises Preliminares	53
4.3.2	Análise Exploratória.....	57
4.3.3	Análise dos Custos e dos Conjuntos de Recursos.....	59
5	PREPARAÇÃO DE DADOS E MODELAÇÃO	65
5.1	CONSTRUÇÃO DO MODELO.....	65
5.2	RESULTADOS EXPERIMENTAIS	69
5.3	EXPERIÊNCIAS ADICIONAIS	76
5.3.1	Modelo com Regras Default	76
5.3.2	Impacto da informação disponível nos resultados.....	78
5.3.3	Utilização do Interest para selecção de regras	80
5.4	RECOMENDAÇÃO DE EQUIPAS	84
5.5	RESUMO DO CAPÍTULO	90
6	AValiação	91

6.1	PERCEPÇÃO DOS UTILIZADORES	91
6.2	DISCUSSÃO	94
7	OPERACIONALIZAÇÃO	96
7.1	PROPOSTA DE IMPLEMENTAÇÃO	96
7.2	MODELO DE RESTRIÇÕES	98
7.3	RECOMENDAÇÃO DE EQUIPAS	101
8	CONCLUSÕES E TRABALHO FUTURO	102
	EPÍLOGO.....	107
	REFERÊNCIAS.....	108
	ANEXO 1 SÍNTESE DA METODOLOGIA CRISP-DM	116
	BUSINESS UNDERSTANDING	116
	DATA UNDERSTANDING	116
	DATA PREPARATION	117
	MODELING.....	118
	EVALUATION	118
	DEPLOYMENT	119
	ANEXO 2 ANÁLISES MULTIVARIADAS.....	120
	ANÁLISE CLASSIFICATÓRIA HIERÁRQUICA	121
	CLASSIFICAÇÃO	123
	ANÁLISE DE COMPONENTES PRINCIPAIS	125
	<i>Recursos</i>	125
	<i>Projectos</i>	128
	ANEXO 3 CAREN	132
	ANEXO 4 PROGRAMAS EM R.....	133
	ANEXO 5 QUESTIONÁRIO.....	139

1 Introdução

O fenómeno da globalização [Hill, Charles W. L. (2001)], em relação ao qual as tecnologias de informação têm hoje em dia grande responsabilidade, obriga a que todos os agentes que intervêm na sociedade, em particular as empresas, estejam receptivos e preparados para a mudança, por forma a garantir a sua sobrevivência num mercado mais amplo e competitivo [Gordon, S. R. et al. (2003)].

As empresas dos dias de hoje localizam-se praticamente onde querem em busca das melhores condições para cumprirem as suas missões. Este facto só tem sido possível à custa dos progressos verificados nas últimas décadas nas telecomunicações e na capacidade de computação. [Laudon, K. C. et al. (2002)].

As possibilidades quase ilimitadas, apesar de ainda não serem completamente compreendidas, oferecidas actualmente pela Internet às empresas, permitem que estas obtenham uma penetração de mercado tal, que não lhes é difícil estar em todo o lado ao mesmo tempo: permite-lhes dar a conhecer em todo o lugar e em qualquer momento, a sua oferta de valor (o que têm para vender), e, inversamente, permite-lhes procurar em todo o mundo o que necessitam de comprar. Este paradigma de fazer negócio, *e-business*, complementa a visão tradicional que caracteriza a relação entre empresas (*B2B - Business to Business*); e entre empresas e clientes finais (*B2C - Business to Consumer*) [Gordon, S. R. et al. (2003)].

Porém, a Internet e a globalização exigirão agilidade e rapidez nos processos de negócio. Quando a empresa entrar na fase inevitável do comércio electrónico, torna-se necessário que internamente a sua cadeia de valor seja leve, assente em processos de negócio altamente integrados. Uma tal integração exige que os sistemas de informação da empresa estejam dotados com pacotes de *software* do tipo *ERP (Enterprise Resource Planning)* [Gordon, S. R. et al. (2003)]. Este tipo de aplicativos permitem automatizar processos manuais – incompatíveis com o comércio electrónico - e reduzir custos operacionais [Laudon, K. C. et al. (2002)].

A presença dos computadores em praticamente todo lado é, hoje em dia, uma realidade inquestionável. A grande maioria dos processos que nos rodeiam são grandemente suportados por meios computacionais. Como resultado destas evidências, verifica-se que a quantidade de informação disponível actualmente é absolutamente brutal: durante o séc. XX, o volume de informação gerado e mantido por algumas empresas cresceu cerca de 100.000 vezes [Berry, Michael J. A. et al. (2000)]! A informação assume-se assim como um factor vital e indispensável para a gestão e competitividade das empresas, sendo que esta pode e deve ser considerada como um activo de elevada importância neste enquadramento. Isto é, estamos em plena era da “revolução da informação e do conhecimento” [Turban, Efraim et al. (2001)].

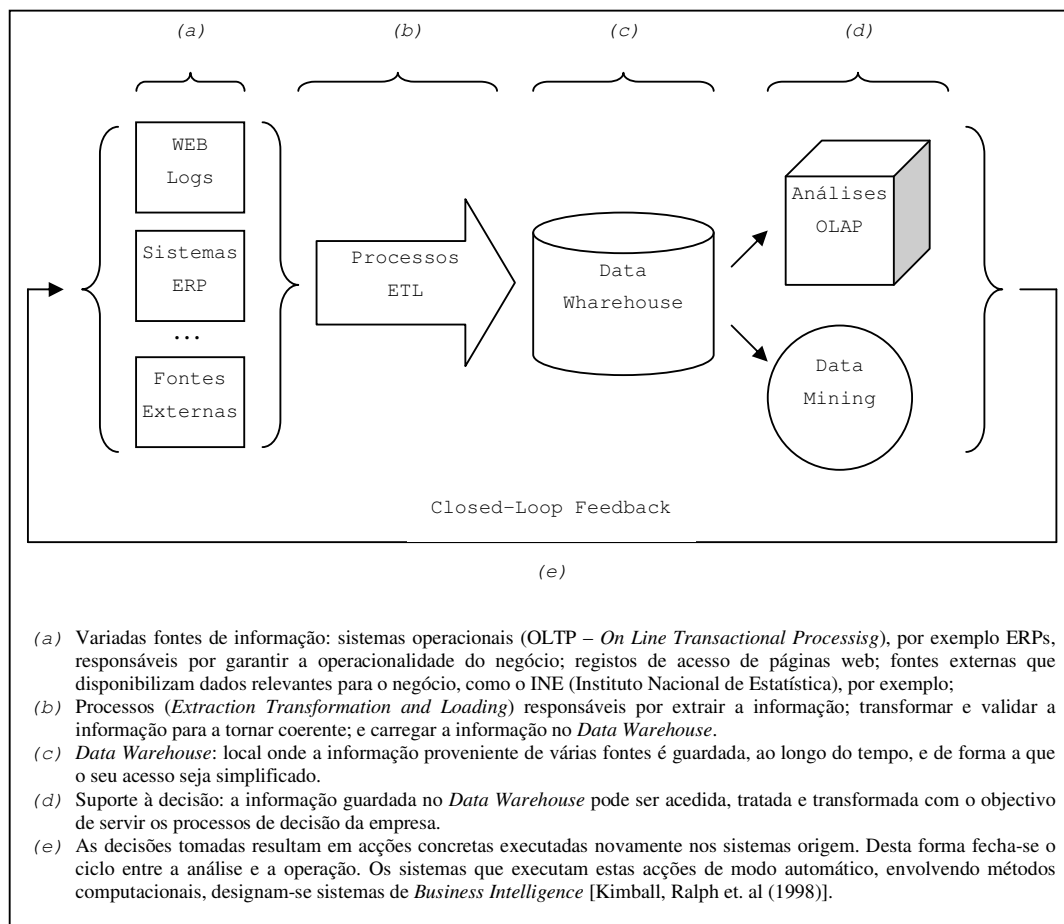
Com o objectivo de dotar as organizações com capacidades de “memória”, surgiram neste contexto nos últimos anos, os sistemas do tipo *Data Warehouse* [Inmon, W. H. (1996)], cuja função é o armazenamento eficaz e coerente da informação gerada ao longo do tempo pelas várias fontes de informação associadas a uma empresa. Pelas suas características técnicas, é suposto que um *Data Warehouse* disponibilize a informação certa, no sítio certo, no tempo certo, com o custo certo, no sentido de suportar as decisões certas. Usualmente, a análise da informação armazenada num *Data Warehouse* é efectuada por ferramentas baseadas em tecnologia *OLAP – On Line Analytical Processing* [Jarke, Matthias et. al (2003)], [Kimball, Ralph (1996)], [Westerman, Paul (2001)].

Uma vez que estes volumes de informação podem esconder padrões interessantes e úteis do ponto de vista de negócio, de que forma é que é possível converter esta informação em conhecimento?

Variados sectores do meio científico, designadamente o estatístico e o informático, têm-se empenhado no sentido de serem descobertas técnicas capazes de transformar estes imensos volumes de informação em conhecimento. Em termos computacionais designa-se vulgarmente esta transformação por: *Data Mining*, *Pattern Recognition*, ou, *Knowledge Discovery in Data Bases (KDD)* [Witten, Ian H. et al. (2000)], [Ripley, B. D. (2001)]. A aplicabilidade prática das acções de *Data Mining* são várias [Berry,

Michael J. A. et al. (1997)], [Mena, Jesus (1999)]: apoio à investigação científica; controlo eficiente de processos de fabrico; marketing; *CRM – Customer Relationship Management*; criação de web sites dinâmicos – e-marketing; detecção de fraude; etc.

A figura que se segue [Kimball, Ralph et. al (1998)] apresenta, de forma abstracta e obviamente simplificada - em virtude deste contexto - uma proposta de visão conceptual da arquitectura de sistemas de informação de uma empresa da actualidade:



Fundamentalmente, as tarefas do *Data Mining* são [Berry, Michael J. A. et al. (2000)]: Classificação, Regressão, Previsão, *Clustering* e Associação. O quadro seguinte sintetiza cada uma destas tarefas, com exemplos, e apresenta algumas das técnicas mais utilizadas para as executar [Sharma, Subhash (1996)], [Adamo, Jean-Marc (2001)], [Witten, Ian H. et al. (2000)]:

Classificação	Ex: Classificar clientes: bons, médios e maus; conceder crédito, ou não conceder crédito; indemnização fraudulenta, ou não; cliente vai abandonar, ou não.	Análise Discriminante; Árvores de Decisão; Classificadores Bayesianos
Regressão	Caso particular da classificação: classe a prever quantitativa. Ex: Estimar o rendimento total de uma família; lucro de um negócio.	Regressão Linear; Regressão Local; <i>Nearest Neighbour</i> ; Árvores de Regressão; Redes Neurais; Algoritmos Genéticos
Previsão	Caso particular da classificação e da regressão: considera o factor tempo. Ex: Prever a evolução de cotações em bolsa.	Séries temporais
Clustering	Agrupar "casos" mais semelhantes entre si. Ex: <i>Clusters</i> de clientes ou de produtos.	Classificação hierárquica; <i>K-means</i>
Associação	Ex: <i>Market Basket Analysis</i>	Regras de Associação

Existe neste momento um conjunto vasto de ferramentas de *Data Mining*, tais como, por exemplo: Intelligent Miner da IBM¹; Clementine da SPSS²; Enterprise Miner da SAS³; S-Plus e Insightful Miner 2 da Insightful Corporation⁴; See5/C5.0, Cubist e Magnum Opus da RuleQuest⁵ e o R⁶. Empresas como a Microsoft⁷ e a Oracle⁸, estão atentas a estas movimentações do mercado e começam também a apresentar soluções neste domínio. Academicamente existem também alguns casos de soluções para *Data Mining*, como o Weka⁹ por exemplo - este *software* acompanha e é um complemento de Witten, Ian H. et al. (2000).

¹ www.ibm.com

² www.spss.com

³ www.sasinstitute.com

⁴ www.insightful.com

⁵ www.rulequest.com

⁶ www.r-project.org

⁷ www.microsoft.com

⁸ www.oracle.com

⁹ www.cs.waikato.ac.nz/~ml/weka/

Este tema é muito vasto e apesar de ser extremamente envolvente e interessante, o foco principal desta tese gravita à volta da aplicação prática das regras de associação. Em particular, estudará a aplicação de um modelo de regras de associação a um método de apoio ao planeamento de recursos humanos em equipas de projectos.

A actividade de planeamento de equipas é fundamental para as empresas prestadoras de serviços, quando estes estão estruturados e organizados por projectos: as vendas deste tipo de empresas são materializadas através de projectos com determinada duração e determinado objectivo; cada projecto tem associado um conjunto de recursos (humanos) da empresa para executar as tarefas necessárias; cada recurso pode estar associado a mais do que um projecto. No entanto, há uma série de desafios com que os responsáveis por efectuar esta actividade se deparam. Neste enquadramento, esta tese propõe o desenvolvimento e a utilização de um sistema de recomendação de recursos humanos, baseado em regras de associação, cujo objectivo é auxiliar e acrescentar valor ao referido processo de planeamento, ou seja, permitir que o responsável por esta actividade possa constituir as equipas de projecto mais adequadas, através das recomendações disponibilizadas por este sistema, independentemente do seu conhecimento acerca dos recursos humanos da empresa, da dimensão da empresa, das características do projecto e do cliente respectivo.

Ao longo desta tese serão estudadas as seguintes hipóteses:

- Se um sistema de recomendação baseado em regras de associação é aplicável ao domínio do planeamento de recursos humanos para equipas de projectos.
- Se o sistema desenvolvido e proposto responde aos desafios que irão ser apresentados.
- Se o conhecimento necessário para efectuar o planeamento de equipas pode ser obtido a partir dos dados resultantes da actividade da própria empresa ou organização utilizando técnicas automáticas de descoberta desse conhecimento.

Este documento está organizado da seguinte forma:

- No capítulo 2 *Regras de Associação* será feita uma introdução breve do conceito “regra de associação” e serão abordadas as problemáticas associadas aos algoritmos para descoberta de regras, à selecção de regras e ao pós processamento e exploração de regras de associação.
- Algumas aplicações práticas das regras de associação serão referidas no capítulo 3 *Aplicação de Regras de Associação*. Será dada especial ênfase à aplicação de regras de associação a sistemas de recomendação.
- O capítulo 4 *Recomendação de Recursos Humanos para Equipas de Projectos* apresentará o caso prático utilizado para desenvolver e testar o modelo proposto por esta tese. Este modelo foi desenvolvido segundo a metodologia *CRISP-DM* que também será apresentada no início deste capítulo.
- No capítulo 5 *Preparação de Dados e Modelação* serão descritos os passos que foram dados para preparar os dados, para construir o modelo e para o avaliar experimentalmente.
- A avaliação deste modelo segundo o conceito do *CRISP-DM*, ou seja, segundo os objectivos do negócio é apresentada no capítulo 6 *Avaliação*. Esta avaliação foi efectuada analisando a percepção dos potenciais utilizadores deste sistema, face às recomendações produzidas por este, através dos resultados de um inquérito organizado para o efeito.
- No capítulo 7 *Operacionalização* será apresentada uma proposta de implementação deste sistema.
- As conclusões e a apresentação de trabalho futuro serão apresentados no capítulo 8 *Conclusões e Trabalho Futuro*, ao que se seguirão as referências e os anexos.

2 Regras de Associação

2.1 Introdução

A descoberta de regras de associação é hoje em dia uma das mais populares tarefas de *Data Mining*. A isso se deve a grande aplicabilidade em problemas de negócio reais e ao facto de serem de fácil compreensão, mesmo para não peritos em *Data Mining* [Hipp, J. et al. (2000)].

Tipicamente, a sua aplicação é efectuada em contextos de empresas da área de retalho, em particular em análises do tipo *market basket analysis* – um cliente que compra x , compra y com a probabilidade $c\%$. O conhecimento que é extraído através deste método pode ser aplicado com o objectivo de aumentar potencialmente o volume de vendas: criação de uma promoção (por exemplo: oferecer o produto y a quem comprar x); colocar os produtos x e y lado a lado na loja para promover a compra por impulso; etc. Foi neste enquadramento que este conceito foi apresentado pela primeira vez [Agrawal, R. et al. (1993)]. No entanto, torna-se óbvio que estas técnicas têm uma excelente aplicabilidade em vastos sectores de actividade sem ser o retalho [Hipp, J. et al. (2000)].

Resumidamente, uma regra de associação é uma expressão $X \Rightarrow Y$, onde X e Y são conjuntos de *itens*. O significado desta expressão é o seguinte: dada uma base de dados D de transacções (onde cada transacção $T \in D$ é um conjunto de *itens*), $X \Rightarrow Y$ representa que quando uma transacção T contém X , então T provavelmente também contém Y . Esta probabilidade observada nos dados, a que chamamos *confiança* da regra, define-se como sendo a percentagem de transacções que contém X em conjunto com Y , em relação às transacções que contém X . Ou seja, a confiança de uma regra pode ser vista como um estimador da probabilidade condicional de Y ocorrer dado que X se observa:

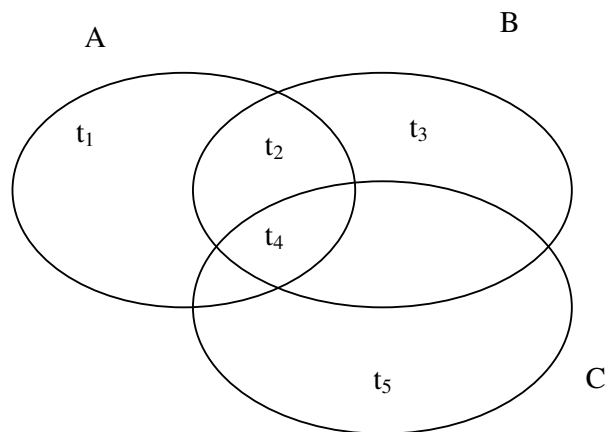
$$conf(X \Rightarrow Y) = P(Y | X) = \frac{P(X \cap Y)}{P(X)}$$

À probabilidade observada de ocorrer X e Y em simultâneo (numerador da expressão em cima), chamamos *suporte* da regra. Enquanto que a confiança mede a força da regra, o suporte corresponde à significância estatística da mesma [Tan, P-N. et al. (2000)].

Considere-se o seguinte exemplo com 5 transacções e 3 *itens*, onde os t_i 's representam as 5 transacções; e A, B e C representam os 3 *itens*. Se uma linha tiver um “1” significa que esse item está presente nessa transacção; caso contrário, esse item não está presente nessa transacção.

	A	B	C
t_1	1	0	0
t_2	1	1	0
t_3	0	1	0
t_4	1	1	1
t_5	0	0	1

O mesmo exemplo, representado por um *diagrama de venn*:



Para a regra $A \Rightarrow B$ temos:

$$\text{Suporte}(A) = P(A) = 3/5$$

$$\text{Suporte}(B) = P(B) = 3/5$$

$$\text{Suporte}(A \Rightarrow B) = P(A \cap B) = 2/5$$

$$\text{Confiança}(A \Rightarrow B) = P(B | A) = P(A \cap B) / P(A) = (2/5) / (3/5) = 2/3$$

Isto significa que se A está presente numa transacção (se um cliente compra A, ...), então B tem $(2/3) * 100 = 66,67\%$ de probabilidade de estar presente também (... então, esse cliente compra B com uma probabilidade de 66,67%). Esta regra aplica-se a $(2/5) * 100 = 40\%$ dos casos – suporte da regra.

De seguida será abordada a problemática associada à descoberta de regras de associação.

2.2 Descoberta de Regras de Associação

Quando este conceito foi introduzido pela primeira vez [Agrawal, R. et al. (1993)], foi apresentado igualmente um algoritmo (designado AIS) para a descoberta deste tipo de regras. Este algoritmo caracteriza-se por produzir regras cujos consequentes possuem apenas um *item*.

A mecânica que está por trás dos algoritmos para a descoberta deste tipo de regras pode ser dividida em dois subproblemas:

- 1 – Geração de *Large Itemsets*, isto é, geração de todas as combinações de *items* que tenham suporte transaccional superior a um determinado limite – *minsupport*.
- 2 – Para um dado *Large Itemset*, gerar todas as regras que tenham confiança superior a um determinado limite – *minconf*.

Os algoritmos dedicados à descoberta de *large itemsets* efectuam múltiplas passagens pelos dados. Durante a primeira passagem, contam o suporte dos *items* individuais e determinam quais destes é que são *large* (frequentes), ou seja, quais é que têm suporte mínimo. Em cada passagem subsequente, estes algoritmos baseiam-se no conjunto de *itemsets* considerados *large* na passagem anterior. Este conjunto inicial é utilizado para gerar novos *itemsets* potencialmente *large*, designados por *itemsets candidatos*. De seguida, é contado o suporte actual para estes *itemsets* candidatos, durante a passagem pelos dados. No fim desta contagem, são determinados quais os candidatos efectivamente *large*, os quais serão a base para o próximo passo. Este processo acaba quando não existirem mais *itemsets* novos.

No AIS a contagem dos *itemsets* candidatos é feita durante a passagem pelos dados. Mais especificamente, após ler uma transacção é determinado qual dos *itemsets* considerado *large* na passagem anterior pelos dados é que está presente na transacção.

Novos *itemsets* candidatos são gerados estendendo estes *itemsets large* com outros *itens* na transacção.

No entanto a estratégia do AIS resulta na geração de *itemsets* desnecessários e na contagem de demasiados candidatos que depois não se revelam frequentes.

Com o objectivo de otimizar a geração de candidatos do AIS, surgem então os algoritmos *Apriori* e *AprioriTID* [Agrawal, R. et al. (1994)].

Estes algoritmos geram os *itemsets* candidatos a ser contados numa passagem, utilizando apenas os *itemsets* considerados frequentes na passagem anterior – sem considerar as transacções da base de dados. Fundamentalmente, a intuição básica é que cada *subset* de um *itemset large* tem de ser também *large*. Por este motivo, a geração de *itemsets* candidatos com k *itens* pode ser gerada juntando *itemsets large* com $k-1$ *itens*, e eliminando aqueles que contenham qualquer *subset* que não é *large*. Este procedimento resulta na geração de um número muito menor de *itemsets* candidatos.

Entre os dois novos algoritmos apresentados em [Agrawal, R. et al. (1994)], a diferença reside fundamentalmente na forma como gerem as passagens pela base de dados. O *Apriori* passa sempre pela base de dados para contar o suporte dos *itemsets* candidatos; o *AprioriTID* apenas passa pela base de dados para contar o suporte dos candidatos de tamanho 1, sendo que guarda em memória os *itemsets large* para as passagens subsequentes.

As inúmeras experiências que os autores levaram a cabo, revelaram que qualquer um destes algoritmos apresentou um desempenho muito superior em relação ao outro algoritmo para descoberta de regras de associação: o AIS. As experiências testaram vários níveis de suporte mínimo e, também, vários *data sets*.

Comparando o *Apriori* e o *AprioriTID*, verificou-se que o *AprioriTID* tem um desempenho superior quando o conjunto de candidatos cabe em memória, e inferior caso contrário. Neste contexto foi proposta uma solução algorítmica híbrida –

AprioriHybrid – cuja estratégia é utilizar o *Apriori* quando o conjunto de candidatos não cabe em memória, passando então o algoritmo para o *AprioriTID*, quando o conjunto de candidatos cabe em memória.

Partindo desta plataforma de trabalho, muitos outros autores propuseram alternativas no sentido de obter desempenhos superiores aos algoritmos *Apriori* e *AprioriTID*, na tarefa da descoberta de regras de associação.

Em Savasere, A. et al. (1995) foi proposto um algoritmo paralelizável (*partition*) cujo modo de funcionamento é o seguinte:

- a base de dados é dividida em partições que caibam em memória;
- para cada partição identifica-se o conjunto de candidatos;
- um conjunto frequente aparece pelo menos numa partição (prova-se que);
- ao fim da primeira passagem tem-se o super conjunto dos candidatos;
- uma segunda passagem conta em toda a base de dados os conjuntos candidatos.

Esta proposta revelou-se muito eficiente e os resultados experimentais levados a cabo, mostraram melhores resultados do que os obtidos com o *Apriori* (o *partition* garante no máximo duas passagens pela base de dados).

Outro trabalho nesta área foi Toivonen H. (1996). Aqui o autor propôs efectuar uma amostra da base de dados para obter o conjunto dos candidatos. Esta estratégia baseia-se no facto do tempo para a descoberta deste conjunto crescer de forma linear com o tamanho da amostra. De seguida conta-se o suporte deste conjunto no resto dos dados. À custa da definição de fronteira negativa, consegue-se controlar se o processo falhou algum conjunto frequente na amostra dos dados, ou não. Em Domingo C. et al. (1998) este conceito foi estendido, sendo que a amostra cresce durante a execução, em função das necessidades – amostra dinâmica.

Mais recentemente foi proposto [Brin, S. et al. (1997)] o método DIC (*Dynamic Itemset Counting*). O objectivo fundamental desta técnica – generalização do *Apriori* - é reduzir

o número de passagens pela base dados, melhorando, assim, o desempenho da descoberta de associações. O procedimento adoptado por este algoritmo é fazer paragens periódicas em cada M transacções da base de dados. Em cada paragem é adicionado um conjunto frequente de maior dimensão que será contado nas passagens ulteriores. Quando um conjunto é contado em toda a base de dados e não tem suporte mínimo, pode ser eliminado do conjunto frequente.

Outros trabalhos na área dos algoritmos para descoberta de regras de associação são: Mannila, H. et al. (1996), Srikant, R. et al. (1997), Zaki, M. J. et al. (1997)a, Zaki, M. J. et al. (1997)b, Han, J. et al. (1999), Han, J. et al. (2000), Lin, W. et al. (2000), Wang, K. et al. (2000)a. Em Hipp, J. et al. (2000) é apresentado um *survey* sobre algoritmos para a descoberta de regras de associação, onde os algoritmos estudados apresentaram um desempenho comparável.

Em Srikant, R. et al. (1996) os autores propuseram uma estratégia para lidar com atributos quantitativos. Esta passa por discretizar os atributos quantitativos, sendo que a dificuldade encontrada neste enquadramento é saber em quantos intervalos se devem dividir os dados. Esta dificuldade foi endereçada pela introdução do conceito de *K-Completeness*. Outro trabalho nesta área é Tsai, P. et al. (2001).

2.3 Selecção de Regras

O objectivo dos algoritmos apresentados anteriormente, é a descoberta, de forma cada vez mais eficiente, de todas as regras de associação que possam existir nos dados. Se, por um lado, a descoberta de todas as regras que possam existir nos dados pode ser considerado um ponto forte destes algoritmos - são completos - por outro lado, esta característica resulta num número extremamente elevado de regras geradas [Liu B. Et al. (1999)]. Através da aplicação de técnicas de *Data Mining*, como já foi exposto, consegue-se descobrir padrões não explícitos nos dados para, desta forma, a capacidade de interpretação dos mesmos ser aumentada. Acontece que se o utilizador for sobrecarregado com um número elevado de padrões, não vai ter capacidade suficiente para os analisar, logo não serão de muita utilidade para ele. Torna-se então evidente a necessidade de encontrar métodos para evitar a produção de muitas regras de

associação, ou para seleccionar as que se apresentam com mais interesse para os utilizadores.

O conjunto “suporte / confiança” é utilizado no modelo original do problema da descoberta de regras de associação. O suporte é necessário, dado que representa a significância estatística de um padrão. Do ponto de vista do marketing, por exemplo, o suporte de um conjunto de *itens* pode justificar a realização de uma promoção desses mesmos *itens*, porque não faz sentido gastar esforços numa campanha para um número reduzido de potenciais alvos. Os algoritmos para a descoberta de regras de associação são muito sensíveis a este parâmetro, pelo que o suporte pode ser considerado como um meio eficaz para reduzir o volume do output destes algoritmos (quanto mais elevado, menor o número de regras geradas). No entanto, o suporte por si só pode não ser muito fiável como medida de interesse para uma regra, já que suportes elevados podem resultar de padrões triviais, ou óbvios, nos dados [Tan, P-N. et al. (2000)]. Por exemplo, no contexto de retalho, um analista de negócio pode não considerar interessante a regra “pão \Rightarrow leite” (regra com um suporte elevado), visto esta não lhe revelar qualquer tipo de conhecimento que ele não possua *à priori*, isto é, do seu ponto de vista, o conhecimento extraído a partir desta regra é óbvio.

Como foi descrito atrás, a confiança de uma regra “A \Rightarrow B”, pode ser escrita em termos de probabilidade condicional: $\text{confiança}(A \Rightarrow B) = P(B|A) = P(A \cap B) / P(A)$. Esta expressão não considera $P(B)$. A probabilidade condicional $P(B|A)$ pode ser igual a $P(B)$, caso A e B sejam independentes. Neste caso, $P(B)$ pode ser superior à confiança mínima e, sendo assim, a regra “A \Rightarrow B” é válida. Por exemplo: suponha-se que $P(\text{leite}) = 80\%$; se “leite” não estiver correlacionado com “salmão”, então a regra “salmão \Rightarrow leite” tem confiança = 80%, sendo que este valor pode ser suficiente para que esta regra seja válida, apesar de os dois *itens* serem independentes! Este aspecto apresenta-se como sendo um ponto fraco da medida de confiança, tendo, por isso mesmo, sido contestada em Brin, et al. (1997). Sendo assim, o interesse de uma regra pode ser medido tendo em conta o desvio de $P(B|A)$ relativamente ao pressuposto de que A e B são independentes.

Várias medidas foram propostas neste enquadramento. O *interest* de uma regra é definido como:

$$interest(A \Rightarrow B) = P(A \cap B) / P(A)P(B)$$

Equação 2.1

O seu valor é o quociente entre a probabilidade conjunta observada e a probabilidade sob independência. Esta medida tem a particularidade de ser simétrica: $interest(A \Rightarrow B) = interest(B \Rightarrow A)$. A selecção de regras, segundo este critério, torna indiferente a selecção da regra $A \Rightarrow B$, ou da regra $B \Rightarrow A$, dado que estas possuem o mesmo valor para o *interest*. Partindo do pressuposto que uma implicação ($A \Rightarrow B$) pode ser escrita da seguinte forma: $\neg(A \wedge \neg B)$, chegamos à *conviction*:

$$conviction(A \Rightarrow B) = P(A)P(\neg B)/P(A \cap \neg B)$$

Equação 2.2

As propriedades da *conviction* são:

- $0 < conviction(A \Rightarrow B) < \infty$;
- A e B são estatisticamente independentes sse $conviction(A \Rightarrow B) = 1$ (regras pouco interessantes);
- $0 < conviction(A \Rightarrow B) < 1$ sse $P(B|A) < P(B)$ (isto é, B é correlacionado com A negativamente – regras, em geral, pouco interessantes);
- $1 < conviction(A \Rightarrow B) < \infty$ sse $P(B|A) > P(B)$ (isto é, B é correlacionado com A positivamente – regras interessantes).

Resumindo, a *conviction* é verdadeiramente uma medida de implicação, uma vez que é direcciona; é máxima quando a implicação é perfeita; e considera tanto a probabilidade do antecedente da implicação, quanto a do conseqüente.

Os quadros seguintes [Adamo, Jean-Marc (2001)] exemplificam estes conceitos:

	<i>Itens</i>						
	a	b	c	¬b	d	e	f
1	1	1	0	0	1	1	0
2	1	1	1	0	1	1	0
3	1	0	1	1	0	1	0
4	1	0	1	1	0	1	0
5	0	0	1	1	1	0	1
6	0	0	1	1	1	0	1
7	0	0	1	1	0	0	1
8	0	0	1	1	0	0	1

Tabela 2.1 – Exemplos de transacções de itens

Regra $x \Rightarrow y$	Suporte(x)	Suporte(y)	Suporte($x \Rightarrow y$)	Confiança($x \Rightarrow y$)	<i>Interest</i> ($x \Rightarrow y$)	<i>Conviction</i> ($x \Rightarrow y$)
$a \Rightarrow b$	0,50	0,25	0,25	0,50	$2 > 1$	1,50
$a \Rightarrow c$	0,50	0,875	0,375	0,75	$0,875 < 1$	0,50
$a \Rightarrow \neg b$	0,50	0,75	0,25	0,50	$0,666 < 1$	0,50
$a \Rightarrow d$	0,50	0,50	0,25	0,50	$1 = 1$	1
$a \Rightarrow e$	0,50	0,50	0,50	1	$2 > 1$	∞
$a \Rightarrow f$	0,50	0,50	0	0	$0 < 1$	0,50

Tabela 2.2 – Varias medidas para regras formadas a partir das transacções da Tabela 2.1

Segundo o modelo suporte/confiança, qualquer regra - excepto $a \Rightarrow f$ - tem potencial para ser válida, em virtude dos seus valores de suporte e confiança. No entanto, os *itens* “a” e “c”, e “a” e “¬b” estão correlacionados negativamente, sendo que, neste caso, as regras respectivas são pouco interessantes. De igual modo, os *itens* “a” e “d” são independentes, logo, a regra respectiva é também desinteressante. Através dos valores da *conviction*, verificamos que todos estes casos são correctamente endereçados, isto é, só a primeira e a quinta regras são válidas.

No seguimento deste raciocínio, *itens* independentes resultam em regras pouco interessantes, foi apresentado em Liu B. et al. (1999) um trabalho onde o teste do *qui-quadrado*:

$$\chi^2 = \sum \frac{(f - f_o)^2}{f}$$

é utilizado para testar se os *itens* estão correlacionados¹. Caso os *itens* sejam independentes (χ^2 baixo), a regra é rejeitada. Caso sejam correlacionados, importa saber se a correlação é positiva ou negativa. Se o suporte observado é superior ao suporte esperado, a correlação é positiva, logo a regra é aceite. Caso contrário, a regra é rejeitada. A tabela de contingência (2×2) seguinte [Liu B. et al. (1999)] ilustra um exemplo neste enquadramento, relativo a uma regra do tipo “ $A \Rightarrow B$ ” (frequências observadas):

	B	$\neg B$	
A	frequência de ($A \Rightarrow B$)	frequência de ($A \Rightarrow \neg B$)	frequência de (A)
$\neg A$	frequência de ($\neg A \Rightarrow B$)	frequência de ($\neg A \Rightarrow \neg B$)	frequência de ($\neg A$)
	frequência de (B)	frequência de ($\neg B$)	Nº total de casos – transacções

Tabela 2.3 - Tabela de contingência para a regra “ $A \Rightarrow B$ ” cujo objectivo é o teste da independência entre A e B

Os resultados experimentais levados a cabo por Liu B. et al. (1999), revelaram que tanto o número de regras geradas, quanto o número de condições das mesmas foi muito reduzido. O factor “capacidade de interpretação” deste modelo de *data mining* ficou, assim, largamente beneficiado.

A abordagem para esta questão em Bayardo R. et al. (1999) foi diferente. O que estes autores propuseram em primeiro lugar foi a definição de uma ordem parcial entre regras \leq_{sc} , tal que dadas duas regras r_1 e r_2 :

¹ “Entre duas variáveis ligadas por uma relação estatística diz-se que existe correlação. Indica-se, assim, que os fenómenos não estão indissoluvelmente ligados, mas, sim, que a intensidade de um é acompanhada tendencialmente pela intensidade do outro, no mesmo sentido ou em sentido inverso. “ [Murteira, B. et al. (2002)]

$r1 <_{sc} r2$ sse:

$$\text{suporte}(r1) \leq \text{suporte}(r2) \wedge \text{confiança}(r1) < \text{confiança}(r2)$$

∨

$$\text{suporte}(r1) < \text{suporte}(r2) \wedge \text{confiança}(r1) \leq \text{confiança}(r2)$$

$r1 = r2$ sse:

$$\text{suporte}(r1) = \text{suporte}(r2) \wedge \text{confiança}(r1) = \text{confiança}(r2)$$

Neste contexto, define-se um que conjunto de regras $R \subseteq U$ (sendo que U é o conjunto de todas as regras) é *SC-ótimo* sse:

$$\forall r1 \in R, \neg \exists r2 \in U : r1 <_{sc} r2$$

A figura seguinte [Bayardo R. et al. (1999)] mostra graficamente um conjunto de regras representadas num sistema de duas coordenadas, sendo que o eixo dos xx representa o suporte; e o eixo dos yy representa a confiança. As regras representadas por pequenos quadrados a negro, representam o conjunto de regras *SC-ótimo*, ou seja, as regras que “caem na fronteira”, para além da qual não existem mais regras.

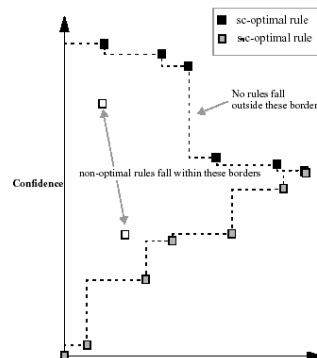


Figura 2.1 - Conjunto de regras representadas num sistema de duas coordenadas, sendo que o eixo dos xx representa o suporte; e o eixo dos yy representa a confiança

O que foi demonstrado em Bayardo R. et al. (1999) é que para muitas medidas de interesse – incluindo o suporte, confiança, *interest* e a *conviction* - as regras mais interessantes caem na fronteira SC, sendo que todas as outras são desprezáveis. Deste

modo consegue-se reduzir o número de regras produzidas e, assim, aumentar capacidade de interpretar os modelos gerados.

Após estudar e comparar, em Tan, P-N. et al. (2000), uma série de medidas de interesse para regras de associação, os autores evidenciaram a importância destas medidas considerarem a correlação estatística entre *itens*; e reforçaram a ideia de que o suporte, significância estatística de um padrão, é relevante para eliminar logo à partida uma série de padrões que envolvam *itens* não correlacionados, ou correlacionados negativamente.

Recapitulando, verificou-se que através de uma série de medidas de interesse para regras de associação: consegue-se controlar o número de regras geradas - evitando que regras pouco interessantes sejam geradas; ou, consegue-se fazer *pruning* das regras geradas menos interessantes. Com efeito, estas aspectos é relevante, uma vez que os algoritmos para a descoberta de regras de associação possuem a característica de serem completos, isto é, percorrem exaustivamente o espaço de procura, e, como tal, produzem um número elevado de regras. Potencialmente, estes algoritmos podem descobrir todas as regras que existem nos dados.

2.4 Pós Processamento e Exploração de Regras de Associação

Tipicamente, o resultado das aplicações que implementam estes algoritmos é uma lista de regras em formato texto. Consegue-se aumentar a capacidade de interpretar estas listas, ao seleccionar, a partir destas, apenas as regras que se apresentam com mais interesse para o utilizador, conforme já foi apresentado. No entanto, o facto das regras de associação serem apresentadas em formato texto não facilita a tarefa necessária de pós processamento (*rule mining*) para inspeccionar e explorar o conjunto de regras produzidas. [Jorge, A. et al. (2002)a]

É neste contexto que têm surgido sistemas para navegar nestas extensas listas de regras e para visualizar este tipo de modelos de *Data Mining*. Desta forma é possível encontrar subconjuntos de regras interessantes, e, conseqüentemente, aumenta-se a capacidade de

interpretar estes modelos [Ma et al. (2000)a], [Ma et al. (2000)b], [Wettshereck (2002)] e [Neves 2002].

O *Clustering* de Regras de Associação [Lent, B., et al. (1997)] permite representar visualmente, numa grelha de duas dimensões, *clusters* de regras, aos quais é aplicado um algoritmo que transforma estes *clusters* em regras mais gerais.

Em [Jorge, A. et al. (2002)a] e [Neves 2002] foi apresentado o *Pear – Post Processing Environment for Association Rules*. O *Pear* é um sistema para pós processamento de regras que permite que o utilizador seleccione e navegue no conjunto de regras geradas, e com capacidades de visualização. Este sistema está implementado num ambiente *web*, cuja versão cliente corre num *browser* internet, onde está implementado um conjunto de operadores responsáveis por transformar um conjunto de regras noutros conjuntos diferentes. A navegação é efectuada através de um conjunto de operadores bem definidos e com uma semântica clara e intuitiva. Este processo de selecção e exploração de regras de associação, ao contrário do que acontece na selecção de regras através de medidas objectivas de interesse como *interest* e *conviction*, permite que o utilizador, de forma implícita, use o seu conhecimento sobre o domínio em causa.

2.5 Resumo do Capítulo

Neste capítulo foi introduzido o conceito regra de associação, abordou-se a problemática associada à descoberta deste tipo de regras e foram referidas técnicas para seleccionar as regras com mais interesse. No final foi dada uma nota sobre o pós processamento e exploração de regras de associação.

No capítulo seguinte será abordada a aplicação de regras de associação em variados contextos.

3 Aplicação de Regras de Associação

Existem numerosas aplicações práticas de *data mining* que utilizam regras de associação. O exemplo de referência desta utilização surge em contextos de retalho, onde o objectivo é estudar as associações entre produtos que existem nas várias transacções efectuadas pelos clientes – conforme já foi referido. De facto, a determinação dos produtos que um cliente provavelmente comprará em conjunto, poderá ser muito útil para as tarefas de planeamento e de marketing. No entanto, a utilidade das regras de associação estende-se à análise de dados com outras características, tais como: as inscrições de alunos por disciplina, a ocorrência de determinadas palavras em ficheiros de texto, as visitas de utilizadores a páginas web, entre outros [Brin, S. et al. (1997)].

Os investigadores desta área, têm desenvolvido alguns trabalhos onde as regras de associação têm assumido um papel fundamental na resolução de diversos tipos de problemas, designadamente problemas de *clustering* [Lent, B., et al. (1997)] e de classificação [Liu B. et al. (1998)]. Outra aplicação de regras de associação com interesse é a sua utilização como base de certo tipo de sistemas de recomendação [Goldberg, D. et al. (1992)], [Resnick, P., et al. (1997)] e [Jorge, A. et al. (2002)b].

3.1 *Clustering* de Regras de Associação

O *clustering* de regras de associação foi referido no capítulo anterior, como sendo uma via para a visualização gráfica, a duas dimensões, deste tipo de modelos. Esta técnica de descoberta de regras de associação (*ARCS - Association Rule Clustering System*), proposta em Lent, B. et al. (1997), é útil quando se pretende segmentar os dados. O objectivo da sua aplicação é formar um conjunto reduzido regras de associação - mais gerais - a partir de um conjunto maior, constituído por regras mais específicas. Considere-se o exemplo extraído de Lent, B. et al. (1997), representado pela figura seguinte:

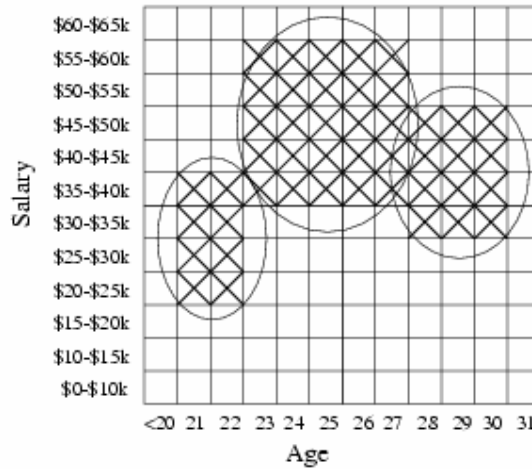


Figura 3.1 - Grelha bidimensional para representar regras de associação com consequentes iguais

Todas as regras representadas nesta grelha - assinaladas com uma cruz - têm consequentes iguais. Exemplos: *Salário = \$20K-\$25K \wedge Idade = 21 \Rightarrow Consequente*; *Salário = \$35K-\$40K \wedge Idade = 25 \Rightarrow Consequente*. Nesta figura consegue-se identificar três *clusters* de regras, devidamente assinaladas por circunferências. O objectivo da arquitectura proposta em Lent, B. et al. (1997) é substituir as regras pertencentes ao mesmo *cluster* por uma única regra mais geral. Por exemplo, o *cluster* mais à esquerda, formado por oito regras, é representado pela regra: *(\$20K < Salário \leq \$40K) \wedge (21 \leq Idade \leq 22) \Rightarrow Consequente*. No exemplo da figura, todas as regras podiam ser substituídas apenas por três – uma por cada *cluster* encontrado.

Para além de permitir visualizar modelos de regras de associação, os resultados experimentais apresentados por Lent, B. et al. (1997), mostraram que o ARCS possui um desempenho em geral equivalente ao do C4.5 [Quinlan, J. R. (1993)], sendo que, no entanto, o volume de regras geradas pelo primeiro é muito inferior. Sendo assim, os modelos criados pelo ARCS são mais fáceis de interpretar.

3.2 Regras de Associação para Classificação

A descoberta de regras de classificação tem como objectivo encontrar um conjunto reduzido de regras na base de dados que forme um classificador preciso [Quinlan, J. R. (1993)]. A descoberta de regras de associação (como já foi apresentado) tem como objectivo encontrar todas as regras na base de dados que satisfaçam as restrições: suporte mínimo e confiança mínima. Para a descoberta de regras de associação o alvo das regras não está predeterminado, enquanto que para a descoberta de regras de classificação existe um, e apenas um, alvo predeterminado: a classe a prever.

Em Liu B. et al. (1998) foi proposta a integração destas duas técnicas de *Data Mining* com o objectivo de construir um classificador com elevada precisão. Esta integração foi conseguida focando a investigação na descoberta de um subconjunto especial de regras de associação, referido como: *Classification Association Rules (CARs)*. Foi então proposto um novo algoritmo para a construção de um classificador baseado no conjunto de *CARs* descobertas previamente.

Neste enquadramento, as *CARs* caracterizam-se por serem regras de associação, cujo consequente (lado direito da regra) é restrito ao atributo classe do classificador. Para este efeito foi efectuada a adaptação de um algoritmo para a descoberta de regras de associação (neste caso foi o *Apriori*), de modo a produzir como resultado todas as *CARs* que satisfaçam as restrições: suporte mínimo e confiança mínima.

O primeiro passo a dar no sentido de se efectuar a “classificação associativa” é gerar o conjunto completo de todas as *CARs* que satisfazem as restrições impostas pelo utilizador: suporte mínimo e confiança mínima. Esta geração é obtida através da execução do algoritmo *CBA-RG (Classification Based on Association – Rule Generator)*.

O segundo passo é a construção de um classificador com base nas *CARs* geradas. O algoritmo proposto para a construção do classificador é o: *CBA-CB (Classification Based on Association – Classifier Builder)*. Antes de prosseguir com descrição da

construção do classificador, importa definir uma ordem total entre as regras geradas. Esta definição será utilizada na selecção de regras para o classificador.

Definição: Dadas duas regras r_i e r_j , $r_i \succ r_j$ (r_i precede r_j , ou r_i tem uma precedência mais elevada do que r_j) se:

1. a confiança de r_i é superior à confiança de r_j , ou
2. as suas confianças são iguais, mas o suporte de r_i é superior ao suporte de r_j , ou
3. as confianças e os suportes de r_i e r_j são iguais, mas r_i é gerada antes de r_j ;

Seja R o conjunto das regras geradas e D os dados de treino. A ideia base do algoritmo *CBA-CB* é seleccionar um conjunto de regras de elevada precedência em R para cobrir D . O classificador proposto tem o seguinte formato:

$$\langle r_1, r_2, \dots, r_n, \text{default_class} \rangle,$$

Expressão 3.1

onde $r_i \in R$, $r_a \succ r_b$ se $b > a$. *default_classe* é a classe por defeito. Para classificar um novo caso, a primeira regra que satisfizer o caso irá classificá-lo. Se não existir nenhuma regra que seja aplicável ao caso, será atribuída a classe por defeito, tal como no *C4.5* [Quinlan, J. R. (1993)].

Foram feitas várias experiências [Liu B. et al. (1998)], no sentido de verificar o desempenho deste classificador em relação ao desempenho do *C4.5* (*release 8*). As experiências mostraram que, desde que o suporte mínimo seja definido em 1% ou 2%, o classificador produzido é mais preciso do que o *C4.5*.

Liu B. et al. (1998) permitiu provar a aplicabilidade das técnicas para a descoberta de regras de associação, em tarefas de classificação. No seguimento deste trabalho, muitos outros têm igualmente surgido [Wang, K. et al. (2000)b] [Li, W. et al. (2001)]. Por

exemplo, em Jovanoski, V. et al. (2001), foi apresentada uma nova versão do *Apriori*, o *Apriori-C*, propositadamente modificado para resolver problemas de classificação.

Na próxima secção será possível verificar que este modelo – *CBA* - tem semelhanças com certos tipos de sistemas de recomendação, designadamente com os baseados em regras de associação.

3.3 Sistemas de Recomendação¹

É frequente termos que tomar certas decisões, sem que, contudo, tenhamos a experiência pessoal suficiente sobre as várias alternativas possíveis. Nas mais diversas actividades do dia a dia, apoiamo-nos em recomendações feitas por outras pessoas; em “*recommendation letters*”; em artigos de revistas; ou em *surveys*. [Resnick, P. et al. (1997)]. Neste contexto, o objectivo dos sistemas de recomendação, igualmente designados por *collaborative filtering* [Goldberg, D. et al. (1992)], é, de acordo com Pennock D. M., et al. (2000), prever as preferências de um utilizador, com base nas preferencias de um grupo de utilizadores. Para Sarwar, B. et al. (2001), os sistemas de recomendação são uma nova e poderosa tecnologia para extrair valor adicional para o negócio a partir da sua base de dados de utilizadores: os sistemas de recomendação beneficiam os utilizadores, dando-lhes a possibilidade de encontrar os *itens* de que gostam ou necessitam e, por outro lado, ajudam o negócio ao gerar mais vendas.

O volume de informação disponível tem crescido a uma velocidade bem mais elevada do que capacidade existente para a processar. Por exemplo: constantemente são lançados no mercado novos livros e são publicados novos artigos em jornais e conferências. A tecnologia tem reduzido de forma drástica as barreiras à publicação e distribuição de informação [Sarwar, B. et al. (2001)]. Consequentemente, a recepção de informação não desejada ou irrelevante, geralmente referida por *information overload* [Wei, Y. Z. et al. (2003)], é considerada um problema para muitas pessoas. Por este motivo é que está a ser investido um esforço significativo em investigação, no sentido

¹ “Recommender Systems”

de serem construídos instrumentos de suporte que garantam que a informação certa é entregue às pessoas certas no tempo certo.

Os motores de busca, por exemplo os referidos em Breese, J. S. et al. (1998) e Wei, Y. Z. et al. (2003), podem auxiliar este processo. Tipicamente são baseados em *queries* que identificam características intrínsecas dos *itens* pesquisados. A procura de documentos de texto (por exemplo páginas *web*) utiliza *queries* que contêm palavras ou que descrevem conceitos que são desejados nos documentos devolvidos. A pesquisa de títulos de *CDs*, por exemplo, necessita da identificação do artista desejado, do género, ou do período de tempo. No entanto, os motores de busca não são personalizados para um utilizador individual e têm a tendência para não entregar o volume de informação apropriado.

É com o objectivo de ultrapassar as limitações dos motores de busca que, segundo Wei, Y. Z. et al. (2003), os sistemas de recomendação têm sido defendidos. Estes sistemas não utilizam qualquer informação sobre o conteúdo (por exemplo: palavras, autor, descrição) dos *itens*, mas, de outro modo, são baseados nos padrões de preferências de outros utilizadores. Estes sistemas seguem o pressuposto de que um bom processo de encontrar conteúdos interessantes, é encontrar outras pessoas com interesses similares, para recomendar conteúdos que essas pessoas apreciem [Breese, J. S. et al. (1998)]. Pennock D. M., et al. (2000) reforça esta ideia ao afirmar que a eficácia de qualquer algoritmo de *collaborative filtering* está fundamentada no pressuposto de que as preferências humanas estão correlacionadas – se não estivessem, então não seria possível efectuar este tipo de previsões.

Num sistema de recomendação típico, as recomendações são disponibilizadas pelos utilizadores como *input*; este é então agregado e direccionado para os receptores apropriados [Resnick, P., et al. (1997)]. Estes sistemas produzem um *score* da probabilidade prevista, e / ou uma lista das *N-melhores* recomendações de *itens*, para um determinado utilizador [Sarwar, B. et al. (2001)].

Os sistemas de recomendação estão a tornar-se rapidamente numa ferramenta crucial para o comércio electrónico através da *web*. Neste enquadramento, os desafios com que estes sistemas se deparam são a sua escalabilidade (os sistemas modernos têm a necessidade de procurar, em tempo real, dezenas de milhões de utilizadores com preferências equivalentes – utilizadores “vizinhos”); e a qualidade das recomendações para o utilizador (os utilizadores necessitam de recomendações em que possam confiar para encontrar o que necessitam ou gostam) [Sarwar, B. et al. (2001)].

Diversas técnicas têm sido utilizadas para efectuar as recomendações [Wei, Y. Z. et al. (2003)]. Breese, J. S. et al. (1998) identifica duas grandes categorias de algoritmos para efectuar a predição das preferências dos utilizadores:

- Os algoritmos *Memory-based* operam em toda a base de dados dos utilizadores para efectuar as recomendações. Estes algoritmos empregam técnicas estatísticas para encontrar um conjunto de utilizadores, conhecidos por utilizadores “vizinhos”, que possuam um histórico de preferências semelhante com o do utilizador actual. Após encontrar os utilizadores “vizinhos”, estes algoritmos combinam as suas preferências para produzir as *N-melhores* recomendações.
- Contrariamente, os algoritmos do tipo *Model-based*, utilizam a base de dados dos utilizadores para construir modelos, sendo que estes modelos são depois utilizados para efectuar as recomendações. O processo de construção destes modelos é executado por algoritmos de *machine learning* [Mitchell, Tom M. (1997)], como por exemplo: *redes bayesianas*, *clustering* e *modelos de regras*, em particular, modelos de regras de associação tal como foi apresentado em Sarwar, B. et al. (2000) e Jorge, A. et al. (2002)b.

Ainda neste trabalho [Breese, J. S. et al. (1998)], foi feita uma comparação do desempenho de vários algoritmos. Os resultados experimentais, evidenciaram que, para uma grande maioria das condições, as redes bayesianas com árvores de decisão nas folhas (*model-based*), e os métodos correlativos (*memory-based*) superaram o desempenho (precisão) do *clustering bayesiano (model-based)* e da similaridade vectorial (*memory based*). Em Pennock D. M., et al. (2000) foi apresentada uma

abordagem híbrida entre estas duas categorias de algoritmos. O desempenho do algoritmo desenvolvido segundo esta nova abordagem foi comparado com outros quatro algoritmos – dois *memory-based*, e dois *model-based* – sendo que os resultados mostraram que a abordagem híbrida apresenta melhores resultados.

De acordo com Sarwar, B. et al. (2001), o maior problema relacionado com o desempenho dos sistemas de recomendação convencionais é a procura de “vizinhos” numa grande população de utilizadores. Com efeito, os sistemas de recomendação que utilizam algoritmos de pesquisa de utilizadores “vizinhos” (*user-based*) deparam-se com uma série de desafios, tais como:

- A grande dispersão da informação – na prática, os sistemas de recomendação comerciais são utilizados para lidar com conjuntos de dados de elevada dimensão (por exemplo, a *Amazon.com* – recomenda livros – e *CDnow.com* recomenda CDs). Nestes sistemas, mesmo os utilizadores mais activos compraram bem menos do que 1% do total de *itens* disponíveis (1% de 2 milhões de livros são 20.000 livros). Consequentemente, um sistema de recomendação baseado em algoritmos que procuram o “vizinho” mais próximo podem não ser capazes de gerar qualquer recomendação para um utilizador em particular. Como resultado a precisão das recomendações pode ser pobre.
- Escalabilidade – os algoritmos que procuram os “vizinhos” mais próximos requerem uma carga computacional que cresce com o número de utilizadores e com o número de *itens*. Com milhões de utilizadores e *itens*, um sistema de recomendação com estas características terá sérios problemas relacionados com escalabilidade.

Sendo assim, Sarwar, B. et al. (2001) sugerem uma abordagem diferente: explorar, em primeiro lugar, a relação que existe entre *itens*, em vez de explorar a relação que existe entre utilizadores – algoritmos *item-based*. As recomendações são efectuadas ao encontrar *itens* que são equivalentes a outros *itens* que o utilizador gostou ou já escolheu. Estes algoritmos são capazes de atingir resultados com a mesma qualidade dos algoritmos *user-based* com uma carga computacional inferior.

Wei, Y. Z. et al. (2003) defende que a melhor forma de produzir recomendações é permitir que diferentes métodos de produzir recomendações possam coexistir. Para atingir este objectivo, foi proposto um sistema para coordenar os *outputs* dos diversos métodos, de tal forma que apenas as melhores recomendações são apresentadas ao utilizador. Neste sistema, os vários métodos são representados por agentes competindo entre si num mercado aberto para conseguir apresentar as suas recomendações ao utilizador.

Por seu turno, Resnick, P., et al. (1997) enquadrou os sistemas de recomendação num espaço definido por 5 dimensões:

- Conteúdo das recomendações. Desde a mais básica informação, do tipo: “recomendar”, ou “não recomendar”; até às anotações textuais não estruturadas.
- Forma como as recomendações são recolhidas:
 - explicitamente pelos utilizadores do sistema – sistemas como GroupLens e EachMovie [Kleinberg, J. et al. (2003)] perguntam explicitamente ao utilizador para registar as suas preferências, as quais são então utilizadas para efectuar as recomendações.
 - implicitamente através de recolha automática da informação – mecanismo utilizado por diversos sítios de comércio electrónico, como por exemplo a *Amazon.com* [Kleinberg, J. et al. (2003)], onde o histórico das escolhas dos utilizadores é guardado e agregado para ser utilizado nas recomendações de *itens* equivalentes às escolhas efectuadas pelos clientes.
- A fonte que forneceu as recomendações pode ser anónima, ou pode ser identificada.
- Como a informação, as várias recomendações, é agregada: votos pesados, análise de conteúdo.
- Utilização das recomendações: as não recomendações podem ser descartadas; os *itens* podem ser ordenados de acordo com valorizações numéricas; ou, apresentação dos *itens* acompanhadas simultaneamente pelas valorizações numéricas.

3.4 Avaliação de Sistemas de Recomendação

Os investigadores desta área têm utilizado diversas métricas para avaliar a precisão dos sistemas de recomendação. Estas podem ser agrupadas em dois grupos principais [Sarwar, B. et al. (2001)]:

- As *métricas estatísticas* avaliam a precisão do sistema comparando os *scores* obtidos pelas recomendações, com as preferências efectivas dos utilizadores (disponibilizadas por estes, também na forma de um *score*). O *Erro Absoluto Médio (EAM)* entre as preferências dos utilizadores e as predições efectuadas pelo sistema é uma métrica frequentemente utilizada. O *EAM* mede o desvio das recomendações em relação ao seu verdadeiro valor – a preferência real de um utilizador específico. Para cada par preferência / predição $\langle p_i, q_i \rangle$ esta métrica calcula o erro absoluto entre estes, ou seja $|p_i - q_i|$. O *EAM* é a média dos N pares preferência / predição. Formalmente:

$$EAM = \frac{\sum_{i=1}^N |p_i - q_i|}{N}$$

Quanto mais baixo for o *EAM*, mais precisas são as recomendações. A *Raíz do Erro Quadrático Médio (REQM)* é também utilizada como medida estatística para avaliar a precisão dos sistemas de recomendação.

- As *métricas de suporte à decisão* avaliam quão eficaz é um motor de predição na tarefa de ajudar um utilizador a encontrar os *itens* que pretende, no conjunto de todos os *itens*. As métricas deste tipo mais utilizadas são *reversal rate*, *weighted errors* e *ROC sensitivity*.

O *EAM* foi igualmente referido e utilizado em Breese, J. S. et al. (1998), quando as recomendações têm o formato de um *score* individual. Quando as recomendações têm o formato de uma lista das *N-melhores* recomendações, foram empregues, neste trabalho, métricas amplamente utilizadas pela comunidade de investigadores de “extracção de

informação”, designadamente o *recall*, a *precision* e o *F1*. Estas foram igualmente utilizadas em Sarwar, B. et al. (2000) e Jorge, A. et al. (2002)b, para os mesmos efeitos.

Em van Rijsbergen, C. A. (1979) estas métricas foram estudadas e o seu significado, no contexto “sistemas de recomendação”, é o seguinte:

- o *recall* corresponde à proporção de respostas correctas e é uma estimativa da probabilidade de se obter pelo menos uma recomendação relevante.
- a *precision* corresponde à qualidade de cada recomendação individual – estimativa da probabilidade de cada recomendação individual estar correcta.
- o *F1* combina o *recall* e a *precision* com pesos iguais. O seu intervalo de valores possíveis varia entre 0 e 1, sendo que maiores valores indicam melhores recomendações. É útil para sintetizar as outras duas medidas e pode ser utilizado para encontrar a sua melhor combinação. Formalmente o *F1* é representado da seguinte forma:

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

Equação 3.1

A representação formal do *recall* e da *precision* será apresentada mais à frente, após se expor os vários métodos estudados para conseguir avaliar a precisão dos sistemas de recomendação.

Para obter o valor destas métricas, a metodologia seguida em Sarwar, B. et al. (2000), Sarwar, B. et al. (2001), Breese, J. S. et al. (1998) e Jorge, A. et al. (2002)b foi começar por dividir os dados disponíveis em dois grandes conjuntos: conjunto de treino e conjunto de teste. Esta divisão foi efectuada considerando 80% dos dados para treino e 20% dos dados para teste. Em Sarwar, B. et al. (2001) é igualmente feita uma referência à validação cruzada (10 folhas). Em Witten, Ian H. et al. (2000) podem ser estudados com mais detalhe estes dois métodos de validação de modelos de *data mining* – o *hold out* e o *cross validation* respectivamente – entre outros. Neste contexto, o conjunto de

treino foi utilizado para construir o modelo de recomendação, quer para os casos *memory based* (neste caso o modelo é o próprio conjunto de treino, utilizado para encontrar os “vizinhos” mais próximos), quer para os casos *model based*. O conjunto de teste, por sua vez, foi manipulado no sentido de ser “escondida” uma porção das escolhas de cada utilizador – o que pretendemos prever. Chamou-se a este conjunto *hidden*. Ao conjunto de teste sem esta porção chamou-se conjunto observável. O caso particular em que a porção “escondida” corresponde apenas a uma única escolha efectuada por cada utilizador, foi designado por *All but One Protocol* em Breese, J. S. et al. (1998). Para obter as métricas para avaliar a precisão dos sistemas em estudo são geradas as recomendações tendo como base o conjunto de dados de treino e, como *input* para cada utilizador, as escolhas presentes no conjunto observável. Ao serem comparadas as recomendações assim efectuadas, com as escolhas que foram “escondidas” (o que se pretendia prever) obtêm-se os valores para as várias métricas.

A figura seguinte ilustra de forma simplificada todos estes passos:

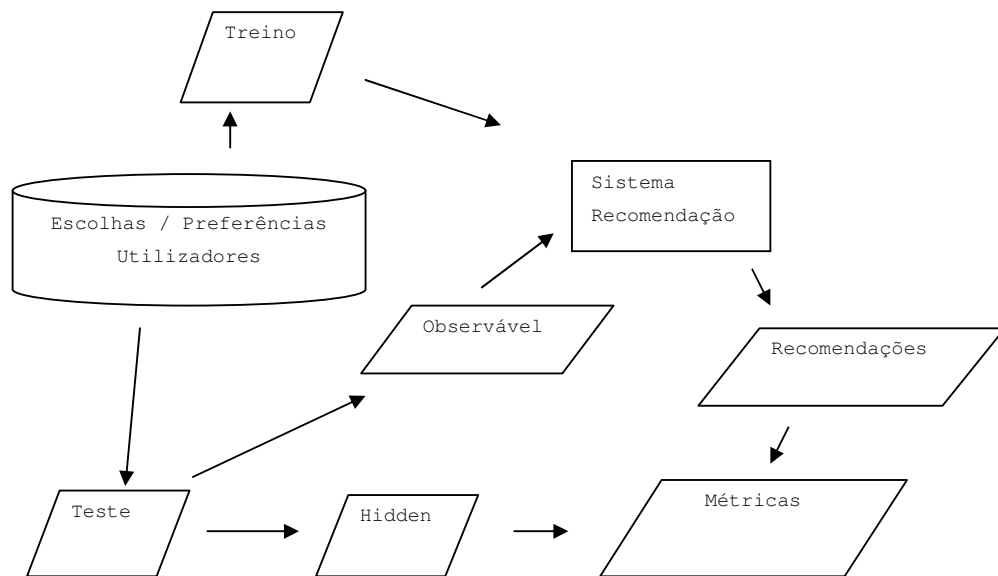


Figura 3.2 - Síntese dos passos necessários para avaliar um sistema de recomendação

Assim, o *recall* é definido formalmente da seguinte forma:

$$Recall = \frac{|Hidden \cap Recommendations|}{|Hidden|}$$

Equação 3.2

cujo significado é o quociente entre o número de respostas correctas devolvidas pelo sistema (numerador da *Equação 3.2*) e o número total de respostas correctas (denominador da *Equação 3.2*). Esta medida é aplicada quando as recomendações têm o formato de uma lista das *N-melhores* recomendações, logo o *recall* tem a tendência de crescer com o valor de *N*. A *precision* é definida formalmente da seguinte forma:

$$Precision = \frac{|Hidden \cap Recommendations|}{|Recommendations|}$$

Equação 3.3

ou seja, é o quociente entre o número de respostas correctas devolvidas pelo sistema (numerador da *Equação 3.3*) e o número de recomendações devolvidas pelo sistema (denominador da *Equação 3.3*). Quando o *N* aumenta o número de recomendações aumenta (denominador da *Equação 3.3*), logo a *precision* tem a tendência para diminuir. Os numeradores da *Equação 3.2* e da *Equação 3.3* são iguais. Estas expressões diferem apenas no denominador. Isto significa que se o *N* for igual a 1, e se o sistema tiver condições para responder em todos os casos (o que nem sempre é possível devido, por exemplo, à dispersão dos dados, como já foi referido), o *recall* é igual à *precision*.

3.5 Sistemas de Recomendação e Regras de Associação

Os sistemas de recomendação têm sido aplicados em muitos domínios [Wei, Y. Z. et al. (2003)], tais como o comércio de livros (*Amazon.com*) e CDs (*CDnow.com*) [Sarwar, B. et al. (2001)]; pesquisa de artigos e *netnews* (*GroupLens*) [Resnick, P., et al. (1997)]; entre outros. Em Jorge, A. et al. (2002)b foram utilizadas regras de associação para

construir um modelo de recomendação (*Model-based*), cujo objectivo era melhorar a utilização de um sítio *web*. Com este sistema, cada utilizador pode receber recomendações, enquanto navega pelo sítio, em função das páginas que foram acedidas nessa sessão de acesso *web*. Estas recomendações são listas de ligações (*links*) com interesse para o utilizador. Para atingir este objectivo, a ideia é construir o modelo de recomendação começando por gerar regras de associação a partir dos dados presentes no registo (*log*) de acessos do sítio *web*. De seguida as páginas visitadas pelos utilizadores são comparadas com os antecedentes destas regras. Os consequentes das N regras que satisfazem o critério de comparação, com a confiança mais elevada, tornam-se assim nas recomendações.

Formalmente: se o modelo de recomendação, constituído pelo conjunto das regras de associação, for representado por M ; o conjunto observável das escolhas efectuadas pelo utilizador actual por O ; e o conjunto das recomendações resultantes por R , define-se este sistema por:

$$R = \{consequente(r_i) \mid r_i \in M \wedge antecedente(r_i) \subseteq O \wedge consequente(r_i) \notin O\}$$

Expressão 3.2

Ou seja, o conjunto de recomendações R é formado pela reunião dos consequentes das regras que pertencem ao modelo M , e que, simultaneamente, possuem antecedentes que estão contidos no conjunto observável, sendo que estes consequentes não pertencem ao conjunto observável.

Se o objectivo for obter as N -melhores recomendações, são seleccionadas a partir de R as recomendações correspondentes às N regras com a confiança mais elevada. A construção de um algoritmo para a implementação da *Expressão 3.2* deve considerar que entre estas N regras podem existir consequentes repetidos e, como tal, as medidas apresentadas podem evidenciar algumas distorções. Por exemplo, um determinado conjunto observável O , devolve as seguintes regras:

Antecedente	Consequente	Confiança
Ant1	A	0,9
Ant2	B	0,85
Ant3	A	0,7
Ant4	C	0,5
Ant5	D	0,4

Se, considerando consequentes repetidos, o N for igual a 3, as recomendações respectivas são: {A, B} – correspondem aos consequentes das 3 primeiras regras. Ao eliminarmos deste conjunto de regras as que possuem consequentes repetidos, as recomendações associadas ao mesmo valor de N (3) são: {A, B, C}. Considerando as fórmulas do *recall* e da *precision*, é expectável que a remoção das regras com o mesmo consequente tenha impacto nos seus valores – os numeradores destas duas fórmulas tenderão a subir; o mesmo acontecerá ao denominador da *precision*.

Quando comparados com a escolha aleatória de sítios, os resultados experimentais apresentados em Jorge, A. et al. (2002)b mostraram vantagens na utilização do sistema proposto.

Com este trabalho [Jorge, A. et al. (2002)b] é possível constatar que, de facto, existem semelhanças entre este modelo de recomendação baseado em regras de associação e o modelo *CBA*, apresentado na secção anterior, para efectuar tarefas de classificação. As diferenças fundamentais residem no facto do *CBA* ser baseado num conjunto de regras cujo consequente é restrito à classe a prever, enquanto que no modelo de recomendação essa restrição não existe; e reside no facto do critério proposto no *CBA* para seleccionar a regra que irá determinar qual a classe a prever (*Expressão 3.1*), ser ligeiramente diferente do critério formalizado através da *Expressão 3.2* para efectuar as recomendações.

Nesta tese, foi desenvolvido um sistema de recomendação aplicado ao domínio do planeamento de recursos humanos em equipas de projectos, como será descrito no próximo capítulo.

4 Recomendação de Recursos Humanos para Equipas de Projectos

Nas empresas prestadoras de serviços, quando estes estão organizados e estruturados por projectos, a planificação e estruturação de equipas é uma actividade fundamental. Para estas empresas, as suas vendas são materializadas através de projectos com determinada duração e determinado objectivo; cada projecto tem associado um conjunto de recursos (humanos) da empresa para executar as tarefas necessárias; cada recurso pode estar associado a mais do que um projecto. Esta actividade de planeamento é complexa dado que lida com diversas variáveis, tais como: as características técnicas e pessoais dos recursos humanos da empresa, bem como a sua disponibilidade; as características do projecto; as características do cliente; entre outras. Sendo assim, existem alguns desafios genéricos para quem tem a responsabilidade de efectuar este tipo de planeamento, entre os quais se destacam os seguintes:

- Onde é que se pode encontrar, explícita ou implicitamente, a informação necessária para a tarefa de planeamento de recursos?
- De que forma é que esta informação pode estar organizada para facilitar o seu acesso?
- Sendo as empresas organizações dinâmicas e em crescimento, será possível concentrar esta informação em colaboradores chave? E se estes colaboradores abandonarem a empresa?
- Onde é que se pode pedir uma segunda opinião face às escolhas efectuadas?
- É possível obter com facilidade um conselho ou uma recomendação para efectuar uma escolha deste tipo?

A dimensão das empresas, o número de recursos humanos afectos a projectos e o número e diversidade de projectos, acentuam proporcionalmente estes desafios.

Neste enquadramento, o contributo desta tese é aplicar parte dos conceitos teóricos abordados nos capítulos anteriores ao desenvolvimento de um sistema de recomendação

de recursos humanos para equipas de projectos, cujo objectivo é acrescentar valor ao processo mencionado em cima.

O caso real de negócio em estudo é uma empresa de serviços cuja designação comercial é *Enabler – Solutions for Retailing*. Os dados reais provenientes deste caso permitirão validar se o sistema de recomendação desenvolvido consegue dar resposta aos desafios apresentados anteriormente.

A concepção e resolução deste caso prático foi levada a cabo utilizando a metodologia *CRISP-DM* [Chapman, Pete et al. (2000)], cuja apresentação será efectuada no ponto seguinte.

4.1 Metodologia

A execução de projectos de *Data Mining*, é uma actividade relativamente recente – a primeira conferência do *ACM Special Interest Group on Knowledge Discovery in Data and Data Mining*¹ ocorreu em 1995. As entidades pioneiras nesta área foram seguindo cada uma o seu próprio caminho, definindo as suas próprias estratégias e definindo os seus próprios métodos. É, então, legítimo que estas entidades coloquem questões tais como: O que estamos a fazer está correcto? Será que quando novas entidades decidirem entrar neste meio, terão que aprender, como nós o fizemos inicialmente, através do método tentativa / erro? E do ponto de vista de um fornecedor de serviços, como é que se pode mostrar aos clientes que o *Data Mining* está suficiente maduro para ser adoptado como factor chave nos seus processos de negócio?

Foi com o objectivo de endereçar todas estas questões, que surgiram várias propostas para a criação de metodologias standard para o desenvolvimento de projectos de *Data Mining*, entre as quais se encontra o *CRISP-DM* (*Cross-Industry Standard Process for Data Mining*). Esta metodologia nasceu no seio de um consórcio formado pela DaimlerChrysler², a SPSS³ e a NCR⁴.

¹ <http://www.acm.org/sigkdd/>

² <http://www.daimlerchrysler.de>

³ <http://www.spss.com>

⁴ <http://www.ncr.com>

Na realidade, o *CRISP-DM* é um modelo de processos para *Data Mining*, independente da área de negócio e da tecnologia a utilizar. Como é de fácil dedução, esta metodologia tem como objectivo fazer com que grandes projectos de *Data Mining* se tornem mais rápidos, mais baratos, mais fiáveis e mais fáceis de gerir. Contudo, até projectos de *Data Mining* de pequena envergadura podem beneficiar com a aplicação do *CRISP-DM*.

Esta metodologia é descrita em termos de um modelo hierárquico de processos, que consiste num conjunto de tarefas representadas por quatro níveis de abstracção (do mais geral para o mais específico): Fases, Tarefas Genéricas, Tarefas Especializadas e Instâncias de Processos, conforme está representado na figura seguinte [Chapman, Pete et al. (2000)]:

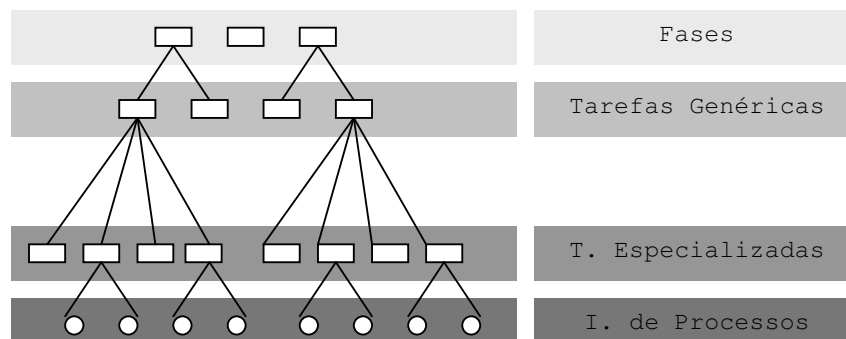


Figura 4.1 - Representação dos quatro níveis da metodologia CRISP-DM

As tarefas genéricas, são apresentadas de forma suficientemente geral para cobrir todas as situações de *Data Mining*. É suposto que estas tarefas sejam tão completas e estáveis quanto possível. Por completas entende-se que cubram todo o processo de *Data Mining* e todas as possíveis aplicações de *Data Mining*; e por estáveis entende-se que o modelo deve manter-se válido, mesmo para novos desenvolvimentos não previstos, conforme, por exemplo: novas técnicas de modelagem.

No terceiro nível (Tarefas Especializadas), descreve-se como as acções do nível genérico devem ser executadas em certas situações específicas.

Continuando a percorrer esta hierarquia, é no seu quarto nível (Instâncias de Processos) que se registam as acções, decisões e resultados de um determinado projecto de *Data Mining*. Uma instância de um projecto está organizada de acordo com as tarefas definidas nos níveis superiores, mas representa o que de facto aconteceu num determinado projecto, em vez do que acontece em geral.

O primeiro nível desta hierarquia, isto é, o ciclo de vida de um projecto de *Data Mining* segundo este modelo, é dividido em seis fases principais: Compreensão do Negócio (*Business Understanding*), Compreensão dos Dados (*Data Understanding*), Preparação dos Dados (*Data Preparation*), Modelação (*Modeling*), Avaliação (*Evaluation*) e Operacionalização (*Deployment*). Nesta metodologia, a sequência através da qual estas fases são executadas não é muito rígida.

A figura seguinte [Chapman, Pete et al. (2000)] ilustra graficamente este ciclo:

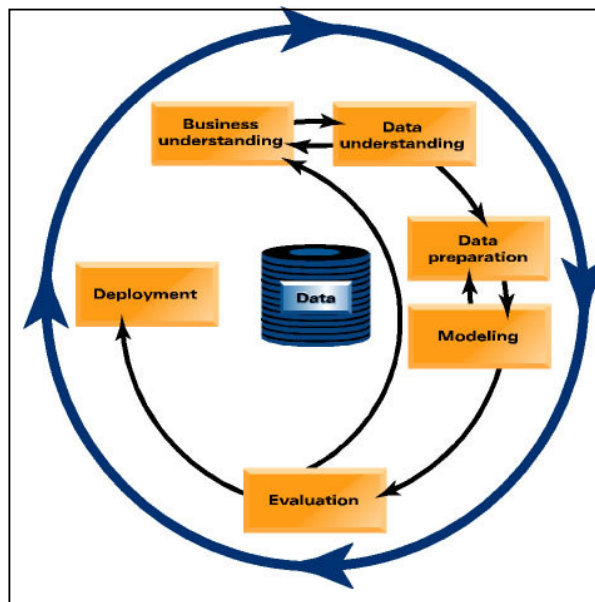


Figura 4.2 - Representação das fases do modelo de referência do CRISP-DM

Sucintamente, o objectivo de cada uma destas fases é:

- Compreensão do negócio – entender os objectivos do projecto e seus requisitos, sob o ponto de vista do negócio, para serem convertidos na definição de um problema de *Data Mining*. Definir um plano preliminar para atingir estes objectivos.
- Compreensão dos Dados – recolha dos dados seguida por actividades que visem a familiarização com estes, de maneira a identificar problemas relacionados com a qualidade dos dados, para descobrir perspectivas iniciais dentro dos dados, ou para detectar subconjuntos interessantes que permitam formular hipótese sobre informação (escondida) não disponível.
- Preparação dos Dados – construção dos *data sets* finais (que serão utilizados na construção dos modelos) a partir dos dados originais.
- Modelação – selecção e aplicação de várias técnicas de modelação de dados; optimização de parâmetros. Tipicamente, existem diversas técnicas para o mesmo problema de *Data Mining*. Dado que algumas destas técnicas possuem requisitos específicos em relação à forma dos dados, é frequente ter-se que voltar à fase anterior.
- Avaliação – do ponto de vista da análise dos dados, os modelos construídos nesta fase podem parecer ter elevada qualidade. Antes de prosseguir para a fase de operacionalização, é importante rever os passos dados para a construção do modelo, no sentido de garantir que ele cumpre os objectivos do negócio. Um objectivo chave é determinar se existe alguma questão importante de negócio que não foi considerada. É após esta fase que se decide implementar os resultados do processo de *Data Mining*, ou não.
- Operacionalização – utilização dos modelos criados em ambiente operacional.

As próximas secções deste capítulo estão estruturadas da seguinte forma: inicialmente será apresentado o problema, ao que se seguirá a análise dos dados em estudo. As tarefas de análise de dados deste documento, bem como todos os programas computacionais que tiveram que ser desenvolvidos, foram efectuados utilizando o R⁵. O R é uma linguagem [Venables, W. N. et al. (2000)] e um ambiente para computação estatística e gráficos [Venables, W. N. et al. (1999)]. Pelas suas características, o R pode

⁵ Disponível gratuitamente em www.r-project.org

também ser aplicado em tarefas de *Data Mining* [Torgo, L. (2002)]. A preparação dos dados, a escolha e o desenvolvimento do modelo serão apresentados depois, ao que se seguirá, no capítulo 6 *Avaliação*, a avaliação do modelo gerado. Este ciclo será fechado com uma proposta de utilização da solução final, no capítulo 7 *Operacionalização*.

4.2 Compreensão do Negócio

4.2.1 Caracterização da *Enabler*⁶

A *Enabler*, fundada em 1997, faz parte do grupo empresarial *Sonae* e o seu ramo de actividade prende-se com a prestação de serviços na área de sistemas de informação para empresas do sector do retalho. Os mercados onde actua são o Português, o Brasileiro e o Europeu (extra Portugal). A sua presença local traduz-se nos seus escritórios de Portugal (Porto e Braga), Inglaterra, Alemanha e Brasil - onde está situado o seu centro de desenvolvimento de *software*.

Sendo uma empresa especializada na área de retalho, a sua proposta de valor caracteriza-se por oferecer a esta indústria uma solução completa, constituída por uma combinação entre tecnologia e experiência no sector:

- serviços de consultoria para redefinição da arquitectura dos sistemas de retalho, em função de novos processos de negócio;
- capacidades de gestão de projectos através de procedimentos e metodologias bem formalizadas;
- gestão de programas (“super projectos”) – quando são necessárias alterações radicais de toda a estrutura de tecnologia de informação de uma empresa, a *Enabler* gere a complexidade do programa e coordena a intervenção dos diferentes fornecedores;
- desenvolvimento e implementação de *software*;
- capacidades de integração de sistemas e de pacotes de *software*;
- suporte aplicacional no final da implementação dos projectos.

⁶ Dados retirados do relatório de actividades 2002, disponível em www.enabler.com

Desde 1997 que a *Enabler* tem subido o seu volume de vendas de forma significativa. Em 2002 este volume atingiu o valor de 24,1 milhões de Euros, o que representa uma subida de 15% relativamente a 2001. 95% do volume de vendas têm origem nos *serviços prestados*, representando os *projectos* 84% (20,2 milhões de Euros), e *suporte* 11% (2,6 milhões de Euros).

Em Portugal é constituída por cerca de 220 colaboradores. Estes colaboradores estão enquadrados na actividade da empresa de acordo com a sua organização, que, basicamente, pode ser resumida em cinco *unidades*:

- *Comissão Executiva* – administração da empresa.
- *Unidade Comercial* – vendas e gestão de conta.
- *Unidade de Delivery*⁷ – execução e gestão de projectos.
- *Unidade de Serviços Profissionais* – suporte técnico e funcional ao cliente
- *Unidade de Suporte ao Negócio* – recursos humanos, infra-estrutura, qualidade, desenvolvimento organizativo.

Tal como já foi referido, são os *projectos* que representam a maior fatia do seu volume de vendas. A *unidade* com uma ligação mais directa aos *projectos* é a *unidade de delivery*. Esta *unidade* tem, por sua vez, uma organização interna que está representada na figura seguinte:

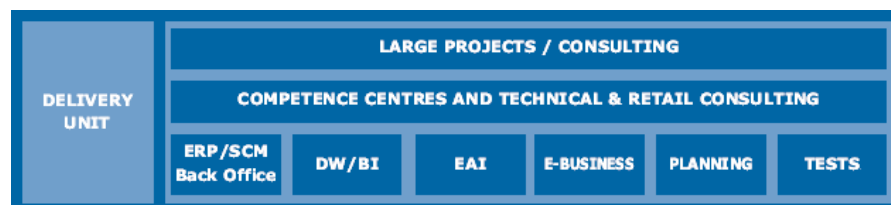


Figura 4.3 - Organização interna da unidade de delivery da Enabler

⁷ Dada a vocação multinacional da empresa, o idioma corporativo adoptado é o inglês, pelo que serão apresentados na descrição da actividade da empresa muitos termos nesta língua.

Os elementos que compõem esta unidade são:

- *centros de competência – pools* que agrupam os colaboradores por valências técnicas específicas. Estas têm associado um responsável – *resource manager* – cuja função, entre outras, é gerir a alocação dos colaboradores respectivos nos vários *projectos* da empresa.
- *grandes projectos / consultoria – pool* formada por colaboradores que desempenham funções de gestão de topo dentro da hierarquia de um *projecto*.

O percurso de carreira dos colaboradores pertencentes aos *centros de competência*, tem três níveis: 1 – *Analistas Juniores*; 2 – *Analistas Seniores*; 3 – *Gestores de Projecto*.

A *pool* dos *grandes projectos / consultoria* é formada apenas por colaboradores de nível 4 – *Delivery Managers*.

Os directores (*principals*) são de nível 5 e os elementos da administração são de nível 6.

Com o objectivo de monitorizar a actividade dos seus colaboradores, e assim facilitar as suas tarefas de planeamento e controlo de gestão, a *Enabler* utiliza uma ferramenta para registo electrónico de relatórios de actividade – *time reports*. Esta ferramenta é da *Evolve*⁸ e o seu nome é *Service Sphere*. Diariamente os colaboradores da *Enabler* cumprem o procedimento de preencher o formulário disponibilizado por esta ferramenta para registarem as horas despendidas com determinada tarefa. Estas tarefas podem estar associadas a determinados *projectos*, ou podem ser respeitantes a períodos de férias do colaborador, a formação, *suporte*, ou qualquer outro tipo de actividade não relacionada com *projectos*.

Os tempos registados pelos colaboradores em *projectos* devem ser aprovados pelo respectivo *gestor de projecto*, de acordo com o procedimento definido e adoptado pela *Enabler*. O *Service Sphere* tem implemento todo o processo de *workflow* que permite,

⁸ www.evolve.com

não só o registo dos *time reports*, mas também a sua submissão para aprovação posterior. O fluxograma seguinte descreve este processo:

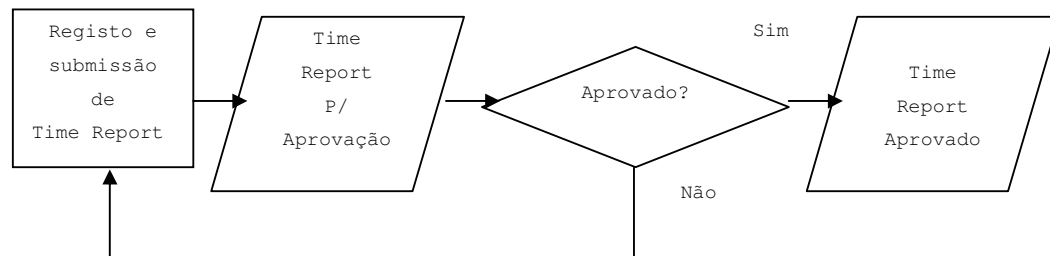


Figura 4.4 - Descrição gráfica do processo de registo e aprovação de time reports

Como nota complementar importa referir que a *Enabler* trabalha também com colaboradores subcontratados a outras empresas, sendo que estes estão também sujeitos ao procedimento de registo de horas na ferramenta mencionada. A figura seguinte mostra um exemplo de um *time report*:

The screenshot shows the 'Time Report Entry' interface in a Microsoft Internet Explorer browser. The page title is 'servicesphere.' and the user is logged in as '<Nome Colaborador>'. The 'Time Report Period' is set from 30/06/2003 to 06/07/2003, and the status is 'Approved'. The last submitted date is 04/07/2003.

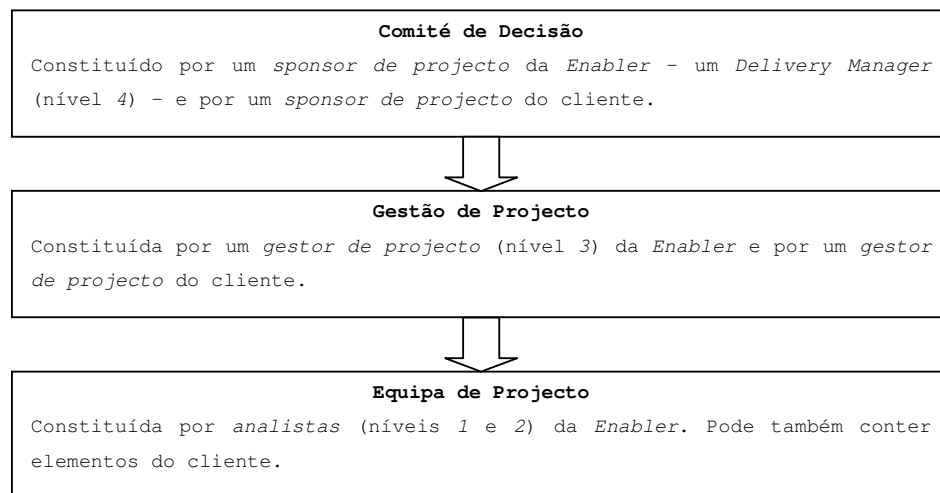
Type	Approver	Client Project/Org.Unit	Activity Task	Position/[Role] Location	Nonbillable Billable Override	Seg 30 Jun	Ter 1 Jul	Qua 2 Jul	Qui 3 Jul	Sex 4 Jul	Sáb 5 Jul	Dom 6 Jul	Total
Salaried		Enabler PT Geral - 244	Formação	[Project Manager] Portugal	No No		08.00		01.50				09.50
> Copy Row													
Salaried		<Cliente> <Projecto>	DEV-Gestão do Projecto	GP Portugal	No No			01.00					01.00
> Copy Row													
Salaried		<Cliente> <Projecto>	DEV-Gestão do Projecto	GP Portugal	No No	08.00		07.00	08.00	08.00			31.00
> Copy Row													
Total						08.00	08.00	08.00	09.50	08.00	00.00	00.00	41.50

Figura 4.5 - Exemplo de um time report no Service Sphere

Nesta é possível observar o aspecto do interface com o utilizador – aplicação *web* – bem como o nome do colaborador (1), o nome de quem aprova o registo (2), o *projecto* (3), a tarefa (4) e as horas registadas (5):

Os *projectos* são estruturados através de *propostas comerciais* que são entregues aos clientes após estes solicitarem a colaboração da *Enabler* em determinado contexto. É através deste documento – *proposta comercial* - que se define o *âmbito* de cada *projecto*. Este *âmbito* é estruturado através dos cinco seguintes eixos:

- *Funcional*: o que se pretende fazer - desenvolvimento de *software*, integração de aplicação, testes; e, para cobrir que necessidade funcional – processo de entreposto, processo de loja, cálculo de receitas comerciais, necessidade analítica, planeamento de gama.
- *Entrega*: que produtos resultantes vão ser entregues – documentos, relatórios, programas.
- *Temporal*: plano, tipicamente um gráfico de *gant*, que enquadrará as actividades e entregas do *projecto*, no tempo.
- *Organizacional*: localizar fisicamente as entregas e definição da estrutura interna do *projecto* – hierarquia usualmente definida da seguinte forma:



- *Financeiro*: valor comercial (monetário) do projecto, condições e modo de pagamento.

Antes de um *projecto* ter início, o cliente tem que aceitar formalmente o *âmbito* proposto pela *Enabler* na sua *proposta comercial*.

Normalmente, quem elabora as *propostas comerciais* são os *gestores de projecto*. Ao fazê-lo, estes têm que planear as necessidades de recursos humanos do *projecto*, em função do seu *âmbito*, ao que se segue a respectiva requisição para o *gestor de centro de competência - resource manager* - correspondente. Este responderá de acordo com a disponibilidade de recursos humanos do momento, face ao calendário de *projecto* apresentado.

Estas requisições podem ser de colaboradores específicos, ou de colaboradores não especificados, caracterizados apenas de forma abstracta através de determinadas valências técnicas pretendidas.

Quando um *gestor de projecto* pede um colaborador específico, fá-lo baseando-se nas suas experiências passadas, ou fá-lo baseado-se no conhecimento comum sobre os vários colaboradores, que está, de certa forma, disperso pela *Enabler*. Nem sempre é possível satisfazer as suas pretensões, dado que esses colaboradores podem não estar disponíveis no momento certo – indisponíveis porque podem estar alocados a outros *projectos*, em férias, etc.

Todo este processo de constituição de equipas para *projectos* (*Figura 4.6*) é também suportado pelo *Service Sphere*. Isto é, esta ferramenta permite que o *gestor de projecto* constitua e visualize a equipa do seu *projecto*; e permite ao *resource manager* ter uma visão global acerca dos planos de alocação dos colaboradores do seu *centro de competência*, e, conseqüentemente, permite-lhe ter presente quem está disponível e quando.

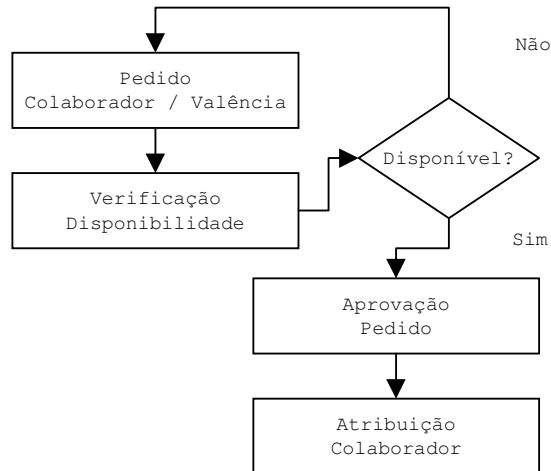


Figura 4.6 - Descrição gráfica do processo de constituição de equipas para projectos

Em síntese, o *Service Sphere* é um *ERP – Enterprise Resource Planing*⁹, especialmente concebido para empresas de serviços com as características da *Enabler*, que cobre, de forma modular, grande parte das suas necessidades operacionais, entre as quais foram apresentadas anteriormente: o processo de registo e aprovação de *time reports*; e o processo de constituição de equipas (requisição / aprovação de colaboradores) em *projectos*.

A *Figura 4.7* mostra um exemplo de uma equipa de *projecto* registada nesta ferramenta. Importa referir que, uma vez que se está a abordar um módulo diferente, mas da mesma aplicação, o aspecto do interface deste é muito equivalente ao do módulo de registo e aprovação de *time reports* que foi apresentado anteriormente. Nesta figura é possível identificar o *projecto* (1) e os vários *recursos humanos* (2) que colaboraram, colaboram ou que irão colaborar no *projecto* durante toda a sua vida útil.

⁹ Ver capítulo: *1 Introdução*

Microsoft Internet Explorer - <Projecto>

Address: http://prt01sph01/servlet/evolve.pts.web.SSS?POID=0.25000.6060%280.0-1393829*0%21_344%29&task=projectPositions

servicesphere

<Nome Colaborador>
Refresh Logout Help

Home Personal Clients Opportunities Projects Resources Messages Requests Forecasts Time and Expense Options

>>> New Team >>> New Team from Template >>> Team Builder

<Projecto> (1)

<Projecto>		> Edit > New position > Delete	
> Request Commitment > Advanced Scheduling > Delete		> Staff	
GP			
Details	Qualifications Required	Commitments	
Dates	Project Planning, Tracking and Reporting (Advanced);	Open	No
Duration	Project Risk Management (Beginner);	<Nome Colaborador>	Assigned 01/05/2003 - 31/12/2003
Not Billable	Scope Management & Change Control (Advanced);	(2)	
Title	Teambuilding (Advanced);		
Role	Change Management (Advanced)		
Use Role Rate			
Sponsor			
Details	Qualifications Required	Commitments	
Dates	Project Planning, Tracking and Reporting (Expert);	Open	No
Duration	Teambuilding (Expert);	<Nome Colaborador>	Assigned 01/05/2003 - 31/12/2003
Not Billable	Scope Management & Change Control (Expert);	(2)	
Title	Project Risk Management (Expert);		
Role	Change Management (Expert)		
Use Role Rate			
Analista 1			
Details	Qualifications Required	Commitments	
Dates	ProC (Advanced);	Open	No
Duration	SQL (Advanced)	<Nome Colaborador>	Assigned 01/05/2003 - 31/12/2003
Not Billable		(2)	
Title			
Role			
Use Role Rate			
Analista 2			
Details	Qualifications Required	Commitments	
Dates	ProC (Advanced);	Open	No
Duration	SQL (Advanced)	<Nome Colaborador>	Assigned 01/05/2003 - 31/12/2003
Not Billable		(2)	
Title			
Role			
Use Role Rate			

Figura 4.7 - Exemplo de uma equipa de projecto registada no Service Sphere

4.2.2 Definição do Problema

Conforme se verificou aquando da caracterização da *Enabler*, são os *projectos* que representam a sua actividade mais significativa e lucrativa. Há um conjunto de processos administrativos que são indispensáveis para a estruturação e monitorização destes *projectos*, tal como, por exemplo: o *processo de elaboração de propostas comerciais*.

Uma das tarefas contempladas na elaboração deste tipo de documentos, é o *planeamento do projecto*, sendo que este planeamento refere-se tanto ao enquadramento das entregas no tempo (quando?), quanto à constituição da própria equipa do projecto (quem?). Em relação a este último aspecto foi referido que os critérios seguidos para estas escolhas (quem?) são a sensibilidade do *gestor do projecto* (o seu próprio “conhecimento”) derivada das suas experiências anteriores (com quem trabalhou anteriormente para atingir determinados objectivos; e com que grau de sucesso esses objectivos foram atingidos); ou o “conhecimento” comum sobre os vários colaboradores, que está, de certa forma, disperso pela *Enabler*. É importante referir que a concretização destas escolhas está condicionada à disponibilidade dos colaboradores pretendidos.

Sendo a actividade de planeamento e constituição de equipas em *projectos* uma actividade complexa, é então legítimo colocar as seguintes questões (algumas destas são instâncias particulares dos desafios genéricos colocados no início deste capítulo):

- Onde é que um *gestor de projecto* recém chegado à *Enabler*, portanto sem experiências anteriores, pode encontrar o “conhecimento” referido?
- Será que este “conhecimento” pode estar “escondido” e/ou armazenado em qualquer parte da *Enabler*?
- O registo histórico dos *time reports* produzidos ao longo do tempo armazena a actividade real dos vários colaboradores da *Enabler*. As “escolhas” feitas no passado pelos colaboradores que efectuaram a actividade de planeamento, estão

assim guardadas e reflectidas neste histórico. Será que este histórico pode conter o “conhecimento” que está a ser referido?

- Será que este “conhecimento” terá que ser concentrado em determinados colaboradores chave? Sendo a *Enabler* uma empresa em franco crescimento, será que isto pode ser comportável no futuro? E se esses elementos abandonarem a empresa?
- Onde é que um colaborador da *Enabler* pode pedir uma segunda opinião face às escolhas efectuadas?
- Um colaborador da *Enabler* pode obter com facilidade um conselho ou uma recomendação para efectuar uma escolha deste tipo?

O conjunto de questões anterior pode ainda ser incrementado com a seguinte:

- De que forma é que, neste contexto, um modelo de *data mining* poderia acrescentar valor?

No capítulo anterior foram abordados alguns exemplos de aplicação prática de regras de associação. Em particular foram apresentados casos concretos de sistemas de recomendação baseados em modelos de regras de associação [Sarwar, B. et al. (2000)] e [Jorge, A. et al. (2002)b]. Sucintamente, o objectivo destes sistemas é auxiliar o utilizador a encontrar os *itens* de que necessita. Com base nestes fundamentos teóricos, será apresentada de seguida uma proposta de um sistema de recomendação, baseado em regras de associação, que visa dar resposta às questões apresentadas em cima. Ou seja, este sistema tem como objectivo recomendar recursos humanos para equipas de projectos e, como tal, pretende auxiliar o referido processo de planeamento.

No capítulo 2 *Regras de Associação*, foi introduzido o conceito *regra de associação* e foi dado um exemplo clássico de aplicação prática desta técnica de *data mining*: análise de cestos de compras de um hipotético supermercado.

O problema que se pretende resolver está relacionado com o planeamento e constituição de equipas em projectos. A empresa em questão é uma empresa de prestação de serviços – a *Enabler* – e não um supermercado.

Se, por analogia com o exemplo clássico, considerarmos que um colaborador representa um artigo; e que um projecto num determinado dia representa um *cesto*, podemos reformular o exemplo anterior da seguinte forma:



Figura 4.8 - Actividade dos recursos em projectos, vista como um problema de análise de custos

No entanto, é possível pensar em diferentes hipóteses para agregar estes dados em cestos com granularidades distintas: um projecto pode ser, por si só, considerado um cesto; um projecto numa semana pode ser considerado um cesto; ou, um projecto num mês pode ser considerado um cesto. As experiências levadas a cabo nesta tese foram efectuadas considerando um cesto representado por um projecto num determinado dia, conforme exemplo da *Figura 4.8*. A elaboração de experiências com cestos de granularidade diferente, bem como o estudo e análise do seu impacto na qualidade dos resultados finais será alvo de trabalho futuro.

A partir destes *cestos* (*Figura 4.8*) podem ser geradas regras de associação do tipo $A \Rightarrow B$, com suporte s e confiança c respectivos, cujo significado é: se A trabalhar num determinado projecto / dia (*cesto*), então B tem $c\%$ de probabilidade de trabalhar nesse mesmo projecto / dia (*cesto*). As associações entre colaboradores que provêm destas regras, podem e devem ser estudadas e interpretadas para serem aplicadas a situações práticas.

A informação necessária para a construção destes *cestos* encontra-se no registo histórico dos *time reports* produzidos ao longo do tempo pelos vários colaboradores da *Enabler*.

É de esperar que este histórico contenha, de forma latente, “conhecimento” que deverá ser descoberto pelo modelo de regras gerado, tal como, por exemplo: os critérios que conduziram às escolhas feitas no passado, sendo que estes critérios são condicionados pelas valências técnicas de cada colaborador, a sua disponibilidade, o objectivo a atingir com o projecto e os sucessos atingidos no passado em circunstâncias semelhantes. É também aceitável que os colaboradores, ao trabalharem juntos nas mesmas equipas por longos períodos de tempo, criem determinados laços pessoais e profissionais que contribuam positivamente para o sucesso de projectos futuros, caso voltem a fazer parte das mesmas equipas. O modelo de regras deve portanto encontrar estas associações entre os colaboradores.

Para auxiliar a tarefa de planeamento e constituição de equipas, a proposta desta tese é então construir um protótipo de um motor para um sistema de recomendação *model based*, baseado em regras de associação, tal como foi apresentado no sub capítulo 3.5 *Sistemas de Recomendação e Regras de Associação*. Este sistema deverá receber como *input* um conjunto de colaboradores, ao que deverá responder com uma recomendação de colaboradores.

Sintetizando os passos necessários, o objectivo é então construir um modelo apropriado de regras de associação, a partir dos dados históricos dos *time reports*. Para este efeito utilizar-se-á uma implementação do *Apriori*. Este modelo servirá de base para um protótipo de um motor para um sistema de recomendação de colaboradores a construir ulteriormente. A *Figura 4.9* ilustra este processo.

Este sistema deverá responder às questões apresentadas no início desta secção. Para saber se este objectivo foi atingido é fundamental definir conceitos de avaliação apropriados. Do ponto de vista experimental, a qualidade deste sistema pode ser medida através do *Recall* e da *Precision*, tal como foi exposto no capítulo anterior. Na prática, a

percepção dos colaboradores da *Enabler* face à adequação das recomendações efectuadas por este sistema, pode ser estudada e interpretada. Este estudo e esta interpretação foram efectuados através da realização de um inquérito via questionário escrito junto de uma amostra de colaboradores da *Enabler*. Mais adiante serão apresentados os resultados obtidos por estes métodos de avaliação.

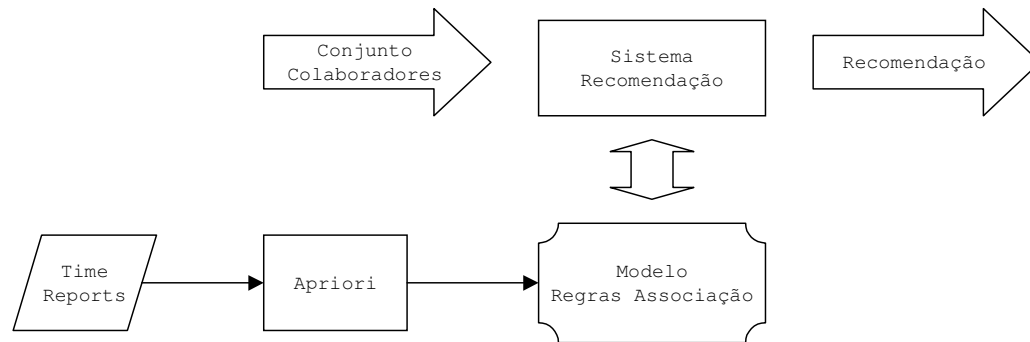


Figura 4.9 - Arquitectura do sistema de recomendação de recursos proposto

4.3 Compreensão dos Dados

4.3.1 Análises Preliminares

Conforme já foi referenciado, os dados disponíveis para este estudo têm origem na aplicação informática para registo de *time reports* da *Enabler* – o *Service Sphere*. Mais especificamente, estes dados referem-se ao histórico desta aplicação compreendido entre os meses de (inclusive) Setembro 2001 e de Novembro 2002, o que faz, ao todo, um total de 15 meses. Os dados em bruto foram carregados para uma tabela, numa base de dados *mysql*¹⁰, cujo nome é “*dados*”.

A estrutura desta tabela é:

```
desc dados;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Pool_Name | varchar(50) | YES | | NULL | |
| Sigla | varchar(5) | YES | | NULL | |
| Resource_Name | varchar(50) | YES | | NULL | |
| Project_Code | varchar(25) | YES | | NULL | |
| Project_Name | varchar(50) | YES | | NULL | |
| Client_Name | varchar(50) | YES | | NULL | |
| From_Date | date | YES | | NULL | |
| To_Date | date | YES | | NULL | |
| Activity_Name | varchar(50) | YES | | NULL | |
| Cell_Date | date | YES | | NULL | |
| Value_Minutes | int(11) | YES | | NULL | |
| Billable | varchar(50) | YES | | NULL | |
| Approved_Date | date | YES | | NULL | |
| Subject_Note | varchar(50) | YES | | NULL | |
| Note | varchar(50) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
15 rows in set
```

Tabela 4.1 Estrutura da tabela com os dados relativos à actividade real dos recursos da *Enabler*

Antes de prosseguir é importante evidenciar que, sendo o *Service Sphere* uma aplicação em língua inglesa, todos os atributos desta tabela estão escritos neste idioma. Por este motivo é que daqui em diante os *colaboradores* serão designados por *recursos*, uma vez que é a tradução mais directa de *Resource_Name* – nomenclatura utilizada pela aplicação.

¹⁰ Disponível gratuitamente em www.mysql.com

A partir desta tabela, foi criada uma outra, “*dados_trunc*”, retirando para este fim todos os registos cujo código de projecto é igual a “”. Isto é, foram retirados todos os registos relativos a actividades não referentes a projectos (actividades identificadas por *Enabler Geral*), tais como por exemplo: *Férias*, *Gestão Corrente*, *Sem Alocação*, *Formação*, etc. Por não serem actividades sujeitas a planeamento (e por isso fora do âmbito deste estudo), foram igualmente retirados os registos relacionados com as actividades de suporte da *Enabler*.

Informação relevante relacionada com estas tabelas:

	Nº de Registos	Nº de Recursos	Nº de Projectos	Nº de Cestos	Nº de Dias	Minutos Totais
Tabela “dados”	169.104	302	886	36.834	484	34.151.385
Tabela “dados_trunc”	80.244	290	812	26.234	480	20.702.375
Diferença	88.860	12	74	10.600	4	13.449.010

Tabela 4.2 Informação relacionada com as tabelas mencionadas

A diferença do número de registos entre as duas tabelas está distribuída, conforme está representado na ***Figura 4.10***.

O facto do número de recursos ser diferente nas duas tabelas, evidencia que, no período em análise, houve 12 recursos que não participaram em projectos.

Dos 74 projectos a mais na tabela “*dados*” em relação à tabela “*dados_trunc*”, 72 referem-se a projectos de suporte; 1 é *null*; o projecto que sobra, tem o código igual a “”, ou seja, são as actividades “*Enabler Geral*”.

Uma vez que a descoberta das regras de associação não necessita de toda a informação presente em “*dados_trunc*”, foi criada a tabela “*cestos*”, cuja estrutura está representada na ***Tabela 4.3***.

	Nº Registos	Percentagem
Enabler Geral	: 46.990	(27,79%)
Suporte	: 41.799	(24,72%)
Registos Nulos	: 71	(0,04%)
Sub Total (1)	: 88.860	(52.55%)
Dados a Analisar (2)	: 80.244	(47,45%)
Total (1) + (2)	: 169.104	(100%)

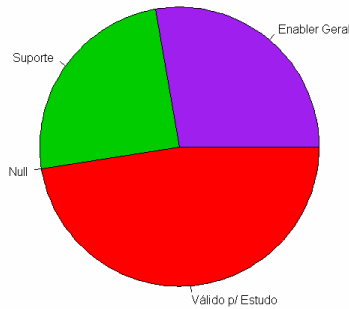


Figura 4.10 Diferença entre o número de registos totais e válidos para estudo

```
desc cestos;
```

Field	Type	Null	Key	Default	Extra
project_code	varchar(25)	YES		NULL	
cell_date	date	YES		NULL	
resource_name	varchar(50)	YES		NULL	
value_minutes	int(11)	YES		NULL	

4 rows in set

Tabela 4.3 Estrutura da tabela utilizada na geração das regras de associação

Tal como foi possível verificar no sub capítulo 2.2 *Descoberta de Regras de Associação*, os algoritmos tradicionais para a descoberta deste tipo de regras, como o *Apriori* por exemplo, efectuem contagens de *itens*, num universo específico de transacções, para gerar as regras. No entanto, a “quantidade” de *itens* em cada transacção não é considerada. Por exemplo, se uma transacção - correspondente a um cesto de um hipotético supermercado - contiver 2 unidades de pão e 20 unidades de leite, estes algoritmos consideram que esta transacção é equivalente à transacção: uma

unidade de pão e uma unidade de leite. É de esperar que este peso – a “quantidade” de *itens* – tenha influência na descoberta de associações através deste método. Por este motivo, optou-se por manter o atributo *value_minutes* na tabela *cestos*, para futuramente ser possível construir regras de associação que considerem a presença do recurso (*item*) nos *cestos* e também a respectiva “quantidade” do mesmo. Esta acção será trabalho futuro, pelo que não será efectuada no âmbito desta tese. Este atributo ajudará também a caracterizar melhor os *recursos* e os *projectos* da *Enabler*.

Nesta tabela (*cestos*), a informação de todas as actividades, de um recurso, numa data, num projecto, foi agregada num único registo. Por esse motivo, o seu número de registos difere do número de registos de “*dados_trunc*”, ou seja é de: 70.212 (80.244 – 70.212 = 10.032).

4.3.2 Análise Exploratória

Com o objectivo de ter uma perspectiva inicial dos dados, foram criadas, a partir da tabela “cestos”, várias variáveis. Estas variáveis caracterizam fundamentalmente os recursos, os projectos, os cestos e os dias. Para estas variáveis foram calculadas uma série de medidas de localização (*Mínimo*, *1º Quartil*, *Mediana*, *Média*, *3º Quartil* e *Máximo*) e de dispersão (*Desvio Padrão*). O objectivo destas medidas é chamar a atenção para aspectos e padrões de maior interesse nos dados [Guimarães, Rui Campos et al. (1997)], [Murteira, Bento J. F (1993)] e [Murteira, Bento J. F et al. (2002)]. Estas variáveis e estas medidas estão representadas no quadro seguinte:

	Min	1º Q	Mediana	Média	Intervalo de confiança para a média, a 95%		3º Q	Max	Desvio Padrão
					Limite Inferior	Limite Superior			
Dias por Recurso	1	83,25	192,50	170,20	159,15	181,32	257	316	95,92
Cestos por Recurso	1	93,50	245,50	242,10	221,35	262,87	334	1.199	179,59
Projectos por Recurso	1	4,25	13	15,47	13,88	17,07	21	82	13,79
Minutos por Recurso	120	24.050	77.780	71.390	66.121,84	76.653,16	111.500	163.600	45.559,73
Dias por Projecto	1	7	16	32,31	29,43	35,18	38	309	41,73
Cestos por Projecto	1	7	16	32,31	29,43	35,18	38	309	41,73
Recursos por Projecto	1	2	4	5,53	4,95	6,10	6	134	8,35
Minutos por Projecto	-960	1.200	4.059	25.500	20.798,19	30.192,89	17.390	820.000	68.192,13
Recursos por Cesto	1	1	2	2,68	2,64	2,71	3	64	2,89
Minutos por Cesto	-480	180	480	789,10	776,12	802,17	960	11.910	1.076,15
Recursos por Dia	1	9	145	102,90	95,90	109,80	168	226	77,53
Cestos por Dia	1	7	61	54,65	50,94	58,37	92	128	41,40
Projectos por Dia	1	7	61	54,65	50,94	58,37	92	128	41,40
Minutos por Dia	180	2.940	59.020	43.130	40.103,53	46.156,37	71.080	133.600	33.744,50

Tabela 4.4 Variáveis para caracterizar os dados em estudo dos dados em estudo

Estes valores indicam uma grande dispersão em todas as variáveis. Como seria de esperar, os testes à normalidade efectuados a cada uma destas variáveis (mais adiante verificar-se-á a importância destes testes para a tomada de decisões em relação à validade da aplicação de determinadas técnicas estatísticas), mostraram que nenhuma delas tem uma aproximação estatisticamente significativa em relação a esta distribuição. Ao interpretar estas medidas, deve-se ter em consideração que as médias e os desvios padrões são muito sensíveis à presença de *outliers*.

Os valores negativos, assinalados por elipses, representam que existiram estornos nestes registos de tempos. Por exemplo: para acertar um *time report*, por qualquer motivo, um recurso pode registar tempos negativos de forma a anular tempos introduzidos anteriormente. O período em análise contemplou um *projecto* cujos registos de tempos negativos foi superior aos registos de tempos positivos; e um *cesto* em circunstâncias idênticas.

De realçar nestes dados, o número médio de cestos por recurso (242,10), o que faz antever suportes baixos neste problema ($242,10 / 26234 = 0,92\%$). Este facto pode ser explicado pelo seguinte: as unidades de cada recurso disponíveis são escassas (um recurso tem, teoricamente, 8 horas por dia de disponibilidade) contrariamente a um *item* de um problema clássico de descoberta de regras de associação (podem existir milhares de unidades em *stock* de um artigo num supermercado). Isto faz com que um recurso não possa aparecer em muitos cestos num dia (é razoável que apareça pelo menos num!). Inversamente, um artigo de supermercado pode, de facto, aparecer em imensos cestos num dia.

O número médio de projectos por recurso (15,47); o número médio de dias de um projecto (32,31); e o número médio de recursos por projecto (5,53) são igualmente indicadores interessantes para caracterizar a *Enabler* e os seus tipos de projectos.

Dado que a chave de cada cesto (transacção) é a data e o projecto, os dias por projecto e os cestos por projecto representam exactamente a mesma coisa.

De acordo com Hipp, J. et al. (2000), um problema clássico de descoberta de regras de associação tem, tipicamente, cestos com 10 a 20 *itens* em média, num total de 1.000 a 100.000 *itens*. Ora, os dados que estamos a estudar, evidenciam que estamos perante um problema com dimensão completamente diferente, ou seja, o número médio de recursos (*itens*) por cesto é de 2,68; e o número total de recursos é 290.

4.3.3 Análise dos Cestos e dos Conjuntos de Recursos

Dado que se pretende construir um modelo baseado em regras de associação, é importante aprofundar o conhecimento sobre os cestos e sobre os conjuntos de recursos – os *item sets* deste problema.

Como já foi referido, existem 26.234 cestos e 290 recursos nos dados em estudo. O número de cestos por recurso está distribuído de acordo com os seguintes gráficos (*box plot* e histograma):

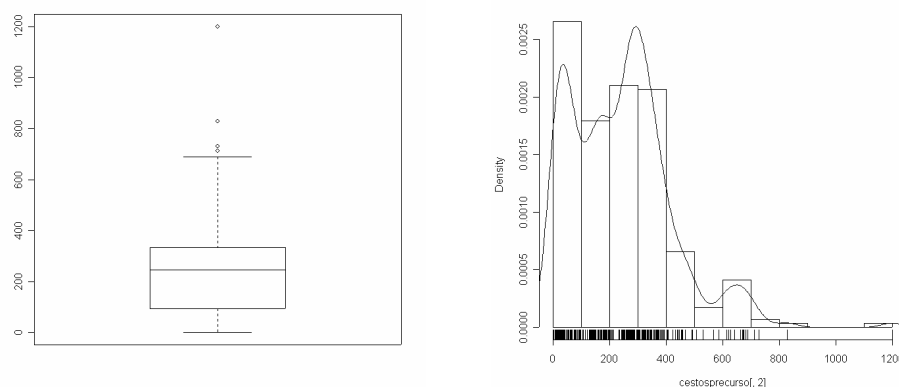


Figura 4.11 Distribuição do número de recursos por cesto

Uma vez que o número de cestos por recurso é o *support count* dos conjuntos de recursos de tamanho 1 (*item sets* de tamanho 1), estes gráficos representam também a distribuição destes valores.

Os valores mínimo (1) e máximo (1.199) da variável cestos por recurso que foram apresentados anteriormente, conduzem-nos aos limites inferiores e superiores do suporte dos conjuntos de recursos de tamanho 1. Estes valores são, respectivamente: $1 / 26.234 = 0,00\%$; e $1.199 / 26.234 = 4,57\%$. À medida que se formam conjuntos de recursos a partir dos conjuntos de dimensão 1, estes novos conjuntos vão-se tornando cada vez mais específicos. Logo, os seus suportes vão ser sempre menores ou iguais aos suportes dos conjuntos que lhes deram origem. Por este motivo, o valor apresentado

para o limite superior dos conjuntos de tamanho 1 (4,57%), vai ser o valor máximo do suporte para os dados deste problema, o que reforça a ideia apresentada anteriormente (aquando da apresentação do suporte para o valor médio de recursos por cesto - $242,10 / 26234 = 0,92\%$) de que este problema apresenta valores baixos para o suporte.

O espaço de procura de conjuntos de *itens* tem 2^{290} conjuntos. O valor do suporte mínimo (parâmetro do *Apriori*) tem impacto neste valor – número de conjuntos –, da seguinte forma:

	Min	1° Q	Mediana	3° Q	Máx
Cestos por Recurso	1	93,5 (25%)	245,5 (50%)	334 (75%)	1.199 (100%)
Número de Recursos		73	145	217	290
Suporte Mínimo	$\geq 0\%$	$\geq 0,36\%$	$\geq 0,94\%$	$\geq 1,27\%$	$\geq 4,57\%$
Número Máx. Conjuntos	$\leq 2^{290}$	$\leq 2^{217}$	$\leq 2^{145}$	$\leq 2^{73}$	0

Tabela 4.5 Efeito do suporte mínimo no espaço de procura

Algumas notas sobre o quadro anterior:

- Há 290 recursos em estudo.
- Cada quartil (1° Q, Mediana, 3° Q e Máximo) representa, sucessivamente, 25% do total de recursos (290).
- O suporte é igual ao *número de cestos / total de cestos* ($\text{total de cestos} = 26.234$).
- 25 % dos recursos, ou seja, 73 recursos, têm um *support count* inferior a 93,5. Isto é equivalente a dizer que o suporte é inferior a 0,36% ($93,5 / 26.234$), para 73 recursos.
- Sendo assim, se o *suporte mínimo* for definido acima de 0,36%, sabe-se à partida que 73 recursos (e todos os conjuntos de recursos que contenham pelo menos um destes 73 recursos) não cumprem esta restrição, pelo que não serão considerados pelo algoritmo de descoberta de regras de associação (*Apriori*, por exemplo).
- O espaço de procura ficará desta forma restrito a, no máximo, 2^{217} conjuntos de recursos ($217 = 290 - 73$).

- Este raciocínio repete-se para os restantes quartis apresentados neste quadro. Sendo porém importante evidenciar que se o suporte mínimo for definido acima de 4,57% (1.199 / 26.234) o espaço de procura não conterá nenhum conjunto.

Os 290 recursos, ou conjuntos de recursos de tamanho 1, resultam em 11.632 conjuntos de recursos de tamanho 2. Com estes conjuntos de dimensão 2, já é possível gerar regras de associação (um antecedente e um consequente), desde que as condições *suporte mínimo* e *confiança mínima* o permitam.

Os *support counts* destes conjuntos têm associadas as seguintes medidas de localização e dispersão:

Min	1° Q	Mediana	Média	3° Q	Máx	Desvio Padrão
1	1	4	14,5	15	232	26,43

O *box plot* e o histograma que se seguem, dão uma ideia de como estes valores se distribuem:

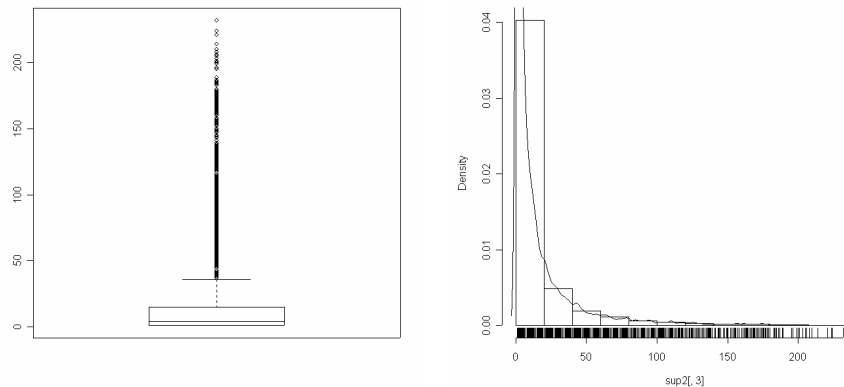


Figura 4.12 Distribuição do support count dos conjuntos de recursos de tamanho 2

Estes dados apresentam um elevado número de *outliers*, pelo que a média e o desvio padrão apresentados devem ser vistos com algum cuidado.

Os factos mais evidentes nestes dados são:

- 75% ($0,75 * 11.632 = 8.724$) dos conjuntos de tamanho 2 têm suportes inferiores a $15 / 26.234 = 0,06 \%$; e que o suporte mais elevado, neste caso, é de $232 / 26.234 = 0,88\%$.
- Se o suporte mínimo for definido acima de 0,88%, significa que não serão encontrados conjuntos de tamanho 2 nos dados. Consequentemente, não serão também encontrados conjuntos de dimensão superior. Uma vez que para gerar regras de associação é necessário encontrar, pelo menos, conjuntos de dimensão 2, isto significa que se o suporte for definido acima deste valor (0,88%), não serão geradas quaisquer regras a partir destes dados.

Estas conclusões consolidam cada vez mais a ideia de que os suportes destes dados são, de facto, muito baixos. Conforme já foi referido, esta evidência pode ser justificada pelo facto dos *itens* deste problema - os recursos da *Enabler* - terem características diferentes dos *itens* de um problema do tipo *market basket analysis* - artigos de um supermercado, por exemplo.

Outra conclusão que se pode tirar, a partir das análises efectuadas aos conjuntos de tamanho 2, é que, de uma maneira geral, os recursos da *Enabler* trabalham muito uns com os outros de forma dispersa (apesar de haver um núcleo restrito de associações fortes entre alguns destes recursos).

Antes de concluirmos esta fase da metodologia que está a ser seguida, importa conhecer os cestos com algum detalhe:

- Dos 26.234 cestos deste problema, 75% (cerca de 19.676) têm 3 ou menos de três recursos. Ou seja, só 6.558 cestos (25%) é que têm potencial para gerar regras que envolvam 4 ou mais recursos.
- Por outro lado, há 25% de cestos (6.558) com um recurso apenas. Estes cestos são considerados na contagem dos suportes dos conjuntos de recursos, em particular, na contagem dos suportes dos conjuntos de tamanho 1. No entanto,

não escondem qualquer tipo de informação relativamente às associações que possam existir nos dados, e, como tal, não vão ser utilizados na geração das regras.

- O maior cesto que está presente neste dados tem 64 recursos. Atendendo ao facto de que estamos perante um contexto de projectos, este número não deixa de ser surpreendente, o que leva a tirar algumas conclusões relativamente à dimensão do projecto a que este cesto pertence. Com efeito e após confirmação, verifica-se que este cesto pertence a um projecto de grande dimensão, de acordo com os critérios apresentados anteriormente (projecto 40).

Os gráficos seguintes dão uma ideia da forma como é que estes dados estão distribuídos:

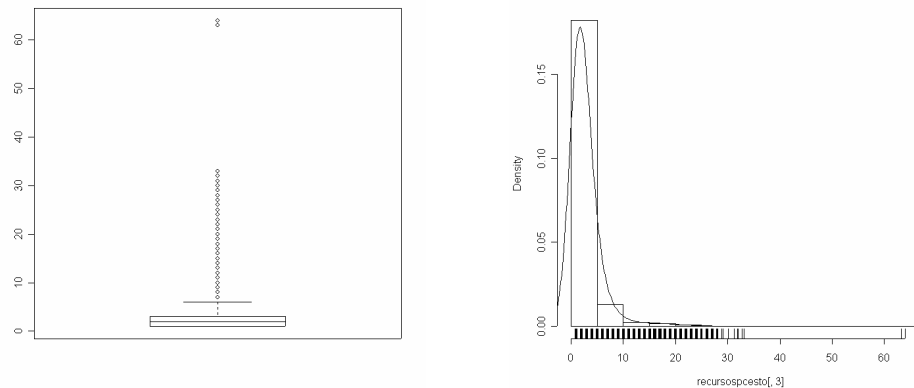


Figura 4.13 Distribuição do número de recursos por cesto

Com estes dois gráficos, concluímos esta fase.

O conhecimento que neste momento existe sobre os dados, ajuda a conhecer melhor o problema e o contexto que o envolve – a *Enabler*; os seus recursos; e os seus projectos - e é um importante contributo para as próximas fases da metodologia, visto que partimos

para elas com alguns pressupostos sobre os dados que ajudarão na tomada de algumas decisões. Ou seja:

- Sabe-se de onde provêm os dados para as fases seguintes e qual o formato destes dados. Em relação ao conteúdo, sabe-se também que a informação foi filtrada no sentido de se obter apenas dados relacionados com projectos – os que são relevantes no contexto de planeamento que está a ser perseguido. Existe a noção exacta dos volumes de informação envolvidos.
- Sabe-se o valor máximo para o suporte dos recursos (4,45%).
- Sabe-se que se o suporte mínimo for definido acima de 0,88%, não serão descobertas quaisquer regras a partir destes dados. Este facto permitirá evitar que se efectuem experiências desnecessárias nos próximos passos.
- Sabe-se que cerca de 25% dos 26.234 cestos não irão contribuir para a constituição das regras, uma vez que têm um recurso apenas.

As análises que foram apresentadas foram complementadas por análises multivariadas, que estão representadas no *Anexo 2 Análises Multivariadas*. Estas permitiram identificar grupos de recursos mais activos em projectos – irão aparecer em mais regras. Este conhecimento pode ser utilizado em trabalho futuro (não desenvolvido nesta tese) com o objectivo de diminuir a dimensão do problema, logo com vantagens ao nível da carga computacional – mantendo a mesma qualidade dos resultados - ao eliminar dos dados os recursos menos activos em projectos.

O próximo passo será então a preparação dos dados para posterior construção e selecção do melhor modelo de regras que servirá de base para o sistema de recomendação de recursos.

5 Preparação de Dados e Modelação

5.1 Construção do Modelo

É nesta fase que o modelo de regras, que irá servir de base ao sistema de recomendação de recursos, será criado. Antes, contudo, será necessário preparar os dados para que possam ser utilizados pelo sistema de geração de regras que se escolheu.

Para gerar as regras de associação, seleccionou-se uma implementação em *Java*¹ do *Apriori*, o *Caren*² [Azevedo, P. J. (2003)]. Esta implementação do *Apriori* foi desenvolvida com o objectivo de implementar um sistema de classificação baseado em regras de associação [Liu B. et al. (1998)], tal como foi descrito no sub capítulo 3.2 *Regras de Associação para Classificação*. No *Anexo 3 Caren* é apresentada uma breve descrição dos parâmetros do *Caren* que foram utilizados.

Em termos de preparação de dados, foi necessário efectuar duas operações:

- primeiro, foi necessário normalizar os dados, visto que o *Caren* é sensível a letras maiúsculas e minúsculas, e os dados originais tinham os mesmos projectos representados tanto por letras maiúsculas, quanto por letras minúsculas;
- segundo, foi necessário criar um ficheiro de texto com os cestos, a partir da tabela *cestos*, com as suas chaves (a chave de cada cesto é composta pelo *projecto* e pelo *dia*) separadas dos recursos por “;”.

¹ <http://java.sun.com>

² Disponível gratuitamente em: www.di.uminho.pt/~pja/class/caren.html

Exemplo de um excerto deste ficheiro de texto:

```
...
(002F)2002-05-20;Almeida, Rosário P.
(002F)2002-05-20;Correia, Paulo R.
(002F)2002-05-20;Duarte, Carlos A.
(002F)2002-05-20;Santos, Jorge R.
(002F)2002-05-20;Sousa, Ricardo A.
(002F)2002-05-21;Almeida, Rosário P.
(002F)2002-05-21;Correia, Paulo R.
(002F)2002-05-21;Santos, Jorge R.
(002F)2002-05-21;Sousa, Ricardo A.
(002F)2002-05-22;Correia, Paulo R.
(002F)2002-05-22;Duarte, Carlos A.
...
```

Neste exemplo consegue-se identificar que o primeiro cesto ((002F)2002-05-20) tem cinco recursos: “Almeida, Rosário P.”, “Correia, Paulo R.”, “Duarte, Carlos A.”, “Santos, Jorge R.” e “Sousa, Ricardo A.”; que o segundo cesto ((002F)2002-05-21) tem quatro recursos: “Almeida, Rosário P.”, “Correia, Paulo R.”, “Santos, Jorge R.” e “Sousa, Ricardo A.”; que o terceiro cesto ((002F)2002-05-22) tem dois recursos: “Correia, Paulo R.”, “Duarte, Carlos A.”; etc.

O modelo de recomendação, bem como o modo de o avaliar, propostos nesta tese, seguem estratégias apresentadas em Sarwar, B. et al. (2000), Sarwar, B. et al. (2001), Breese, J. S. et al. (1998) e Jorge, A. et al. (2002)b. Desta forma, e sintetizando os passos descritos no Sub Capítulo 3.5 *Sistemas de Recomendação e Regras de Associação*, seguem-se os princípios que foram adoptados para a construção e avaliação deste sistema:

- Este modelo descreve-se formalmente através da *Expressão 3.2*.
- Para proceder à sua avaliação, a totalidade dos cestos foi dividida de forma aleatória em dois conjuntos diferentes: *Treino* e *Teste*. A percentagem pela qual foi efectuada esta divisão foi, respectivamente, 80% e 20% do número total de cestos.

- O conjunto de *Treino* foi utilizado para gerar as regras para o modelo de recomendação.
- Para cada cesto do conjunto de *Teste*, foi apagado, de forma aleatória, um único recurso. O conjunto total de recursos que foi apagado desta forma foi chamado *Hidden* (pode também ser visto como um conjunto de cestos com um recurso cada – aquilo que se pretende prever). O conjunto dos cestos aos quais foi retirado um recurso, foi chamado *Observável*.
- O conjunto de recomendações $\{r_1, r_2, \dots, r_n\}$ para um determinado cesto do conjunto *Observável*, pode ser representado por:

$$\{ \langle chave_cesto, r_1 \rangle, \langle chave_cesto, r_2 \rangle, \dots, \langle chave_cesto, r_n \rangle \}$$

- Ao conjunto formado pela reunião dos conjuntos de recomendações para todos os cestos do conjunto *Observável*, chamou-se *Recomendações*.
- Este modelo foi avaliado através da comparação do conjunto de recomendações que ele faz (*Recomendações*), dado o conjunto *Observável*, com os recursos que pertencem ao conjunto *Hidden*.
- O número N de recomendações produzidas para cada cesto do conjunto *Observável* pode variar. Cada modelo de recomendação irá ser utilizado com diferentes valores de N e em cada um destes casos serão avaliadas as medidas: *Recall*, *Precision* e *F1*.

O esquema seguinte é uma instância particular da *Figura 3.2*, aplicada a este caso específico:

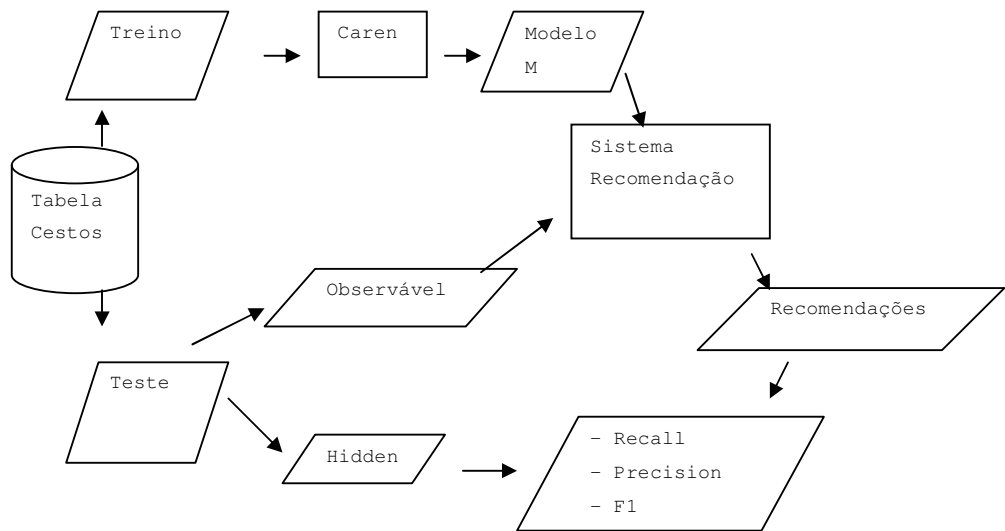


Figura 5.1 - Passos necessários para avaliar o sistema de recomendação de recursos

Alguns dados importantes:

	Nº Registos	Nº Cestos
Tabela Cestos	70.212	26.234 - (100%)
Treino	56.337	20.987 - (80%)
Teste	13.875	5.247 - (20%)
Observável	8.628	3.015
Hidden	5.247	5.247

O conjunto *Observável* tem 3.015 cestos porque existem $5.247 - 3.015 = 2.232$ cestos em *Teste* com um recurso apenas. Se um cesto contém apenas um recurso, ao “esconder” este seu recurso, este cesto fica com zero recursos, logo desaparece do conjunto *Observável*. Este valor indica que o número máximo de recomendações quando o N é igual a 1, é de 3.015. Este número é máximo porque podem existir alguns cestos no conjunto *Observável* que não consigam encontrar em M pelo menos uma regra que cumpra os critérios para a formação do conjunto R , e, como tal, não vão ter recomendações associadas.

O facto do número máximo de recomendações ser diferente do número de recursos em *hidden*, quando o $N = 1$, implica que o *Recall* nunca será igual à *Precision*. Isto é, os denominadores da *Equação 3.2* e da *Equação 3.3* nunca são iguais, para estes dados.

Importa referir que, para implementar todos os passos ilustrados na *Figura 5.1* foram desenvolvidos no âmbito deste trabalho uma série de programas em R (disponíveis no *Anexo 4 Programas em R*), designadamente para separar a base de dados inicial no conjuntos *Treino* e *Teste*; para preparar o conjunto de *Treino* para que este pudesse ser utilizado pelo *Caren*; para criar os conjuntos *Observável* e *Hidden*; para implementar o protótipo do motor do sistema de recomendação propriamente dito; para gerar todas as recomendações relativas ao conjunto *Observável*; e, por fim, para calcular o *Recall*, a *Precision* e o *F1*.

5.2 Resultados Experimentais

Foram gerados variados modelos para diferentes valores do suporte mínimo e da confiança mínima. Cada um destes modelos foi utilizado, conjuntamente com o conjunto *Observável*, como entrada do sistema de recomendação desenvolvido no âmbito deste trabalho, o qual, por sua vez, produziu os conjuntos *Recomendações* respectivos. As várias combinações experimentadas para o suporte mínimo e para a confiança mínima, conduziram aos seguintes resultados:

<i>Suporte</i> <i>Mínimo</i>	<i>Nº Mínimo</i> <i>Cestos</i>	<i>Confiança</i> <i>Mínima</i>	<i>Nº Regras</i> <i>Geradas</i>	<i>Nº de "Não</i> <i>Respostas"</i>	<i>% "Não</i> <i>Respostas"</i>
0,005	105	0,5	629	2.238	74,23%
0,005	105	0,1	903	1.341	44,48%
0,003	63	0,5	8.023	1.897	62,92%
0,003	63	0,1	8.957	338	11,21%

Tabela 5.1 Dados resultantes das diferentes combinações de suporte e confiança experimentadas

Nº mínimo de cestos (“número de cestos do conjunto *Treino*” * “Suporte Mínimo”) significa que só os conjuntos de recursos, independentemente do seu tamanho, que

aparecem num número de cestos superior a este valor é que irão ser considerados no processo de geração de regras do *Apriori*.

Mais atrás foi referido que o número máximo de recomendações, caso N seja igual a 1, é 3.015. Isto porque podem existir alguns cestos no conjunto *Observável* sem recomendações associadas. O N° de “*Não Respostas*” representa exactamente este valor para cada um dos modelos gerados. A percentagem da coluna seguinte é o quociente entre o número de *não respostas* e o número de cestos do conjunto *Observável* (3.015).

Estes valores permitem tirar algumas conclusões:

- O parâmetro “suporte mínimo” tem um impacto muito superior ao impacto da “confiança mínima”, no que diz respeito ao número de regras geradas – este efeito já era esperado, no seguimento do que foi exposto no sub capítulo 2.3 *Seleção de Regras*.
- O número de regras geradas não tem grande influência na “quantidade” de recomendações efectuadas (número de não respostas). Neste aspecto, o valor da confiança mínima tem um impacto bem mais elevado.

A questão que se coloca neste momento é a de avaliar também a qualidade das recomendações efectuadas pelos vários modelos, pois cada um destes pode, por exemplo, estar a recomendar pouco, mas bem; ou pode estar, de forma inversa, a recomendar muito, mas com menos qualidade. O ideal é, obviamente, conseguir atingir um ponto que maximize estes dois factores.

De início serão apresentados os resultados obtidos pelos modelos de regras gerados com confiança mínima igual a 0,5, considerando para este efeito diferentes valores de N .

N	Suporte Mínimo = 0,005			Suporte Mínimo = 0,003			
	Recall	Prec.	F1	Recall	Prec.	F1	Rnd
1	0,054	0,360	0,093	0,091	0,425	0,150	0,003
2	0,059	0,299	0,098	0,100	0,352	0,155	0,007
3	0,061	0,269	0,099	0,103	0,318	0,155	0,010
5	0,063	0,241	0,100	0,105	0,282	0,153	0,017
10	0,065	0,218	0,101	0,109	0,252	0,152	0,034
20	0,066	0,210	0,101	0,111	0,236	0,151	0,069

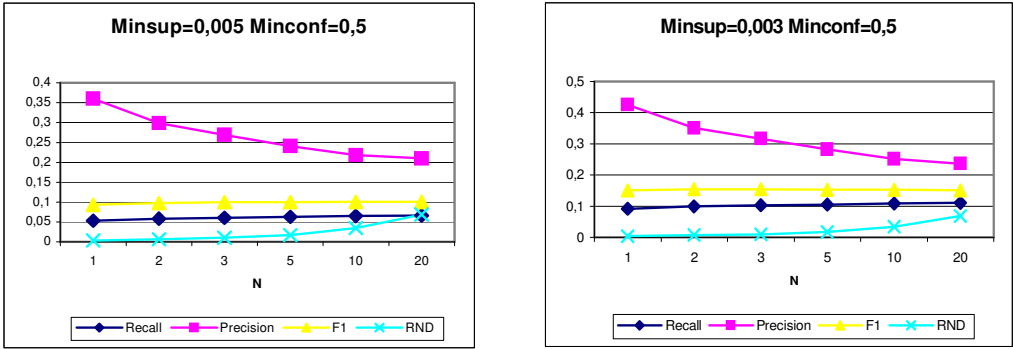


Figura 5.2 Recall, Precision e F1 para as duas combinações de suporte mínimo e confiança mínima experimentadas, para vários valores de N

Os valores para a coluna *Rnd* - escolha aleatória de recursos - são obtidos dividindo o valor de N por 290, e estão a servir como valor de referência. Mais à frente (sub capítulo 5.3.1 *Modelo com Regras Default*) serão também utilizados como referência, os valores do *recall* e da *precision* para as recomendações por defeito (os recursos mais prováveis *à priori*). Quando $N = 1$, a recomendação por defeito para todos os cestos do conjunto observável é o recurso com o suporte mais elevado no conjunto de treino; quando o $N = 2$, as recomendações por defeito para todos os cestos do conjunto observável são os dois recursos com o maior suporte no conjunto de treino; e assim sucessivamente.

Pelos valores do *recall* apresentados, pode-se concluir que para maior parte dos valores de N , a probabilidade de obtermos pelo menos uma recomendação correcta é quase

sempre superior à escolha aleatória de recursos (coluna “Rnd”). A excepção verifica-se para o modelo gerado com o suporte mínimo igual a 0,005, quando o N é igual a 20. Nesta situação é preferível a escolha aleatória de recursos.

Para os restantes casos, o valor do *recall* é superior à escolha aleatória de recursos, sendo que esta ordem de grandeza varia (aproximadamente) entre]2, 16[vezes para o primeiro modelo; e entre]2, 26[, vezes para o segundo modelo. Conforme seria de esperar pela definição desta medida, apresentada anteriormente, os valores do *recall* têm a tendência para crescer com o N .

Em relação à *precision*, verifica-se que cada recomendação individual obtida através do primeiro modelo, quando N é igual a um, tem 36% de probabilidade de estar correcta. Nas mesmas condições, as recomendações individuais obtidas pelo segundo modelo têm 42,5% de probabilidade de estarem correctas. Uma vez que a *precision* vai-se tornando menos interessante à medida que o N aumenta, estes são os valores mais elevados para esta medida. Os limites inferiores (quando o N é igual a 20) para esta medida são 21% e 23,6%, respectivamente para o primeiro e segundo modelos.

Os valores do $F1$ no primeiro modelo, revelam que quanto maior for o N , melhor é a combinação *precision / recall*. Em relação ao segundo modelo, o padrão identificado revela que as duas medidas obtêm a melhor combinação quando o N é igual a 2.

Ao analisar estes resultados, verifica-se a tendência para obtermos valores para o *recall* baixos e, inversamente, valores para a *precision* elevados. Observando a definição e as fórmulas associadas a estas medidas, justifica-se este resultado pelo facto deste modelos terem um número elevado de *não respostas*, como foi apresentado mais atrás. Assim, estes modelos recomendam poucas vezes, mas quando o fazem, efectuem-no com qualidade.

Estes dados permitem concluir que uma alteração no suporte mínimo de 0,005 para 0,003, permitiu obter um sistema de recomendação com resultados mais relevantes e

interessantes. Importa então apresentar e analisar qual o impacto da confiança mínima neste contexto.

Desta forma, a experiência apresentada imediatamente antes foi repetida alterando o valor da confiança mínima para 0,1.

Suporte Mínimo = 0,005				Suporte Mínimo = 0,003			
N	Recall	Prec.	F1	Recall	Prec.	F1	Rnd
1	0,089	0,277	0,134	0,147	0,287	0,194	0,003
2	0,106	0,213	0,142	0,194	0,208	0,201	0,007
3	0,115	0,189	0,143	0,217	0,168	0,189	0,010
5	0,121	0,163	0,139	0,240	0,127	0,168	0,017
10	0,125	0,138	0,131	0,261	0,095	0,140	0,034
20	0,127	0,128	0,128	0,272	0,076	0,119	0,069

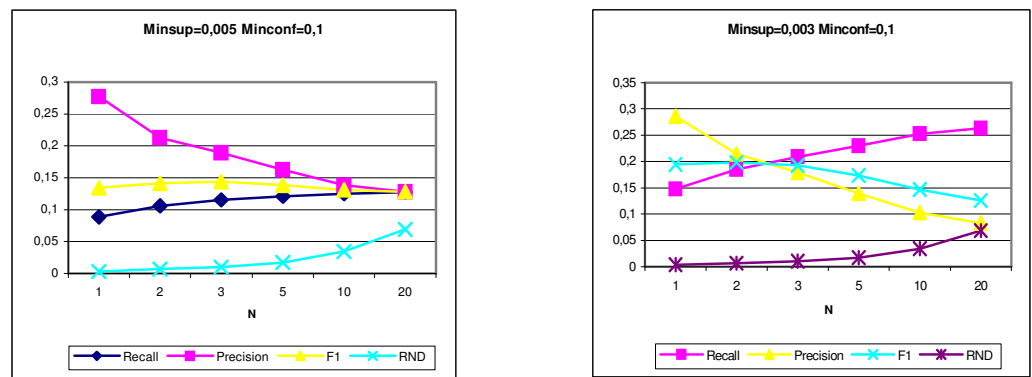


Figura 5.3 Recall, Precision e F1 para as duas combinações de suporte mínimo e confiança mínima experimentadas, para vários valores de N

Observa-se que os valores do *recall* são superiores aos valores apresentados no caso precedente. Este facto acentua a vantagem em assumir as recomendações efectuadas por este modelo, em relação à escolha aleatória de recursos. Com efeito, esta opção apresenta-se como sendo entre]2, 25[vezes melhor do que a escolha aleatória para o modelo gerado com o suporte mínimo igual a 0,005; e entre]4, 42[vezes melhor para o outro modelo.

A justificação para esta ocorrência, prende-se com o facto destes modelos estarem associados a um valor de *não respostas* muito inferior. Com mais recomendações, a probabilidade de acertar em, pelo menos uma, é, portanto, maior. Por outro lado, um numero de respostas superior implica que a probabilidade individual de cada resposta estar correcta diminua. É por este motivo que os valores da *precision* são menores neste caso.

O *F1* permite identificar que o valor de *N* que maximiza a combinação do *recall* e da *precision* é 3, para o primeiro modelo; e 2 para o segundo.

É igualmente relevante observar que é no modelo cujo suporte mínimo igual a 0,003, e cuja confiança mínima é igual a 0,1, que se verifica claramente através dos vários valores de *N* testados que o valor do *recall* torna-se superior ao valor da *precision*. Ou seja, neste modelo, quando o *N* é superior a 3, a probabilidade de ter pelo menos uma resposta certa é superior à probabilidade individual de cada recomendação estar correcta. Isto explica-se pelo reduzido número de *não respostas* associado (338). Quanto mais baixo for este número (*não respostas*), menor é o valor de *N* que permite identificar esta regularidade. Esta afirmação deduz-se a partir das fórmulas respectivas.

Após analisar estes quatro modelos, a escolha do modelo adequado para servir de base ao sistema de recomendação recai no modelo que apresenta melhores valores para o *recall*: modelo associado aos valores 0,003 e 0,1 para, respectivamente, o *suporte mínimo* e para a *confiança mínima*.

E porquê este modelo?

- Este modelo tem limites para o *suporte mínimo* e para a *confiança mínima* inferiores aos dos restantes modelos. Este facto implica que este contenha todas as regras que fazem parte dos outros modelos apresentados. Desta forma, este modelo acerta nas mesmas situações do que os restantes, e potencialmente pode acertar em mais situações, através das regras adicionais que o constituem.

- É legítimo que um utilizador de um sistema de recomendação sinta que este é de pouca utilidade caso o número de *não respostas* seja, de facto, muito elevado. O custo de acesso ao sistema pode não ser recompensado, se eventualmente muitas perguntas ficarem sem resposta. Este é o modelo que apresenta o valor mais baixo de *não respostas*.
- Um volume de respostas superior, aumenta a probabilidade de acertos, de acordo com a fórmula do *recall*. Como foi possível constatar, foi este modelo que apresentou os valores mais reduzidos para não respostas e, consequentemente, mais elevados para o *recall*.

É possível testar valores mais baixos para o *suporte mínimo* e para a *confiança mínima*. No entanto, tal escolha não parece razoável para o sistema de recomendação dado que tentar valores mais baixos do que 0,003 para o *suporte mínimo*, implica considerar conjuntos de recursos que aparecem em menos do que 63 cestos; e valores mais baixos do que 0,1 para a *confiança mínima*, implica seleccionar regras com menos de 10% de probabilidade de ocorrência.

Os valores do *recall* e *precision* obtidos pelo modelo seleccionado é semelhante ao obtido em circunstâncias análogas noutros trabalhos, nomeadamente Jorge, A. et al. (2002)b.

5.3 Experiências Adicionais

5.3.1 Modelo com Regras *Default*

Os vários modelos que foram experimentados, utilizaram a escolha aleatória de recursos como medida de referência. O que se propõe efectuar neste ponto é avaliar o desempenho do modelo constituído por regras *default*. Os resultados obtidos por este modelo serão igualmente utilizados como uma medida de referência (em alternativa à escolha aleatória de recursos). Antes de apresentar os resultados obtidos importa definir o que são regras *default*.

As regras *default* são regras com consequente, mas sem antecedente, tal como:

$$\emptyset \Rightarrow \textit{Consequente}$$

cujo significado é: “antecedente por defeito” (qualquer antecedente) implica *Consequente*. Estas regras possuem suporte igual à confiança, que, por sua vez, é igual ao suporte do respectivo consequente no conjunto de treino (conjunto a partir do qual foram criadas estas regras). As regras que constituem este modelo são então ordenadas por ordem decrescente em relação à confiança (ou suporte) de cada uma delas. As recomendações obtidas a partir deste modelo são geradas da seguinte forma: quando N é igual a um, todos os cestos do conjunto observável obtêm como recomendação o consequente da regra *default* com a confiança (ou suporte) mais elevada, isto é, o recurso com suporte mais elevado; quando N é igual a dois, todos os cestos do conjunto observável obtêm como recomendações os consequentes das duas regras *default* com as confianças (ou suportes) mais elevadas (os dois recursos com os suportes mais elevados); e assim sucessivamente. O *CAREN* gera este tipo de regras.

De seguida apresentam-se os resultados obtidos por este modelo, para vários valores de N :

N	Recall	Precision	F1	RND
1	0,011	0,019	0,014	0,003
2	0,017	0,015	0,016	0,007
3	0,029	0,013	0,018	0,010
5	0,034	0,012	0,018	0,017
10	0,057	0,010	0,017	0,034
20	0,108	0,009	0,017	0,069

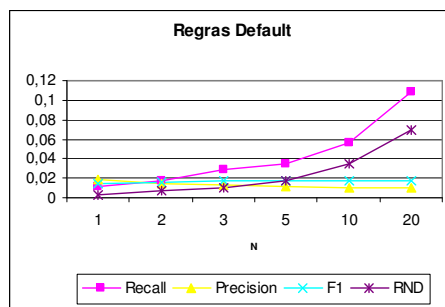


Figura 5.4 *Valore do Recall, Precision e F1, para o modelo constituído pelas regras default, para vários valores de N*

A partir destes dados observa-se que é sempre vantajoso assumir as recomendações por *default* em relação à escolha aleatória de recursos.

Se estes resultados forem comparados com os resultados apresentados anteriormente, obtidos para o modelo gerado com suporte mínimo igual a 0,003 e confiança mínima igual a 0,1, observa-se que é sempre vantajoso assumir as recomendações efectuadas pelo sistema de recomendação apresentado, em relação às recomendações obtidas por *default*. Em modo gráfico verifica-se que a proporção de respostas correctas é menor no modelo formado por regras *default*, para qualquer valor de N :

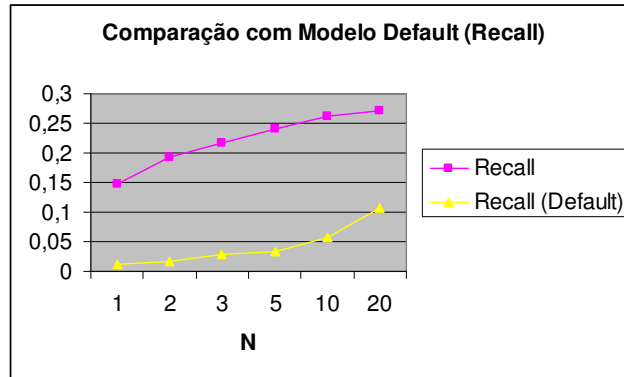


Figura 5.5 Comparação da proporção de respostas correctas (Recall), obtidas pelo modelo constituído pelas regras com suporte mínimo = 0,003 e confiança mínima = 0,1, e pelo modelo constituído pelas regras default

O gráfico seguinte permite igualmente verificar que a qualidade individual de cada recomendação é inferior no modelo formado por regras *default*, para qualquer valor de *N*:

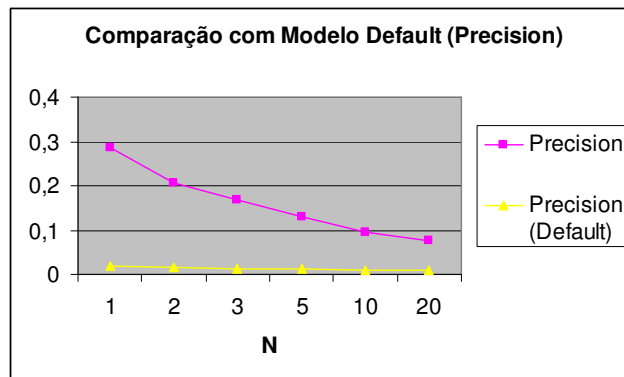


Figura 5.6 Comparação dos valores da precision, obtidas pelo modelo constituído pelas regras com suporte mínimo = 0,003 e confiança mínima = 0,1, e pelo modelo constituído pelas regras default

5.3.2 Impacto da informação disponível nos resultados

Os cestos só com um recurso não escondem qualquer tipo de relações que possam existir nos dados. Desta forma, não são geradas regras de associação a partir destes

cestos. Tal como se verificou mais atrás, esta situação resulta num número de cestos diferente entre o conjunto *Observável* e o conjunto *Hidden*.

Por estes motivos, retirou-se estes cestos dos dados de *teste*, e repetiu-se estas experiências com estes novos dados. O objectivo foi medir os impactos desta acção no *recall* e assim simular experimentalmente situações em que a recomendação só é feita quando há um mínimo de informação observada (pelo menos um recurso).

Os resultados obtidos foram (*suporte mínimo* = 0,003; *confiança mínima* = 0,1):

Nº de cestos de Teste, Observável e Hidden: 3.015

Nº de Não Respostas: 338

N	Recall	Prec.	F1	Rnd
1	0,255	0,287	0,270	0,003
2	0,338	0,208	0,257	0,007
3	0,378	0,168	0,233	0,010
5	0,418	0,127	0,197	0,017
10	0,455	0,095	0,158	0,034
20	0,474	0,076	0,131	0,069

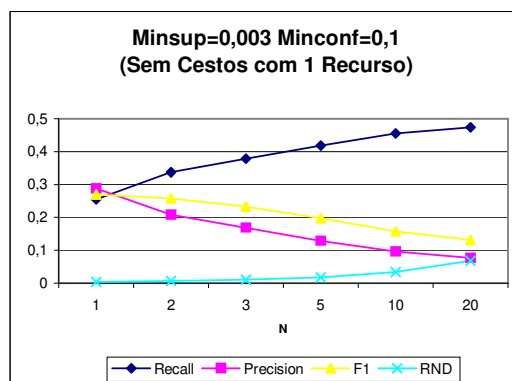


Figura 5.7 Recall, Precision e F1 com o modelo sem os cesto de tamanho 1

Observa-se a partir destes resultados que o valor do *recall* foi beneficiado ao serem retirados os cestos só com um recurso dos dados de *teste*. Isto é, quando *N* é igual a 1, a probabilidade de se conseguir pelo menos uma resposta correcta é de aproximadamente

26%. Este impacto no *recall* explica-se pelo facto do seu denominador (o *hidden*) ter diminuído significativamente.

No âmbito de trabalho futuro (não sendo, portanto, trabalho desenvolvido nesta tese), serão efectuadas experiências para estudar o impacto nos resultados, de se retirarem dos dados também os cestos de dimensão 2, 3, e assim sucessivamente. Desta forma será possível analisar o valor do *recall* (e da *precision*), em função da informação disponível.

5.3.3 Utilização do *Interest* para selecção de regras

O modelo de regras gerado com o *suporte mínimo* igual a 0,003 e a *confiança mínima* igual a 0,1 é constituído por 8.957 regras. Todo o processo de criação das recomendações para o conjunto *observável* (3.015 cestos), a comparação destas recomendações com os cestos de *hidden* e o cálculo das medidas que estão a ser utilizadas, demora cerca de 3 horas, para cada valor de *N* que foi testado (1, 2, 3, 5, 10, 20). Ou seja, em média, o tempo necessário para cada recomendação individual é de cerca de 3,58 segundos. Este desempenho foi conseguido num computador pessoal equipado com um processador *Intel Pentium IV* a 2GHz, com 512 MB de memória RAM.

No sub capítulo 2.3 *Seleccção de Regras*, foram apresentados alguns métodos cujo objectivo é reduzir o volume total de regras geradas, escolhendo apenas as que se apresentam com mais interesse para os utilizadores.

Uma vez que mais regras implicam um desempenho computacional inferior para o sistema de recomendação, sendo que o inverso é também válido, ou seja, menos regras equivalem a um desempenho computacional superior, foram igualmente efectuadas experiências no sentido de diminuir a dimensão do modelo de regras, na expectativa de um desempenho superior (em termos de tempo de processamento, mantendo valores aproximados para o *recall*).

Dada uma regra $A \Rightarrow B$, o *interest* foi apresentado neste sub capítulo como sendo uma medida que considera tanto $P(A)$ quanto $P(B)$: $interest(A \Rightarrow B) = P(A \cap B) / P(A)P(B)$. Outra maneira de escrever esta medida é: $interest(A \Rightarrow B) = confiança(A \Rightarrow B) / P(B)$. Com esta medida pode-se eliminar dos modelos as regras que contenham *itens* do antecedente não correlacionados [Brin, et al. (1997)] com *itens* do consequente.

Uma vez que o resultado do *Caren* inclui também o valor da confiança para cada uma das regras geradas, para obter o *interest* é necessário dividir a confiança de cada uma das 8.957 regras pelo suporte dos consequentes respectivos. Os resultados obtidos resumem-se através destas estatísticas e gráficos:

Min.	1° Q	Mediana	Média	3° Q.	Max.	Desvio P.
1.000	53.520	63.640	69.860	89.150	159.900	29.745

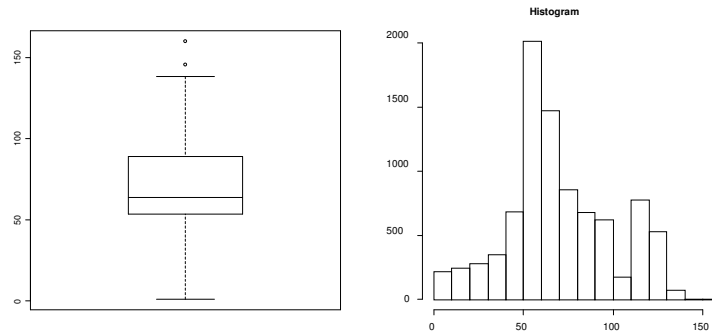


Figura 5.8 Distribuição dos valores do interest

Para reduzir o tamanho do modelo, utilizou-se 3 valores do *interest*: 54, 64 e 89, isto é, procedeu-se à eliminação das regras cujo *interest* é inferior a estes valores. Estes correspondem, respectivamente, ao 1° quartil, à mediana e ao 3° quartil. O impacto destes valores no tamanho do modelo é, respectivamente, 6.635 regras; 4.419 regras; e 2.243 regras.

Para obter o valor do *recall* e da *precision*, foi necessário gerar previamente o conjunto das recomendações (os testes foram efectuados com N igual a 1) para cada um destes 3 “modelos parciais”, ao que se seguiu a sua comparação com o conjunto *hidden*. A

Tabela 5.2 apresenta os resultados obtidos. A primeira linha já foi apresentada anteriormente (com o modelo completo), apenas está presente para servir de referência.

<i>Interest</i> >	<i>Nº</i> <i>Regras</i>	<i>Recall</i>	<i>Precision</i>	<i>F1</i>	<i>Min.</i> <i>Totais</i> <i>(aprox.)</i>	<i>Seg. p/</i> <i>recomendação</i> <i>(aprox.)</i>	<i>Nº Não</i> <i>Respostas</i>
0	8957	0,147	0,287	0,194	180	3,58	338
54	6635	0,069	0,418	0,118	133	2,65	2151
64	4419	0,052	0,488	0,095	88	1,75	2451
89	2243	0,016	0,518	0,032	45	0,90	2849

Tabela 5.2 Dados obtidos através dos variados modelos que resultaram da aplicação do *interest* para a selecção de regras. Nesta tabela pode-se observar o valor do *interest* utilizado para seleccionar as regras; o número de regras geradas, o *recall*; a *precision*; o *F1*; os minutos totais necessários para gerar as 3.015 recomendações; o tempo médio por recomendação (em segundos); e o número de não respostas.

A primeira percepção é que há, com efeito, ganhos em relação ao tempo, ou seja: um modelo maior implica mais tempo de processamento. O gráfico que se segue permite inclusivamente identificar uma relação linear entre o *interest* (consequentemente, o tamanho do modelo) e o tempo de processamento associado (para as 3.015 recomendações):

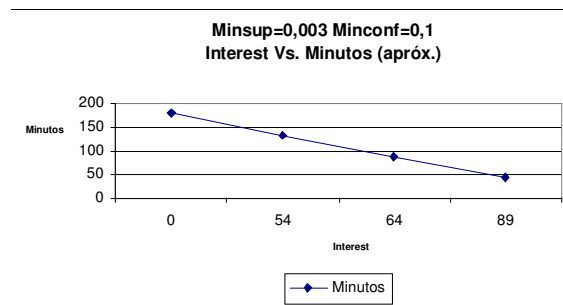


Figura 5.9 Relação entre o tempo necessário para processar todas as recomendações relativas ao conjunto observável, e os modelos obtido através de diferentes valores do *interest*.

O mesmo não pode ser dito em relação ao *recall* e ao número de não respostas.

O *recall* cai abruptamente quando passamos da situação em que não há restrição do *interest* (é igual a 0), para a situação em que se eliminam do modelo as regras cujo *interest* é inferior a 54. A partir deste valor, o *recall* apresenta mais estabilidade.

Estes factos podem ser visualizados através do gráfico seguinte:

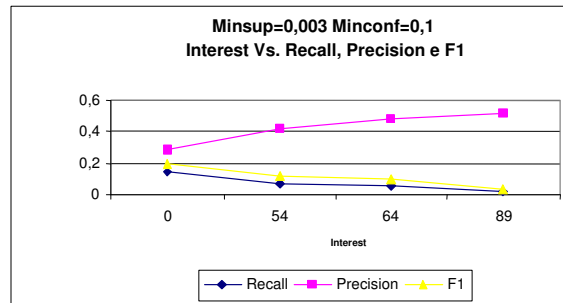


Figura 5.10 Relação entre os valores do Recall, Precision e F1, e os modelos obtido através de diferentes valores do interest

O número de *não respostas* dá um grande “salto” quando impomos a primeira restrição baseada no *interest*, ao modelo, após o que se mantém mais estável. Este comportamento pode ser observado através do seguinte gráfico:

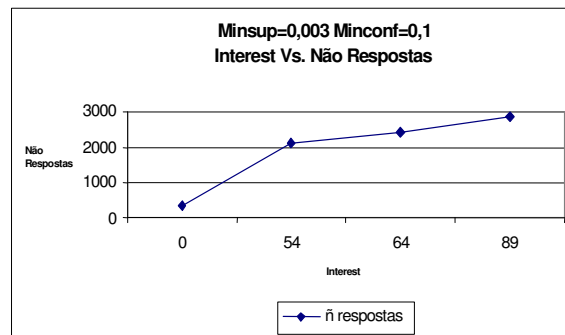


Figura 5.11 Relação entre o número de não respostas, e os modelos obtido através de diferentes valores do interest

O grande o bjectivo desta experiência foi tentar encontrar um modelo com menos regras, logo mais rápido a recomendar, mas que mantivesse um desempenho semelhante nas recomendações efectuadas.

Os resultados obtidos mostraram que em relação ao tempo conseguiu-se cumprir este objectivo. Em relação à capacidade de efectuar recomendações semelhantes, verificou-se que os vários modelos alternativos tentados, recomendam menos (valor mais elevado para as *não respostas*) e com menos qualidade (valores piores para o *recall*).

Sendo assim, as experiências efectuadas não conseguiram demonstrar a vantagem de reduzir a dimensão deste modelo de regras de associação através do *interest*. Uma interpretação mais aprofundada destes resultados, bem como a realização de mais experiências neste domínio ficam para trabalho futuro.

5.4 Recomendação de Equipas

A partir do protótipo do motor do sistema de recomendação que foi descrito ao longo do capítulo 4 *Recomendação de Recursos Humanos para Equipas de Projectos*, foi desenvolvida uma nova funcionalidade: *Recomendação de Equipas*.

Basicamente esta funcionalidade pode ser caracterizada do seguinte modo: dada uma equipa *E1*, constituída pelos recursos $\{a, b, c\}$, o sistema irá recomendar uma outra equipa *E2*, constituída pelos recursos $\{x, y, z\}$, ao substituir um recurso da primeira equipa, por outro recurso que considere mais apropriado. O novo recurso terá que ser do mesmo *nível* e da mesma *pool* do que o recurso que foi substituído. O caso particular desta situação é quando o sistema recomenda a mesma equipa, ou seja, não procede a nenhuma alteração da equipa inicial. Pode igualmente acontecer o caso do sistema não ter capacidade para recomendar qualquer equipa.

O objectivo desta funcionalidade é possibilitar ao *gestor do projecto*, ou ao *delivery manager*, a identificação de uma oportunidade de otimizar a equipa, no final do processo de constituição da mesma. É importante clarificar que o conceito “otimizar a equipa” segue os critérios deste sistema de recomendação.

Exemplo:

Suponha-se que um gestor de projecto constituiu a seguinte equipa de projecto (esta equipa foi gerada de forma aleatória com dados reais):

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Castro, Jorge L.	ERP/Back Office Resources	3
Mendes, José M.	ERP/Back Office Resources	1

Ao aplicar esta nova funcionalidade a esta equipa, o sistema produziu a seguinte recomendação:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Rocha, Manuela C.	ERP/Back Office Resources	3
Mendes, José M.	ERP/Back Office Resources	1

Ou seja, o sistema trocou o recurso *Castro, Jorge, L.* pelo recurso *Rocha, Manuela C.* Repare-se que estes dois recursos são do mesmo nível - 3 - e pertencem ambos à mesma *pool* – *ERP/Back Office Resources*.

O algoritmo que implementa esta nova funcionalidade é o seguinte:

```

recomendar_equipa(e)
/* e - equipa que se pretende otimizar */

    para cada equipa ei
        /* ei é sub equipa de e, de tamanho = [(tamanho de e) - 1], por se retirar desta
        sub equipa o recurso ri. A partir de e, é possível criar [tamanho de e] sub
        equipas ei*/
            gerar as recomendações de recursos de ei
            escolher a melhor destas recomendação, que seja do nível e da pool de ri
            /* a melhor recomendação corresponde à regra com confiança mais elevada.
            */

    de entre as [tamanho de e] recomendações seleccionadas no ciclo anterior,
    escolher a que tiver a confiança mais elevada

    recomendar a equipa formada pela sub equipa ei, que deu origem à recomendação
    seleccionada no passo anterior, e a recomendação associada

```

Este algoritmo permite concluir que a *recomendação de equipas* não é mais do que uma aplicação prática, alternativa, da funcionalidade de *recomendação de recursos*. Foi desenvolvida uma implementação em R deste algoritmo que se encontra no *Anexo 4 Programas em R*.

De seguida vai-se mostrar a lógica deste algoritmo através do exemplo anterior:

Pretende-se saber qual é a recomendação associada à equipa:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Castro, Jorge L.	ERP/Back Office Resources	3
Mendes, José M.	ERP/Back Office Resources	1

Para tal, prossegue-se com a geração das recomendações de recursos associadas às 3 sub equipas de tamanho 2, que é possível formar.

1ª Sub Equipa:

Recurso	Pool	Nível
Castro, Jorge L.	ERP/Back Office Resources	3
Mendes, José M.	ERP/Back Office Resources	1

Recurso Retirado:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4

Recomendação_Recurso ("Castro, Jorge L.", "Mendes, José M.") :

Sup.	Conf.	Recurso	Pool	Nível
0.00500	0.42857	Alves, André J.	ERP/Back Office Resources	2
0.00424	0.36327	Santos, Inês S.	ERP/Back Office Resources	2
0.00405	0.34694	Rocha, Manuela C.	ERP/Back Office Resources	3
0.00357	0.30612	Ribeiro, Maria E.	DW/BI/EAI Resources	1
0.00705	0.29249	Braga, Martinho A.	ERP/Back Office Resources	2
0.00319	0.27347	Goes, Henrique M.	Project Management / Consulting Resources	4
0.00319	0.27347	Sousa, André L.	ERP/Back Office Resources	1
0.00314	0.26939	Henriques, Sílvia T.	ERP/Back Office Resources	1
0.00310	0.26531	Nunes, João A.	ERP/Back Office Resources	2
0.00457	0.18972	Cunha, Avelino	CPC Resources	0

0.00343	0.14229	Belo, João D.	Project Management / Consulting Resources	4
0.00319	0.13241	Teixeira, José M.	ERP/Back Office Resources	2
0.00300	0.12451	Nogueira, João P.	ERP/Back Office Resources	3

Selecciona-se desta lista de recomendações, a melhor, cujo respectivo recurso é de nível 4 e simultaneamente pertence à *pool*: *Project Management / Consulting Resources*. Esta selecção está assinalada a **bold**. Neste caso, esta recomendação coincide com o recurso retirado da sub equipa que deu origem a esta lista.

2ª Sub Equipa:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Mendes, José M.	ERP/Back Office Resources	1

Recurso Retirado:

Recurso	Pool	Nível
Castro, Jorge L.	ERP/Back Office Resources	3

Recomendação_Recurso ("Goes, Henrique M.", "Mendes, José M."):

Sup.	Conf.	Recurso	Pool	Nível
0.00500	0.42857	Alves, André J.	ERP/Back Office Resources	2
0.00424	0.36327	Santos, Inês S.	ERP/Back Office Resources	2
0.00405	0.34694	Rocha, Manuela C.	ERP/Back Office Resources	3
0.00357	0.30612	Ribeiro, Maria E.	DW/BI/EAI Resources	1
0.00319	0.27347	Sousa, André L.	ERP/Back Office Resources	1
0.00314	0.26939	Henriques, Sílvia T.	ERP/Back Office Resources	1
0.00310	0.26531	Nunes, João A.	ERP/Back Office Resources	2

Selecciona-se desta lista de recomendações, a melhor, cujo respectivo recurso é de nível 3 e simultaneamente pertence à *pool*: *ERP/Back Office Resources*. Esta selecção está assinalada a **bold**.

3ª Sub Equipa:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Castro, Jorge L.	ERP/Back Office Resources	3

Recurso Retirado:

Recurso	Pool	Nível
Mendes, José M.	ERP/Back Office Resources	1

Recomendação_Recursos ("Goes, Henrique M.", "Castro, Jorge L."):

Sup.	Conf.	Recurso	Pool	Nível
0.00705	0.29249	Braga, Martinho A.	ERP/Back Office Resources	2
0.00329	0.21362	Alves, André J.	ERP/Back Office Resources	2
0.00319	0.20743	Mendes, José M.	ERP/Back Office Resources	1
0.00310	0.20124	Rocha, Manuela C.	ERP/Back Office Resources	3
0.00457	0.18972	Cunha, Avelino	CPC Resources	0
0.00429	0.17787	Nunes, João A.	ERP/Back Office Resources	2
0.00343	0.14229	Belo, João D.	Project Management / Consulting Resources	4
0.00319	0.13241	Teixeira, José M.	ERP/Back Office Resources	2
0.00300	0.12451	Nogueira, João P.	ERP/Back Office Resources	3

Selecciona-se desta lista de recomendações, a melhor, cujo respectivo recurso é de nível 1 e simultaneamente pertence à pool: ERP/Back Office Resources. Esta selecção está assinalada a bold. Neste caso, esta recomendação coincide com o recurso retirado da sub equipa que deu origem a esta lista.

O próximo passo é seleccionar, de entre as recomendações de recursos já seleccionadas, a que apresentar a confiança mais elevada. Neste caso foi:

Sup	Conf	Recurso	Pool	Nível
0.00405	0.34694	Rocha, Manuela C.	ERP/Back Office Resources	3

Sendo assim, a equipa recomendada é:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Rocha, Manuela C.	ERP/Back Office Resources	3
Mendes, José M.	ERP/Back Office Resources	1

Antes de terminar com a descrição desta funcionalidade, é importante expor um padrão muito interessante que foi identificado através das várias experiências efectuadas. Este padrão surge ao efectuar uma recomendação a partir de uma equipa que, por sua vez, já era o resultado de uma recomendação. Quando sucede esta situação, os resultados obtidos foram sempre a mesma equipa. Por exemplo:

- considere-se a equipa {a, b, c};

- ao efectuar a recomendação da equipa {a, b, c}, obtém-se a equipa {a, x, c};
- ao efectuar a recomendação da equipa {a, x, c}, obtém-se a novamente a equipa {a, x, c}.

Este dado evidencia alguma estabilidade no processo de recomendação de equipas, ou seja, deduz-se que há um ponto de paragem no processo iterativo de recomendação de equipas. A interpretação e justificação deste padrão nos dados não vai ser explorada nesta tese, ficando, portanto, para trabalho futuro.

A alteração do algoritmo para a recomendação de equipas para que este proceda à substituição de mais do que um recurso por recomendação, será também alvo de trabalho futuro.

5.5 Resumo do Capítulo

Neste capítulo obteve-se um modelo (suporte mínimo = 0,003 e confiança mínima = 0,01) com uma proporção de recomendações correctas de cerca de 15%, quando N é igual a 1. Verificou-se também que este modelo recomenda melhor do que a escolha aleatória de recursos e do que as recomendações por *default*. É, contudo, importante evidenciar que nos casos em que este modelo “não recomenda bem” segundo o critério do *Recall*, as recomendações efectuadas podem, ainda assim, ser consideradas adequadas. Esta avaliação ficará para mais tarde quando for apresentada a conclusão sobre o estudo efectuado para medir a percepção de uma amostra de recursos da *Enabler*, face à adequação das recomendações efectuadas por este sistema. Este estudo foi baseado nas respostas disponibilizadas por esta amostra de recursos a um inquérito elaborado e distribuído para este efeito.

Foi também apresentada uma nova funcionalidade desenvolvida a partir da “recomendação de recursos”, a “recomendação de equipas”.

Nos próximos capítulos será apresentada a avaliação efectuada a este sistema, segundo os critérios de negócio – de acordo com a metodologia *CRISP-DM* -, e serão formulados diversos cenários possíveis para uma possível implementação deste modelo.

6 Avaliação

O desempenho do sistema de recomendação de recursos, apresentado no capítulo 4 *Recomendação de Recursos Humanos para Equipas de Projectos*, foi avaliado através das medidas *recall* e *precision*. Observando os numeradores das definições formais destas medidas, respectivamente a *Equação 3.2* e a *Equação 3.3*, verifica-se que estas consideram apenas as recomendações exactas.

Considere-se por exemplo o seguinte caso: ao cesto formado pelos recursos {a, b, c} foi retirado o recurso {c}. Ou seja, {a, b} é o conjunto observável e {c} é o conjunto *hidden*. Se este conjunto observável for aplicado como *input* do sistema de recomendação de recursos e este, por sua vez, devolver a recomendação {d}, então esta recomendação não irá contribuir positivamente para o valor do *recall*, dado que {c} não é igual a {d} - conforme se pode confirmar pela *Equação 3.2*.

Este exemplo não significa que a recomendação efectuada (neste caso {d}) não seja adequada em determinado contexto. Por outro lado, mesmo que o sistema tivesse recomendado {c}, não significaria que esta seria a mais adequada em todas as situações.

6.1 Percepção dos Utilizadores

Por estes motivos considerou-se relevante estudar a percepção dos (potenciais) utilizadores deste sistema, face à adequação das recomendações que este produz. Para efectuar este estudo foi elaborado um questionário (*Anexo 5 Questionário*) que foi passado a uma amostra de recursos da *Enabler*. Esta amostra foi retirada de um universo de 56 recursos (potenciais utilizadores deste sistema), sendo que este universo é constituído por recursos com ligações (directas ou indirectas) à actividade de planeamento de equipas: elementos da Comissão Executiva; níveis IV das áreas de *delivery* e comercial; e níveis III das áreas de *delivery* e comercial. O plano de amostragem seleccionado foi a amostragem aleatória simples [Vicente, P. et al. (2001)]. A dimensão da amostra é de 17 recursos, ou seja $17/56 = 30,4\%$ do tamanho da população.

A elaboração do questionário seguiu os princípios expostos em [Hill, M (2002)]. Este questionário foi dividido em 3 partes, correspondendo cada uma delas a diferentes utilizações deste sistema:

- Recomendação de recursos – foram apresentadas 6 equipas de projecto, geradas de forma aleatória, às quais se aplicou a funcionalidade de recomendação de recursos desenvolvida no âmbito desta tese. Pediu-se aos recursos da amostra que exprimissem a sua percepção em relação ao grau de adequação das recomendações apresentadas.
- Recomendação de equipas – foram apresentadas 6 equipas de projecto, geradas de forma aleatória, às quais se aplicou a funcionalidade de recomendação / optimização de equipas desenvolvida no âmbito desta tese. Pediu-se aos recursos da amostra que exprimissem a sua percepção em relação ao grau de adequação das recomendações apresentadas.
- Construção interactiva de equipas – foi dado um exemplo de como seria possível criar uma equipa com quatro elementos, a partir de um recurso inicial (dado), adicionando uma recomendação – um recurso – em cada passo deste processo. Neste caso, este sistema foi utilizado para “navegar” de forma interactiva pelos recursos da *Enabler*. No final deste processo, pediu-se aos recursos da amostra que exprimissem a sua percepção em relação à adequação da equipa final formada, bem como ao processo que conduziu à formação da equipa final apresentada (esta parte do questionário teve duas questões).

Os recursos apresentados neste questionário foram recursos reais da *Enabler*, portanto conhecidos dos elementos da amostra seleccionada. O grau de adequação de cada recomendação foi medido de acordo com a seguinte escala:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Tabela 6.1 Escala de adequação das recomendações utilizada no inquérito desenvolvido

Os resultados finais do preenchimento do inquérito foram compilados no sentido de se obter o valor médio da percepção dos recursos da amostra em relação à adequação das recomendações apresentadas. A partir de testes *t* para as médias, obteve-se um intervalo de confiança, a 95%, para as médias da população. A *Figura 6.1* sintetiza os resultados obtidos.

	Média	Intervalo Confiança a 95%	
		Limite Inf.	Limite Sup.
Primeira Parte			
Recomendação de recursos:	3,31	3,03	3,59
Segunda Parte			
Recomendação de equipas:	3,80	3,59	4,02
Terceira Parte			
Construção Interactiva de uma Equipa (Equipa Final):	3,88	3,57	4,19
Construção Interactiva de uma Equipa (Processo):	3,65	3,34	3,96

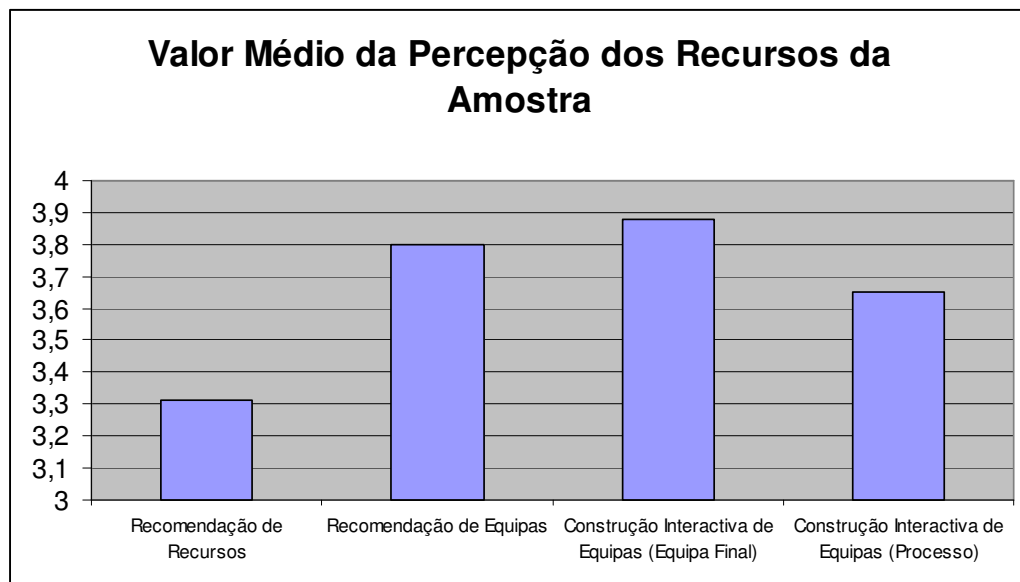


Figura 6.1 Resultados do preenchimento do inquérito

A partir desta tabela e deste gráfico pode-se concluir, com uma confiança de 95%, que os (potenciais) utilizadores do sistema de recomendação não têm, em média, uma percepção negativa sobre a adequação das recomendações efectuadas por este.

6.2 Discussão

Os recursos da amostra seleccionada complementaram o preenchimento do inquérito com alguns comentários relevantes neste enquadramento. Em síntese estes comentários podem ser divididos em dois:

1º Comentário

Contexto em que a recomendação está a ser efectuada: as equipas apresentadas no inquérito foram geradas de forma aleatória, sem se considerar as valências das pessoas, o cliente, o tipo de projecto, etc. Sendo assim, é difícil avaliar a adequação das recomendações, dado que, por exemplo, a equipa formada pelos recursos {a, b} pode ser adequada para o projecto x, mas inadequada para o projecto y.

Esta observação é relevante e sustenta a necessidade do modelo de restrições proposto no sub capítulo 7.2 *Modelo de Restrições*. No entanto, é importante ter presente que o pressuposto que está por de trás da construção de um modelo de *data mining* é que este consiga descobrir padrões que não são explícitos nos dados. Assim, é de esperar que o modelo deste sistema de recomendação assuma implicitamente um determinado contexto, em função dos recursos que lhe forem apresentados.

2º Comentário

A recomendação de recursos baseada na informação sobre o histórico de alocações, tende a viciar a formação de equipas. Isto é, as equipas tendem a ser constituídas sempre pelos mesmos elementos, sendo portanto difícil integrar novos elementos em equipas que trabalharam frequentemente juntas no passado.

Com efeito este comentário é pertinente, contudo este é o pressuposto do sistema desenvolvido – recomendar recursos com base nas escolhas feitas no passado. A integração de novos elementos em equipas que trabalharam juntas no passado, será

conseguida por outros meios – o sistema de recomendação de equipas não é um sistema automático de constituição de equipas. Entretanto é importante referir que o modelo de regras de associação deverá ser actualizado periodicamente, com o objectivo de descobrir as novas associações que vão aparecendo nos dados.

7 Operacionalização

7.1 Proposta de Implementação

O processo de requisição de recursos para os projectos, é efectuado, tal como foi apresentado em 4.2.2 *Caracterização da Enabler*, através da aplicação *Service Sphere*. Daí que faça sentido que o processo de recomendação de recursos parta deste ponto. A ideia é estender a funcionalidade desta aplicação com a inclusão do protótipo do motor do sistema de recomendação de recursos desenvolvido e avaliado nas fases anteriores.

Sendo assim, o diagrama apresentado anteriormente (*Figura 4.6*), sobre o processo de constituição de equipas para projectos, deve ser incrementado com esta nova funcionalidade, tal como está ilustrado através da *Figura 7.1*:

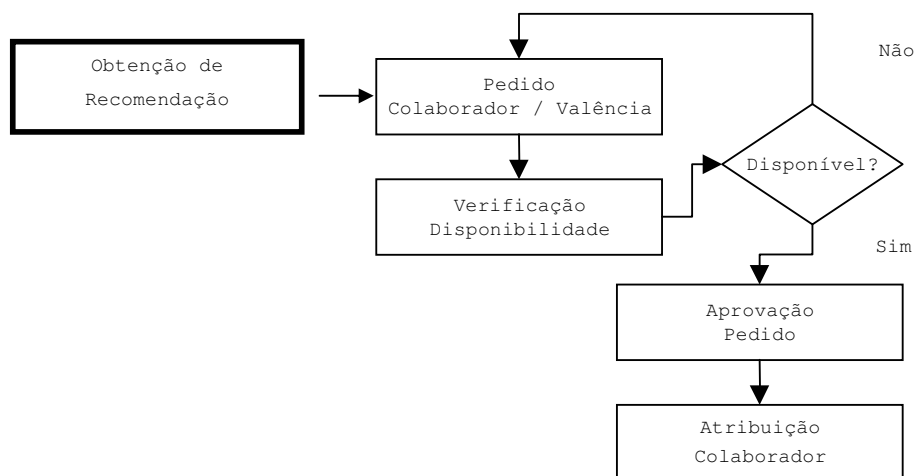


Figura 7.1 Descrição gráfica do processo de constituição de equipas para projectos utilizando o sistema de recomendação de recursos

Para integrar esta nova funcionalidade na aplicação (*Service Sphere*), será necessário alterar o seu interface, através de desenvolvimento adequado, de forma a que este passe a conter uma chamada a esta nova funcionalidade. A *Figura 7.2* apresenta um exemplo de como o interface desta aplicação podia ser modificado para atingir este objectivo (a

nova opção está devidamente assinalada). Caso o utilizador seleccione esta nova opção, ser-lhe-ia apresentada a respectiva lista de recomendações (Figura 7.3).

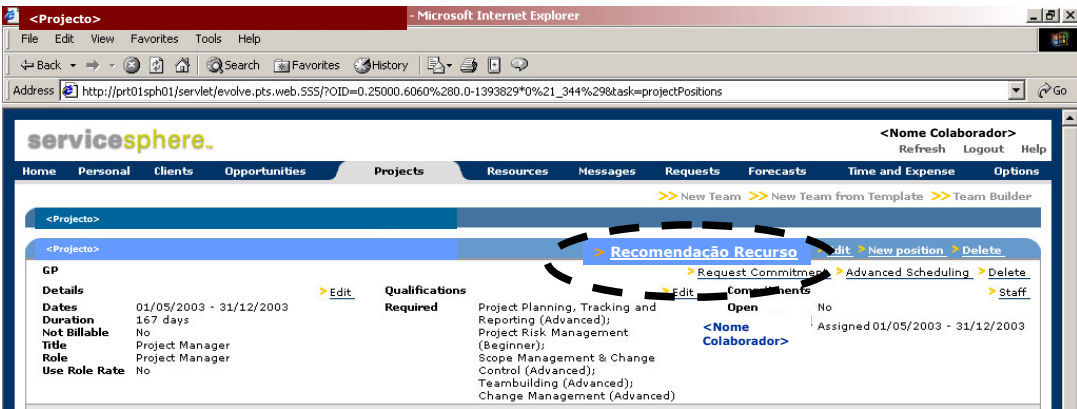


Figura 7.2 Proposta para integração da funcionalidade de recomendação de recursos no interface actual do Service Sphere

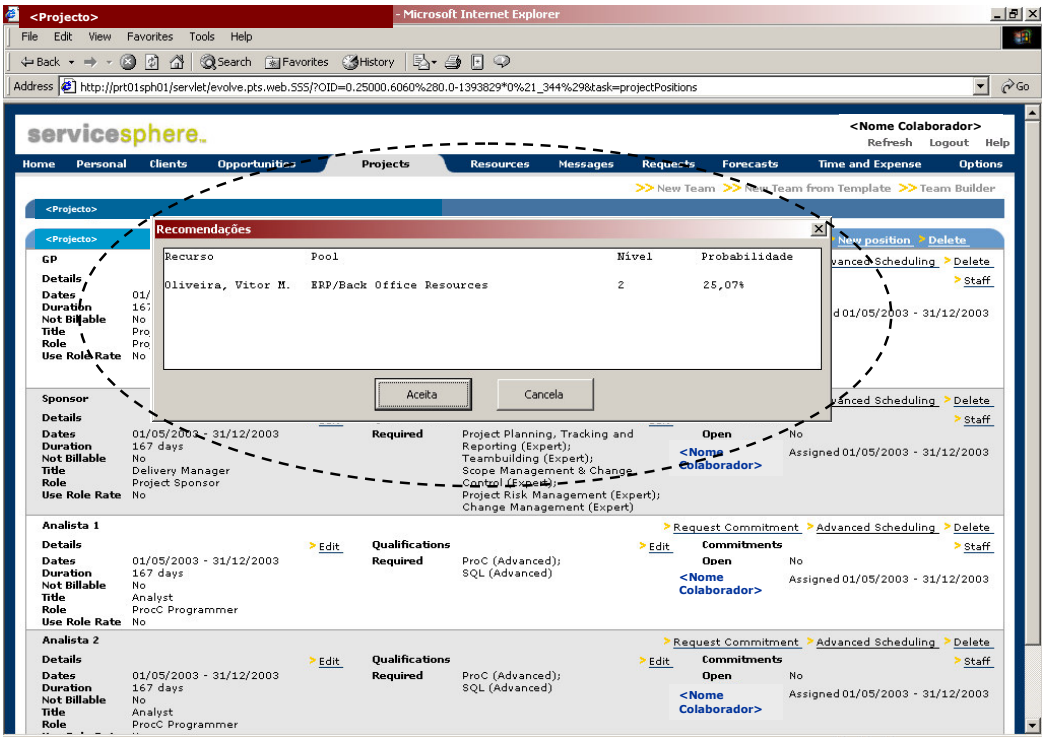


Figura 7.3 O sistema recomenda um recurso

O número N de recomendações obtidas, pode ser um parâmetro da aplicação. Se este número for superior a *um* o utilizador terá que seleccionar a recomendação que pretende adicionar à equipa, a partir da lista de N recomendações, antes de carregar no botão *Aceita*.

Com esta nova funcionalidade integrada no *Service Sphere*, o gestor de projecto tem à disposição uma ferramenta que lhe permite: obter recomendações de recursos; obter “segundas opiniões” face às escolhas de recursos que pretende para a sua equipa; e, simultaneamente, consegue prosseguir com o processo normal de requisição / constituição de equipas em projectos.

.

7.2 Modelo de Restrições

Este sistema de recomendação é estritamente baseado num modelo de regras de associação. É de esperar que este modelo de regras contenha “conhecimento” que está guardado, de forma latente, nos dados que lhe deu origem – o histórico dos *time reports*. Esse “conhecimento” está subjacente às escolhas feitas no passado e que desta forma estão guardadas neste registo histórico.

No entanto, é de esperar que determinados contextos impliquem que as recomendações produzidas estejam sujeitas a certas restrições, tais como:

- *A disponibilidade dos recursos no momento*. Requisitar um recurso que não está disponível nos períodos pretendidos (alocado a outros projectos, em férias, em formação, etc) pode resultar numa perda de tempo. Isto porque, muito provavelmente, a resposta do respectivo *resource manager* será negativa. Porém, é importante que a requisição de recursos nestas circunstâncias seja possível, dado que, por vezes, estas situações permitirem “negociar a troca” de recursos entre projectos.

- *Necessidade de um recurso de uma pool de recursos específica.* Tal como foi apresentado em 4.2.1 *Caracterização da Enabler*, as *pools* de recursos agrupam recursos de valências técnicas semelhantes. As melhores recomendações deste sistema podem apontar para recursos pertencentes a *pools* diferentes daquelas de que o projecto necessite. Neste caso, as recomendação serão muito provavelmente inadequadas na prática, apesar de serem as melhores do ponto de vista teórico, de acordo com o modelo de recomendação adoptado.
- *Necessidade de um recurso de um nível específico.* A explicação dada no ponto anterior pode ser facilmente transcrita para este ponto.
- *Necessidade de um recurso com valências técnicas específicas.* As explicações efectuadas nos dois pontos anteriores podem ser facilmente transcritas para este ponto. Apesar das *pools* agruparem recursos com valências técnicas semelhantes, pode ser importante especificar em concreto qual a que se pretende, caso a caso.

Por este motivo é que se propõe uma mudança na arquitectura deste sistema: a inclusão de um módulo de restrições, cuja função será “filtrar” as recomendações efectuadas, em função de determinados critérios previamente seleccionados.

Este módulo estará ligado ao sistema de recomendação e ao *Service Sphere*, dado que é nesta aplicação que está registada a informação sobre cada recurso: o seu nível, a *pool* a que pertence, as suas valências técnicas e o seu plano de alocação – quando e em que projectos irá trabalhar futuramente; quando é que irá ter férias; quando é que terá formação; e quando é que estará disponível. A nova arquitectura está representada na *Figura 7.4*.

Do ponto de vista de utilização, o que irá acontecer quando um utilizador seleccionar a opção *Recomendar Recurso*, é que a aplicação irá perguntar, através de uma nova janela, quais as restrições que o utilizador pretende que sejam consideradas para a recomendação que deseja efectuar. A figura *Figura 7.5* mostra um exemplo com esta nova janela.

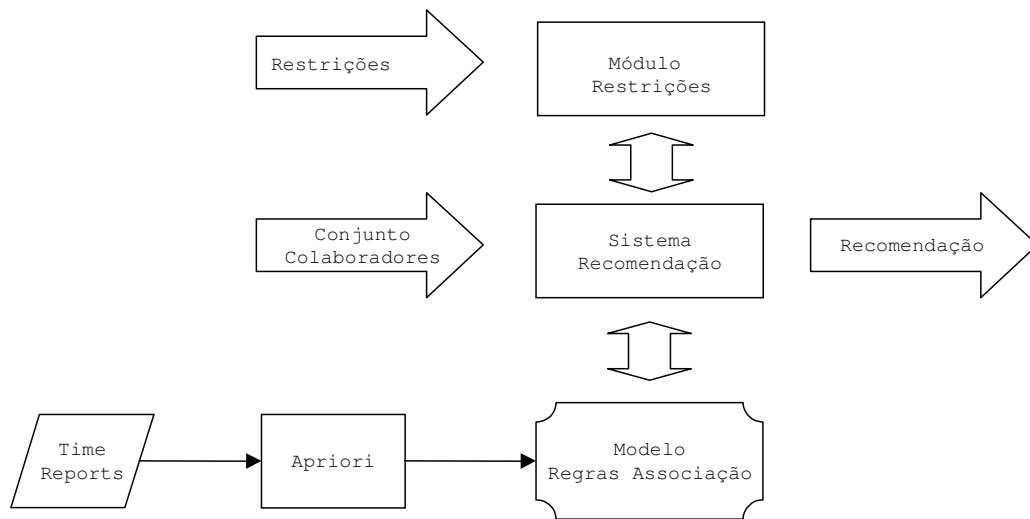


Figura 7.4 Arquitectura do sistema de recomendação incremental com o modelo de restrições

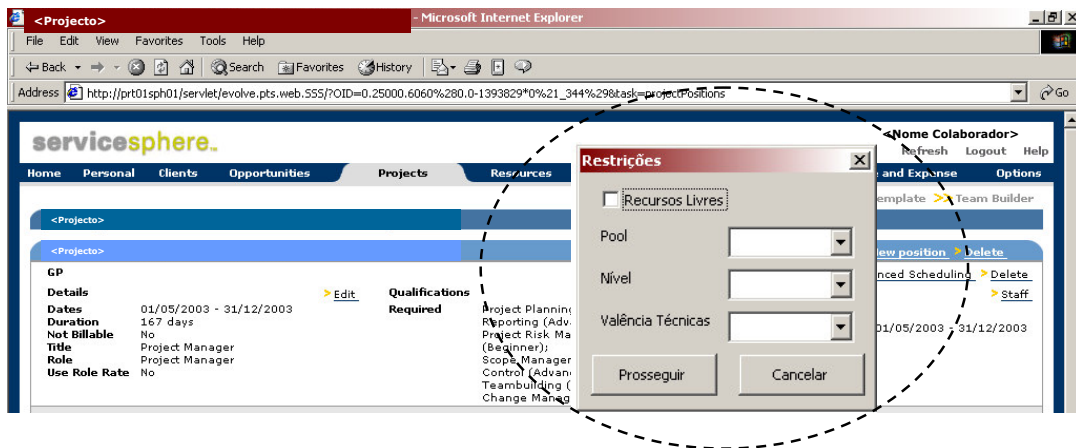


Figura 7.5 Introdução das restrições no sistema

O algoritmo que implementa o sistema de recomendação de acordo com a *Expressão 3.2*, terá que ser devidamente alterado para filtrar as recomendações produzidas, em função das restrições definidas pelo utilizador através da janela mostrada na *Figura 7.5*.

7.3 Recomendação de Equipas

Para integrar esta funcionalidade no *Service Sphere*, deve-se seguir a estratégia que está a ser apresentada. Ou seja, o interface deve ser alterado, através de desenvolvimento adequado, para que seja adicionada uma chamada a esta nova funcionalidade.

A figura seguinte mostra um exemplo deste interface:

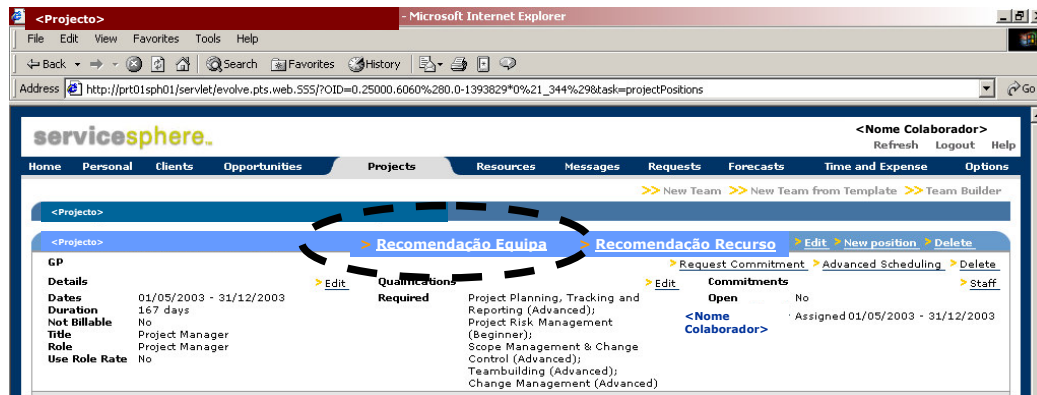


Figura 7.6 Incorporação da funcionalidade de recomendação de equipas no interface do Service Sphere

8 Conclusões e Trabalho Futuro

Nesta dissertação foi proposto um método de apoio ao planeamento de recursos humanos. Como suporte deste método foi desenvolvido um sistema de recomendação de recursos, tendo como base um modelo de regras de associação construído a partir da informação histórica sobre projectos reais: quem trabalhou com quem; em que projectos; e durante quanto tempo. A escolha de um modelo baseado em regras de associação para este efeito, segue o pressuposto de que o histórico referido guarda de forma implícita “conhecimento” relevante para este processo de planeamento: as escolhas que foram feitas no passado e o contexto em que estas escolhas foram efectuadas; e segue o pressuposto de que um modelo de regras de associação tem a capacidade para “descobrir” este “conhecimento” de forma a que este seja aplicável em situações futuras de planeamento, através do sistema de recomendação.

Sendo esta actividade de planeamento complexa e fundamental para as empresas organizadas por projectos, o objectivo deste método é responder ao conjunto de desafios que surgem neste enquadramento e que foram apresentados no início do *capítulo 4*:

- *Onde é que se pode encontrar, explicita ou implicitamente, a informação necessária para a tarefa de planeamento de recursos?*
- *De que forma é que esta informação pode estar organizada para facilitar o seu acesso?*

Esta tese mostrou que a informação necessária para efectuar este tipo de planeamento pode estar guardada implicitamente numa organização. No caso em estudo - a *Enabler* - esta informação está guardada no histórico dos *time reports* preenchidos pelos seus recursos que participam em projectos. A informação deste histórico foi utilizada para construir o modelo de regras de associação entre recursos. No entanto é importante referir que para implementar um sistema de recomendação com estas características, uma organização tem que dispor de um sistema de informação que permita registar a actividade dos seus recursos, nos diversos projectos, ao longo do tempo. Caso contrário

a construção do modelo de recomendação pode estar comprometida, ou a metodologia a aplicar teria que ser diferente.

- *Sendo as empresas organizações dinâmicas e em crescimento, será possível concentrar esta informação em colaboradores chave? E se estes colaboradores abandonarem a empresa?*

Um sistema com estas características auxilia e é um complemento à actividade humana de planeamento de equipas. Contudo, minimiza a dependência que pode existir entre a organização e determinados recursos chave - recursos com informação mais global sobre o histórico de projectos da empresa e sobre os vários colaboradores da empresa. Por outro lado a utilização frequente deste sistema pode ajudar a “formar” novos recursos chave, dando-lhes a possibilidade de ter acesso a “conhecimento” sobre as relações que existem entre os vários recursos da empresa.

- *Onde é que se pode pedir uma segunda opinião face às escolhas efectuadas?*
- *É possível obter com facilidade um conselho ou uma recomendação para efectuar uma escolha deste tipo?*

Mostrou-se que este sistema pode, com efeito, servir para confirmar a escolha de um recurso, ou para dar conselhos ou recomendações para a selecção de recursos em determinado projecto. Os resultados experimentais obtidos pelo *recall* mostraram que, quando N é igual a 1, a proporção de respostas correctas ronda os 15%, isto é, em 15% das vezes há uma recomendação relevante; de igual modo, para N igual a 5, por exemplo, e quando se removem dos dados os cestos só com um recurso, o valor do *recall* é próximo dos 42%. Os resultados do inquérito apresentado, mostraram que a sensibilidade dos potenciais utilizadores deste sistema, face às recomendações que este efectua, é positiva.

Para efectuar esta tese, estudou-se e apresentou-se um conjunto de trabalhos relevantes nesta área. Nestes são dados diversos exemplos de aplicações práticas de *data mining*, em diversos domínios diferentes. Em particular foram estudadas aplicações práticas de

regras de associação e de sistemas de recomendação. Com esta tese demonstrou-se a aplicabilidade prática de *data mining* em mais um domínio, isto é, estudou-se a aplicação prática de um sistema de recomendação, baseado em regras de associação, a um tipo de problemas diferente: o planeamento de recursos humanos para equipas de projectos.

Um projecto de *data mining* pode pressupor um investimento avultado em ferramentas sofisticadas de *software*. Nesta tese utilizou-se a base de dados *mysql* para guardar, tratar e transformar a informação necessária para construir o modelo; utilizou-se o R para desenvolver todos os programas necessários e para efectuar todas as análises estatísticas necessárias; e utilizou-se uma implementação do *Apriori*, o *Caren*. Dado que todas estas ferramentas encontram-se disponíveis gratuitamente na Internet, mostrou-se que é possível levar a cabo um projecto de *data mining*, sem ter que efectuar esforço financeiro em ferramentas de *software*.

A aplicação de uma metodologia de *data mining*, neste caso o *CRISP-DM*, ajudou organizar e a estruturar todo o trabalho necessário para levar a cabo esta tese. O desenvolvimento do caso prático apresentado ajudou a conhecer melhor a *Enabler*, os seus recursos e os seus projectos.

O sistema proposto é escalável. A partir do sistema de recomendação de recursos construiu-se o sistema de recomendação de equipas. Este sistema foi apreciado positivamente pelos recursos da amostra que foi seleccionada para preencher o inquérito, como se pode verificar pelos seus resultados. A avaliação teórica deste sistema fica para trabalho futuro. Como trabalho futuro fica igualmente mais uma extensão da recomendação de recursos: “optimização de equipas”. Esta é uma abordagem híbrida entre as funcionalidades de recomendação de recursos e recomendação de equipas, ou seja, dada uma equipa, este sistema deve recomendar N recursos adicionais e, simultaneamente, deve substituir os recursos da equipa que considere conveniente.

Os dados utilizados na construção do modelo de regras de associação compreendem um período de 15 meses – desde Setembro de 2001 até Novembro de 2002. Dada a dinâmica das organizações - novos recursos; recursos que mudam de área ou são promovidos; novos tipos de projectos; etc - é aceitável supor que este modelo pode perder as suas capacidades de previsão com o tempo. Deste modo entende-se que este modelo deve ser refeito periodicamente no sentido de manter o seu desempenho. Como trabalho futuro fica o estudo de métodos e técnicas para garantir a “validade” actual do modelo, e para medir o período mínimo pelo qual o modelo deve ser actualizado.

O desempenho deste sistema foi medido utilizando métricas de *information retrieval*: o *recall* e a *precision*. No entanto estas medidas consideram apenas as recomendações exactas, sem considerar se as que não o são, são adequadas ou não. Por este motivo propõe-se para trabalho futuro o estudo do impacto de se utilizar como numerador destas métricas uma medida de adequação (um somatório de um *score* de adequação) das recomendações efectuadas, em vez do número de recomendações correctamente efectuadas. Considere-se por exemplo o seguinte caso:

- os dados de teste contêm, por hipótese, apenas um cesto;
- este cesto é constituído pela equipa $\{a, b, c\}$;
- a esta equipa retira-se aleatoriamente o recurso $\{b\}$ (recurso *Hidden*);
- se a recomendação ($N = 1$) para a equipa observável $\{a, c\}$ for $\{b\}$, então o *recall* é 1 ($|Hidden \cap Rec| / |Hidden| = 1/1$); se a recomendação anterior fosse $\{x\}$, então o *recall* seria 0 ($|Hidden \cap Rec| / |Hidden| = 0/1$).
- no entanto, quando o sistema recomenda o recurso $\{x\}$, este pode ser de certa forma adequado para trabalhar com a equipa $\{a, c\}$, apesar do respectivo resultado do *recall* (0)
- assim, se for utilizado um *score* de adequação, e se este for, por exemplo, 0,9 para a recomendação $\{x\}$, a nova medida a propor em trabalho futuro poderia ter o valor 0,9 ($score \wedge Hidden = 0,9/1$).

O custo de aceitar uma recomendação não adequada não foi estudado no âmbito desta tese e tem condições para ser trabalho com interesse para efectuar futuramente.

A construção das regras de associação para este trabalho foi efectuada com ajuda de uma implementação do *Apriori*. No entanto este algoritmo considera apenas a presença dos *itens* nas transações, desprezando, portanto, o peso de cada *item* na transação. Imagine-se o que sucede frequentemente com os *managers* da *Enabler* – trabalham em muitos projectos, mas poucos minutos em cada um deles. Com o *Apriori* é natural que surjam muitas associações entre *managers* e recursos de nível I, II e III. No entanto, se estas associações fossem pesadas com os minutos trabalhados em cada cesto, obtinha-se um tipo de associações diferente. Assim propõe-se como trabalho futuro a utilização de uma modificação do *Apriori* para que este passe a contemplar também a quantidade dos *itens* em cada transacção.

Epílogo

Possuir “conhecimento” permite decidir sobre as actividades futuras. O *data mining* consiste na descoberta de conhecimento, tendências ou padrões interessantes, em grandes conjuntos de dados.

Nesta tese esta temática foi abordada e foi desenvolvido um modelo que permite a descoberta de conhecimento que conduz à tomada de decisões sobre planeamento de recursos em equipas de projectos.

Sendo a actividade de planeamento de equipas fundamental para as empresas organizadas por projectos, possuir o conhecimento certo para executar esta tarefa de planeamento torna-se assim uma factor chave de sucesso para estas empresas.

Isto é:

“o segredo do sucesso está em saber algo que mais ninguém sabe” [Aristotle Onassis].

Referências

Adamo, Jean-Marc (2001), *Data Mining for Association Rules and Sequential Patterns*, New York: Springer-Verlag.

Agrawal, R., Imielinski, T. e Swami, A. (1993), “Mining association rules between sets of items in large data bases”. In proc. of the ACM SIGMOD Int’l Conf. On Management of Data (ACM SIGMOD ’93), Washington, USA, May 1993.

Agrawal, R. e Srikant, R. (1994), “Fast Algorithms for Mining Association Rules”. In Proc. of the 20th Int’l Conf. On Very Large Databases (VLDB’94), Santiago, Chile, June 1994.

Azevedo, P. J. (2003), “CAREN – A Java Based Apriori Implementation for Classification Purposes”. Technical Report, January 2003. Consultado a 30 de Março de 2003 em www.di.uminho.pt/~pja/class/caren.html

Bayardo R., Agrawal R. (1999), “Mining the Most Interesting Rules”. In Proc. of the 5th International Conference on Knowledge Discovery and Data Mining, KDD - 99, 145-153.

Berry, Michael J. A. e Linoff, Gordon S. (1997), *Data Mining Techniques for Marketing, Sales and Customer Support*, New York: John Wiley & Sons, Inc.

Berry, Michael J. A. e Linoff, Gordon S. (2000), *Mastering Data Mining*, New York: John Wiley & Sons, Inc.

Breese, J. S., Heckerman, D. e Kadie, C. (1998) “Empirical Analysis of Predictive Algorithms for Collaborative Filtering”. Appears in Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July, 1998. Morgan Kaufmann Publisher.

Brin, S., Motwani, R., Ullman, J. D. e Tsur, S. (1997), “Dynamic itemset counting and implications rules for market basket data”. In Proc. of the ACM SIGMOD Int’l Conf. on Management of Data, 1997.

Chapman, Pete et al. (2000), *CRISP – DM 1.0 Step-by-step data mining guide*, consultado a partir de <http://www.crisp-dm.org> em Novembro de 2002.

Domingo C., Gavalda R. e Watanabe O. (1998), “On-line Sampling Methods for Discovering Association Rules”. Draft.

Goldberg, D., Nichols, D., Oki, B. M. e Terry, D. (1992), “Using Collaborative Filtering to Weave an Information Tapestry”. Comun. ACM 35, 12 (Dec. 1992), 61 – 70.

Gordon, S. R., Gordon J. R. (2003), *Information Systems – A Management Approach*, New York: John Wiley & Sons, Inc.

Guimarães, Rui Campos e Cabral, José A. Sarsfield, (1997), *Estatística*, Lisboa: McGraw Hill.

Han, J. e Fu, Y. (1999), “Mining Multiple-Level Association Rules in Large Databases”. IEEE Transactions on Knowledge and Data Engineering, Vol 11. Nº 5. 1999.

Han, J., Pei, J. e Yin, Y. (2000), “Mining Frequent Patterns Without Candidate Generation”. SIGMOD 2000.

Hill, Charles W. L. (2001), *International Business*, McGraw-Hill.

Hill, M. (2002), *Investigação por Questionário*, Lisboa: Edições Sílabo.

Hipp, J., Guntzer, U. e Nakhaeizadeh, G. (2000), “Algorithms for Association Rule Mining – A General Survey and Comparison”. SIGKDD 2000.

Inmon, W. H. (1996), *Building The Data Warehouse*, New York: John Wiley & Sons, Inc.

Jarke, Matthias et. al (2003), *Fundamentals of Data Warehouses*, New York: Springer-Verlag.

Jorge, A., J. Poças e P. Azevedo (2002)a, “Post-processing Operators for Browsing Large Sets of Association Rules”, in Proceedings of Discovery Science 2002 Eds, Steffen lange, Ken Satoh, Carl H. Smith, Springer-Verlag, LNCS 534, 2002.

Jorge, A., Alves, M. A. e Azevedo, P. (2002)b, “Recommendation With Association Rules: A Web Mining Application”, in Proceedings of Data Mining and Warehouses, a sub-conference of information society 2002, EDS. Mladenec, D., Grobelnik, M., Josef Stefan Institute, October 2002.

Jovanoski, V. e Lavrac, N. (2001), “Classification rule learning with APRIORI-C”. In ECML/PKDD'01 workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning, pages 81--92. ECML/PKDD'01 workshop notes, September 2001.

Kimball, Ralph (1996), *Data Warehouse ToolKit*, New York: John Wiley & Sons, Inc.

Kimball, Ralph et. al (1998), *The Data Warehouse Lifecicle Toolkit*, New York: John Wiley & Sons, Inc.

Kleinberg, J., Sandler, M. (2003), “Convergent Algorithms for Collaborative Filtering”. Proc. 4th ACM Conference on Electronic Commerce, 2003.

Laudon, Kenneth C. e Laudon, Jane P. (2002), *Management Information Systems*, New Jersey: Prentice Hall.

Lent, B., Swami, A., e Widom, J. (1997), “Clustering Association Rules”. Proceedings of the IEEE International Conference on Data Engineering, 1997, pp. 220-231.

Li, W., Han, J. e Pei, J. (2001), “CMAR: Accurate and Efficient Classification Based on Multiple Class Association Rules”. Simon Fraser University 2001.

Lin, W., Ruiz, C. e Alvarez, S. (2000), “A New Adaptative-Support Algorithm for Association Rule Mining”. Worchester Polytechnic Institute. 2000.

Liu B., Hsu, W. e Ma Y. (1998), “Integrating Classification and Association Rule Mining”. KDD98, New York 1998.

Liu B., Hsu W. e Ma Y. (1999), “Pruning and Summarizing the Discovered Associations”, In Proc. of the 5th International Conference on Knowledge Discovery and Data Mining, KDD-99, 125-134.

Ma, Y., Wong, K. e Liu, B. (2000)a, “Effective Browsing of the Discovered Associations Rules Using the Web”. National University of Singapore 2000.

Ma, Y., Wong, K. e Liu, B. (2000)b, “Web for Data Mining: Organizing and Interpreting the Discovered Rules Using the Web”, School, SIGKDD Explorations, ACM SIGKDD, Volume 2, Issue 1, July 2000.

Mannila, H. e Toivonen, H. (1996), “Multiple uses of frequent sets and condensed representations”. University of Helsinki 1996.

Mena, Jesus, (1999), *Data Mining Your Website*, Boston: Digital Press.

Mitchell, Tom M., (1997), *Machine Learning*, WCB / McGraw-Hill.

Murteira, Bento J. F, (1993), *Análise Exploratória de Dados*, Lisboa: McGraw-Hill.

Murteira, Bento J. F, Ribeiro, Carlos S., Silva, João A. e Pimenta, Carlos, (2002), *Introdução à Estatística*, Lisboa: McGraw-Hill.

Pennock, D. M., Horvitz, E., Lawrence, S. and Giles C. L. (2000), “Collaborative Filtering by Personality Diagnosis: A Hybrid Memory – and Model – Based Approach”. In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000), pp. 473-480, Morgan Kaufmann, San Francisco, 2000.

Neves, João Poças das (2002), “Ambiente de Pós-processamento para Regras de Associação”. Tese de Mestrado em Análise de Dados e Sistemas de Apoio à Decisão, Faculdade de Economia Universidade do Porto.

Quillan, J. R., (1993), *C4.5: Programs for machine learning*, San Francisco: Morgan Kaufmann.

Reis, Elizabeth, (2001), *Estatística Multivariada Aplicada*, Lisboa: Edições Sílabo.

Resnick, P. e Varian, H. (1997), “Recommender Systems”. Communications of ACM, Vol. 40, No. 3, March 1997.

Ripley, B. D. (2001), *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press.

Sarwar, B., Karypis, G., Konstan, J. e Reid, J. (2000), “Analysis of Recommendation Algorithms for E-Commerce”. In Proceedings of the ACM EC'00 Conference. Minneapolis, MN. pp. 158-167.

Sarwar, B., Karypis, G., Konstan, J. e Reid, J. (2001), “Item-based Collaborative Filtering Recommendation Algorithms”. Appears in WWW10, May 1-5, 2001, Hong Kong.

Savasere, A., Omiecinski, E., e Navathe, S. (1995), “An efficient algorithm for mining association rules in large databases”. In Proc. of the 21th Int’l Conf. On Very Large Databases (VLDB’95), Zurich, Switzerland, September 1995.

Sharma, Subhash (1996), *Applied Multivariate Techniques*, New York: John Wiley & Sons, Inc.

Srikant, R. e Agrawal, R. (1996), “Mining Quantitative Association Rules in Large Relational Tables”. ACM SIGMOD 96.

Srikant, R., Vu, Q., e Agrawal, R. (1997), “Mining Association Rules with Item Constraints”. Proc. 3th Int’l Conf. Knowledge Discovery and Data Mining KDD 97.

Tan, P-N., Kumar V. (2000), “Interestness Measures for Association Patterns: A Perspective”. KDD 2000 Workshop on Postprocessing in Machine Learning and Data Mining.

Toivonen H. (1996), “Sampling Large Databases for Association Rules”. Proceedings of the 22nd VLDB Conference, 1996.

Torgo, L. (2002), *Data Mining with R: learning by case studies*. Consultado a 8 de Abril de 2003 em <http://www.liacc.up.pt/~ltorgo/DataMiningWithR/>.

Tsai, P., e Cheng, C. (2001), “Mining Quantitative Association Rules in a Large Database of Sales Transactions”. Journal of Information Science and Engineering. 2001.

Turban, Efraim e Aronson, Jay E. (2001), *Decision Support Systems and Intelligent Systems*. New Jersey: Prentice Hall.

van Rijsbergen, C. A. (1979), *Information retrieval*. London: Butterworths.

Venables, W. N. e Ripley, B. D. (1999), *Modern Applied Statistics with S-Plus. Third Edition*. New York: Springer.

Venables, W. N. e Ripley, B. D. (2000), *S Programming*. New York: Springer.

Vicente, P., Reis, E., Ferrão, F. (2001), *Sondagens – A amostragem como factor decisivo da qualidade*. Lisboa: Edições Sílabo.

Wang, K., He, Y. e Han, J. (2000)a, “Mining Frequent Itemsets Using Support Constraints”. VLDB Jornal 2000.

Wang, K., Zhou, S. e He, Y. (2000)b, “Growing Decision Trees On Support-Less Association Rules”. KDD 2000.

Wei, Y. Z., Moreau, L. and Jennings, N. R. (2003), “Recommender Systems: A Market-Based Design”. Proc. 2nd International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS03).

Westerman, Paul (2001), *Data Warehousing Using the Wal-Mart Model*, San Francisco: Morgan Kaufmann.

Wettshereck, D., (2002), “A KDDSE-independent PMML Visualizer”, in Proc. of IDDM-02, workshop on Integration aspects of Decision Support and Data Mining, (Eds) Bohanec, M., Mladenic, D., Lavrac, N., associated to the conference ECML/PKDD 02, Helsinki, Finland, 2002.

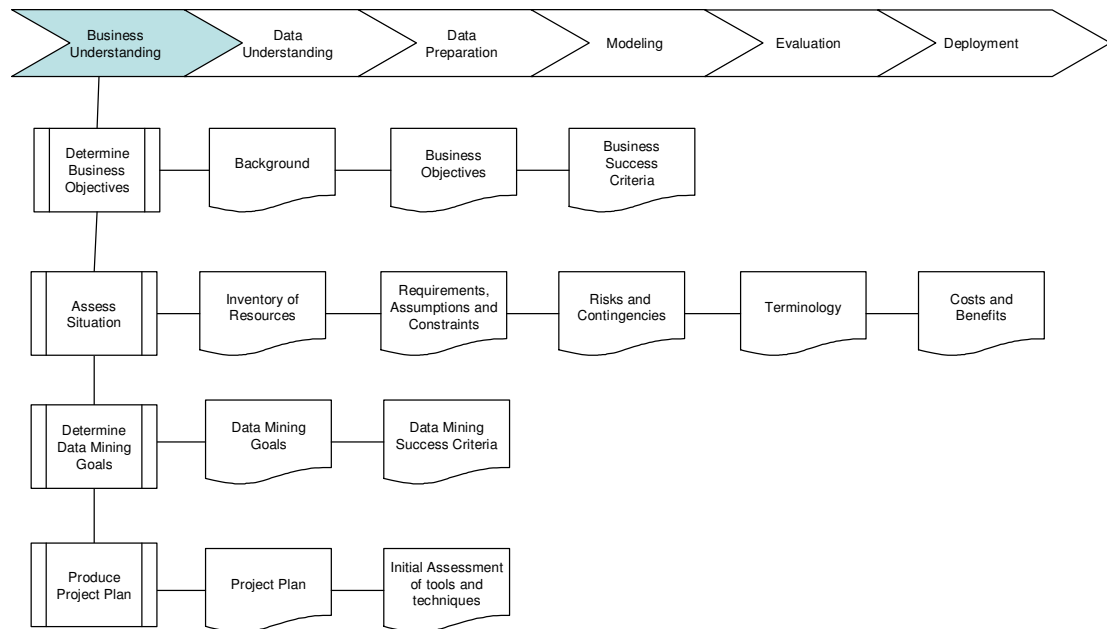
Witten, Ian H. e Frank, Eibe (2000), *Data Mining*, San Francisco: Morgan Kaufmann.

Zaki, M. J., Parthasarathy, S., Ogihara, M. e Li, W. (1997)a, “New Algorithms for Fast Discovery of Association Rules”. 3th Int’l Conf. KDD97.

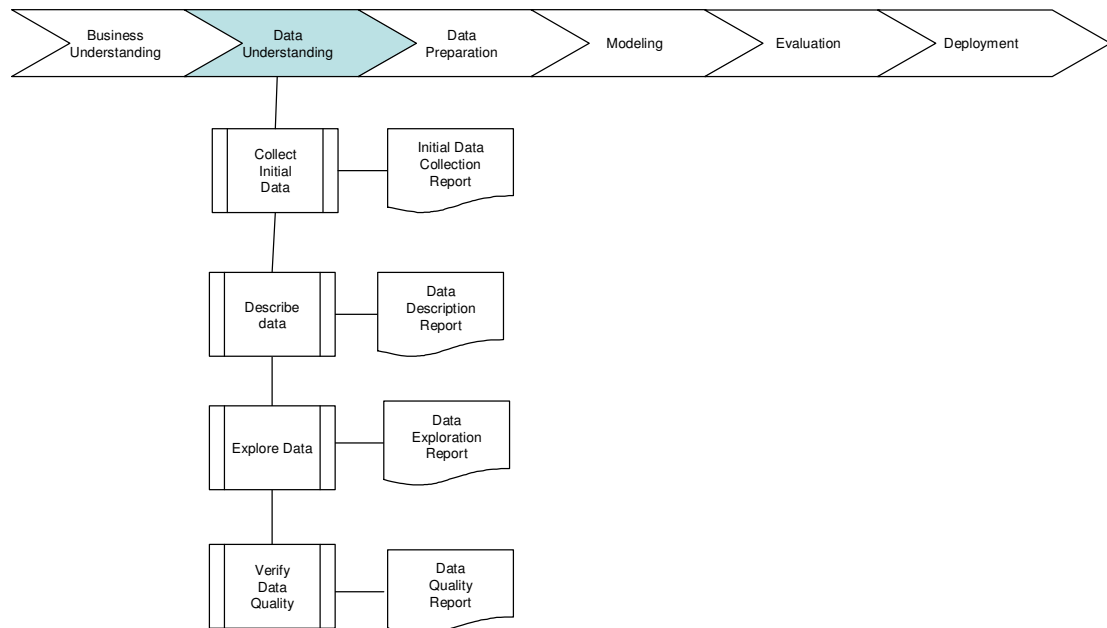
Zaki, M. J., Parthasarathy, S., e Ogihara, M. (1997)b, “Parallel Algorithms for Discovery of Association Rules”. Data Mining and Knowledge Discovery 1997.

Anexo 1 Síntese da Metodologia CRISP-DM

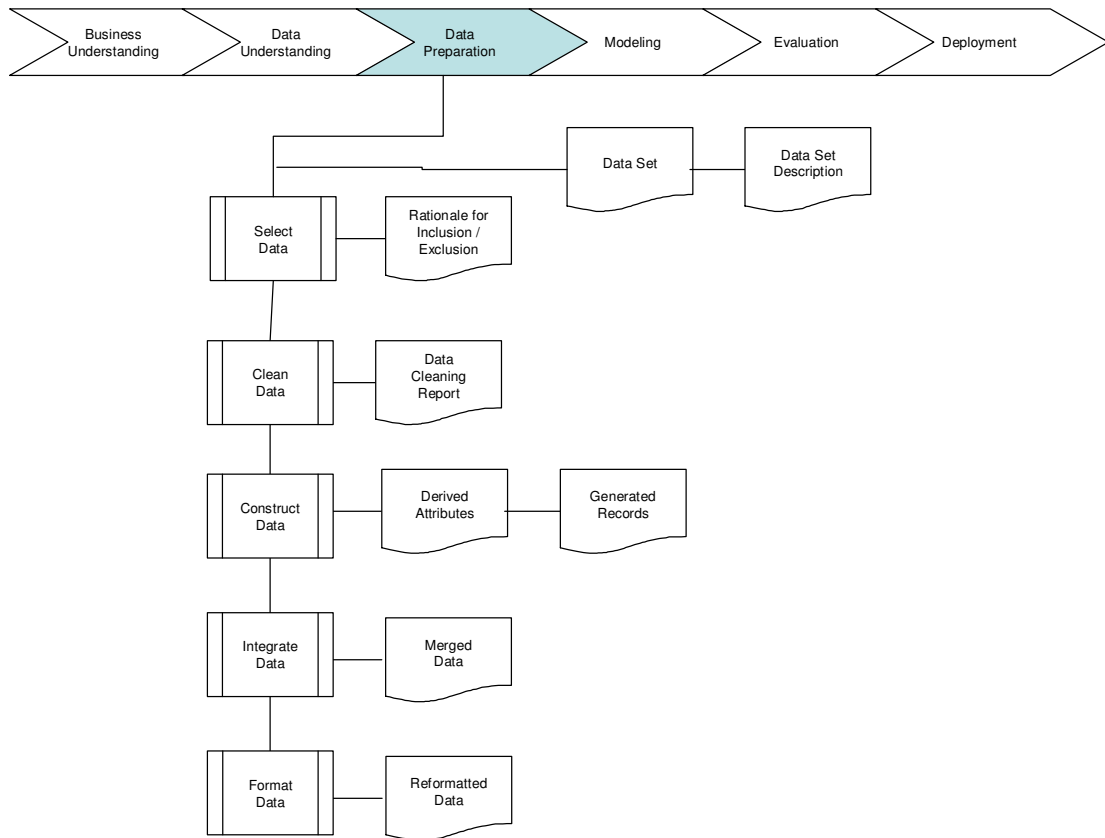
Business Understanding



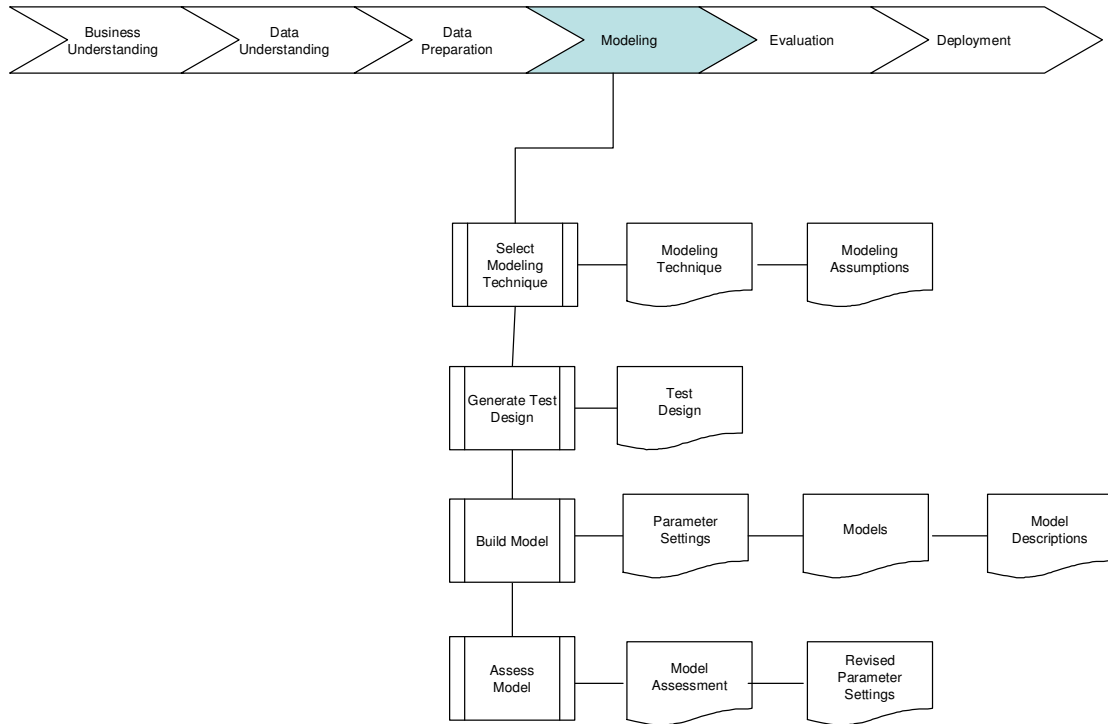
Data Understanding



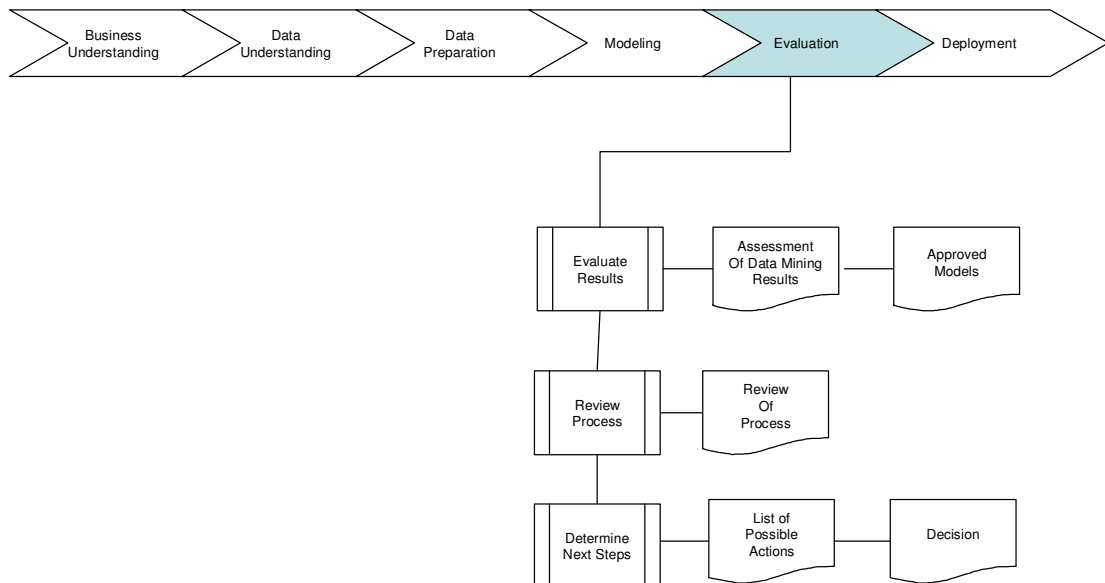
Data Preparation



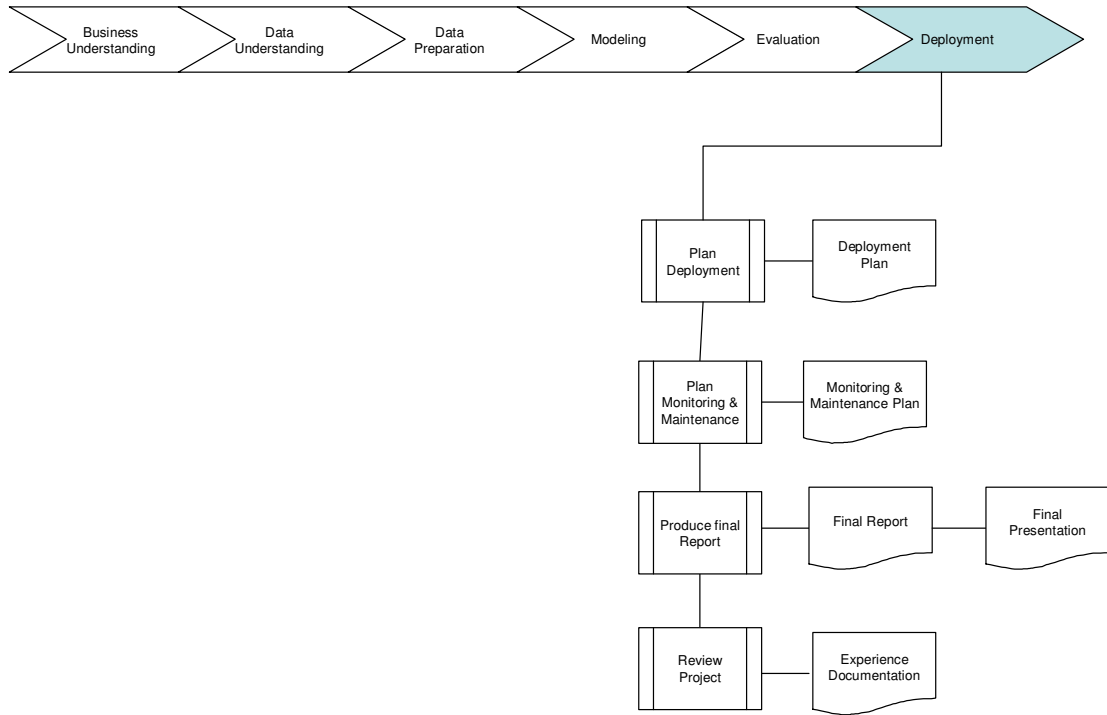
Modeling



Evaluation



Deployment



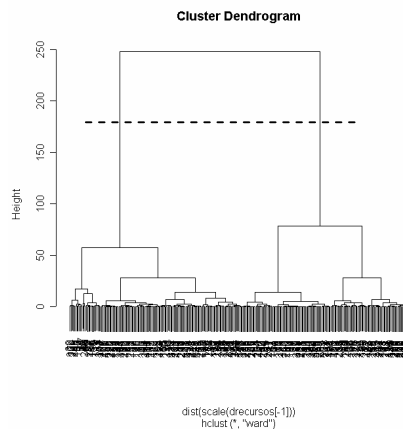
Anexo 2 Análises Multivariadas

Para aprofundarmos um pouco mais o conhecimento sobre estes dados, a informação relacionada com os recursos e com os projectos foi compilada em dois grandes *data sets* distintos, para tratamento posterior. O primeiro *data set* é constituído pelas variáveis: Dias por Recurso, Cestos por Recurso, Projectos por Recurso e Minutos por Recurso; o segundo *data set* contém as variáveis: Dias por Projecto, Cestos por Projecto, Recursos por Projecto e Minutos por Projecto.

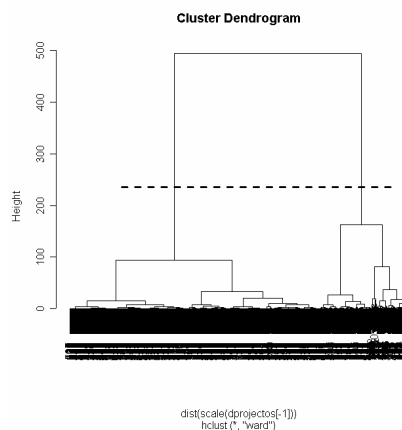
A motivação inicial foi saber se era possível encontrar *clusters* de recursos e de projectos. Para atingir este objectivo foi efectuada uma análise classificatória hierárquica, utilizando o índice de *ward* [Sharma, Subhash (1996)] e [Reis, Elizabeth (2001)] a ambos *data sets*. Este tipo de técnica permite agrupar dentro de grupos muito diferentes entre si, os casos que apresentam as maiores semelhanças possíveis. O segundo passo foi tentar explorar as correlações que existem nos dados, e avançar para uma análise de componentes principais [Sharma, Subhash (1996)] e [Reis, Elizabeth (2001)], também aos dois *data sets*. Por fim, houve o interesse de combinar o conhecimento extraído por todas estas técnicas, para uma interpretação mais aprofundada.

Análise Classificatória Hierárquica

Em relação aos recursos, o *dendograma* seguinte permite identificar (com ajuda da linha a tracejado) dois *clusters*, cada um com, respectivamente, 143 e 147 recursos, sendo que o primeiro representa os recursos com menos dias, com menos cestos, com menos projectos e com menos minutos (valores médios); em oposição, o segundo cluster representa os recursos com valores médios mais elevados para as mesmas variáveis.



Ao efectuar a mesma operação em relação aos projectos, conseguimos também identificar, através do próximo *dendograma*, dois *clusters* nos dados. Cada *cluster* de projectos tem, respectivamente, 600 e 212 projectos, sendo que o primeiro *cluster* representa os projectos com menos dias, com menos cestos, com menos recursos e com menos minutos (valores médios); do modo contrário, o outro *cluster* representa os projectos com valores médios mais elevados nas mesmas variáveis.



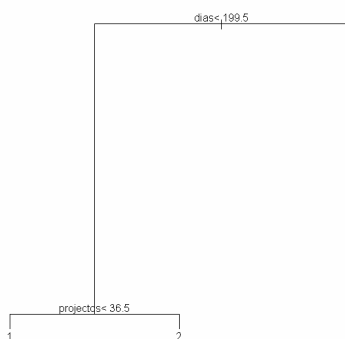
A conclusão que podemos retirar desta primeira análise é que os recursos podem ser divididos em dois grandes grupos: recursos com uma actividade mais intensiva em projectos (mais dias, mais custos, mais projectos e mais minutos) – tipicamente estas características enquadram-se em recursos da área de *delivery* tais como, por exemplo: “*Quesado, João F.*”, “*Loja, Luísa R.*”, “*Ribeiro, Susana C.*”; e recursos com uma actividade menos intensiva em projectos (menos dias, menos custos, menos projectos e menos minutos). Neste último grupo estão incluídos os recursos que têm uma actividade mais intensiva noutras tarefas da empresa, tais como, por exemplo, actividades relacionadas com a administração da empresa, secretariado, ou gestão de conta. Exemplos: “*Brás, Jorge S.*”, “*Bessa, Adelina L.*”, “*Gomes, Sílvia L.*”.

De igual modo, os projectos podem ser divididos em dois grandes grupos: projectos de grande dimensão (mais dias, mais custos, mais recursos e mais minutos) conforme, por exemplo: *712M*, *208B*, *712K*; e projectos de dimensão mais pequena (menos dias, menos custos, menos recursos e menos minutos). Exemplos: *719O*, *301N*, *P_108D*.

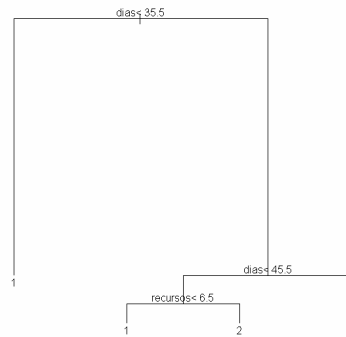
Classificação

Para entender de que forma é que estes *clusters* são formados, é possível efectuar uma análise discriminante [Sharma, Subhash (1996)] e [Reis, Elizabeth (2001)] a estes dados, ou pode-se construir uma árvore de decisão para este efeito [Venables, W. N. et al (1999)] e [Witten, Ian H. et al. (2000)]. Com estas técnicas, para além de se conseguir entender de que forma é que estes grupos foram formados, consegue-se igualmente classificar novos recursos e novos projectos nas classes formadas pelos *clusters* identificados anteriormente. Uma vez que estas variáveis não evidenciam (verificado em 4.3.3 – *Análises Exploratórias*) uma aproximação estatisticamente significativa à distribuição normal (pressuposto da análise discriminante linear), optou-se por construir uma árvore de decisão (método não paramétrico).

A árvore de decisão seguinte permite identificar que as variáveis que influenciam a formação de *clusters* de recursos são “dias” e “projectos”. Ou seja, se um recurso tiver mais do que 199,5 dias, então é classificado no *cluster* dos recursos com a actividade mais intensiva (*cluster 2*). Caso contrário, se tiver mais do que 36,5 projectos, então também é classificado neste *cluster*. Um recurso é classificado no *cluster* que corresponde aos recursos com uma actividade menos intensiva (*cluster 1*), se tiver menos do que 195,5 dias e se tiver menos do que 36,5 projectos.



Em relação aos projectos, a árvore de decisão seguinte permite identificar, do modo equivalente, que são os “dias” e os “recursos” que influenciam a criação dos *clusters* de projectos. Se um projecto tiver menos do que 35,5 dias, é classificado no *cluster* dos projectos de dimensão mais pequena (*cluster 1*). Se um projecto tiver mais do que 45,5 dias, então é classificado no *cluster* dos projectos de maior dimensão (*cluster 2*). Caso um projecto tenha entre 35,5 e 45,5 dias, então é necessário verificar o número de recursos para o classificar. Nesta situação, se um projecto tiver menos do que 6,5 recursos é classificado no *cluster* dos projectos de menor dimensão (*cluster 1*); caso contrário, é classificado nos projectos de grande dimensão (*cluster 2*).



Perante estes resultados, conclui-se que é muito provável que existam fortes correlações nestes dados (mais dias implicam mais minutos, por exemplo), pelo que foi decidido avançar para uma análise de componentes principais. Esta análise permite estender e complementar o conhecimento que se conseguiu reter sobre estes dados, até agora. Por outro lado, esta análise permite reduzir a dimensão do problema [Sharma, Subhash (1996)] e [Reis, Elizabeth (2001)].

Análise de Componentes Principais

Recursos

Começando, uma vez mais, pelos recursos, o primeiro passo em direcção a esta análise foi determinar as correlações que efectivamente existem nos dados, através da seguinte matriz de correlações:

	dias	cestos	projectos	minutos
dias	1,0000000	0,8437427	0,4747585	0,9547502
cestos	0,8437427	1,0000000	0,7340959	0,7677498
projectos	0,4747585	0,7340959	1,0000000	0,3159184
minutos	0,9547502	0,7677498	0,3159184	1,0000000

Esta matriz permite confirmar, com efeito, que estes dados apresentam correlações positivas e fortes (à excepção de “dias” com “projectos”; e “projectos” com “minutos”). Assim sendo, estão reunidas as condições para avançar com a análise de componentes principais [Sharma, Subhash (1996)] e [Reis, Elizabeth (2001)]. O quadro que se segue mostra as quatro componentes extraídas a partir destes dados, de onde se pode concluir que as duas primeiras componentes (as que interessam reter) representam 96,90% da variância total:

	Comp.1	Comp.2	Comp.3	Comp.4
Standard deviation	1,7565065	0,8891359	0,30747373	0,1719942
Proportion of Variance	0,7713288	0,1976407	0,02363502	0,0073955
Cumulative Proportion	0,7713288	0,9689695	0,99260450	1,0000000

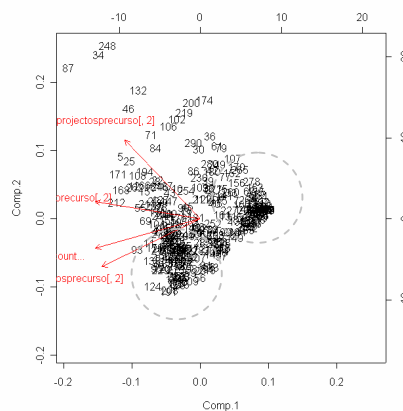
A tabela seguinte permite analisar os *loadings* (combinações lineares das variáveis originais para cada componente principal [Venables, W. N. et al. (1999)]) e as correlações entre as variáveis originais e as várias componentes principais:

	Correlações:				Loadings:			
	Comp.1	Comp.2	Comp.3	Comp.4	Comp.1	Comp.2	Comp.3	Comp.4
dias	-0,951	-0,265	0,293	0,089	-0,542	-0,298	0,317	0,719
cestos	-0,955	0,154	0,296	0,030	-0,544	0,173	-0,821	
projectos	-0,688	0,713	0,153	0,644	-0,392	0,802	0,425	-0,149
minutos	-0,891	-0,433	0,258	0,237	-0,508	-0,487	0,214	-0,678

A análise deste quadro permite concluir que a primeira componente é um factor tamanho (os *loadings* da primeira componente têm todos o mesmo sinal). Assim, é possível evidenciar e distinguir os recursos com valores elevados a todas as variáveis; dos recursos com valores mais baixos em todas as variáveis. As variáveis com maior peso na formação desta componente são os custos e os dias, ao que se seguem os minutos e, por fim, os projectos. No fundo, o que esta componente ajuda a distinguir são os recursos mais activos em projectos dos recursos menos activos em projectos.

A segunda componente opõe fundamentalmente os recursos com muitos projectos e poucos minutos, aos recursos com poucos projectos e muitos minutos. Dito de outra forma, esta componente separa os recursos que participam em poucos projectos, independentemente de terem um envolvimento elevado nestes; dos recursos que não têm um envolvimento tão elevado nos projectos em que participam, apesar de participarem em muitos projectos.

Um alternativa eficaz para analisar estas componentes é fazê-lo de forma gráfica [Venables, W. N. et al. (1999)]. O gráfico seguinte mostra as 4 componentes, e, simultaneamente, mostra também os recursos representados nas duas primeiras componentes.



Através deste gráfico, conseguimos identificar duas grandes “nuvens” de recursos (identificados pelos círculos a tracejado). Uma por cima do eixo dos *xx*, no seu lado positivo; e outra aproximadamente por cima do eixo dos *yy*, no seu lado negativo.

A primeira nuvem mostra um grande grupo de recursos bem representado na primeira componente, mais especificamente no seu lado positivo. Isto significa que os recursos que pertencem a este grupo têm (por ordem decrescente do peso de cada variável nesta componente) poucos custos, poucos dias, poucos minutos e poucos projectos. Tipicamente este perfil encaixa em recursos pouco activos em projectos: recursos da “*pool de suporte*”, dado que tem uma participação reduzida em (poucos) projectos; recursos que chegaram à empresa, ou que a abandonaram, durante o período em análise; e alguns tipos de recursos sub contratados. Esta intuição foi confirmada ao identificar alguns dos recursos presentes nesta nuvem (recursos 156, 278, 162 e 101, por exemplo).

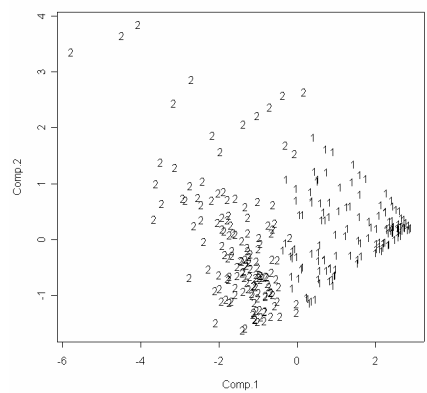
O segundo grupo, está bem representado na segunda componente – no seu lado negativo. Ou seja, os recursos que a ele pertencem têm poucos projectos e muitos minutos. Ainda que com menor peso, são também recursos com muitos dias e poucos custos (poucos custos porque trabalham em poucos projectos). É legítimo deduzir que estas características enquadram-se em recursos da “*pool de delivery*”, com funções de análise / desenvolvimento (poucos projectos, apesar do envolvimento nestes ser elevado). Esta dedução foi confirmada através da identificação de alguns recursos presentes nesta nuvem (recursos 124, 191 e 208, por exemplo).

É de esperar que os recursos que se encontram do lado oposto da primeira nuvem referida anteriormente, sejam gestores de projecto. Estes recursos são muito activos em projectos (muitos custos, muitos dias, muitos minutos e muitos projectos – considerando, obviamente os pesos destas variáveis na formação da primeira componente principal). Os recursos 69, 212 e 171, por exemplo, confirmaram esta expectativa.

Os *managers*, os gestores de centro de competência, os comerciais e os membros da administração, têm, em princípio, muitos projectos, apesar do seu envolvimento nestes

ser reduzido. Devem, portanto, estar bem representados na segunda componente, mais concretamente no seu lado positivo. Ao identificar os recursos 87, 34, 248, 132, 200, 174, 219, 71, 106 e 36, por exemplo, confirmou-se, uma vez mais, esta intuição.

Para finalizar a análise que se está a ser efectuada aos recursos da *Enabler*, a próxima proposta é ver, através do próximo gráfico, como os *clusters* de recursos identificados nos passos anteriores através da análise classificatória hierárquica, são representados por estas componentes principais.



A análise deste gráfico permite constatar que a primeira componente consegue discriminar de forma extremamente aceitável os dois *clusters*.

O próximo passo é então repetir estas análises para o *data set* dos projectos.

Projectos

A matriz de correlações seguinte permite confirmar que tipos de correlações existem entre as variáveis deste *data set*.

	dias	cestos	recursos	minutos
dias	1,0000000	1,0000000	0,5382805	0,7023301
cestos	1,0000000	1,0000000	0,5382805	0,7023301
recursos	0,5382805	0,5382805	1,0000000	0,6348114
minutos	0,7023301	0,7023301	0,6348114	1,0000000

Esta matriz permite confirmar de imediato (correlação perfeita = 1) o que já foi referido anteriormente em relação aos “dias por projecto” e aos “cestos por projecto”, isto é, representam exactamente a mesma coisa. Isto deve-se ao facto da chave dos cesto ser constituída pela data da actividade e pelo projecto.

As restantes variáveis apresentam correlações elevadas, estando, portanto, assim reunidas as condições para prosseguir com a análise de componentes principais. O quadro que se segue mostra as quatro componentes extraídas a partir destes dados, de onde se pode concluir que as duas primeiras componentes (as que interessam reter) representam 92,16% da variância total:

	Comp.1	Comp.2	Comp.3	Comp.4
Standard deviation	1,7541595	0,7806029	0,55998528	2,399334e-08
Proportion of Variance	0,7692689	0,1523352	0,07839588	1,439201e-16
Cumulative Proportion	0,7692689	0,9216041	1,00000000	1,000000e+00

A tabela seguinte permite analisar os *loadings* e as correlações entre as variáveis originais e as várias componentes principais:

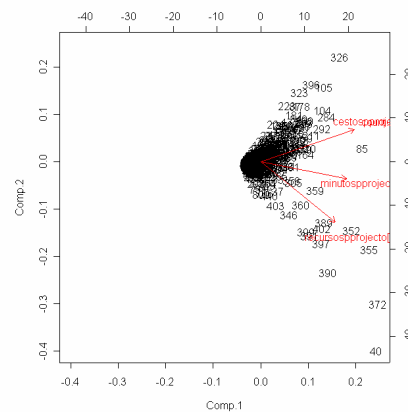
	Correlações:				Loadings:			
	Comp.1	Comp.2	Comp.3	Comp.4	Comp.1	Comp.2	Comp.3	Comp.4
dias	0,939	0,323	0,287	0,172	0.536	0.414	-0.204	0.707
cestos	0,939	0,323	0,287	0,172	0.536	0.414	-0.204	-0.707
recursos	0,751	-0,608	0,183	0,607	0.428	-0.779	-0.457	
minutos	0,864	-0,174	0,243	0,050	0.493	-0.223	0.841	

A interpretação destas componentes é então a seguinte:

A primeira componente é, também neste caso, um factor tamanho – os seus *loadings* têm todos o mesmo sinal. O que significa que esta componente opõe projectos de grande dimensão (muitos dias, muitos minutos e muitos recursos) a projectos de dimensão mais reduzida (poucos dias, poucos minutos e poucos recursos).

A segunda componente, basicamente, identifica os projectos com muitos recursos, ao destacá-los dos projectos que têm poucos recursos.

O gráfico seguinte mostra as 4 componentes, e, simultaneamente, mostra também os recursos representados nas duas primeiras componentes.

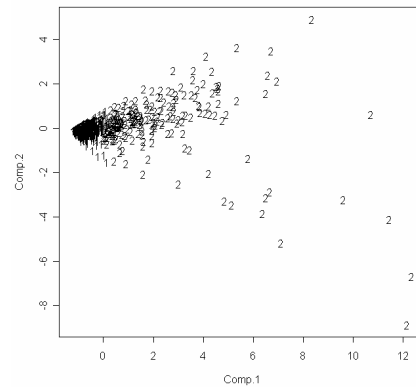


A grande “nuvem” de projectos que aparece no centro do eixo das coordenadas, significa que existe um número elevado de projectos mal representados por estas duas componentes.

É possível identificar dois projectos perto do eixo dos xx e com um afastamento significativo em relação à origem. Estes projectos são o 85 e 359. Estes projectos estão bem representados na primeira componente, e, de acordo com a sua interpretação efectuada anteriormente, estes projectos são, portanto, de grande dimensão. Apesar de estarem mais afastados do eixo dos xx (e por isso terem uma pior representação na primeira componente), conseguem-se identificar outros projectos, cuja projecção neste eixo está também significativamente afastada da origem, sendo, então, projectos de grande dimensão. Estes projectos são: 105, 326, 284, 389, 402, 397, 390, 352, 355, 372 e 40.

Em extremos opostos do eixo dos yy , situam-se os projectos 40 e 325. Pela interpretação efectuada sobre a segunda componente, este facto significa que o primeiro projecto tem muitos recursos e o segundo poucos recursos. Seguindo este raciocínio, identifica-se igualmente os projectos 397, 390, 355 352 e 372, como sendo projectos com muitos recursos; e os projectos 323, 396 e 105, como sendo projectos com poucos recursos.

Da mesma forma como fechamos a análise sobre os recursos, o gráfico seguinte mostra de que forma é que os *clusters* de projectos, identificados através da análise classificatória hierárquica efectuada num dos passos anteriores, são representados pelas duas primeiras componentes principais:



Este gráfico permite concluir que a primeira componente consegue discriminar de forma satisfatória os dois *clusters*.

Anexo 3 *Caren*

O *Caren* funciona a partir da linha de comando, e os parâmetros principais de execução que foram utilizados são:

```
C:\>java aprioribas DatasetName minsup minconf -s; -ocsRulesFile
```

O “*DatasetName*” é o ficheiro que contém os cestos; “*minsup*” e “*minconf*” são, respectivamente, o suporte mínimo e a confiança mínima; a opção “-s” define qual o separador entre o identificador dos cestos e o item (neste caso foi “;”); e a opção “-o” configura o formato do resultado, ou seja a lista final de regras, e o nome do ficheiro que o irá conter - neste caso o formato de saída escolhido foi o “cs” – *comma separator*.

O ficheiro que contém o resultado final com este formato, é composto pelos seguintes atributos (para cada regra gerada): o suporte; a confiança; o consequente; e o conjunto dos *itens* que fazem parte do antecedente, sendo que cada um destes é separado dos demais por: “ & “.

Anexo 4 Programas em R

```
> f.cria.cestos
function(db, t, k, i){

#Cria cestos do tipo chave transacção (k), item (i), a partir de uma tabela (t) de uma
base de dados (db) de transacções.

sqlQuery(db, paste ("select concat(", paste(k, collapse=", "), "), ", paste(i,
collapse=", "), " from", t))

}

> f.indices.treino
function(d, t) {

# Dá os índices dos dados de treino

sample(1:nrow(d), as.integer(t*nrow(d)))

}

> f.indices.treino.tran
function(d, t) {

# Dá os índices dos dados de treino, a partir de um data frame (d) com transacções.

f.indices.treino(as.data.frame(levels(d[[1]])), t)

}

> f.ler.db
function(db, t){

# Lê a base de dados

sqlQuery (db, paste ("select * from", t))

}

> f.teste
function(df, i) {

# A partir do data frame (df) com as transacções, dá o conjunto de treino

df[is.element(df[[1]], as.data.frame(levels(df[[1]]))[-i,]),]

}

> f.treino
function(df, i) {

# A partir do data frame (df) com as transacções, dá o conjunto de treino

df[is.element(df[[1]], as.data.frame(levels(df[[1]]))[i,]),]

}

> f.hidden.sample
function(d) {

d[sample(1:nrow(d),1),1]
```

```

}

> f.prepare_to_evaluate
function () {

# Função para "fazer tudo de uma vez" ...

library("RODBC")
madsad.DB<-odbcConnect("madsadodb")
cestos<-f.cria.cestos(madsad.DB, "cestos", c("project_code", "cell_date"),
"resource_name")
i.treino<-f.indices.treino.tran(cestos,0.8)
treino<-f.treino(cestos,i.treino)
teste<-f.teste(cestos,i.treino)
i.hidden<-by(cbind(1:nrow(teste),teste), teste[,1][,drop=T],f.hidden.sample)
hidden<-teste[i.hidden,]
observable<-teste[-i.hidden,]

}

> f.antecedentes.string
function(a)
{

# Dada uma strig de itens separados por " & ", devolve um vector de itens

i<-regexpr("&",a[1])[1]
if (i==1) a[1]
else c(substr(a[1], 1, i-3), f.antecedentes.string(substr(a[1], i+3, nchar(a[1]))))

}

>f.regra.contida
function(m,o) {

# Verifica se o antecedente da regra m está contida no conjunto observável o

!(is.element(F,is.element(f.antecedentes.string(as.character(m[[4]])),o)))

}

> f.recomendar
function(o,m,N) {

# Dado um conjunto observavel de itens, um modelo de regras e um valor N, devolve as
recomendações

m<-m[!(is.element(m[[3]],o)),] #retirar de m todas as regras cujo conseqüente pertence a
"o"
m<-m[apply(m,1,f.regra.contida,o),] #"m" passa a conter apenas as regras cujo
antecedente está contido em "o" e cujo conseqüente não pertence a "o"

m<-m[,-4]

##retirar todas as regras cujo conseqüente é repetido....
i<-1
while(i<nrow(m))
if (m[(i+1),3]==m[i,3])
m<-m[-(i+1),]
else
i<-i+1
}
##---

if (nrow(m)>1) {
##-- Buble Sort
flag<-T
while (flag) {

```

```

    flag<-F
    for (i in 1:(nrow(m) -1))
      if (m[i,2]<m[i+1,2]) {
        aux<-m[i,]
        m[i,]<-m[i+1,]
        m[i+1,]<-aux
        flag<-T
      }
    }
  ##
}

if (nrow(m)<N) N<-nrow(m)
m<-m[1:N,]

#levels(m[is.element(m[[2]],s),][[3]][,drop=T])

levels(m[[3]][,drop=T])

}

>f.item.observable
function(o) {
  as.character(o[[2]][,drop=T])
}

>f.intersect
function(k,h,r) {

  if (length(r[names(r)==k])!=0) intersect( as.character( h[ h[[1]] == k,][[2]]),
  r[names(r) == k][[1]])

}

# Instruções p/ avaliar o modelo

# observable vem da função prepare_to_evaluate mostrada em cima
l.observable<-by(observable,observable[[1]][,drop=T],f.item.observable)

#modelo_regras é o objecto do R que contém o conjunto de regras de associação. N é o
numero de recomendações que se pretende.
rec<-lapply(l.observable,f.recomendar,modelo_regras,N)
rec2<-rec[rec!="character(0)"]

intersect.h.r<-apply(as.array(names(rec2)),1,f.intersect,hidden,rec2)

m.intersect.h.r<-length(intersect.h.r[intersect.h.r!="character(0)"]) # "m" significa:
"módulo"

m.hidden<-nrow(hidden)

recall<-m.intersect.h.r/m.hidden

m.rec<-sum(sapply(rec2,length))

precision<-m.intersect.h.r/m.rec

f1<-(2*recall*precision)/(recall+precision)

> f.recomendar.recurso3
function(o,m,N,r) {

  # Dado um conjunto observável de itens, um modelo de regras e um valor N, devolve as N
primeiras recomendações com os suportes e confianças respectivos. R é o Data Frame com
os recursos, a sua pool, e o seu nível

m<-m[!(is.element(m[[3]],o)),] #retirar de m todas as regras cujo consequente pertence a
"o"
m<-m[apply(m,1,f.regra.contida,o),] # "m" passa a conter apenas as regras cujo
antecedente está contido em "o" e cujo consequente não pertence a "o"

```

```

if (nrow(m)>0) {
  m<-m[,-4] #retirar o antecedente das regras do modelo m

  #retirar todas as regras cujo consequente é repetido...; obtem-se melhores valores para
  as medidas testadas (Recall, Precision e F1).
  i<-1
  while (i<nrow(m))
    if (m[(i+1),3]==m[i,3])
      m<-m[-(i+1),]
    else
      i<-i+1
  #---

  if (nrow(m)>1) {
    ##-- Buble Sort
    flag<-T
    while (flag) {
      flag<-F
      for (i in 1:(nrow(m) -1))
        if (m[i,2]<m[i+1,2]) {
          aux<-m[i,]
          m[i,]<-m[i+1,]
          m[i+1,]<-aux
          flag<-T
        }
      }
    }
    ##--

    if (nrow(m)<N) N<-nrow(m)
    m<-m[1:N,]
    rec<-cbind(m[1,-3],r[1,])
    rec<-rec[-1,]
    for (i in 1:N) {
      rec.temp<-cbind(m[i,-3],r[r[[1]]==as.vector(m[i,3]),])
      rec<-rbind(rec,rec.temp)
    }

    rec
  }

}

f.calcula.interest(t, r) {

  # Dado um data frame "t" com cestos e um modelo "r" de regras, devolve um data frame de
  regras com o Interest calculado

  interest<-0
  sup1<-summary(t[[2]][drop=T], maxsum=length(levels(t[[2]][drop=T])))
  sup1<-sup1/(length(levels(t[[1]][drop=T])))
  for (i in 1:nrow(r)) {
    interest[i]<-r[i,2]/sup1[names(sup1)==r[i,3]]
  }

  interest
}

f.contido(a, b) {

  flag<-TRUE
  contido<-TRUE
  i<-1
  while (flag) {
    if (!is.element(a[i], b)) {
      flag<-FALSE
      contido<-FALSE
    }
    else {
      i<-i+1
    }
  }
}

```



```

    if (i>length(a)) flag<-FALSE
  }
}
contido

}

f.suport.count(is, custos.lista) {

  f.conta<-function(lista, is) {

    i<-0
    if (f.contido(is, lista)) i<-1
    i

  }

  suporte.count<-lapply(custos.lista, f.conta, is)
  sum(suporte.count)

}

> f.recomendar.equipa
function(e,m,r) {

# Dado uma equipa de tamanho n, testa previsões de recomendações para equipas de tamanho
(n-1). A recomendação com suporte mais elevado constituirá, com os restantes (n-1)
elementos, a equipa recomendada.

  rec<-cbind(m[1,-(3:4)],r[1,])
  rec<-rec[-1,]
  equipa.pool.nivel<-r[1,]
  equipa.pool.nivel<-equipa.pool.nivel[-1,]

  for (i in 1:length(e)) {
    equipa.pool.nivel.temp<-r[r[[1]]==e[i],]
    equipa.pool.nivel<-rbind(equipa.pool.nivel,equipa.pool.nivel.temp)
  }
  for (i in 1:length(e)) {
    rec.temp<-f.recomendar.recurso3(e[-i],m,50,r)
    print(rec.temp)
    if (is.null(rec.temp)) {
      }
    else {
      if (nrow(rec.temp)>0) {
        j<-1
        flag=T
        while (flag) {
          if ((equipa.pool.nivel[i,2]==rec.temp[j,4])
(equipa.pool.nivel[i,3]==rec.temp[j,5])) {
            flag<-F
            rec[i,]<-rec.temp[j,]
          }
          else {
            j<-j+1
            if (j>nrow(rec.temp)) flag<-F
          }
        }
      }
    }
  }
  print(rec)
  if(nrow(rec)>0) {
    if (is.na(rec[1,2])) max<-0 else max<-rec[1,2]
    i.max<-1
    for (i in 1:nrow(rec))
      if (!is.na(rec[i,2]))
        if (rec[i,2]>max) {
          max<-rec[i,2]
          i.max<-i
        }
  }
}

```

```
    c(e[-i.max],as.vector(rec[i.max,3]))
  }
  else
    cat("Não é possível recomendar outra equipa")
}
```

Anexo 5 Questionário

Questionário

Introdução

No âmbito da minha tese de mestrado desenvolvi um protótipo de um motor de um sistema de recomendação de recursos / equipas, baseado em regras de associação construídas a partir dos dados históricos dos *time reports* do *Service Sphere*, cujo objectivo é auxiliar a tarefa de planeamento e constituição de equipas em projectos.

Neste enquadramento, solicito e agradeço desde já a sua cooperação no preenchimento deste questionário que visa medir a sua percepção em relação ao grau de **adequação** das recomendações produzidas por este sistema. Nesta acção serão necessários cerca de 10 minutos.

Este questionário está dividido em 3 partes, correspondendo cada uma delas a diferentes utilizações deste sistema:

1. *Recomendação de recursos* – dado um conjunto de recursos, este sistema irá recomendar um recurso adicional, indicando a probabilidade associada a esta recomendação.
2. *Recomendação de equipas* – dado um conjunto de recursos, este sistema irá recomendar um outro conjunto, do mesmo tamanho, substituindo um dos recursos que lhe pertencem por outro que considere mais apropriado.
3. *Construção interactiva de equipas* – a partir de um recurso inicial, o utilizador vai construindo uma equipa, adicionando uma recomendação – um recurso – em cada passo deste processo.

As respostas obtidas através deste questionário serão tratadas confidencialmente. Sendo assim, não serão identificadas no documento final as pessoas que forneceram a informação que está a ser aqui solicitada.

Grato pela cooperação,

Miguel Veloso.

Primeira Parte - Recomendação de Recursos

Nesta secção serão apresentadas 6 equipas de projecto, geradas de forma aleatória, às quais se aplicou o protótipo do sistema desenvolvido neste âmbito. Pede-se que exprima a sua percepção em relação ao grau de **adequação** das recomendações apresentadas. É igualmente pedido que refira quais seriam as suas recomendações, face às 6 equipas apresentadas.

Equipa nº 1:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Soares, Marco A.	DW/BI/EAI Resources	3
Teixeira, José M.	ERP/Back Office Resources	2
Cunha, Cristina P.	DW/BI/EAI Resources	1

Recomendação:

Recurso	Pool	Nível	Probabilidade
Oliveira, Vitor M.	ERP/Back Office Resources	2	43,27%

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria:_____

Equipa nº 2:

Recurso	Pool	Nível
Machado, Maria A.	Project Management / Consulting Resources	4
Ferreira, Paulo F.	DW/BI/EAI Resources	3
Braga, Martinho A.	ERP/Back Office Resources	2
Martins, Sérgio M.	E-commerce Resources	1

Recomendação:

Recurso	Pool	Nível	Probabilidade
Castro, Jorge L.	ERP/Back Office Resources	3	47,97%

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria:_____

Equipa nº 3:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Gaspar, Manuel M.	ERP/Back Office Resources	3

Recomendação:

Recurso	Pool	Nível	Probabilidade
Castro, Rui M.	ERP/Back Office Resources	2	33,6%

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria:_____

Equipa nº 4:

Recurso	Pool	Nível
Machado, Maria A.	Project Management / Consulting Resources	4
Rocha, Manuela C.	ERP/Back Office Resources	3

Recomendação:

Recurso	Pool	Nível	Probabilidade
Lebreiro, Carlos F.	ERP/Back Office Resources	2	32,35%

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria:_____

Equipa nº 5:

Recurso	Pool	Nível
Fernandes, José A.	Project Management / Consulting Resources	4
Carvalho, Maria L.	DW/BI/EAI Resources	3
Martins, Isabel M.	Quality & Testing	2
Portela, Hugo M.	DW/BI/EAI Resources	2
Morim, Elisa R.	E-commerce Resources	1
Oliveira, António I.	DW/BI/EAI Resources	1

Recomendação:

Recurso	Pool	Nível	Probabilidade
Ferreira, Paulo F.	DW/BI/EAI Resources	3	72,83%

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria:_____

Equipa nº 6:

Recurso	Pool	Nível
Torres, Mário N.	ERP/Back Office Resources	3
Soares, Maria E.	ERP/Back Office Resources	3
Carvalho, Nuno S.	E-commerce Resources	2

Recomendação:

Recurso	Pool	Nível	Probabilidade
Pinho, João A.	ERP/Back Office Resources	3	61,13%

Para esta recomendação, como classifica o seu grau de adequação:

Muito Inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria:_____

Segunda Parte – Recomendação de Equipas

Às equipas apresentadas na secção anterior foi aplicada a funcionalidade de recomendação / optimização de equipas, do sistema em estudo. De igual modo, pede-se que exprima a sua percepção em relação ao grau de **adequação** das recomendações apresentadas. É igualmente pedido que refira quais seriam as suas recomendações, face às 6 equipas apresentadas.

Equipa nº 1:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Soares, Marco A.	DW/BI/EAI Resources	3
Teixeira, José M.	ERP/Back Office Resources	2
Cunha, Cristina P.	DW/BI/EAI Resources	1

Recomendação:

Recurso	Pool	Nível
Soares, Marco A.	DW/BI/EAI Resources	3
Teixeira, José M.	ERP/Back Office Resources	2
Cunha, Cristina P.	DW/BI/EAI Resources	1
Quesado, João F.	Project Management / Consulting Resources	4

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria: _____

Equipa nº 2:

Recurso	Pool	Nível
Machado, Maria A.	Project Management / Consulting Resources	4
Ferreira, Paulo F.	DW/BI/EAI Resources	3
Braga, Martinho A.	ERP/Back Office Resources	2
Martins, Sérgio M.	E-commerce Resources	1

Recomendação:

Recurso	Pool	Nível
Machado, Maria A.	Project Management / Consulting Resources	4
Ferreira, Paulo F.	DW/BI/EAI Resources	3
Braga, Martinho A.	ERP/Back Office Resources	2
Sousa, Ricardo A.	E-commerce Resources	1

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria:_____

Equipa nº 3:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Gaspar, Manuel M.	ERP/Back Office Resources	3

Recomendação:

Recurso	Pool	Nível
Goes, Henrique M.	Project Management / Consulting Resources	4
Rocha, Manuela C.	ERP/Back Office Resources	3

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria:_____

Equipa nº 4:

Recurso	Pool	Nível
Machado, Maria A.	Project Management / Consulting Resources	4
Rocha, Manuela C.	ERP/Back Office Resources	3

Recomendação:

Recurso	Pool	Nível
Rocha, Manuela C.	ERP/Back Office Resources	3
Goes, Henrique M.	Project Management / Consulting Resources	4

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria:_____

Equipa nº 5:

Recurso	Pool	Nível
Fernandes, José A.	Project Management / Consulting Resources	4
Carvalho, Maria L.	DW/BI/EAI Resources	3
Martins, Isabel M.	Quality & Testing	2
Portela, Hugo M.	DW/BI/EAI Resources	2
Morim, Elisa R.	E-commerce Resources	1
Oliveira, António I.	DW/BI/EAI Resources	1

Recomendação:

Recurso	Pool	Nível
Fernandes, José A.	Project Management / Consulting Resources	4
Martins, Isabel M.	Quality & Testing	2
Portela, Hugo M.	DW/BI/EAI Resources	2
Morim, Elisa R.	E-commerce Resources	1
Oliveira, António I.	DW/BI/EAI Resources	1
Ferreira, Paulo F.	DW/BI/EAI Resources	3

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria: _____

Equipa nº 6:

Recurso	Pool	Nível
Torres, Mário N.	ERP/Back Office Resources	3
Soares, Maria E.	ERP/Back Office Resources	3
Carvalho, Nuno S.	E-commerce Resources	2

Recomendação:

Recurso	Pool	Nível
Torres, Mário N.	ERP/Back Office Resources	3
Carvalho, Nuno S.	E-commerce Resources	2
Pinho, João A.	ERP/Back Office Resources	3

Para esta recomendação, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Neste caso, a sua recomendação seria: _____

Terceira Parte – Construção Interactiva de Uma Equipa

Nesta fase será construída (passo a passo) uma equipa de trabalho com quatro recursos, utilizando para este efeito o sistema em estudo. Neste caso, este sistema será utilizado para “navegar” de forma interativa pelos recursos da Enabler. No final deste processo, pede-se que exprima a sua percepção em relação à **adequação** da equipa final formada.

Para iniciar o processo, foi gerado aleatoriamente um nome de um manager, ao que foi aplicada a funcionalidade de recomendação de recursos. O nome do manager assim gerado foi:

Recurso	Pool	Nível
Quesado, João F.	Project Management / Consulting Resources	4

Recomendações para [“Quesado, João F.”]:

Recurso	Pool	Nível	Probabilidade
Costa, Paula G.	ERP/Back Office Resources	3	24,85%
Oliveira, Vitor M.	ERP/Back Office Resources	2	21,30%
Teixeira, José M.	ERP/Back Office Resources	2	19,92%

Recurso escolhido: “Costa, Paula G.”

Recomendações para [“Quesado, João F.”, “Costa, Paula G.”]:

Recurso	Pool	Nível	Probabilidade
Oliveira, Vitor M.	ERP/Back Office Resources	2	25,07%
Teixeira, José M.	ERP/Back Office Resources	2	21,37%
Guerra, Pedro M.	ERP/Back Office Resources	1	16,62%
Ribeiro, José P.	DW/BI/EAI Resources	1	13,95%
Faria, Isabel M.	ERP/Back Office Resources	2	13,21%
Fernandes, José A.	Project Management / Consulting Resources	4	13,21%

Recurso escolhido: “Oliveira, Vitor M.”

Recomendações para [“Quesado, João F.”, “Costa, Paula G.”, “Oliveira, Vitor M.”]:

Recurso	Pool	Nível	Probabilidade
Teixeira, José M.	ERP/Back Office Resources	2	77,78%
Guerra, Pedro M.	ERP/Back Office Resources	1	16,62%
Ribeiro, José P.	DW/BI/EAI Resources	1	13,95%
Faria, Isabel M.	ERP/Back Office Resources	2	13,21%
Fernandes, José A.	Project Management / Consulting Resources	4	13,21%

Recurso escolhido: “Guerra, Pedro M.” (partindo do pressuposto que se pretende um recurso de nível 1 na equipa)

Equipa final:

Recurso	Pool	Nível
Quesado, João F.	Project Management / Consulting Resources	4
Costa, Paula G.	ERP/Back Office Resources	3
Oliveira, Vitor M.	ERP/Back Office Resources	2
Guerra, Pedro M.	ERP/Back Office Resources	1

Para esta equipa, como classifica o seu grau de adequação:

Muito inadequada	Inadequada	Nem adequada nem inadequada	Adequada	Muito Adequada
1	2	3	4	5

Como classifica a adequação do processo que conduziu à constituição desta equipa:

Muito inadequado	Inadequado	Nem adequado nem inadequado	Adequado	Muito Adequado
1	2	3	4	5

Comentários globais_____
