Universidade de Brasília
Departamento de Ciência da Computação

Um modelo baseado em inteligência artificial para a gestão do conhecimento aplicado ao processo de desenvolvimento de software

Sandro Carlos Vieira

Dissertação de mestrado

Orientador: Prof. Dr. Li Weigang

Co-orientador: Prof. Dr. Marcelo Ladeira

Universidade de Brasília
Departamento de Ciência da Computação

Um modelo baseado em inteligência artificial para a gestão do conhecimento aplicado ao processo de desenvolvimento de software

Sandro Carlos Vieira

Orientador: Prof. Dr. Li Weigang

Co-orientador: Prof. Dr. Marcelo Ladeira

Brasília (DF), 21 de julho de 2008

Catalogação da publicação

Vieira, Sandro Carlos.

Um modelo baseado em inteligência artificial para a gestão do conhecimento aplicado ao processo de desenvolvimento de software. / Sandro Carlos Vieira. Brasília, 2008.

146p;il;29,5cm

Dissertação de mestrado. Departamento de ciência da computação. Universidade de Brasília, Brasília.

1.Inteligência artificial. 2.Gestão do conhecimento. 3. Desenvolvimento de software.

CDU 004

Nominata

Universidade de Brasília – UnB

Reitor: Roberto Armando Ramos de Aguiar
Vice-reitor: José Carlos Balthazar

Departamento de ciência da computação – DCC

Chefe do departamento: Profa. Dra. Célia Ghedini Ralha
Coordenador de pós-graduação: Prof. Dr. Li Weigang

Sandro Carlos Vieira

Um modelo baseado em inteligência artificial para a gestão do conhecimento aplicado ao processo de desenvolvimento de software

Dissertação de mestrado submetida à avaliação
como requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação

Aprovado em:_____ / _____ / _____

Banca examinadora

_____

Li Weigang, Dr.
(Universidade de Brasília)
(Orientador)

_____

Paulo César Guerreiro da Costa, Dr.
(George Mason University)
(Examinador)

_____

Paulo Sérgio Cugnasca, Dr.
(Escola Politécnica - Universidade de São Paulo)
(Examinador)

Agradecimentos

Este trabalho começou muito antes de meu ingresso na Universidade de Brasília; na realidade ele começou antes mesmo que eu freqüentasse a primeira escola regular, com os ensinamentos que recebi de meu pai Osvaldo (in memoriam) e de minha mãe Maria Terezinha, ela que, entre outras coisas, me ensinou a ler e a escrever as primeiras palavras. Quero, portanto, dirigir a eles meus primeiros agradecimentos. Nesse primeiro núcleo familiar, tive a oportunidade de observar meu irmão Osvaldo trilhando os caminhos da educação apontados por meus pais, e esse exemplo que serviu, também, como inspiração para minhas realizações fica aqui registrado.

Ainda no núcleo familiar, registro meu sincero e carinhoso agradecimento à minha esposa Sandra e aos meus filhos Junior e Rafael que, compreendendo a importância deste trabalho, souberam me apoiar e repartir com os estudos o tempo que lhes caberia. Ao pequeno Arthur, que por estar presente há apenas três anos em nossas vidas ainda não compreendia essa mesma importância, fica também um agradecimento especial por me mostrar que nenhum tempo, por menor que seja, é insuficiente para ganhar um sorriso espontâneo e com ele a certeza de que também o dia foi ganho.

Aos professores do Departamento de Ciência da Computação da UnB registro também meu sincero agradecimento por todos os conhecimentos que compartilharam; entre esses gostaria de ressaltar os trabalhos do professor Dr. Li Weigang e do professor Dr. Marcelo Ladeira, que me orientaram ao longo de toda a pesquisa, tornando-a possível. São pessoas dedicadas e comprometidas como essas que, com o amor que demonstram ao ofício de ensinar, me permitem acreditar no futuro da educação deste país.

Agradeço, também, aos amigos e colegas do Banco do Brasil que, de várias formas, ajudaram na realização dessa pesquisa. Registro aqui um agradecimento especial aos amigos Anderson Nobre e Almir Delon, que viabilizaram a realização dessa pesquisa, possibilitando sua conciliação com os trabalhos do Banco e aos amigos Amarildo, Antônio Henrique, Isaac e Jaine que compartilharam comigo vários momentos dessa caminhada.

Ao alunos da graduação em ciência da computação da UnB André Ávila e Maurício Franceschini que participaram do desenvolvimento da ferramenta para avaliação e suporte ao modelo proposto, registro meu agradecimento e reconhecimento pela importante ajuda.

Por último, mas não menos importante, agradeço a Deus que, entre outras coisas, me deu o dom da vida e a oportunidade de compartilhá-la com todas essas pessoas.

Resumo

Desenvolver *software* é uma atividade comum para muitas empresas em diversas áreas da sociedade moderna, ainda que esse não seja o foco de seus negócios. Essa realidade se faz presente pela percepção de que a maioria dos produtos e serviços oferecidos é suportada por sistemas de computação, notadamente quando se fala de grandes empresas e de processos que envolvem valores vultuosos. No ramo financeiro, por exemplo, temos organizações, que dedicam alguns milhares de profissionais ao desenvolvimento de sistemas para a automação de suas atividades e à manutenção de outros que vêm sendo empregados há muito tempo. Nesse contexto, o processo de desenvolvimento de *software* têm sido objeto de constante preocupação por parte das administrações, porém a maioria dos trabalhos se volta ao controle de recursos e a melhoria e padronização dos processos atuais, negligenciando boa parte do conhecimento que está presente nesse ciclo. A presente pesquisa foca sua atenção sobre essa questão, com a proposição de um modelo, baseado em formalismos de inteligência artificial, que busca agregar mecanismos para a gestão do conhecimento envolvido no processo de desenvolvimento de *software*. O modelo aqui descrito foi construído e aplicado a um estudo de caso, em uma grande instituição financeira, obtendo-se resultados promissores quanto à sua utilização como forma de incrementar o reuso de soluções já desenvolvidas, evitar duplicidade de esforços na construção de soluções similares e também de propiciar uma alternativa eficaz para a rastreabilidade de características e detalhes sobre tais soluções.

Palavras-chaves: Inteligência artificial; gestão do conhecimento; engenharia de *software*; desenvolvimento de *software*.

Abstract

Software development has been a common activity to many companies in modern society, even that this activity are not in the core of its business. This reality is already evidenced by the perception that computer systems are strongly coupled to services and products, especially when big organizations and a great amount of money is involved. In the financial area, for example, many of this organizations employs over a thousand professionals to develop computer systems designed to support its business and to keep running software already in use for a long time. In this context, software development process has been considered as a critical mission by the organization's CIOs, but major efforts have been driven to adjust it into process patterns and resources control, keeping knowledge management outside this cycle. This research describes a model designed to improve knowledge management in software development process. The model described in this paper was constructed and applied as a case study in a financial organization, and the collected results illustrate the efficiency of the system to improve software reuse and to avoid duplicated efforts in solution design.

Keywords: Artificial intelligence; knowledge management; software engineering; software development.

Lista de figuras

Lista de fórmulas

Lista de tabelas

SUMÁRIO

1 Introdução

Este capítulo descreve o problema abordado na presente pesquisa, identificando o contexto atual em que está inserido o processo de desenvolvimento de *software* -mais especificamente o *software* destinado a execução em *mainframes*-, suas perspectivas e as soluções já tentadas. Descreve, também, os objetivos desta pesquisa, as linhas gerais adotadas durante sua condução e as contribuições obtidas.

1.1 Definição do problema

1.1.1 Situação atual

O desenvolvimento de *software*, embora não seja a atividade fim para a maioria das empresas, está de tal forma ligado aos negócios, que grande parte das organizações dedica parte de seus recursos a atividades relacionadas à área de tecnologia. Essa situação decorre da necessidade premente de desenvolver produtos com um ciclo de vida cada vez menor, aliada à exigência de rigorosos controles para que se mantenha a competitividade frente à concorrência.

Em grandes empresas, essa realidade vem sendo vivenciada há vários anos, fazendo com que grande parte das aplicações hoje em uso sejam originadas de projetos desenvolvidos há muito tempo. Essa situação é agravada, ainda, pela inexistência de documentação aderente aos aplicativos em uso, pela grande quantidade de intervenções (manutenções) realizadas de maneira emergencial e pela utilização de linguagens que não propiciam ao desenvolvedor a possibilidade de controlar e gerenciar a evolução dos aplicativos.

Grande parte dos principais sistemas utilizados por essas organizações padece, ainda, desse mal; são, em geral, sistemas antigos, com alto índice de manutenção, documentação inadequada ou inexistente, e a necessidade freqüente de interagir com outros sistemas de concepção diferente. Grande parte desses sistemas corporativos é escrita em linguagem Cobol, ou similares, e são executados diariamente em *mainframes* instalados em centros de processamento de dados de imensa capacidade. Sua execução ocorre em rotinas normalmente chamadas de *Batch*, pela característica de processamento em lotes. Também é comum esses sistemas corporativos contarem com uma interface baseada no padrão estabelecido pelos terminais 3270 (com evoluções, tal como o uso de cores) mas presa à restrição da apresentação de textos em uma área formada por 25 linhas e 80 colunas.

O fim dos *mainframes* e dos sistemas escritos em Cobol vem sendo há muito tempo

preconizado por várias correntes de especialistas em informática. Muitos chegavam a dizer, na década de 90, que os *mainframes* seriam rapidamente substituídos por computadores de menor porte cuja capacidade de processamento vem aumentando progressivamente desde seu lançamento. No entanto, se a capacidade de processamento vem crescendo a cada dia, o mesmo ocorre com a necessidade de processamento; o volume de transações processadas a cada dia tem crescido na mesma proporção, notadamente por empresas do ramo financeiro -especialmente bancos e administradoras de cartão de crédito-, fazendo com que os *mainframes* continuem sendo a escolha de boa parte dessas instituições.

O mesmo fenômeno se repete em relação à linguagem Cobol; embora muitos especialistas venham afirmando, desde a década de 1990, o seu fim iminente, muitas organizações ainda mantém em seus CPDs boa parte das transações suportadas por sistemas escritos nessa linguagem. E não se trata apenas de sistemas legados, como se poderia supor; um artigo publicado na revista ComputerWorld (2006) apresenta o resultado de uma pesquisa com executivos de empresas americanas que aponta resultados na direção contraria da opinião desses especialistas: dos 362 gerentes da área de Tecnologia da Informação (TI) que responderam a pesquisa, 62% usam o Cobol em seus ambientes de processamento, e 52% afirmaram que ainda fazem uso dessa tecnologia em novos produtos que estão sendo desenvolvidos.

Esse fenômeno se justifica pela convicção de muitos especialistas e executivos da área de TI de que a performance de sistemas escritos em Cobol, executados em *mainframes* ainda não pode ser alcançada em outras plataformas. Junte-se a isso os pesados investimentos que precisariam ser feitos para migrar as mais de 10 bilhões de linhas de código Cobol que se estimam estar em produção na atualidade e a preocupação quanto à confiabilidade necessária para substituir aplicações que movimentam cifras na casa dos bilhões de Reais diariamente - somente o Banco do Brasil processa aproximadamente 1,7 bilhão de transações por mês, com valor médio superior a R$30,00 (ComputerWorld, 2007) e é possível supor que essa realidade tende a permanecer por muito tempo em vários centros de processamento de dados em todo o mundo.

Os gráficos mostrados nas figuras 1.1 e 1.2 apresentam a situação das organizações de TI, conforme os resultados da pesquisa citada:

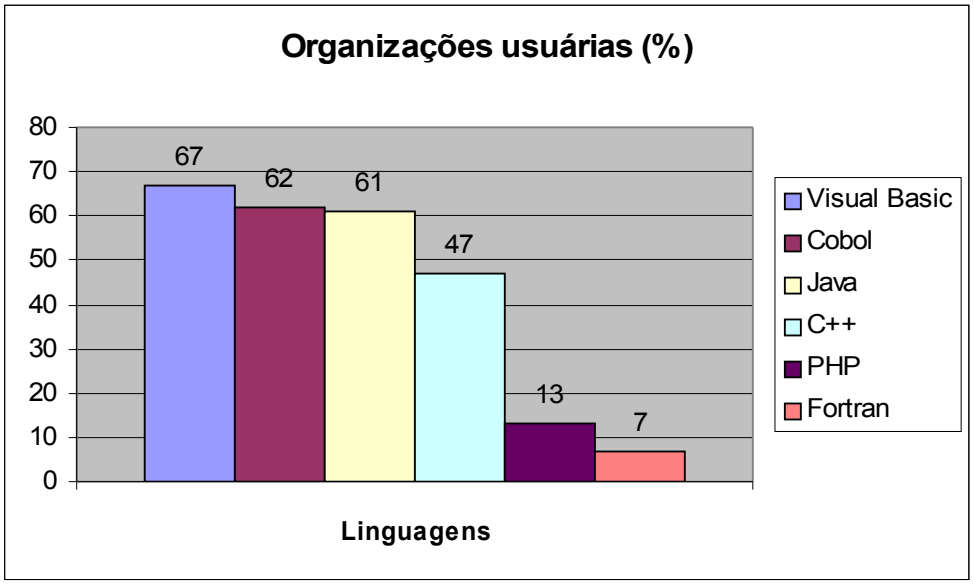

Figura 1.1 – As linguagens utilizadas pelas organizações.
(Computerworld,2006)

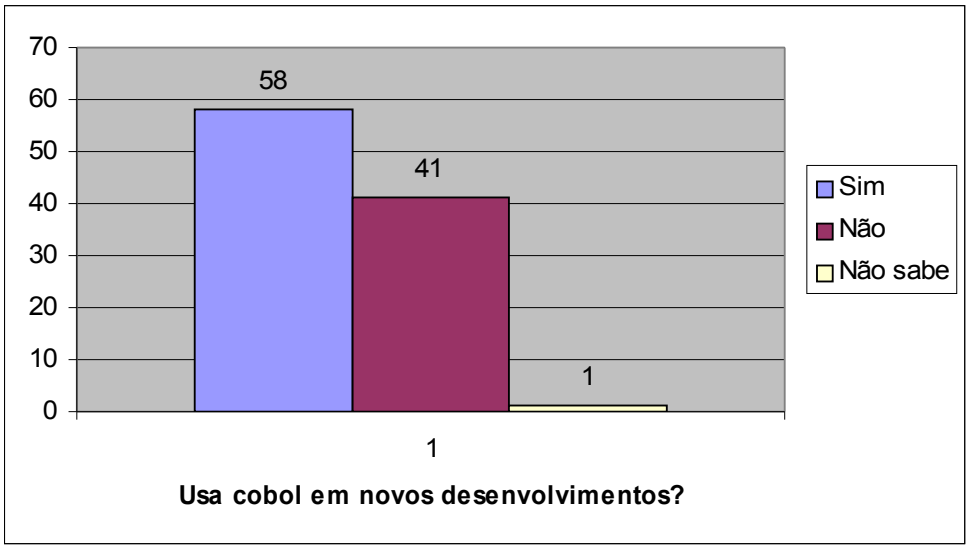

Figura 1.2 – Organizações usuárias do Cobol que estão desenvolvendo novos produtos nessa linguagem.
(Computerworld, 2006)

Essa realidade implica, diretamente, em dois problemas básicos para a área de desenvolvimento de *software*: a dificuldade de manter e evoluir os aplicativos existentes e a dificuldade em reusar soluções já conhecidas para abreviar o tempo de desenvolvimento e melhorar a qualidade dos novos aplicativos. Esses dois problemas podem ser adequadamente

representados pela perda do conhecimento que é utilizado para iniciar o desenvolvimento do *software*.

A programação orientada a objetos surgiu como um paradigma para resolver parte das questões aqui apresentadas. Sua adoção esbarra, no entanto, em um esforço enorme para migrar toda a base instalada para tecnologias capazes de suportar a orientação a objetos. É preciso considerar, também, que o volume de processamento suportado por aplicações escritas em linguagem cobol executadas em rotinas *batch* pressupõe requisitos de performance que não são comumente oferecidos pelas linguagens que suportam orientação a objetos.

É dentro desse contexto que se situa o presente trabalho, no qual se apresenta uma proposta para dotar o processo de desenvolvimento de *software* de uma ferramenta capaz de gerir o conhecimento empregado em sua produção. Cabe destacar que não se pretende abranger todo o processo de desenvolvimento de *software*, mas somente as fases iniciais – de compreensão do negócio e levantamento de requisitos -, nas quais a maior parte do conhecimento que será utilizado no ciclo de vida do produto se faz necessária e, em geral, acaba sendo perdida.

1.1.2 As soluções já tentadas

As principais contribuições para a solução desse problema, têm vindo da área de engenharia de *software*, que orienta profissionais da área quanto aos principais aspectos que devem ser considerados durante o desenvolvimento de novos produtos. Essas contribuições, no entanto, voltam-se muito mais ao controle e organização do processo de desenvolvimento de *software* ligando-se aos aspectos qualitativos do produto final do que aos aspectos da gestão do conhecimento envolvido nas diversas fases da produção, sendo, por conseqüência, desprovida de preocupação quanto ao ciclo de vida dos produtos e quanto ao reuso, em projetos similares.

A aplicação da engenharia de *software* em grandes empresas, é geralmente complementada com atividades ligadas à gestão de projetos, uma vez que é de fundamental importância o controle e acompanhamento de custos e recursos envolvidos no desenvolvimento de novos produtos e serviços.

A gestão de projetos presta, nesse sentido, uma contribuição importante ao planejamento da área tecnológica, uma vez que a decisão sobre desenvolver, ou não, um aplicativo para suporte a alguma atividade negocial se liga diretamente aos custos e prazos envolvidos. Pouco adiantaria a uma empresa se empenhar para desenvolver uma aplicação que somente ficaria pronta após a oportunidade negocial já ter se esgotado.

A figura 1.3 ilustra a situação descrita, indicando as principais contribuições das áreas de engenharia de *software* e da gestão de projetos.

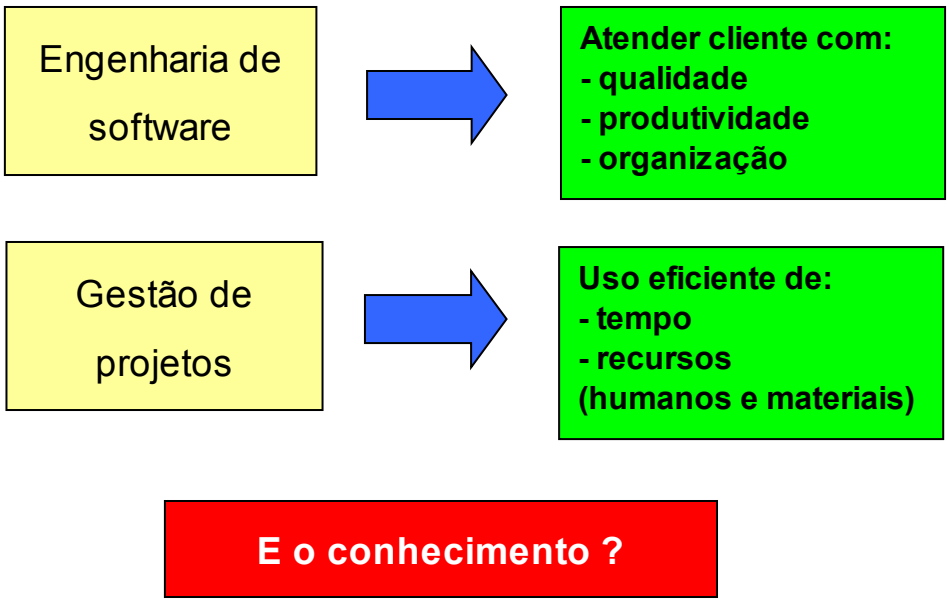

Figura 1.3 – As contribuições da engenharia de software e da gestão de projetos

Identificam-se, também, entre as soluções tentadas para a melhoria do processo de desenvolvimento de *software*, a existência de várias ferramentas para controle e aprimoramento do processo de produção de *software*, usualmente denominadas de CASE (*Computer Aided Software Engeneering*). Entre elas, destaca-se a suíte de aplicações "Rational Rose" comercializada pela IBM, que suporta várias etapas envolvidas no processo de desenvolvimento de *software*. Essas ferramentas, no entanto, apóiam-se no paradigma da programação orientada a objetos, chegando inclusive a geração automática de código nas linguagens C, C++, Java ou similares.

Embora essas duas áreas apresentem uma notável contribuição ao processo de desenvolvimento de *software*, restam, sem resposta, as duas principais questões propostas no âmbito desse trabalho: como gerir de forma apropriada o conhecimento que aflora durante a fase de construção das aplicações e como melhorar o processo de desenvolvimento em ambiente de grande porte?

1.1.3 Os problemas não resolvidos

Embora se tenha vivenciado um avanço significativo na área de desenvolvimento de *software* nos últimos anos, notadamente pelas contribuições da engenharia de *software*, com a

proposição de modelos de gestão do processo, tais como o CMM/CMMI (*Capability Maturity Model / Integration*), o ITIL (*IT Infrastructure Library*) e o MPS-BR (Melhoria do Processo de S*oftware* Brasileiro), observa-se que esses modelos voltam sua atenção para o processo de desenvolvimento, estabelecendo pontos de controle e acompanhamento que permitem a obtenção de produtos de qualidade e processos de produção gerenciáveis e repetíveis. A questão da gestão do conhecimento utilizada no processo não é tratada com a profundidade adequada em nenhum desses modelos.

Em um ambiente de uma grande organização, com diversas áreas demandando por soluções de tecnologia para suportar os seus negócios, o problema agrava-se, pois muitas vezes solicitações similares, são direcionadas para equipes de desenvolvimento diferentes, causando uma duplicidade de esforços que se situa na contramão das necessidades de um desenvolvimento rápido e com utilização eficiente dos recursos disponíveis. Essa situação é ilustrada na figura 1.4, na qual dois produtos com parte similar dão origens a dois aplicativos totalmente distintos.

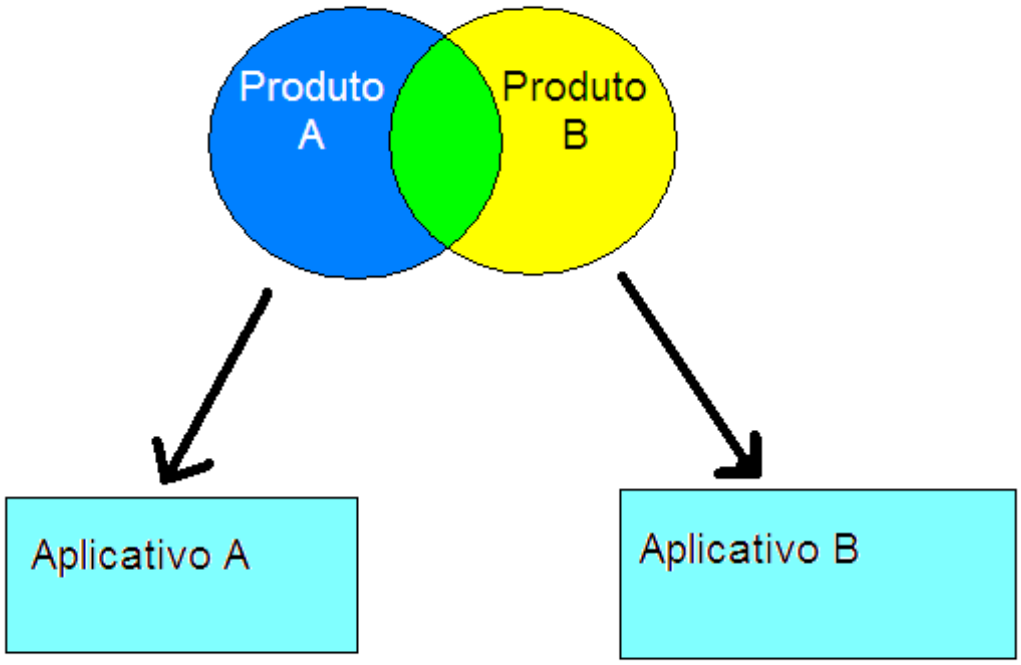

Figura 1.4 – Necessidades com partes em comum e atendidas por sistemas distintos

Não se pretende, no âmbito desse trabalho, substituir as práticas recomendadas por esses modelos. O que se pretende é preencher a lacuna existente em relação ao gerenciamento do conhecimento gerado durante as fases do desenvolvimento das aplicações, especialmente nas fases de compreensão do negócio e de levantamento de requisitos.

1.2 Objetivos deste trabalho

1.2.1 Objetivo geral

Propor um modelo que permeie o processo de desenvolvimento de *software* – especialmente as fases de compreensão do negócio e de levantamento de requisitos –, e permita identificar, com o apoio de ferramentas de inteligência artificial, elementos que possam ser reutilizados de soluções anteriores, definições e características que possam encontrar similares em situações já tratadas, ou em fase de desenvolvimento, e melhorar a rastreabilidade das características dos produtos desenvolvidos.

1.2.2 Objetivos específicos

Estabelecer um modelo de apoio ao desenvolvimento de *software*, baseado nas áreas de conhecimento da engenharia de *software* e de gestão de projetos que permita agregar a gestão do conhecimento ao processo de desenvolvimento;

Identificar elementos que permitam avaliar a similaridade entre solicitações de serviços de TI iniciadas pelas diversa áreas demandantes, baseado na combinação de atributos discretos, ou discretizáveis, e de descrições textuais dos objetivos e funcionalidades a serem implementadas;

Estruturar um protótipo de ferramenta de apoio às fases de compreensão do negócio e de levantamento de requisitos do processo de desenvolvimento de *software* , aplicável a ambiente de grande porte que permita avaliar os resultados obtidos pela aplicação do modelo proposto. Essa ferramenta deve apresentar as seguintes características:

- Apresentar um ambiente para o registro das solicitações de serviços de TI que descrevam as principais características do negócio que se deseja automatizar;
- Permitir a vinculação das solicitações com as atividades desenvolvidas para seu atendimento;
- Apresentar um ambiente para o registro dos requisitos do negócio a ser suportado pela aplicação;
- Permitir a representação dessas informações em uma base de dados que possa ser utilizada para recuperar informações relevantes;
- Extrair informações dessa base e interagir com o usuário apresentando-lhe funcionalidades semelhantes para possível reuso;
- Associar a solução técnica construída para atendimento dos requisitos funcionais;
- Registrar os principais aspectos envolvidos durante as fases de elicitação de

requisitos, modelagem e construção da solução para atendimento das necessidades do cliente.

1.2.3 Contribuição esperada

Como resultado desta pesquisa, espera-se que o processo de desenvolvimento de *software* passe a incorporar os benefícios que podem ser propiciados pela gestão do conhecimento, refletindo tal alteração em economia de tempo e de recursos para a produção de aplicativos e melhoria de sua qualidade. Espera-se, também, que a utilização de formalismos de inteligência artificial na construção desse modelo e na preparação de uma ferramenta de suporte sejam capazes de minimizar a sobrecarga sobre o processo de desenvolvimento que normalmente ocorre pela inclusão de novas atividades.

1.3 Áreas de pesquisa relacionadas

O presente trabalho lança sua atenção sobre três áreas:

- a área de gestão do conhecimento, na qual se insere o ponto principal a ser abordado, que é a criação de mecanismos para uma a melhoria da eficiência do processo de desenvolvimento de *software*;
- a área de inteligência artificial, que apresenta um conjunto de mecanismos capazes de suportar uma ferramenta que trate adequadamente da gestão do conhecimento nesse processo, e
- a área de engenharia de *software*, que com seu notável avanço vêm conseguindo auxiliar o gerenciamento de todo o processo de desenvolvimento, auxiliando os desenvolvedores na obtenção de melhores produtos e os gerentes e executivos de TI num melhor gerenciamento dos custos e riscos envolvidos.

1.4 Organização deste documento

Este documento foi organizado em uma seqüência de capítulos que descrevem a trajetória da pesquisa realizada, apresentando cada uma das etapas de construção do modelo e da ferramente de suporte de maneira detalhada como forma de facilitar sua compreensão e permitir o prosseguimento da pesquisa em trabalhos futuros. Nesse sentido, as informações foram ordenadas da seguinte forma:

Capitulo 1: Descreve o problema que motivou a realização da pesquisa, contextualizando a realidade vivenciada na área e delimita os objetivos da proposta de trabalho;

Capítulo 2: Apresenta uma revisão da literatura e a fundamentação teórica para os principais conceitos e técnicas que são utilizados, abordando as áreas de inteligência artificial, gestão do conhecimento e engenharia de *software*;

Capítulo 3: Em complemento à revisão bibliográfica inserida no capítulo 2, este capítulo apresenta os resultados obtidos por pesquisas correlatas à área tratada. São mostrados os resultados de artigos científicos publicados recentemente e que, de alguma forma, tenham contribuído com o presente trabalho;

Capitulo 4: Este capítulo descreve detalhadamente o modelo construído, em todas as suas fases e etapas, e apresenta todos os elementos que são fundamentais para o correto entendimento do trabalho realizado.

Capítulo 5: Descreve a implementação da ferramenta que foi construída para suporte e validação do modelo;

Capítulo 6: Apresenta a primeira parte do estudo de caso realizado, descrevendo desde a fase de entendimento do negócio até a determinação dos objetivos e estabelecimento de critérios para a avaliação. Optou-se por dispor os resultados em um capítulo a parte, para permitir sua melhor exploração e análise;

Capítulo 7: Segunda parte do estudo de caso, na qual são apresentados e analisados os resultados obtidos;

Capitulo 8: Apresenta as conclusões do trabalho e as indicações dos trabalhos futuros a serem feitos em continuação a pesquisa.

2 Fundamentação teórica

Esta pesquisa fundamenta-se nos princípios de inteligência artificial, da gestão do conhecimento e da engenharia de *software*, e procura aliar contribuições de cada uma dessas áreas como forma de prover uma solução ao problema apresentado no capítulo anterior. São apresentados a seguir um breve resumo das linhas de pesquisa envolvidas e os principais elementos dessas áreas que foram avaliados no âmbito deste trabalho.

2.1 Inteligência artificial

Em seu livro Inteligência Artificial, Luger (2004) define a Inteligência Artificial (IA) como a capacidade de uma máquina executar alguma tarefa que, para ser executada por um ser humano, requereria o emprego de inteligência. Essa definição apresenta uma abordagem interessante, à medida que a sua comparação com uma atividade comumente presente no cotidiano da pessoas facilita a compreensão do termo, sem a necessidade de definições mais complexas em termos técnicos.

Para a consecução dos objetivos propostos neste trabalho, foram investigadas três linhas de IA, que se destacam, seja pela capacidade de prover os recursos necessários, seja pelo uso em aplicativos cuja funcionalidade guarde alguma semelhança com esta proposta. Essas linhas estão brevemente descritas a seguir.

2.1.1 Raciocínio baseado em casos

A área conhecida como raciocínio baseado em casos apresenta um modelo de aproximação da forma como os seres humanos utilizam seu cérebro para inferir resultados a partir de experiências passadas. Sua proposição é atribuída aos trabalhos iniciados por Kolodner (1993), no qual se fundamentaram as bases para essa nova forma de resolução de problemas, e que vislumbrava uma nova fronteira no campo da inteligência artificial à medida em diminuía sensivelmente a dependência de um especialista no domínio da aplicação para a produção de regras, geralmente utilizadas por sistemas especialistas.

Os trabalhos de Kolodner materializavam as idéias defendidas anteriormente por Schank (1991) que associavam o processo de aprendizado ao de memorização e descreviam o que ela acreditava ser uma das formas comumente utilizada pelo cérebro humano. A teoria cognitiva predominante na época de seus estudos apontava para o aprendizado simbólico e

associava a inteligência à capacidade de manipular esses símbolos com a utilização de regras.

É certo que a mente humana consegue inferir resultados de maneira bastante eficiente quando utiliza experiências passadas como ponto de partida para a solução de problemas. Consideremos, por exemplo, a intenção de calcular o valor da raiz quadrada de 35,9. O conjunto de operações matemáticas necessárias para obter tal valor é conhecido e plenamente aplicável, mas essa seqüência de operações pode requer uma quantidade grande de esforços para obter o valor final. Se, por outro lado, uma aproximação razoável do valor nos for suficiente, podemos afirmar que o valor é maior que 5 e menor que 6 apenas partindo do conhecimento de que a raiz quadrada do número 25 é igual a 5 e a do número 36 é igual a 6. Nesse exemplo, pode-se demonstrar, de maneira bastante simples, o potencial do método, que permite a obtenção de soluções para problemas presentes pela adaptação de outros resultados já conhecidos.

Essa forma de raciocínio é bastante útil, principalmente quando um grande grupo de usuários precisa investigar e apontar soluções para problemas que lhes são apresentados. Nessa abordagem, a possibilidade de acessar os resultados já obtidos pelos demais integrantes do grupo pode representar uma significativa economia de esforços, além de contribuir para a padronização desses resultados.

A aplicação desse método, embora de compreensão bastante fácil, requer alguns cuidados, entre os quais pode-se destacar os seguintes:

representação do conhecimento – Como o método se baseia na busca e adaptação de casos passados para inferir uma solução para o problema atual, é extremamente importante que a representação desses casos seja feita de forma a facilitar a pesquisa. A representação de dados por valores discretos, permite, via de regra, que se encontre com maior facilidade valores iguais ou próximos aos valores desejados, enquanto que elementos descritos em linguagem natural apresentam maior dificuldade;

utilização de pesos – Muitas vezes para determinar o quão similar um caso é de outro conhecido, são considerados vários fatores com a adoção de pesos diferenciados para cada fator.

Tome-se, por exemplo, a intenção de determinar o grau de similaridade entre três carros, considerando apenas os atributos cor e marca; suponha-se que o conjunto para estudo seja o indicado na tabela a seguir, no qual o termo IS x-y é usado para designar o índice de similaridade existente entre os casos x e y:

Caso	Veículo	Fabricante	Cor	Grau de similaridade
1	Gol	VW	Vermelho	IS 1-2 = 1
2	Palio	Fiat	Vermelho	IS 2-3 = 0
3	Golf	VW	Verde	IS 3-1 = 1

Tabela 2.1 – Exemplo de aplicação de CBR

Nesse exemplo, a determinação do grau de similaridade considerou dois atributos (fabricante e cor), resultando em um valor igual para comparação entre os veículos 1 e 2 e entre os veículos 1 e 3. Apresenta-se, nesse momento, uma importante questão que precisaremos responder para obter resultados significantes com a utilização do método. “O que é mais importante ao determinar o grau de similaridade: a cor do veículo ou o fabricante? A resposta para essa questão não é tão simples quanto pode parecer a primeira vista e exige alguma investigação adicional. É preciso determinar, primeiramente, que uso faremos da informação extraída dessa comparação: se a informação será utilizada para um fabricante de tintas, a cor pode ser mais relevante; já sob a ótica de um departamento de compras de peças, por exemplo, a informação sobre o fabricante poderá ter relevância muito maior.

Este exemplo, ainda que bastante simples, nos faz perceber que para determinar o grau de similaridade entre casos devemos ter em mente qual a aplicação que queremos dar aos resultados obtidos e nos traz presente a necessidade de que, qualquer modelo, construído com base nesse tipo de raciocínio irá requer uma calibragem e sucessivos refinamentos, que somente poderão ser feitos com o auxílio de um especialista no domínio do problema que se pretende resolver.

A figura 2.1 ilustra o funcionamento de um sistema de diagnóstico utilizando o mecanismo de raciocínio baseado em casos, no qual um problema que se pretende resolver é encapsulado em um novo caso para comparação e recuperação de casos similares a partir de uma base de casos anteriormente tratados. O ciclo prossegue com a reutilização (e eventual adaptação) de algum dos casos recuperados, sua revisão e retenção (ou aprendizado) que é a etapa responsável pela geração de novos casos que poderão ser também utilizados para a solução dos próximos casos.

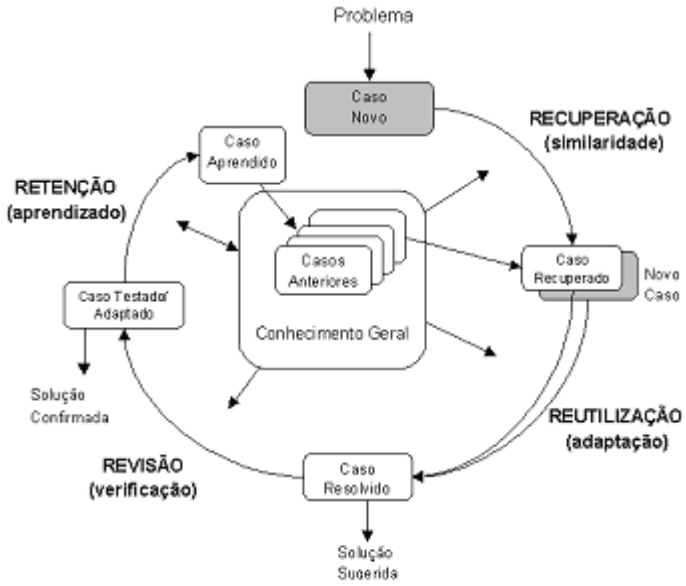

Figura 2.1 – Funcionamento típico de um sistema de raciocínio baseado em casos

2.1.2 Redes neurais artificiais

A percepção de que o cérebro humano processa informações de uma forma completamente diferente do computador digital convencional (Haykin, 2001) têm motivado o trabalho com redes neurais artificiais.

A compreensão dessas diferenças não é intuitiva, posto que a maioria das pessoas identifica o computador como uma máquina capaz de fazer rapidamente um conjunto de operações que o homem levaria muito tempo para tratar. Essa idéia aplica-se, de fato, à tarefas repetitivas e de processo controlado e seqüencial. Quando se trata de operações mais complexas, em que muitas informações devem ser processadas paralelamente e adaptadas ao contexto para a obtenção de um resultado, o cérebro humano demonstra sua infinita superioridade sobre as máquinas. A estrutura responsável pela visão, por exemplo, deve ser capaz de receber informações de um conjunto enorme de elementos dispostos ao seu redor e

processá-los simultaneamente para formar uma correta percepção do ambiente. Discernir um rosto de outros, em meio a uma multidão de pessoas, é uma tarefa simples até para uma criança, mas extremamente complexa e onerosa mesmo para os computadores mais sofisticados.

Uma rede neural artificial tenta simular o funcionamento do cérebro humano, por meio de dispositivos que sejam capazes de reagir a determinados estímulos. Sabe-se que o cérebro humano é formado por um complexo conjunto de estruturas, entre as quais temos as células nervosas, chamadas de neurônios, interligadas entre si; essas ligações, chamadas de sinapses, estabelecem através de pequenos impulsos elétricos uma relação de excitação entre os neurônios que levam a percepção de tudo o que nos cerca e determina nossa relação com o ambiente no qual estamos inseridos. É através de impulsos elétricos propagados pelos neurônios que temos, entre outras coisas, a percepção de todos os nossos sentidos.

A figura 2.2 ilustra a composição de um neurônio.

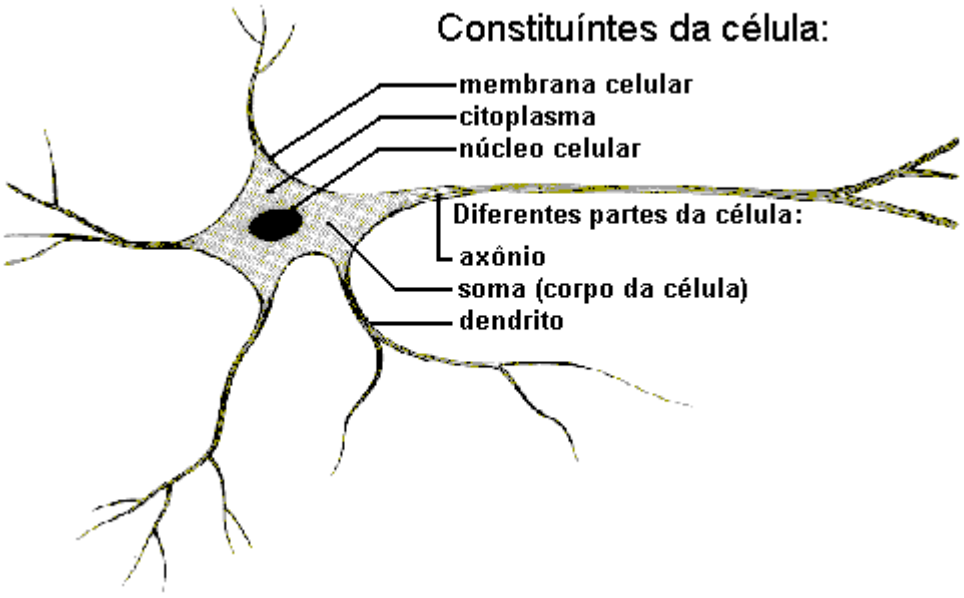

Figura 2.2 – Constituição típica de um neurônio.
(fonte:Carvalho,A.2001)

Os principais componentes dos neurônios são:

- Os dendritos, que tem por função, receber os estímulos transmitidos pelos outros neurônios;
- O corpo de neurônio, também chamado de soma, que é responsável por coletar e combinar informações vindas de outros neurônios;

- O axônio, que é constituído de uma fibra tubular que pode alcançar até alguns metros, e é responsável por transmitir os estímulos para outros neurônios.

Uma rede neural artificial tenta simular as funções desempenhadas pelo cérebro e, embora existam algumas variações, seu funcionamento é baseado principalmente em estruturas denominadas de perceptrons que tentam modelar o funcionamento de um neurônio humano.

> Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:
>
> 1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
>
> 2. Forças de conexão entre neurônios, conhecidos como pesos sinápticos, são utilizados para armazenar o conhecimento adquirido. (Haykin, 2001, p.28)

Essa definição de Haykin expressa os fundamentos da construção de redes neurais, cuja estrutura e funcionamento materializam uma aproximação matemática para a função executada por um neurônio e suas interconexões. Os neurônios são modelados por estruturas chamadas perceptrons que recebem entradas e se interligam em camadas com pesos diferenciados numa aproximação das ligações sinápticas realizadas pelo cérebro humano. O simbolismo matemático de um perceptron está representado na figura 2.3:

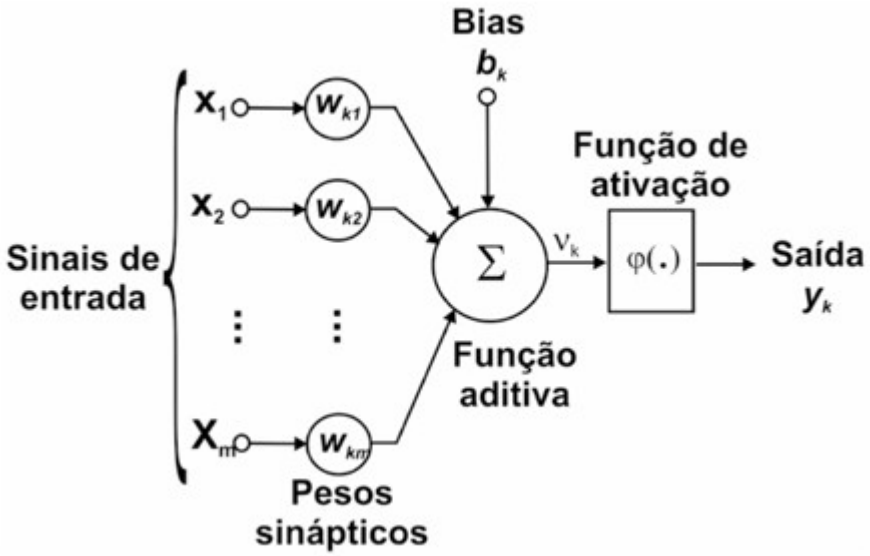

Figura 2.3 – Representação de um perceptron

(Adaptado de Haykin, 2001)

Nessa representação percebe-se a existência de um conjunto de entradas (x_1 ... x_n) as quais são aplicados pesos relativos (w_1 ...w_n); o somatório desses sinais é então aplicado a um

componente que decide (através de uma função de ativação) se o sinal recebido é suficiente ou não para ativar a saída do perceptron. A ativação ou não da saída, em função dos sinais de entrada , que corresponde a aquisição de conhecimento do cérebro humano, é obtida por uma etapa de treinamento da rede neural, onde ocorre o processo que se define como aprendizagem.

Esse modelo é a base para a construção de redes neurais artificiais e apresenta uma ampla possibilidade de interligações e a construção de vários tipos de redes. Essas redes podem então ser treinadas para que um dado conjunto de entradas produza uma determinada saída pela modificação dos pesos relativos. Nessa situação, podemos construir um mecanismo capaz de identificar similaridades entre padrões de entrada e produzir saídas que correspondam a uma interpretação adequada a esse estímulo.

2.1.3 Redes bayesianas

As redes bayesianas são construídas a partir dos estudos probabilísticos conduzidos pelo matemático Thomas Bayes e publicados em 1763, dois anos após sua morte. Em seu trabalho, Bayes apresenta uma abordagem adequada para a representação e tratamento do conhecimento incerto. Entende-se aqui, por conhecimento incerto, aquele que pode apresentar alguma representação não exata, parcial ou aproximada da realidade.

Esse formalismo serve de base para a construção de redes que têm a forma de grafos acíclicos direcionados, cujos nós representam as variáveis do domínio e os arcos as dependências entre as variáveis. Cabe ressaltar aqui a necessidade de que as variáveis sejam condicionalmente independentes para que possa ser aplicado o Teorema de Bayes.

A cada um dos nós associa-se um tabela de probabilidades e, a partir da propagação da influência de um conjunto de variáveis conhecidas pode-se calcular os valores para cada uma dessas probabilidades e, conseqüentemente realizar inferências a partir de novos casos a serem avaliados.

A figura 2.4 ilustra o funcionamento típico de uma rede bayesiana para determinar a probabilidade da ocorrência de chuva a partir da evidência de que a grama está molhada e das tabelas de probabilidades condicionais associadas a cada uma das variáveis. É um exemplo clássico da literatura, que resume de forma simples tal funcionamento.

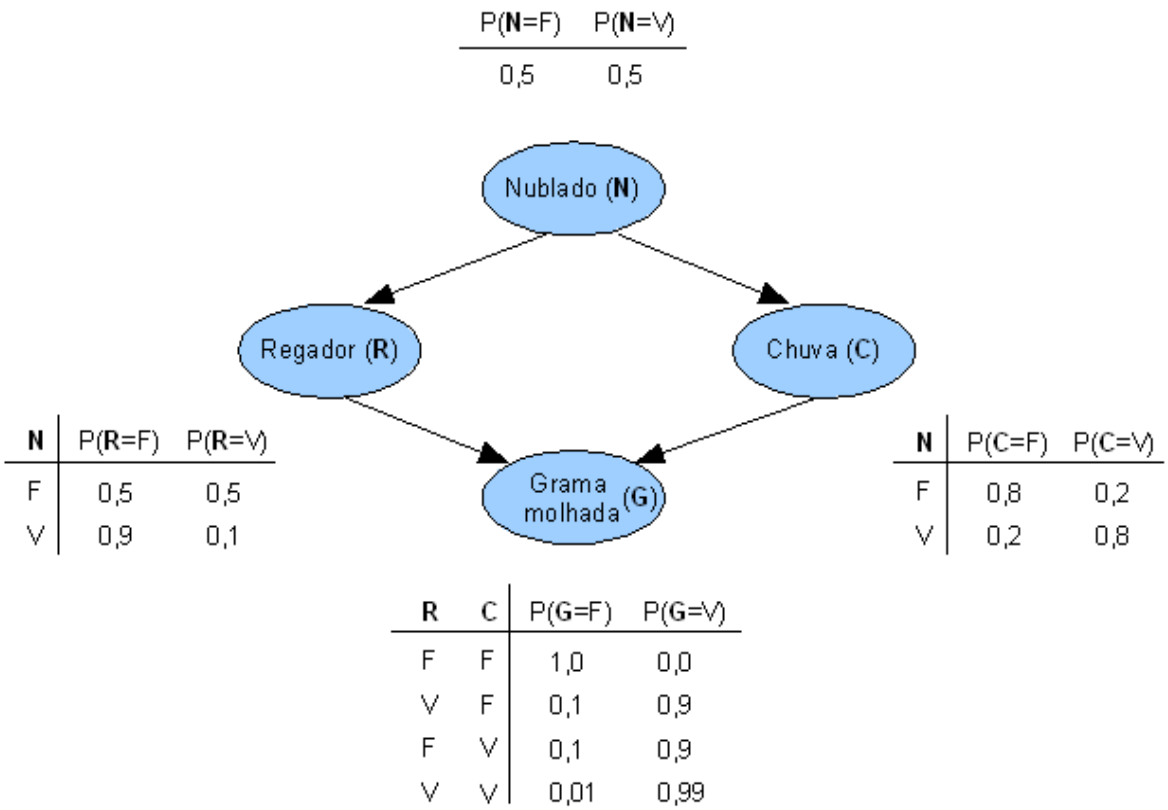

Figura 2.4 – Esquema típico de construção de uma rede bayesiana.

(adaptado de Murphy, 1998)

A utilização de redes bayesianas na determinação de similaridades de padrões, vêm sendo estudada já há bastante tempo, podendo-se destacar o trabalho conduzido por Eric Horvitz (1995) em um projeto da empresa Microsoft, denominado Lummiére, que culminou com a adoção dessa tecnologia para a construção dos assistentes utilizados pelos aplicativos da família MS-Office na interação com usuários, provendo-lhes a informação necessária para a conclusão de uma dada tarefa a partir de uma pesquisa efetuada com base em redes desse tipo.

2.2 Mineração de Dados

A área conhecida por essa denominação volta suas atenções ao processamento de informações com o objetivo de extrair informações úteis ou relevantes de um conjunto de dados do qual, não se tenha um conhecimento prévio. Essa abordagem apresenta uma possibilidade extremamente rica e de interesse da comunidade científica à medida que o crescente uso de sistemas de processamento eletrônico de informações tem dado origem a grandes bases de dados que, muitas vezes, não voltavam sua atenção a estruturação da

informação e, dessa forma, negligenciavam esse precioso bem.

Buscando suprir essa lacuna e recuperar as informações relevantes em meio a uma quantidade enorme de informações disponíveis em formatos diversos, têm sido desenvolvidas várias técnica para o processamento dessas informações, visando a obtenção de informações sobre relacionamentos e sobre a relevância dos dados. Em geral esse processo envolve uma etapa de pré-processamento das bases de dados, o reconhecimento de padrões e uma análise estatística sobre os resultados obtidos.

O exemplo ilustrado na figura 2.5 representa as etapas envolvidas no processo de mineração de dados, que consistem em uma seqüência de estágios destinadas a transformar dados em conhecimento.

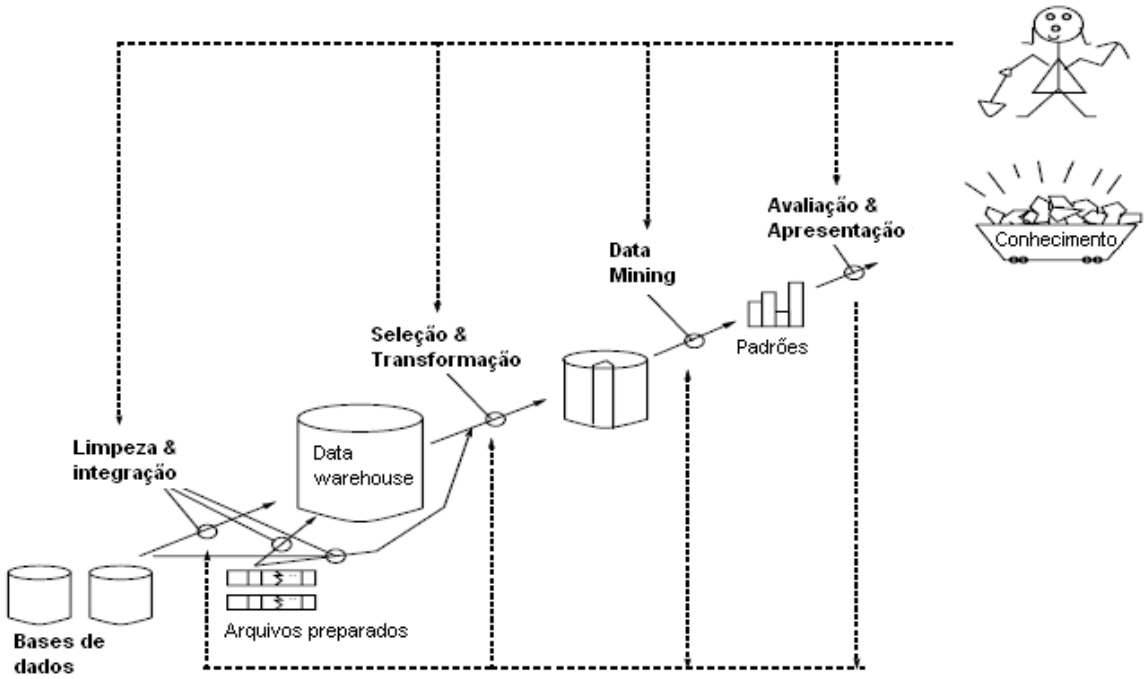

Figura 2.5 – Estágios do processo de mineração de dados

(Han, 2001)

Esses estágios incluem as atividades necessárias para, partindo de uma base dados não estruturada, aplicar processos para limpeza e integração dos dados e armazená-las em estruturas adequadas, aplicar procedimentos para seleção e transformação dos dados, efetuar sua mineração, avaliar e apresentar os resultados na forma de um conhecimento útil sobre o domínio ali representado.

As técnicas de *mineração de dados* permitem, por exemplo, a construção de classificadores, de categorizadores e de clusterizadores, que têm sido objeto de extensos estudos porquanto permitam explicitar relações importantes e que poderiam ser desperdiçadas em meio a uma massa enorme de dados insignificantes. Uma das áreas que utiliza esses

conceitos é o combate ao terrorismo que busca, em meio a milhares de mensagens que circulam diariamente na rede mundial de computadores, identificar aquelas que possam representar alguma ameaça e tomar as medidas preventivas para evitar ou minimizar seus efeitos. A utilização na área de negócios também se vale dessas técnicas, buscando identificar padrões de comportamento que possam prever ou incrementar o consumo de determinados produtos.

Derivada das pesquisas com *mineração de dados*, tem-se a área que se preocupa com a extração dessas mesmas relações sobre bases de informações exclusivamente textuais, normalmente designada por *Textual Data Mining* ou simplesmente *Text Mining*. Esse procedimento, que se vale das mesmas técnicas de processamento e de análise estatística dos resultados, permite extrair relações relevantes de um conjunto de informações não adequadamente estruturadas.

2.3 Processamento de linguagem natural

Uma dificuldade típica em trabalhos com as metodologias descritas nos itens anteriores consiste no tratamento de elementos descritos em linguagem natural. Nessa situação, a comparação de elementos se torna especialmente difícil face as inúmeras possibilidades de expressar uma mesma situação, fato ou objeto com o uso de sinônimos, ou com a flexão de palavras para expressar corretamente gênero, número e grau.

Há de se compreender, ainda, que qualquer descrição em linguagem natural é fortemente dependente do contexto em que está inserida, fazendo com que sua análise deva se dar sobre mais de uma ótica.

Luger (2004) ao tratar a questão da desconstrução da linguagem e de sua análise simbólica descreve sete níveis definidos por linguistas como importantes para se lidar com a questão da linguagem natural:

- A prosódia, ligada ao ritmo e entonação da linguagem que, embora de difícil formalização é imprescindível para a compreensão de expressões artísticas e religiosas;
- A fonologia, que trata da combinação dos sons para a formação da linguagem;
- A morfologia, que trata da constituição das palavras, na qual se situa o conjunto de regras para a formação e derivação de palavras;
- A sintaxe, que cuida da combinação de palavras, sentenças e frases;
- A semântica, ligada ao significado de palavras, frases e sentenças;

- A pragmática, que estuda a forma de uso da linguagem;
- O conhecimento do mundo, que inclui aspectos da percepção do ambiente e da interação social na formação da linguagem.

Cada um desses níveis apresenta contribuições específicas em determinados contextos; para a compreensão de uma poesia, por exemplo, a prosódia - ligada a ritmo e emoção - tem efeito muito maior que a semântica, apesar de neste nível estar enxerta a questão de significância das palavras. Analogamente, aplicações que tratem reconhecimento de voz humana, ou sua geração, devem ocupar-se principalmente de questões ligadas à fonologia das palavras.

Um modelo que se pretenda capaz de extrair informações relevantes para um dado contexto em uma porção de texto deve ser capaz de abstrair essas situações fazendo as aproximações e conversões necessárias para que a comparação de elementos possa retornar resultados adequados, ainda que tenham sido utilizadas construções distintas da linguagem para representar uma mesma coisa. Diversas abordagens vêm sendo experimentadas nessa área, cada qual com suas vantagens e desvantagens, o que torna a escolha de uma determinada técnica extremamente difícil e dependente da situação que se deseja tratar. São descritas nos itens seguintes duas das abordagens mais comumente presentes nas pesquisas envolvendo a mineração de dados em elementos textuais: o modelo espaço vetorial e *Textual Case Based Reasoning* (TCBR).

2.3.1 Modelo espaço vetorial

Esse modelo foi inicialmente descrito por Salton (1971) e se apresenta com uma alternativa simples e eficiente para o processamento de elementos descritos em linguagem natural, que corresponde à representação do conjunto de termos de um documento na forma de uma matriz A, de dimensões m por n ($A_{m \times n}$). Cada linha da matriz corresponde a um documento da base e cada uma das entradas da matriz $A_{i,j}$ corresponde à freqüência relativa de cada termo desse documento. Segundo Berry (2004) o maior benefício desse modelo é que pode ser explorada a estrutura algébrica do espaço vetorial; no entanto as dimensões m e n tendem a crescer rapidamente, prejudicando a eficiência do modelo. Para resolver essa questão, têm sido propostas diversas alternativas, que vão desde a clusterização dos termos em cada documento à modelos probabilísticos, objetivando sempre a preservação da informação e sua representação em um modelo de tratamento mais fácil e simples.

São freqüentes na literatura exemplos e metodologias para a representação dos termos

contidos em um documento na forma de um vetor do espaço. A cada conjunto de termos de um documento é atribuído um vetor, de forma que a distância entre dois vetores represente a similaridade entre os documentos a eles associados. Quanto menor o distanciamento entre os vetores maior a similaridade, e vice-versa.

Uma das abordagens possíveis e largamente empregadas para nessa área é o modelo conhecido por TF-IDF (*term frequency – inverse document frequency)*, que permite processar os elementos presentes em uma porção de texto, fazendo a distinção entre a relevância dos termos a partir de valores de freqüência com que cada termo existe em uma base usada para comparação.

A idéia central deste modelo consiste em avaliar cada um dos elementos (termos) constantes no fragmento de texto em estudo, considerando, para isto, a freqüência com que o termo ocorre nesse texto e compará-los à freqüência com que esse mesmo termo ocorre nos demais casos registrados na base. Quanto mais freqüente é um determinado termo no conjunto de textos que compõem a base de comparação, menor será sua relevância para efeito de mineração de dados. Isso equivale a dizer que um termo que ocorra poucas vezes na base de casos e que ocorra muitas vezes no documento deverá ter, para efeito de comparação, uma relevância maior que termos que são mais freqüentemente encontrados nos documentos. Essa abordagem apresenta uma característica interessante para a comparação de elementos descritos em linguagem natural, posto que cada termo do documento poderá apresentar uma contribuição diferente para apuração do índice final de similaridade entre os casos comparados. O exemplo a seguir ilustra uma utilização desse modelo para determinar o grau de similaridade entre o nome de pessoas, imaginando que se deseja encontrar, a partir do nome, alguma relação de parentesco.

Exemplo de aplicação do modelo TF-IDF

Casos para comparação disponíveis na base:

Id-caso	Nome
A	José de Bragança Mendelev
B	José Eurico da Silva Santos
C	José da Silva Bragança
D	José Gomes dos Santos Silva

Tabela 2.2 – Exemplo de base de casos

Caso em estudo → X = José da Silva Santos Mendelev

Freqüência dos termos (considera cada nome da base de casos como um documento).

Termo (t)	Freqüência	Freqüência invertida (fi)
José	4	0,2500
Silva	3	0,3333
Santos	2	0,5000
Mendelev	1	1

Tabela 2.3 – Exemplo de processamento dos termos do caso em estudo

Admitindo-se (para efeito didático) que o índice de similaridade entre o caso X e um caso Y qualquer possa ser descrito pela $IS(X,Y)=\sum(f(t)\,x\,fi(t))$ formula para os termos (*t*) que ocorram simultaneamente em X e Y ,temos a seguinte relação:

*f(*José*)* = 1 ; *fi(*José*)* = 0,2500

*f(*Silva*)* = 1 ; *fi(*Silva*)* = 0,3333

*f(*Santos*)* = 1; *fi(*Santos*)* = 0,5000

*f(*Mendelev*)* = 1 ; *fi(*Mendelev*)* = 1

IS (X-A) = 1 x 0,2500 + 1 x 1 = 1,2500

IS (X-B) = 1 x 0,2500 + 1 x 0,3333 + 1 x 0,5000 = 1,0833

IS (X-C) = 1 x 0,2500 + 1 x 0,3333 = 0,5833

Este resultado mostra uma característica interessante do método, uma vez que o termo "Mendelev" por ser o mais raro na base de casos foi determinante para a apuração do índice final de similaridade. Apesar de entre os nomes X e A haver apenas dois termos com ocorrência simultânea, eles foram considerados mais similares do que os nomes X e B que apresentavam 3 termos iguais.

É interessante ressaltar que o modelo inicialmente proposto como parte do trabalho de Salton tem sido objeto de diversas análises e alterações, que buscam agregar alguma heurística como fator de discernimento entre termos mais, ou menos, relevantes.

2.3.2 Textual case based reasoning (TCBR)

O ramo designado por este nome vem ganhando espaço nas pesquisas envolvendo técnicas para a recuperação de informação em documentos, uma vez que sua proposta agrega elementos baseados no domínio específico da solução que se está buscando para melhorar a qualidade da informação recuperada e estabelecer índices capazes de identificar similaridades entre documento.

Brüninghaus(2001) apresenta uma rica análise sobre o processamento de textos com essa metodologia, elencando as condições que precisam ser atendidas para que se tenham resultados adequados. Entre essas condições, está a necessidade de uma profunda análise sintática do texto, em relação ao tipo de extração de informação que se deseja. De forma análoga à apresentada por Luger(2004), o contexto em que se dará a utilização da informação pode ser determinante não apenas para essa análise, mas, também, para o estabelecimento da abordagem que deve ser utilizada. Ao se tratar textos que representem ofertas de emprego, por exemplo, uma classificação quanto à área de atuação ou quanto à região geográfica em que se situa a vaga pode ser muito mais significativa que a similaridade entre duas ofertas.

A abordagem proposta supre uma deficiência do modelo vetorial que baseia sua avaliação de similaridades no tratamento das freqüências com que cada termo é encontrado no documento em relação a um conjunto de documento sobre os quais se pretende fazer a avaliação. A dificuldade reside em que aquisição de conhecimentos no domínio da aplicação precisa ser cuidadosamente mapeada, de forma a se obter um vocabulário apropriado e representativo. Algumas técnicas vêm sendo estudadas para minimizar o esforço necessário à construção desse vocabulário, associando paradigmas do processamento de linguagem natural, tais como o estudo de trigramas, e de sintagmas nominais e de identificação de radicais formadores dos termos.

Sua implementação, no entanto, ainda exige um esforço muito maior para o mapeamento do domínio da aplicação, na qual existe grande intervenção de especialistas, e uma maior complexidade dos algoritmos de processamento, com o conseqüente consumo de maior quantidade de recursos computacionais.

É interessante notar que muitas das idéias introduzidas por essa abordagem podem ser agregadas ao modelo de recuperação de informações por análise vetorial, entre os quais se destaca a construção de um dicionário de termos e palavras chaves no domínio da aplicação específica, que pode ser utilizados para enriquecer os resultados daquele modelo.

2.4 Gestão do conhecimento

Informações divulgadas pela Organização Mundial do Comércio, em relatórios disponíveis em seu site (WTO, 2008), estimam que mais da metade das riquezas produzidas no mundo estejam direta ou indiretamente ligadas ao conhecimento; essa mudança vem se consolidando com o passar do tempo, fazendo com que a riqueza da produção industrial e agropecuária ceda espaço a serviços e bens cujo valor está intimamente ligado ao conhecimento agregado em sua geração ou produção.

Nesse contexto, a gestão do conhecimento vem sendo apontada como a grande ponte que liga as empresas ao sucesso em seu ramo de atividade. Há muito tempo que o valor das empresas não se mede apenas pelos seus ativos, representados por imóveis, móveis e outros bens cujo valor possa ser facilmente determinado para constar em registros contábeis; sabe-se hoje que o verdadeiro valor das empresas inclui uma parcela de difícil mensuração, que é representado pelo conhecimento utilizado para fazer acontecer seus negócios. Esse conhecimento está disperso nas organizações e reside, muitas vezes, apenas nas mentes de um grupo de funcionários trazendo um risco enorme à continuidade dos negócios e a capacidade de gerar os resultados desejados. Essa constatação fez com que a gestão do conhecimento ganhasse um importante papel nas organizações atuais e que vários estudos se desenvolvessem sobre esse tema.

A definição de conhecimento carrega, no entanto, uma carga de conceitos que não são de compreensão tão direta; é importante para isso, definir alguns termos que são usados neste trabalho e ajudam a compreender melhor do que se trata:

- Dado: é o símbolo ou conjunto de símbolos utilizados para representar algum fenômeno, objeto ou evento;
- Informação: É o dado associado à sua interpretação
- Conhecimento: É a informação estruturada e associada ao contexto na qual se insere;

O grande problema que se apresenta na gestão do conhecimento, é que, embora a representação de dados e informações seja relativamente simples, a representação do contexto passa a requerer a capacidade de tratar elementos de difícil representação. Estruturar a informação e associá-la a um determinado contexto permite que esse conhecimento possa ser explorado, relacionado a outros e utilizado para melhorar a eficiência e a qualidade de processos e produtos.

Nesse sentido, o conjunto de informações contextualizadas e adequadamente estruturadas pode representar um valioso diferencial para qualquer organização, porém não basta apenas gerar o conhecimento; é necessário, também, criar um ciclo virtuoso no qual o conhecimento gerado em qualquer etapa de um processo possa ser explicitado, representado e disponibilizado para uso pelos demais integrantes da organização. Esse ciclo é ilustrado na figura 2.3.

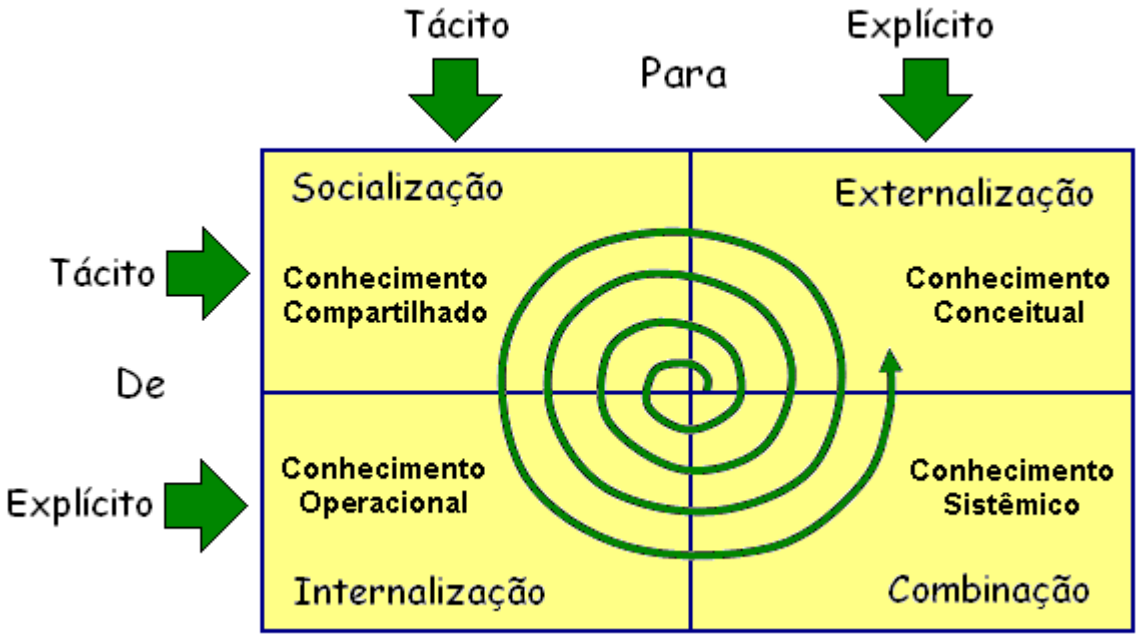

Figura 2.6 – A espiral do conhecimento

(Adaptado de Nonaka, 1997)

Nesse modelo, os conhecimentos são classificados entre tácitos e explícitos; os conhecimentos tácitos são aqueles de domínio dos funcionários da organização e que são empregados diariamente em suas atividades, mas cuja sistematização não encontra uma representação específica e regulamentada pela empresa. Um funcionário de uma oficina mecânica sabe, por exemplo, que antes de suspender um veículo em um elevador para efetuar algum reparo deve aguardar pela saída do condutor do veículo, embora tal restrição não esteja, em geral, expressa nas normas ou regulamento. Esse tipo de conhecimento é utilizado a todo instante em muitas atividades e o principal problema para a gestão do conhecimento é que, embora pareça óbvio, contextualizar atividades como essa não é tão simples.

O modelo pressupõe a existência de quatro fases distintas, na qual o conhecimento está presente: a fase de socialização, que ocorre quando o conhecimento é empregado na presença dos demais integrantes da equipe, e permite-se, portanto, uma percepção do contexto no qual

foi empregado; a fase de interiorização, quando após a exposição a um evento qualquer, suas peculiaridades são assimiladas. As fases seguintes consistem na combinação do conhecimento adquirido, que pode dar origem a novos conhecimentos e na externalização desses novos conhecimentos.

A idéia que se insere no âmbito deste projeto é de capturar os conhecimentos que são externalizados tanto pelo cliente quanto pelo desenvolvedor do *software*, aplicar-lhes um tratamento adequado para que passem a compor uma base de conhecimentos que permeie todo o projeto e que esteja disponível para o auxilio em novos projetos similares ou correlatos.

2.4 Engenharia de software

Em seu livro Engenharia de software, Pressman (1995, p.34) afirma que "a engenharia de *software* é um rebento da engenharia de sistemas e de *hardware*". De fato, conceitua-se que a engenharia de sistemas é responsável não apenas pela produção do *software*, mas de criar as condições necessárias a sua correta execução e para o atingimento dos fins propostos, enquanto a engenharia de *hardware* se encarrega do projeto de componentes eletrônicos capazes de desempenhar determinadas atividades. O processo de desenvolvimento de *software*, esteve, por muito tempo, à sombra da engenharia de *hardware*, com suas atividades conduzidas de maneira quase artesanal.

Embora a etapa de elaboração do *software* requeira uma boa dose de criatividade, a expansão da utilização de aplicações para praticamente todas as atividades ligadas ao dia-a-dia das pessoas e ao suporte de negócios que envolvem quantias cada vez maiores de dinheiro exigia que essa etapa pudesse ser acompanhada e controlada como forma de garantir que os produtos seriam confeccionados dentro de uma estimativa razoável de tempo, custo e qualidade/confiabilidade.

Essa preocupação deu um grande impulso à pesquisa de métodos e procedimentos de controle para cada das fases que compõem o ciclo de desenvolvimento de *software*, sendo hoje empregada na maior parte das organizações como uma condição essencial à sua sobrevivência.

A engenharia de *software* divide o processo de construção de aplicativos em um conjunto de etapas ou fases, sendo consensualmente aceitas pela área a utilização de, pelo menos, as seguintes:

- Planejamento do projeto: consiste na etapa inicial, na qual serão verificadas a viabilidade e a conveniência de desenvolver uma solução aderente às necessidades do negócio ou a possibilidade adquirir no mercado uma solução existente para tal fim. Essa etapa preliminar é uma importante ferramenta utilizada pelas empresas como forma de evitar o gasto desnecessário de recursos;
- Levantamento de requisitos: primeira etapa de construção da solução, propriamente dita. Nesta etapa existe uma grande interação do cliente com os técnicos para que sejam compreendidas todas as funcionalidades que o aplicativo deve possuir, as restrições que deve cumprir (tais como tempo de resposta e plataforma de acesso, entre outras). Acordam-se nesse momento que funcionalidades serão atendidas e também aquilo que não fará parte da solução. Essa etapa é concluída, em geral, com a elaboração de um documento de requisitos de sistema, listando todos os requisitos funcionais e não funcionais que devem ser atendidos e o compromisso das partes que tal solução atende aos objetivos propostos;
- Projeto e construção do sistema: etapa na qual se elaboram os modelos de dados, se define a arquitetura de processamento e armazenamento das informações e a construção do conjunto de programas e rotinas que comporão o sistema como um todo;
- Testes e validação: essa etapa consiste na verificação do atendimento dos requisitos para cada um dos módulos construídos e para o todo o conjunto. Usualmente, os testes são divididos em testes unitários, testes de integração, testes de sistemas e teste de aceitação;
- Implantação do sistema: etapa que consiste na colocação em produção do aplicativo desenvolvido. Essa etapa precisa ser atentamente acompanhada para garantir que a solução seja corretamente executada, notadamente em grandes organizações, nas quais os ambientes de desenvolvimento e de produção são isolados, operados por pessoas diferentes e, eventualmente com diferenças entre versões de produtos instalados;
- Manutenção e evolução: etapa que se estende por todo o ciclo de vida do aplicativo desenvolvido, e que compreende a correção de erros não detectados durante a fase de testes e de intervenções destinadas a manter o produto

adequado às necessidades do negócio, inclusive no que tange a regulamentação legal porventura envolvida.

É importante ressaltar, neste ponto, a estreita ligação existente entre as áreas de engenharia de *software* e de gerenciamento de projetos. Embora a engenharia de *software* delimite adequadamente as fases e os elementos resultantes de cada uma delas para a continuação das atividades nas fases posteriores, não se apresentam ali os elementos para controle e acompanhamento de prazos e recursos envolvidos. A área de gestão de projetos se apresenta, nesse ponto, com uma importante contribuição, posto que trata exatamente dessa questão.

A gestão de projetos, com seu conjunto de práticas normalmente aceitas e recomendadas pelo mercado, norteia sua atuação pelo conjunto de recomendações descritas pelo PMI-*Project Management Institute* (referência mundial em gerenciamento de projetos) em sua publicação conhecida por PMBOK-*Project Management Body of Knowledgement*, e subdivide a atividade de gestão de projetos em nove gerências, brevemente descritas a seguir.

- Gerência de integração: descreve os processos requeridos para certificar-se que os vários elementos do projeto estão propriamente coordenados.
- Gerência do escopo: descreve os processos requeridos para garantir que o projeto atenderá estritamente as necessidades acordadas com o cliente, nem mais, que poderia prejudicar o tempo e recursos originalmente previstos, e nem menos, o que poderia inviabilizar a solução.
- Gerência do tempo: descreve os processos requeridos para garantir que o projeto seja completado dentro do prazo.
- Gerência de custo: descreve os processos requeridos para que o projeto seja executado dentro do orçamento aprovado.
- Gerência da qualidade: descreve os processos requeridos para garantir que o projeto vai atender as necessidades descritas pelo cliente, bem como adequar-se a todos os dispositivos legais que regulam o negócio suportado.
- Gerência de recursos humanos: descreve os processos requeridos para o uso racional das pessoas envolvidas no projeto, em suas diversas fases, priorizando o emprego das competências adequadas a cada fase do projeto.
- Gerência de comunicações: descreve os processos requeridos para garantir rápida e adequada geração, coleção, disseminação, armazenamento e disposição final das informações do projeto.

- Gerência de riscos: descreve os processos relacionados a identificar, analisar e descrever a forma de responder aos riscos do projeto, em sua ocorrência.
- Gerência de aquisições: descreve os processos requeridos para adquirir bens e serviços relativos ou necessários à consecução dos objetivos do projeto.

A conjugação das melhores práticas da engenharia de *software* com a adoção das áreas do gerenciamento de projetos (ou pelo menos de parte delas) vêm sendo empregada com sucesso por muitas empresas, em uma tentativa de obter um maior controle sobre sua produção de *software*.

2.5 Avaliação de similaridade

Uma das principais necessidades da presente pesquisa é a adoção de um método que permita comparar a similaridade entre os documentos considerados. Para isso, foram avaliados alguns dos principais métodos e algoritmos usualmente aceitos pela comunidade científica:

2.5.1 Coeficiente de similaridade de Jaccard

O coeficiente de similaridade de Jaccard, também conhecido por índice de Jaccard é uma medida usada para comparar a similaridade entre dois conjuntos de dados. Na apuração desse coeficiente são consideradas tanto as igualdades, quanto as desigualdades.

A apuração do coeficiente e da distância de Jaccard são feitas através das fórmulas 2.1. e 2.2, respectivamente.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad (2.1)$$

onde: $J(A,B)$ *é o coeficiente de Jaccard entre os conjuntos* A *e* B*;*

Fórmula 2.1 – Apuração do coeficiente de Jaccard

$$J_d(A,B) = 1 - J(A,B) \qquad (2.2)$$

onde: $J_d(A,B)$ *é a distância de Jaccard entre os conjuntos* A *e* B*;*
$J(A,B)$ *calculado pela fórmula* (2.1)

Fórmula 2.2 – Apuração da distância de Jaccard

A avaliação de similaridade utilizando o coeficiente ou a distância de Jaccard fornece resultados iguais para a comparação entre os conjuntos A e B, independentemente da ordem

utilizada na comparação. Essa propriedade a torna especialmente útil quando a relação que se procura entre os conjuntos é de identidade; no caso da presente pesquisa, deseja-se que a comparação ocorra de maneira que, se A está contido integralmente em B, ainda que B seja muito maior que A, então a similaridade de A sobre B deverá ser igual a um.

Seja o exemplo mostrado na tabela 2.1:

	1	2	3	4
Seqüência 1 →	C	A	S	O
Seqüência 2 →	C	E	D	O

Tabela 2.1 – Exemplo do cálculo do coeficiente de Jaccard entre duas seqüências

O número de caracteres presentes na disjunção das seqüências 1 e 2 é igual a quatro, assim como a quantidade de caracteres presentes na união das seqüências é, também, igual a quatro. Nessa situação, tanto o coeficiente quanto a distância de Jaccard são iguais a meio.

2.5.2 Distância de Hamming

A distância de Hamming é uma medida da similaridade entre duas seqüências (ou dois vetores quaisquer) que representa o grau de dificuldade para se transformar uma seqüência na outra, através da aplicação de substituições, ou o número de erros que existe na comparação das duas seqüências

Tome-se o exemplo mostrado na tabela 2.2.

	1	2	3	4
Seqüência 1 →	C	A	S	A
Seqüência 2 →	C	A	P	A
Substituições necessárias	-	-	s	-

Tabela 2.2 – Exemplo da distância de Hamming entre duas seqüências

Nesse exemplo, basta apenas uma substituição na coluna três da seqüência 2 para transformá-la na seqüência 1. Isso equivale a dizer que a distância de Hamming entre as duas seqüências é igual a um.

2.5.3 Distância de Levenshtein

A distância de Levenshtein é uma métrica para apuração da similaridade entre seqüências, baseada na distância de Hamming, mas que admite, porém, um conjunto maior de operações sobre as seqüências. Além da operação de substituição utilizada na medição por Hamming, são consideradas nessa proposição as operações de inclusão e de exclusão de

elementos. O exemplo da tabela 2.3 ilustra essas operações:

	1	2	3	4	5
Seqüência 1 →	C	A	S	A	L
Seqüência 2 →	C	A	P	A	-
Operações necessárias	-	-	s	-	i

Tabela 2.3 – Exemplo da distância de Levenshtein entre duas seqüências

Tem-se, nesse exemplo, a necessidade de substituir um caractere (s), na coluna três e de inserir um caractere (i) na coluna 5 para transformar a seqüência dois na seqüência um, e portanto, a distância de Levenshtein para esse caso é dita como igual a dois.

2.5.4 Distância de Damerau-Levenshtein

Esse modelo é uma extensão ao modelo proposto por Levenshtein, com a agregação da operação de transposição (permutação da posição) entre dois elementos contíguos da mesma seqüência, conforme exemplificado na tabela 2.4.

	1	2	3	4	5
Seqüência 1 →	L	A	P	S	O
Seqüência 2 →	C	A	S	P	A
Operações necessárias	s	-	t	-	i

Tabela 2.4– Exemplo da distância de Damerau-Levenshtein entre duas seqüências

Neste exemplo, tem-se uma distância total igual a três, pois a transposição do caractere presente na coluna três é suficiente para igualar os conteúdos dessa coluna e também da coluna quatro; temos, portanto, a realização de três operações sobre a seqüência dois para transformá-la na seqüência um: uma substituição (s), uma transposição (t) e uma inclusão (i). É interessante notar que os métodos de Hamming e Levenshtein exigem, no mínimo quatro operações para fazer a mesma transformação.

Esse modelo, ou alguma de suas variações, é comumente utilizado em aplicações que pretendam determinar a similaridade entre duas seqüências. Notadamente no campo da genética, a comparação de cadeias de genomas o utiliza intensamente para determinar a similaridade entre amostras de DNA, normalmente representadas por um seqüência de letras (A-T-C-G-U) que correspondem às bases hidrogenadas ali presentes.

O modelo oferece também um importante subsídio para a comparação de vetores, podendo ser utilizado como base para a determinação da qualidade da ordenação de seus elementos em relação à um outro vetor conhecido.

3 O Estado da arte

O propósito deste capítulo é apresentar, em complemento à fundamentação teórica explanada no capítulo anterior, trabalhos de pesquisa que versam sobre assuntos correlatos e que propõem soluções para problemas que guardem semelhança com os objetivos deste trabalho, em especial aqueles que tratem dos modelos de IA e de processamento de linguagem natural que serão usados na construção do modelo proposto. Pretende-se que essa apresentação seja capaz de apresentar uma visão geral sobre os temas aqui tratados e fornecer subsídios para a escolha por uma ou outra metodologia que será usada no suporte ao presente trabalho de pesquisa.

Embora as três linhas de inteligência artificial apresentadas no capítulo anterior pudessem ser utilizadas para a construção de um modelo que atendesse aos objetivos propostos neste trabalho, a linha de raciocínio baseado em casos foi inicialmente escolhida para seu desenvolvimento, pelas suas características e similaridade com os principais processos envolvidos na pesquisa e, portanto dedicaremos aqui especial atenção aos recentes trabalhos que vêm sendo conduzidos nessa área.

3.1 Pesquisas em raciocínio baseado em casos

Este tópico trata de pesquisas relacionadas ao modelo de raciocínio baseado em casos que, por sua abrangência ou completude quanto aos estudos conduzidos e quanto aos relatos apresentados são citações freqüentes em trabalhos na área. Procura-se aqui destacar os trabalhos que, mantendo essas premissas, estejam de alguma forma relacionado com os objetivos da presente pesquisa ou que possam apresentar alguma contribuição significativa na construção do modelo proposto.

Uma das referências mais freqüentes quando se trata de raciocínio baseado em casos (CBR, do inglês *Case Based Reasoning*) é o trabalho de Aamodt (1994), no qual são descritas e avaliadas algumas variações de metodologia e aproximações geralmente empregadas em sua utilização. Esse trabalho faz uma extensa revisão sobre as principais metodologias e aplicações de CBR, descrevendo e analisando as etapas envolvidas que, conforme o autor, podem ser hierarquizada em um conjunto de procedimentos e métodos para representar o conhecimento, recuperar as informações, reusá-las, revisar e reter as informações necessárias.

Os tópicos a seguir explanam sinteticamente as idéias apresentadas pelo autor em seu trabalho:

Representação de casos: Um mecanismo de raciocínio que atue sobre casos é extremamente dependente dessa coleção de casos. Uma vez que se pretende obter a solução para um problema a partir de experiências passadas é necessário que o procedimento de busca e de reconhecimento de similaridade sejam eficientes, tanto a nível de qualidade dos resultados quanto do tempo consumido em sua busca. Isso torna a representação dos casos a principal questão primária a ser resolvida, posto que é necessário construir um conjunto de informações que permita, ao mesmo tempo, uma indexação apropriada e uma organização tal que sua recuperação seja útil na solução dos novos problemas;

Recuperação de casos: A recuperação de um caso parte da descrição de um novo problema e termina quando a melhor solução, entre as disponíveis na base, é encontrada. Essa atividade depende do estabelecimento de objetivos claros sobre quais informações são relevantes para o problema e, essa definição em si, é dependente do contexto em que se insere a aplicação. Nesse ponto é preciso definir também qual o tipo de similaridade a ser considerada, já que existe a possibilidade de considerar similaridades a níveis sintáticos, semânticos e de funcionalidades. O processo geralmente é dividido em etapas que envolvem a identificação das funcionalidades, com o conseqüente descarte de elementos pouco significativos, da determinação de um conjunto inicial de soluções candidatas e de uma seleção, nesse conjunto, daquela ou daquelas que pareçam oferecer a melhor solução, ou a melhor seleção para o caso mais próximo do atual.

Reuso de casos: Essa etapa envolve duas possibilidades distintas; a cópia da solução, indicada quando a melhor solução encontrada puder ser diretamente aplicável na solução do novo problema, ou a adaptação da solução anterior. No caso de adaptação da solução, descreve-se a possibilidade de adaptar tanto os resultados quanto os métodos utilizados para solucionar o caso anterior.

Revisão dos casos: O método para a revisão dos casos envolve a avaliação da solução gerada e o aprendizado com seu sucesso ou falha. Quando uma solução é aplicada com sucesso é necessário reter essa informação e utilizá-la para orientar novas buscas sobre a base de conhecimentos. Ao contrário, quando um solução apontada resulta em falha é preciso determinar as razões dessa falha que podem estar ligadas à representação inadequada do caso na base, ou até mesmo a uma mudança de contexto. A detecção de uma falha pode dar origem a explanações sobre sua ocorrência que também poderão ser reusados no futuro, seja na previsão de novas falhas ou na adoção de medidas apara evitá-las ou tratá-las mais facilmente.

Retenção de casos: O processo de retenção de casos, também designado como processo de aprendizado, consiste em se apropriar dos resultados anteriores para melhorar a qualidade das soluções obtidas pelo sistema, o que pode ser feito tanto nos casos de sucesso quanto nas falhas. A primeira etapa desse processo consiste na extração do conhecimento que pode ser representado pelo conjunto formado pelo problema descrito e pela solução obtida pelo sistema. É preciso avaliar qual o mecanismo utilizado para a obtenção da solução para determinar se deverá ser feita a inclusão desse novo caso na base, se o caso utilizado deverá ser atualizado ou generalizado para incluir a solução obtida ou mesmo se nenhuma providência precisará ser tomada. Essa decisão sobre a retenção é, geralmente, extremamente dependente do domínio da aplicação. É preciso considerar, ainda, os mecanismos de indexação e de integração de novos casos a base de dados que, como parte final do processo devem ser pensados de forma a manter as características do sistema em relação à qualidade dos resultados obtidos e da quantidade de tempo consumido para a pesquisa.

As pesquisas versando sobre CBR foram importante apoio na construção deste trabalho. De fato, boa parte dos processos usualmente descritos para o suporte de processos baseados em CBR estão presentes na solução que se idealiza.

3.2 Pesquisas em Textual Case Based Reasoning

Derivada da linha de pesquisas em raciocínio baseado em casos, a área denominada por *Textual Case-Based Reasoning* ou simplesmente TCBR se apresenta como uma opção para o tratamento de situações em que os casos não podem ser descritos simplesmente em termos discretos ou discretizáveis. Para isso são propostas abordagens que visam processar os elementos textuais e extrair deles as informações relevantes para a composição e identificação de casos.

As principais referências em trabalhos com TCBR, citadas em diversas outras publicações são os trabalhos conduzidos por Lenz(1998), Brüninghaus(2001).

O trabalho de Lenz, estabelece uma importante comparação entre TCBR e os mecanismos tradicionais de recuperação de informações (IR, do inglês *Information Retrieval*). O resultado dessa comparação é mostrado na tabela 3.1 e têm especial significado para essa pesquisa, tendo em vista que o processamento de informações dispostas em blocos de texto é uma das situações que, obrigatoriamente, deverá ser tratada.

	IR	Textual CBR
Representação dos documentos	Conjunto de termos e índices obtidos a partir de avaliações estatísticas	Conjunto de funcionalidades estabelecias durante a aquisição do conhecimento
Medida de similaridade	baseada em cálculos sobre a freqüência dos termos	Baseada na teoria do domínio
Aplicação a novos domínios	fácil	requer aquisição de conhecimento
Conhecimento do domínio	Não é considerado	Requerido
Informação não textual	Não pode ser usada	Pode ser integrada
Avaliação	Bem definida	Não suficientemente tratada, ainda

Tabela 3.1 – Resumo da comparação entre IR e Textual CBR (Lenz,1998)

A análise das informações dessa tabela apontam que a aplicação de TCBR pode permitir uma avaliação mais rica em detalhes na comparação de elementos de texto que a abordagem tradicional usada por IR, notadamente por incluir elementos do domínio específico da aplicação. Essa característica, ao mesmo tempo em que favorece a obtenção de melhores resultados estabelece um conjunto de limitações quanto a sua expansão a outros domínios e quanto a necessidade de aquisição de conhecimento, o que requer, em geral, a participação de especialistas na avaliação ou supervisão das etapas de treinamento.

A partir dessas conclusões, Lenz prossegue sua avaliação sobre o emprego de técnicas de processamento de linguagem natural (NLP – do inglês *Natural Language Processing*), explorando o emprego de dois grupos:

a) Técnicas sofisticadas: sobre as quais ele conclui que, além de dispendiosas em termos de consumo de recursos computacionais, apresentavam problemas para o tratamento de documentos que não estavam adequadamente formatados sob o ponto de vista gramatical ou que apresentavam muitos termos desconhecidos nos dicionários em uso;

b) Técnicas simples (ou rasas): que em, contraposição às sofisticadas, apresentaram melhores resultados, apesar da necessidade de muito esforço manual na classificação dos termos.

Relata-se no artigo que o emprego de técnicas de TCBR estavam, naquele momento, sendo utilizadas em três projetos conduzidos pelo grupo com avaliação positiva pelos respectivos usuários.

Brüninghaus, em uma linha de trabalho que complementou as pesquisas de Lenz, aprofundou as questões levantadas e analisou o papel da Extração de informações (IE – do

inglês *Information Extraction*) no contexto da aplicação de TCBR. Em sua abordagem, foi considerada a importante evolução das técnicas de NLP que, segundo o autor, são capazes de gerar vocabulários sobre domínios específicos a partir de um conjunto pequeno de amostras. O artigo relata o uso da ferramenta AutoSlog, proposta por Rillof(1996), nos experimentos realizados por Brüninghaus em busca da identificação de informações relevantes em trechos de documentos, demonstrando como o estabelecimento de um conjunto de regras permitia a recuperação adequada das informações desejadas. Cabe aqui ressaltar que da leitura do artigo depreende-se que a quantidade de esforços necessárias para a construção desse conjunto de regras não foi pequena, assim como também a dependência das regras em relação ao domínio da aplicação é fortemente evidenciada.

Apesar do esforço evidenciado no artigo, os resultados obtidos com essa opção de processamento dos elementos textuais foram considerados bem-sucedidos pela autora que os utilizou para alimentar uma aplicação denominada SMILE (do inglês *SMart Index Learner*) e cuja função principal era a determinação, sobre uma base de textos, de um conjunto de informações relevantes para tratamento pelo *framework* CATO (Aleven, 1997), que é um sistema de CBR usado para ensinar estudantes de direito a construir argumentações legais a partir casos representados por sentenças judiciais.

A avaliação dos resultados indicou o atingimento dos objetivos desejados, comprovando a idéia que o processamento a que foi submetido o texto teve a capacidade de manter e explicitar as informações relevantes que deveriam ser identificadas para o correto entendimento dos casos.

3.3 Pesquisas em processamento de linguagem natural

Datam da década de 50 os primeiros trabalhos de pesquisa envolvendo o processamento de linguagem natural, que tinham por objetivo estabelecer melhorias no processo de comunicação entre seres humanos e computadores, permitindo que a interação se desse de forma mais natural possível.

Essa idéia ganhou notoriedade no filme “2001-Uma odisséia no espaço” de Stanley Kubrick, baseado na obra de Arthur C. Clarke, na forma do robô HAL 9000 que interagia com os demais personagens em diálogos conduzidos em linguagem natural. Exageros à parte, o pequeno robô sintetiza os desafios dessa área de pesquisa que consistem, basicamente, em dotar sistemas computacionais da capacidade de interagirem com seres humanos em sua linguagem natural, ao contrário da interação habitual em que o ser humano é que precisa

transformar em linguagem compreensível pelas máquinas os elementos dessa interação.

Processar elementos descritos em linguagem natural envolve, no entanto, uma série de atividades complexas, que partem da compreensão das estruturas lingüísticas na direção da extração de seu significado principal. Há que se considerar toda sorte de fatores que interferem na construção de uma frase, sentença ou parágrafos descritos dessa forma, desde variações de vocabulários entre os interlocutores para descrição do mesmo objeto até a ocorrência de erros gramaticais ou léxicos.

É preciso, portanto, considerar uma série de etapas nesse tratamento e que cada uma dessas etapas pode ter uma significância diferente dentro do contexto da extração da informação. De acordo com Navaux(2008?) essas etapas podem ser resumidas da seguinte forma:

Análise morfológica que busca identificar palavras ou expressões isoladas em sentenças e classificá-las de acordo com sua categoria gramatical. Durante a análise morfológica tratam-se, também, de substituições que mantenham o sentido do texto e da redução da flexão de termos (plural ou gênero, por exemplo);

Análise sintática que avalia a construção das frases em busca do reconhecimento de padrões de construção da linguagem e adequação a regras gramaticais. Essa avaliação é extremamente importante face a ambigüidade que pode existir na construção de frases, períodos e orações. Essa análise pode ser baseada em gramáticas regulares, gramáticas livres de contexto e gramáticas sensíveis ao contexto, entre outras de menor expressão.

Análise semântica que busca a identificação do significado das palavras, frases e sentenças e se desdobra em semântica léxica e sem semântica gramatical. A compreensão da relação entre as palavras pode ser tão importante quanto à compreensão da palavra, em si, dependendo do contexto em que se faz a análise.

Análise pragmática que corresponde a interpretação do todo e complementa as análises anteriores. A análise pragmática busca a diferenciação entre os elementos que tenham relevância ou não para a compreensão do texto, ou de elementos que modifiquem o sentido que uma frase parece ter originalmente.

A composição entre essas etapas (e outras não citadas aqui) leva a construção de estruturas voltadas à compreensão do texto que permitem estabelecer parâmetros de classificação e aproximar sua compreensão através do uso de padrões.

Esse processamento é, no entanto, um processo extremamente complexo e, face as extensas variações e possibilidades suportadas pelas regras gramaticais de cada idioma, ainda oneroso em termos de consumo de recursos computacionais e humanos na identificação e

validação de regras de processamento. Apesar desse custo, os sistemas voltados ao processamento de linguagens naturais apresentam bom desempenho geral, principalmente quando o domínio de sua aplicação está adequadamente mapeado.

A avaliação das pesquisas sobre TCBR e sobre o processamento de linguagem natural foi decisiva na escolha da metodologia a ser adotada no tratamento dos elementos de texto que estão presentes no modelo proposto: a necessidade de um conjunto de informações muito maior sobre o domínio e a dificuldade na aquisição de um vocabulário apropriado, associados aos resultados mostrados levaram a decisão pela adoção de um modelo baseado no modelo espaço vetorial pela abordagem TF-IDF, com o enriquecimento incremental de termos do domínio e pré-processamento para eliminação ou substituição de alguns vocábulos.

3.4 Aplicações de IA em desenvolvimento e manutenção de software

A área de engenharia de *software* têm sido objeto de muitas pesquisas e freqüentes contribuições são encontradas na literatura, principalmente em áreas que visam a padronização e adequação de processos à modelos de referência.

Em um artigo que se propõe a dar uma visão geral da aplicação de inteligência artificial a ambientes de engenharia de *software*, Silva (2005) faz uma extensa abordagem desses conceitos.

Ambientes de engenharia de *software* são descritos como um conjunto de ferramentas capaz de fornecer apoio automático para as atividades comumente presentes nos processo de desenvolvimento e manutenção de *software*, tais como especificação, desenvolvimento e reengenharia de *software*.

O estudo categoriza os ambientes de acordo com três tipos principais de modelos empregados nesse processo.

Modelos abstratos que fornecem moldes de solução para problemas comuns, em nível de detalhamento não diretamente associado a uma organização específica;

Modelos instanciados ou executáveis, que são modelos prontos para serem submetidos a execução por uma máquina de processos. Um modelo instanciado é considerado como uma instância de um modelo abstrato.

Modelos executados que mantém registros sobre execuções passadas de determinados processos, incluindo eventos e modificações realizadas no modelo associado.

São abordados no artigo, também, diversas linhas de pesquisa na área de inteligência artificial e suas principais características e, em seguida alguns modelos de ambientes que

fazem uso dessas técnicas para apoio ao desenvolvimento e manutenção de *software*.

Como conclusão desse trabalho, Silva observa que o apoio de técnicas de inteligência artificial apresentam grande potencial de contribuição, que já se materializa nos ambientes avaliados e que podem gerar subsídios importantes para a melhoria do processo através do controle de recursos e de direitos de acesso, da coleta de métricas e disponibilização de análises sobre seu desempenho.

Um outro estudo, em área correlata à presente pesquisa foi conduzido por Shirabad (2001) e trata da aplicação de técnicas de mineração de dados para melhoria do processo de manutenção do que o autor chama de *software* Legado.

O artigo apresenta a manutenção de *software* em sistemas legados como uma atividade dispendiosa e como grande consumidora de tempo e recursos nas organizações, descrevendo a experiência da aplicação de métodos de *mineração de dados* como uma ferramenta capaz de auxiliar nessa tarefa. A abordagem do artigo passa pelo descobrimento de relações relevantes de manutenção, que, em geral, não estão expressas, seja pela falta de documentação ou por sua não condizência com a situação atual dos programas.

De acordo com o autor, sistemas legados fazem parte da realidade da comunidade de desenvolvimento do *software*, e ainda estão presentes em muitas atividades essenciais na sociedade moderna. Esses sistemas foram desenvolvidos há muito tempo atrás, e representam algumas centenas de bilhões de linhas de código, de forma que sua substituição não é uma tarefa simples.

A manutenção desses sistemas é, portanto, uma atividade imprescindível e com alto grau de dificuldade e, apesar de representar importante aspecto do ciclo de vida dos produtos e da abundância desse tipo de *software*, pouco esforço acadêmico vem sendo direcionado nesse sentido.

A base para o estudo realizado foi um sistema utilizado pela empresa Mitel corporation, utilizado em um sistema de telefonia (PBX) denominado SX 2000, originalmente criado em 1983. O *software* de gerenciamento desse sistema foi desenvolvido utilizando as linguagens Mitel Pascal e assembly. O código do sistema era composto por, aproximadamente, 1,9 milhão de linhas de código, distribuídas em mais de 4700 arquivos.

As informações sobre a atualização de código fonte eram controladas por um outro *software*, também desenvolvido pela MITEL, chamado SMS. Nesse sistema, ficavam armazenadas as informações sobre as intervenções já realizadas no sistema, tais como a história de problemas detectados e a solução aplicada a cada um deles. O uso do SMS permitiu construir um histórico das atualizações aplicadas ao sistema relacionando cada um

dos arquivos envolvidos nesse processo.

Outra ferramenta utilizada no estudo, voltada à exploração do código foi um programa chamado TKSee, utilizada para registrar a interação do usuário (programador) com os módulos do sistema.

Conforme os autores do artigo, uma das mais importantes perguntas envolvidas na manutenção de um sistema, e que precisa ser respondida para uma intervenção eficiente é descrita por: “Sobre que outros arquivos ou rotinas eu também devo ter conhecimento, ou, o que mais pode ser relevante para esse trecho de código ?”

Essa questão precisa, de fato, ser respondida inúmeras vezes a cada dia, em diferentes locais e organizações, e a habilidade para respondê-la é essencial para a atividade de manutenção de *software*. Na busca de solução a essa questão, foi direcionada a pesquisa, que aplicou uma mescla de técnicas de IA e de engenharia de *software*, conforme descritas nos itens a seguir.

O estudo envolveu duas áreas da engenharia de *software*, e tentou combinar técnicas aplicadas a cada uma delas para estabelecer de uma maneira mais clara, precisa e simples o relacionamento entre os diversos módulos que compunham o sistema. Essas técnicas estão descritas, de forma simplificada.

Knowledge Based Software Engineering (KBSE): A área assim denominada aplica conceitos de inteligência artificial para auxiliar o desenvolvimento de *software*, tendo a capacidade de auxiliar os programadores na representação e na dedução de relações entre os diversos componentes de um sistema. Essa abordagem tende, no entanto, a demandar muitos recursos computacionais e depender de uma base extremamente rica em conhecimentos, na avaliação dos autores.

Inductive Methods in Software Engineering : A outra área aplicada no estudo foi a utilização de métodos indutivos em engenharia de *software*, a qual apesar de vir recebendo menos atenção, pode ser uma ferramenta importante para encontrar novas relações entre conceitos. A utilização dessa técnica em manutenção de sistemas abre a possibilidade de um sistema incorporar conhecimentos através da exploração de exemplos, examinando casos resolvidos com sucesso sem a participação de um especialista. Espera-se, no entanto, que o ganho obtido por um modelo construído dessa forma seja menor que o de um modelo construído com a participação de um especialista.

O ponto de partida para o estudo foi a exploração das atividades de manutenção realizadas no código fonte. O objetivo era identificar, a partir desses registros, relações de relevância entre a manutenção dos diversos objetos, tentando obter interconexões intrínsecas

entre elementos; a maior dificuldade, segundo o estudo, é que essas relações existem entre objetos que podem ou não estar documentados e que tenham sido geradas em momentos diferentes do ciclo de vida dos projetos – como conseqüência do projeto original ou de manutenção feitas posteriormente.

O primeiro passo, consistiu em definir o conceito de "Relevância", a partir dos dados disponíveis, e isso foi tratado como um problema de classificação. Foi necessário definir classes, ou categorias, e associar cada instância do conceito como uma classe.

O Estudo foi feito com duas classificações distintas: a primeira agrupava os dados em três classes distintas (Não relevante, potencialmente relevante e relevante), enquanto a segunda simplificava essa classificação, usando apenas duas classes (não relevante e relevante).

O processo consistiu, então dos seguintes passos:

- Encontrar pares de objetos que eram relevantes, potencialmente relevantes ou não relevantes, entre si;
- Criar os conjuntos de dados para o aprendizado de máquina;
- Executar o algoritmo de aprendizado de máquina para induzir o conceito de relevância.

O conjunto de dados para treinamento foi dividido em três grupos distintos, conforme o período de tempo utilizado na coleta, e a cada um deles aplicou-se uma das heurísticas descritas a seguir:

Co-presence: Identifica que dois objetos são potencialmente relevantes, se foram verificados na mesma sessão de usuário;

Co-update: Dois objetos são relevantes uma para o outro, se os dois foram atualizados concomitantemente, e

Non-Relevance: Dois objetos que nunca foram objeto de atualização simultânea são não relevantes, um para o outro.

A aplicação dessas heurísticas sobre um conjunto de dados formado pelos mesmos pares de objetos poderia levar, então, a um conflito de classes, quando uma heurística associasse um par com um classe diferente daquela atribuída por outra heurística. Para a solução desse problema, foi proposta uma estratégia que propunha a prevalência de uma classe sobre outra, conforme indicado abaixo:

- Relevante / Potencialmente relevante → Relevante

- Não relevante / Potencialmente relevante → Potencialmente relevante

A classificação dos dados e montagem da árvore de indução foi feita com o uso do algoritmo c5.0, da RuleQuest Research, que implementa solução comercial de uma evolução dos algoritmos c4.5 e jar48, ambos de domínio público, destinados a esse tipo de classificação.

Os resultados da pesquisa apontaram que, para os objetivos propostos, a classificação em apenas duas classes distintas, utilizando a discretização dos atributos numéricos permitiu a obtenção de resultados médios melhores que os demais, e a precisão na descoberta de relações relevantes (as que interessam ao processo de manutenção do *software*). Embora o artigo tenha sido apresentado inicialmente em 2001, a realidade quanto à existência de *software* legado, sobretudo nas grandes organizações, permanece praticamente inalterada. Boa parte dos sistemas utilizados no controle de atividades rotineiras, e que fazem parte da vida de grande parte das pessoas continua sendo suportada por aplicações escritas há muito tempo. Um exemplo concreto desse tipo de *software* e o controle de contas correntes, que, em praticamente todos os grandes bancos, é feita em sistemas antigos (a maioria escritos em linguagem Cobol).

A manutenção desse tipo de sistema é, e presumivelmente continuará sendo por um bom período de tempo, consumidora de grandes esforços das áreas de desenvolvimento de *software* das organizações, sobretudo pela inexistência de mecanismos capazes de auxiliar os desenvolvedores a identificar todos os objetos envolvidos em uma determinada atividade de manutenção.

A utilização de técnicas de IA, para a mineração dos registros de manutenções já efetuadas em busca de relacionamentos de relevância, tais como proposto no artigo aponta um caminho que pode levar ao racionamento desses esforços, bem como melhorar a qualidade das intervenções e reduzir o volume de erros encontrados nesse processo.

4 O modelo proposto

Este capítulo descreve em detalhes a construção do modelo idealizado para possibilitar uma melhoria no processo de desenvolvimento de *software* pela agregação de uma etapa destinada a auxiliar na gestão do conhecimento presente nesse processo. São apresentadas aqui todas as etapas cumpridas em seu desenvolvimento, em uma abordagem que parte do problema como um todo e o subdivide em elementos menores para facilitar seu tratamento e compreensão.

4.1 Nicho atacado

Este trabalho se apresenta como uma contribuição ao processo de desenvolvimento de *software*, aliando a preciosa contribuição que pode ser dada pela gestão do conhecimento aos processos tradicionalmente sugeridos pela engenharia de *software* e pela gestão de projetos.

Cabe ressaltar que toda atividade de controle e acompanhamento a que estão sujeitos os desenvolvedores de sistemas gera uma sobrecarga de trabalho, que impacta nos prazos para a construção dos aplicativos e é vista pelos técnicos como uma burocracia de pouca utilidade – especialmente quando são produzidas dezenas de formulários e documentos que jamais serão consultados. Desta forma, é especialmente importante que qualquer alteração, ou acréscimo, que se faça ao processo de desenvolvimento seja suportado por uma ferramenta que minimize essa sobrecarga e evidencie as vantagens da adoção do modelo sugerido.

A figura 4.1 ilustra a alteração proposta, com o acréscimo de mais uma linha de processos ao ciclo de desenvolvimento de *software* aos dois atualmente mais utilizados pelas empresas, que são a engenharia de *software* e a gestão de projetos, descritos no capítulo 2.

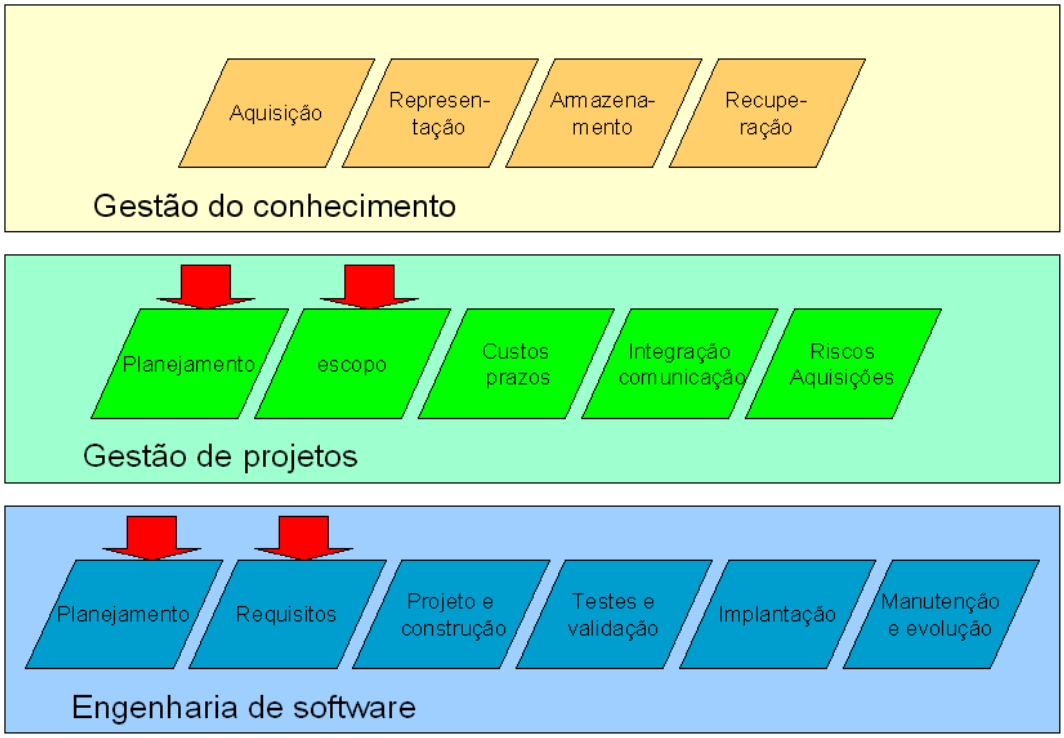

Figura 4.1 - Acréscimo de uma nova linha de processos para a gestão do conhecimento

Destaca-se que a adoção dessa nova linha implicará em alterações em alguns dos processos já existentes, notadamente nas fases iniciais.

Para suportar o modelo proposto e validar os resultados obtidos com sua utilização, propõe-se a construção de uma ferramenta que se apresente como uma alternativa às práticas de desenvolvimento de *software* usualmente empregadas, dentro de um ambiente propício à explicitação, representação e compartilhamento do conhecimento necessário para o desenvolvimento de aplicações de *software*.

Espera-se que tal ferramenta seja capaz de propiciar a gestores de produtos (clientes) e desenvolvedores de sistema o acesso ao conhecimento empregado na construção da solução, bem como o acesso a soluções previamente empregadas, tendo, desta forma, a capacidade de prover os subsídios necessários ao reuso de módulos, funções e outros componentes de *software*.

4.2 Metodologia

O trabalho se inicia pela confirmação dos pontos já identificados em que a gestão do conhecimento pode acrescentar maior contribuição ao processo de desenvolvimento de *software*. A partir destes pontos elaborou-se uma proposta de alteração no processo de interação entre as áreas demandantes (cliente) e desenvolvedora de soluções, para que o conhecimento explicitado durante essa etapa possa ser adequadamente registrado e reaproveitado.

Identifica-se, nas fases iniciais do processo a existência de uma grande interação entre especialistas no domínio do negócio que se pretende automatizar e da área de tecnologia. Essa interação apresenta uma situação desejável para que o conhecimento sobre o negócio e sobre a sistematização dos produtos e serviços envolvidos seja externalizado e registrado para futuro reaproveitamento. É importante registrar também o contexto no qual esse conhecimento se insere, posto que não é possível conceituar conhecimento dissociado de contexto.

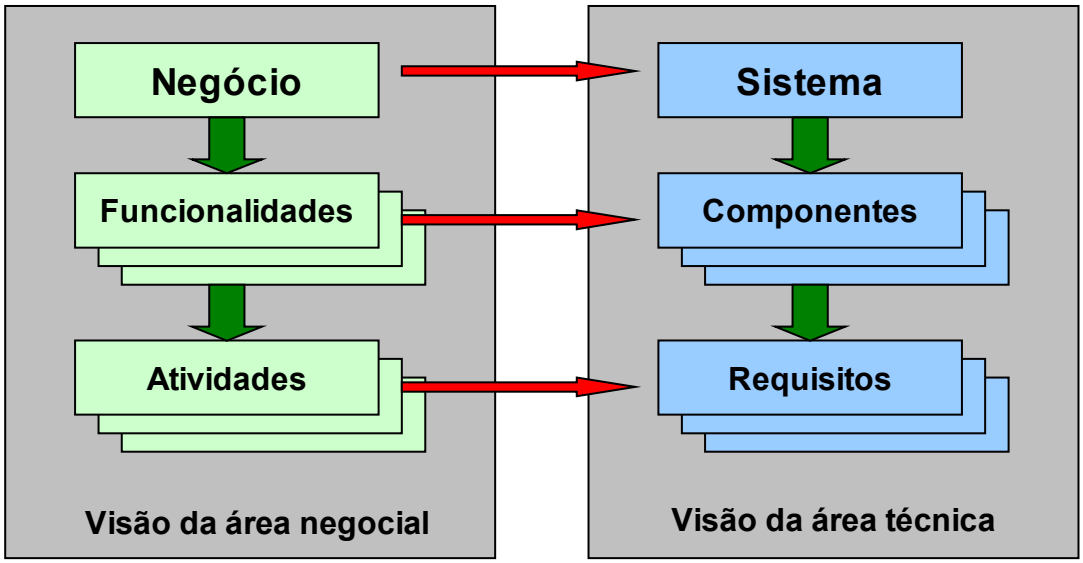

Figura 4.2 – Correlação entre as visões das áreas de negócio e técnica

A figura 4.2 ilustra a interação entre as áreas e as correspondências entre as visões de cada uma; enquanto a área negocial foca sua atenção nos objetivos negociais e nos resultados que se pretende obter para a organização a área técnica busca estruturar recursos tecnológicos (*hardware* e *software*) para viabilizar tal negócio.

Para suportar o modelo proposto, faz-se necessário o desenvolvimento de uma ferramenta de *software* capaz de produzir os benefícios esperados sem acrescentar uma sobrecarga adicional de trabalho. Uma importante decisão a respeito dessa ferramenta é relativa a sua própria construção; embora sua aplicação vise, inicialmente, suportar o

desenvolvimento de *software* para execução em mainframes e linguagens de programação tradicionais, os requisitos para sua construção recomendam a adoção de técnicas modernas, interface gráfica e execução em ambiente Web.

4.3 Descrição do modelo

O objetivo do modelo é a obtenção de um índice capaz de expressar a similaridade entre duas ou mais solicitações – a solicitação atual, que está em fase de elaboração e outras, recuperadas da base de casos.

Esse índice deve ser expresso através de um número real, variando no intervalo de zero até o valor máximo de um, representando, respectivamente, que nenhuma similaridade foi encontrada ou que as solicitações são idênticas para efeito dos atributos considerados na verificação. Essa identificação poderá então ser utilizada pelo demandante do serviço para vincular à sua solicitação informações sobre outras solicitações similares já atendidas (ou em andamento) e pela equipe técnica que, valendo-se dessa informação poderá intensificar o reuso de soluções no atendimento de novos pedidos.

Figura 4.3 – Fluxo de informações previsto para o modelo

A figura 4.3 apresenta, em termos gerais, o fluxo de interações para o modelo proposto que pode ser descrito pela seguinte seqüência de eventos:

1. O demandante inicia a solicitação de serviço de TI e a envia para o aplicativo de controle;
2. O sistema pesquisa por solicitações similares à situação atual e as apresenta, na forma de uma lista ordenada;
3. O demandante verifica as sugestões e vincula aquelas que poderão ser úteis ao desenvolvimento da solicitação, seja pelo reuso de componentes já desenvolvidos, seja por conter informações relevantes;
4. O sistema registra os vínculos selecionados;
5. Ao iniciar o atendimento da solicitação, o responsável técnico pelo atendimento (executante) da solicitação verifica os vínculos informados pelo demandante;
6. O executante registra na solicitação quais os vínculos que foram utilizados(se houver);
7. O executante registra as novas funcionalidades que foram desenvolvidas para o atendimento da solicitação (se houver) e, opcionalmente, converte a solicitação em um caso da base de conhecimentos;

A partir da análise desse conjunto de informações iniciais, tomou-se a primeira decisão a respeito do modelo, que consistiu na escolha do formalismo de Inteligência Artificial que seria utilizado para suportá-lo. A escolha recaiu sobre Raciocínio Baseado em Casos (CBR), justificada pelos seguintes fatores:

- O fluxo previsto para o modelo permite identificar claramente as atividades que deverão ser desempenhadas por seus componentes, entre as quais se destacam: a recuperação da informação, o reuso de soluções já preparadas, a avaliação da sugestão de reuso feita pelo sistema e a retenção de novos casos na base para consulta e uso nas próximas solicitações. Essas quatro atividades constituem o ciclo básico da metodologia e apontam para sua aplicação direta;
- Deseja-se que a similaridade seja expressa na forma de um índice. Ainda que a classificação quanto a existência, ou não, de similaridade fosse uma indicação razoável quanto a possibilidade de reuso de soluções anteriores, é desejável conhecer a magnitude dessa similaridade, permitindo, com isso, que as solicitações com maior índice sejam verificadas prioritariamente;

- O aprendizado de novos casos deve ocorrer da forma mais automatizada possível;
- Deseja-se que a intervenção de especialistas seja minimizada após a colocação do modelo em funcionamento.

Em contrapartida, é preciso considerar que devem ser tratados elementos não estruturados (descritos em linguagem natural) e que a abordagem de CBR, ainda que consideradas as extensões do modelo TCBR (*Textual Case Based Reasoning*) não se adapta aos objetivos propostos, principalmente, por exigir uma mapeamento detalhado dos termos utilizados no domínio da aplicação para apresentar resultados adequados.

Para o tratamento desses elementos, o modelo espaço vetorial apresenta características bastante adequadas, pois:

- Permite identificar similaridades entre elementos de texto baseado em critérios de semântica, sintaxe e de morfologia;
- Permite o processamento de um largo espectro de vocábulos ainda que não se tenha muito conhecimento sobre eles;
- Possibilita diferenciação entre a relevância de termos conforme sua freqüência nos documentos da base e atualização desse valor a cada novo caso agregado;
- Permite a obtenção de índice de similaridade entre documentos a partir dos termos presentes em cada um dos documentos;
- Permite fazer o enriquecimento da capacidade de identificação de similaridade, com a utilização de técnicas de pré-processamento do texto e a inclusão de dicionários de termos específicos do domínio da aplicação.

A análise desses fatores leva a conclusão que o modelo ideal deveria, assim, utilizar os formalismos de raciocínio baseado em casos para o processamento dos elementos estruturados e empregar técnicas baseadas no modelo espaço vetorial para o tratamento dos elementos descritos em linguagem natural. Essa constatação levou a construção de um modelo híbrido, no qual o índice final de similaridade será computado a partir dos índices parciais que indicarão a similaridade entre os elementos estruturados e entre os elementos não estruturados.

A observação do diagrama de atividades do modelo explicita, também, a existência de dois grupos de componentes: os componentes de interface, responsáveis pela interação de

demandantes e de executantes com o sistema de controle das solicitações; e, componentes de suporte à recuperação de informações para reuso.

Os componentes de interface subdividem-se em interface com o demandante e com o executante e podem ser entendidos como um conjunto de formulários (para serem preenchidos em tela) para o registro das informações. Esses componentes são melhor detalhados adiante.

Os componentes de suporte à recuperação de informações constituem-se no cerne do modelo e são detalhadamente descritos a seguir.

A primeira função básica que deve ser cumprida é a mensuração da similaridade da solicitação de serviços atual com outras existentes na base de conhecimento, e para isso é preciso compreender primeiramente a composição da estrutura da solicitação.

4.3.1 A solicitação de serviços de TI

Uma solicitação de serviços é um pedido formal, feito por um departamento vinculado a uma diretoria de negócios, para que seja desenvolvida uma aplicação, uma nova aplicação, uma nova funcionalidade ou para que seja alterada uma funcionalidade existente em uma aplicação para suportar tecnologicamente os negócio que são desenvolvidos pela empresa. Uma solicitação contém uma descrição de seus objetivos, das funcionalidades que devem ser implementas ou alteradas e elementos que caracterizam produtos relacionados, canais de disponibilização, riscos e benefícios envolvidos em sua adoção. Esses elementos podem ser subdivididos em dois grupos:

a) elementos estruturados → são aqueles que podem ser representados por valores numéricos, símbolos ou mnemônicos, cujo valor pode ser adequadamente previsto e mapeado;
b) elementos não estruturados → correspondem aos elementos que serão descritos em linguagem natural. Sobre esses elementos não é possível fazer qualquer previsão quanto aos valores e formatos que serão utilizados. Muitas vezes, em complemento às informações digitadas são utilizados anexos, que detalham ou complementam as informações das solicitações, também em formatos diversos.

A solicitação de serviços de TI é, portanto, o primeiro elemento de estudo do modelo, de forma que as informações possam ser adequadamente representadas, tanto para suprir as necessidades dos módulos de suporte ao reconhecimento de similaridades, quanto para suprir as necessidades de demandantes e executantes. Com base nesse estudo foram definidos os

elementos mostrados na tabela 4.1 para serem tratados.

Campo	Descrição / observações	Tipo
Objetivo do serviço solicitado	Descrição sucinta do serviço que está sendo solicitado	Não estruturado (linguagem natural)
Funcionalidades a serem implementadas	Descrever detalhadamente todas as funcionalidades sobre o serviço que está sendo solicitado	Não estruturado (linguagem natural)
Benefícios esperados com sua adoção	Registrar os benefícios esperados em relação aos aspectos financeiro, imagem, controle	Estruturado Valores tabelados
Riscos incorridos em caso de não atendimento	Registrar os riscos a que se sujeita a e organização em caso de não atendimento da solução. Avaliar os aspectos financeiro, imagem e controle	Estruturado Valores tabelados
Periodicidade	Informar a periodicidade de utilização da solução, tanto em relação à interface com os usuários, quanto em relação à execução de rotinas de atualização e de troca de informações	Estruturado Valores tabelados
Legislação	Informar os normativos relacionados ao serviço. Permitir a impostação de até cinco normativos, informando o tipo, o número ou referência e a data de publicação	Estruturado Valores tabelados
Principal sistema relacionado	Informar a sigla do principal sistema envolvido no atendimento da solicitação (se houver)	Estruturado Valores tabelados
Principais produtos relacionados	Informar quais os principais produtos envolvidos com a solicitação, entre os grandes grupos de produtos afetos aos negócios da organização	Estruturado Valores tabelados
Canais de disponibilização	Selecionar em uma lista previamente cadastrada quais os canais de disponibilização da solução	Estruturado Valores tabelados

Tabela 4.1 – Elementos estruturados e não estruturados que serão tratados pela aplicação.

4.4 Identificação de similaridade entre casos

Os componentes de suporte são elementos que sustentam o mecanismo de recuperação das informações e mensuração da similaridade entre os casos considerados. As funções que devem ser desempenhadas por esses componentes estão descritas nos itens apresentados a seguir.

4.4.1 Estrutura do caso

Um caso corresponde a uma solicitação de serviços de TI; temos então um conjunto de casos que já foram objeto de tratamento pelo sistema e constituem a base de casos sobre a qual são feitas as pesquisas. A solicitação de serviços que está em fase de elaboração, ainda não faz parte da base de casos do sistema, mas possui todas as informações para formar um novo caso. Essa solicitação é referida no decorrer desta pesquisa como “caso em estudo”.

Tomando por base a solicitação de serviços que está em elaboração pelo demandante (o caso em estudo), o sistema deve ser capaz de determinar sua similaridade com os demais casos registrados na base. Para fazer essa determinação devem ser considerados tanto os elementos estruturados quanto os não estruturados. Cabe aqui ressaltar a primeira decisão de projeto que foi segregar esses elementos e fazer o tratamento em separado de cada grupamento. Essa decisão foi tomada porque a comparação de elementos estruturados é uma tarefa muito mais simples que a comparação dos elementos não estruturados e pode funcionar como um filtro para que somente os casos pré-selecionados sejam completamente avaliados.

Cada caso contém, portanto, um conjunto de elementos estruturados e um conjunto de elementos não estruturados

4.4.2 Tratamento dos elementos estruturados

A tarefa sobre esses elementos consiste em determinar um índice de similaridade entre a nova solicitação, doravante denominada simplesmente como caso em estudo, e as demais solicitações registradas, as quais serão referidas por base de casos. A apuração desse desse índice deve considerar todos os elementos estruturados presentes na solicitação e levar em conta que cada um desses elementos poderá ter um contribuição diferenciada na composição do índice de similaridade. Com esse propósito, e atendendo a esses requisitos, foi construída o primeiro elemento de avaliação do índice de similaridade entre as solicitações, que será aqui definido como índice de similaridade entre elementos estruturados (ISee) cujo cálculo se dará pela fórmula indicada em (4.1).

$$ISee(A,X) = \sum_{i=1}^{i=num-elem} IS_i \;\; x \;\; Peso_i \qquad (4.1)$$

em que : $ISee_{(A,X)}$ é o índice de similaridade entre os elementos estruturados do caso A em relação a um caso X , recuperado da base ;
IS_i é o índice de similaridade entre o $i-$ésimo elemento estruturado de cada caso ;
$Peso_i$ é o peso relativo ao $i-$ésimo elemento dos casos ;

Formula 4.1– Calculo do índice de similaridade entre elementos estruturados.

Os pesos relativos à cada um dos elementos estruturados da solicitação serão inicialmente atribuídos com base na opinião de especialistas no domínio da aplicação e serão refinados durante a fase de validação e testes. Na implementação da ferramenta de suporte à aplicação esse fato deve ser considerado de tal maneira que a alteração dos valores possa ser feita facilmente.

4.4.3 Tratamento dos elementos não estruturados

Os elementos não estruturados constituem uma fonte extremamente importante na avaliação da similaridade e seu tratamento deve ser tal que permita obter a maior quantidade possível de informações relevantes entre as descrições constantes nos campos objetivo e funcionalidade da solicitação de serviços de TI. No entanto a descrição feita em linguagem natural apresenta uma série de dificultadores para que isso seja feito, entre as quais pode-se destacar:

- Utilização de sinônimos para descrever termos similares;
- Ausência de uma estrutura padronizada para descrever as funcionalidades desejadas;
- Possibilidade da ocorrência de erros de ortografia;
- Utilização de sinais gráficos (acentos, caracteres especiais, números);

Essas dificuldades podem (e devem) ser atenuadas pela adoção de mecanismos de limpeza e enriquecimento do texto. Esse mecanismo fará o pré-processamento de cada um dos elementos descritos em linguagem natural, buscando a eliminação de sinais inadequados, a redução de termos a seus sinônimos preferenciais e o reconhecimento de termos específicos do domínio da aplicação.

O esquema proposto para o pré-processamento do texto está indicado na figura 4.4, e é implementado em um dos componentes de suporte.

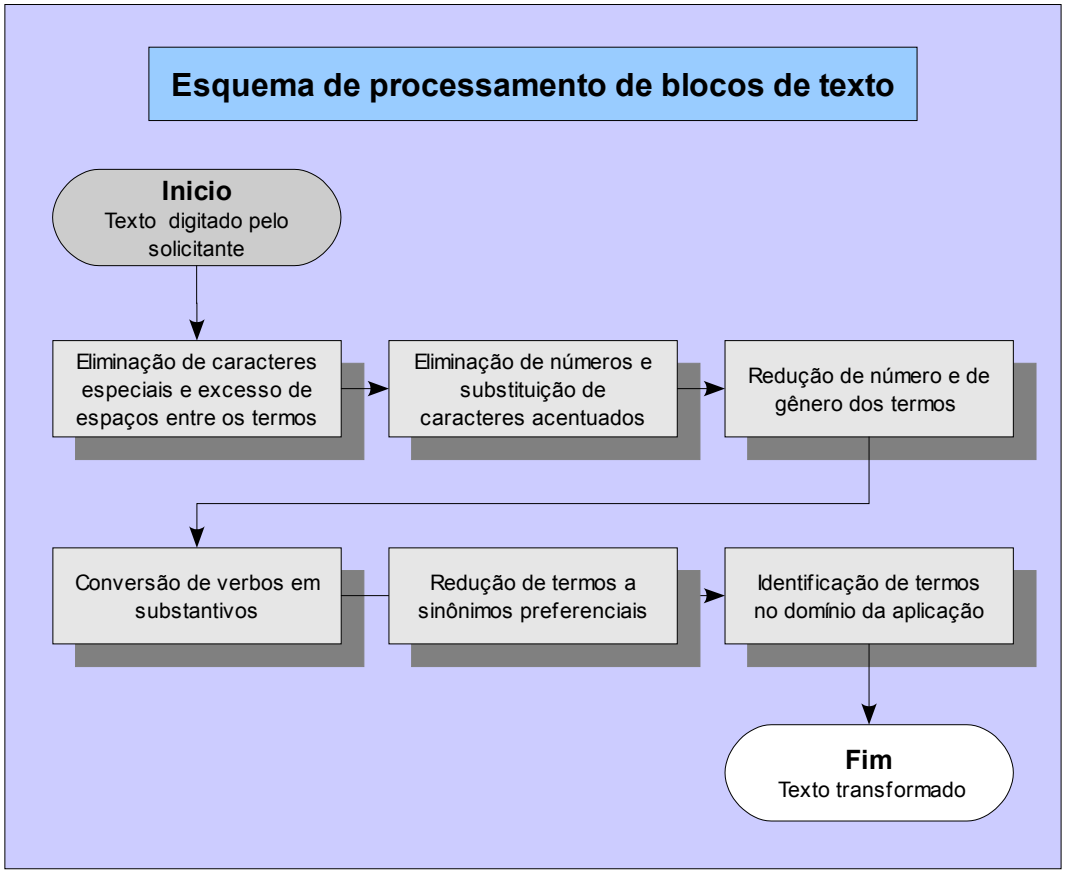

Figura 4.4 – Esquema de processamento de blocos de texto

A partir do texto resultante desse pré-processamento, que consiste em aplicar cada uma das etapas indicadas no esquema ao bloco de texto, como um todo, serão formadas as matrizes de termos (geralmente referidas na literatura como *bag of words*), compostas por um termo e um número que representa a freqüência desse termo no documento sob Análise.

Após a formação, essa matriz será comparada com as matrizes obtidas a partir do processamento dos campos objetivo e funcionalidades das solicitações que estão na base de conhecimentos do sistema, atuando como mandatária na formação das demais, ou seja, as matrizes das solicitações com as quais será comparada terão sua lista de termos limitada aos termos da matriz do caso atual. Essa definição permite encontrar correspondências diretas entre o texto da solicitação atual e fragmentos de texto de solicitações anteriores.

A transformação de um bloco de texto em uma matriz de freqüência de palavras (ou termos) está exemplificada na figura 4.5, e, assim como o pré-processamento do texto é implementada por um componente de suporte:

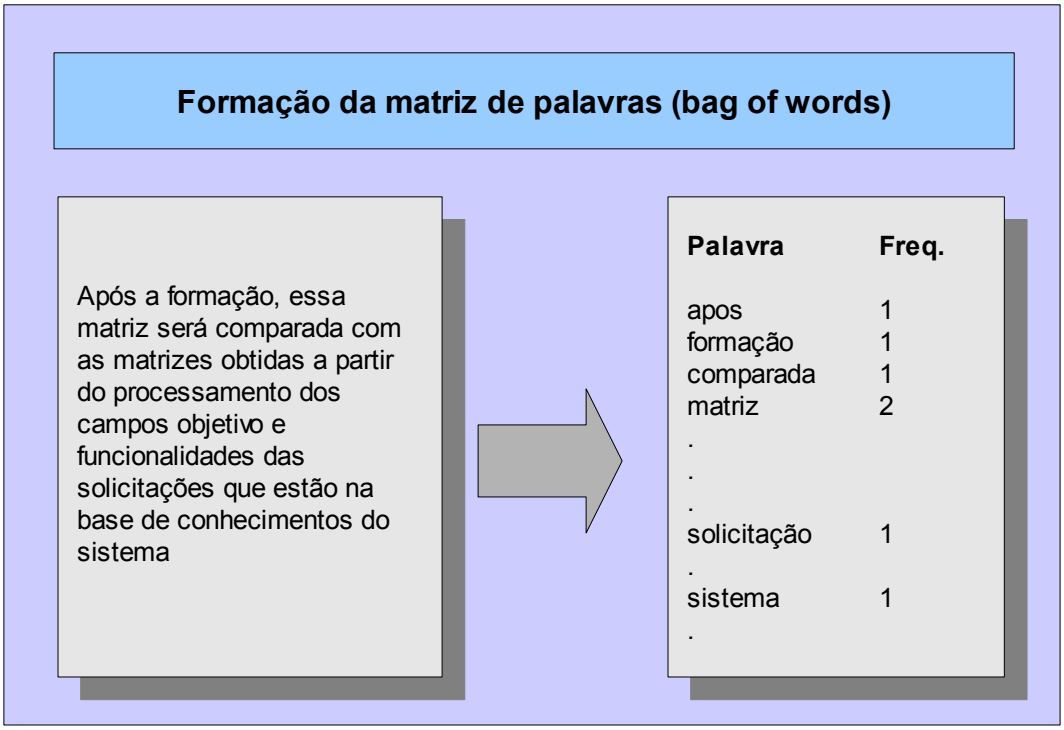

Figura 4.5 – Formação da matriz de freqüências de palavras a partir do texto

A construção da matriz de freqüência de palavras é a última etapa de pré-processamento antes da comparação dos elementos não estruturados das solicitações. Uma vez obtidos os termos e respectivas freqüências, e sabida a freqüência de todos os termos que já estão na base de conhecimentos, pode-se determinar a significância de cada um desses termos, aplicando fórmula descrita em (4.2), baseada nas proposições do modelo espaço-vetorial descritas no capítulo 2:

$$S_i(A) = tf_i(A) \quad x \quad IDF(T) \qquad (4.2)$$
$$S_i(X) = tf_i(X) \quad x \quad IDF(T)$$

em que : $S_i(A)$ *é o valor da significância do* $i-$*ésimo termo no caso em estudo ;*
$S_i(X)$ *é o valor da significância do* $i-$*ésimo termo no caso recuperado da base ;*
T *é o* $i-$*ésimo termo da matriz de frequência de palavras do caso ;*
$tf_i(A)$ *é a frequência do* $i-$*ésimo termo no caso em estudo ;*
$tf_i(X)$ *é a frequência do* $i-$*ésimo termo no caso recuperado da base ;*
$IDF(T)$ *é o valor da frequencia invertida do termo* T *, na base ;*

Fórmula 4.2 – Determinação da significância dos termos dos documentos a serem comparados.

O cálculo do valor do IDF é feito sobre todos os documentos registrados na base de conhecimentos, utilizando-se a fórmula (4.3):

$$IDF(T) = \ln \frac{\text{Qtde\_total\_dcto}}{\text{Qtde\_dcto} \subset \text{T}} \qquad (4.3)$$

em que : $IDF(T)$ é o valor que representa a frequência invertida associada ao termo t ;
Qtde_total_dcto *é a quantidade total de documentos na base ;*
Qtde_dcto $\subset$ T *é a quantidade de documentos que contenham o termo T ;*

Fórmula 4.3 – Determinação do IDF para cada um dos termos da base.

Uma vez definida a significância de cada termo em cada um dos documentos que estão sendo comparados, é feita a medida da similaridade entre eles, com a aplicação da fórmula (4.4), que identifica a distância entre as significâncias relativas.

$$IS_i(A, X) = S_i(A) \quad x \quad \sqrt{1 - \left(S_i(A) - S_i(X)\right)^2} \qquad (4.4)$$

em que : $IS_i(A, X)$ é o índice de similaridade do $i-$ésimo termo no caso em estudo , em relação ao caso recuperado da base ;
$S_i(A)$ é o valor da significância do $i-$ésimo termo no caso em estudo ;
$S_i(X)$ é o valor da significância do $i-$ésimo termo no caso recuperado da base ;

Fórmula 4.4 – Determinação do índice de similaridade entre termos de dois documentos.

A última etapa da avaliação dessa similaridade consiste em fazer um somatório dos índices de similaridade relativos de cada um dos termos da lista, obtendo o índice de similaridade global para os elementos não estruturados. A fórmula relativa a esta operação está descrita em (4.5):

$$ISne(A, X) = \sum_{i=1}^{i=qtd-termos} IS_i(A, X) \qquad (4.5)$$

em que : $ISne(A, X)$ é o Indice de similaridade entre os elementos não estruturados ;
$IS_i(A, X)$ é o indice de similaridade entre os termos do caso em estudo em relação aos termos do caso recuperado da base ;
$qtd-termos$ é a quantidade de termos da lista de palavras no caso em estudo ;

Fórmula 4.5 – Determinação do índice final de similaridade entre elementos não estruturados.

O índice ISne representa a similaridade existente entre os elementos não estruturados

nos dois casos comparados (referidos nas fórmulas como A e X) . A composição desse índice com o obtido para os elementos estruturados (ISee) é então efetuada para a obtenção do índice final de similaridade entre as dois casos comparados.

A composição entre os índices deve levar em conta a existência de pesos distintos para os elementos estruturados e não estruturados. Considerando todos esses elementos temos, enfim, a fórmula final da similaridade entre os casos, que está indicada em (4.6).

$$IS(A,X) = \frac{ISee(A,X) \;\; x \;\; Pee \;\; + \;\; ISne(A,X) \;\; x \;\; Pne}{Pee \;\; + \;\; Pne} \qquad (4.6)$$

em que : $IS_{(A,X)}$ *é o índice final de similaridade entre os casos A e X ;*
$ISee_{(A,X)}$ *é o índice de similaridade entre os elementos estruturados ;*
Pee *é o peso relativo aos elementos estruturados ;*
$ISne_{(A,X)}$ *é o índice de similaridade entre os elementos não estruturados ;*
Pne *é o peso relativo aos elementos não estruturados ;*

Fórmula 4.6 – Determinação do índice final de similaridade entre dois casos.

A etapa de identificação de casos similares se encerra quando todas as solicitações foram comparadas (considerando a existência do limiar para comparação dos elementos não estruturados) e montada uma lista, em ordem decrescente, das solicitações similares para apresentação ao demandante, que poderá então navegar entre as diversas solicitações e vincular as solicitações por ele escolhidas à nova solicitação que está sendo elaborada.

Apresenta-se no apêndice A deste documento um exemplo de como esses índices são calculados em função dos diversos elementos que compõem cada solicitação.

4.4.5 Registro da solicitação

Após a elaboração da solicitação pelo demandante, com a possível vinculação de casos similares, a solicitação deverá ser gravada na base de solicitações, com todos os seus elementos. Nesse momento, encerra-se a primeira etapa prevista para o ciclo do modelo descrito na figura 4.3..

O registro será feito através da gravação das informações em uma base de dados que conterá as informações registradas pelo demandante e estará acessível para verificação pelo executante.

É preciso ressaltar que, entre a impostação de uma nova solicitação de serviços e o início de seu desenvolvimento, existe uma negociação entre demandante e executante em relação à prioridade de atendimento, pois freqüentemente as demandas por novas soluções de TI para suporte aos negócios das organizações são em quantidade superior à capacidade de

atendimento.

Essa realidade não é, no entanto, objeto deste estudo e seu tratamento não será feito aqui. Apenas como referência, a área de gestão de projetos (abordada no capítulo 2) oferece um conjunto de métodos para auxiliar as áreas nessa definição de prioridades.

4.4.6 Atendimento da solicitação

Uma vez que a solicitação tenha seu atendimento autorizado, é designado um (ou mais) responsáveis por seu atendimento. Esse atendimento é feito, no modelo atual, por meio de uma seqüência de processos que resulta na construção da solução de TI para atendimento da solicitação. Na maioria das grandes organizações esse processo é controlado e obedece uma série de regras, muitas das quais são freqüentemente objeto de ações de auditoria que visam certificar sua aderência a padrões aceitos internacionalmente. Cita-se aqui, como exemplo, a inclusão de regras no "Acordo de Basiléia" que normatiza o funcionamento de instituições financeiras, para a verificação da conformidade dos processos de desenvolvimento de *software* por parte dessas instituições.

A engenharia de *software* (também abordada no capítulo 2) apresenta contribuições significativas nessa área, oferecendo um conjunto de metodologias e ferramentas para permitir o desenvolvimento seguro e aderente a padrões.

Uma das primeiras etapas técnicas do processo de desenvolvimento é o levantamento dos requisitos para a solução demandada, etapa esta que, muitas vezes, é feita com a participação direta do demandante, seja através de reuniões ou do preenchimento de questionários para elicitação das funcionalidades que, embora descritas na solicitação, freqüentemente não estão claras o bastante para seu atendimento.

Nessa etapa do processo, se encaixa uma nova fase, que antecede o levantamento de requisitos: trata-se da verificação de solicitações similares, que passa a ser possível graças a vinculação pelo demandante dos registros de solicitações que exibam algum grau de similaridade com a atual. Essa verificação somente ocorre no processo atual de desenvolvimento de *software* quando o responsável pelo atendimento da solicitação detém o conhecimento sobre a existência de tal similaridade, o que geralmente ocorre, quando houve sua participação no desenvolvimento.

Verificadas as solicitações vinculadas à solicitação atual, o responsável técnico pode, então, avaliar a possibilidade de reusar componentes desenvolvidos anteriormente, conhecer detalhes sobre as implementações anteriores – identificando riscos e oportunidades-, avaliar a existência de aspectos legais envolvidos, enfim, ter seu trabalho sensivelmente facilitado.

Uma vez que uma solução apontada pelo sistema pode ser reutilizada para o atendimento da nova solicitação é importante que essa informação seja também registrada no sistema. No ciclo clássico do CBR essa atividade é descrita como etapa de avaliação das soluções estimadas e provê uma informação que permite reforçar a indicação da similaridade entre o caso utilizado e o problema inicial, funcionando como um *feedback* do qual o mecanismo de identificação de similaridades pode se valer para melhorar seu índice de acertos.

Outra atividade importante que também se faz presente nesta etapa é o registro das funcionalidades desenvolvidas para o atendimento da solicitação, de detalhes sobre sua construção e de aspectos legais e negociais envolvidos. Esse registro será extremamente útil se, na avaliação da solicitação, o executante identificar que se trata de um novo caso que deverá ser armazenado na base de conhecimentos.

Identificamos nessa fase, portanto, a existência de três importantes atividades do ciclo do raciocínio baseado em casos: o reuso de soluções anteriores, a avaliação das soluções estimadas pelo sistema e o aprendizado de novos casos.

Essa etapa não demanda a construção de mecanismos complexos, mas somente de uma interface adequada para que o registro dessas informações possa ser feito de forma simples e direta, com o mínimo de acréscimos às atividades já desenvolvidas pela área técnica quanto ao controle das solicitações que estão sendo atendidas. Ressalta-se aqui que a existência de mecanismos de controle sobre o atendimento das solicitações já é uma praxe das áreas de tecnologia de grandes organizações, ainda que sua finalidade, em geral, esteja ligada a processos de controle e administração de recursos humanos e materiais.

4.5 Avaliação e suporte ao modelo

Propõe-se, para avaliação e suporte às funcionalidades do modelo, a construção de uma ferramenta capaz de automatizar o processo descrito. Essa ferramenta fundamenta-se nos princípios de gestão do conhecimento, engenharia de *software* e de inteligência artificial apresentados nos capítulos anteriores.

A idéia consiste em construir um ambiente de trabalho para que o cliente (demandante da solução) e o responsável técnico pela construção do *software* registrem, durante as fases de desenvolvimento, todos os aspectos relevantes e fatos importantes associados ao processo de desenvolvimento. Pretende-se que a ferramenta seja capaz de fazer a associações do tipo: necessidade negocial → solução de projeto → implementação de *software*; a esse conjunto

associam-se as definições legais e negociais que fazem parte da solução ou que tiveram impacto direto ou indireto em sua adoção, e ainda os problemas encontrados durante sua modelagem e construção.

Esse conjunto de informações, devidamente contextualizadas, será então armazenado em uma base de conhecimentos que integra o ambiente da ferramenta, permitindo o rápido acesso a elementos similares utilizados no projeto atual ou em projetos já desenvolvidos, propiciando o reuso de soluções e evitando que o direcionamento dado para uma determinada solução se depare com os mesmos problemas já enfrentados em situações similares.

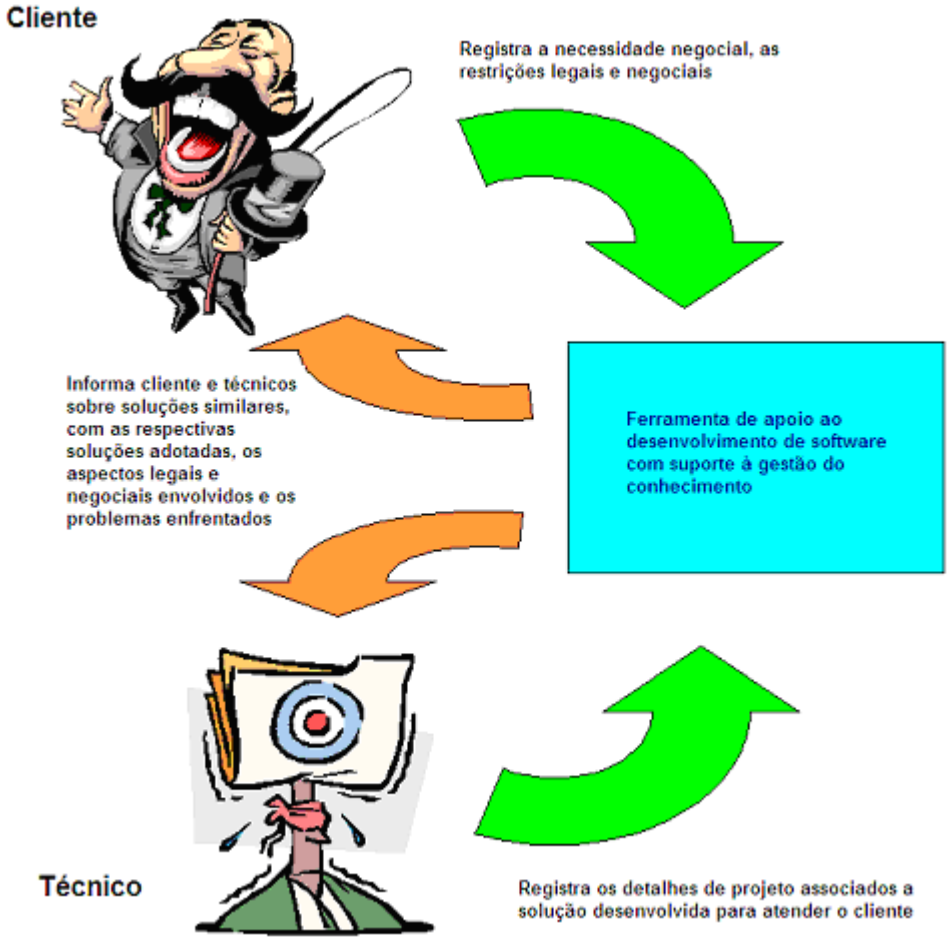

Figura 4.6 – Diagrama básico da ferramenta proposta

A figura 4.6 apresenta uma visão macro de como a ferramenta interage com cliente (demandante da solução) e com profissionais da área técnica envolvida, prestando-lhes apoio nas fases de elaboração da solicitação e de construção da solução de TI.

5 Implementação

O modelo, descrito em detalhes no capítulo anterior, estabelece um conjunto de procedimentos e métodos que visam apurar a similaridade entre duas solicitações de serviço. Neste capítulo se faz a descrição da ferramenta que foi construída para seu suporte e avaliação e que recebeu a denominação de FSSAIA (Ferramenta para a Solicitação de Serviços Assistida por Inteligência Artificial). No desenvolvimento dessa ferramenta, foram priorizadas a utilização de soluções de *software* livre e o direcionamento para ambiente Web, como forma de torná-la facilmente integrável a outras ferramentas de controle de solicitações de serviços. São descritos a seguir os requisitos, a arquitetura geral da aplicação e os diversos módulos que a compõem. A construção da ferramenta considera aspectos do domínio da aplicação, sendo, portanto, customizada para essa finalidade.

5.1 Requisitos da ferramenta de suporte ao modelo

Como todo produto de *software*, também a construção da ferramenta proposta deve seguir uma metodologia adequada, com o objetivo de se obter uma solução que seja capaz de atender as necessidades apresentadas. Dessa forma, após a abordagem inicial, que cumpre a etapa de iniciação e de planejamento, cabe detalhar os requisitos mínimos que devem ser atendidos. Tais requisitos são divididos em dois grupos, como usualmente se faz na engenharia de *software*, respondendo o primeiro pelas necessidades funcionais da ferramenta - aquilo que a ferramenta deve fazer - e o segundo pelas necessidades não funcionais.

No curso da pesquisa identificaram-se os seguintes requisitos, que foram detalhados e complementados durante a construção do estudo de caso.

a) Funcionais → Correspondem às funcionalidades que o aplicativo deve apresentar para atender aos requisitos pretendidos com seu uso:

- Permitir o registro, pelo cliente, da necessidade de desenvolvimento de *software* para o suporte a uma determinada atividade;
- Permitir o registro de um conjunto de funcionalidades que devem ser atendidas para suprir tal necessidade;
- Registrar em uma base de dados todas as requisições impostadas pelo cliente (demandante da solução);
- Permitir a inclusão, alteração, consulta e exclusão de necessidades e de funcionalidades registradas;

- Para cada nova solicitação impostada, fazer a pesquisa de solicitações similares já registradas na base, devolvendo ao usuário uma lista de casos para que ele possa verificá-los e vincular à solicitação aqueles que possam representar contribuições importantes ao processo de desenvolvimento de *software*;
- Permitir o registro, pelos desenvolvedores de aplicações, das soluções idealizadas e construídas para o atendimento das funcionalidades solicitadas;
- Permitir consultas a partir de parâmetros negociais ou técnicos;
- Permitir que sejam utilizados anexos, em qualquer formato, para complementar as informações; deve alertar, no entanto, aos usuários que os campos de descrição das funcionalidades não deve ser preenchido com termos do tipo "conforme anexo" para evitar a impossibilidade de recuperação das informações.

b) Não funcionais → Correspondem aos requisitos de segurança, performance, e outros que, embora não estejam ligados diretamente às funcionalidades básicas do aplicativo devem ser considerados em seu desenvolvimento.

- Interface gráfica, disponível em ambiente de intranet, utilizando formulário padronizado, com caixas de seleção para os elementos cuja seleção possa ser prevista. Ainda nesse caso, deve estar disponível a opção de seleção "outros" que, quando selecionada habilita o preenchimento de um campo adicional para informação pelos usuários do valor a ser atribuído. Os campos selecionados com essa opção (outros) são desconsiderados para efeito da apuração do índice de similaridade;
- Acessível para execução em ambiente web durante, pelo menos, todo o horário comercial (8:00 às 18:00);
- Apresentar tempo de resposta inferior a 5 segundos para paginação entre as telas de entrada de dados e para a pesquisa de similaridades na base;
- Pré-processar os anexos em formato conhecido, registrando os termos em campo separado, com possibilidade de atribuição de peso diferenciado em relação aos estabelecidos para os campos informados diretamente na interface da aplicação (considerou-se que essa implementação não era imprescindível para avaliação do modelo e foi ser deixada como trabalho futuro)

5.2 O Ambiente de desenvolvimento

Como parte da premissa para o desenvolvimento da ferramenta que suportará o modelo e permitirá sua validação, foram selecionados os aplicativos sobre os quais sua construção se fundará. Nessa etapa foram avaliadas as alternativas disponíveis para o estabelecimento de uma infra-estrutura de *software* estável e testada que permitisse que o desenvolvimento fosse feito sem preocupações quanto a influências desses componentes no resultado final. Partindo dessa premissa, foram selecionados os seguintes componentes.

- Sistema Operacional: Utilizamos no desenvolvimento e testes a versão 7.0 do Mandriva Linux e também a versão XP do Microsoft Windows, obtendo resultados idênticos quanto aos índices de similaridade em ambos os sistemas;
- Servidor HTTP: Optou-se por utilizar a versão 2.2.4 do Servidor Apache, considerando suas boas qualidades quanto aos aspectos de segurança e estabilidade, e também o fato de ser ele a referência principal na Web em aplicativos da espécie;
- Banco de dados: A escolha recaiu sobre a plataforma MySQL, que também apresenta boas características quanto a estabilidade, desempenho, recuperação segura das informações e também é referência em *software* livre para SGBDs;
- Linguagem de programação: Fizemos a opção pelo uso da linguagem PHP, considerando principalmente as características de facilidade de uso, disponibilidade de funções apropriadas para comparação de dados e manipulação de *strings*. Sabemos que essa escolha poderá prejudicar a performance final da aplicação pelo fato de ser uma linguagem interpretada mas, como o objetivo da ferramenta é o de provar a correção do modelo, entendemos que a solução poderá ser aplicada, ressalvando-se essa questão para trabalhos futuros.

5.3 A arquitetura da aplicação FSSAIA

A figura 5.1 apresenta a arquitetura utilizada pela ferramenta. Nela pode ser observadas a presença de uma interface com o demandante e de uma aplicação de controle das solicitações, que não integram, diretamente, o escopo desta pesquisa, mas que, para permitir a avaliação do modelo proposto foram também implementadas. Além da interface com o demandante, existe também a interação da ferramenta com o executante do serviço solicitado, seja para a verificação de solicitações vinculadas seja para pesquisas que por sua conta entenda necessárias e com o especialista para a calibragem de pesos e atualizações sobre a base de termos mantidas pela aplicação.

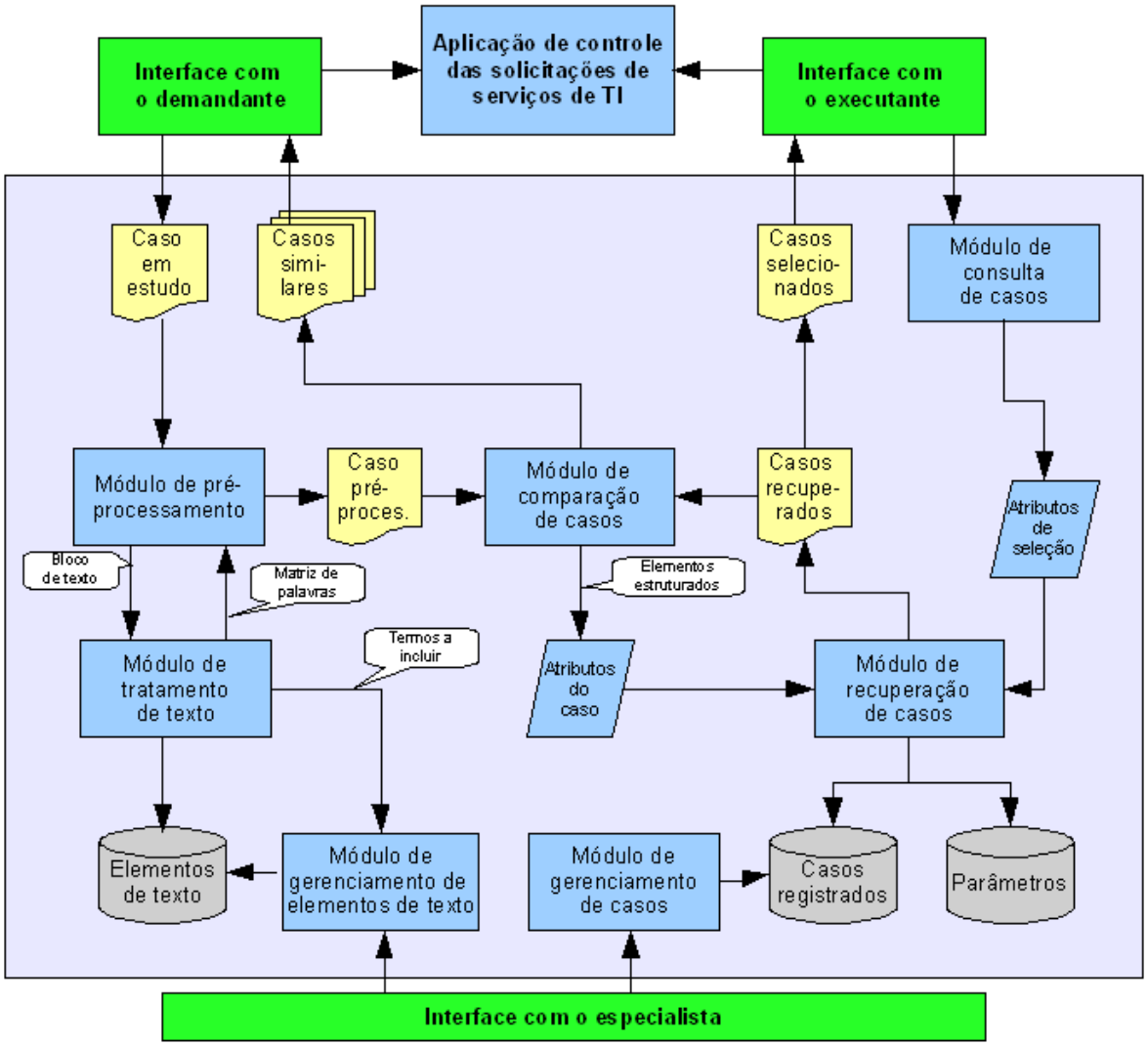

Figura 5.1 – Arquitetura da aplicação de suporte ao modelo proposto

5.4 A interface com o usuário

A interface é o primeiro elemento de interação do usuário com o modelo e o ponto de partida para todo o processo. Ela é responsável pelo registro inicial da solicitação que será processada e por interagir com os mecanismos de suporte para a obtenção de casos similares. Entre suas atribuições está a de implementar um formulário básico para o registro da solicitação, com os campos estruturados e não estruturados identificados na fase de modelagem e as funções necessárias ao registro dessa nova solicitação na base de dados do sistema.

Por uma questão de simplificação, optou-se pela construção no formato de uma página Web, na qual estão dispostos em um menu localizado na parte esquerda o acesso tanto à interface do usuário, quanto a interface técnicas e de manutenção.

A interface foi implementada com o seguinte aspecto visual:

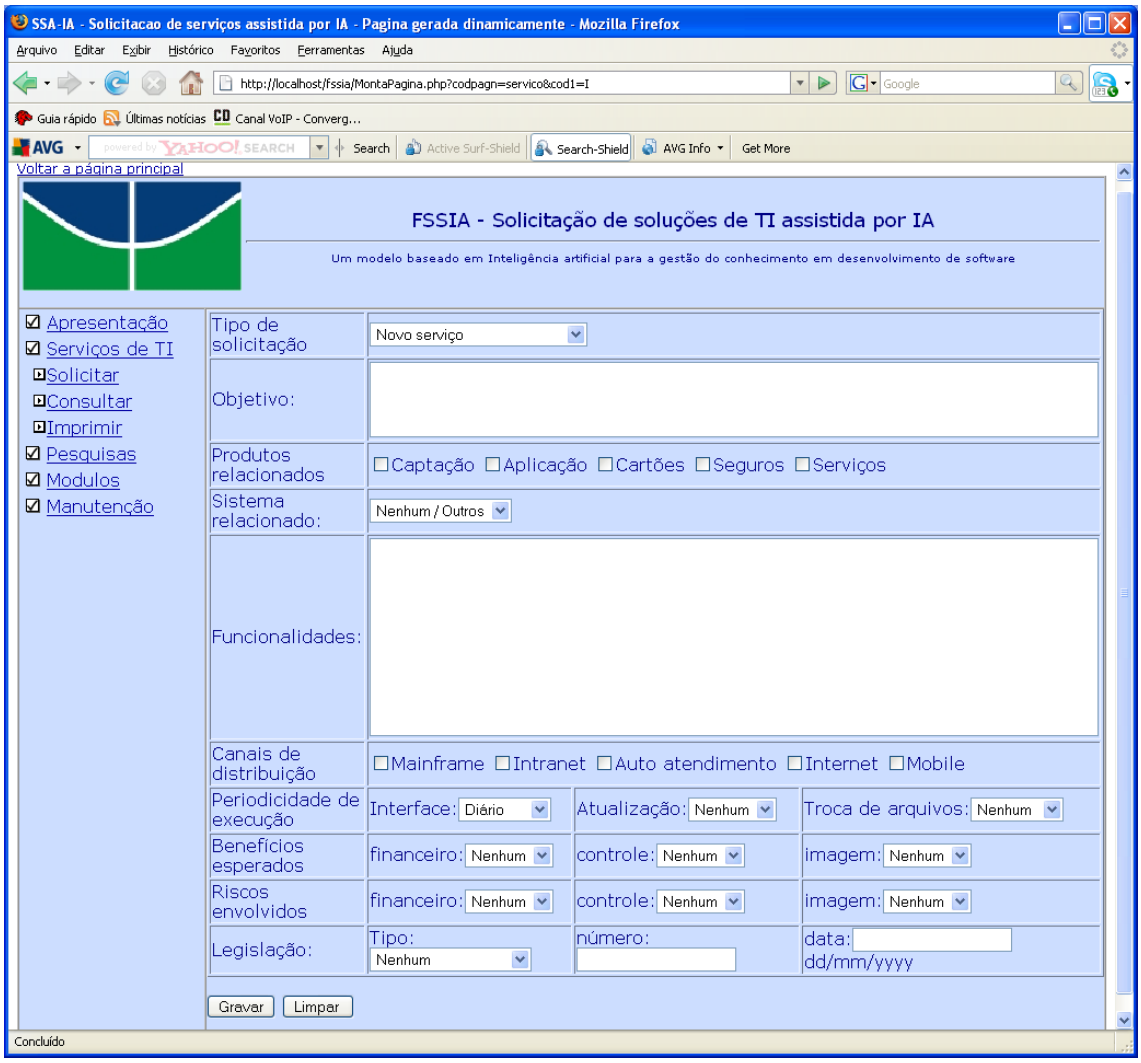

Figura 5.2 – A interface com o demandante da solicitação

Optou-se pela construção de uma interface simples, na qual todos os elementos estruturados estão disponíveis através de caixas de seleção ou botões do tipo “ratio box”, evitando, dessa forma, a impostação de valores desconhecidos pelo sistema; os campos objetivo e funcionalidades permitem a descrição textual das implementações que devem ser providenciadas pela área de TI.

Nessa primeira versão não foi incluída a funcionalidade de permitir e tratar anexos, o que deverá ser feito nos trabalhos futuros. Os anexos são compostos, em geral, por documentos que complementam e detalham as solicitações, tais como cópias de legislações aplicáveis, planilhas com exemplos de cálculos, imagens e documentos com formatações diversas.

5.5 Interface com o executante do serviço

A interface com o executante oferece a possibilidade de consultar a solicitação de serviços, com todos os campos informados pelo demandante e os vínculos que foram informados. Ao verificar os vínculos, o executante deve tomar a decisão por reusar, ou não, alguma solução previamente elaborada (ou em fase de elaboração), bem como tomar conhecimento de fatos relevantes que foram registrados quando de seu desenvolvimento.

Está disponível, também, nessa interface do executante, o acesso para avaliação quanto a utilidade das soluções apontadas pelo sistema e confirmadas pelo demandante como sendo similares ao caso atual.

A figura 5.3 ilustra a interface do executante.

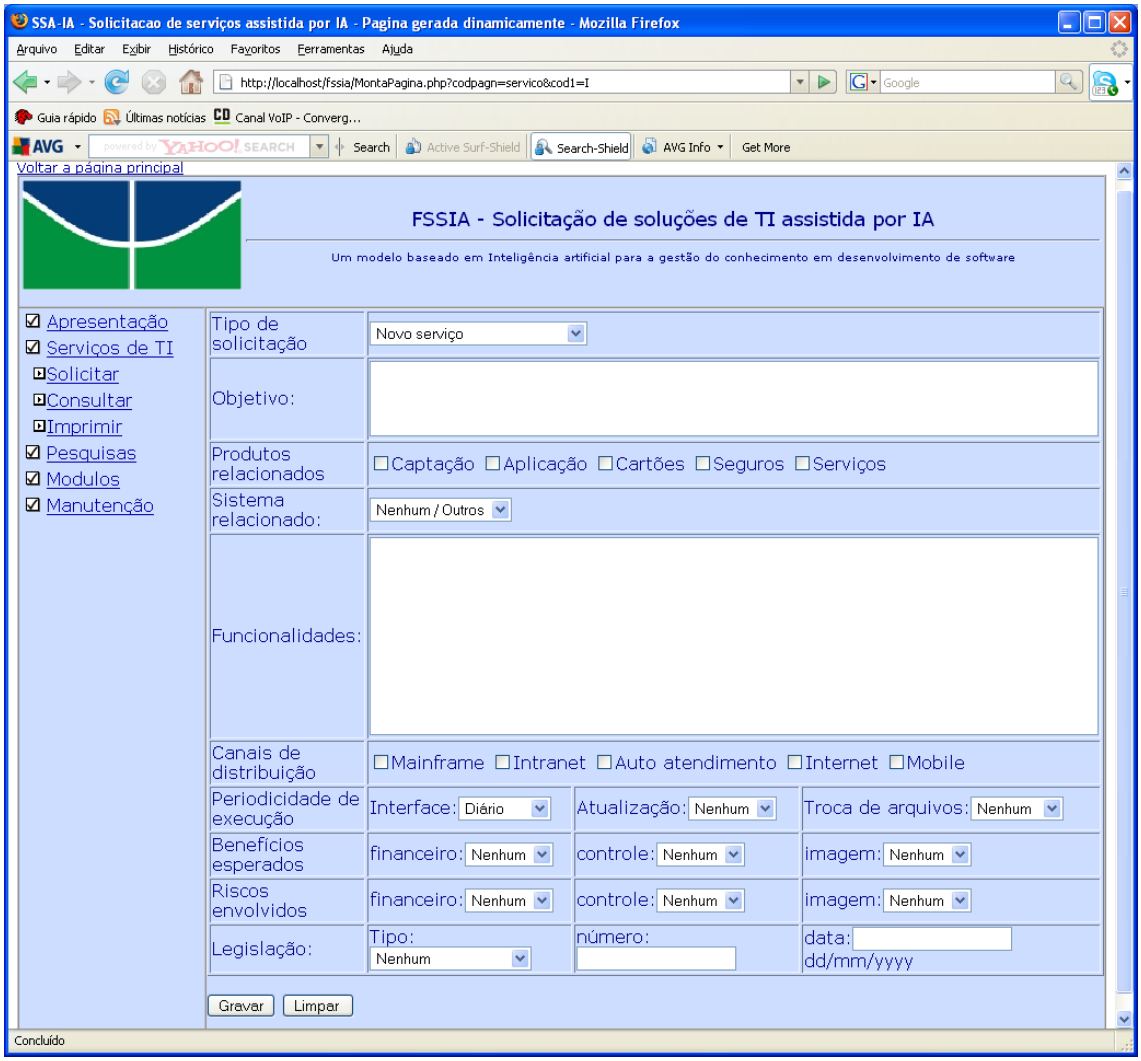

Figura 5.3 – A interface com o executante (técnico responsável pelo atendimento da solicitação)

5.6 Especificação dos módulos de determinação da similaridade

Descreve-se, nos itens seguintes cada um dos módulos que foi implementado para desempenhar a função de reconhecer casos similares através do modelo de processamento explanado na capítulo 4. Como aquele capítulo descreve detalhadamente as funcionalidades e metodologia a serem empregadas, a abordagem desses itens será feita a nível de detalhamento dos objetivos e das entradas e saídas de cada módulo, permitindo assim a compreensão do seu funcionamento.

5.6.1 Módulo de pré-processamento

Objetivos: Converter um caso para estudo, contendo elementos estruturados, provenientes de itens de seleção no módulo de interface, e elementos não estruturados, correspondentes aos itens de texto em linguagem natural, digitados pelo usuário em um caso em formato padrão, para avaliação pelo módulo de comparação.

Entradas: Caso para estudo, contendo os seguintes atributos e formatos especificados na tabela 5.1.

Saídas: Caso pré-processado, contendo a estrutura de dados indicada na tabela 5.2:

Nome do elemento	Tipo / Formato	Observações
Tipo de solicitação	Estruturado / Numérico	Contém a informação sobre o tipo de serviço solicitado.
Principais produtos relacionados	Estruturado / Numérico	Contém a informação sobre as grandes áreas associadas com a solicitação do serviço (valores tabelados a serem definidos).
Objetivo	Não estruturado / Texto	Contem uma descrição resumida dos objetivos da solicitação.
Funcionalidades	Não estruturado / Texto	Contém uma descrição detalhada sobre as funcionalidades que devem ser atendidas pelo produto resultante da solicitação.
Legislação_tipo	Estruturado / Numérico	Contém informação associada ao tipo de normativo que regula a solicitação (valores tabelados a serem definidos).
Legislação_número	Não estruturado / texto	Contém uma complementação sobre a legislação relacionada.
Legislação_data	Não estruturado / texto	Contém a data de referência para a legislação indicada.
Canais de distribuição	Estruturado / numérico [até 3 informações]	Valores previamente cadastrados para representar os possíveis canais de utilização do serviço.
Sistema relacionado	Estruturado / numérico	Informação do principal sistema relacionado com a solicitação.
Periodicidade	Estruturado / numérico	Contém uma combinação de valores relativos à periodicidade de execução, atualização e interface.
Benefícios esperados	Estruturado / numérico	Contém uma combinação de valores relativos aos benefícios esperados com a implementação da solução.
Riscos envolvidos	Estruturado / numérico	Contém uma combinação de valores relativos aos riscos envolvidos com a não implementação da solução.

Tabela 5.1 – Especificação dos atributos de entrada do módulo de pré-processamento

Nome do elemento	Tipo / Formato	Observações
Todos os campos da estrutura de entrada	Mesmo formato da estrutura de entrada	Os campos devem ser mantidos para que no momento em que se desejar inserir um novo caso na base, todas as informações já estejam disponíveis.
Objetivo processado	Matriz, de dimensão n x 2, (adotar n = 10 para a primeira versão)	Contém uma lista de palavras, convertidas pelo módulo de processamento de textos, associadas a um valor numérico que represente sua relevância em relação ao conjunto de texto.
Funcionalidades	Matriz, de dimensão n x 2, (adotar n = 50 para a primeira versão)	Contém uma lista de palavras, convertidas pelo módulo de processamento de textos, associadas a um valor numérico que represente sua relevância em relação ao conjunto de texto.

Tabela 5.2 – Especificação dos atributos de saída do módulo de pré-processamento

5.6.2 Módulo de tratamento de textos

Objetivos: Criar uma matriz de "n" registros formados por uma palavra e um valor numérico associado, representando a relevância da palavra no texto. Utilizar a metodologia TFIDF para obtenção desse valor.

Entradas: Bloco de texto, retirado da solicitação de serviços de TI, e quantidade de palavras que deve constar na lista de saída, conforme tabela 5.3.
Saídas: Matriz [1 ..100] elementos, contendo a estruturação mostrada na tabela 5.4.

Nome do elemento	Tipo / Formato	Observações
Texto em linguagem natural	Não estruturado / Texto	Contém toda a cadeia de caracteres que forma o bloco de texto. Havendo necessidade, pode-se utilizar esse campo, em combinação com o campo tipo de informação para passar o nome do arquivo contendo o texto a processar.
Tipo de informação	Estruturado / Numérico	Contém a descrição do conteúdo do elemento 'texto em linguagem natural' . Utilizar os seguintes códigos : 1-campo contém o texto a processar ; 2-campo contém o nome do arquivo que contém o texto a processar
Tamanho da lista de palavras	Não estruturado / Numérico	Contém um valor discreto entre 1 e 100, que representa o tamanho máximo de elementos que estarão contidos na matriz de saída.

Tabela 5.3 – Especificação da estrutura de entrada do módulo de processamento de textos

Nome do elemento	Tipo / Formato	Observações
Palavra	Estruturado / Texto	Contém a palavra resultante do processamento do bloco de texto, substituída ou ajustada para seu 'sinônimo' preferencial.
Relevância	Não estruturado / Numérico	Contém um valor numérico entre 0 e 1 (mínima e máxima relevância, respectivamente), que represente a relevância da palavra em relação ao conjunto de texto que está sendo processado e em relação à freqüência invertida com que essa palavra aparece nos demais documentos registrados na base de casos.

Tabela 5.4 – Especificação da estrutura de saída do módulo de processamento de textos

5.6.3 Módulo de comparação de casos

Objetivos: Comparar o caso em estudo, após o pré-processamento, com os casos recuperados da base de casos, atribuindo um índice de similaridade entre os casos comparados e produzindo uma lista ordenada de casos selecionados, cujo índice de similaridade esteja acima de um dado patamar (parâmetro do sistema) ajustado durante a fase de calibração do sistema.

Entradas:

1) Caso em estudo, após o pré-processamento, no formato descrito para a saída do módulo de pré-processamento;

2) Lista de casos recuperados pelo módulo de recuperação de casos, a partir de um conjunto mínimo de similaridades entre os elementos estruturados.

Saídas: Lista de casos selecionados, ordenados pelo índice de similaridade entre os casos recuperados para comparação e o caso em estudo.

5.6.4 Módulo de recuperação de casos

Objetivos: Recuperar, na base de casos, os casos que atinjam um índice mínimo de

similaridade, quando comparados somente os elementos estruturados com os do caso em estudo.

Entradas:

1) Elementos estruturados do caso em estudo;

2) quantidade máxima de casos a recuperar.

Saídas: Lista de casos selecionados, ordenados pelo índice de similaridade (prévio) entre os casos recuperados para comparação e o caso em estudo.

Observação: Na comparação dos casos da base com o caso em estudo, deve-se comparar primeiramente os elementos estruturados; somente os casos que atinjam um valor mínimo nesse critério são levados a comparação completa de elementos estruturados e não estruturados (feita pelo módulo descrito em 5.6.3).

5.6.5 Módulo de manutenção de elementos de texto

Objetivos: Este módulo deve prover os recursos para gerenciar a base de termos, que contém as palavras, as ações e substituições que devem ser feitas durante a fase de pré-processamento do texto e os valores associados a cada uma dessas palavras, considerando sua freqüência nos documentos armazenados na base de casos.

Entradas: Não há.

Saídas: Não há.

Observação:Durante a fase de pré-processamento, poderão ser encontradas novas palavras que não estejam registradas entre os elementos constantes da base de elementos de texto. Essas palavras sãor marcadas (durante o pré-processamento) para posterior classificação por um especialista no domínio do negócio; enquanto não é feita essa classificação, esses termos são tratados da forma como aparecem no texto, ou seja, não é feita nenhuma substituição ou exclusão sobre esses termos.

A estrutura inicialmente proposta para a base de termos está indicada na tabela 5.5.

Campo	Formato	Observações
Termo	String	Contém o termo, grafado sem acentuação e caracteres especiais (p. ex.“ç”, #, &)
Categoria	Numérico	Contém a categoria a qual pertence o termo (artigo, preposição, verbo, substantivo, etc....)
Ação	Numérico	Código que determina o comportamento que deve ser adotado ao processar um texto contendo o termo. Usar a seguinte convenção: 1-Manter; 2-Excluir; 3-Substituir por; 4-Classificar
Complemento da ação	String	Contém o termo a ser usado quando a ação descrita for a de código 3 (substituir por), grafado na mesma forma que o campo termo
Freqüência	Numérico	Contém um número inteiro indicando a freqüência relativa ao termo, na base de casos mantida pelo sistema
Relevância	Numérico	Contém um número real que indica a relevância relativa ao termo, obtida pelo processamento dos documentos em relação à sua freqüência nos casos registrados na base

Tabela 5.5-Estrutura da tabela de termos

5.6.6 Módulo de gerenciamento de casos

Objetivos: Este módulo deve prover os recursos para gerenciar a base de casos mantida pelo sistema, permitindo consultar, alterar e excluir as informações ali registradas. Esse módulo deve prover dois tipos de acionamento: um para inclusão de novos casos através de uma chamada a partir do módulo de interface e outro com uma interface própria e opções para acesso direto aos casos registrados na base.

Entradas:

a) Tipo de acionamento : 1-Acionamento para inclusão de novo caso na base de casos, a partir do módulo de interface; 2 – Acionamento para opções de consulta, alteração e exclusão de casos diretamente na base.

b) Caso para registro: Estrutura, no formato de caso pré-processado (saída do módulo de pré-processamento) para registro na base, quando o acionamento se der pelo tipo 1.

Saídas: Não há.

Observação: Este módulo deve interagir com o módulo de recuperação de casos para implementar as opções de consulta, inclusão e exclusão de casos.

5.6.7 Módulo de atualização de casos

Objetivos: Este módulo deve ser executado periodicamente, para atualizar as estatísticas sobre os termos constantes da base, que podem sofrer alterações em função da inclusão de novos casos. Devem ser atualizados os pesos relativos a cada um dos termos considerados na composição da base de casos.

Entradas: Não há.

O acionamento deve ser agendado conforme a freqüência de atualização dos casos, ou a cada período regular de tempo, conforme se verifique a degradação da performance da aplicação.

Saídas: Não há.

Após o processamento os casos deverão estar atualizados na base.

5.6.8 Módulo de gerenciamento de solicitações de serviços

Objetivos: Este módulo deve prover os recursos para gerenciar a base de solicitações de serviços mantida pelo sistema, permitindo consultar, alterar e excluir as informações ali registradas. Ele deve prover dois tipos de acionamento: um para inclusão de novas solicitações de serviços através de uma chamada a partir do módulo de interface e outro com uma interface própria e opções para acesso direto às solicitações registradas na base.

Entradas: Tipo de acionamento : 1-Acionamento para inclusão de novas solicitações na base, a partir do módulo de interface; 2 – Acionamento para opções de consulta, alteração e exclusão de solicitações diretamente na base.

Saídas: Não há.

6 Estudo de caso: Desenvolvimento de software em um banco público (parte I – Preparação do experimento)

Para a avaliação do modelo proposto foi desenvolvido um estudo de caso, aplicado à área de desenvolvimento de *software* de uma grande organização do setor financeiro, em consonância com a proposição do modelo e ferramenta. Os experimentos e os resultados aqui relatados foram produzidos tomando por base a metodologia CRISP-DM (*Cross-Industry Standard Process for Data Mining*), que apresenta uma abordagem adequada aos objetivos desta pesquisa e é uma referência mundial em trabalhos da espécie. São tratadas neste capítulo quatro das seis fases descritas no modelo de referência, cobrindo as etapas de entendimento, coleta e preparação dos dados e modelagem a ser utilizada na condução dos experimentos.

A etapa de avaliação, por sua importância para a compreensão dos resultados, foi disposta no capítulo 7, no qual se faz uma análise aprofundada dos experimentos realizados. A etapa de implementação será abordada apenas sob a ótica de um planejamento, uma vez que, por tratar-se de um trabalho acadêmico, o escopo da pesquisa se encerra com a avaliação dos resultados obtidos. A figura 6.1 ilustra as fases do modelo de referência empregado.

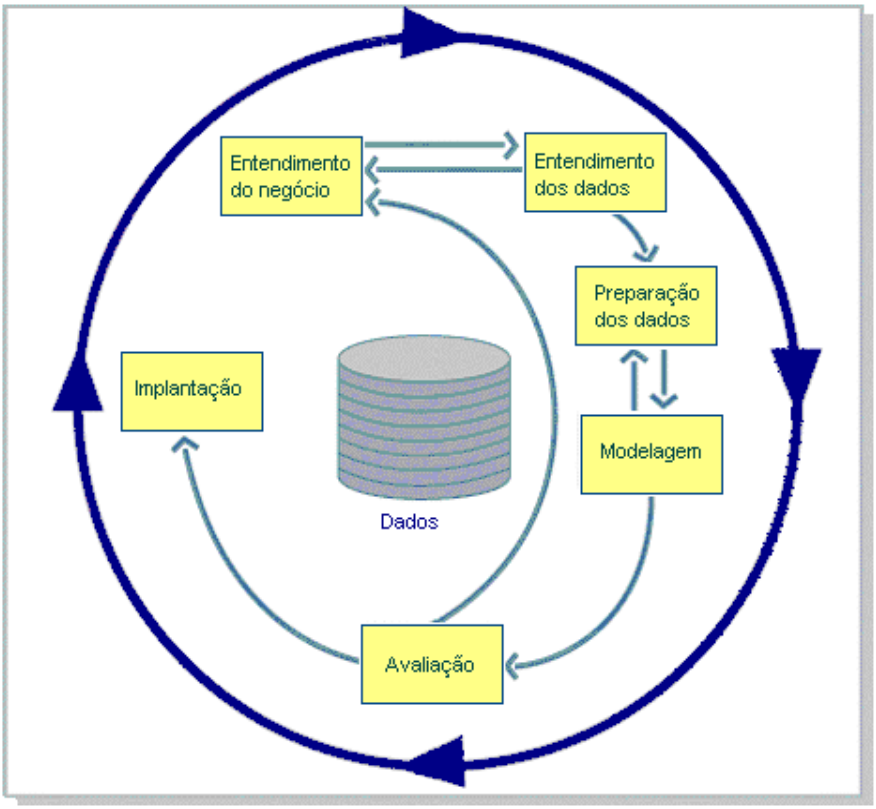

Figura 6.1 – As etapas do modelo de referência CRISP-DM

(Fonte:http://www.crisp-dm.org)

6.1 Entendimento do negócio

Para a compreensão do contexto em que se fez a avaliação do modelo proposto, torna-se necessário explanar, em linhas gerais, alguns detalhes sobre a organização alvo do estudo de caso. Os itens descritos a seguir permitem construir o entendimento do negócio da organização, caracterizam a importância do processo de desenvolvimento de *software* para seus negócios, apresentam o problema a ser tratado nesta pesquisa e estabelecem os objetivo e expectativas para o trabalho a ser realizado.

6.1.1 Caracterização da organização

A empresa objeto do presente estudo de caso é uma sociedade de economia mista (parte das ações em poder de acionistas particulares e parte pertencentes ao governo federal), com atuação no ramo financeiro. A instituição atua fortemente como agente de políticas públicas, notadamente na área de fomento a produção agroindustrial em que detém 83% da carteira de crédito total do país. Além da área agroindustrial, sua atuação abrange, também, a área de crédito comercial em geral, com produtos voltados ao atendimento de empresas de pequeno, médio e grande porte e também ao atendimento de pessoas físicas.

A instituição conta hoje com uma rede de mais de 15 mil pontos de atendimento, dispersos geograficamente por todo o país, sendo o Banco com maior penetração em termos absolutos de municípios atendidos. Conta, ainda, com uma ampla rede de auto-atendimento com aproximadamente cem mil terminais eletrônicos (BB, 2008) posicionados em locais de grande movimento de pessoas, e com uma infra-estrutura para atendimento por *call center* e através de auto-atendimento por celular- o único do país, neste momento.

Para atendimento de toda essa estrutura, a instituição conta com uma diretoria de tecnologia, com aproximadamente três mil pessoas divididas entre as áreas de suporte a produção e de desenvolvimento de aplicativos.

A área de desenvolvimento de aplicativos responde pela produção do *software* necessário a automatização de todos os processos da instituição, sendo demandada diariamente para a construção de novos aplicativos, para a evolução dos aplicativos existentes, agregando-lhes novas funcionalidades ou adequando-os às alterações legais ou de mercado e respondendo, também, pela manutenção dos aplicativos em produção quanto ao atendimento de eventuais falhas ocorridas durante sua execução.

Essa área, que conta com aproximadamente mil e quinhentas pessoas entre funcionários de quadro próprio e empregados terceirizados, transfere mensalmente uma média

de 15 mil módulos de programa para o ambiente de produção, atendendo a demanda por novos aplicativos e por alterações e correções de aplicativos existentes, e é subdividida em gerências de desenvolvimento, com atribuições semelhantes entre si, voltada, cada uma, ao atendimento de um determinado grupo de produtos e de diretorias.

A empresa vem adotando, há aproximadamente sete anos um ciclo de aperfeiçoamento, que vise garantir a aderência de seus processos a normas internacionais, fruto da preocupação com a continuidade dos negócios da organização e também da exigência de convenções internacionais, como o acordo de Basiléia, por exemplo.

Apesar dos esforços que a área vem empreendendo, e dos significativos avanços já conseguidos, a gestão do conhecimento empregado nesse processo ainda se destaca como um dos pontos em que mais se pode avançar.

6.1.2 Situação atual

A área de tecnologia da empresa, explanada em números gerais na introdução deste capítulo, atende, em termos de desenvolvimento de *software* e de infra-estrutura de processamento todo o conglomerado, sendo demandada por diversas diretorias para a construção de diferentes aplicativos.

Durante essa fase de construção existe uma grande interação entre os profissionais da área técnica e da área negocial, na qual são abordados os mais diversos aspectos do negócio que a solução pretende suportar. No entanto, face à grande quantidade de pessoas trabalhando nas áreas envolvidas, a intensa rotatividade de pessoas e a ausência de uma ferramenta de apoio, todo esse conhecimento acaba não sendo registrado, ou registrado de forma tal que seu reuso fica absolutamente prejudicado. Observa-se, também, a freqüente ocorrência de intervenções corretivas ou evolutivas que são efetuadas por equipe distinta daquela que participou da construção do aplicativo, sendo tal atividade dificultada pela ausência desse conhecimento.

A figura 6.2 ilustra a situação descrita: diretorias diferentes apresentam solicitações cujos componentes apresentam similaridade entre si, porém, como o desenvolvimento será feito na área de tecnologia por equipes de desenvolvimento diferentes, muitas soluções que poderiam ser compartilhadas acabam sendo desenvolvidas em duplicidade.

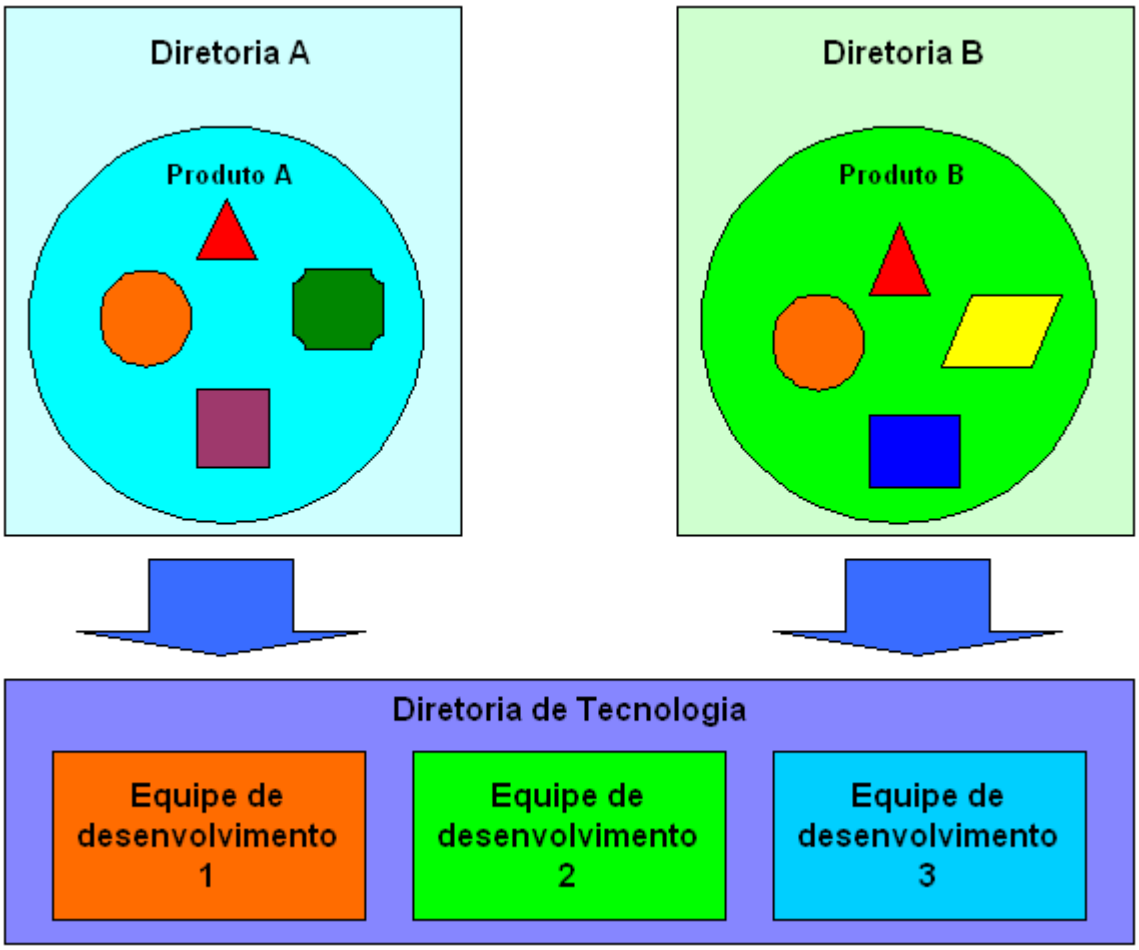

Figura 6.2 – Demandas de diversas diretorias por soluções de tecnologia

Essa duplicidade no desenvolvimento de componentes similares aumenta o esforço consumido no atendimento, provoca dubiedade entre as soluções, que, por vezes, fazem a mesma coisa de forma completamente diferente, e aumenta o consumo de recursos nas atividades de manutenção que poderão vir a ser demandadas posteriormente.

6.1.3 Objetivos do modelo proposto

- Identificar, durante a fase de preparação pelo demandante, as similaridades existentes entre a solicitação que está sendo apresentada e as demais solicitações já atendidas pela diretoria de tecnologia e permitir a revisão e a vinculação de elementos comuns;
- Permitir ao executante do serviço que tome conhecimento e reuse soluções já aplicadas à problemas semelhantes, diminuindo a quantidade de esforço no desenvolvimento e nas futuras manutenções da solução e evitando a duplicidade de soluções para um mesmo problema.

6.1.4 Determinação das metas de mineração de dados

Para o escopo deste trabalho, pretende-se avaliar a capacidade do modelo proposto em obter um resultado, na forma de um índice que represente a similaridade entre dois documentos representados por solicitações de serviços em TI. Para o alcance desse objetivo, estabeleceram-se os seguintes critérios:

a) Um documento é considerado similar a um outro já registrado na base da dados, quando índice final de similaridade (IS) entre eles é superior a 0,5 (cinco décimos), em uma escala que varia de 0 (zero) a 1(um), na qual o valor zero representa que não existe qualquer semelhança entre as solicitações e o valor 1 representa que as solicitações são idênticas, para efeito dos atributos considerados.
b) Somente são comparadas completamente (atributos estruturados e não estruturados) as solicitações que apresentem um índice de similaridade entre os elementos estruturados (ISee) superior à um limiar, fixado inicialmente em 0,5 (cinco décimos) em uma escala igualmente construída com valores na faixa de 0 (zero) a 1 (um) e ajustado, durante a etapa de testes, para 0,55.

Os resultados finais obtidos pela composição dos índices de similaridade entre elementos estruturados (ISee) e não estruturados (ISne) são computados com a atribuição de pesos específicos para cada um desses índices e apresentados ao usuário sob a forma de uma lista ordenada (com valores decrescentes) das solicitações que representam casos similares ao caso em estudo.

6.2 Entendimento dos dados

6.2.1 Coleta inicial de dados

Os dados utilizados para a realização dos experimentos foram extraídos de um sistema utilizado pela organização alvo para o registro e controle de solicitações de serviços em TI. Esse sistema é utilizado pela organização há aproximadamente quatro anos e conta com mais de 30.000 solicitações registradas em sua base. Apesar de conter uma quantidade enorme de informações, sua estruturação foi concebida apenas para o controle dos pedidos e entregas e da quantidade de esforço envolvido. Não existe ali a preocupação com a possibilidade de recuperar e reusar as informações como forma de diminuir a quantidade de esforços gasta com o atendimento das solicitações.

6.2.2 Descrição dos dados

O sistema empregado para o controle das solicitações de serviços de TI mantém o registro de diversas informações, desde a fase em que a solicitação é impostada, até o momento em que o solicitante atesta que a solução desenvolvida atende aos requisitos pretendidos. Durante esse ciclo varias informações são agregadas à solicitação inicial, porém, na maioria das vezes, com o viés de controle de recursos empregados no atendimento.

Para os objetivos do presente trabalho, foram extraídas apenas as informações que pudessem agregar valor ao processo de identificação de similaridades entre solicitações já atendidas e novas solicitações, durante a fase de construção da solicitação ou do início do seu atendimento. Dentre essas informações, destacam-se os campos objetivo da solicitação e funcionalidades a serem atendidas, além de atributos que identificam os produtos e sistemas envolvidos e canais de disponibilização.

6.2.3 Exploração/verificação da qualidade dos dados

Durante a verificação inicial dos dados que estavam disponíveis para a realização do estudo de caso, percebeu-se que havia grande disparidade entre a qualidade e a quantidade de informações contidas em cada solicitação de serviço; enquanto algumas solicitações traziam descrições detalhadas e precisas sobre o objetivo pretendido e sobre as funcionalidades que se pretendiam implementar, outras traziam descrições imprecisas e superficiais acerca dos mesmos campos. Outra situação comumente encontrada nas solicitações existentes nessa base

é a utilização de anexos, em formatos diversos, o que dificulta bastante qualquer processo de recuperação de dados, principalmente quando tais anexos abordam assuntos e aspectos que não fazem parte dos objetivos da solicitação.

A utilização de descrições imprecisas é um dos pontos que se identificou, durante essa avaliação, como uma provável causa para a insatisfação dos demandantes dos serviços com a qualidade de alguns produtos desenvolvidos.

6.3 Preparação dos dados

6.3.1 Seleção dos dados

A organização alvo estrutura sua área de tecnologia em gerências de atendimento, que atualmente são em número de cinco. Cada uma dessas gerências atende a solicitações destinadas a um grupo de produtos ou serviços, guardando estreita relação das atividades entre si; seus processos e métodos de trabalhos são semelhantes, assim como a organização da fila de entrada de serviços e a forma de entrega dos produtos desenvolvidos. Essa semelhança permite que seja escolhida uma entre essas gerências para a aplicação do modelo e avaliação de resultados com a suposição de que a validação do modelo para essa gerência o permitiria estender às demais.

Das solicitações apresentadas a essa diretoria, foi separado, de forma aleatória, um grupo de cinqüenta solicitações entre as registradas na base cujo atendimento já tenha sido providenciado. Os dados referentes a esse grupo de solicitações foram transcritos para a aplicação FSSIA como forma de validar a metodologia proposta.

Durante essa coleta, optou-se por trazer para a aplicação soluções com diferentes níveis de detalhamento e quantidade de informações, até porque era preciso avaliar o comportamento do modelo frente a essas situações, e por não coletar dados de solicitações cujo detalhamento estivesse contido em anexos (exceto quando os anexos apenas complementavam o detalhamento existente). Essa decisão implicou na necessidade de estabelecer uma primeira categorização das solicitações quanto ao nível de detalhamento presente na solicitação, e que foi especialmente útil durante a análise dos resultados.

6.3.2 Tratamento e formatação dos dados

A aplicação construída teve por premissas a utilização de soluções de *software* livre em todas as suas etapas. Dessa forma, as informações que estavam nas solicitações de serviços, registradas em banco de dados IBM/DB2, foram migradas para a aplicação,

alimentando uma base de dados suportada por um SGBD de código aberto (MySQL). Durante essa transcrição, que foi feita a partir da interface da aplicação FSSIA, foram complementados os dados que não estavam explicitamente declarados. Após esse tratamento, cada registro de solicitação passou a conter as mesmas informações, formatadas e organizadas como se descreve adiante.

Para efeito de tratamento, os dados foram agrupados em dois tipos:

a) Elementos estruturados → Quando o valor que descreve o elemento podia ser expresso em termos numéricos ou mnemônicos e sua representação em uma base de dados pudesse ser diretamente comparados, e
b) Elementos não estruturados → Quando sua descrição é feita com o uso de linguagem natural. Esses elementos requerem que seja feito um tratamento adequado para que se possa compará-los em busca de similaridades.

As informações utilizadas, suas respectivas descrições, tipos e valores utilizados no modelo estão descritas na tabela 6.1.

Seq.	Nome do campo	Descrição	Tipo	Valores
1	Tipo de solicitação	Descreve o tipo de solicitação de serviço de TI	Estrut.	1-Novo serviço 2-Alteração de funcionalidade já existente
2	Produtos relacionados	Descreve com que produtos bancários se relaciona a solicitação de serviço de TI	Estrut.	1-Captação 2-Aplicação 4-Cartões 8-Seguros 16-Serviços
3	Canal de distribuição	Relaciona todos os possíveis canais em que a solução deverá ser disponibilizada	Estrut.	1- Mainframe 2- Intranet 4-Auto-atendimento 8-Internet 16-Mobile
4	Periodicidade de execução	Descreve com que periodicidade deverá ser executada, atualizada ou estabelecida a troca de arquivos com outros sistemas	Estrut.	**gr1:Execução** 1-Diário 2-Semanal 4-Mensal 8-Anual/eventual **gr2-Atualização** 16-Diário 32-Mensal 64-Anual/eventual 128-Nenhum **gr3:Troca de arquivos** 256-Diário 512-Semanal 1024-Mensal 2048-Anual/eventual 4096-Nenhum
5	Benefícios esperados	Relaciona os possíveis benefícios para a empresa que serão obtidos com a implementação	Estrut.	**gr1:Financeiro** 1-Nenhum 2-Baixo 4-Médio 8-Alto **gr2:Controle** 16-Nenhum 32-Baixo 64-Médio 128-Alto **gr3:Imagem** 256-Nenhum 512-Baixo 1024-Médio 2048-Alto

Tabela 6.1 – Descrição dos elementos de dados selecionados para o estudo de caso
(continua na página seguinte)

Seq.	Nome do campo	Descrição	Tipo	Valores
6	Riscos envolvidos	Relaciona os riscos a que serão expostos a empresa em caso de não implementação da solução	Estrut.	**gr1:Financeiro** 1-Nenhum 2-Baixo 4-Médio 8-Alto **gr2:Controle** 16-Nenhum 32-Baixo 64-Médio 128-Alto **gr3:Imagem** 256-Nenhum 512-Baixo 1024-Médio 2048-Alto
7	Legislação associada	Descreve qual o tipo de legislação a que está associada a solicitação	Estrut.	**gr1:Tipo** 1-Nenhum 2-Lei federal 4-Lei estadual 8-Lei municipal 16-Portaria/resolução 32-Normativo interno **gr2-Numero da lei** **gr3-data**
8	Sigla do sistema	Informa, se houver, qual a sigla do principal sistema envolvido com a solicitação	Estrut.	1-Nenhum 2-COP 4-IDA 8-GST 16-SEC 32-VPC 64-XER
9	Objetivo da solicitação	Descreve qual o objetivo pretendido com a implementação da solicitação	Texto	Descrição em linguagem natural
10	Funcionalidades	Descreve quais as funcionalidades que deverão ser implementadas pela área de Tecnologia para o atendimento da solicitação	Texto	Descrição em linguagem natural

Tabela 6.1 – Descrição dos elementos de dados selecionados para o estudo de caso

(continuação da página anterior)

6.4 Modelagem dos testes

6.4.1 Técnica empregada

Embora o objetivo do modelo proposto seja avaliar novas solicitações no momento em que estão sendo impostadas no sistema de acompanhamento e controle, era necessário realizar um número significativo de comparações e confrontá-las com o resultado esperado para avaliar sua capacidade de identificar corretamente a similaridade entre as diversas solicitações. A alternativa escolhida para suprir essa necessidade foi fazer a comparação entre cada uma das solicitações registradas na base com todas as demais e confrontar os resultados obtidos automaticamente pela aplicação (FSSIA) com a indicação de especialistas conhecedores do domínio tratado. Essa avaliação foi feita por um grupo de três especialistas (pessoas com conhecimento e experiência no trato de solicitações de serviços de TI) e os resultados utilizados para aferição do modelo foram construídos através do consenso entre essas indicações.

A avaliação focou suas atenções sobre dois aspectos:

a) qualidade da especificação e do detalhamento da solicitação, resultando em dois grupos: grupo 1, composto por solicitações com boa qualidade e detalhamento, e grupo 2, no qual a qualidade ou a quantidade de informações foi considerada pobre. Essa classificação será especialmente útil durante a fase de análise dos resultados;
b) existência, ou não, de similaridade com as demais solicitações. Nesse quesito, foram identificadas três situações possíveis: Inexistência ou insignificância da similaridade (baixa ou nenhuma similaridade), existência de alguma similaridade significativa (média similaridade) e existência de grande similaridade entre as solicitações (alta similaridade);
c) as solicitações identificadas como possuindo média similaridade ou alta foram então ordenadas pelos especialistas conforme suas avaliações, em ordem decrescente da similaridade existente entre elas.

Com base nas avaliações efetuadas, as solicitações foram manualmente tabuladas, utilizando as convenções mostradas nas tabelas 6.2 e 6.3

Indicação do especialista	Valores
Informações suficientes e adequadas	1
Informações insuficientes ou inadequadas	0

Tabela 6.2 Indicação quanto à suficiência e adequação das informações contidas na solicitação de serviços.

Indicação do especialista	Faixa de valores
Baixa ou nenhuma similaridade	0,00 até 0,49
Média similaridade	0,50 até 0,69
Alta similaridade	0,70 até 1,00

Tabela 6.3 – Faixa de valores para comparação dos resultados do sistema

Essa técnica permitiu a construção de uma matriz de similaridade esperada entre as solicitações registradas na base com os valores obtidos para a comparação de cada par, permitindo assim fazer uma primeira avaliação dos resultados obtidos pela aplicação do modelo, em relação a esse domínio específico, quanto a correta classificação das solicitações.

No entanto, os objetivos do modelo pressupõem não apenas a correta classificação quanto à existência de similaridades, mas também a correta ordenação dos índices que representam. Essa ordenação foi objeto da segunda avaliação a que se submeteram os resultados obtidos pelo sistema.

As faixas de valores da tabela 6.3 foram inicialmente estabelecidos como uma estimativa, e se mostraram adequadas durante a avaliação; esses valores foram usados como base para a análise estatística dos resultados que visava para determinar o grau de acerto que, para os objetivos desse trabalho, serão apontados em termos de acurácia, sensibilidade, especificidade e da precisão quanto à correta ordenação do índice final de similaridade apontado. Os resultados da avaliação dos especialistas estão sinteticamente descritos na tabela 6.4:

Id_solicitação	Qualidade da especificação	Solicitações com alta similaridade	Solicitações com média similaridade	Ordem de similaridade
1	0	-	26, 29	
2	1	-	-	-
3	1	-	-	-
4	1	-	8, 9	-
5	0	-	28	28
6	1	-	-	-
7	1	-	31	-
8	1	9	4	9, 4
9	1	8	4, 12, 27	8
10	1	-	11, 21, 22, 23, 24	-
11	1	-	10, 21,	-
12	0	26	13	26, 13
13	1	-	13, 26	-
14	1	-	36	36
15	0	-	-	-
16	0	-	36	36
17	0	-	-	-
18	1	-	45, 46	
19	1	31	17	31, 17
20	0	-	45	45
21	1	-	10, 21, 22, 23, 24	-
22	1	21, 24	10, 23	21, 24
23	1	-	10, 21, 22, 25	-
24	1	24	10, 21, 25	24
25	1	-	23	23,
26	1	12	-	12
27	1	-	4	4,
28	0	5	36	5, 36
29	0	-	-	-
30	1	32	40	32, 40
31	1	19	7, 26	19
32	1	30, 40	-	30, 40
33	0	-	34, 45	-
34	1	-	33, 45	-
35	0	-	-	-
36	0	-	-	-
37	0	-	15, 36	-
38	0	-	10, 21, 22, 24, 25	-
39	1	-	41	-
40	1	32	30	32, 30
41	0	-	39, 45	-
42	1	-	45	-
43	1	-	-	-
44	0	-	33, 34	-
45	1	-	42, 50	-
46	0	-	43, 45, 47	-
47	0	-	43, 46,	-
48	1	-	36, 49, 50	-
49	1	-	48	-
50	1	-	46, 48	-

Tabela 6.4 – Indicação de similaridade apontada pelos especialistas no domínio da aplicação

6.4.2 Descrição dos testes

O objetivo dos testes é verificar a capacidade do modelo quanto a identificação de similaridades entre duas solicitações quaisquer, através do estabelecimento de um índice que represente adequadamente essa situação. Uma vez que tal objetivo seja alcançado, torna-se viável a reutilização de componentes e procedimentos, bem como o conhecimento prévio de problemas que possam surgir.

O índice final de similaridade é sempre obtido pela composição dos índices parciais obtidos por duas etapas de comparação: elementos estruturados (primeira fase da comparação) e elementos em linguagem natural, ou não estruturados (segunda fase da comparação). A contribuição de cada um desses elementos no índice final precisa ser ajustada para a obtenção de resultados adequados e, para tal, no início da etapa de testes foi feita essa calibragem. Para permitir a observação de valores intermediários para todos os casos, o limiar de comparação que é utilizado como filtro entre as duas etapas foi ajustado inicialmente para 0 (zero), fazendo com que todas as solicitações fossem processadas tanto em relação aos elementos estruturados quanto em relação aos não estruturados. Ao final da calibragem dos pesos dos atributos estruturados, procedeu-se ao estabelecimento definitivo do limiar que indica quais solicitações terão os elementos textuais processados.

Finda a calibragem o sistema foi considerado apto ao processamento da comparação de todas as solicitações, trazendo, como resultado, uma matriz de similaridade global. Sobre essa matriz de similaridade fez-se a primeira análise de resultados, baseada nos critérios estabelecidos no item 6.4.3.

No capítulo 7 apresenta-se um planejamento mais detalhado dos testes e dos cenários que foram avaliados em cada uma das etapas de avaliação.

6.4.3 Critérios de avaliação do modelo

Os resultados da aplicação são considerados bem sucedidos quando os valores resultantes da análise estatística dos índices de similaridade entre as solicitações apontam para a capacidade de agregar facilidades ao processo de desenvolvimento de *software*.

Essa análise leva em conta duas características, descritas a seguir:

a) quanto a classificação correta das solicitações similares, considerando a existência de três classes distintas (não similar, similar e muito similar), e

b) quanto a correta ordenação dos índices de similaridade entre as solicitações processadas, considerando a ordem apontada pelos especialistas no domínio.

Para avaliação dos resultados da classificação, foi utilizada uma matriz de confusão construída a partir da comparação entre os resultados esperados e os resultados obtidos; essa comparação visou determinar o grau de acerto do sistema em relação à indicação prévia de que se dispunha. Os resultados foram posicionados em cada uma das células, conforme a classificação do sistema e a classificação prévia por um especialista.

Esperado (especialista) \ Obtido (sistema)	**Verdadeiro**	**Falso**
Verdadeiro	VP	FN
Falso	FP	VN

Tabela 6.5– Formatação geral da matriz de confusão utilizada

Na avaliação dos resultados, as matrizes construídas são organizadas da seguinte forma:

linhas: Cada linha expressa o resultado de uma classificação esperada, de acordo com uma avaliação prévia feita por especialistas no domínio do problema;

colunas: Cada coluna apresenta os resultados obtidos pelo sistema, com a aplicação do modelo para a classificação esperada;

VP : Representam os valores de verdadeiros positivos, ou seja, aqueles valores em que a classificação obtida automaticamente pelo sistema coincidiu com o valor esperado para elementos pertencentes a classe tratada;

VN: Representam os valores de verdadeiros negativos, ou seja, valores em que a expectativa baseada na indicação dos especialistas era de não pertencerem a classe tratada e assim foram corretamente classificados;

FP: Representam os casos em que a classificação feita pelo sistema indicou pertencerem a classe tratada enquanto a expectativa dos especialistas era que não pertencessem;

FN: Representam os elementos classificados pelo sistema como não pertencentes à classe e que os especialistas consideraram pertencer.

Para a análise da classificação foram obtidos os valores de acurácia, sensibilidade, erros, e de precisão, cujas definições e fórmulas de cálculo, para efeito deste trabalho, são as seguintes:

a1) Acurácia → Porcentagem global de acertos quanto a existência de similaridade (independente de grau) entre duas solicitações avaliadas, apurada de acordo com a formula indicada em (6.1).

$$A = 100 \; x \; \frac{(VP+VN)}{(VP+VN+FP+FN)} \qquad (6.1)$$

em que : A = acurácia ;
VP = número de verdadeiros positivos ;
VN = número de verdadeiros negativos ;
FP = número de falsos positivos ;
FN = número de falsos negativos.

Formula 6.1 – Apuração da acurácia.

a2) Sensibilidade → Capacidade de distinguir os valores com similaridade dentro do conjunto de solicitações em que ela efetivamente exista, calculado por:

$$S = 100 \; x \; \frac{VP}{(VP+FN)} \qquad (6.2)$$

em que : S = sensibilidade ;
VP , VN , FP e FN como descrito em 6.1 *;*

Formula 6.2 – Apuração da sensibilidade.

a3) Especificidade → Corresponde a capacidade de distinguir os valores não significativos dentro do conjunto, representada pela taxa de acertos em relação aos valores classificados corretamente como não similares:

$$E = 100 \; x \; \frac{VN}{(VP+VN)} \qquad (6.3)$$

em que : E = especificidade ;
VP , VN , FN como descrito em 6.1 *;*

Formula 6.3 – Apuração da especificidade.

a4) Média harmônica → Corresponde a uma média ponderada dos valores de especificidade e sensibilidade e dá a informação quanto a habilidade do classificador em acertos, tanto para valores de verdadeiros positivos quanto para valores de verdadeiros negativos:

$$F1 = 100 \; x \; \frac{(2 \; x \; S \; x \; E)}{(S+E)} \qquad (6.4)$$

em que : F1 = média harmônica ;
S e E como definidas em (6.2) e (6.3) ;

Formula 6.4 – Apuração da média harmônica

Essas medidas foram comparadas com as seguintes réguas de avaliação, estabelecidas, também, pelo consenso entre os especialistas:

Quanto a acurácia do modelo

Qualificação	Faixa de valores
Inaceitável	Abaixo de 50%
Aceitável	Entre 50% e 80%
Desejável	Acima de 80%

Tabela 6.6 – Régua de comparação para classificação da acurácia.

Quanto à sensibilidade do modelo

Qualificação	Faixa de valores
Inaceitável	Abaixo de 50%
Aceitável	Entre 50% e 80%
Desejável	Acima de 80%

Tabela 6.7 – Régua de comparação para classificação da sensibilidade.

c) Quanto à especificidade do modelo

Qualificação	Faixa de valores
Inaceitável	Acima de 50%
Aceitável	Entre 20% e 50%
Desejável	Abaixo de 20%

Tabela 6.8 – Régua de comparação para classificação quanto ao nível de erros.

d) Quanto à média harmônica do modelo

Qualificação	Faixa de valores
Inaceitável	Abaixo de 50%
Aceitável	Entre 50% e 80%
Desejável	Acima de 80%

Tabela 6.9 – Régua de comparação para a precisão dos índices encontrados.

Em relação à ordenação, faz-se uma avaliação da precisão quanto a taxa de acerto do modelo, utilizando-se para isso a comparação entre dois vetores montados com as ordens de similaridades apontados pelo sistema e pelos especialistas, utilizando a métrica conhecida como distância de Damerau-Levenshtein explicada em detalhes no capítulo 3, exceto quanto a deleção de elementos no final do vetor gerado pelo sistema.

A taxa de acerto é obtida pelo distanciamento entre esses dois vetores, considerando para isso o tamanho do vetor indicado pelo especialista e a quantidade de movimentações (trocas, inserções ou exclusões) necessárias para igualar a esse o vetor com a ordem obtida pelo sistema. O exemplo abaixo ilustra essa apuração:

Ordem indicada pelo especialista

31	28	14	27	11

Ordem do sistema

31	14	28	11

Tem-se, então, um vetor de tamanho = 5, obtido pela indicação dos especialistas, com 1 troca e 1 inserção necessárias para igualar os dois.

Para apuração da taxa de acerto, foram avaliadas as ordenações obtidas para todas as solicitações e computada uma média ponderada pelo tamanho dos vetores. A fórmula 6.5 indica como será feita essa apuração.

$$Ta = \frac{\sum_{i=1}^{i=ns} \frac{(Tv_i - E_i)}{Tv_i}}{ns} \qquad (6.5)$$

em que : Ta *é a taxa de acerto global do sistema ;*
Tv_i *é o tamanho do* $i-$*ésimo vetor ;*
E_i *é a quantidade de erros (trocas / inversoes / exclusões) ;*
ns *é o número de solicitações processadas ;*
Obs.: *Desconsiderar as solicitações em que* $Tv_i = 0$ *;*

Fórmula 6.5 – Apuração da taxa de acertos quanto a ordenação obtida pelo sistema.

6.5 Ambiente de avaliação

A avaliação dos resultados foi feita em três equipamentos distintos, todos com a mesma configuração de *software* instalada, divergindo apenas quanto aos recursos de *hardware* disponível. Todas as comparações resultaram na mesma matriz de similaridade, fato que estava de acordo com o esperado, posto que os dados e os pesos relativos a cada atributo estavam igualmente configurados.

Essa tripla avaliação teve a finalidade de verificar se os resultados poderiam ser de alguma forma afetados por diferenças de *hardware* e ainda avaliar o tempo de execução gasto em verificação mais extensa, já que seriam efetuadas 2.500 comparações entre solicitações e estima-se que o modelo, quando colocado em produção, deverá ter a capacidade de tratar bases extensas em curto espaço de tempo.

Os equipamentos utilizados apresentavam a seguinte configuração:

Componentes	Un.	Equipamento 1	Equipamento 2	Equipamento 3
Marca e modelo		Megaware-MegaHome	Positivo	Positivo Móbile
Processador		Pentium 4 - Modelo 506	Pentium D (Dual Core) Modelo D-925	Celeron M - Modelo 430
Freqüência de Clock	MHz	2,66	2	1,73
Memória cache L1	Kb	16	2 x 64	32
Memória cache L2	Mb	1024	2 x 1024	1024
Memória RAM	Mb	512	1024	512
Capacidade do disco rígido	Gb	80	160	60
Velocidade do disco rígido	rpm	7200	7200	5400

Tabela 6.10 – Equipamentos utilizados nos testes realizados

Todos os equipamentos utilizavam a mesma configuração de *software*, que era a seguinte:

- Sistema Operacional Windows XP – service pack 2
- Servidor HTTP Apache versão 2.2.4
- Banco de dados MySQL versão 5.0.37
- Módulo de extensão PHP versão 5.2.3

7 Estudo de caso: desenvolvimento de software em um banco público (parte II – Avaliação dos resultados)

Este capítulo descreve a o planejamento e a avaliação dos resultados obtidos pelo emprego do modelo proposto ao estudo de caso iniciado no capítulo 6. Apresenta-se não apenas uma avaliação quantitativa e qualitativa dos resultados, mas, também, uma análise de fatores que influenciaram a obtenção de tais resultados. Espera-se que essa análise seja capaz de facilitar avaliações quanto a real possibilidade de seu uso e direcionar trabalhos futuros.

7.1 Planejamento dos testes

Para uma correta avaliação do modelo, os testes foram conduzidos em duas etapas distintas, assim idealizadas:

1 ª etapa: Avaliação do modelo:

A primeira etapa consistiu na construção de uma matriz de similaridade global, obtida a partir das solicitações registradas na base de conhecimentos do sistema, destinada a validação do modelo, como um todo. Esta etapa envolveu a comparação exaustiva de cada uma das solicitações da base com todas as outras, para, em seguida, obter os índices de acurácia, sensibilidade, especificidade, média harmônica,e taxa de acertos. Assim que tais índices se situaram dentro dos valores desejáveis, pelo ajuste dos pesos para cada um dos elementos tratados pela aplicação, passou-se a segunda etapa da avaliação.

2 ª etapa: Avaliação do comportamento do modelo em diferentes cenários

Na segunda etapa do teste, foram selecionadas algumas novas solicitações para processamento pela ferramenta de suporte ao modelo, divididas conforme uma classificação prévia dos índices de similaridade apresentados.

- **Cenário 1** – Solicitações com baixa ou nenhuma similaridade com as demais solicitações registradas na base de casos mantida pelo sistema (índices de similaridade esperados na faixa de 0,000 até 0,499);
- **Cenário 2** – Solicitações com média similaridade na base de casos (índices de similaridade esperados na faixa de 0,500 até 0,699).
- **Cenário 3** – Solicitações com alta similaridade em relação às solicitações

registradas na base de casos (índices esperados na faixa de 0,700 até 1,000).

Os resultados obtidos pela aplicação do modelo a cada um desses cenários serão então avaliados, quanto à sua correção e quanto a fatores que possam influenciar positivamente ou negativamente nos índices apurados, permitindo dessa forma prever seu comportamento quando submetido a situações reais.

7.2 Avaliação do modelo

Essa avaliação consiste na primeira etapa dos testes descritos na seção 7.1 e foram realizados em todos os equipamentos listados na seção 6.5, com resultados idênticos, quanto aos valores dos índices de similaridade obtidos. Esse resultado, embora previsível, nos leva a uma primeira verificação importante que é a independência de plataforma de *hardware* e *software*. Os resultados para essa etapa dos testes estão assinalados de acordo convenção de formatação mostrada na tabela 7.1.

Tipo de resultado	Formato do resultado mostrado na planilha
Verdadeiro positivo → Resultado corretamente classificado quanto a existência de similaridade entre as solicitações para valores em que a expectativa era de obtenção de um índice médio ou alto de similaridade entre o caso comparado e o caso registrado na base.	**0,999**
Verdadeiro negativo → Resultado corretamente classificado quanto a inexistência de similaridade. Enquadram-se nesta situação todos os casos em que a similaridade esperada deveria ser classificada como baixa ou nenhuma.	0,499 ou - (traço)
Falso Positivo → Resultado incorretamente classificado para solicitações em que a expectativa de similaridade era pouca ou nenhuma e os resultados obtidos pelo sistema apontaram para a existência de média ou alta similaridade	**0,555**
Falso Negativo → Resultado incorretamente apurado pelo sistema, nos casos em que a expectativa era de obtenção de média ou alta similaridade e os valores obtidos situaram-se na faixa de baixa ou nenhuma similaridade.	0,455

Tabela 7.1–Formatação utilizada nas tabelas de resultado

Os índices de similaridade obtidos pela aplicação do modelo com o auxílio da ferramenta FSSIA, nesta primeira etapa da avaliação, estão descritos na tabela 7.2, que foi

dividade em quatro páginas para facilitar a leitura dos dados em razão da quantidade de informações e de sua extensão. Note-se que essa tabela é o resultado da execução de 2500 comparações entre solicitações e está assim organizada:

linhas: Temos em cada uma das linhas o identificador da solicitação que estava sendo comparada com as demais, ou seja a linha 1 apresenta o resultado da comparação da solicitação de número 1 com todas as demais solicitações da base;

colunas: Apresenta a solicitação com a qual foi feita a comparação. Na coluna 4 da linha 1, por exemplo, encontramos o índice de similaridade obtido quando a solicitação de número 1 foi comparada com a solicitação de número 4 e assim sucessivamente.

As células preenchidas com um traço representam resultados cuja comparação dos elementos estruturados resultou em um valor abaixo do limiar estabelecido, e portanto, não foi efetuada a comparação de elementos não estruturados. Esses valores são assumidos como de baixa ou nenhuma similaridade.

Ressalta-se que o resultado da comparação entre as células é dependente de qual célula é o ponto de partida da comparação, uma vez que a comparação dos elementos de texto é direcionada pela matriz de termos resultantes do processamento do caso em estudo. Para exemplificar essa situação, considere-se a seguinte situação:

Texto 1: O carro deverá ser pintado na cor vermelha;

Texto 2: A execução completa dos serviços implica que o carro deverá ser pintado na cor vermelha, as rodas na cor azul e os para-choques na cor verde.

Na comparação da frase 1 com a frase 2, obtém-se 100% de similaridade, posto que todos os seus termos estão ali contidos; o mesmo não ocorre quando se inverte a comparação, uma vez que nem todos os termos da frase 2 estão contidos na frase 1.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	**1,000**	0,334	-	0,442	**0,515**	-	-	0,401	**0,510**	-	-	**0,521**	0,471
2	0,296	**1,000**	-	-	0,245	0,318	-	0,282	-	0,444	0,449	-	-
3	-	-	**1,000**	-	0,222	0,216	-	0,215	0,228	0,210	0,200	-	0,235
4	0,404	-	-	**1,000**	0,466	-	-	**0,686**	**0,735**	-	-	0,454	0,382
5	**0,504**	0,249	0,222	0,464	**1,000**	-	-	-	0,434	0,343	0,301	-	0,473
6	-	0,317	0,216	-	-	**1,000**	0,494	-	0,291	-	-	0,356	0,277
7	-	-	-	-	-	0,489	**1,000**	-	-	-	-	-	-
8	0,392	0,331	0,215	**0,670**	-	-	-	**1,000**	**0,780**	0,382	-	0,427	0,333
9	0,458	-	0,228	**0,726**	0,421	0,334	-	**0,773**	**1,000**	0,342	-	**0,504**	0,431
10	-	0,414	0,210	-	0,308	-	-	0,347	0,312	**1,000**	**0,525**	-	-
11	-	0,431	0,200	-	0,299	-	-	-	-	**0,549**	**1,000**	-	-
12	0,381	-	-	0,350	-	0,280	-	0,376	0,388	-	-	**1,000**	**0,631**
13	0,415	-	0,235	0,385	0,418	0,266	-	0,335	0,420	-	-	**0,731**	**1,000**
14	-	-	-	0,345	-	-	-	-	-	-	-	0,206	-
15	-	-	-	-	-	0,295	0,348	-	-	-	-	-	-
16	-	-	-	-	-	0,289	-	-	-	-	-	-	-
17	-	0,286	-	-	-	0,273	0,319	0,332	0,340	0,295	-	0,399	0,427
18	-	-	0,219	-	-	0,313	0,306	-	-	0,362	0,222	-	-
19	-	0,318	-	0,304	-	0,338	-	0,327	-	-	-	0,392	0,314
20	-	0,269	0,320	-	-	0,259	0,409	-	-	-	-	0,271	0,200
21	-	0,403	0,326	0,282	0,305	-	-	-	-	**0,539**	0,470	-	-
22	0,311	0,415	0,202	0,369	0,348	0,295	0,350	0,370	0,365	**0,602**	0,474	0,410	0,309
23	-	0,438	0,351	-	0,303	-	-	-	-	**0,512**	0,479	-	-
24	-	0,451	0,306	-	0,346	0,295	0,342	-	0,346	**0,600**	0,435	-	0,315
25	0,290	0,374	-	0,302	0,255	0,252	0,293	0,316	0,319	0,337	0,440	0,368	-
26	0,433	0,325	-	0,385	0,382	0,310	-	0,327	0,368	-	-	**0,725**	0,484
27	0,415	-	-	0,490	0,390	-	-	0,377	0,459	0,359	-	0,477	0,368
28	0,481	-	-	0,373	**0,683**	-	-	-	0,334	-	-	-	-
29	0,482	0,316	0,202	0,471	0,493	0,315	0,335	0,359	0,466	0,337	0,320	**0,509**	**0,506**
30	-	-	-	-	-	-	-	-	-	-	-	-	-
31	-	0,334	-	-	-	0,400	**0,591**	-	0,263	-	0,275	0,444	0,333
32	-	-	-	-	-	-	-	-	-	-	-	-	-
33	-	-	-	-	-	0,240	0,345	-	-	-	-	-	-
34	-	-	0,289	-	-	0,329	0,307	-	-	0,202	0,202	-	-
35	-	-	-	-	-	0,387	-	-	-	-	-	-	0,381
36	-	-	-	0,316	-	0,305	-	0,282	0,298	-	-	-	-
37	-	-	-	-	-	0,271	0,356	-	-	-	-	0,345	-
38	0,251	0,411	-	0,328	0,275	0,307	-	0,363	0,376	**0,541**	**0,508**	0,380	-
39	-	0,239	0,274	0,263	-	0,285	0,304	-	-	0,222	0,250	0,317	-
40	-	-	-	-	-	-	-	-	-	-	-	-	-
41	-	0,306	0,477	-	0,300	0,332	0,344	-	0,297	0,235	0,332	-	-
42	-	0,205	-	-	-	-	0,392	-	-	-	-	-	-
43	-	-	0,245	-	-	0,203	0,330	-	-	-	-	-	-
44	-	-	0,229	-	0,203	0,373	0,303	-	-	0,216	0,326	-	-
45	-	-	0,229	-	-	0,280	0,343	-	-	0,303	0,329	-	-
46	-	0,344	0,245	0,368	0,355	0,383	0,327	-	0,346	0,305	0,401	-	0,360
47	-	-	0,239	-	0,276	0,308	0,291	-	-	0,235	0,366	-	-
48	-	0,316	0,327	0,250	0,301	0,338	0,313	0,309	0,252	-	-	0,428	0,325
49	-	0,343	0,313	-	-	0,389	0,396	-	-	-	-	0,396	0,279
50	-	0,317	0,261	-	0,350	0,218	0,235	-	-	0,308	0,374	-	-

Tabela 7.2 – Resultados da comparação efetuada pelo sistema
(continua na página seguinte)

	14	15	16	17	18	19	20	21	22	23	24	25	26
1	-	-	-	-	-	-	-	-	0,363	-	-	0,338	0,656
2	-	-	-	0,372	-	0,339	0,241	0,442	0,419	0,473	0,441	0,395	0,435
3	-	-	-	-	0,219	-	0,267	0,283	0,202	0,387	0,294	-	-
4	0,441	-	-	-	-	0,401	-	0,294	0,424	-	-	0,364	0,596
5	-	-	-	-	-	-	-	0,386	0,390	0,336	0,380	0,273	0,572
6	-	0,291	0,309	0,343	0,289	0,384	0,259	-	0,306	-	0,305	0,272	0,420
7	-	0,343	-	0,420	0,278	-	0,318	-	0,369	-	0,343	0,320	-
8	-	-	-	0,443	-	0,435	-	-	0,443	-	-	0,455	0,480
9	-	-	-	0,476	-	-	-	-	0,438	-	0,387	0,453	0,498
10	-	-	-	0,441	0,375	-	-	0,664	0,646	0,576	0,595	0,364	-
11	-	-	-	-	0,222	-	-	0,504	0,488	0,497	0,422	0,468	-
12	0,206	-	-	0,481	-	0,377	0,219	-	0,340	-	-	0,304	0,801
13	-	-	-	0,533	-	0,386	0,200	-	0,311	-	0,339	-	0,618
14	1,000	0,431	0,332	0,461	-	0,354	-	-	-	-	-	-	0,494
15	0,415	1,000	0,512	-	-	-	0,340	-	-	-	-	-	-
16	0,273	0,377	1,000	0,501	-	0,336	-	-	-	-	-	-	0,525
17	0,286	-	0,463	1,000	-	0,495	0,258	0,383	0,425	0,341	-	-	0,561
18	-	-	-	-	1,000	-	0,258	-	-	0,203	-	-	-
19	0,321	-	0,412	0,514	-	1,000	0,381	0,362	0,359	-	-	0,305	0,466
20	-	0,357	-	0,402	0,258	0,424	1,000	0,331	-	-	-	-	-
21	-	-	-	0,407	-	0,301	0,248	1,000	0,575	0,568	0,532	0,424	-
22	-	-	-	0,515	-	0,360	-	0,671	1,000	0,520	0,757	0,458	-
23	-	-	-	0,424	0,203	-	-	0,606	0,533	1,000	0,471	0,519	-
24	-	-	-	-	-	-	-	0,656	0,867	0,487	1,000	0,458	-
25	-	-	-	-	-	0,284	-	0,415	0,435	0,480	0,429	1,000	0,440
26	0,426	-	0,447	0,536	-	0,419	-	-	-	-	-	0,308	1,000
27	-	-	-	0,493	-	-	-	0,339	0,419	-	-	0,329	0,514
28	0,294	-	-	-	-	-	-	-	-	-	-	-	-
29	0,273	0,245	-	0,399	-	0,353	-	-	0,436	0,303	0,400	0,336	0,514
30	-	-	-	-	-	-	-	-	-	-	-	0,205	-
31	0,320	-	0,438	0,546	0,325	0,766	0,294	0,427	0,418	0,374	0,350	0,335	0,501
32	-	-	-	-	-	-	-	-	-	-	-	-	-
33	-	0,266	0,301	-	0,359	-	0,226	-	-	-	-	-	-
34	-	0,312	0,332	-	0,260	-	0,259	-	0,210	-	0,219	0,220	-
35	0,493	-	0,437	0,564	-	0,472	0,364	-	-	-	-	-	0,591
36	0,383	0,356	0,471	0,479	-	0,357	-	-	-	-	-	-	0,491
37	0,287	0,358	-	-	0,222	0,312	0,305	-	-	-	-	-	0,375
38	0,200	-	-	0,416	-	-	0,278	0,523	0,569	0,410	0,542	0,527	0,350
39	-	0,276	-	-	0,260	-	0,357	0,305	0,286	-	0,286	0,265	-
40	-	-	-	-	-	-	-	-	-	-	-	-	-
41	-	0,327	-	-	0,273	-	0,344	0,295	0,326	-	0,329	0,307	-
42	-	-	-	-	0,455	0,396	0,346	-	-	-	-	-	-
43	-	-	-	-	0,423	-	0,375	-	-	-	-	-	-
44	-	0,220	-	-	0,274	-	0,245	-	-	-	0,233	0,298	-
45	-	-	-	-	0,506	-	0,500	0,360	-	-	0,317	-	-
46	-	0,281	-	0,530	0,606	0,357	0,402	0,391	0,455	0,393	0,454	0,379	-
47	-	0,203	-	-	0,524	-	0,264	0,268	-	-	0,327	-	-
48	-	0,312	-	0,481	0,314	0,379	0,295	0,465	0,396	0,349	0,376	-	0,520
49	-	-	-	-	0,337	0,409	0,245	0,413	-	-	-	0,294	0,494
50	-	-	-	0,421	0,358	-	0,388	0,407	0,344	0,392	0,337	-	-

Tabela 7.2 – Resultados da comparação efetuada pelo sistema

(continua na página seguinte)

	27	28	29	30	31	32	33	34	35	36	37	38	39
1	0,469	0,414	0,546	-	-	-	-	-	-	-	-	0,265	-
2	-	-	0,314	-	0,347	-	-	-	-	-	-	0,402	0,304
3	-	-	0,202	-	-	-	-	0,282	-	-	-	-	0,318
4	0,463	0,321	0,458	-	-	-	-	-	-	0,419	-	0,359	0,291
5	0,407	0,534	0,550	-	-	-	-	-	-	-	-	0,266	-
6	-	-	0,310	-	0,390	-	0,240	0,386	0,387	0,415	0,271	0,342	0,306
7	-	-	0,320	-	0,633	-	0,267	0,302	-	-	0,337	-	0,372
8	0,499	-	0,364	-	-	-	-	-	-	0,414	-	0,400	-
9	0,542	0,288	0,455	-	0,316	-	-	-	-	0,443	-	0,385	-
10	0,385	-	0,376	-	-	-	-	0,202	-	-	-	0,479	0,222
11	-	-	0,296	-	0,291	-	-	0,202	-	-	-	0,443	0,296
12	0,404	-	0,437	-	0,367	-	-	-	-	-	0,293	0,301	0,299
13	0,417	-	0,477	-	0,354	-	-	-	0,326	-	-	-	-
14	-	0,297	0,269	-	0,363	-	-	-	0,484	0,584	0,336	0,200	-
15	-	-	0,243	-	-	-	0,266	0,319	-	0,486	0,370	-	0,343
16	-	-	-	-	0,330	-	0,235	0,284	0,355	0,532	-	-	-
17	0,330	-	0,288	-	0,473	-	-	-	0,458	0,510	-	0,321	-
18	-	-	-	-	0,293	-	0,314	0,260	-	-	0,222	-	0,260
19	-	-	0,356	-	0,681	-	-	-	0,393	0,451	0,307	-	-
20	-	-	-	-	0,364	-	0,226	0,259	0,370	-	0,342	0,298	0,449
21	0,343	-	-	-	0,338	-	-	-	-	-	-	0,434	0,367
22	0,468	-	0,420	-	0,373	-	-	0,210	-	-	-	0,496	0,337
23	-	-	0,288	-	0,361	-	-	-	-	-	-	0,391	-
24	-	-	0,429	-	0,317	-	-	0,219	-	-	-	0,518	0,353
25	0,340	-	0,305	0,205	0,326	-	-	0,220	-	-	-	0,435	0,294
26	0,419	-	0,420	-	0,403	-	-	-	0,458	0,458	0,280	0,260	-
27	1,000	0,316	0,536	-	-	-	-	-	-	0,428	-	0,305	0,328
28	0,378	1,000	0,501	-	-	-	0,261	0,500	-	0,595	0,427	-	-
29	0,492	0,358	1,000	-	0,357	0,202	0,203	0,287	-	0,407	-	0,261	0,260
30	-	-	-	1,000	-	0,837	-	-	-	-	-	-	0,201
31	-	-	0,379	-	1,000	-	-	-	0,414	0,500	0,254	0,279	-
32	-	-	0,202	0,738	-	1,000	-	-	-	-	-	-	-
33	-	0,261	0,203	-	-	-	1,000	0,415	-	0,465	0,422	-	0,255
34	-	0,331	0,276	-	-	-	0,415	1,000	-	-	0,250	0,273	0,356
35	-	-	-	-	0,485	-	-	-	1,000	0,477	-	-	-
36	0,318	0,323	0,307	-	0,339	-	0,297	-	0,358	1,000	0,365	0,263	0,320
37	-	0,305	-	-	0,261	-	0,329	0,250	-	0,525	1,000	0,303	0,322
38	0,355	-	0,264	-	0,294	-	-	0,287	-	0,331	0,325	1,000	0,352
39	0,251	-	0,232	0,201	-	-	0,255	0,334	-	0,353	0,289	0,295	1,000
40	-	-	-	0,668	-	0,652	-	-	-	-	-	-	-
41	-	0,209	0,305	-	0,256	-	0,352	0,301	-	0,430	0,356	0,351	0,673
42	-	-	-	0,222	0,405	0,219	-	0,231	-	-	0,320	-	0,415
43	-	-	-	-	-	-	0,359	0,324	-	-	-	-	0,384
44	-	0,298	-	-	-	-	0,595	0,681	-	-	0,334	-	0,361
45	-	-	-	-	-	-	0,327	0,270	-	-	0,344	-	0,372
46	-	-	0,346	-	0,373	-	0,330	0,447	-	-	0,281	0,387	0,413
47	-	-	-	-	-	-	0,445	0,384	-	-	0,216	0,294	0,359
48	0,330	-	0,284	-	0,371	-	0,262	0,229	0,272	0,521	0,394	0,253	0,391
49	0,332	-	-	-	-	-	-	0,273	-	-	0,322	0,326	0,424
50	-	-	-	-	0,422	-	-	0,373	-	-	-	-	0,460

Tabela 7.2 – Resultados da comparação efetuada pelo sistema

(continua na página seguinte)

	40	41	42	43	44	45	46	47	48	49	50
1	-	-	-	-	-	-	-	-	-	-	-
2	-	0,318	0,205	-	-	-	0,365	-	0,335	0,338	0,292
3	-	0,501	-	0,245	0,229	0,229	0,245	0,239	0,368	0,306	0,261
4	-	-	-	-	-	-	0,385	-	0,278	-	-
5	-	0,313	-	-	0,203	-	0,381	0,269	0,359	-	0,350
6	-	0,393	-	0,203	0,375	0,297	0,370	0,291	0,389	0,434	0,218
7	-	0,379	0,331	0,308	0,289	0,358	0,305	0,258	0,328	0,416	0,235
8	-	-	-	-	-	-	-	-	0,342	-	-
9	-	0,309	-	-	-	-	0,366	-	0,252	-	-
10	-	0,235	-	-	0,216	0,347	0,287	0,235	-	-	0,279
11	-	0,411	-	-	0,320	0,365	0,377	0,305	-	-	0,338
12	-	-	-	-	-	-	-	-	0,324	0,329	-
13	-	-	-	-	-	-	0,328	-	0,315	0,330	-
14	-	-	-	-	-	-	-	-	-	-	-
15	-	0,321	-	-	0,220	-	0,273	0,203	0,337	-	-
16	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	0,361	-	0,396	-	0,285
18	-	0,273	0,422	0,423	0,274	0,528	0,554	0,495	0,342	0,350	0,360
19	-	-	0,337	-	-	-	0,315	-	0,380	0,417	-
20	-	0,493	0,419	0,418	0,245	0,524	0,379	0,264	0,320	0,294	0,460
21	-	0,324	-	-	-	0,291	0,390	0,221	0,371	0,374	0,310
22	-	0,391	-	-	-	-	0,414	-	0,392	-	0,310
23	-	-	-	-	-	-	0,330	-	0,353	-	0,309
24	-	0,401	-	-	0,233	0,343	0,409	0,333	0,338	-	0,316
25	-	0,318	-	-	0,370	-	0,357	-	-	0,269	-
26	-	-	-	-	-	-	-	-	0,331	0,364	-
27	-	-	-	-	-	-	-	-	0,341	0,372	-
28	-	0,209	-	-	0,359	-	-	-	-	-	-
29	-	0,347	-	-	-	-	0,345	-	0,313	-	-
30	0,668	-	0,222	-	-	-	-	-	-	-	-
31	-	0,258	0,361	-	-	-	0,377	-	0,367	-	0,405
32	0,642	-	0,219	-	-	-	-	-	-	-	-
33	-	0,438	-	0,457	0,432	0,469	0,387	0,401	0,357	-	-
34	-	0,301	0,231	0,326	0,579	0,270	0,385	0,332	0,229	0,281	0,358
35	-	-	-	-	-	-	-	-	0,297	-	-
36	-	0,358	-	-	-	-	-	-	0,347	-	-
37	-	0,407	0,290	-	0,320	0,387	0,269	0,216	0,369	0,399	-
38	-	0,390	-	-	-	-	0,348	0,324	0,263	0,345	-
39	-	0,620	0,311	0,289	0,311	0,330	0,337	0,322	0,329	0,341	0,333
40	1,000	-	0,304	-	0,201	-	-	-	-	-	-
41	-	1,000	0,345	0,381	0,340	0,382	0,369	0,357	0,410	0,352	0,416
42	0,300	0,480	1,000	0,458	0,217	0,548	0,481	0,244	0,340	0,390	0,395
43	-	0,466	0,398	1,000	0,267	0,501	0,514	0,486	0,423	-	0,381
44	0,201	0,352	0,217	0,267	1,000	0,348	0,420	0,287	0,216	0,205	0,348
45	-	0,396	0,477	0,528	0,330	1,000	0,476	0,360	0,354	-	0,528
46	-	0,430	0,438	0,603	0,435	0,509	1,000	0,595	0,429	0,471	0,404
47	-	0,372	0,244	0,525	0,287	0,377	0,551	1,000	0,400	0,303	0,278
48	-	0,418	0,292	0,352	0,216	0,395	0,447	0,272	1,000	0,532	0,603
49	-	0,370	0,339	-	0,205	-	0,393	0,320	0,470	1,000	0,421
50	-	0,470	0,392	0,437	0,412	0,552	0,415	0,278	0,495	0,432	1,000

Tabela 7.2– Resultados da comparação efetuada pelo sistema

7.2.1 Análise dos resultados da primeira etapa

A avaliação dos resultados para essa primeira etapa foi feita de acordo com as condições e critérios estabelecidas no item 6.4.3 (Modelagem dos testes – critérios de avaliação) e, para tal, procedeu-se primeiramente a construção de uma matriz de confusão, com os valores obtidos pelo sistema.

Uma matriz de confusão é normalmente construída a partir da comparação entre os resultados esperados e os resultados obtidos, visando determinar o grau de acerto do sistema em relação à indicação prévia de que se dispõe. Os resultados são então posicionados em cada uma das células, conforme a classificação do sistema e a classificação prévia por um especialista.

Esperado \ Obtido	**Verdadeiro**	**Falso**
Verdadeiro	VP	FN
Falso	FP	VN

Tabela 7.3 – Formatação geral da matriz de confusão utilizada

Na avaliação dos resultados, as matrizes construídas estão organizadas da seguinte forma:

linhas: Cada linha expressa o resultado de uma classificação esperada, de acordo com uma avaliação prévia feita por especialistas no domínio do problema;

colunas: Cada coluna apresenta os resultados obtidos pelo sistema, com a aplicação do modelo para a classificação esperada;

VP : Representam os valores de verdadeiros positivos, ou seja, aqueles valores em que a classificação obtida automaticamente pelo sistema coincidiu com o valor esperado para elementos pertencentes a classe tratada;

VN: Representam os valores de verdadeiros negativos, ou seja, valores em que a expectativa baseada na indicação dos especialistas era de não pertencerem a classe tratada e assim foram corretamente classificados;

FP: Representam os casos em que a classificação feita pelo sistema indicou pertencerem a classe tratada enquanto a expectativa dos especialistas era que não pertencessem;

FN: Representam os elementos classificados pelo sistema como não pertencentes à classe e que os especialistas consideraram pertencer.

Os resultados obtidos pelo sistema e mostrados na tabela 7.2 foram consolidados e

divididos em três classes distintas, mostrada na tabela 7.4; para utilizar a avaliação proposta na forma do item 4 do capítulo 6, devemos construir uma matriz de confusão para cada uma das classes e apurar os valores médios para determinação dos índices de acurácia, sensibilidade, especificidade e média harmônica, conforme as fórmulas elencadas no item 6.4.3.

Temos então, as seguintes matrizes de confusão.

Matriz geral:

Esperado \ Obtido	**Similaridade baixa**	**Similaridade média**	**Similaridade alta**
Similaridade baixa	2343	16	1
Similaridade média	13	63	0
Similaridade alta	0	2	12

Tabela 7.4 – Matriz de confusão consolidada

Classe: similaridade baixa ou nenhuma

Esperado \ Obtido	**Similaridade baixa**	**Demais**
Similaridade baixa	2343	17
Demais	15	75

Tabela 7.5 –Matriz para a classe similaridade baixa

Classe: Similaridade média

Esperado \ Obtido	**Similaridade média**	**Demais**
Similar	63	13
Demais	19	2355

Tabela 7.6 –Matriz para a classe similaridade média

Classe similaridade alta

Esperado \ Obtido	**Similaridade alta**	**Demais**
Muito similar	12	0
Demais	2	2436

Tabela 7.7 –Matriz para a classe similaridade alta

Matriz resultante (valores médios/normalizados das classes anteriores)

Esperado \ Obtido	Verdadeiro	Falso
Verdadeiro	806 (0,329)	10 (0,004)
Falso	12 (0,005)	1622 (0,662)

Tabela 7.8 – Matriz de confusão resultante

Os valores indicados entre parênteses na matriz 7.8 são os resultado normalizados, em relação a amostra de 2500 casos, desconsiderados os 50 casos em que a comparação foi feita contra a própria solicitação.

Com base nos valores obtidos e nos critérios estabelecidos, procedeu-se ao cálculo dos índices de avaliação do modelo:

a) Quanto à acurácia do modelo.

A = 100 * (VP+VN)/(VP+FP+VN+FN)

A = 100 * (0,329 + 0,662)/ 1

A = 99,1% → Desejável

b) Quanto à sensibilidade do modelo.

S = 100 * (VP)/(VP+FN)

S = 100 * (0,329)/(0,329+0,004)

S = 98,8% → Desejável

c) Quanto à especificidade.

E = 100 * (VN)/(VN + FP)

E = 100 * (0,662)/(0,662+0,005)

E = 99,2% → Desejável

d) Quanto à média harmônica.

F1 = 2 * S * E /(S+E)

F1 = 2 * 0,988 * 0,992 / (0,988+0,992)

P = 99,0% → Desejável

Em relação ao ordenamento dos índices obtidos pelo sistema em comparação aos indicados pelos especialistas, os resultados obtidos estão descritos na tabela 7.9:

Id_solicitação	Ordem de similaridade indicada pelo especialista	Ordem de similaridade obtida pelo sistema	Tv	Te
1	26-29	26-29-12-5-9	2	0
2	-	-	0	0
3	-	-	0	0
4	9-8	9-8-26-28	2	0
5	28	26-29-28	1	1
6	-	-	0	0
7	31	31	1	0
8	9- 4	9-4	2	0
9	8-4-12-27	8-4-27	4	1
10	21-22-23-24-11	21-22-24-23-11	5	1
11	10-21	10-21	2	0
12	26-13	26-13	2	0
13	13-26	13-26-17	2	0
14	36	36	1	0
15	-	16	0	0
16	36	36-26-17	1	0
17	-	26-36	0	0
18	-	46-45	0	0
19	31-17	31-17	2	0
20	45	45	1	0
21	22-23-24-10	22-23-10-24	4	1
22	21-24-10-23	24-21-10-23	4	0
23	21-22-10-24	21-22-10-24	4	0
24	22-21-10-25	22-21-10-38	4	1
25	23	-	1	1
26	12	12-17	1	0
27	4	29-26	1	1
28	5-36	5-36-29	2	0
29	-	26-12-13	0	0
30	32-40	32-40	2	0
31	19	19-7-17-26	1	0
32	30- 40	30-40	2	0
33	45-34	-	2	2
34	45-33	44	2	2
35	-	26-17	0	0
36	-	-	0	0
37	36-15	36	1	1
38	22-24-25-10-21	22-10-24-25-21-11	5	2
39	41	41	1	0
40	32, 30	30-32	2	1
41	39-45	39	2	1
42	45	45	1	0
43	-	46-45	0	0
44	34-33	34-33	2	0
45	50-42	50-43-17	2	1
46	43-45-47	18-43-47-44	3	2
47	46-43-18	46-43-18	3	0
48	49-50-36	50-49-36-26	3	1
49	48	-	1	1
50	-	45	0	0

Tabela 7.9 – Ordenação dos índices obtidos pelo sistema

A avaliação dessas informações aponta para o seguinte resultado, apurado conforme a fórmula definida no item 6.4.3

Taxa de acerto = Σ(Tv-E)/Tv / ns
Taxa de acerto = 27,65 / 39 (considerando apenas as solicitações em que Tv > 0)
Taxa de acerto = 70,90% → Boa

Os resultados obtidos revelam a boa aderência do modelo aos objetivos propostos e sua capacidade de identificar similaridades entre solicitações pode ser considerada como ótima, de acordo com critérios previamente estabelecidos. Percebe-se, de fato, que a aplicação do modelo somente não identificaria similaridades em 7,5% dos casos o que nos permite dizer que para a grande maioria das solicitações o sistema identificaria casos semelhantes e poderia ser utilizado com sucesso para auxiliar tanto o demandante na lapidação de sua solicitação, quanto o desenvolvedor no reuso de soluções já apresentadas.

Ainda em relação aos resultados obtidos nesta etapa do teste, chamou a atenção o fato de que boa parte dos erros encontrados pelo aplicativo ocorriam quando se comparava uma solicitação cuja qualidade da especificação foi considerada fraca pelos especialistas com as solicitações 17 e 26; investigando essa relação, constatou-se que essas duas solicitações eram as de maior nível de detalhamento de toda a base, com uma descrição textual rica das funcionalidades a implementar. Quando uma comparação de termos entre um conjunto pequeno de vocábulos se deu contra um conjunto extenso, boa parte desses vocábulos pode ser identificada, fato esse que justifica a incidência de erros sobre as mesmas solicitações.

Há que se considerar, no entanto, que esses resultados foram obtidos sobre uma base relativamente pequena de casos e que representam solicitações direcionadas a apenas uma das gerências de desenvolvimento da organização alvo. É preciso que sejam feitos mais testes, e eventualmente novas calibragens dos pesos, para que se possa afirmar que tais resultados poderão ser generalizados para todas as demais gerência de desenvolvimento de aplicativos.

7.3 Avaliação do comportamento do modelo em diferentes cenários

Finda a primeira etapa dos testes, o modelo foi considerado apto ao processamento de solicitações em diferentes cenários, para nova análise e avaliação dos resultados. Nesta etapa, um conjunto de cinco solicitações identificadas como pertencentes aos cenários escolhidos foi submetido ao processamento para que se faça uma análise detalhada dos resultados obtidos e para que seja possível estabelecer uma previsão quanto ao padrão de comportamento frente a cada um desses casos.

7.3.1 Cenário 1: Solicitações com baixa ou nenhuma similaridade.

As solicitações escolhidas para a avaliação deste caso precisavam atender a um determinado conjunto de características para possibilitar uma avaliação correta, entre as quais pode-se destacar:

a) Serem direcionadas à mesma gerência escolhida para os casos selecionados para integrar a base de solicitações;

b) Serem direcionadas, prioritariamente, a algum dos sistemas que continham uma base com boa diversidade de casos;

c) Tratarem de assuntos diversos dos tratados nos casos anteriores.

A conjunção desses três fatores nos permitiria explorar a capacidade do modelo e do sistema que o suporta quanto ao tratamento de solicitações que, embora não similares às registradas anteriormente, apresentavam uma configuração que as conduzisse ao tratamento de todos os elementos estruturados e não estruturados, além de representarem amostras de diversos pontos da classe, com índices de similaridades situados na faixa de 0,000 à 0,499 (de 0 a quatrocentos e noventa e nove milésimos).

O resultado desse processamento está expresso nas tabelas seguintes, com as respectivas análises.

Cenário 1 → Caso 1: Solicitação A

Por tratar-se de uma solicitação com baixa ou nenhuma similaridade, a expectativa para esse caso era de que todos índices encontrados pelo aplicativo fossem de valor menor ou igual a 0,499.

Ident. Solicitação	Caso da Base	IS	ISee	ISne
A	41	0,418	0,719	0,283
A	39	0,401	0,684	0,274
A	44	0,392	0,648	0,277
A	45	0,374	0,718	0,219
A	22	0,348	0,603	0,234
A	46	0,341	0,751	0,157
A	34	0,331	0,684	0,172
A	18	0,329	0,683	0,170
A	47	0,310	0,648	0,159
A	38	0,309	0,605	0,175
A	43	0,286	0,680	0,109
A	33	0,262	0,607	0,107
A	32	0,221	0,712	0,000
A	30	0,221	0,712	0,000
A	20	0,209	0,675	0,000
A	40	0,199	0,641	0,000
A	31	0,195	0,628	0,000
A	42	0,189	0,610	0,000
A	35	0,000	0,555	0,000
A	36	0,000	0,528	0,000
A	16	0,000	0,520	0,000

Tabela 7.10 – Resultado 1ª Comparação – cenário 1

A tabela acima apresenta o resultado obtido pela comparação da solicitação identificada como solicitação A. Para facilitar a leitura e entendimento dos dados, foram selecionados apenas os 20 casos mais relevantes e utilizada a mesma notação da tabela 7.2 quanto à correção dos dados. A tabela completa, com todos os 50 casos processados pode ser consultada no apêndice B deste documento.

Observa-se na tabela que todos os resultados obtidos pelo sistema confirmam os índices esperados que, para esse grupo de solicitações se situava na faixa de baixa ou nenhuma similaridade, com índices variando entre 0,000 e 0,499. Durante a comparação foram utilizados os pesos e limiares determinados na etapa 1 do processamento e, portanto, somente foram comparadas integralmente aquelas solicitações cujo índice de similaridade entre os elementos estruturados situava-se na faixa de 0,600 a 1,000.

É importante notar, também, que os critérios utilizados para selecionar o caso levaram

a obtenção de índices de similaridade bastante elevados na comparação dos elementos estruturados; esse fato se deve a que muitos desses elementos encontravam correspondência direta com alguns casos registrados na base. Já em relação aos elementos não estruturados, todos os resultados apresentaram índices muito baixos, sendo que, em 38 casos comparados não havia nenhuma similaridade entre eles.

Cenário 1 → Caso 2: Solicitação B

Semelhante ao caso anterior, de baixa ou nenhuma similaridade, a expectativa era de que todos os índices fossem de valor menor ou igual a 0,499.

Ident. Solicitação	Caso da Base	IS	ISee	ISne
B	36	0,522	0,715	0,435
B	37	0,476	0,739	0,357
B	15	0,472	0,777	0,334
B	49	0,457	0,708	0,344
B	48	0,450	0,814	0,286
B	45	0,399	0,629	0,296
B	8	0,399	0,604	0,307
B	7	0,388	0,665	0,264
B	20	0,376	0,657	0,250
B	27	0,368	0,643	0,245
B	43	0,364	0,629	0,244
B	44	0,361	0,665	0,225
B	41	0,350	0,771	0,161
B	33	0,350	0,777	0,158
B	39	0,314	0,735	0,124
B	9	0,307	0,639	0,157
B	29	0,300	0,607	0,163
B	38	0,274	0,603	0,126
B	6	0,239	0,771	0,000
B	3	0,220	0,710	0,000

Tabela 7.11 – Resultado da 2ª comparação – cenário 1

Na segunda comparação realizada para teste, houve a ocorrência de uma solicitação, identificada como similar pelo sistema em contraposição à expectativa de que todas as comparações resultassem em índices abaixo de 0,499. Avaliando os resultados, alguns detalhes explicam essa situação.

Em primeiro lugar é preciso considerar a qualidade da especificação contida na solicitação, que apresentava uma descrição bastante pobre das funcionalidades a implementar. Essa situação, já discutida na etapa 1, prejudica a capacidade do modelo na identificação de similaridades pela existência de poucos termos a comparar; quando uma solicitação nessa

situação é comparada com outras que sejam ricas em detalhes, e conseqüentemente com muitos termos comuns ao domínio, é comum que esses termos elevem o índice de similaridade entre os elementos não estruturados e em decorrência dessa elevação, também o índice final seja elevado.

Outra consideração a fazer a respeito dessa comparação é que, apesar de considerada não similar pelos especialistas que a avaliaram, a solicitação trata do fornecimento de informações à orgão externo; ainda que se trate de outras informações, obtidas de outro sistema e destinada à outra entidade, também o caso 36 trata do fornecimento de informações. Existe, portanto, similaridade entre as funcionalidades solicitadas ainda que se estime que de nada serviria ao executante do serviço valer-se de qualquer informação presente naquele caso. Nessa situação, identificando-se a baixa possibilidade de reuso, o próprio demandante poderia optar por não vincular as solicitações.

7.3.2 Cenário 2: Solicitações com média similaridade.

Assim como no cenário anterior, as solicitações escolhidas para a avaliação deste caso precisavam atender a um determinado conjunto de características, que as permitisse classificar como mantendo um grau médio de similaridade, o que, em termos práticos, significa dizer que parte das soluções desenvolvidas poderiam ser reutilizadas, ou que o conhecimento explicitado em algum dos casos poderia ser útil ao atendimento da solicitações presentes. Foram enquadradas nesse caso, solicitações que atendessem aos seguintes critérios

a) Serem direcionadas à mesma gerência escolhida para os casos selecionados para integrar a base de solicitações;

b) Serem direcionadas, prioritariamente, a algum dos sistemas que continham uma base com boa diversidade de casos;

c) Tratarem de assuntos correlatos aos tratados nos casos anteriores.

d) Não compartilharem um conjunto extenso de funcionalidades e termos comuns ao domínio que as pudesse caracterizar como de alta similaridade

Cenário 2 → Caso 3: Solicitação C

A expectativa para esse cenário era pela obtenção de valores entre 0,500 e 0,699 para um grupo de solicitações indicadas pelos especialistas.

Ident. Solicitação	Caso da Base	IS	ISee	ISne
C	6	0,659	0,684	0,648
C	36	0,515	0,809	0,383
C	17	0,481	0,787	0,344
C	16	0,448	0,747	0,314
C	31	0,446	0,856	0,261
C	26	0,431	0,817	0,257
C	35	0,414	0,782	0,248
C	14	0,409	0,814	0,228
C	19	0,377	0,926	0,131
C	12	0,318	0,607	0,189
C	38	0,311	0,605	0,179
C	37	0,211	0,680	0,000
C	34	0,000	0,561	0,000
C	33	0,000	0,484	0,000
C	32	0,000	0,414	0,000
C	50	0,000	0,427	0,000
C	39	0,000	0,561	0,000
C	49	0,000	0,531	0,000
C	48	0,000	0,563	0,000
C	47	0,000	0,491	0,000

Tabela 7.12 – Resultado 1ª Comparação – cenário 2

Nessa comparação, tivemos a correta identificação do único caso apontado como similar na verificação feita pelos especialistas, e a identificação de um caso que não exibia similaridade com a solicitação atual, conforme essa mesma avaliação. Essa incorreção pode ser justificada pelo elevado índice de similaridade existente entre os elementos estruturados de ambas as solicitações. Tratava-se de solicitação de uma funcionalidade bastante diversa daquela constante da solicitação C, porém formulada para o mesmo sistema e com impactos e periodicidades de execução semelhantes. Havia também uma rica descrição das funcionalidades a implementar nessa solicitação, o que, em geral, implica na existência de diversos termos no domínio da aplicação Percebe-se, na tabela, que o índice de similaridade entre os elementos estruturados atingiu a casa dos 80% nesse caso. Quanto à solicitação de número 6, as funcionalidades ali solicitadas são de fato semelhantes à da solicitação C e provavelmente poderiam auxiliar na implementação da nova solução, embora se destinasse a sistema alvo diferente.

Cenário 2 → Caso 4: Solicitação D

Expectativa de índices semelhantes ao caso anterior.

Ident. Solicitação	Caso da Base	IS	ISee	ISne
D	46	0,620	0,788	0,545
D	47	0,527	0,750	0,427
D	43	0,456	0,782	0,310
D	45	0,430	0,821	0,254
D	18	0,428	0,821	0,251
D	48	0,427	0,636	0,333
D	34	0,409	0,680	0,287
D	49	0,384	0,710	0,237
D	39	0,377	0,680	0,242
D	41	0,371	0,680	0,233
D	42	0,342	0,859	0,109
D	7	0,309	0,701	0,132
D	3	0,289	0,602	0,149
D	20	0,279	0,671	0,103
D	50	0,231	0,746	0,000
D	44	0,222	0,715	0,000
D	33	0,187	0,602	0,000
D	40	0,000	0,575	0,000
D	2	0,000	0,569	0,000
D	6	0,000	0,557	0,000

Tabela 7.13 – Resultado 1ª Comparação – cenário 2

Nesse caso, a comparação encontrou apenas dois casos similares na base, confirmando a expectativa de avaliação dos especialistas. Tratava-se de uma solicitação para o desenvolvimento de transação de verificação dos cadastro do cliente que estivesse

participando de leilões de café realizados pelo governo federal. Esse tipo de solicitação não é muito freqüente na base, mas, apesar dessa baixa freqüência o sistema conseguiu identificar corretamente duas solicitações que apresentavam média similaridade com a atual. A primeira fazia referência à implementação de funcionalidade semelhante (validação de informações) sobre outros aspectos que devem ser considerados na condução das atividades do mesmo sistema. A segunda solicitação tratava, também, de aspectos similares e era direcionada ao mesmo sistema, embora as funcionalidades solicitadas fossem diferentes.

7.3.3 Cenário 3: Solicitações com alta similaridade.

Assim como nos cenários anteriores, foram definidos alguns critérios para a avaliação deste caso. Esperava-se que as solicitações recuperadas e apontadas como de alta similaridade pudessem oferecer alta capacidade de reuso das funcionalidades implementadas; Para atender a essa premissa, as solicitações deveriam:

a) ser direcionadas à mesma gerência escolhida para os casos selecionados para integrar a base de solicitações;

b) ser direcionadas, prioritariamente, a algum dos sistemas que continham uma base com boa diversidade de casos;

c) tratar de assuntos correlatos aos tratados nos casos anteriores, e

d) compartilhar funcionalidades e termos comuns ao domínio que as pudesse caracterizar como de alta similaridade.

Cenário 3 → Caso 5: Solicitação E

Para esse cenário, esperava-se que a solicitação encontrasse na base de casos do sistema alguma solicitação para a qual o índice de similaridade deveria situar-se acima de 0,700

Ident. Solicitação	Caso da Base	IS	ISee	ISne
E	5	0,701	0,785	0,664
E	28	0,531	0,665	0,471
E	13	0,513	0,812	0,378
E	1	0,501	0,787	0,373
E	12	0,491	0,602	0,441
E	29	0,477	0,786	0,338
E	49	0,420	0,613	0,333
E	17	0,419	0,676	0,304
E	9	0,389	0,857	0,179
E	27	0,389	0,822	0,194
E	4	0,368	0,752	0,196
E	48	0,347	0,680	0,197
E	46	0,322	0,605	0,195
E	8	0,321	0,611	0,191
E	2	0,319	0,649	0,170
E	22	0,318	0,615	0,185
E	24	0,304	0,613	0,165
E	19	0,302	0,605	0,166
E	31	0,290	0,605	0,149
E	6	0,274	0,602	0,127
E	38	0,241	0,610	0,076
E	21	0,000	0,579	0,000

Tabela 7.14 – Resultado 1ª Comparação – cenário 3

Verifica-se na tabela 7.14 a existência de similaridade entre a solicitação avaliada e

quatro outras solicitações constantes da base de casos. Esse resultado foi considerado excelente pelos especialistas que o avaliaram, visto que a solicitação de número 5 era de fato bastante similar a solicitação E, e requisitava informações semelhantes, obtidas do mesmo sistema porém direcionadas a órgãos diferentes dentro da organização. A reutilização de funcionalidades nesse caso seria praticamente certa e implicaria em uma economia considerável de esforços. Destaca-se no processamento dessa solicitação a identificação de outras duas solicitações (números 28 e 13) que também apresentavam similaridades com a solicitação atual e que poderiam também auxiliar no desenvolvimento da solução. A solicitação de número 1, embora não tenha sido considerada como similar na avaliação dos especialistas apresenta alto grau de similaridade entre os elementos estruturados e descreve funcionalidades também afetas ao sistema tratado. Há de se considerar também que o índice obtido para esse caso foi de apenas 0,501 o que a coloca em uma região extremamente próxima da fronteira definida.

7.4 Avaliação de performance

Embora o objetivo deste trabalho seja a obtenção de um modelo aplicável ao processo de *software* e a ferramenta desenvolvida o tenha sido apenas para demonstrar sua correção, durante os teste foram feitas também medições do tempo gasto para fazer a comparação dos elementos, e, nesse ponto os resultados obtidos apontam para a necessidade de se promoverem melhoras na performance geral do aplicativo. Apenas para efeito de registro, no melhor caso as 2500 comparações efetuadas consumiram 46 segundos para serem concluídas e 93 segundos no pior caso, dependendo do equipamento utilizado. Esse tempo foi consumido quase que exclusivamente na comparação dos termos descritos em linguagem natural, resultado relativamente previsível pois o processamento dos termos de cada solicitação era refeito a cada comparação, já que, por uma opção de projeto que visava sua simplificação, tal resultado não estava sendo guardado na base. Esse resultado aponta fortemente para a necessidade de se adotar tal prática antes de expandir o uso da aplicação sobre uma base maior e fica como indicação para os trabalhos futuros.

8 Conclusão e trabalhos futuros

O presente trabalho envolveu a identificação e caracterização de um problema com o qual profissionais e organizações da área de informática freqüentemente se deparam, que é a falta da gestão do conhecimento ligado ao processo de desenvolvimento de *software*. Para sua solução, foi proposto um modelo com características especialmente planejadas para permitir a identificação e o reuso de soluções similares que já tenham sido desenvolvidas.

Os resultados obtidos pela aplicação do modelo proposto demonstram com clareza o potencial existente para sua utilização, e são bastante promissores em relação aos benefícios que poderá trazer ao processo de desenvolvimento de *software*.

Ainda que aplicado sobre uma base relativamente pequena de casos, foi possível identificar similaridades que, a princípio, não seriam percebidas a não ser que a equipe responsável pelo desenvolvimento da nova solução tivesse, também, participado da solução anterior. Como o índice de rotatividade das equipes é relativamente alto, um interstício de um ou dois anos pode significar que toda a equipe já tenha sido rodiziada de funções, até porque a crescente demanda pelo desenvolvimento de produtos de *software* obriga os administradores da área a utilizarem eficientemente os recursos humanos disponíveis, viabilizando o atendimento das demandas prioritárias das organizações.

Há que se considerar, também, a crescente opção das empresas pela adoção de práticas de gerenciamento de projetos na condução das equipes de desenvolvimento de *software*, com a adoção de equipes matriciais de especialistas em cada uma das atividades (levantamento de requisitos, modelagem de dados, codificação, etc) em detrimento à manutenção de equipes fixas focadas no atendimento de um dado produto. Essa situação, ao mesmo tempo em que permite explorar de maneira eficiente o potencial de trabalho de cada profissional do grupo, direcionando-o à um conjunto de atividades nas quais ele tenha desenvolvido algum tipo de especialização, seja através de formação teórica ou prática, exige que se adotem mecanismos eficientes para o gerenciamento do conhecimento envolvido no processo sob pena de inviabilizar manutenções futuras ou até mesmo a continuidade do negócio, sob certas condições.

A utilização de um modelo híbrido nos permitiu trabalhar com um conjunto de elementos estruturados e não estruturados. Essa abordagem se revelou como um importante diferencial, uma vez que a situação vivenciada pela maioria das organizações ocorre dessa forma. São raros os casos em que as informações dispersas em sistemas de controle e

acompanhamento, geralmente utilizados, constituem-se somente de um ou de outro tipo. A possibilidade de trabalhar a combinação desses dois tipos de informação conjuntamente, aplicando pesos específicos a cada grupo de informação e a cada um dos elemento de forma mais específica foi avaliada como uma característica positiva do modelo.

Também em função dessa característica avalia-se que o universo de aplicações nas quais o modelo pode ser utilizado com sucesso seja muito maior que aquele tratado no estudo de caso. Observa-se esse tipo de informação, composta por dados estruturados e não estruturados, presente em vários ramos da sociedade. Apenas como exemplo, podemos citar duas outras situações em que essa situação se faz presente:

a) Prontuários médicos: Ainda que essa área já tenha sido objeto de avaliação por diversos trabalhos de pesquisa, é fácil notar que o prontuário de um paciente apresenta claramente uma estruturação semelhante àquela tratada no estudo de caso, com informações estruturadas (sexo, idade, estado civil, por exemplo) e outras não estruturadas, tais como a anamnese do paciente e as prescrições e indicações médicas. Aqui se apresenta, também, uma oportunidade de gerir o conhecimento presente nesses documentos para a obtenção de informações úteis para o tratamento de novos casos.

b) Processos judiciais: Muitas pesquisas tem sido feitas, também, nessa área, até porque a quantidade de informações produzidas no curso de um processo é, em geral, muito grande. Ocorre, no entanto, que a quantidade de blocos de texto a processar em busca de similaridades torna extremamente difícil essa tarefa. A adoção de um sistema híbrido que fizesse uma triagem inicial sobre a relevância de cada documento, baseado em parâmetros estruturados poderia simplificar essa tarefa e reduzir o gasto de tempo e o consumo de recursos na obtenção de informações relevantes.

Os exemplos acima dão apenas uma idéia do potencial de aplicação da metodologia aqui proposta. É certo que muitas outras áreas poderiam valer-se também de tal proposição, assim como é certo também que a adoção de um modelo automatizado para o processamento de informações em busca de conhecimento útil não é livre de falhas e não se pode garantir que sempre levará a resultados corretos. A incerteza é uma das poucas certezas que se pode ter quando se lida com mineração de dados em busca de conhecimento.

Ainda em relação ao modelo proposto é preciso destacar que se trata de uma primeira avaliação; novos experimentos e pequisas deverão ser feitos em prosseguimento ao presente trabalho como forma de confirmar e melhorar os resultados aqui apresentados, notadamente pela avaliação de formalismos alternativos para a identificação de similaridades e para a construção de índices cada vez mais representativos da proximidade dos casos avaliados.

Trabalhos futuros

Durante a preparação e execução dos testes com o modelo e com a aplicação, algumas situações que surgiram indicavam a possibilidade de outras formas de solução. Essas situações estão descritas a seguir e ficam como meta para a continuidade dos trabalhos

- **Calibragem dos pesos relativos a cada atributo**: Percebeu-se, durante a fase de calibragem, que o ajuste manual de pesos é uma atividade delicada, sensível e em grande dose, empírica. Conhecendo-se os valores que se aplicam a uma dada base de casos, o trabalho poderia ser facilitado pela utilização de uma rede neural, como topologia adequada, e treinamento sobre esses casos. Supõe-se que tal procedimento poderia levar a um ajuste melhor dos pesos e com menor consumo de tempo para sua obtenção.
- **Tratamento dos documentos utilizados no levantamento de requisitos:** Durante a avaliação do problema, foram identificadas as etapas iniciais do processo de desenvolvimento de *software* como aquelas em que a explicitação do conhecimento ocorre de forma mais intensa. Entre essas, a fase de levantamento de requisitos é também extremamente rica em conhecimentos que se devem buscar aproveitar em etapas seguintes e não foi tratada no escopo de nossa avaliação. O modelo proposto, no entanto, se aplica perfeitamente a essa fase, restando fazer o mapeamento dos elementos que devem tratados para que se possa utilizá-lo a partir dos documentos de requisitos do sistema, elaborados no início dos trabalhos de desenvolvimento de novas soluções.
- **Comparação com outras metodologias:** Durante a pesquisa foram identificadas várias linhas na área de Inteligência Artificial que poderiam ser utilizadas para suporte ao modelo. Essas linhas foram abordadas nos capítulos 2 e 3 e, apesar do direcionamento da pesquisa apontar sua aderência aos formalismos utilizados, não se pode afirmar que não seriam obtidos resultados melhores com a adoção de outras técnicas. O processamento dos elementos não estruturados, em especial, é uma área onde os resultados obtidos pela comunidade acadêmica ainda não apontam claramente para a adoção preferencial de uma ou outra técnica, e sim para caraterísticas que tornam adequada a utilização de uma dada metodologia em um determinado contexto. Entende-se que uma avaliação dessa natureza requer a implementação das demais técnicas de avaliação de similaridades entre elementos de texto, o que estava

claramente fora dos objetivos deste trabalho, mas pode apresentar uma contribuição significativa para a continuidade da pesquisa.

- **Revisão dos atributos considerados:** Ainda na fase de calibragem pode-se notar que determinados atributos, tais como benefícios esperados e riscos envolvidos apresentavam uma contribuição muito pequena na apuração do índice de similaridade, quando comparados aos demais atributos. A supressão desses atributos, para efeito de comparação e obtenção do índice de similaridade entre os elementos estruturados não causaria variações significativas, conforme observação do estudo de caso.
- **Performance geral do aplicativo de suporte ao modelo:** Ainda que tal preocupação não fizesse parte dos objetivos iniciais da pesquisa, percebe-se que existe a necessidade de melhorar a performance do aplicativo para permitir sua aplicação. Notou-se, ainda, que essa melhoria pode ser conseguida com alterações relativamente simples na estrutura da aplicação, tais como salvar o resultado do processamento dos termos em linguagem natural e indexar termos para pesquisa..

Referência bibliográfica

AAMODT, Agnar; PLAZA, Enric. **Case-Based Reasoning:** Foundational Issues, Methodological Variations, and System Approaches. In: AI Communications, Vol 7, nr. 1, 1994.

BB. **Conheça o BB em grandes números**. Disponível em: http://www.bb.com.br. Acesso em 21 de agosto de 2008.

BERRY, Michael W. **Survey of text mining:**clustering, classification and retrieval. 1st edition, New York, USA. Springer, 2004.

BRÜNINGHAUS, Stefanie; ASHLEY, Kevin D. **The role of information extraction for textual CBR.** Proceedings of the 4th International Conference on Case-Based Reasoning. Vancouver, BC, Canada , 2001.

CARVALHO, André P. L. F. **Redes neurais Artificiais.** Disponível em: http://www.icmc.usp.br/~andre/research/neural/index.htm. Acesso em 22 de junho de 2008.

CRISP-DM. Cross Industry Standard Process for Data Mining. Disponível em: http://www.crisp-dm.org. Acesso em 22 de junho de 2008.

ESPÍNDOLA, R.; et.al. **Uma abordagem baseada em gestão do conhecimento para gerência de requisitos em desenvolvimento distribuído de software**. Rio Grande do Sul, Brasil. PUC/RS - Programa de pós-graduação em ciência da computação.[]

FELDMAN, Ronen et al. **Text mining at term level.** In: Proceedings of the 2nd European Symposium on principles of Data Mining. Nantes, France, 1998.

GANE, C.; SARSON, T. **Análise estruturada de sistemas**. 8ª edição. Rio de Janeiro, Brasil. Editora LTC, 1983.

HAN, Jiawei; KAMBER Micheline. **Data Mining:** concepts and Techiniques. 1st edition. San Francisco, USA. Morgan Kaufmann, 2001.

HAYKIN, Simon. **Redes Neurais:** princípios e prática. 2ª edição. Porto Alegre, Brasil. Bookman, 2001.

HYYRÖ, Heikki. **A bit-vector algorithm for computing Levenshtein and Damerau edit distances.** In: Nordic Journal of Computing, Vol. 10 , 2003.

JOACHIMS, Thorsten. **A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for text categorization.** Proceedings of the Fourteenth International Conference on Machine Learning (ICML´97), 1997.

JURAFSKY, Daniel; MARTIN, James H. **Speech and Language Processing:** an introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. 2ª edição. Upper Saddle River, New Jersey, USA. Prentice Hall, 2008.

KOLODNER, Janet L. **Case-Based Learning.** 1st edition, Los Altos, USA. Morgan Kaufmann, 1993.

KUNZE, Mirjam; HÜBNER, André. **CBR on Semi-structured Documents: The experiencebook and the FAllQ Project.** Proceedings 6th German Workshop on CBR, 1998.

LAROSE, D.T. **Discovering knowledge in data**: an introduction to data mining. Hoboken, New Jersey, USA. Wiley Interscience, 2005.

LEAKE, David B. **Case-Based Reasoning:** Experiences, Lessons, and Future Directions. 1st edition, The MIT Press, 1996.

LENZ, Mario. **Textual CBR and Information Retrieval – A comparison.** Proceedings of 6th German Workshop on CBR, 1998

LIEBOWITZ, J.; WILCOX, L.C. **Knowledge Management and its Integrative Elements**. Boca Raton. CRC Press, 1997.

LUGER, G. F. **Inteligência Artificial**: Estruturas e estratégias para a solução de problemas complexos. 3ª edição. Porto Alegre, Brasil. Bookmann, 2004.

MARTIN, Robert C. **Agile software development:** principles, patterns and practices. 1st edition, Upper Saddle River , New Jersey, USA. Prentice Hall, 2003.

MURPHYK, Kevin P. **A Brief Introduction to Graphical Models and Bayesian Networks.** Disponível em:http://www.cs.berkeley.edu/~murphyk/Bayes/bayes.html. Acesso em 22 de junho de 2008

NONAKA, Ikujiro; TAKEUCHI, Hirotaka. **Criação do Conhecimento na Empresa.** 9ª edição. Rio de Janeiro, Brasil. Editora Campus, 1997

PARREIRAS, F. S.; BAX, M. P. A gestão de conteúdos no apoio à engenharia de software. KMBrasil, 2003. São Paulo, 2003.

PARREIRAS, F. S.; OLIVEIRA, G. S. **Análise comparativa de processos de desenvolvimento de software sob a luz da gestão do conhecimento**: um estudo de caso de empresas mineiras. Simpósio Brasileiro de qualidade de software. Brasília, Brasil, 2004.

PRESSMAN, Roger S. **Engenharia de Software.** 6ª edição. São Paulo, Brasil. McGraw-Hill do Brasil, 2006.

RECIO, Juan A. et al. **Extending jCOLIBRI for Textual CBR.** Proceedings of the 6th International Conference on Case-Based Reasoning, 2005

RUSSEL, Stuart J; NORVIG, Peter. **Inteligência Artificial:** uma abordagem moderna. 2ª edição. Rio de Janeiro, Brasil. Editora Campus, 2004.

SALTON, Gerard; McGILL Michael. **Introduction to Modern Information Retrieval.**1st edition, New York, USA, McGraw-Hill, 1984.

SALTON, Gerard; BUCKLEY, Christopher. **Term-weighting aproaches in automatic text retrieval.** In:Information Processing and Management Journal, Vol 24, número 5, 1998

SHIRABAD, J. S.; LETHBRIDGE, T. C.; MATWIN S. **Supporting Maintenance of Legacy Software with Data Mining Techniques**. Proceedings of the 2000 conference of the Centre for Advanced Studies on Collaborative research. Mississauga, Ontario, Canadá, 2001.

SILVA, Renato A. C. **Inteligência artificial aplicada a ambientes de engenharia de software: uma visão geral**. In: INFOCOMP Journal of Computer Science, vol. 4, 2005.

SOMERVILLE, Ian. **Engenharia de software**. 2ª edição. São Paulo, Brasil. Editora Campus, 1998.

WERNER, C. M. L.; et.al. **OdysseyShare**: Desenvolvimento colaborativo de componentes. Rio de Janeiro, Brasil. COPPE/UFRJ - Programa de engenharia de sistemas e computação, 2003.

WERNER, C. M. L.; MURTA, L. G. P. **Uma máquina de processos de desenvolvimento de software baseada em agentes inteligentes**. Rio de Janeiro, Brasil. COPPE/UFRJ - Programa de engenharia de sistemas e computação. 2002.

WTO. **Bases de dados da Associação Mundial de Comércio (WTO)**. Disponível em: http://www.wto.org/english/res_e/statis_e/statis_e.htm. Acesso em 22 de junho de 2008.

ZIMMERMANN, T.; et. al**. Mining Version Histories to Guide Software Changes.** Proceedings of the 26th International Conference on Software Engineering ICSE '04. Scotland, UK, 2004.

Glossário

Para melhor compreensão deste trabalho, relacionam-se abaixo alguns termos e os respectivos significados que lhes é atribuído neste texto.

- ***Software*** – Conjunto de artefatos associados e necessários à perfeita execução de um programa de computador, incluindo o código fonte, os módulos executáveis e a documentação contendo a descrição de todas as etapas envolvidas em sua execução.
- **Interface** – Parte do sistema responsável pela interação com outros sistemas, outros módulos do mesmo sistema ou com o usuário do sistema.
- **Cliente** – Solicitante do desenvolvimento de um determinado *softwar*e para suportar um produto ou serviço.
- **Desenvolvedor** – Responsável pelo desenvolvimento do software para atendimento de uma solicitação do cliente.
- **Desenvolvimento de *software*** – Conjunto de etapas envolvidas no processo de criação de um novo *software* ou de alguma nova funcionalidade em um software existente.
- **Manutenção de *software*** – Intervenções destinadas a corrigir alguma funcionalidade existente em um *software*.
- **Levantamento de requisitos** – Etapa do processo de desenvolvimento de *software* que consiste na determinação das funcionalidades que devem ser suportadas pelo *software* a ser construído.
- ***Mainframe*** – Computador de grande capacidade de processamento e armazenamento, utilizado em situações nas quais é requerida a capacidade de processar simultaneamente grandes volumes de informação.
- **Negócio** – Cada um dos produtos e serviços relacionados com a atividade fim da empresa.
- **Aplicativo** – *Software* construído com o propósito especifico de auxiliar o funcionamento ou o controle de um determinado negócio da empresa.
- **Sistemas corporativos** – Sistemas de propósito geral, destinados a prover as condições necessárias à execução dos demais aplicativos (p. ex. cadastro de clientes).
- **Rotinas *batch*** - Conjunto de procedimentos executados de maneira seqüencial a um grupo de registros do mesmo tipo.
- **Rotinas *on-line*** – Procedimentos executados em tempo de iteração com o usuário.

Apêndices

Apêndice A: Exemplo de tratamento de solicitação

Exemplo de tratamento de uma solicitação (elementos estruturados e não estruturados) pelo modelo proposto.

Sejam D1 → nova solicitação, e

C3 → uma solicitação qualquer registrada na base de casos.

D1: nova solicitação		Resultado da comparação do atributo de D1 com Cn			C3: um dos casos existentes na base (Cn, para n=3)	
Atributo	**Valor (*)**	**Atributo igual**	**Peso**	**Result.**	**Atributo**	**Valor**
Tipo_solicitação	1	1	0,1	0,1	Tipo_solicitação	1
Prod_relacion	22 =(2+4+16)	2/5 (*)	0,9	0,36	Prod_relacion	6 =(2+4)
Canal_dist	28 =(4+8+16)	2/5 (*)	0,8	0,32	Canal_dist	20 =(4+16)
Period_exec	1041 =(1+16+1024)	2/4 (**)	0,5	0,25	Period_exec	17 = (1+16)
Benef_esprd	81 =(1+16+64)	1/3 (***)	0,7	0,21	Benef_esprd	146 =(2+16+128)
Risco_envolvd	45	3/3	0,4	0,12	Risco_envolvd	45
Legis_tipo	4	1	0,2	0,2	Legis_envolvd	4
Legis_num	11322	1	0,8	0,8	Legis_num	11322
Legis_data	20/11/1995	1	0,6	0,6	Legis_envolvd	20/11/1995
Resultado final: ISee_D1_C3 ➔➔ **(dos elementos estruturados)**				**2,86**	←←←←←	

(*) Considera que 2 dos 5 aspectos possíveis são iguais na solicitação D1 e no caso C3.

(**) Considera que 2 dos quatro aspectos possíveis para o atributo canal_dist são iguais.

(***) Considera que 1 dos 3 aspectos possíveis para o atributo benefício são iguais.

Após o cálculo, normalizar utilizando o somatório de todos os pesos = 5,00

➔ Isee_D1_C3* = 2,86/5,00 = 0,572

Como Isee_D1_C3* é maior que o limiar estabelecido na tabela de configuração do sistema (Limiar_Isee = 0,55), o caso C3 será filtrado para a comparação completa (envolvendo os elementos não estruturados)

Supondo que apenas o caso C3 na base tenha superado o limiar estabelecido no arquivo de configuração, comparar os elementos não estruturados desse caso com os da nova solicitação (documento D1, nesse exemplo).

Se houver mais casos, deve-se comparar todos eles e apresentar ao usuário uma lista dos casos selecionados, ordenados em ordem decrescente do Indice final de similaridade.

D1: nova solicitação		**Resultado da comparação do atributo de D1 com Cn**			**C3: um dos casos existentes na base (Cn, para n=3)**	
Atributo	**Valor**	**Result. do parser**	**Peso**	**Result.**	**Atributo**	**Valor**
Objetivo	Bla, bla, bla ....bla, bla	0,30 (*)	0,8	0,24	Objetivo	Bla, blum, zip, ploc
Funcionalidades	Zip, Zap, Zum ...Ploc, Plat, Zem	0,43 (*)	1	0,43	Funcionalidades	Zap, Zep, Zip, Bla, Bla ... Bla, Blum
Resultado final: ISne_D1_C3 ➔➔ **(dos elementos NÃO estruturados)**				**0,67**	⬅⬅⬅⬅⬅	

(*) O resultado do parser é obtido pela comparação dos elementos não estruturados,

Após o cálculo, normalizar utilizando o somatório de todos os pesos = 1,8

➔ **ISne_D1_C3* = 0,67/1,8 = 0,372**

➔ **IS_final_D1_C3 = ISee_D1_C3* x Pee + ISne_D1_C3* x Pne**

➔ **IS_final_D1_C3 = 0,572 x 0,50 + 0,372 x 0,80**

➔ **IS_final_D1_C3 = 0,286 + 0,298**

➔ **IS_final_D1_C3 = 0,584**

Após o cálculo, normalizar utilizando a soma dos pesos = (1,3)

➔ **IS_final_D1_C3* = 0,449**

Apêndice B: Tabelas completas de comparação dos cenários

Ident. Solicitação	Caso da Base	IS	Isee	Isne
A	1	0,000	0,424	0,000
A	2	0,000	0,566	0,000
A	3	0,000	0,494	0,000
A	4	0,000	0,529	0,000
A	5	0,000	0,528	0,000
A	6	0,000	0,526	0,000
A	7	0,000	0,487	0,000
A	8	0,000	0,459	0,000
A	9	0,000	0,494	0,000
A	10	0,000	0,576	0,000
A	11	0,000	0,576	0,000
A	12	0,000	0,449	0,000
A	13	0,000	0,520	0,000
A	14	0,000	0,481	0,000
A	15	0,000	0,481	0,000
A	16	0,000	0,520	0,000
A	17	0,000	0,524	0,000
A	18	0,329	0,683	0,170
A	19	0,000	0,558	0,000
A	20	0,209	0,675	0,000
A	21	0,000	0,567	0,000
A	22	0,348	0,603	0,234
A	23	0,000	0,524	0,000
A	24	0,000	0,566	0,000
A	25	0,000	0,568	0,000
A	26	0,000	0,414	0,000
A	27	0,000	0,459	0,000
A	28	0,000	0,571	0,000
A	29	0,000	0,598	0,000
A	30	0,221	0,712	0,000
A	31	0,195	0,628	0,000
A	32	0,221	0,712	0,000
A	33	0,262	0,607	0,107
A	34	0,331	0,684	0,172
A	35	0,000	0,555	0,000
A	36	0,000	0,528	0,000
A	37	0,000	0,523	0,000
A	38	0,309	0,605	0,175
A	39	0,401	0,684	0,274
A	40	0,199	0,641	0,000
A	41	0,418	0,719	0,283
A	42	0,189	0,610	0,000
A	43	0,286	0,680	0,109
A	44	0,392	0,648	0,277
A	45	0,374	0,718	0,219
A	46	0,341	0,751	0,157
A	47	0,310	0,648	0,159
A	48	0,000	0,563	0,000
A	49	0,000	0,461	0,000
A	50	0,000	0,567	0,000

Tabela B.1- Resultado completo para a comparação do caso A - cenário 1

Ident. Solicitação	Caso da Base	IS	ISee	ISne
B	1	0,000	0,573	0,000
B	2	0,000	0,498	0,000
B	3	0,220	0,710	0,000
B	4	0,000	0,573	0,000
B	5	0,000	0,533	0,000
B	6	0,239	0,771	0,000
B	7	0,388	0,665	0,264
B	8	0,399	0,604	0,307
B	9	0,307	0,639	0,157
B	10	0,000	0,525	0,000
B	11	0,000	0,455	0,000
B	12	0,000	0,595	0,000
B	13	0,000	0,560	0,000
B	14	0,000	0,566	0,000
B	15	0,472	0,777	0,334
B	16	0,000	0,489	0,000
B	17	0,000	0,561	0,000
B	18	0,195	0,629	0,000
B	19	0,000	0,489	0,000
B	20	0,376	0,657	0,250
B	21	0,000	0,463	0,000
B	22	0,000	0,569	0,000
B	23	0,000	0,455	0,000
B	24	0,000	0,533	0,000
B	25	0,000	0,464	0,000
B	26	0,000	0,524	0,000
B	27	0,368	0,643	0,245
B	28	0,000	0,566	0,000
B	29	0,300	0,607	0,163
B	30	0,000	0,383	0,000
B	31	0,000	0,489	0,000
B	32	0,000	0,419	0,000
B	33	0,350	0,777	0,158
B	34	0,217	0,700	0,000
B	35	0,000	0,560	0,000
B	36	**0,522**	0,715	0,435
B	37	0,476	0,739	0,357
B	38	0,274	0,603	0,126
B	39	0,314	0,735	0,124
B	40	0,000	0,348	0,000
B	41	0,350	0,771	0,161
B	42	0,000	0,594	0,000
B	43	0,364	0,629	0,244
B	44	0,361	0,665	0,225
B	45	0,399	0,629	0,296
B	46	0,217	0,700	0,000
B	47	0,217	0,700	0,000
B	48	0,450	0,814	0,286
B	49	0,457	0,708	0,344
B	50	0,209	0,674	0,000

Tabela B.2- Resultado completo para a comparação do caso B - cenário 1

Ident. Solicitação	Caso da Base	IS	ISee	ISne
C	1	0,000	0,494	0,000
C	2	0,000	0,566	0,000
C	3	0,000	0,423	0,000
C	4	0,000	0,564	0,000
C	5	0,000	0,457	0,000
C	6	0,659	0,684	0,648
C	7	0,000	0,539	0,000
C	8	0,000	0,564	0,000
C	9	0,000	0,529	0,000
C	10	0,000	0,454	0,000
C	11	0,000	0,489	0,000
C	12	0,318	0,607	0,189
C	13	0,000	0,571	0,000
C	14	0,409	0,814	0,228
C	15	0,000	0,498	0,000
C	16	0,448	0,747	0,314
C	17	0,481	0,787	0,344
C	18	0,000	0,526	0,000
C	19	0,377	0,926	0,131
C	20	0,000	0,570	0,000
C	21	0,000	0,462	0,000
C	22	0,000	0,568	0,000
C	23	0,000	0,454	0,000
C	24	0,000	0,496	0,000
C	25	0,000	0,532	0,000
C	26	0,431	0,817	0,257
C	27	0,000	0,529	0,000
C	28	0,000	0,520	0,000
C	29	0,000	0,598	0,000
C	30	0,000	0,378	0,000
C	31	0,446	0,856	0,261
C	32	0,000	0,414	0,000
C	33	0,000	0,484	0,000
C	34	0,000	0,561	0,000
C	35	0,414	0,782	0,248
C	36	0,515	0,809	0,383
C	37	0,211	0,680	0,000
C	38	0,311	0,605	0,179
C	39	0,000	0,561	0,000
C	40	0,000	0,413	0,000
C	41	0,000	0,561	0,000
C	42	0,000	0,487	0,000
C	43	0,000	0,452	0,000
C	44	0,000	0,526	0,000
C	45	0,000	0,491	0,000
C	46	0,000	0,523	0,000
C	47	0,000	0,491	0,000
C	48	0,000	0,563	0,000
C	49	0,000	0,531	0,000
C	50	0,000	0,427	0,000

Tabela B.3- Resultado completo para a comparação do caso C- cenário 2

Ident. Solicitação	Caso da Base	IS	ISee	ISne
D	1	0,000	0,461	0,000
D	2	0,000	0,569	0,000
D	3	0,289	0,602	0,149
D	4	0,000	0,391	0,000
D	5	0,000	0,496	0,000
D	6	0,000	0,557	0,000
D	7	0,309	0,701	0,132
D	8	0,000	0,426	0,000
D	9	0,000	0,391	0,000
D	10	0,000	0,544	0,000
D	11	0,000	0,509	0,000
D	12	0,000	0,382	0,000
D	13	0,000	0,346	0,000
D	14	0,000	0,340	0,000
D	15	0,000	0,515	0,000
D	16	0,000	0,305	0,000
D	17	0,000	0,421	0,000
D	18	0,428	0,821	0,251
D	19	0,000	0,490	0,000
D	20	0,279	0,671	0,103
D	21	0,000	0,500	0,000
D	22	0,000	0,500	0,000
D	23	0,000	0,457	0,000
D	24	0,000	0,499	0,000
D	25	0,000	0,465	0,000
D	26	0,000	0,382	0,000
D	27	0,000	0,391	0,000
D	28	0,000	0,427	0,000
D	29	0,000	0,460	0,000
D	30	0,000	0,539	0,000
D	31	0,000	0,490	0,000
D	32	0,000	0,539	0,000
D	33	0,187	0,602	0,000
D	34	0,409	0,680	0,287
D	35	0,000	0,305	0,000
D	36	0,000	0,313	0,000
D	37	0,000	0,554	0,000
D	38	0,000	0,460	0,000
D	39	0,377	0,680	0,242
D	40	0,000	0,575	0,000
D	41	0,371	0,680	0,233
D	42	0,342	0,859	0,109
D	43	0,456	0,782	0,310
D	44	0,222	0,715	0,000
D	45	0,430	0,821	0,254
D	46	**0,620**	0,788	0,545
D	47	**0,527**	0,750	0,427
D	48	0,427	0,636	0,333
D	49	0,384	0,710	0,237
D	50	0,231	0,746	0,000

Tabela B.4- Resultado completo para a comparação do caso D - cenário 2

Ident. Solicitação	Caso da Base	IS	ISee	ISne
E	1	0,501	0,787	0,373
E	2	0,319	0,649	0,170
E	3	0,000	0,541	0,000
E	4	0,368	0,752	0,196
E	5	0,701	0,785	0,664
E	6	0,274	0,602	0,127
E	7	0,000	0,534	0,000
E	8	0,321	0,611	0,191
E	9	0,389	0,857	0,179
E	10	0,000	0,571	0,000
E	11	0,000	0,500	0,000
E	12	0,491	0,602	0,441
E	13	0,513	0,812	0,378
E	14	0,000	0,419	0,000
E	15	0,000	0,454	0,000
E	16	0,000	0,454	0,000
E	17	0,419	0,676	0,304
E	18	0,000	0,496	0,000
E	19	0,302	0,605	0,166
E	20	0,000	0,523	0,000
E	21	0,000	0,579	0,000
E	22	0,318	0,615	0,185
E	23	0,000	0,571	0,000
E	24	0,304	0,613	0,165
E	25	0,000	0,544	0,000
E	26	0,000	0,566	0,000
E	27	0,389	0,822	0,194
E	28	0,531	0,665	0,471
E	29	0,477	0,786	0,338
E	30	0,000	0,425	0,000
E	31	0,290	0,605	0,149
E	32	0,000	0,461	0,000
E	33	0,000	0,454	0,000
E	34	0,000	0,531	0,000
E	35	0,000	0,560	0,000
E	36	0,000	0,568	0,000
E	37	0,000	0,458	0,000
E	38	0,241	0,610	0,076
E	39	0,000	0,531	0,000
E	40	0,000	0,390	0,000
E	41	0,000	0,566	0,000
E	42	0,000	0,499	0,000
E	43	0,000	0,457	0,000
E	44	0,000	0,496	0,000
E	45	0,000	0,496	0,000
E	46	0,322	0,605	0,195
E	47	0,000	0,496	0,000
E	48	0,347	0,680	0,197
E	49	0,420	0,613	0,333
E	50	0,000	0,579	0,000

Tabela B.5- Resultado completo para a comparação do caso E - cenário 3