

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

SISTEMA DE GESTÃO DE INCIDENTES COM RACIOCÍNIO
BASEADO EM CASOS UTILIZANDO ONTOLOGIA E
ANÁLISE DE TEXTO

JEWITON LÉO GARCIA JÚNIOR

BLUMENAU
2015

2015/2-12

JEWITON LÉO GARCIA JÚNIOR

**SISTEMA DE GESTÃO DE INCIDENTES COM RACIOCÍNIO
BASEADO EM CASOS UTILIZANDO ONTOLOGIA E
ANÁLISE DE TEXTO**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Airan Arinê Possamai, Mestre - Orientador

**BLUMENAU
2015**

2015/2-12

**SISTEMA DE GESTÃO DE INCIDENTES COM RACIOCÍNIO
BASEADO EM CASOS UTILIZANDO ONTOLOGIA E
ANÁLISE DE TEXTO**

Por

JEWITON LÉO GARCIA JÚNIOR

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: _____
Prof. Airan Arinê Possamai, Mestre – Orientador, FURB

Membro: _____
Prof. Samuel Cristhian Schwebel, Mestre – FURB

Membro: _____
Prof. Roberto Heinzle, Doutor – FURB

Blumenau, 08 de dezembro de 2015

Dedico este trabalho aos meus amigos, professores, minha família e especialmente aos meus pais que sempre me apoiaram e acreditaram em mim.

AGRADECIMENTOS

A Deus pelo dom de existir e pela saúde.

À minha família por sempre me incentivar e acreditar em meus projetos, em especial aos meus pais, por contribuírem com diversas ideias e sugestões neste trabalho e durante toda a minha graduação, carreira profissional e vida, a minha irmã por me auxiliar na correção ortográfica de vários textos meus ao longo desta graduação e traduções, ao meu irmão caçula por sempre me incentivar a imaginar formas diferentes de pensar e ver as coisas, e por fim ao meu irmão do meio e meus avós.

Aos meus amigos por estarem presentes ao longo de minha vida, trocando ideias, conversando e contribuindo com suas experiências de vida que me auxiliam a continuar evoluindo.

A todos os professores ao longo da minha graduação que contribuíram com a minha formação, em especial os professores Aurélio Faustino Hoppe, Alexander Roberto Valdameri, Dalton Solano dos Reis, Fabio Rafael Segundo, Gilvan Justino e Joyce Martins.

Ao meu orientador, professor Airan Arinê Possamai por acreditar em mim e me auxiliar no desenvolvimento deste trabalho. E por fim aos professores Daniel Theisges dos Santos e Everaldo Artur Grahl que me auxiliaram com diversas sugestões e ideias no desenvolvimento deste trabalho.

Eu acredito na intuição e na inspiração. A imaginação é mais importante que o conhecimento. O conhecimento é limitado, enquanto a imaginação abraça o mundo inteiro, estimulando o progresso, dando à luz à evolução. Ela é, rigorosamente falando, um fator real na pesquisa científica.

Albert Einstein

RESUMO

Este trabalho tem como objetivo apresentar uma solução computacional para auxiliar a gestão de incidentes para a área de suporte e manutenção de computadores, seguindo o modelo de referência MPS-SV. Especificando e aplicando um modelo de busca por casos de incidentes semelhantes que utiliza técnicas de raciocínio baseado em casos com a biblioteca MyCBR, ontologia fazendo uso da biblioteca Apache Jena e análise de texto utilizando a biblioteca Apache Lucene. Os resultados do desenvolvimento do sistema de gestão de incidentes foram a especificação e aplicação de um modelo de busca por incidentes semelhantes que envolvem um modelo de categorias de incidente representado em uma ontologia, análise de texto levando em consideração a definição do técnico, a descrição do cliente e um modelo de incidente RBC que representa os atributos que interessam comercialmente a empresa na aplicação da solução. Após o desenvolvimento do sistema de gestão de incidentes pode-se concluir que o resultado obtido é interessante, por se fundamentar em conceitos e processos de gestão de incidentes abrangentes e versáteis como o MPS-SV, por ser uma solução que utiliza bibliotecas de código aberto gratuitas para implementar um sistema de gestão de incidentes, que por sua vez possui uma funcionalidade específica para busca por incidentes semelhantes levando em consideração tanto a similaridade entre eles, quanto qual destes incidentes semelhantes pode ser o mais adequado para a situação específica.

Palavras-chave: Análise de texto. Ontologia. Raciocínio baseado em casos.

ABSTRACT

This work aims to present a computational solution to assist the incident management for support and maintenance of computers, based on MPS-SV reference model . Specifying and applying a search engine for similar incidents cases which utilizes case-based reasoning with the MyCBR library, ontology utilizing the Apache Jena library and text analysis using Apache Lucene library. The results of the incident management software development were the specification and application of a search engine for similar incidents that involves categories models of incident represented by an ontology, text analysis considering the technician's definition, the client's description and a CBR incident model which represents the attributes that commercially interest the company in the application solution. After the development of an incident management software it can be concluded that the obtained result is interesting, by being based in comprehensive and versatile concepts and processes of incident management like the MPS-SV, by being a solution that utilizes open source libraries to implement an incident management software, which has an specific functionality to look for similarities between themselves and to see what incident solution is the most suitable for a specific situation.

Key-words: Text analysis. Ontology. Case-based reasoning.

LISTA DE FIGURAS

Figura 1 - Componentes do modelo MPS	17
Figura 2 - Composição de um serviço	22
Figura 3 - Fluxograma do gerenciamento de incidentes.....	23
Figura 4 - Tela de ocorrências do Ocomon	33
Figura 5 - Interface de busca de solução Silva (2002)	34
Figura 6 - Diagrama de caso de uso	37
Figura 7 - Diagrama de atividade de Abertura do Incidente.....	41
Figura 8 - Diagrama de atividade de Fechamento do Incidente.....	42
Figura 9 - Diagrama de sequência Consulta Incidentes Semelhantes.....	43
Figura 10 - Modelo entidade relacionamento.....	44
Figura 11 - Diagrama de classe	46
Figura 12 - Ontologia de Categorias de Incidente	47
Figura 13 - Tabela de símbolos da função de similaridade	48
Figura 14 - Tela inicial da aplicação	62
Figura 15 - Tela de Abertura do Incidente.....	64
Figura 16 - Tela Encerramento do Incidente	65
Figura 17 - Consulta 1 para demonstrar a análise RBC	66
Figura 18 - Consulta 2 para demonstrar a análise RBC	67
Figura 19 - Consulta 1 para demonstrar análise de texto	68
Figura 20 - Consulta 3 para demonstrar a análise RBC	85
Figura 21 - Consulta 4 para demonstrar a análise RBC	86
Figura 22 - Consulta 5 para demonstrar a análise RBC	87
Figura 23 - Consulta 6 para demonstrar a análise RBC	87
Figura 24 - Consulta 2 para demonstrar análise de texto	88
Figura 25 - Consulta 3 para demonstrar análise de texto	89
Figura 26 - Consulta 4 para demonstrar análise de texto	89

LISTA DE QUADROS

Quadro 1 - Principais características do MR-MPS-SV, CMMI e ITIL.....	18
Quadro 2 - Níveis de maturidade do MR-MPS-SV.....	21
Quadro 3 - Detalhamento do caso de uso Cadastrar Incidente.....	38
Quadro 4 - Detalhamento do caso de uso Consultar Incidentes Semelhantes.....	39
Quadro 5 - Matriz de rastreabilidade.....	40
Quadro 6 - Arquitetura do sistema.....	44
Quadro 7 - Atributos do modelo RBC de incidente.....	48
Quadro 8 - Exemplo de incidente.....	50
Quadro 9 - Definições do modelo da ontologia.....	51
Quadro 10 - Trecho da estrutura da ontologia.....	52
Quadro 11 - Modelo da ontologia sendo carregado na interface.....	52
Quadro 12 – Método de criação do projeto do MyCBR.....	53
Quadro 13 – Método de carregamento das estruturas do MyCBR.....	54
Quadro 14 - Método consulta RBC.....	55
Quadro 15 - Cadastramento dos incidentes encerrados na base de casos MyCBR.....	56
Quadro 16 - Bibliotecas utilizadas do Apache Lucene e método de adição de documentos ...	58
Quadro 17 - Etapa de análise de similaridade e aplicação de filtros na consulta de incidentes semelhantes.....	59
Quadro 18 - Etapa de análise de texto na consulta de incidentes semelhantes.....	61
Quadro 19 - Comparação entre características do trabalho desenvolvido e os correlatos.....	69
Quadro 20 - Detalhamento do caso de uso Cadastrar Empresa.....	80
Quadro 21 - Detalhamento do caso de uso Cadastrar Setor.....	80
Quadro 22 - Detalhamento do caso de uso Cadastrar Equipamento.....	81
Quadro 23 - Detalhamento do caso de uso Cadastrar Software.....	82
Quadro 24 - Detalhamento do caso de uso Consultar Incidentes.....	83
Quadro 25 - Detalhamento do caso de uso Cadastrar Técnico.....	84

LISTA DE ABREVIATURAS E SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

AP – Atributos do Processo

API – *Application Programming Interface*

CMMI-SVC – *Capability Maturity Model Integration for Services*

GIN – Gerência de Incidentes

ISO – *International Standardization Organization*

ITIL – *Information Technology Infrastructure Library*

MER – Modelo Entidade Relacionamento

MPS.BR – Melhoria de Processo de Software Brasileiro

MPS-SV – Melhoria de Processo de Software Brasileiro de Serviços

MPS-SW – Melhoria de Processo de Software Brasileiro de Software

MR-MPS-SV – Modelo de Referência MPS de Serviços

MR-MPS-SW – Modelo de Referência MPS de Software

OTRS – *Open Technology Real Services*

OWL – *Web Ontology Language*

RAP – Resultados esperados de Atributos do Processo

RBC – Raciocínio Baseado em Casos

TI – Tecnologia da Informação

UML – *Unified Modeling Language*

XML – *eXtensible Markup Language*

W3C – *World Wide Web Consortium*

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS.....	15
1.2 ESTRUTURA.....	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 MPS.BR.....	16
2.2 MR-MPS-SV	17
2.3 NÍVEL G DE MATURIDADE DO MR-MPS-SV	19
2.4 GERÊNCIA DE INCIDENTES	21
2.4.1 Incidente.....	22
2.4.2 Solicitação de serviço.....	22
2.5 RESULTADOS ESPERADOS COM A IMPLEMENTAÇÃO DO GIN.....	23
2.5.1 GIN 1.....	24
2.5.2 GIN 2.....	25
2.5.3 GIN 3.....	25
2.5.4 GIN 4.....	25
2.5.5 GIN 5.....	26
2.5.6 GIN 6.....	26
2.5.7 GIN 7.....	27
2.6 FUNCIONALIDADE DE BUSCA POR CASOS DE INCIDENTES SEMELHANTES	27
2.6.1 Ontologia.....	27
2.6.2 OWL.....	28
2.6.3 Raciocínio baseado em caso.....	29
2.6.4 Apache Jena	29
2.6.5 MyCBR	30
2.6.6 Apache Lucene.....	30
2.7 TRABALHOS CORRELATOS	31
2.7.1 Ocomon.....	32
2.7.2 OTRS Help Desk	33
2.7.3 Sistema de apoio para otimização das atividades de suporte técnico de uma empresa de desenvolvimento de software.....	34
3 DESENVOLVIMENTO.....	35

3.1	MODELO DE DESENVOLVIMENTO	35
3.2	REQUISITOS	36
3.3	ESPECIFICAÇÃO	36
3.3.1	Diagrama de caso de uso	36
3.3.2	Diagrama de rastreabilidade RF x caso de uso	39
3.3.3	Diagramas de atividade	40
3.3.4	Diagrama de sequência	42
3.3.5	MER	43
3.3.6	Arquitetura do sistema	44
3.3.7	Diagrama de classe.....	44
3.3.8	Modelo da ontologia de categorias de incidente	46
3.3.9	Modelo RBC	47
3.3.10	Modelo de consulta de incidentes semelhantes.....	48
3.4	IMPLEMENTAÇÃO	50
3.4.1	Técnicas e ferramentas utilizadas.....	50
3.4.2	Operacionalidade da implementação	61
3.5	RESULTADOS E DISCUSSÕES.....	68
4	CONCLUSÕES	73
4.1	EXTENSÕES	75
APÊNDICE A – DETALHAMENTO DOS CASOS DE USO		79
APÊNDICE B – EXEMPLOS DE CONSULTA A INCIDENTES SEMELHANTES		85

1 INTRODUÇÃO

Nos últimos anos, com o crescimento e as mudanças no mercado empresarial, cada vez mais as empresas têm aumentado as suas necessidades com relação a qualidade dos serviços de Tecnologia da Informação (TI), exigindo um atendimento rápido e evolutivo. Com isso, a área de TI precisa sempre se manter na vanguarda, mantendo uma prestação de serviços de qualidade com uma abordagem orientada ao cliente (MAGALHÃES; PINHEIRO, 2007, p. 49).

Buscando uma forma de aprimorar e estruturar a prestação de serviços na área de TI, segundo a Associação para Promoção da Excelência do Software Brasileiro (2012, p. 6) “é essencial que os provedores de serviços coloquem a eficiência e a eficácia dos seus processos [...] visando à oferta de serviços conforme padrões internacionais de qualidade.” Para atender a esta necessidade, as empresas de TI podem se basear em diversas metodologias e modelos de referência como o *Information Technology Infrastructure Library* (ITIL), *Capability Maturity Model Integration for Services* (CMMI-SVC) e o modelo de Melhoria de Processo de Software Brasileiro (MPS.BR), que será utilizado neste trabalho que tem como intuito melhorar os processos internos e externos das organizações atuantes na área de TI.

Com isso, o modelo de referência MPS.BR propõe, segundo o Instituto de Tecnologia de Software (2015, n.p.) “impulsionar a melhoria da capacidade de desenvolvimento de software e serviços nas empresas brasileiras”. Sendo que este modelo de referência se divide em quatro áreas específicas: o MPS de Software (MPS-SW), MPS de Serviços (MPS-SV), MPS de Recursos Humanos (MPS-RH) e o MPS de Aquisição. Destes será dado foco aos MPS-SW e principalmente ao MPS-SV.

O MPS-SV possui foco no setor de prestação de serviços de TI, sendo baseado na mesma norma internacional homologado pela *International Standardization Organization* (ISO) ISO/IEC 20000, assim como o ITIL e o modelo CMMI-SVC (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2013a, p. 32). Segundo o Instituto de Tecnologia de Software, o MPS-SV (2015, n.p.) “serve para melhorar tanto os processos de serviços quanto o desempenho nos negócios das organizações públicas e privadas de qualquer porte”.

No entanto, a maior diferença entre eles é que o MPS-SV trata o gerenciamento de incidentes e as solicitações de serviços em conjunto conforme definido na ISO/IEC 20000-1 de 2011, já o ITIL e o CMMI-SVC dividem esses dois processos em etapas

distintas (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2013a, p. 32). Levando em conta a necessidade de melhoria no processo de gerenciamento de incidente dentro das empresas de TI, este trabalho visa desenvolver um software que auxilie os profissionais da área, mais especificamente no suporte e manutenção de computadores. Para que este controle seja feito de forma eficiente, otimizando as etapas de registro, classificação, resolução dos incidentes.

Com o intuito de otimizar a etapa de busca por uma solução ideal para o incidente, que não só resolva o incidente em si, mas que se adéque com relação a custos, tempo, complexidade e riscos que o caso exige. Para isso, este trabalho utiliza a abordagem que possui um modelo de uma ontologia para categorizar os incidentes, análise de texto sobre a definição e a descrição do incidente, e um modelo de sistema de Raciocínio Baseado em Casos (RBC) para buscar a solução que seja o mais similar possível com relação a custos, tempo, complexidade e riscos. Através de estudo dos casos cadastrados e concluídos, o sistema deverá retornar os casos de incidentes mais similares e a partir daí cabe ao técnico verificar se a solução utilizada em casos anteriores se aplica ao incidente analisado.

O desenvolvimento deste trabalho, se mostra relevante no âmbito empresarial, pois ele é uma revisão sobre diversos processos atualmente utilizados, porém por falta de tempo ou investimento acabam sendo gerenciados de forma inadequada (GUIMARÃES et al., 2008). O intuito do desenvolvimento deste software é possibilitar que o atendimento aos incidentes seja rápido e eficiente, para que a empresa possa conhecer a infraestrutura de cada cliente, e assim possa se moldar para poder atender as necessidades.

Outra questão que este trabalho se mostra relevante, é devido a sua funcionalidade específica, que visa facilitar a busca por casos de incidentes semelhantes. E para desenvolver esta busca o trabalho utiliza um modelo de uma ontologia *Web Ontology Language* (OWL) utilizando o Apache Jena. Uma ferramenta de análise textual, o Apache Lucene, em conjunto com um modelo criado em sistema de RBC, determina o grau de similaridade de acordo com as características especificadas do incidente com o MyCBR.

Por fim, este trabalho não propõe apenas o desenvolvimento de uma solução que aplique o modelo de referência MPS.BR-SV, além disso, sistema busca se tornar aplicável a uma empresa de TI da região, levando em conta as necessidades práticas da

empresa. A motivação para o desenvolvimento deste trabalho, surgiu devido a necessidade observada na execução de funções profissionais, em uma empresa de TI da região.

1.1 OBJETIVOS

O objetivo geral deste trabalho é desenvolver uma solução computacional para auxiliar a gerência de incidentes para a área de suporte e manutenção de computadores, com base no modelo de referência MPS-SV, especificando e aplicando um modelo de busca por casos de incidentes semelhantes.

Objetivos específicos do trabalho:

- a) identificar os principais elementos descritos pelo MR-MPS-SV, considerando as necessidades de uma empresa da área de suporte e manutenção de computadores;
- b) especificar um formalismo baseado em ontologias para representação e classificação dos incidentes;
- c) especificar o modelo para análise de similaridade dos casos utilizando o MyCBR;
- d) aplicar análise textual na definição e descrição dos incidentes utilizando o Apache Lucene.

1.2 ESTRUTURA

Este trabalho está organizado em quatro capítulos, sendo eles: introdução, fundamentação teórica, desenvolvimento e conclusões. No capítulo inicial, tem-se uma contextualização do tema do trabalho, apresentação dos objetivos e as justificativas de relevância. No capítulo seguinte é desenvolvida a fundamentação teórica básica para compreensão do trabalho, e no fim deste capítulo são apresentados os trabalhos correlatos. O capítulo três contém o desenvolvimento da ferramenta, bem como seus requisitos, detalhes de implementação relevantes, técnicas e ferramentas utilizadas e por fim resultados e discussões. O capítulo final trata das conclusões, e sugestões de extensão para a ferramenta.

2 FUNDAMENTAÇÃO TEÓRICA

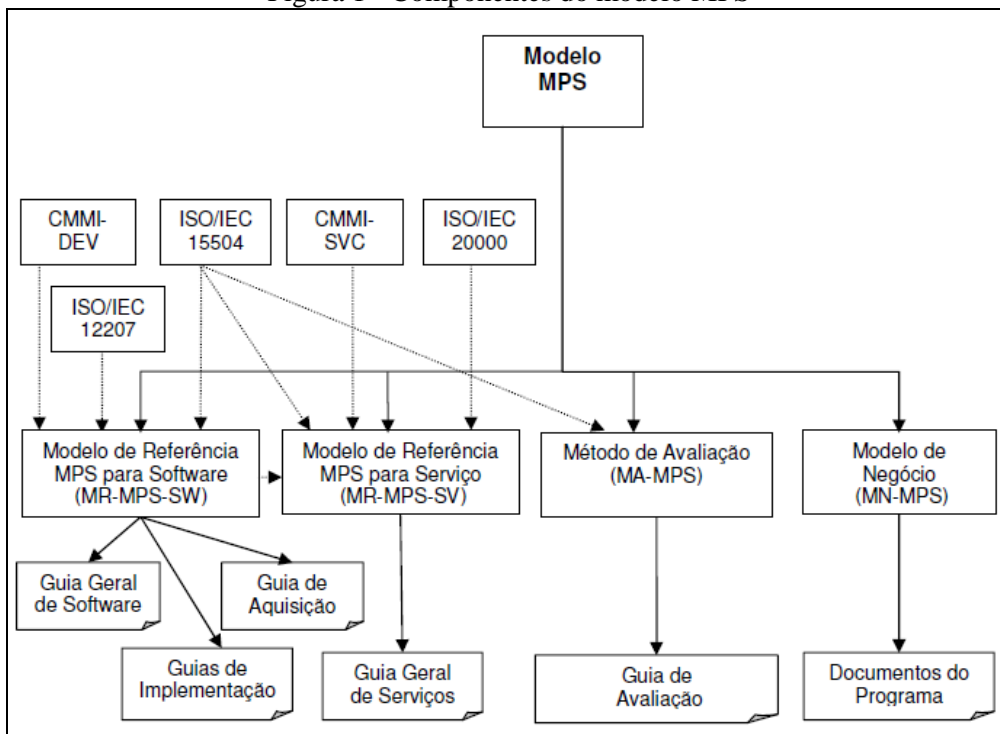
Este capítulo tem como objetivo descrever os principais assuntos necessários para a realização e o entendimento deste trabalho. Os assuntos foram subdivididos nas seguintes partes: a seção 2.1 MPS.BR descreve os conceitos básicos do modelo de referência, a seção 2.2 MR-MPS-SV descreve, de maneira mais aprofundada, o modelo de referência específico para serviços, a seção 2.3 detalha o nível G de maturidade do MR-MPS-SV, a seção 2.4 sobre gerência de incidentes aprofunda conceitos de serviço e incidentes, a seção 2.5 trata dos resultados esperados com a implementação do Gerência de Incidentes (GIN), e por fim a seção 2.6 sobre conceitos para a funcionalidade de busca de incidentes.

2.1 MPS.BR

O programa MPS.BR surgiu como modelo brasileiro para melhoria de processo de software alternativo ao CMMI. Criado pela Associação para Promoção da Excelência do Software Brasileiro (Softex) em 2003, com apoio do Ministério da Ciência e Tecnologia (MCT), Financiadora de Estudos e Projetos (FINEP), Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) e Banco Interamericano de Desenvolvimento (BID) (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2013b).

Os modelos MPS possuem quatro componentes, sendo eles: Modelo de Referência de Software (MR-MPS-SW), Modelo de Referência de Serviços (MR-MPS-SV), Método de Avaliação (MA-MPS) e Modelo de Negócio (MNMPS) como pode ser observado na Figura 1. Este trabalho toma como base o MR-MPS-SV, que como o próprio nome indica é voltado para a área de prestação de serviços. Também pode-se observar na Figura 1 que o MR-MPS-SW está em conformidade com os requisitos dos modelos de referência de processo da Norma Internacional ISO/IEC 15504-2, da ISO/IEC 15504 e do CMMI para Desenvolvimento (CMMI-DEV), que auxiliam na otimização dos processos de software, seja em implementações ou avaliações de processo (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2012, p.14).

Figura 1 - Componentes do modelo MPS



Fonte: MPS.BR Guia Geral de Serviços (2012, p. 14).

Na Figura 1 é possível observar que o MPS-SV está vinculado a duas normas, sendo elas: a ISO/IEC 15504 e a ISO/IEC 20000. A ISO/IEC 15504 é definida segundo a Associação Brasileira de Normas Técnicas (2008) (ABNT) “provê informações sobre conceitos de avaliação de processo e seu uso em dois contextos, o de melhoria de processo e o de determinação de capacidade de processo.”. Já a ISO/IEC 20000 é mais específica, levando em conta o objetivo, pois segundo a Associação para Promoção da Excelência do Software Brasileiro (2012, p. 16), “Esta norma provê a adoção de uma abordagem de processos integrada para a gestão de serviços de TI e alinha-se com as melhores práticas do ITIL para entrega e suporte de serviços.”.

2.2 MR-MPS-SV

Antes de detalhar o modelo de referência MPS-SV, seria interessante descrever comparativamente o MPS-SV com relação aos modelos CMMI e ITIL. Sendo que o MPS-SV se destaca por possuir um foco no mercado nacional e versão mais recente que os outros modelos, além de ter a possibilidade de implementação incremental, e dentre os modelos, embora seja o menos popular atualmente, ele se mostra como o mais acessível para empresas nacionais (ARAÚJO et al., 2014, p. 277).

O guia do MR-MPS-SV detalha os 7 níveis de maturidade, sendo o nível A o mais avançado e o G o nível mais básico do modelo de referência, cada nível possui

processos específicos que são descritos no Quadro 2, e os processos possuem atributos que são representados na coluna mais a direita. No entanto, o foco deste trabalho é um processo específico do nível G de maturidade que é a GIN.

O CMMI-SVC também possui uma classificação com níveis de maturidade, como pode ser observado no Quadro 1. O CMMI se organiza de uma forma diferente tendo uma divisão em duas classificações, uma com 4 níveis de capacidade e a outra com 5 de maturidade, porém, diferente do MPS-SV, para estar no primeiro nível de maturidade do CMMI não é necessária nenhuma atividade específica.

Quadro 1 - Principais características do MR-MPS-SV, CMMI e ITIL

	MR-MPS-SV	CMMI	ITIL
Início	Desenvolvido em 2003	Primeira versão em 2002	Tornou-se padrão em 1990
Abrangência	Nacional	Internacional	Internacional
Última versão	2013	2010	2011
Estruturação dos modelos	- Define níveis de maturidade; - Combinação entre processos e capacidades da organização.	- Gestão do trabalho; - Gestão de processos; - Serviços de criação; - Entrega e suporte. - Define níveis de maturidade;	- Conjunto de práticas que visa alinhar o gerenciamento do serviço de TI com o negócio.
Divisão dos modelos para melhoria dos processos	- 7 níveis de maturidade (processo evolutivo); - Os processos são descritos em termos de propósitos e resultados.	- 4 níveis de capacidade (0 a 3); - 5 níveis de maturidade (1 a 5); - Os processos são descritos em objetivos alcançados.	- Definido por um conjunto de boas práticas; - Divisão de processos por 5 áreas.

Fonte: Adaptado de Sousa (2013).

A questão que se mostra mais relevante na escolha entre ITIL, CMMI e o MPS-SV é segundo Bertolini, Ceratti e Silveira ([2014]) “[...] devido a sua proposta de melhorar o processo de software de uma forma gradual e de baixo custo”. Observando a

forma como o modelo de referência MPS-SV foi construído, o torna mais agradável e atrativo para as empresas brasileiras, pois ele foi construído especificamente pensando na situação do mercado nacional.

Então, o trabalho a ser desenvolvido se encaixa no nível de maturidade G, demonstrado no Quadro 2, onde um dos processos básicos é a GIN. Porém, observando o Quadro 2, podemos chegar a conclusão precipitada de que ainda é pouco pensar em apenas uma das etapas do nível mais baixo do MPS-SV, no entanto segundo dados de uma pesquisa realizada pela Softex o número de avaliações para adequação realizadas foi de 596, e destas, apenas 22 foram na área de serviços (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2015).

2.3 NÍVEL G DE MATURIDADE DO MR-MPS-SV

O nível G é subdividido em 5 partes sendo elas: Entrega de Serviços (ETS), Gerência de Incidentes, Gerência de Nível de Serviço (GNS), Gerência de Requisitos (GRE) e Gerência de Trabalhos (GTR) como pode ser observado no Quadro 2. Segundo a Associação para Promoção da Excelência do Software Brasileiro (2013a, p. 7) "Sua implementação deve ser executada com cautela por estabelecer o início da implantação de melhoria dos processos de gerenciamento de serviços na organização".

Na coluna de atributos do processo (AP), mostrado no Quadro 2, o nível G possui dois processos relacionados, que são AP 1.1 que se refere ao processo a ser executado e atingir os objetivos especificados, e o AP 2.1 se refere ao nível de gerenciamento do processo. O guia estabelece uma série de resultados esperados de atributos do processo (RAP):

- a) RAP 2 : Existe uma política organizacional que seja estabelecida e mantida para o processo;
- b) RAP 3 : A execução do processo seja planejada;
- c) RAP 4: A execução do processo seja monitorada para que os ajustes necessários sejam realizados;
- d) RAP 5: As informações e os recursos serem identificados e devidamente disponibilizados;
- e) RAP 7: A capacitação em termos de formação, treinamento e experiência dos profissionais envolvidos;
- f) RAP 8 : A comunicação entre as partes é planejada e executada;

- g) RAP 10: O processo ser planejado e executado. (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2012, p. 19).

Os maiores desafios para uma organização conseguir os resultados esperados no nível G de acordo com a Associação para Promoção da Excelência do Software Brasileiro (2013a, p. 7) são “(1) mudança de cultura organizacional, orientando a definição e melhoria dos processos de gerenciamento de serviços; (2) definição do conceito acerca do que é “serviço” para a organização.”

No entanto, é importante ressaltar que no nível G, o gerenciamento de serviços pode seguir os procedimentos normais que já são utilizados pela organização, então não é necessário que sejam feitas mudanças radicais nos processos da organização. Com isso, caso estes processos não estejam devidamente documentados e formalizados, ou seja, cabe a empresa descrever e especificar o modo como a prestação de serviço funciona na prática e como os incidentes e serviços são gerenciados.

Quadro 2 - Níveis de maturidade do MR-MPS-SV

Nível	Processos	Atributos de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Trabalhos – GTR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Capacidade – GCA	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência de Continuidade e Disponibilidade dos Serviços – GCD	
	Gerência de Decisões – GDE	
	Gerência de Liberação – GLI	
	Gerência de Riscos – GRI	
	Gerência da Segurança da Informação – GSI	
	Relato de Serviços – RLS	
D	Desenvolvimento do Sistema de Serviços – DSS	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
E	Avaliação de Melhoria do Processo Organizacional – AMP	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Definição do Processo Organizacional – DFP	
	Gerência de Mudanças – GMU	
	Gerência de Recursos Humanos – GRH	
	Gerência de Trabalhos – GTR (evolução)	
F	Aquisição – AQU	AP 1.1, AP 2.1 e AP 2.2
	Gerência de Configuração – GCO	
	Gerência de Problemas – GLP	
	Gerência de Portfólio de Trabalhos – GPT	
	Medição – MED	
G	Entrega de Serviços – ETS	AP 1.1 e AP 2.1
	Gerência de Incidentes – GIN	
	Gerência de Nível de Serviço – GNS	
	Gerência de Requisitos – GRE	
	Gerência de Trabalhos – GTR	

Fonte: MPS.BR Guia Geral de Serviços (2012, p. 24).

2.4 GERÊNCIA DE INCIDENTES

Magalhães e Pinheiro (2007, p. 69) definem o processo de gerenciamento de incidentes como “O processo [...] responsável pelo tratamento e pela resolução de todos os incidentes observados nos serviços de TI, visando ao restabelecimento dos serviços no menor prazo possível”. Já para Abreu e Fernandes (2012, p. 134) a gerência de incidentes “[...] visa restaurar a operação normal de um serviço no menor tempo possível, de forma a minimizar impactos adversos para o negócio”. Após definir a gerência de incidentes, deve-se destacar outras definições importantes para uma melhor compreensão do processo de GIN, que são a de incidente e solicitação de serviço.

2.4.1 Incidente

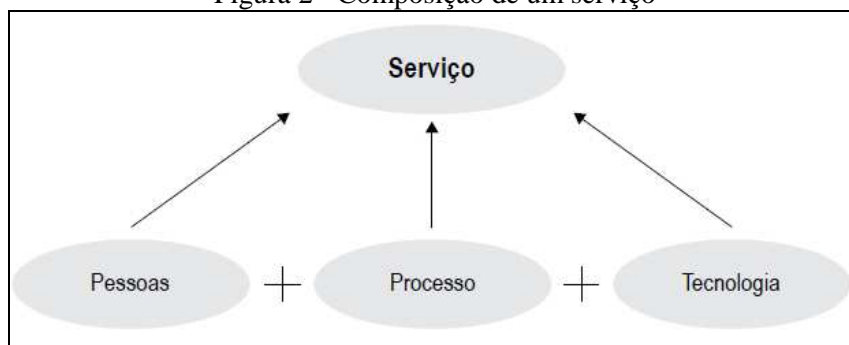
Segundo ISO/IEC 20000-1 (2011, n.p., tradução nossa) incidente é uma “interrupção não planejada de um serviço, uma redução na qualidade de um serviço ou um evento que ainda não impactou o serviço para o cliente”. Então, incidente consiste de qualquer situação imprevista que ocorra durante a prestação de serviço.

Uma situação hipotética que pode ser citada como exemplo, seria que durante o monitoramento de um servidor foi observada uma falha em uma de suas fontes de alimentação, portanto, deve ser efetuada a substituição da mesma para a correção do incidente. Basicamente a gerência de incidentes engloba todo o processo de identificar o problema, registrá-lo, categorizá-lo, resolvê-lo, encerrar e registrar a conclusão.

2.4.2 Solicitação de serviço

Serviço é composto como pode-se observar na Figura 2, por 3 elementos básicos, as pessoas envolvidas (Ex.: clientes ou prestadoras de serviços), os processos envolvidos para executar o serviço e a tecnologia empregada para realizar esta tarefa.

Figura 2 - Composição de um serviço



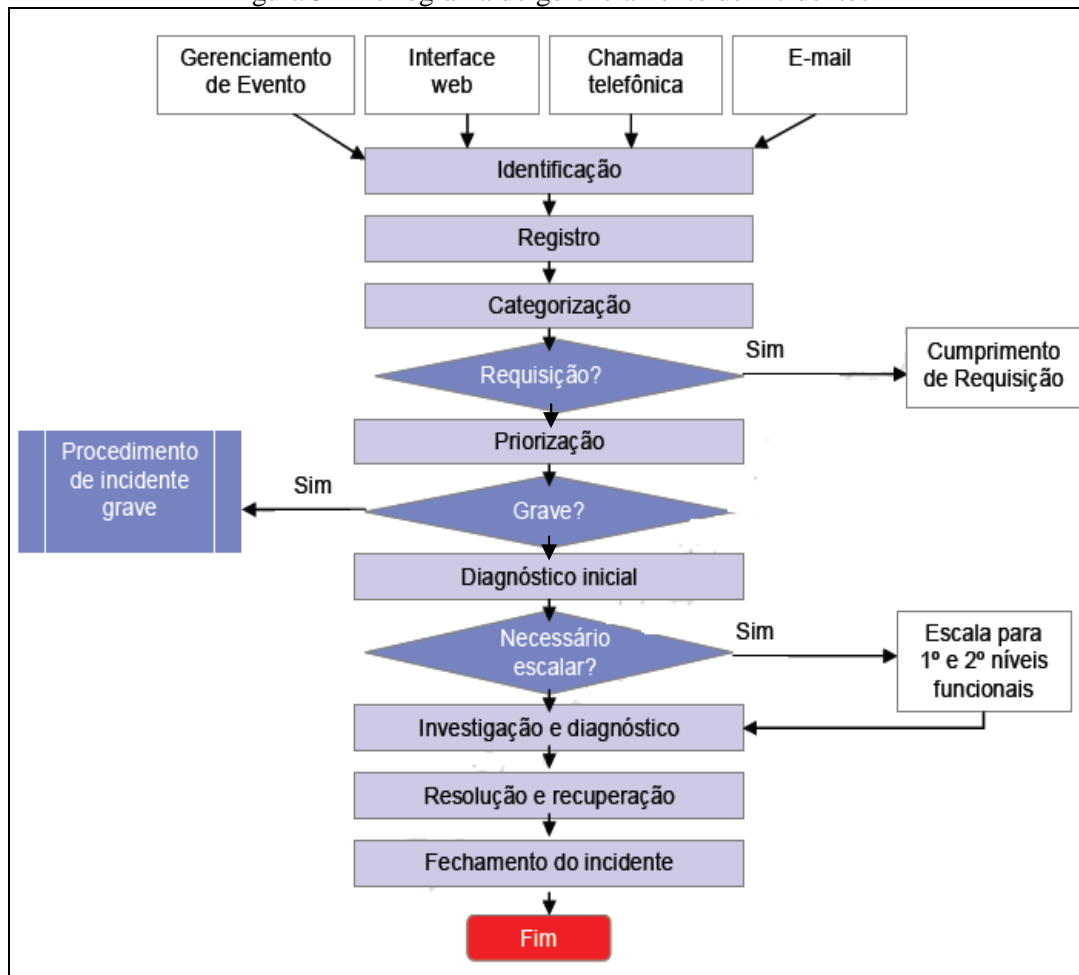
Fonte: Magalhães e Pinheiro (2007, p. 46).

Uma solicitação de serviço refere-se a qualquer solicitação do cliente em relação aos serviços prestados. Pode ser, por exemplo, um telefonema para esclarecimento de uma dúvida, a solicitação para reativação de uma senha de acesso, a solicitação de uma documentação, o esclarecimento de como acessar um serviço, a execução de mudanças de baixo risco pré-aprovadas etc. (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2013a, p. 33).

Embora este processo seja aprofundado nos trechos seguintes deste trabalho, pois este é um dos elementos chaves esperados no processo de implantação do GIN, é importante ilustrar, de forma conceitual, como o processo de solicitação de serviços pode ser organizado de acordo com a Figura 3. Observando a imagem, pode-se fazer o relacionamento da definição de solicitação de serviço e o fluxograma do processo,

aonde basicamente o problema chega a empresa responsável pela prestação de suporte e serviços, e passa pelas etapas de identificação, registro e categorização, a partir daí é realizada a requisição e a separação de acordo com a gravidade do problema, para então ter início a etapa análise, resolução e encerramento.

Figura 3 - Fluxograma do gerenciamento de incidentes



Fonte: Adaptado de Office of Government Commerce (2007, p. 165).

2.5 RESULTADOS ESPERADOS COM A IMPLEMENTAÇÃO DO GIN

O GIN se divide em 7 elementos fundamentais para a implementação, sendo eles:

- GIN 1: uma abordagem para o gerenciamento de incidentes e solicitação de serviço é estabelecida e mantida;
- GIN 2: um sistema de gerenciamento e controle de incidentes e solicitação de serviços é estabelecido e mantido;
- GIN 3: incidentes e solicitações de serviços são registrados e classificados;
- GIN 4: incidentes e solicitações de serviços são priorizados e analisados;

- e) GIN 5: incidentes e solicitações de serviços são resolvidos e encerrados;
- f) GIN 6: incidentes e solicitações de serviços que não progrediram conforme os acordos de nível de serviço são escalonados;
- g) GIN 7: informações a respeito da situação ou progresso de um incidente relatado ou solicitação de serviço são comunicadas às partes interessadas. (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2012, p. 27).

2.5.1 GIN 1

Para cumprir este elemento do Gin é necessário que a organização estabeleça políticas, diretrizes e procedimentos para o tratamento das solicitações de serviços e incidentes. Isto inclui as etapas de registro priorização, tratamento, escalonamento, resolução e comunicação as partes interessadas (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2013a, p. 34).

No entanto, também é interessante ressaltar que o guia de implementação não define como a empresa deve se organizar com relação as questões requisitadas pelo GIN 1. Então, para iniciar a etapa de adequação que possibilita a gestão de incidentes, é necessário que a empresa estruture bem suas políticas, processos e procedimentos na área de solicitações de serviços.

Outra questão relevante que o guia apresenta, é a estrutura necessária para o recebimento dos comunicados de incidentes. Pois, de acordo com a demanda, empresas podem optar por um setor dedicado ao atendimento, em caso de empresas de menor porte, o guia sugere que a própria equipe responsável pelo atendimento aos incidentes tenha a possibilidade de organizar e padronizar respostas e soluções para problemas cotidianos. Os procedimentos geralmente incluem elementos como o passo a passo da solução, a definição dos especialistas necessários, o tempo para resolução, entre outros elementos que podem ser incluídos de acordo com a necessidade (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2013a, p. 34).

Em empresas com funcionamento de horários por turno, ou com subdivisão de clientes por profissional, a definição e a classificação dos procedimentos para resolução de incidentes é um elemento fundamental. Uma situação hipotética que serve de exemplo, seria onde o profissional responsável pelo atendimento a um cliente em

específico, não esteja disponível para atender o chamado, então para que a organização possa solucionar este incidente o procedimento destacado anteriormente se mostra primordial.

2.5.2 GIN 2

Seguindo as recomendações propostas no GIN 1, o outro elemento básico para conseguir implementar o GIN é a utilização de um sistema de gerenciamento e controle de incidentes. Portanto, é interessante destacar que é exatamente neste elemento onde é mencionada a necessidade de buscar uma solução por software. E como veremos nas etapas seguintes, a presença de um sistema pra essa solução se justifica e quanto o processo pode ser facilitado e otimizado com a utilização do mesmo.

2.5.3 GIN 3

Nesta etapa o ponto chave é como as solicitações de serviços e os incidentes são registrados. O guia recomenda que as informações sejam padronizadas, tanto no momento da resolução do incidente, quanto para que depois seja realizada a análise para adoção de soluções definitivas (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2013a, p. 35).

Estão, um dos principais elementos dessa etapa é o registro adequado contendo informações básicas, tais como o serviço, a data do chamado, descrição do pedido e do problema. Outra necessidade que está relacionada a isso é um sistema de classificação e priorização de acordo com as solicitações de serviços, que possa extrair os dados necessários para que o profissional responsável possa organizar a resolução dos incidentes. A vantagem de possuir um controle deste tipo, é que em posse de informações como histórico de serviços prestados, tempo para resolução dos incidentes e soluções aplicadas é possível otimizar os processos da organização e com isso aumentar a satisfação do cliente, por consequência os lucros da empresa prestadora de serviços.

2.5.4 GIN 4

Este elemento pode variar muito de organização para organização, pois a priorização do serviço depende muito da política de negócio da empresa de TI, então é recomendado que haja um Acordo de Nível de Serviço (ANS). A Norma ISO/IEC 20000-2 a priorização dos incidentes deve ser feita considerando a prioridade, os perfis

disponíveis, requisitos competindo pelo recurso, o esforço e o custo para solução e tempo (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2013b).

Embora a prioridade de atendimento possa variar de acordo com a organização, é interessante que a empresa prestadora de serviços trate estas questões diretamente com o cliente caso o mesmo tenha uma grande demanda, pois muitas vezes apenas a classificação interna dos incidentes não consegue representar as reais necessidades do cliente. Um exemplo desta questão seria um incidente complexo que surge, e para solucioná-lo seria necessário o envolvimento de diversos profissionais, portanto o custo seria elevado, no entanto um incidente aparentemente simples está trazendo mais problemas para o cliente do que o incidente que demanda uma solução mais complexa. Por isso, cabe a organização realizar um estudo sobre os processos do cliente e analisar quais são as necessidades básicas de cada cliente, porém é evidente que para a realização deste tipo de análise a demanda e o retorno financeiro do cliente devem justificar o investimento no processo.

2.5.5 GIN 5

Este elemento do GIN trata especificamente de como as solicitações de serviços e incidentes devem ser atendidas mantendo os prazos estabelecidos nos ANSs. Mas, também é interessante ressaltar que a prioridade de atendimento a uma solicitação de serviço ou incidente que inicialmente tenha uma prioridade baixa, pode subir de prioridade com o passar do tempo (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2013a, p. 36).

Além disso, outras questões já mencionadas anteriormente voltam a ser necessárias nesta etapa, como a necessidade de o software disponibilizar as informações necessárias de acordo com o procedimento a ser executado. Pois, a categorização da solução realizada é um elemento chave que juntamente com o conhecimento da infraestrutura do cliente, possibilitam uma melhora no atendimento.

2.5.6 GIN 6

O GIN 6 é a etapa relacionada ao escalonamento das solicitações e incidentes que não progrediram, esta questão de escalonar um serviço é necessário em organizações que possuem um corpo de profissionais extenso e variado. Porém, dependendo do tamanho da empresa talvez estas questões acabem se tornando pouco relevantes ou até desnecessárias.

Basicamente, o guia sugere que se a organização utilizar uma ferramenta para GIN, ela deve permitir que sejam configurados parâmetros para o escalonamento automático, com base no tempo e prioridade. No entanto o sistema deve possibilitar que as regras sejam quebradas de acordo com a necessidade do cliente para com a empresa prestadora de serviços (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO, 2013a, p. 37).

2.5.7 GIN 7

Segundo a Associação para Promoção da Excelência do Software Brasileiro (2013a, p. 37) “É importante que a organização defina e coloque em execução procedimentos para que as partes envolvidas possam tomar conhecimento do andamento das solicitações de serviço e do tratamento dos incidentes”. No entanto, devem ser consideradas as necessidades e formas de trabalho das empresas que prestam serviços e os seus clientes, pois dependendo do tamanho da organização, certas atividades acabam se tornando desnecessárias, ou até mesmo o cliente não possui a disponibilidade para observar o andamento das solicitações.

2.6 FUNCIONALIDADE DE BUSCA POR CASOS DE INCIDENTES SEMELHANTES

Em algumas situações, um incidente ocorre em um ambiente com diversos elementos fornecidos por vários fabricantes com tecnologias diferentes que geram uma série de variantes para resolução do problema. São então feitas reuniões técnicas para buscar uma solução, a qual é inserida nos registros (CORRÊA; GASPAR; OLIVEIRA, [201?]). Como pode-se observar na citação acima, a parte de análise e registro dos incidentes é crucial, e portanto, uma funcionalidade específica para busca de casos de incidentes semelhantes ao pesquisado, se mostra fundamental já que isso poderia otimizar e poupar muitos esforços da empresa.

2.6.1 Ontologia

O conceito de ontologia é originário da Filosofia que trata um ser enquanto ser, e possui como intuito compreender identidades e grupos de identidades, sendo vista como uma forma sistemática descrição da existência. Então para a Filosofia cada área de conhecimento deve possuir a sua própria ontologia para representar a sua área de conhecimento, definida por seus termos e suas teorias (HEINZLE, 2011, p. 106).

Para a da Ciência da Computação ontologia pode ser definida de maneira simples como um catálogo que define os “tipos de coisas” em um domínio em uma linguagem determinada (SOWA, 1999). Pode-se definir ontologia no contexto da ciência da computação, como um conjunto de primitivas de representação para modelar um domínio de conhecimento. Essas primitivas de representação compostas por classes (ou conjuntos), atributos (ou propriedades) e as relações (ou relação entre os membros da classe) (GRUBER, 2009).

Uma ontologia pode ser utilizada para diversas finalidades como pode ser observado na colocação de Almeida e Bax (2003) “Ontologias são utilizadas em projetos de domínios como gestão do conhecimento, comércio eletrônico, processamento de linguagens naturais, recuperação da informação na Web, de cunho educacional, entre outros”. Neste trabalho a ontologia será utilizada para o armazenamento de dados, e a linguagem escolhida foi a OWL.

2.6.2 OWL

Uma ontologia desenvolvida na linguagem OWL é composta por estrutura baseada em uma marcação semântica feita para descrever classes e os relacionamentos entre elas, sendo recomendada pelo *World Wide Web Consortium* (W3C) (CARVALHO; LIMA, 2005). Também é importante ressaltar que:

A OWL é uma revisão da linguagem DAML+OIL. Possuindo mais facilidades para expressar significados e semânticas do que XML, RDF e RDF Schema, embora seja baseada em RDF e RDF Schema e utilize a sintaxe XML. A OWL foi projetada para ser usada por aplicações que necessitem processar o conteúdo de informações, ao invés de somente apresentar a visualização destas informações (CARVALHO; LIMA, 2005).

As ontologias têm uma série de elementos básicos para formalizar o conhecimento, sendo eles: classes, relacionamentos, axiomas e instâncias. Classes são unidades básicas de uma ontologia que representam conjuntos de elementos com os mesmos atributos e formam conceitos que definem um objeto. Estes conceitos representam todas as coisas relacionadas ao domínio que se pretende modelar, incluindo objetos, tarefas, ações e etc. As ligações entre estes conceitos se dão através dos relacionamentos ou relações. As relações descrevem as interações entre os conceitos, as quais representam os relacionamentos semânticos envolvidos no domínio. Os axiomas são regras relativas às relações que devem ser cumpridas pelos elementos de uma ontologia obrigatoriamente, que são as restrições. E por fim as instâncias representam os

elementos ou objetos de uma ontologia, sendo assim, são exemplares individuais das classes (HEINZLE, 2011, p. 107).

O tipo específico de sub-linguagem da OWL que optou-se por ser utilizado foi o OWL DL que garante que o usuário possa utilizar todos os recursos de uma ontologia sem perdas computacionais utilizado para a representação de raciocínio e modelo de conhecimento (W3C, 2014).

2.6.3 Raciocínio baseado em caso

Para um ser humano tentar compreender algo, ele busca em sua memória algo que possa ajudá-lo nesta compreensão, ou seja, o ser humano se recorda de algo que já foi compreendido que lhe é semelhante ao da situação atual para tentar solucionar o problema (CARVALHO, 1996 apud HEINRICH, 2000). Sendo assim, um RBC é semelhante a lógica humana, pois o RBC busca resolver problemas adaptando soluções que foram utilizadas para resolver problemas passados (RIESBECK; SCHANK, 1989, p. 25).

Para exemplificar a funcionalidade e como se utiliza um RBC, pode-se dar um exemplo de um incidente onde um computador não consegue acessar a internet (incidente). Com isso, o técnico pesquisa por problemas semelhantes no sistema, então o sistema, baseado na análise de similaridade, apresenta soluções mais similares já utilizadas para resolver incidentes semelhantes como, por exemplo, verificar a placa de rede, o cabo de rede e etc. Após verificar a solução do incidente, o técnico pode analisar se alguma das soluções foi válida e caso não seja o técnico de suporte deve registrar a nova solução para futuras consultas.

2.6.4 Apache Jena

Este *framework* foi desenvolvido por pesquisadores dos laboratórios da Hewlett-Packard (HP) na divisão do Reino Unido, sendo lançado em 2001 e se tornando *open source* em 2009. Com o intuito de possibilitar, criar, armazenar e manipular dados semânticos (DICKINSON, 2011).

Dentre as diversas funcionalidades que o framework possui para criação de uma ontologia, pode-se destacar a *Application Programming Interface* (API) denominada Jena Ontology API desenvolvida em linguagem neutra, por exemplo, uma classe representada em Java com a classe `OntClass`, pode representar uma classe OWL ou

Resource Description Framework Schema (RDFS) (APACHE JENA, [201?b]). Com isso, é interessante ressaltar outro recurso que o Jena oferece, que é a biblioteca Java voltada para a construção e especificação de ontologias. Embora os recursos disponibilizados pelo framework ainda não ofereçam todas as funcionalidades necessárias a uma ontologia, para este trabalho o Jena se mostra adequado possibilitando a especificação necessária para categorizar os diversos tipos de serviços (APACHE JENA, [201?a]).

2.6.5 MyCBR

O MyCBR é uma ferramenta para o desenvolvimento de sistemas RBC *open-source*, desenvolvida pelo Centro de Competência em RBC da *Deutsches Forschungszentrum für Künstliche Intelligenz GmbH* (DFKI) e a Escola de Informática e Tecnologia da *University of West London* (UWL) (MYCBR, 2015). Pode-se destacar que a ferramenta possui diversas medidas de similaridade para tipos de dados inteiros, de ponto flutuante, para símbolos e taxonomia, entre outros, e cada atributo pode ter várias medidas de similaridade (MYCBR, 2015).

Além do MyCBR Workbench, a ferramenta disponibiliza um *Software Development Kit* (SDK) que disponibiliza as funcionalidades já mencionadas, e possibilita a utilização de projetos criados no MyCBR Workbench, assim como a alteração nas regras do conceito do RBC e alimentação da base de casos (MYCBR, 2015). No entanto a maior deficiência apresentada pelo MyCBR é o uso apenas de arquivos *eXtensible Markup Language* (XML) para armazenar as informações (FERNANDES; KRAUS, 2010).

2.6.6 Apache Lucene

O Apache Lucene é um motor de busca de texto desenvolvido em Java, criado por Doug Cutting em 2000. Trata-se de um projeto de código aberto, com uso livre para aplicações comerciais e de código aberto sobre a licença Apache Foundation (LUCENE, 2012).

O processo de busca da biblioteca é realizado em duas etapas: indexação e pesquisa. A indexação gera uma estrutura de dados indexada para a busca baseada em palavras-chave, e a pesquisa consulta os índices criados na etapa anterior pelas palavras utilizadas como entrada para análise. Por fim, a consulta classifica e organiza os resultados pelo grau de similaridade com o texto (PAMPLONA, 2009).

Algumas das principais vantagens na biblioteca são a escalabilidade e alto desempenho de indexação e os algoritmos de pesquisa:

- a) mais 150 GB por hora;
- b) requisitos de memória RAM de 1MB;
- c) indexação incremental e indexação de grandes cargas de texto;
- d) tamanho do índice de cerca de 20% à 30% do tamanho do texto indexado;
- e) classificação das buscas por grau de similaridade;
- f) buscas por frases, palavras-chave, proximidade, intervalo e outras;
- g) busca por campos (título, autor, conteúdo);
- h) entre muitas outras (LUCENE, 2012).

O Lucene oferece um agradável nível de abstração para um conjunto poderoso de técnicas baseadas no modelo Vetorial e Booleano. O desenvolvedor não precisa conhecer rotinas e algoritmos de indexação, nem de consulta. Basta utilizá-los através da API da biblioteca (PAMPLONA, 2009).

Outra questão relevante com relação ao Lucene, é que para realizar a função de análise de texto, foi utilizado a classe `StandardTokenizer` que é um *tokenizer* baseado em uma gramática construída no JFlex, um analisador léxico feito em Java (JFLEX, 2015). O `StandardFilter` é o responsável por normalizar os tokens extraídos do `StandardTokenizer`. O `LowerCaseFilter` normaliza os tokens para letras minúsculas. Por fim `StopFilter` remove as *stop words* utilizando a lista de *stop words* em inglês (LUCENE, 2015).

2.7 TRABALHOS CORRELATOS

Existem muitas ferramentas comerciais voltadas para o gerenciamento de prestação de serviços, incluindo soluções de código aberto e corporativas, no entanto a maior parte não possui foco específico no tratamento da gestão de incidentes. A maioria das ferramentas encontradas são fundamentadas em conceitos do ITIL, tanto que durante as pesquisas não foram encontradas aplicações de gestão de incidentes com base no modelo de referência MPS-SV.

Além disso, outra ferramenta que é interessante mencionar, é um sistema de otimização das atividades de suporte para uma empresa de software que utiliza RBC. Concluindo, são apresentadas 2 ferramentas comerciais voltadas para gerência de incidentes de serviço que mais se aproximam as regras de negócio da solução proposta

pelo trabalho, e uma outra ferramenta desenvolvida em um trabalho de conclusão de curso.

2.7.1 Ocomon

O Ocomon é uma ferramenta de *Helpdesk* voltado para o gerenciamento de chamados de suporte e controle integrado de inventário para equipamentos de informática. O Ocomon segundo Ribeiro ([200?]) “foi concebido sob a visão de software *OpenSource* sob o modelo *General Public License* (GPL) de licenciamento, utilizando tecnologias e ferramentas livres para o seu desenvolvimento e manutenção.”

É importante ressaltar que a ferramenta foi desenvolvida com as linguagens em PHP, HTML, CSS e Java Script, utilizando o banco de dados MySQL. Podendo ser utilizado em qualquer navegador, sendo necessário apenas um servidor Apache e o MySQL.

Possui dois módulos, um voltado para o gerenciamento de chamados e outro voltado apenas para a manipulação de equipamentos. O módulo mais relevante para este trabalho é o de gerenciamento de chamados com funções como (RIBEIRO, [200?]):

- a) abertura de chamados;
- b) vínculo do chamado com equipamentos;
- c) busca a informações de equipamentos;
- d) acompanhamento de atendimento;
- e) classificação de chamados;
- f) encerramento de chamados.

Como pode-se observa na Figura 4, na tela principal de ocorrências do Ocomon podemos verificar as principais funcionalidades de abertura de chamado, consulta, relatórios e soluções. Além disso, no quadro central têm-se as ocorrências pendentes no sistema e as do usuário em específico, descrevendo a área, problema, contato, local e status do atendimento.

Figura 4 - Tela de ocorrências do Ocomon

OcoMon - Módulo de Ocorrências

Foi encontrada 1 ocorrência pendente para este usuário.

Chamado	Problema	Contato Ramal	Local	Status	válido	RESP.	SOLUC.
5	IMPRESSORA	Marta 8795	CONTABILIDADE	Em atendimento	00:06:30		

Foram encontradas 3 ocorrências pendentes no sistema.

Chamado Área	Problema	Contato Ramal	Local	Tempo válido	Ação	RESP.	SOLUC.
6 HELPDESK	INSTALAÇÃO DE SOFTWARE	Ademar 5264	Recursos Humanos	00:05:04			
7 HELPDESK	OTIMIZAÇÃO DE S.O.	Cassius 2545	Tesouraria	00:03:21			
8 HELPDESK	CONFIGURAÇÃO DE SOFTWARE	Nelso 9652	DIRETORIA	00:02:22			

Fonte: Ribeiro ([200?]).

2.7.2 OTRS Help Desk

O grupo *Open Technology Real Services* (OTRS) é uma comunidade *Open Source* que trabalha no desenvolvimento de ferramentas no ramo empresarial, desenvolveu o OTRS Help Desk que é basicamente uma ferramenta de gerenciamento de serviços que está presente no mercado desde 2001. Tendo como intuito auxiliar empresas de TI nos processos de registro, classificação, priorização e definição de fluxo, para que o gerenciamento de serviço se adapte as regras de cada negócio (RIBEIRO, 2012).

O OTRS Help Desk se fundamenta nos conceitos do ITIL, abrangendo questões como gerenciamento de incidentes, gerenciamento de mudanças, gerenciamento de nível de serviço, gerenciamento de itens de configuração, gerenciamento do conhecimento, melhoria contínua de serviço através de pesquisas de satisfação, definição de papéis, entre outras funcionalidades (RIBEIRO, 2012). Outra questão que esta ferramenta se mostra interessante, é devido a ela possibilitar ajustes e adaptações de acordo com as necessidades da empresa e por possuir licença *Affero General Public License V3* (AGPL) que garante a possibilidade de baixar, instalar, usar e modificar o software, sem nenhum custo. No entanto para suportar esta ferramenta é necessária toda uma estrutura de servidor, banco de dados e servidor web, além de um serviço de e-mail suportado pelo software. (OTRS, 2015).

2.7.3 Sistema de apoio para otimização das atividades de suporte técnico de uma empresa de desenvolvimento de software

Sistema proposto por Silva (2002) tem como objetivo desenvolver uma aplicação com base em conceitos de RBC, para auxiliar os técnicos responsáveis pelo suporte ao usuário com soluções em problemas com software, sendo assim muito semelhante a proposta deste trabalho. Ambos buscam agregar a experiência dos técnicos que utilizam o sistema para serem utilizadas no futuro em soluções semelhantes.

O trabalho utiliza os conceitos de RBC para realizar a análise de similaridade com base em um conjunto de palavras-passe inseridas pelo usuário na hora da consulta, que são comparadas com a descrição dos casos de uso cadastrados na base de conhecimento. Também é interessante ressaltar que para recuperação de dados foi utilizada a técnica do vizinho mais próximo (SILVA, 2002, p. 61).

A busca de palavras-passe funciona da seguinte forma o usuário ao cadastrar um problema indica quais são as palavras chave e qual é o grau de importância de cada uma delas. Para então quando for realizada uma busca basta o usuário inserir as palavras chaves que o sistema retorna os casos indicando o grau de similaridade com as expressões utilizadas como pode ser observado na Figura 5.

Figura 5 - Interface de busca de solução Silva (2002)

Informe as Palavras que devem ser pesquisadas

Palavra 1 GENERAL	Palavra 2 SOL	Palavra 3 ERROR
Palavra 4 MICROSOFT	Palavra 5 ODBC	Palavra 6

Clique no caso para ver detalhes

Caso	Similaridade (%)
9	100.00
1	10.00

Fonte: Silva (2002).

3 DESENVOLVIMENTO

Este capítulo tem como objetivo descrever o desenvolvimento do software, demonstrando o modelo de desenvolvimento, os requisitos, a especificação do projeto, detalhes da implementação e a operacionalidade das principais funcionalidades.

3.1 MODELO DE DESENVOLVIMENTO

Para o desenvolvimento da aplicação, foi adotado um modelo de ciclo de vida específico que não segue exatamente nenhum ciclo de desenvolvimento tradicional, mesmo sendo semelhante ao iterativo-incremental em seus processos. Deve-se ressaltar quais foram as principais etapas do modelo:

- a) especificação do sistema em paralelo com as pesquisas e testes das bibliotecas: na etapa de especificação foram realizados o levantamento e análise dos requisitos, desenvolvimento dos diagramas iniciais e a modelagem da solução. Na etapa de pesquisas e testes das bibliotecas foram realizadas pesquisas com relação a biblioteca para ontologia, RBC e análise de textos;
- b) implementação das funcionalidades básicas e desenvolvimento da interface: nesta etapa foram realizados a implementação dos cadastros e realizados os testes do mesmo. Além disso, foram feitos ajustes nos diagramas para se adequar a solução proposta e o desenvolvimento da interface;
- c) especificação do modelo de análise de incidentes semelhantes: modelagem da ontologia, do modelo de incidente para RBC e do modelo de consulta unindo a ontologia, o modelo RBC de incidente e a análise de texto para a criação da solução;
- d) implementação das funcionalidades específicas: nesta etapa foram realizados a implementação das consultas e do modelo de análise de incidentes semelhantes;
- e) especificação dos diagramas finais e testes.

Mesmo não tendo adotado um modelo específico para o desenvolvimento do software, deve-se destacar que o trabalho foi desenvolvido com base nas recomendações do MR-MPS-SV.

3.2 REQUISITOS

Os requisitos do sistema a ser desenvolvido são:

- a) O sistema deve permitir o cadastro de empresas (Requisito Funcional - RF);
- b) O sistema deve permitir o cadastro de setores (RF);
- c) O sistema deve permitir o cadastro de equipamentos tecnológicos (RF);
- d) O sistema deve permitir o cadastro de softwares (RF);
- e) O sistema deve permitir o cadastro de incidentes (RF);
- f) O sistema deve permitir a consulta geral de incidentes (RF);
- g) O sistema deve permitir a consulta de incidentes indicando o grau de similaridade entre os incidentes mais semelhantes ao consultado (RF);
- h) O sistema deve permitir o cadastro de profissionais que executam funções suporte e manutenção (RF);
- i) O sistema deverá se basear no modelo de referência MPS-SV para área de GIN (Requisito Não Funcional - RNF);
- j) O sistema deve utilizar a linguagem Java 1.8 (RNF);
- k) O sistema deve utilizar o banco de dados MySQL 6.3 (RNF);
- l) O sistema deve utilizar a biblioteca Apache Jena 2.13.0 (RNF);
- m) O sistema deve utilizar a biblioteca MyCBR 3.1 (RNF);
- n) O sistema deve utilizar a biblioteca Apache Lucene 5.3.0 (RNF).

3.3 ESPECIFICAÇÃO

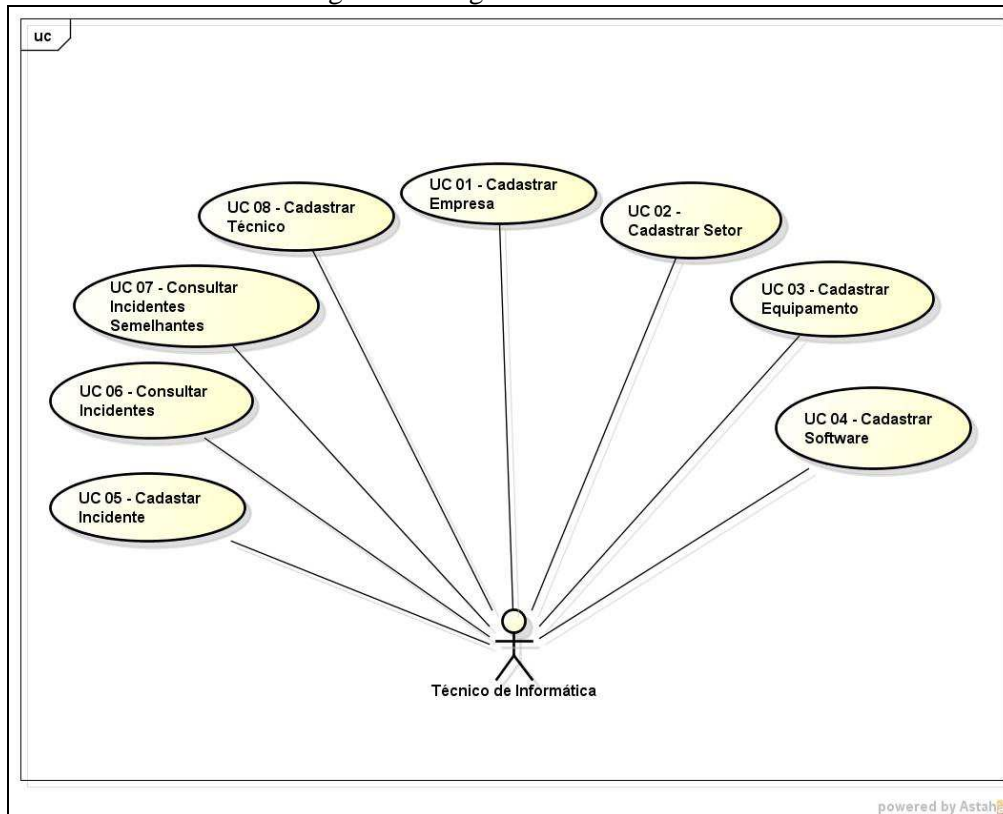
Neste capítulo são apresentados os principais diagramas e modelos desenvolvidos para especificação do software, utilizando alguns dos diagramas da *Unified Modeling Language* (UML) para representações gráficas e textuais, e o modelo conceitual Modelo Entidade Relacionamento (MER) para representação do modelo de dados. Sendo que as ferramentas usadas para a criação dos diagramas e modelos foram o Astah Community 6.9.0 para o UML e MySQL Workbench 6.3 para MER.

3.3.1 Diagrama de caso de uso

A Figura 6 demonstra as possíveis ações para o único ator presente no sistema, o Técnico de Informática. Conforme pode-se observar, o Técnico de Informática tem a possibilidade de realizar todas as operações de inclusão, edição, consulta e exclusão sobre os cadastros de técnico, empresa, setor, equipamento, software e

incidente. Além disso, a Figura 6 apresenta os UC 05 que representa a consulta de incidentes em geral, como é comum em todas as aplicações de gestão de incidentes para o controle e atendimento dos mesmos. O UC 07 mostra a principal funcionalidade do sistema que é a consulta por incidentes semelhantes.

Figura 6 - Diagrama de caso de uso



Com o intuito de exemplificar as principais funcionalidades do sistema descritas na Figura 6, que são os UC 05 e UC 07, representam a manipulação do cadastro de incidente e a consulta de incidentes semelhantes respectivamente. Pode-se observar o Quadro 3, que detalha o processo de manipulação do cadastro de incidentes demonstrando o papel do ator e o do sistema na atividade, e os possíveis caminhos alternativos e exceções ao cenário principal, bem como as pré-condições e pós-condições necessárias para realização do caso de uso.

Quadro 3 - Detalhamento do caso de uso Cadastrar Incidente

UC 05 – Cadastrar Incidente	
Ator	Técnico
Objetivo	Permitir ao técnico cadastrar um incidente desde a etapa de abertura até o encerramento do mesmo.
Pré-condições	Ter previamente cadastrados a empresa, setor e o técnico que serão referenciados no cadastro do incidente. E caso o cadastro referencie o equipamento de um setor ou um software estes também devem estar cadastrados antes da realização da operação.
Cenário principal	<ol style="list-style-type: none"> 1) O técnico seleciona a opção “Cadastros”; 2) O técnico seleciona a opção “Incidente” no submenu de “Cadastros”; 3) O técnico seleciona a opção “Abertura” no item do submenu “Incidente”; 4) O técnico informa o código do incidente; 5) O técnico informa o restante dos dados de abertura do incidente; 6) O técnico informa o próprio nome; 7) O técnico informa o nome da empresa que ocorreu o incidente; 8) O técnico informa o nome do setor da empresa em que ocorreu o incidente; 9) O técnico informa o nome do equipamento que ocorreu o incidente; 10) O técnico informa o nome do software que ocorreu o incidente; 11) O técnico seleciona a opção “Cadastrar”; 12) O sistema registra os dados do incidente no banco de dados.
Cenário alternativo	<p>No passo 3, caso o técnico selecione a opção “Encerramento” no item do submenu “Incidente”:</p> <ol style="list-style-type: none"> 3.1) O técnico informa o código do incidente; 3.2) O técnico informa o restante dos dados de encerramento do incidente; 3.3) O técnico seleciona a opção “Encerrar”; 3.4) O sistema registra na base de incidentes RBC; 3.5) Retorne ao item 12;
Cenário alternativo	<p>No passo 9, caso o técnico não informe o equipamento:</p> <ol style="list-style-type: none"> 9.1) Retorne ao item 11.
Cenário alternativo	<p>No passo 10, caso o técnico não informe o software:</p> <ol style="list-style-type: none"> 10.1) Retorne ao item 11.
Pós-condições	O incidente pode ser consultado no menu de “Consultas” no submenu “Incidentes Semelhantes” e no “Incidentes”.

O Quadro 4 detalha funcionalidade de busca por incidentes semelhantes, descrevendo o papel do ator no caso de uso, e quais são as operações realizadas pelo sistema para retornar a lista de incidentes semelhantes. Pode-se destacar a partir do item 9 do Quadro 4, que é realizado um filtro por categoria que visa diminuir o escopo da pesquisa só para os incidentes classificados naquela categoria da ontologia.

Quadro 4 - Detalhamento do caso de uso Consultar Incidentes Semelhantes

UC 07 – Consultar Incidentes Semelhantes	
Ator	Técnico
Objetivo	Permitir ao técnico encontrar incidentes que sejam semelhantes ao que desejasse.
Pré-condições	Ter previamente cadastrados incidentes que sejam semelhantes ao incidente consultado. E o incidente deve estar com status encerrado, para constar na consulta.
Cenário principal	<ol style="list-style-type: none"> 1) O técnico seleciona o item “Incidentes Semelhantes” no menu “Consultas”; 2) O técnico seleciona a categoria do incidente; 3) O técnico informa a definição e a descrição do incidente; 4) O técnico informa a descrição do equipamento; 5) O técnico informa o nome do software; 6) O técnico informa o restante de informações para avaliação de similaridade; 7) O técnico seleciona a opção “Consultar”; 8) O sistema faz a análise de similaridade com os atributos informados; 9) O sistema faz um filtro por categoria, descrição de equipamento e nome do software; 10) O sistema faz uma análise de busca por palavras em comum na definição e na descrição do incidente; 11) O sistema mostra o resultado da consulta retornando os incidentes mais semelhantes indicando o grau de similaridade entre eles e outros dados do incidente.
Cenário alternativo	No passo 8, caso os atributos não sejam informados: 8.1) Retorne ao item 9.
Cenário alternativo	No passo 9, caso a descrição do equipamento ou nome do software não sejam informados: 9.1) Retorne ao item 10.
Cenário alternativo	No passo 10, caso a definição ou a descrição do incidente não sejam informados: 10.1) Retorne ao item 11.

3.3.2 Diagrama de rastreabilidade RF x caso de uso

O Quadro 5 mostra a matriz de rastreabilidade entre os requisitos na seção 3.2 e o caso de uso da Figura 6.

Quadro 5 - Matriz de rastreabilidade

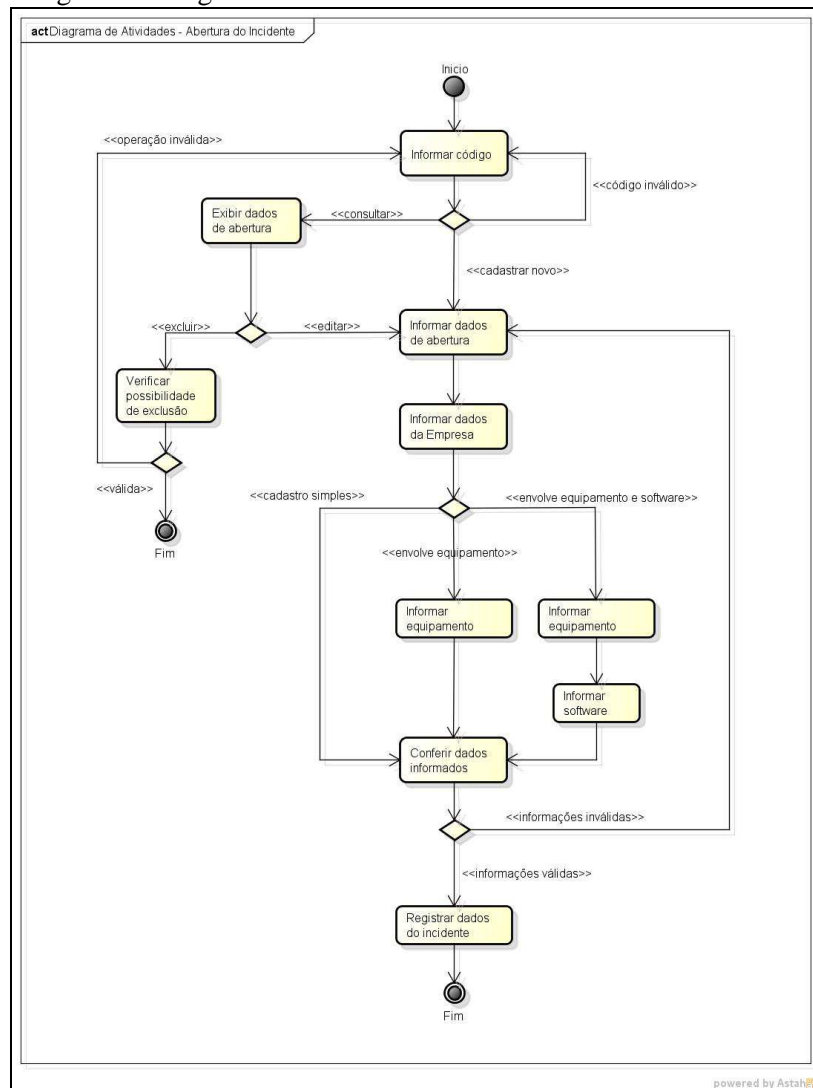
Requisitos Funcionais	Casos de Uso
RF 01: O sistema deve permitir o cadastro de empresas.	UC 01
RF 02: O sistema deve permitir o cadastro de setores.	UC 02
RF 03: O sistema deve permitir o cadastro de equipamentos tecnológicos.	UC 03
RF 04: O sistema deve permitir o cadastro de softwares.	UC 04
RF 05: O sistema deve permitir o cadastro de incidentes.	UC 05
RF 06: O sistema deve permitir a consulta geral de incidentes.	UC 06
RF 07: O sistema deve permitir a consulta de incidentes indicando o grau de similaridade entre os incidentes mais semelhantes ao consultado.	UC 07
RF 08: O sistema deve permitir o cadastro de profissionais que executem funções de suporte e manutenção.	UC 08

3.3.3 Diagramas de atividade

Os diagramas de atividade da Figura 7 e Figura 8 complementam o diagrama de caso de uso do Quadro 3, que descreve detalhadamente as possíveis operações sobre o cadastro de incidentes. Os diagramas de atividade mencionados detalham os processos necessários para realização da abertura na Figura 7 e o encerramento do incidente na Figura 8.

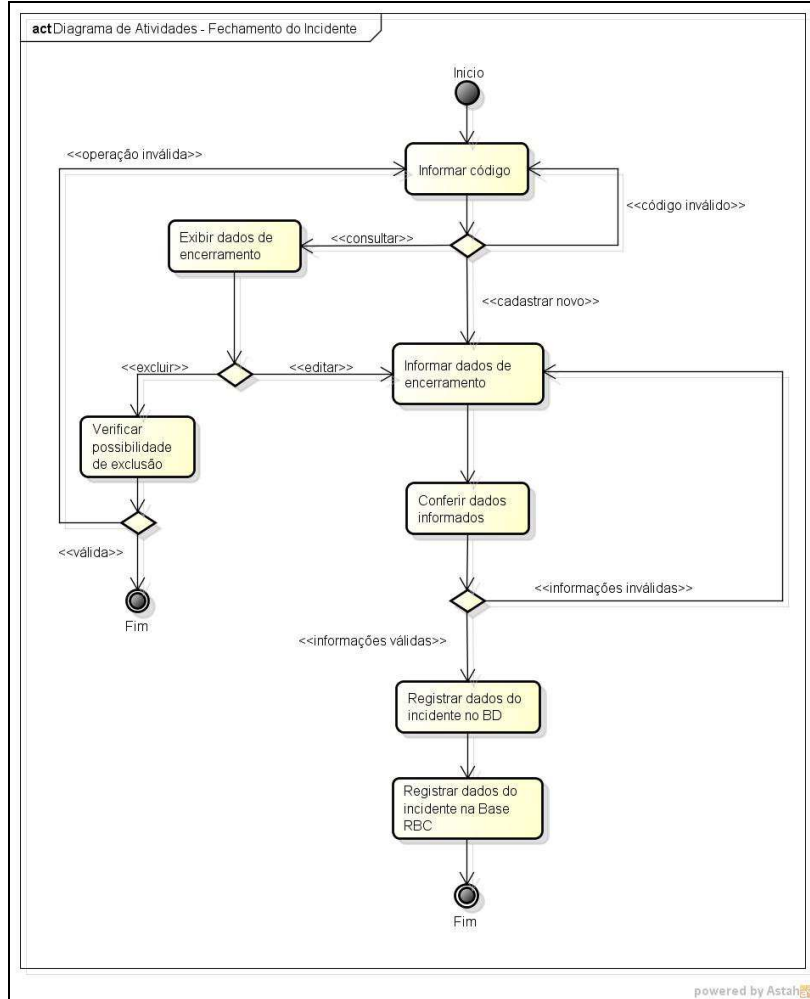
Deve-se destacar que no processo de cadastro de um incidente representado na Figura 7, não é obrigatório a vinculação do incidente a um software ou a um equipamento, pois o sistema visa possibilitar ao técnico desde o cadastramento de um incidente que ocorreu devido a um aterramento, portanto não possui vínculo com nenhum software ou equipamento, até um incidente que ocorreu em um modem que então pode ser vinculado apenas a um equipamento sem um software. No entanto, quando um incidente tem ligação com um software, o técnico deve indicar qual foi o equipamento em que o software estava instalado.

Figura 7 - Diagrama de atividade de Abertura do Incidente



Deve-se ressaltar um detalhe importante na relação entre a Figura 7 e Figura 8 que representam a abertura e o encerramento de um incidente respectivamente, que é a questão de após o término da abertura do incidente, o mesmo é registrado na base de dados, ou seja, no banco de dados. O registro do incidente na base de casos RBC só ocorre após o encerramento do incidente e o registro dos dados no banco.

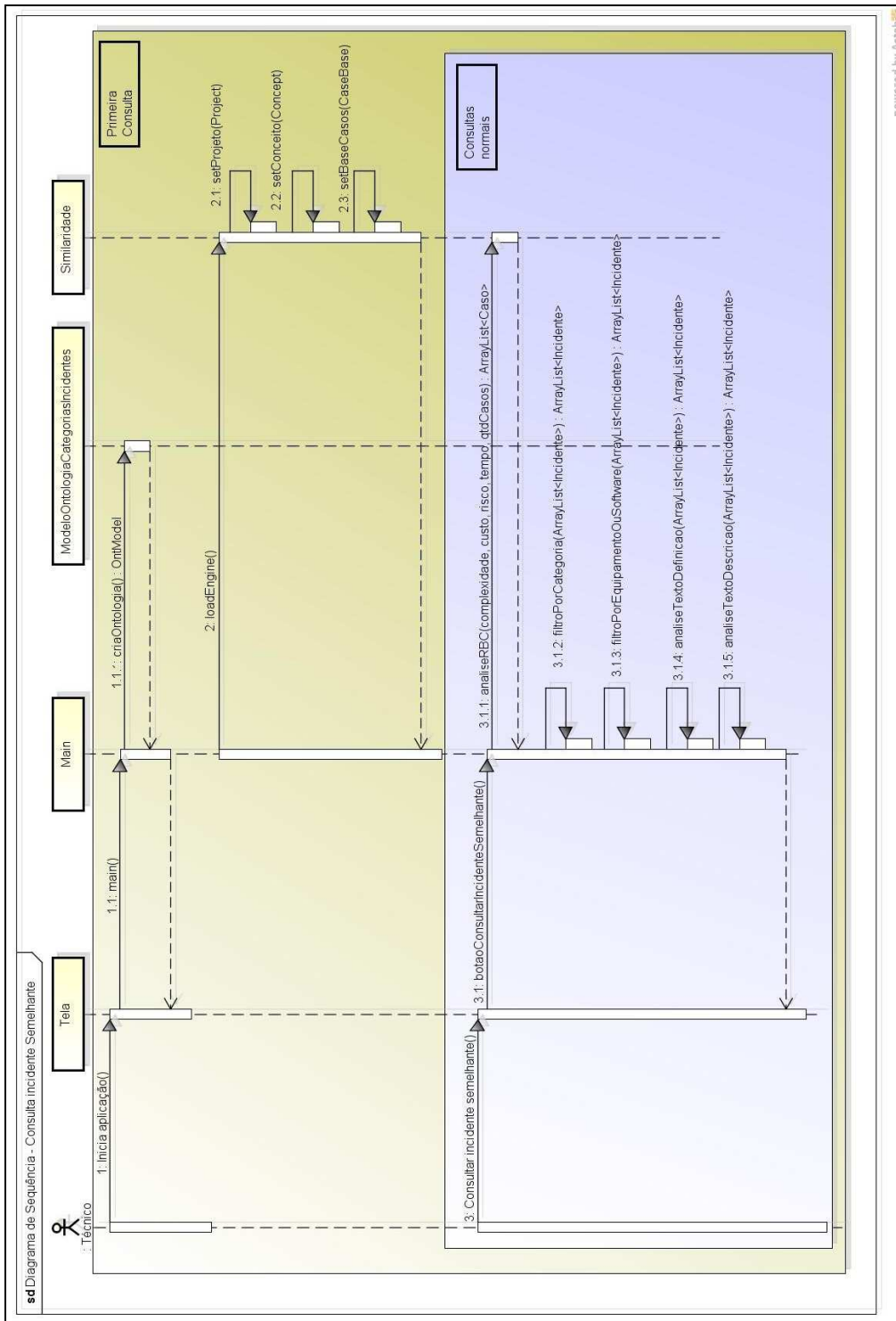
Figura 8 - Diagrama de atividade de Fechamento do Incidente



3.3.4 Diagrama de sequência

O diagrama de sequência da Figura 9, complementa o diagrama de caso de uso do Quadro 4, demonstrando quais são as operações realizadas pelo sistema para retornar a função de consulta a incidentes semelhantes. Deve-se destacar que este diagrama mostra os processos executados em ordem temporal e como a aplicação, se comporta na primeira consulta e nas consultas subsequentes.

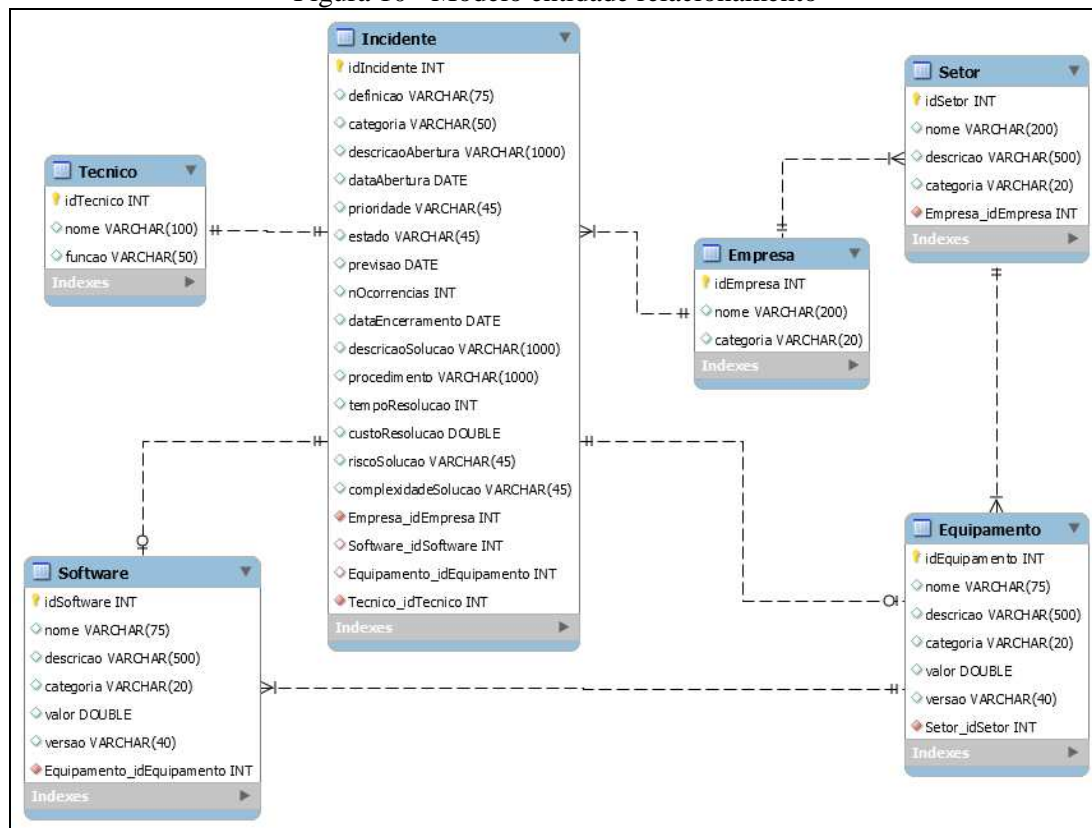
Figura 9 - Diagrama de sequência Consulta Incidentes Semelhantes



3.3.5 MER

O MER, na Figura 10, demonstra a representação da base de dados da aplicação com respectivas tabelas, atributos e relacionamentos.

Figura 10 - Modelo entidade relacionamento



3.3.6 Arquitetura do sistema

O Quadro 6 demonstra a arquitetura do sistema, detalhando quais são os softwares e bibliotecas utilizados, e quais são as suas funções na aplicação. Deve-se ressaltar que a organização da arquitetura do sistema está representada de forma que as camadas mais acima estão mais próximas do usuário.

Quadro 6 - Arquitetura do sistema

Interface com usuário: Cadastros (Incidente, Técnico, Empresa, Setor, Equipamento, Software), Consultas (Incidente, Técnico, Empresa, Setor, Equipamento, Software e Incidentes Semelhantes)			
API de desenvolvimento NetBeans 8.0.2			
MySQL Connector Java 5.1.36	Apache Jena 2.13.0 (Ontologia)	Apache Lucene 5.3.0 (Análise de Texto)	MyCBR 3.1 (RBC)
Banco de Dados MySQL 6.3 (32 bits)		Java 1.8	

3.3.7 Diagrama de classe

A Figura 11 demonstra o diagrama de classe do sistema de gestão de incidentes, primeiramente é interessante ressaltar que o sistema foi desenvolvido com intuito de desenvolver-se um software simples que atendesse as necessidades propostas. Portanto, as funcionalidades do sistema estão concentradas na classe `Main` que também é um

JFrame contendo as funções de cadastro, edição e consulta das estruturas presentes na Figura 10.

Outra questão que deve-se destacar com relação ao diagrama de classe, é que as outras classe do diagrama são para atender as necessidades de implementação das bibliotecas mostradas no Quadro 6. Onde estas não estão diretamente relacionadas as funcionalidades do sistema.

O pacote `Ontologia` possui duas classes, uma sendo a `Categoria` que é uma estrutura para representar os dois nomes de categoria, onde um é de uso interno no sistema e outro para apresentação do usuário. Isto ocorre devido a ontologia representar as categorias logicamente, observou-se a necessidade da criação de uma representação dos nomes das categorias que fosse mais familiar ao técnico.

Outra classe do pacote `Ontologia` é a `ModeloOntologiaCategoriaIncidente` que utiliza a biblioteca Apache Jena para implementar o modelo representado na Figura 12. Esta classe também é responsável por criar a representação da ontologia de categorias de incidente para o usuário, que será utilizado na classe `Main`.

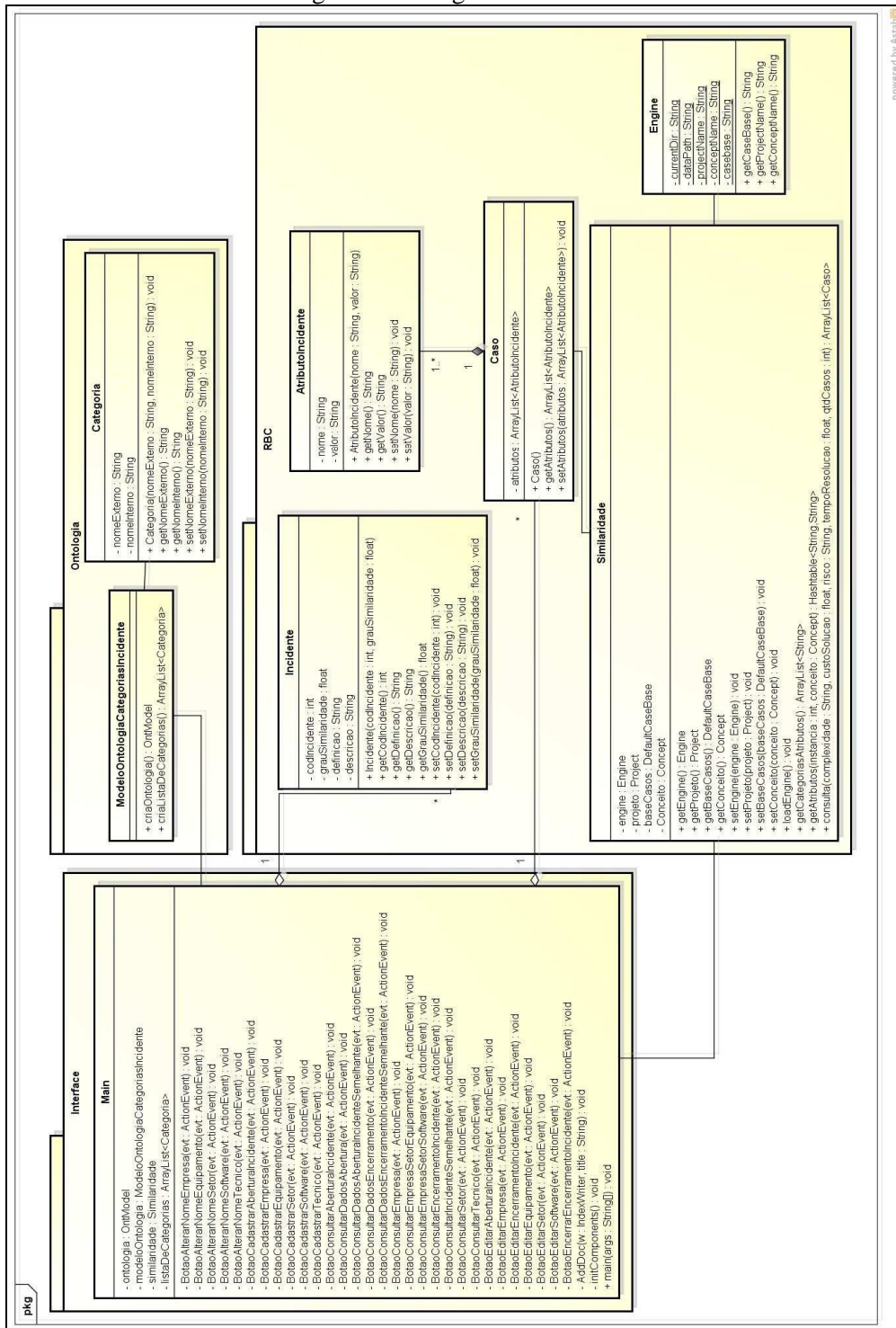
O pacote `RBC` contém as classes necessárias para utilização da biblioteca MyCBR. A classe `AtributoIncidente` basicamente representa um atributo extraído de uma consulta a base de casos, contendo apenas o nome e valor do atributo, e a classe `Caso` possui uma lista dos atributos do caso. Estas duas classes agem em conjunto para representar um caso de um incidente extraído de uma consulta feita na classe `Similaridade`.

A classe `Similaridade` representa os elementos básicos para utilização da biblioteca do MyCBR, o que inclui um objeto da classe `Engine` que é a responsável por criar o projeto e retornar elementos como caminho do projeto, nome do projeto, nome do conceito e o nome da base de casos. Além do objeto da classe `Engine`, a classe `Similaridade` é responsável por carregar o motor do MyCBR com os elementos necessários, que são: o projeto do MyCBR, o conceito de incidente e a base de casos seguindo o conceito já mencionado.

A classe `Incidente` tem o intuito de representar apenas os elementos básicos, que serão utilizados nas operações futuras após a realização da consulta a base de casos. Para que não seja necessário ficar fazendo novamente consultas ao arquivo XML que contém a base de casos, e ao banco de dados para buscar a definição e a descrição dos incidentes analisados, é criada uma estrutura de armazenamento em memória que

contenha estes incidentes, onde os incidentes possam ser incluídos todos de uma vez após verificar quais serão os envolvidos na consulta.

Figura 11 - Diagrama de classe

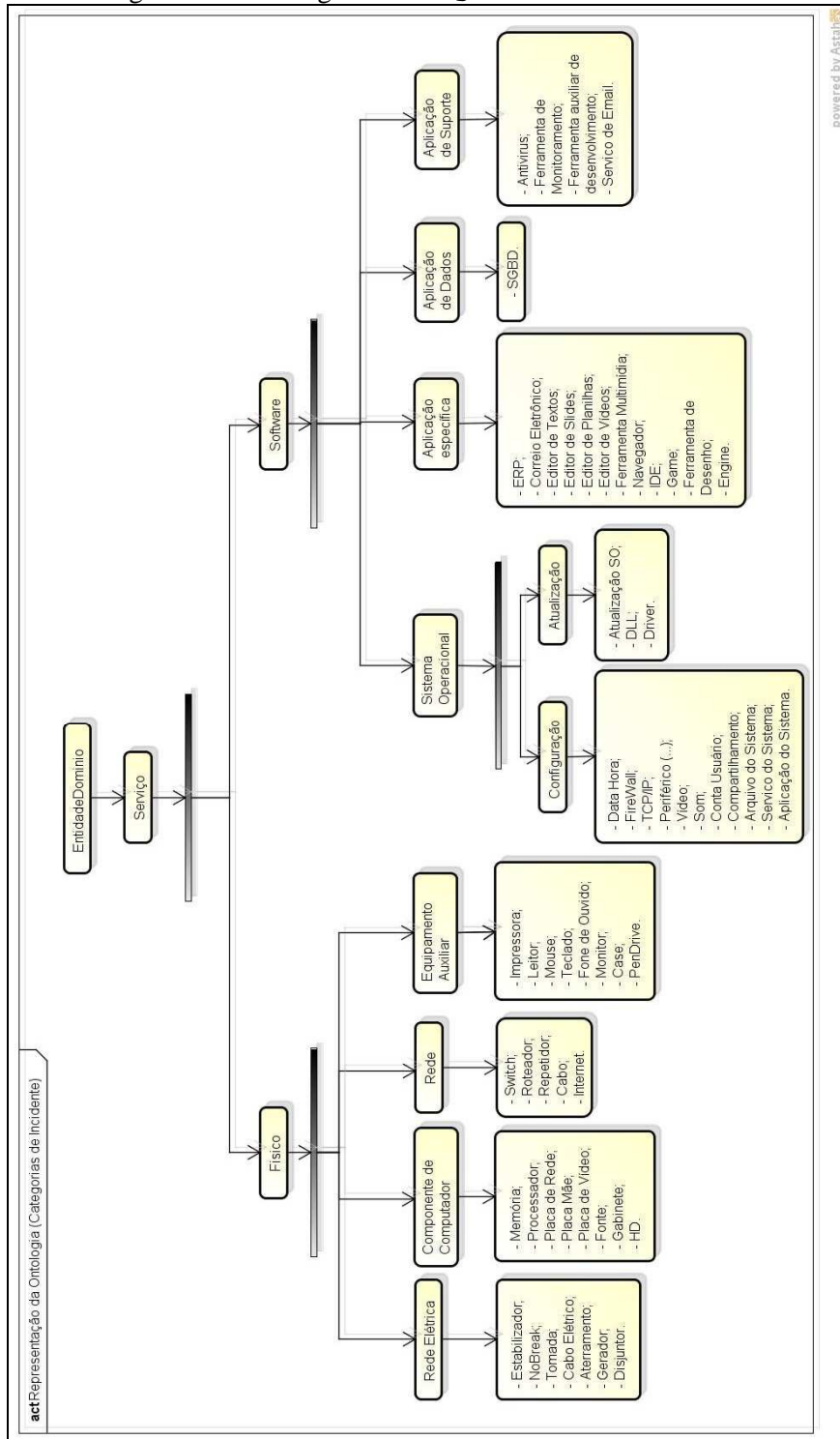


3.3.8 Modelo da ontologia de categorias de incidente

A Figura 12 demonstra o modelo da ontologia de categorias de incidente. Deve-se destacar o modelo que representa as categorias básicas de incidente e que a cada

nível estão categorias específicas dos nodos acima. Outra questão relevante é de que os incidentes são classificados de acordo com os elementos folha da árvore.

Figura 12 - Ontologia de Categorias de Incidente



3.3.9 Modelo RBC

O modelo de RBC, criado para representar o conceito de um incidente, é composto por cinco atributos representados no Quadro 7. Observa-se que todos os

atributos possuem o mesmo peso, com exceção do atributo `Id` que é utilizado para fins de identificação.

Quadro 7 - Atributos do modelo RBC de incidente

Nome dos Atributos	Tipo	Peso	Função de Similaridade
Complexidade	Symbol	1.0	Classificação entre as categorias (Baixa, Média, Alta)
Custo da solução	Float	1.0	Aproximação de valores
Id	Integer	0.0	Sem função de similaridade
Risco da solução	Symbol	1.0	Classificação entre as categorias (Baixo, Médio, Alto)
Tempo p/ resolução	Float	1.0	Aproximação de valores

Para exemplificar uma função de similaridade de classificação entre as categorias, pode-se observar a Figura 13, onde se tem a função de similaridade do atributo `Complexidade`.

Figura 13 - Tabela de símbolos da função de similaridade

Symmetry <input checked="" type="radio"/> symmetric <input type="radio"/> asymmetric		Baixa	Média	Alta
Baixa		1.0	0.0	0.0
Média		0.0	1.0	0.0
Alta		0.0	0.0	1.0

3.3.10 Modelo de consulta de incidentes semelhantes

Para entender o modelo de consulta de incidentes semelhantes, é necessário observar o Quadro 8, que demonstra os atributos de um incidente, incluindo a etapa de abertura e a de encerramento do mesmo. Para realizar esta função que retorna os incidentes, indicando o grau de similaridade entre eles, e o incidente consultado, o sistema realiza uma série de etapas de análise, que são:

- filtro por `Categoria`: onde a consulta restringe o escopo dos resultados apenas há atributos da categoria representada na Figura 12;
- filtro por `Equipamento` ou `Software`: este filtro é opcional e tem a intuição de restringir o escopo dos resultados a um equipamento ou software específico;
- análise de texto: possuindo um peso de 40% no grau de similaridade da função, a análise de texto busca semelhanças entre a `Definição` e a `Descrição` da consulta com os incidentes cadastrados;

- d) análise de RBC: possuindo um peso de 60% no grau de similaridade, realiza a análise de similaridade de acordo com as funções estabelecidas demonstradas no Quadro 7.

Para o desenvolvimento da função de consulta a incidentes semelhantes, optou-se por deixar esta etapa de análise de texto com peso de 40% no grau de similaridade, pois esta etapa visa analisar as definições e descrições dos incidentes e encontrar os casos mais próximos como, por exemplo, no caso do Quadro 8 seriam incidentes com as palavras navegador, FireFox e etc.. No entanto o foco principal da consulta não é simplesmente buscar a melhor solução para o incidente, mas sim buscar uma solução que se adéque aos critérios de custo, tempo, complexidade da solução e risco de aplicação da solução.

Portanto, é atribuído o peso de 60% no grau de similaridade da análise de RBC, pois é essencial para a empresa prestadora saber qual solução se adéqua melhor ao cliente ou mesmo a situação da empresa de TI. Podem-se citar vários exemplos de possíveis casos, como caso onde o cliente precisou de uma solução rápida e os custos eram secundários, então a empresa de TI aplica a solução mais rápida sugerida ou mesmo o caso onde o quadro de funcionários da empresa de TI está sem seus profissionais mais qualificados, a empresa, então, opta por empregar soluções mais simples e demoradas.

Quadro 8 - Exemplo de incidente

Dados de Abertura do Incidente	
Código	102
Definição	O navegador Firefox trava ou não está respondendo
Categoria	Navegador
Descrição	O cliente relatou que o Mozilla Firefox tem travado com frequência
Data de Abertura	14/10/2015
Prioridade	Baixa
Previsão	15/10/2015
Responsável	Jewiton
Nº de Ocorrências	2
Empresa	Banco MAX
Setor	Gerência
Solicitante	José
Equipamento	Computador Gerente 01
Software	Mozilla Firefox
Dados de Encerramento do Incidente	
Data de Encerramento	15/10/2015
Descrição da Solução	Foi realizada a desinstalação do navegador Firefox para a instalação do Google Chrome
Procedimento	Antes de realizar o procedimento realize o backup dos dados do perfil. 1º Instalar o Google Chrome; 2º Importar dados do cliente; 3º Desinstalar o Mozilla Firefox.
Complexidade	Baixa
Risco	Baixo
Tempo	0.15
Custo	50

3.4 IMPLEMENTAÇÃO

Este capítulo descreve detalhes da implementação do sistema como ferramentas e bibliotecas utilizadas, funções implementadas e questões de operacionalidade do software.

3.4.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento do sistema de gestão de incidentes foi utilizado a linguagem de programação Java versão 1.8, com as bibliotecas MyCBR versão 3.1, Apache Jena versão 2.13.0, Apache Lucene versão 5.3.0 e o MyCBR Workbench 3.0.1. Como ambiente de desenvolvimento foi utilizado o NetBeans 8.0.2.

3.4.1.1 Utilização do Apache Jena

O Apache Jena foi utilizado para o desenvolvimento da ontologia que modela as categorias de incidente. A classe responsável por implementar a ontologia é `ModeloOntologiaCategoriasIncidentes`.

Na linha 12 do Quadro 9 é criado um objeto de `OntModel` que receberá o modelo da ontologia. Na linha 14 do Quadro 9 é passado o caminho onde o objeto `ModelMaker` vai criar o arquivo nomeado `CategoriasIncidentes.owl`, e como outro parâmetro é passado `false` para indicar que não se deseja lançar uma exceção, caso o arquivo já exista ele assume o arquivo.

Quadro 9 - Definições do modelo da ontologia

```

1. package Ontologia;
2. import com.hp.hpl.jena.ontology.OntClass;
3. import com.hp.hpl.jena.ontology.OntModel;
4. import com.hp.hpl.jena.ontology.OntModelSpec;
5. import com.hp.hpl.jena.rdf.model.Model;
6. import com.hp.hpl.jena.rdf.model.ModelFactory;
7. import com.hp.hpl.jena.rdf.model.ModelMaker;
8. [...]
9. public class ModeloOntologiaCategoriasIncidentes {
10.     public OntModel CriarOntologia(){
11.         String caminho = "c://"; String arquivo = "file:";
12.         OntModel ontologiaCategoriasIncidentes = null;
13.         try {
14.             ModelMaker maker =
15. ModelFactory.createFileModelMaker(caminho);
16.             Model base =
17. maker.createModel("CategoriasIncidentes.owl", false);
18.             OntModelSpec spec =
19. new OntModelSpec(OntModelSpec.OWL_DL_MEM);
20.             spec.setImportModelMaker(maker);
21.             ontologiaCategoriasIncidentes =
22. ModelFactory.createOntologyModel(spec, base);
23.         } catch (Exception e) {
24.             e.printStackTrace();
25.         }
26.     }
27. }

```

Na linha 16 do Quadro 9 é criado o objeto `OntModelSpec` que recebe como parâmetro o formato em que a ontologia deve ser criada, que no caso foi `OWL_DL_MEM`, referindo-se a sub-linguagem OWL DL e, a mesma, é apenas armazenada em memória. Por fim na linha 18 do Quadro 9 é criado o modelo da ontologia, que recebe como parâmetros o formato e o modelo a ser criado.

Na sequência tem-se o Quadro 10, que representa uma continuação da implementação da classe `ModeloOntologiaCategoriasIncidentes` onde, após a criação do modelo, é iniciado a criação dos objetos `OntClass` que são utilizados para criar os nodos de uma ontologia. No Quadro 10 pode-se observar na linha 26 e 27 a criação dos objetos `entidadeDominio` e `incidente` respectivamente, na linha 28 do

Quadro 10 o objeto incidente é adicionado como sub-classe do objeto entidadeDominio.

Entre as linhas 30 e 33 do Quadro 10 tem-se a criação dos objetos fisico e software, e a adição dos mesmos como sub-classes de incidente. O restante do método CriarOntologia() segue a mesma lógica adicionando as sub-classes específicas de cada OntClass reproduzindo a estrutura representada na Figura 12.

Quadro 10 - Trecho da estrutura da ontologia

```

22. String nomeOntologia = arquivo+caminho+"CategoriasIncidentes.owl";
23. String iniNomeElemento = nomeOntologia + "#";
24. ontologiaCategoriasIncidentes.begin();
25.
26. OntClass entidadeDominio =
    ontologiaCategoriasIncidentes.createClass(iniNomeElemento +
        "EntidadeDominio");
27. OntClass incidente =
    ontologiaCategoriasIncidentes.createClass(iniNomeElemento +
        "Incidente");
28. entidadeDominio.addSubClass(incidente);
29.
30. OntClass fisico =
    ontologiaCategoriasIncidentes.createClass(iniNomeElemento +
        "Fisico");
31. OntClass software =
    ontologiaCategoriasIncidentes.createClass(iniNomeElemento +
        "Software");
32. incidente.addSubClass(fisico);
33. incidente.addSubClass(software);
34. [...]

```

O Quadro 11 ilustra a utilização do Jena na classe Main, onde a ontologia é de fato criada através da chamada do método CriarOntologia() na linha 5 do Quadro 11. Na linha 6 do Quadro 11 tem-se a chamada do método criaListaDeCategorias() que transforma a estrutura das categorias representadas pela ontologia em um ArrayList para preencher os componentes de tela que mostram as categorias.

Quadro 11 - Modelo da ontologia sendo carregado na interface

```

1. import com.hp.hpl.jena.ontology.OntModel;
2. [...]
3. OntModel ontologia;
4. [...]
5. ontologia = modeloOntologia.CriarOntologia();
6. listaDeCategorias = modeloOntologia.criaListaDeCategorias();
7. for (int i = 0; i < listaDeCategorias.size(); i++) {
8.     CaixaSelecaoCategoria.addItem(
        listaDeCategorias.get(i).getNomeExterno());
9.     CaixaSelecaoCategoriaIncidenteSemelhante.addItem(
        listaDeCategorias.get(i).getNomeExterno());
10. }
11. [...]

```

3.4.1.2 Utilização da biblioteca MyCBR e do MyCBR Workbench

O MyCBR Workbench foi utilizado para o desenvolvimento do modelo RBC que representa o conceito de incidente mencionado na seção 3.3.9. A aplicação segue a

definição do modelo e gerencia a base de casos que são registrados em um arquivo XML.

No Quadro 12 é demonstrado o principal método da classe `Engine` que o `createProjectFromPRJ()` utilizado na classe `Similaridade` no método `loadengine()` conforme pode-se observar no Quadro 13. O método `loadengine()` é responsável por criar o objeto `Engine`, acima citado, e atribuir os elementos instanciados nele, que são o projeto, base de casos e conceito de incidente.

Quadro 12 – Método de criação do projeto do MyCBR

```
1. package RBC;
2. import de.dfki.mycbr.core.Project;
3.
4. public class Engine {
5.     [...]
6.     public Project createProjectFromPRJ(){
7.         Project project = null;
8.         try{
9.             project = new Project (data_path + projectName);
10.            while (project.isImporting()){
11.                Thread.sleep(1000);
12.            }
13.        }
14.        catch(Exception ex){
15.            ex.printStackTrace();
16.        }
17.        return project;
18.    }
19. }
```

Quadro 13 – Método de carregamento das estruturas do MyCBR

```

1. package RBC;
2. [...]
3. import de.dfki.mycbr.core.DefaultCaseBase;
4. import de.dfki.mycbr.core.Project;
5. import de.dfki.mycbr.core.casebase.Instance;
6. import de.dfki.mycbr.core.model.AttributeDesc;
7. import de.dfki.mycbr.core.model.Concept;
8. import de.dfki.mycbr.core.model.FloatDesc;
9. import de.dfki.mycbr.core.model.SymbolDesc;
10. import de.dfki.mycbr.core.retrieval.Retrieval;
11. import de.dfki.mycbr.core.retrieval.Retrieval.RetrievalMethod;
12. import de.dfki.mycbr.core.similarity.Similarity;
13. import de.dfki.mycbr.util.Pair;
14. [...]
15. public class Similaridade {
16.     private Engine engine;
17.     private Project projeto;
18.     private DefaultCaseBase baseCasos;
19.     private Concept conceito;
20.
21.     public void loadengine () {
22.         setEngine(new Engine());
23.         setProjeto(getEngine().createProjectFromPRJ());
24.         System.out.println(getProjeto());
25.         setBaseCasos(
26.             (DefaultCaseBase)getProjeto().getCaseBases().get(
27.                 getEngine().getCaseBase());
28.             setConceito(getProjeto().getConceptByID(
29.                 getEngine().getConceptName()));
30.         }
31.     }
32. }
33. [...]

```

No Quadro 14 pode-se observar o método de consulta implementado na classe *similaridade*. Este método basicamente recebe como parâmetros os dados de um incidente, sendo eles: complexidade, custo, risco, tempo e quantidade de casos. Para, a partir disso, realizar uma consulta a base de casos retornando os incidentes mais semelhantes limitado-se pela quantidade de incidentes retornados a quantidade informada no parâmetros *qtdCasos*.

Na linha 31 do Quadro 14, pode-se observar a criação do objeto *Retrieval* da biblioteca do MyCBR utilizado para realizar consultas, que recebe como parâmetros o conceito e a base de casos. Logo abaixo, na linha 33 do Quadro 14, é criado um objeto de *Instance* que recebe o caso retornado pelo método *getQueryInstance()*.

Na linha 32 do Quadro 14, é definido o método de retorno, o *RETRIEVE_SORTED*, que retorna os casos ordenados por ordem de inserção na base de casos. Na linha 33 do Quadro 14, o objeto *Instance* se torna a instância da consulta, ou seja, serão adicionados os atributos dos casos e comparados aos atributos dessa instância.

Na linha 34 do Quadro 14, é criado um objeto *SymbolDesc* que recebe a descrição completa das regras aplicadas ao atributo denominado “Complexidade” no conceito de incidente. Com isso, na linha 35 do Quadro 14, é utilizado o método

`addAttribute()` que recebe como parâmetros a descrição das regras de complexidade e o atributo passado como parâmetro `complexidade` do método `consulta`. Basicamente o que foi feito, é informar a regra de análise de similaridade do atributo a ser comparado, regra a qual foi definida previamente no conceito de incidente e o valor externo a ser comparado com a base de caso.

Quadro 14 - Método consulta RBC

```

28. [...]
29. public ArrayList<Caso> consulta(String complexidade, float
    custoSolucao, String risco, float tempoResolucao, int qtdCasos){
30.     ArrayList<Caso> incidentes = new ArrayList<Caso>();
31.     Retrieval retorno = new Retrieval(getConceito(),
    getBaseCasos());
32.     retorno.setRetrievalMethod(RetrievalMethod.RETRIEVE_SORTED);
33.     Instance query = retorno.getQueryInstance();
34.     SymbolDesc complexidadeDesc = (SymbolDesc)
    getConceito().getAllAttributeDescs().get("Complexidade");
35.     query.addAttribute(
    complexidadeDesc,complexidadeDesc.getAttribute(complexidade));
36.     [...]
37.     retorno.start();
38.     List <Pair<Instance, Similarity>> resultado =
    retorno.getResult();
39.     if(resultado.size()>0){
40.         [...]
41.         if(qtdCasos>=getBaseCasos().getCases().size()){
42.             qtdCasos = getBaseCasos().getCases().size();
43.         }
44.         for(int i = 0; i < qtdCasos; i++){
45.             [...]
46.             for (String s : details.split(",")){
47.                 [...]
48.                 incidente.getAtributos().add(atributo);
49.             }
50.             incidentes.add(incidente);
51.         }
52.     }
53.     return incidentes;
54. }

```

Com isso esse processo se repete para todos os atributos da consulta, e por fim na linha 37 do Quadro 14, é iniciada a consulta de fato e o resultado dela é colocado no objeto `List<Pair<Instance, Similarity>>` denominado `resultado`, que recebe o resultado através do método `getResult()`. Após retornar a consulta, os incidentes são inseridos no `ArrayList<Caso>` para retornar os incidentes mais semelhantes ao consultado por ordem de semelhança. Outra questão importante para se ressaltar nesse trecho de código, é a linha 41 do Quadro 14, que contém um comando condicional `if` para alterar a quantidade de casos retornados para o tamanho da base de casos, ou seja, assim a quantidade de casos retornados é de todos os casos avaliados, pois os filtros serão realizados no método de consulta de incidentes semelhantes, demonstrado na seção 3.4.1.4.

Os incidentes são incluídos na base de casos do MyCBR quando os mesmos são encerrados. Pois, a proposta do sistema é retornar a melhor solução possível para um incidente, levando em consideração os elementos já citados na seção 3.3.10.

No Quadro 15 é demonstrado a utilização da biblioteca do MyCBR na classe Main, através da classe Similaridade e do uso do objeto Instance que é utilizado para criar uma instância de um caso que recebe seus respectivos atributos e é inserido na base de caso. Outro elemento interessante a se destacar com relação a biblioteca, é que o caso só é salvo na base de casos de fato após a chamado do método `save()`.

Quadro 15 - Cadastramento dos incidentes encerrados na base de casos MyCBR

```

1. [...]
2. import de.dfki.mycbr.core.casebase.Instance;
3. [...]
4. similaridade = new Similaridade();
5. similaridade.loadengine();
6. [...]
7. private void BotaoEncerrarEncerramentoIncidenteActionPerformed
   (java.awt.event.ActionEvent evt) {
8. [...]
9.     Instance caso = new Instance(similaridade.getConceito(),
   "Caso" + codIncidente);
10.     try {
11.         caso.addAttribute(
   similaridade.getConceito().getAttributeDesc("ID"),
   codIncidente);
12.         caso.addAttribute(
   similaridade.getConceito().getAttributeDesc("Risco"),
   risco);
13.         caso.addAttribute(
   similaridade.getConceito().getAttributeDesc("Complexidade"),
   complexidade);
14.         caso.addAttribute(
   similaridade.getConceito().getAttributeDesc("CustoDaSolucao"),
   (float)custo);
15.         caso.addAttribute(
   similaridade.getConceito().getAttributeDesc("TempoResolucao"),
   (float)tempo);
16.     } catch (ParseException ex) {
17.         Logger.getLogger(Main.class.getName()).log(Level.SEVERE,
   null, ex);
18.     }
19.     similaridade.getBaseCasos().addCase(caso);
20.     similaridade.getProjeto().save();
21. [...]
22. }
23. [...]

```

3.4.1.3 Utilização do Apache Lucene

A biblioteca do Apache Lucene foi utilizada neste trabalho para fazer a operação de análise de texto, avaliando a similaridade entre a definição e a descrição de vários incidentes, com um caso descrito e informado pelo técnico. No entanto, como a maior parte da utilização da biblioteca foi dentro da funcionalidade de consulta aos incidentes

semelhantes, exemplificada na seção 3.4.1.4, optou-se nesta seção por se destacar as classe utilizadas da biblioteca e o método `addDoc()` demonstrados no Quadro 16.

Deve-se destacar dentre as classes utilizadas da biblioteca do Apache Lucene, a `StandardAnalyzer`, que é o analisador padrão da ferramenta como o nome indica, utiliza o filtros de `StandardTokenizer`, `StandardFilter`, `LowerCaseFilter` e `StopFilter`. Outras classes fundamentais a se destacar, são as classes responsáveis por gerenciar a busca por documentos no índice criado, são elas: o `IndexWriter`, que é responsável por criar e gerenciar o índice, a classe `IndexReader` e a classe `IndexSearcher` são utilizadas para leitura e busca no índice respectivamente.

A classe `ScoreDoc` é utilizada como ponteiro para um documento do índice, pois contém a posição do documento e o grau de similaridade do documento. A classe `TopScoreDocCollector` que é utilizada para recolher os documentos por ordem de pontuação. Por fim, outra questão importante na classe `Main`, é `QueryParser`, responsável pela análise das expressões inseridas no índice. O método `addDoc()` é utilizado para adicionar um documento ao índice de um objeto `IndexWriter`, caso da implementação desse trabalho, ele é utilizado para criação dos índices de definição e descrição de incidentes.

Quadro 16 - Bibliotecas utilizadas do Apache Lucene e método de adição de documentos

```

1. [...]
2. import org.apache.lucene.analysis.standard.StandardAnalyzer;
3. import org.apache.lucene.document.Document;
4. import org.apache.lucene.document.Field;
5. import org.apache.lucene.document.TextField;
6. import org.apache.lucene.index.DirectoryReader;
7. import org.apache.lucene.index.IndexReader;
8. import org.apache.lucene.index.IndexWriter;
9. import org.apache.lucene.index.IndexWriterConfig;
10. import org.apache.lucene.queryparser.classic.QueryParser;
11. import org.apache.lucene.search.IndexSearcher;
12. import org.apache.lucene.search.Query;
13. import org.apache.lucene.search.ScoreDoc;
14. import org.apache.lucene.search.TopScoreDocCollector;
15. import org.apache.lucene.store.Directory;
16. import org.apache.lucene.store.RAMDirectory;
17. [...]
18. private void addDoc(IndexWriter w, String title) throws
    IOException {
19.     Document doc = new Document();
20.     doc.add(new TextField("title", title, Field.Store.YES));
21.
22.     w.addDocument(doc);
23. }
24. [...]

```

3.4.1.4 Implementação da consulta de incidentes semelhantes

A principal funcionalidade deste sistema de gestão de incidentes, é a de consulta de incidentes semelhantes que está na classe `Main`, no método `BotaoConsultarIncidentesSemelhantes()`. Primeiramente, pode-se observar no Quadro 17 na linha 4, a chamada do método `consulta()` que obtém como retorno uma lista de objetos da classe `Caso`. Na sequência, na linha 5 do Quadro 17, é feito um comando de repetição `for` para transferir os casos para a tabela incidente, extraíndo apenas os atributos `id` e `grauDeSimilaridade`, já convertendo o grau de similaridade na escala de 0% à 100% conforme mencionado na seção 3.3.7.

Para fins de simplificação da apresentação do código e melhor compreensão das técnicas utilizadas, optou-se por apenas comentar a aplicação dos filtros feitos através de operações com `select` do banco de dados `MySQL`, que não são o foco do trabalho. Sendo elas, um filtro por categoria de incidente representado pelo comentário na linha 14 do Quadro 17, e outro filtro opcional por nome do software ou por descrição do equipamento conforme o código indica na linha 15 do Quadro 17.

Quadro 17 - Etapa de análise de similaridade e aplicação de filtros na consulta de incidentes semelhantes

```

1. private void BotaoConsultarIncidentesSemelhantesActionPerformed(
2.     java.awt.event.ActionEvent evt) {
3.     [...]
4.     //Análise de similaridade através da lib. MyCBR
5.     casos = similaridade.consulta(complexidade, custo, risco,
6.     tempo, 100);
7.     for(int i = 0; i < casos.size(); i++){
8.         incidentesAuxiliar.add(
9.             new Incidente(
10.                Integer.parseInt(
11.                    casos.get(i).getAtributos().get(0).getValor()),
12.                (Float.parseFloat(
13.                    casos.get(i).getAtributos().get(5).getValor()) * 100)
14.                )
15.            );
16.        }
17.    }
18.    [...]
19.    //Filtro por categoria através de select
20.    //Filtro opcional por Nome do Software ou Descrição do
21.    Equipamento
22.    [...]

```

No Quadro 18, tem-se a continuação da implementação do método `BotaoConsultarIncidentesSemelhantes()`, mais especificamente na etapa de análise de texto do incidente. Na linha 23 do Quadro 18, pode-se observar um comando condicional `if` que é responsável por fazer o filtro da seleção entre análise apenas da definição, da descrição ou de ambos, pois quando apenas o filtro de definição ou descrição é utilizado o peso dessa análise no grau de similaridade total do incidente é de 40%, já se o filtro analisar ambos, cada um recebe um peso de 20% no grau de similaridade total do incidente.

Na linha 25 do Quadro 18, observa-se um comando de repetição `for` para realizar a configuração do peso no grau de similaridade total do incidente para 60%, pois, até então, na execução do código foi realizada apenas a análise de similaridade RBC. Essa configuração foi feita só nessa etapa da codificação, visto que se o usuário não preencher os campos de definição e descrição o sistema considerará que deve ser realizada apenas a análise de similaridade RBC.

Entre as linhas 30 e 32 do Quadro 18, é feito a criação dos objetos `StandardAnalyzer`, `Directory`, `IndexWriterConfig`, basicamente são instanciados um objeto do analisador da ferramenta, um objeto que comporta o `IndexWriter` e um objeto que possui a configuração do `IndexWriter` respectivamente. Com isso na linha 34 do Quadro 18, é criado de fato uma instância do objeto `IndexWriter` que tem como parâmetros o diretório e as configuração do índice.

Na linha 35 do Quadro 18, deve-se observar a estrutura de repetição `for` que percorre a lista de incidentes aplicando o método `addDoc()`, que foi exemplificado no final da seção 3.4.1.3, que no caso está adicionando a definição do incidente específico na instância do objeto `IndexWriter` denominado `def`. Por fim, na linha 38 do Quadro 18, é utilizado o método `close()` da classe `IndexWriter` para que o índice possa ser acessado pela classe `IndexReader`.

Na linha 39 do Quadro 18, é instanciado um objeto denominado `q` da classe `QueryParser` onde é passado a instância do objeto `StandardAnalyzer` e a `string`, contendo a definição do incidente que o usuário inseriu para realizar esta consulta. Na linha 42 do Quadro 18, tem-se a criação de uma instância do objeto `IndexReader` que recebe como parâmetro uma instância do objeto `Directory` contendo o índice das definições dos incidentes já cadastrados, e esta instância do objeto `IndexReader` é passado como parâmetro para a criação de uma instância do objeto `IndexSearcher` denominado `searcherDef`, na linha 43 do Quadro 18.

Na linha 44 do Quadro 18, é criada uma instância do objeto da classe `TopScoreDocCollector` denominado `collectorDef`, que, como o nome da classe sugere, coleta os documentos do índice por ordem de melhor pontuação e recebe como parâmetro a quantidade de documentos que ele deve retornar por acesso ao índice, ou seja, todos os itens do índice serão retornados na busca, mais eles serão retornados de 10 em 10.

Na linha 45 do Quadro 18, ocorre a execução do método `search` da instância `searcherDef` que recebe como parâmetros o valor da `query`, no caso, a definição que o usuário inseriu para consulta e os documentos contendo as definições dos incidentes já cadastrados para que sejam comparados. Na linha 46 do Quadro 18, é passado para o `array` de `ScoreDoc` que recebe os incidentes comparados que tinham alguma semelhança com a definição do usuário.

Na linha 48 do Quadro 18, é feito um comando de repetição `for` que percorre o `array` de `ScoreDoc` contendo os documentos semelhantes, e através da identificação pelo índice que fica no atributo `doc` do objeto `ScoreDoc`, é realizada a configuração do grau de similaridade do incidente, onde é adicionado a porcentagem de similaridade referente a análise de texto que está no atributo `score` do objeto `ScoreDoc`, que, no caso, é uma consulta que envolve a definição e a descrição do incidente, cada um tem peso de 40% no grau de similaridade do incidente. Por fim, o mesmo processo descrito

se repete para a descrição do incidente informado pelo usuário. Outra questão relevante, é que logo abaixo são representados por comentários nas linhas 56 e 58 do Quadro 18, a análise de texto apenas da definição e apenas da descrição do incidente.

Quadro 18 - Etapa de análise de texto na consulta de incidentes semelhantes

```

17. //Análise de palavras chave através da lib. Lucene
18. try {
19.     [...]
20.     //query
21.     Query q = null;
22.     //Analise da descricao e da definicao do incidente
23.     if(!queryStrDefinicao.equals("") &&
!queryStrDescricao.equals("")){
24.         //realizar configuração de peso para 60% da similaridade
MyCBR
25.         for (int i = 0; i < incidentesPrincipal.size(); i++) {
26.             incidentesPrincipal.get(i).setGrauSimilaridade(
27.                 incidentesPrincipal.get(i).getGrauSimilaridade() *
0.6f);
28.         }
29.         //objetos base
30.         StandardAnalyzer analyzerDef = new StandardAnalyzer();
31.         Directory indexDef = new RAMDirectory();
32.         IndexWriterConfig configDef = new
IndexWriterConfig(analyzerDef);
33.         //cria indice
34.         IndexWriter def = new IndexWriter(indexDef, configDef);
35.         for (int i = 0; i < incidentesPrincipal.size(); i++) {
36.             addDoc(def,
incidentesPrincipal.get(i).getDefinicao());
37.         }
38.         def.close();
39.         q = new QueryParser("title",
analyzerDef).parse(queryStrDefinicao);
40.         //busca
41.         int hitsPerPageDef = 10;
42.         IndexReader readerDef = DirectoryReader.open(indexDef);
43.         IndexSearcher searcherDef = new
IndexSearcher(readerDef);
44.         TopScoreDocCollector collectorDef =
TopScoreDocCollector.create(hitsPerPageDef);
45.         searcherDef.search(q, collectorDef);
46.         ScoreDoc[] hitsDef = collectorDef.topDocs().scoreDocs;
47.         //saida
48.         for (int i = 0; i < hitsDef.length; i++) {
49.             incidentesPrincipal.get(hitsDef[i].doc).setGrauSimilaridade(
50. (incidentesPrincipal.get( hitsDef[i].doc).getGrauSimilaridade() +
51. ((hitsDef[i].score*100) * 0.2f));
52.         }
53.         readerDef.close();
54.         //Repete o mesmo processo para descrição do incidente
55.         [...]
56.         //Analise somente da definição do incidente
57.         [...]
58.         //Analise somente da descrição do incidente
59.         [...]
60. }

```

3.4.2 Operacionalidade da implementação

Nesta seção é apresentada a operacionalidade do sistema de gestão de incidentes desenvolvido neste trabalho. Através de exemplos práticos com imagens da utilização da aplicação, demonstrando as principais funcionalidades do sistema. Dentre as

principais funcionalidades, está a de consulta a incidentes semelhantes que é exemplificada nesta seção detalhadamente, para assim, esclarecer os recursos da função.

Primeiramente, tem-se a tela inicial da aplicação demonstrada na Figura 14, que é a tela básica da maioria dos sistemas de gestão de incidentes, onde o técnico costuma observar os incidentes e seus respectivos status. A barra superior denominada Menu Principal contém os menus de cadastros, consultas, configurações e ajuda.

O menu Cadastros possui os itens de Abertura Incidente, Encerramento Incidente, Técnico, Empresa, Setor, Equipamento e Software que correspondem aos cadastros descritos na seção 3.2.

O menu Consultas, por sua vez, contém os itens Incidentes e Incidentes Semelhantes sendo que o primeiro item leva para a tela inicial do sistema e o segundo corresponde a funcionalidade específica. O menu Ajuda possui apenas informações sobre o sistema. Deve-se ressaltar que em algumas das telas apresentadas nesta seção, mais especificamente nas telas que demonstram exemplos de consultas a incidentes semelhantes, aparecem o menu Configurar que possuía os itens com elementos que acabaram não sendo desenvolvidos neste trabalho que são detalhados na seção 4.1.

Figura 14 - Tela inicial da aplicação

Cód.	Categoria	Prioridade	Estado	Abert.	Empresa	Nm. Téc.	Enc.
1	Aplicação específicas(Correio Eletrônico)	Critico (1 hora)	Resolvido	05/09/2015	Metalúrgica Ele...	Jewiton	07/09/2015
2	Configuração(TCP/IP)	Alto (8 horas)	Resolvido	12/09/2015	Metalúrgica Ele...	Jewiton	12/09/2015
3	Configuração(FireWall)	Baixo (48 horas)	Resolvido	12/09/2015	Metalúrgica Ele...	Jewiton	13/09/2015
4	Aplicação específicas(Navegador)	Planejado	Resolvido	15/09/2015	Metalúrgica Ele...	Jewiton	17/09/2015
5	Rede Elétrica(Tomada)	Médio (24 horas)	Resolvido	12/09/2015	Metalúrgica Ele...	Jewiton	13/09/2015
6	Componente Computador(HD)	Baixo (48 horas)	Resolvido	12/09/2015	Metalúrgica Ele...	Jewiton	14/09/2015
7	Aplicação específicas(Navegador)	Baixo (48 horas)	Resolvido	15/09/2015	Metalúrgica Ele...	Jewiton	17/09/2015
8	Configuração(Aplicações SO)	Alto (8 horas)	Resolvido	25/09/2015	Metalúrgica Ele...	Jewiton	26/09/2015
9	Aplicação específicas(Navegador)	Alto (8 horas)	Resolvido	02/10/2015	Metalúrgica Ele...	Jewiton	02/10/2015
10	Aplicação específicas(Navegador)	Baixo (48 horas)	Resolvido	02/10/2015	Metalúrgica Ele...	Jewiton	03/10/2015
11	Aplicação específicas(Navegador)	Alto (8 horas)	Resolvido	03/10/2015	Metalúrgica Ele...	Jewiton	05/10/2015
12	Aplicação específicas(Navegador)	Planejado	Resolvido	03/10/2015	Metalúrgica Ele...	Jewiton	06/10/2015
13	Aplicação específicas(Navegador)	Baixo (48 horas)	Resolvido	07/10/2015	Metalúrgica Ele...	Jewiton	08/10/2015
14	Aplicação específicas(Navegador)	Médio (24 horas)	Resolvido	07/10/2015	Metalúrgica Ele...	Jewiton	08/10/2015
15	Aplicação específicas(Navegador)	Baixo (48 horas)	Resolvido	07/08/2015	Metalúrgica Ele...	Jewiton	09/08/2015
17	Atualização(SO)	Planejado	Resolvido	07/10/2015	Metalúrgica Ele...	Jewiton	10/10/2015
18	Atualização(SO)	Baixo (48 horas)	Resolvido	07/10/2015	Metalúrgica Ele...	Jewiton	08/10/2015
19	Configuração(Serviços SO)	Médio (24 horas)	Resolvido	08/10/2015	Metalúrgica Ele...	Jewiton	10/10/2015
20	Configuração(Serviços SO)	Planejado	Resolvido	08/10/2015	Metalúrgica Ele...	Jewiton	09/10/2015
21	Configuração(Serviços SO)	Alto (8 horas)	Resolvido	08/10/2015	Metalúrgica Ele...	Jewiton	08/10/2015
22	Atualização(SO)	Alto (8 horas)	Resolvido	08/10/2015	Metalúrgica Ele...	Jewiton	08/10/2015
23	Atualização(SO)	Planejado	Resolvido	09/10/2015	Metalúrgica Ele...	Jewiton	11/10/2015
24	Configuração(Serviços SO)	Baixo (48 horas)	Resolvido	09/10/2015	Metalúrgica Ele...	Jewiton	10/10/2015
25	Atualização(SO)	Planejado	Resolvido	09/10/2015	Metalúrgica Ele...	Jewiton	11/10/2015
26	Configuração(Serviços SO)	Planejado	Resolvido	09/10/2015	Metalúrgica Ele...	Jewiton	13/10/2015
27	Atualização(SO)	Alto (8 horas)	Resolvido	09/10/2015	Metalúrgica Ele...	Jewiton	09/10/2015
28	Configuração(Serviços SO)	Planejado	Resolvido	10/10/2015	Metalúrgica Ele...	Jewiton	13/10/2015

Na Figura 14, abaixo do Menu Principal, pode-se observar o campo Código que recebe o código do incidente digitado pelo usuário, que deseja saber os dados de abertura ou encerramento, através da interação com os botões Consultar Dados de

Abertura e Consultar Dados de Encerramento respectivamente. Logo a esquerda tem-se o botão Atualizar, que realiza a atualização da tabela denominada Tabela de Incidentes.

A Tabela de Incidentes contém as colunas Cód. que possui o código dos incidentes, Categoria que tem a categoria do incidente, Prioridade que contém o nível de prioridade do incidente, Estado que tem o status do incidente, Abert. que possui a data de abertura do incidente, Empresa que contém o nome da empresa, Nm. Téc. que contém o nome do técnico responsável pelo incidente e, por fim, Enc. que possui a data de encerramento do incidente. O restante dos dados que aparecem na Figura 14 dentro da Tabela de Incidentes correspondem a incidentes cadastrados seguindo a formação mencionada acima neste parágrafo.

A Figura 15, demonstra um exemplo de abertura de um incidente. Onde são informados o código do incidente, a definição feita pelo técnico, a categoria que o incidente se encaixa, a descrição realizada pelo cliente, data de abertura, o nível de prioridade, o nome do técnico responsável e a quantidade de ocorrências do incidente.

Concluído o preenchimento dos dados de abertura do incidente, são preenchidos os dados da empresa, que contém o nome da empresa, o nome do setor e o solicitante. Então, são informados os dados do equipamento e dados do software que são apenas seus respectivos nomes.

Deve-se destacar que para realizar a referência a um técnico, empresa, setor, equipamento ou software este deve estar previamente cadastrado. Abaixo dos campos da Figura 15, tem os botões de cadastro, edição e exclusão. Na parte superior da tela da Figura 15, ao lado de Código pode-se observar o botão de consulta.

Além disso, é importante destacar que o sistema possui validações para garantir que o técnico não referencie campos obrigatórios inexistentes ou deixe os mesmos vazios. Estes campos são: Definição, Data de Abertura, Responsável, Empresa e Setor. O campo Código também possui validação para não permitir que o campo fique vazio e que o código não se repita. Ressaltando que todas as validações são feitas após a interação com os botões de Cadastrar e Editar.

Figura 15 - Tela de Abertura do Incidente

Sistema de Gestão de Incidentes

Cadastrados Consultas Ajuda

Dados de Abertura do Incidente

Código: 16

Definição: O navegador Firefox esta travando frequentemente e tem apresentado lentidão

Categoria: Aplicação específicas(Navegador)

Descrição: O Cliente relatou que o navegador Mozilla Firefox tem travado com frequência, e tem apresentado lentidão na utilização de funções do navegador e o plugin do Adobe Reader

Data de Abertura: 08/09/2015

Prioridade: Baixo (48 horas)

Previsão: 11/09/2015

Responsável: Jewiton

Nº de Ocorrências: 5

Dados Empresa **Dados Equipamento** **Dados Software**

Empresa: Transportadora Flash Nome: Computador Faturamento 01 Nome: Mozilla Firefox

Setor: Financeiro

Solicitante: Ezio

Após a conclusão do incidente pelo técnico, o mesmo vai até a tela de Encerramento Incidente, onde informa o código do incidente no campo Código da Figura 16, e em seguida interage com o botão Consultar que preenche o campo Definição com a definição do respectivo incidente na tela de Encerramento do Incidente representado na Figura 16. Então, o técnico preenche os campos Data de Encerramento, Descrição da Solução, Procedimento, Complexidade de Aplicação da Solução, Risco de Aplicação da Solução, Tempo para Resolução e Custo para Resolução.

Abaixo dos dados de encerramento do incidente da Figura 16, tem-se os botões de encerramento, edição e exclusão. Acima, ao lado do campo Código, o botão de consulta. Deve-se destacar que todos os cadastros possuem o mesmo design de posicionamento de campos e botões, então compreendendo a disposição das telas de abertura e encerramento do incidente, não haverá dificuldades no entendimento dos outros cadastros do sistema.

Figura 16 - Tela Encerramento do Incidente

Sistema de Gestão de Incidentes

Cadastros Consultas Ajuda

Dados de Encerramento do Incidente

Código: 16

Definição: O navegador Firefox esta travando frequentemente e tem apresentado lentidão

Data de Encerramento: 11/09/2015

Descrição da Solução: Foi reiniciado o navegador sem o carregamento do navegador sem o plugin do Adobe Reader e o navegador funcionou normalmente, então foi realizado a atualização do navegador e respectivo plugin

Procedimento:

- 1° Iniciar o Firefox em modo de segurança segurando a tecla Shift ao abrir o Firefox para desativar as extensões temporariamente;
- 2° Realizar a atualização do navegador;
- 3° Realizar a atualização do plugin;
- 4° Clicar no botão Restaurar o Firefox irá restaurar o Firefox ao seu estado original,

Complexidade de Aplicação da Solução:

Risco de Aplicação da Solução:

Tempo para Resolução(hrs): 0.3

Custo para Resolução(R\$): 50

Na Figura 17, pode-se observar a tela de consulta de incidentes semelhantes que possui as seguintes características: o caixa de seleção de categoria do incidente, que possui as categorias representadas conforme mencionado na seção 3.3.8, os campos de texto de definição e descrição do incidente, que são analisados pela biblioteca Apache Lucene, a caixa de seleção que recebe o tipo do incidente que aplica um filtro opcional por hardware ou software, onde o técnico pode digitar no campo ao lado denominado Nome Hardware/Software o nome a descrição do equipamento ou o nome do software para que sejam apresentados apenas resultados específicos do equipamento ou software descrito, os cinco campos seguintes, que são número de ocorrências, custo, tempo, complexidade e risco, são os campos analisados pela biblioteca do MyCBR. A direita temos o botão de Consultar que gera a lista de incidentes semelhantes e aplica as operações mencionadas acima, preenchendo a tabela ordenada por grau de similaridade.

Esta tabela é composta pelos campos $Sim(\%)$, que corresponde ao grau de similaridade que este incidente tem com os dados inseridos pelo técnico acima, o campo $Cód.$ que representa o código do incidente em questão, o campo $Definição$ que contém a definição do incidente, o campo $Categoria$ e por fim o campo $Técnico$ que contém o nome do responsável por solucionar o incidente. Acima da tabela tem-se o campo que recebe o código do incidente para realizar consultas aos dados de abertura e encerramento do incidente.

Figura 17 - Consulta 1 para demonstrar a análise RBC

Sistema de Gestão de Incidentes

Cadastrados Consultas Configurações Ajuda

Consulta Incidentes Semelhantes

Categoria: Aplicação específicas(Navegador)

Definição: navegador trava

Descrição:

Tipo de incidente: Nenhum Nome Hardware/Software:

Nº de Ocorrências: 1

Custo Solução(R\$): 100

Tempo p/ Resolução(hrs): 1

Complexidade de Aplicação da Solução: Média

Risco de Aplicação da Solução: Médio

Incidentes Semelhantes

Código:

Sim. (%)	Cód.	Definição	Categoria	Técnico
83,303	13	O navegador Firefox trava ou não está respondendo	Aplicação específicas(Navegador)	Jewiton
73,66	32	O navegador Firefox trava ou não está respondendo	Aplicação específicas(Navegador)	Jewiton
62,901	14	Erro com plugins no navegador Mozilla Firefox	Aplicação específicas(Navegador)	Jewiton
54,865	4	Problema com um plugin para navegador	Aplicação específicas(Navegador)	Jewiton
46,307	7	Navegador fechando eventualmente	Aplicação específicas(Navegador)	Jewiton
45,374	11	Navegador apresentando janelas indesejadas durante a na...	Aplicação específicas(Navegador)	Jewiton
45,31	31	Navegador apresentando janelas indesejadas durante a na...	Aplicação específicas(Navegador)	Jewiton

No entanto, a questão mais interessante a ser abordada não são os elementos que compõem a tela de Consulta Incidente Semelhantes, mas sim como a funcionalidade de consulta pode ser utilizada, para isso foram elaborados alguns exemplos interessantes que demonstram a utilidade da funcionalidade, que são exemplificados nos parágrafos seguintes.

Observando os dados inseridos na consulta realizada na Figura 17, tem-se um incidente que ocorreu em um navegador onde o técnico busca um incidente que solucione o travamento do navegador, que tenha o custo de solução de aproximadamente R\$ 100, com tempo de resolução aproximado de 1 hora que tenha ocorrido apenas uma vez e possui complexidade e risco médios. Após pressionar o botão de consulta, ele obtém o retorno de uma série de incidentes, destacando-se os de código 13 e 32 que possuem respectivamente grau de similaridade de 83.303 % e 73.66 %, pois ambos contêm a definição idêntica e mencionam as palavras navegador e trava. Porém, o técnico tem interesse em saber qual das duas soluções aplicadas em incidentes anteriores se adequa melhor para o incidente atual, com isso, o técnico realiza um ajuste nos valores da consulta que pode ser observado na Figura 18, onde o custo seja mais barato e a solução seja mais rápida, além disso, o técnico deseja saber se uma das soluções possui complexidade e riscos baixos. Então, após realizar a consulta, pode-se observar que o incidente de código 32, que antes apresentava grau de similaridade de

73.66 % agora possui grau de similaridade de 91.342 %, pois o incidente 32 possui um custo de solução de R\$ 50, com tempo de resolução de 15 minutos e com a complexidade e o risco baixos, em quanto isso o incidente 13 diminuiu um pouco o seu grau de similaridade que passou a ser de 83.215 % devido a possuir um custo de R\$ 100, ter um tempo de resolução de 1 hora e apresentar uma complexidade baixa e risco médio.

Figura 18 - Consulta 2 para demonstrar a análise RBC

Sim. (%)	Cód.	Definição	Categoria	Técnico
91,342	32	O navegador Firefox trava ou não está respondendo	Aplicação específicas(Navegador)	Jewiton
83,215	13	O navegador Firefox trava ou não está respondendo	Aplicação específicas(Navegador)	Jewiton
63,988	7	Navegador fechando eventualmente	Aplicação específicas(Navegador)	Jewiton
62,991	31	Navegador apresentando janelas indesejadas durante a na...	Aplicação específicas(Navegador)	Jewiton
62,946	11	Navegador apresentando janelas indesejadas durante a na...	Aplicação específicas(Navegador)	Jewiton
62,492	16	O navegador Firefox esta travando frequentemente e tem ap...	Aplicação específicas(Navegador)	Jewiton
62,448	12	O navegador Google Chrome apresentando uma lentidão e...	Aplicação específicas(Navegador)	Jewiton

Observando o exemplo acima, pode-se pensar qual a diferença de utilizar o Apache Lucene a um `select` de um banco de dados, afinal ambos podem fazer o filtro por palavras contidas na definição de incidente. No entanto, o Apache Lucene possui um recurso que faz a análise de similaridade entre os textos, isto quer dizer que através da quantidade de textos inseridos no índice da consulta feita pela biblioteca, a mesma avalia a quantidade de vezes que a expressão aparece nos textos e quanto mais comum for a expressão nos textos, menor o peso, o contrário também é verdadeiro, caso a expressão apareça em apenas um dos textos esta possui um peso alto.

Na Figura 19, pode-se observar a consulta na categoria atualização de sistema operacional, na definição do incidente têm-se as palavras `erro` e `aplicativo` que podem ser observados na definição do incidente de código 22, e nos 30, 25, 29, 17 e 27 a palavra `windows` aparece na definição, portanto essa é uma palavra de menor peso tanto que todos os incidentes possuem no máximo 53 % de grau de similaridade. No

entanto, como a palavra erro aparece apenas nos incidentes 18 e 22 essa palavra possui maior peso. Deve-se ressaltar, que como neste exemplo, foram preenchidos os campos de definição e descrição, ambos representam 40%, porcentagem a qual é igualmente dividida entre os dois, portanto as diferenças entre o grau de similaridade do incidente 18 e o 22 acontece pois o incidente 18 é o único que possui as palavras instalar, service pack, bem como este incidente também é o único a possuir na descrição a expressão Ocorreu após baixar o service pack. Por isso, o grau de similaridade do incidente de código 18 é de 84.973 %. Caso haja interesse, nos apêndices deste trabalho, são detalhados outros exemplos que demonstram a análise RBC e de texto.

Figura 19 - Consulta 1 para demonstrar análise de texto

The screenshot shows the 'Sistema de Gestão de Incidentes' interface. The search criteria are as follows:

- Categoria:** Atualização(SO)
- Definição:** Erro ao instalar o service pack 1 requisitado por um aplicativo windows
- Descrição:** Ocorreu após baixar o service pack
- Tipo de incidente:** Nenhum
- Nome Hardware/Software:** (empty)
- Nº de Ocorrências:** 1
- Custo Solução(R\$):** 100
- Tempo p/ Resolução(hrs):** 1
- Complexidade de Aplicação da Solução:** Média
- Risco de Aplicação da Solução:** Médio

The results table is as follows:

Sim. (%)	Cód.	Definição	Categoria	Técnico
84,973	18	Erro 0x8024200D ao instalar o Service Pack 1 do Windows 7	Atualização(SO)	Jewiton
55,172	22	Erro 0xc000007b ao abrir aplicativo	Atualização(SO)	Jewiton
53,05	30	Sistema Operacional Windows 8 fica pedindo para atualizar...	Atualização(SO)	Jewiton
52,605	25	Problemas para realizar o upgrade do Windows 8.1 para o ...	Atualização(SO)	Jewiton
52,436	29	Mensagem de tela azul Windows 8 que não especifica o pro...	Atualização(SO)	Jewiton
52,369	17	Atraso ao desligar, reiniciar ou fazer logoff do Windows 7	Atualização(SO)	Jewiton
52,07	27	Não é possível configurar as opções de rede no Windows 8	Atualização(SO)	Jewiton

3.5 RESULTADOS E DISCUSSÕES

O objetivo principal deste trabalho foi desenvolver uma solução computacional para auxiliar empresas de TI na gestão de incidentes, especificamente para empresas de menor porte onde o modelo MPS-SV se mostra mais interessante. Além disso, o trabalho se propõe a desenvolver um modelo de busca por casos de incidentes semelhantes. Verificou-se que ambos os objetivos foram alcançados através de um sistema de gestão de incidentes que possibilita a gerencias de incidentes, técnicos responsáveis, empresas, setores, equipamentos e softwares, e de uma funcionalidade

específica para busca de incidentes semelhantes com o uso de técnicas de ontologia, RBC e análise de texto.

Para analisar os trabalhos correlatos, serão levados em consideração os resultados esperados com a implementação do GIN do MPS-SV apresentados na seção 2.5, além de funcionalidades e recursos utilizados para otimizar a gestão dos incidentes. No Quadro 19, pode-se observar uma breve comparação entre as características do trabalho desenvolvido e os trabalhos correlatos.

Quadro 19 - Comparação entre características do trabalho desenvolvido e os correlatos

	Sistema desenvolvido	Ocomon	OTRS Help Desk	Silva (2002)
MR para GIN	MPS-SV	ITIL	ITIL	Nenhum
Abrangência da Gestão	Hardware e Software	Hardware e Software	Hardware e Software	Software
GIN 1	Possui	Possui	Possui	Não faz parte do escopo
GIN 2	Possui	Possui	Possui	Não faz parte do escopo
GIN 3	Possui	Possui	Possui	Possui
GIN 4	Possui	Possui	Possui	Não faz parte do escopo
GIN 5	Possui	Possui	Possui	Possui
GIN 6	Não faz parte do escopo	Possui	Possui	Não faz parte do escopo
GIN 7	Não faz parte do escopo	Não Possui	Possui	Não faz parte do escopo
Análise de texto ou busca	Através de análise de texto da definição e descrição do incidente (Apache Lucene)	Busca por banco de dados (MySQL)	Busca por banco de dados (Não especificado)	Através de busca de palavras chave
Análise de negócio	Através de análise de modelo RBC (MyCBR)	Análise estatística	Análise estatística	Não faz parte do escopo

O primeiro elemento do quadro, se refere ao modelo de referencia utilizado para a solução na gestão de incidentes no caso do Ocomon e do OTRS Help Desk se baseiam no ITIL. Embora o ITIL seja um modelo claramente mais detalhado nos processos que ele abrange com relação ao MPS-SV, o ITIL acaba se mostrando como uma opção que possui maior custo e consumo de tempo para adoção dos padrões.

O MPS-SV propõe um modelo mais barato, que aproveite os processos já utilizados na empresa, apenas acrescentando os processos que não são aplicados ainda. Então, basicamente o MPS-SV busca estabelecer os critérios primordiais que a empresa de TI deve adotar, deixando o modo como esses padrões serão implementados a critério da empresa. Tendo um modelo mais versátil e abrangente, adequando-se assim ao mercado nacional.

O sistema desenvolvido por Silva (2002) não adota nenhum modelo de referência e se encaixa na gestão de software. O sistema desenvolvido, Ocomon e o OTRS propõem uma gestão de hardware e software.

Com relação ao GIN 1, tanto o sistema desenvolvido quanto o Ocomon e o OTRS, possuem uma abordagem de gerenciamento de incidentes que é mantida através da utilização dos sistemas em questão. No entanto, o sistema de Silva (2002) possui o escopo de encontrar a solução para problemas de suporte técnico de empresas de desenvolvimento de software.

Com relação aos GIN 2, GIN 3, GIN 4 e GIN 5, tanto o sistema desenvolvido quanto o Ocomon e o OTRS possibilitam o gerenciamento e controle de incidentes, o registro e a classificação de incidentes, a priorização e a análise de incidentes e por fim o registro da resolução e o encerramento de incidentes. O sistema de Silva (2002) atende apenas aos GIN 3 e GIN 5.

O GIN 6 requer o escalonamento dos incidentes não atendidos para priorização dos mesmos, sendo assim apenas as ferramentas comerciais Ocomon e OTRS possuem esta funcionalidade. O sistema desenvolvido neste trabalho possibilita a categorização dos incidentes, no entanto não possui uma funcionalidade que gerencia os incidentes de acordo com a sua prioridade e os escalona conforme o tempo. Assim como o sistema de Silva (2002) também não.

Isto ocorreu devido a ambos os trabalhos acadêmicos possuírem um foco específico no auxílio a busca de soluções de incidentes, como pode ser observado na coluna sobre análise de texto ou busca. Onde os sistemas comerciais apenas realizam buscas através de consultas de banco de dados, os trabalhos acadêmicos realizam análises de texto mais complexas categorizando os resultados por grau de similaridade.

No entanto o sistema desenvolvido neste trabalho, se destaca com relação ao sistema de Silva (2002), pois o mesmo possibilita a busca pela definição feita pelo técnico e a descrição feita pelo cliente. Realizando uma análise de similaridade automática que classifica e categoriza as expressões conforme a frequência que a mesma aparece nos textos procurados, ou seja, se uma expressão contida na consulta é frequente em vários dos textos pesquisados essa expressão possui menor peso, e se uma expressão contida na consulta é rara nos textos procurados ela possui maior peso. Já no trabalho de Silva cabe ao usuário definir o grau de importância de cada expressão, que também possui suas vantagens, pois possibilita ao usuário escolher exatamente o nível

de prioridade de cada expressão. No ambiente comercial, esta opção pode se tornar pouco funcional por exigir que o usuário classifique cada expressão, sendo que a busca pode conter diversas expressões, então este processo de classificação pode se tornar cansativo.

Deve-se ressaltar que isso ocorre, pois o sistema de Silva (2002) foi desenvolvido a mais de 10 anos atrás sem a utilização de nenhuma biblioteca de análise de texto. Já o sistema desenvolvido neste trabalho, utiliza os recursos da biblioteca Apache Lucene.

Por fim, a última coluna do Quadro 19, trata da análise de negócio. Onde o sistema desenvolvido neste trabalho utiliza uma análise específica direcionada a tratar cada incidente, para que o técnico possa optar pela solução que se adéque com relação a custo, tempo, complexidade e risco da solução a ser aplicada. As ferramentas comerciais Ocomon e o OTRS utilizam uma análise estatística voltada para a gerencia do negócio e não para busca da melhor solução de acordo com o incidente e a situação.

A principal dificuldade encontrada no desenvolvimento do trabalho, foi a utilização da biblioteca do MyCBR. Desde o início do projeto, a utilização da biblioteca do MyCBR se mostrou complicada. Devido primeiramente a dificuldade em se encontrar exemplos que demonstrassem o funcionamento da mesma, sobrando apenas um breve tutorial dos desenvolvedores que mostrava como utilizar as funcionalidades básicas da biblioteca. Porém, este tutorial fazia uso de classes da biblioteca que haviam sido descontinuadas, e para aumentar o clima de incerteza, a documentação da biblioteca está incompleta apresentando vários métodos sem nenhuma explicação.

Durante as etapas iniciais do desenvolvimento, chegou a ser cogitada a utilização de outra biblioteca, pois até então, no segundo mês de desenvolvimento não se tinha conseguido desenvolver um exemplo utilizando a biblioteca. No entanto, ao fim destes dois meses, foram encontrados dois exemplos funcionais que utilizavam a biblioteca com as classes atuais da mesma. Analisando os exemplos, o tutorial dos desenvolvedores da biblioteca e a documentação, foi possível elaborar um modelo RBC para representar incidente para que a aplicação pudesse utilizar os recursos propostos.

Após a compreensão do funcionamento da biblioteca, o processo de criação no modelo e a utilização do mesmo foi rápida. No entanto, foram observados alguns problemas em métodos específicos da biblioteca que não funcionam corretamente.

O restante do desenvolvimento do sistema ocorreu conforme o esperado, sem maiores problemas. Deve-se destacar a boa documentação, a quantidade de materiais e exemplos que as bibliotecas Apache Jena e Apache Lucene possuem.

4 CONCLUSÕES

O objetivo deste trabalho era de desenvolver uma solução computacional para auxiliar na gerencia de incidentes na área de TI, com base no modelo de referencia MPS-SV, especificando um modelo de busca de incidentes semelhantes, que otimiza-se o reaproveitamento de soluções já aplicadas para novos casos. Primeiramente é interessante observar separadamente os objetivos específicos estabelecidos previamente que podem ser observados na seção 1.1.

Sendo o primeiro dos objetivos identificar os principais elementos propostos pelo MPS-SV, levando em consideração a realidade de empresas de pequeno e médio porte da região do Vale do Itajaí. Para isso, foi realizado um estudo específico sobre a área de GIN do modelo de referência, observando os principais resultados esperados com a implantação do MPS-SV.

Portanto, é fundamental comparar-se os resultados esperados com a implementação do GIN apresentados na seção 2.5, com os que este trabalho atende, os GIN 1, GIN 2, GIN 3, GIN 4 e GIN 5 são atendidos com o sistema desenvolvido, pois o mesmo propõe uma abordagem de gerenciamento dos incidentes que possibilite o registro, classificação, priorização e resolução dos incidentes. No entanto, o GIN 6, que consiste em escalonamento dos incidentes não atendidos com o tempo não foi implementado no sistema, e o GIN 7 que consiste em comunicar as partes interessadas, também não foi previsto pelo sistema.

O objetivo específico seguinte, tratava sobre a especificação de um modelo de ontologia para representar as categorias dos incidentes com intuito de possibilitar a classificação e subdivisão dos incidentes nestas categorias. Este objetivo foi atendido através da utilização da biblioteca Apache Jena, conforme pode ser observado na seção 3.3.8, onde é demonstrado o modelo criado, sendo que na seção 3.4.2, o mesmo foi aplicado no cadastramento dos incidentes, onde o técnico classifica os incidentes de acordo com a sua respectiva categoria.

O objetivo específico seguinte, tratava sobre a especificação e aplicação de um modelo de análise RBC, visando questões comerciais, ou seja, um modelo que fizesse uma análise se a solução dos incidentes retornados se adequavam em questões de custo, tempo, complexidade e risco. Este objetivo também foi cumprido, como pode ser observado na seção 3.3.9, onde é demonstrado o modelo desenvolvido, e na seção 3.4.2 se pode observar a aplicação do mesmo.

Por fim, o último objetivo específico de aplicar análise textual na definição e na descrição dos incidentes para otimizar a busca por incidentes semelhantes foi cumprido, através da utilização da biblioteca Apache Lucene. Esta análise é aplicada na etapa de consulta a incidentes semelhantes, conforme se pode observar na descrição do caso de uso Quadro 4, e a aplicação da mesma na realização de consultas na seção 3.4.2.

Pode-se observar que o objetivo principal foi cumprido, pois este trabalho apresenta um sistema que possibilita a gestão dos incidentes seguindo o modelo de referência MPS-SV, e que foi especificado e aplicado um modelo de busca por incidentes semelhantes, conforme pode ser observado na seção 3.3.10 a especificação, e na seção 3.4.2 a sua aplicação.

Dentre as ferramentas utilizadas, a única que se mostrou problemática e que talvez não deveria ter sido utilizada, foi a biblioteca do MyCBR que acabou prejudicando o desenvolvimento do trabalho, os motivos são detalhados na seção 3.5. O Apache Jena se mostrou uma ferramenta interessante que atendeu as necessidades deste trabalho. O Apache Lucene é uma grande biblioteca, sendo fundamental para o desenvolvimento deste trabalho, deve-se ressaltar que a mesma foi explorada apenas de maneira básica.

As principais contribuições deste trabalho foram primeiramente no campo comercial. Pois ele é uma solução barata utilizando apenas ferramentas gratuitas e abertas no seu desenvolvimento. Sendo de fácil utilização, aplicando apenas a gestão direcionada ao cliente e a solução dos incidentes. Além de não exigir grandes mudanças nos processos das empresas, devido ao modelo simples e flexível adotado do MPS-SV, que apenas aponta os objetivos a serem atendidos dando liberdade para a empresa optar pela sua forma de atendimento destes objetivos.

Outra contribuição interessante do trabalho, é na questão de que o mesmo especifica um modelo de consulta de incidentes semelhantes. Fazendo uma análise que leva em consideração a melhor solução a ser aplicada para o incidente, bem como qual dessas soluções interessa mais a empresa.

Também é fundamental ressaltar, que embora a solução aplicada seja simples e talvez não seja a solução ideal, a mesma une uma série de conceitos como ontologia, RBC e análise de textos e mostra como é possível obter soluções interessantes, aproveitando e unindo estes diversos conceitos científicos. Este trabalho abre caminho para uma série de possíveis melhorias, pois a ideia de unir a busca de uma solução

semelhante, junto com a análise de qual das soluções semelhantes se adéqua melhor ao caso é intrigante, pois este mesmo conceito pode ser aplicado para as mais diversas áreas.

Por fim, deve-se destacar as limitações deste trabalho que em sua maioria ocorreram por falta de tempo ou por não pertencerem a escopo fundamental do trabalho. A primeira delas é que a funcionalidade básica do sistema, que é de exclusão dos cadastros, não foi implementada. Outra funcionalidade que visava possibilitar a edição da ontologia e do modelo RBC pelo usuário, não foram implementadas. Além disso, o sistema teve que passar por uma simplificação na interface, que acaba tornando a usabilidade do mesmo menos agradável do que se pretendia na ideia original.

4.1 EXTENSÕES

As sugestões de extensões para trabalhos futuros são:

- a) substituir a biblioteca MyCBR pela jColibri CBR ou buscar outra alternativa melhor;
- b) utilizar na análise de texto da biblioteca Apache Lucene filtros de *Stopwords* e *Stemming*;
- c) utilizar na análise de texto da biblioteca Apache Lucene reconhecimento de sinônimos;
- d) possibilitar a edição ontologia de modelo de categorias de incidente ao usuário;
- e) possibilitar a edição e ajustes nos parâmetros de análise RBC;
- f) realizar estudos com relação as outras áreas que o MR-MPS.BR trata;
- g) otimizar o modelo de análise de incidentes semelhantes;
- h) otimizar o modelo da ontologia;
- i) otimizar o modelo RBC;
- j) implementar o GIN 6 para a aplicação do sistema em empresas de grande porte.

REFERÊNCIAS

- ABREU, Vladimir F.; FERNANDES, Aguinaldo A. **Implantando a Governança de TI da Estratégia à Gestão dos Processos e Serviços**. Rio de Janeiro: Brasport, 2012.
- ALMEIDA, Mauricio B.; BAX, Marcello P. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. **SciELO**, Brasília, v. 32, n. 3, p. 7-20, Set./Dez. 2003.
- APACHE JENA. **Jena Ontology API**. [S.l.], [201?a]. Disponível em: <<http://jena.apache.org/documentation/ontology/>>. Acesso em: 27 ago. 2015.
- APACHE JENA. **Uma Introdução a RDF e à API RDF de Jena**. [S.l.], [201?b]. Disponível em: <https://jena.apache.org/tutorials/rdf_api_pt.html>. Acesso em: 27 ago. 2015.
- ARAUJO, Larissa L. et al. Experiência de Implantação do MR-MPS-SV no Service Desk da ECO Sistemas. In: SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, 13. 2014, Blumenau. **Anais...** Blumenau: [s.n.], 2014. p. 271-278.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 15504-1: 2008**. [S. l.], 2008.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 20000-2: 2013**. [S.l.], 2013.
- ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS-SV: 2012**. [S.l.], 2013a.
- ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **Guia Geral MPS de Serviços**. [S.l.], 2012.
- ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **MPS-SV**. [S.l.], 2014. Disponível em: <<http://www.softex.br/mpsbr/mps/servicos/>>. Acesso em: 01 set. 2015.
- ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **Saiba como começou o programa MPS.BR**. [S.l.], 2013b. Disponível em: <<http://www.softex.br/saiba-como-comecou-o-programa-mps-br/>>. Acesso em: 23 mar. 2015.
- BERTOLINI, Cristiano; CERATTI, Paulo R.; SILVEIRA, Sidnei R. Implementação do MPS.BR na empresa Digifred Sistemas para Gestão Pública: Um Estudo de Caso. In: CESNORS, [?]., [2014], Frederico Westphalen. **Anais...** Frederico Westphalen: UFSM, [2014]. p. 1-20. Disponível em: <http://w3.ufsm.br/cesnors/images/Implementa%C3%A7%C3%A3o_do_MPS.BR_na_empresa_Digifred_Sistemas_para_Gest%C3%A3o_P%C3%BAblica_Um_Estudo_de_Caso.pdf>. Acesso em: 23 mar. 2015.
- CARVALHO, Cedric L.; LIMA, Júnio C. **Ontologias - OWL (Web Ontology Language)**. Goiânia, 2005. 16 p. Disponível em: <http://www.portal.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_004-05.pdf>. Acesso em: 20 maio 2015.

- CORRÊA, Felipe P.; GASPAR, Luciano; OLIVEIRA, Elisamara. **Estudo de Caso: gestão de incidentes e problemas com base nas práticas da ITIL em uma instituição bancária - Primeira Parte.** [S.l.], [201?]. Disponível em: <<http://www.getinews.com.br/volumes/184/estudo-de-caso-gestao-de-incidentes-e-problemas-com-base-nas-praticas-da-til-em-uma-instituicao-bancaria-primeira-parte>>. Acesso em: 19 maio 2015.
- DICKINSON, Ian. Elo perdido. In: **Linux Magazine**, [S.l.], n. 80, p. 63-69, jul. 2011.
- FERNANDES, Anita M. R; KRAUS, Helton M. Ferramenta para Ensino da Técnica de Raciocínio Baseado em Casos. In: **COMPUTER ON THE BEACH**, 1, 2010, Florianópolis-SC. **Anais...** Local: Universidade do Vale do Itajaí – UNIVALI, 2010. p. 1-6.
- GRUBER, Tom. **Ontology.** [S.l.], 2009. Disponível em: <<http://tomgruber.org/writing/ontology-definition-2007.htm>>. Acesso em: 20 maio 2015.
- GUIMARÃES, Lucyene C. et al. Projetos de Tecnologia da Informação: Caracterização da Gestão de Projetos de TI no estado de Pernambuco. In: **ENEGEP**, 28., 2008, Rio de Janeiro. **Anais...** Rio de Janeiro: ABEPRO, 2008. p. 1-15. Disponível em: <http://www.abepro.org.br/biblioteca/enegep2008_tn_sto_076_538_12018.pdf>. Acesso em 19 maio 2015.
- HEINRICH, Luciane T. **Sistema de informação aplicados a lojas de confecções do Alto Vale do Itajaí – SC utilizando raciocínio baseado em casos.** 2000. 87 f. Trabalho de conclusão de curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- HEINZLE, Roberto. **Um modelo de engenharia do conhecimento para sistemas de apoio a decisão com recursos para raciocínio abduutivo.** 2011. 247 f. Tese (Doutorado em Engenharia e Gestão do Conhecimento) – Programa de Pós Graduação em Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis.
- INSTITUTO DE TECNOLOGIA DE SOFTWARE. **O modelo MPS.Br.** [S.l.], [201?]. Disponível em: <<http://its.org.br/o-que-e-mpsbr/o-modelo-mps-br/>>. Acesso em: 18 mar. 2015.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 20000-1: 2011.** [S.l.], 2011. Disponível em: <<http://www.praxiom.com/iso-20000-definitions.htm>>. Acesso em: 02 abr. 2015.
- JFLEX. **JFlex.** [S.l.], 2015. Disponível em: <<http://jflex.de/>>. Acesso em: 29 out. 2015.
- LUCENE. **Apache Lucene Core.** [S.l.], 2012. Disponível em: <<http://lucene.apache.org/core/#lucenetm-features>>. Acesso em: 20 out. 2015.
- LUCENE. **Lucene 5.3.1 core API.** [S.l.], 2015. Disponível em: <https://lucene.apache.org/core/5_3_1/core/index.html>. Acesso em: 29 out. 2015.
- MAGALHÃES, Ivan L.; PINHEIRO, Walfrido B. **Gerenciamento de serviços de TI na prática: uma abordagem com base na ITIL.** São Paulo: Novatec, 2007.
- MYCBR. **Welcome.** [S.l.], 2015. Disponível em: <<http://mycbr-project.net/index.html>>. Acesso em: 23 maio 2015.

OFFICE OF GOVERNMENT COMMERCE. **ITIL** – The Official Introduction to the ITIL Service Lifecycle. London, 2007. Disponível em:

<<http://www.training.com.br/download/ITIL%20v3%20Study%20Guide.pdf>>.

Acesso em: 18 mar. 2015.

OTRS. **Download OTRS Open Source Help Desk Software and free features**. [S.l.],

2015. Disponível em: <<https://www.otrs.com/download-open-source-help-desk-software-otrs-free/>>. Acesso em: 04 mar. 2015.

PAMPLONA, Vitor. **Introdução ao Apache Lucene**. [S.l.], 2009. Disponível em: <<http://vitorpamplona.com/wiki/Introdu%C3%A7%C3%A3o%20ao%20Apache%20Lucene>>. Acesso em: 20 out. 2015.

RIBEIRO, Flávio. **O Ocomon**. [S.l.], [200?]. Disponível em:

<<http://ocomonphp.sourceforge.net/>>. Acesso em: 01 abr. 2015.

RIBEIRO, José. **OTRS** - Revolucione seu Help Desk com esta ferramenta. [S.l.], 2012.

Disponível em: <<http://www.vivaolinux.com.br/artigo/OTRS-Revolucione-seu-Help-Desk-com-esta-ferramenta.html>>. Acesso em: 04 mar. 2015.

RIESBECK, Christopher K.; SCHANK, Roger C. **Inside case-based reasoning**.

Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1989.

SILVA, Carlos E. de S. **Sistema de apoio para otimização das atividades de suporte técnico de uma empresa de desenvolvimento de software, utilizando raciocínio baseado em casos**. 2002. 72 f. Trabalho de conclusão de curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SOUSA, Pedro. **MR-MRS-SV x CMMI-SVC**. [S.l.], 2013. Disponível em:

<<https://prezi.com/z9etmwvgtrqq/mr-mrs-sv-x-cmmi-svc/>>. Acesso em: 31 mar. 2015.

SOWA, J. F. **Building, sharing and merging ontologies**. [S.l.], 1999. Disponível

em:<<http://www.jfsowa.com/ontology/index.htm>>. Acesso em: 20 maio 2015.

W3C. **OWL Web Ontology Language: Guide**. [S.l.], 2004. Disponível em:

<<http://www.w3.org/TR/owl-guide/>>. Acesso em: 28 out. 2015.

APÊNDICE A – Detalhamento dos casos de uso

Neste apêndice, são apresentadas as descrições completas dos casos de uso do sistema desenvolvido neste trabalho previstos na seção 3.3.1, que não foram apresentados até então, o Quadro 20 apresenta o caso de uso *Cadastrar Empresa*, o Quadro 21 apresenta o caso de uso *Cadastrar Setor*, o Quadro 22 apresenta o caso de uso *Cadastrar Equipamento*, o Quadro 23 apresenta o caso de uso *Cadastrar Software*, o Quadro 24 apresenta o caso de uso *Consultar Incidentes* e o Quadro 25 apresenta o caso de uso *Cadastrar Técnico*.

Quadro 20 - Detalhamento do caso de uso Cadastrar Empresa

UC 01 – Cadastrar Empresa	
Ator	Técnico
Objetivo	Permitir ao técnico cadastrar, consultar, editar e excluir empresas.
Pré-condições	Em caso de edição, consulta e exclusão é necessário que a empresa já tenha sido cadastrada previamente.
Cenário principal	1) O técnico seleciona a opção “Cadastrados”; 2) O técnico seleciona a opção “Empresa”; 3) O técnico opta por cadastrar uma nova empresa; 4) O técnico informa os dados básicos da empresa: - Nome da empresa; - Categoria da empresa. 5) O técnico seleciona a opção “Cadastrar”; 6) O sistema registra os dados da empresa.
Cenário alternativo	No passo 4, após informar o nome da empresa, o técnico opta por consultar os dados da empresa. 4.1) O técnico opta por consultar a empresa; 4.2) O técnico seleciona a opção “Consultar”; 4.3) O sistema mostra os dados da empresa consultada.
Cenário alternativo	Após o passo 4.3, o técnico opta por editar os dados da empresa. 4.3.1) O técnico opta por editar os dados da empresa; 4.3.2) O técnico realiza as alterações nos dados da empresa; 4.3.3) O técnico seleciona a opção “Editar”; 4.3.4) Retorne para o item 6.
Cenário alternativo	Após o passo 4.3, o técnico opta por excluir a empresa. 4.3.1) O técnico opta por excluir a empresa. 4.3.2) O técnico seleciona a opção “Excluir”; 6.1) O sistema exclui a empresa.
Cenário alternativo	No passo 4.3.1, o técnico opta por editar o nome da empresa. 4.3.1.1) O técnico seleciona a opção “Alterar Nome”; 4.3.1.2) O técnico informa o novo nome da empresa; 4.3.1.3) Retorne para o item 6.
Cenário de exceção	Obrigatoriedade no preenchimento do campo de nome da empresa. O sistema exibe uma mensagem informando a situação.
Cenário de exceção	Obrigatoriedade de que não existe nenhum setor vinculado a empresa na base de dados para realizar a exclusão. O sistema exibe uma mensagem informando a situação.
Cenário de exceção	Obrigatoriedade de que não exista outra empresa com o mesmo nome. O sistema exibe uma mensagem informando a situação.
Pós-condições	Uma empresa foi cadastrada, editada ou excluída.

Quadro 21 - Detalhamento do caso de uso Cadastrar Setor

UC 02 – Cadastrar Setor	
Ator	Técnico
Objetivo	Permitir ao técnico cadastrar, consultar, editar e excluir setores das empresas.
Pré-condições	A empresa associada ao setor deve ser previamente cadastrada. Em caso de edição, consulta e exclusão é necessário que a empresa e o seu respectivo setor já tenham sido cadastrados previamente.
Cenário principal	1) O técnico seleciona a opção “Cadastrados”; 2) O técnico seleciona a opção “Setor”; 3) O técnico opta por cadastrar um novo setor; 4) O técnico informa os dados básicos da empresa: - Nome empresa. 5) O técnico informa os dados básicos do setor:

	<ul style="list-style-type: none"> - Nome setor; - Descrição; - Categoria (Trivial, Importante, Fundamental); <p>6) O técnico seleciona a opção “Cadastrar”;</p> <p>7) O sistema registra os dados do setor.</p>
Cenário alternativo	<p>No passo 5, após informar o nome do setor, o técnico opta por consultar os dados do setor.</p> <p>5.1) O técnico opta por consultar o setor;</p> <p>5.2) O técnico seleciona a opção “Consultar”;</p> <p>5.3) O sistema mostra os dados do setor consultado.</p>
Cenário alternativo	<p>Após o passo 5.3, o técnico opta por editar os dados do setor.</p> <p>5.3.1) O técnico opta por editar os dados do setor;</p> <p>5.3.2) O técnico realiza as alterações nos dados do setor;</p> <p>5.3.3) O técnico seleciona a opção “Editar”;</p> <p>5.3.4) Retorne para o item 7.</p>
Cenário alternativo	<p>Após o passo 5.3, o técnico opta por excluir o setor.</p> <p>5.3.1) O técnico opta por excluir o setor.</p> <p>5.3.2) O técnico seleciona a opção “Excluir”;</p> <p>7.1) O sistema exclui o setor.</p>
Cenário alternativo	<p>No passo 5.3.1, o técnico opta por editar o nome do setor.</p> <p>5.3.1.1) O técnico seleciona a opção “Alterar Nome”;</p> <p>5.3.1.2) O técnico informa o novo nome do setor;</p> <p>5.3.1.3) Retorne para o item 7.</p>
Cenário de exceção	<p>Obrigatoriedade no preenchimento do campo de nome da empresa e do setor. O sistema exibe uma mensagem informando a situação.</p>
Cenário de exceção	<p>Obrigatoriedade de que não existe nenhum equipamento vinculado ao setor na base de dados para realizar a exclusão. O sistema exibe uma mensagem informando a situação.</p>
Cenário de exceção	<p>Obrigatoriedade de que não exista outra empresa com o mesmo nome. E que não exista um setor com o mesmo nome na empresa. O sistema exibe uma mensagem informando a situação.</p>
Pós-condições	<p>Um setor foi cadastrado, editado ou excluído de uma empresa.</p>

Quadro 22 - Detalhamento do caso de uso Cadastrar Equipamento

UC 03 – Cadastrar Equipamento	
Ator	Técnico
Objetivo	Permitir ao técnico cadastrar, consultar, editar e excluir equipamentos dos setores das empresas.
Pré-condições	A empresa e o setor associados devem ser previamente cadastrados. Em caso de edição, consulta e exclusão é necessário que a empresa, o setor e o respectivo equipamento já tenham sido cadastrados previamente.
Cenário principal	<ol style="list-style-type: none"> 1) O técnico seleciona a opção “Cadastros”; 2) O técnico seleciona a opção “Equipamento”; 3) O técnico opta por cadastrar um novo equipamento; 4) O técnico informa os dados básicos da empresa: <ul style="list-style-type: none"> - Nome empresa. 5) O técnico informa os dados básicos do setor: <ul style="list-style-type: none"> - Nome setor. 6) O técnico informa os dados básicos do equipamento: <ul style="list-style-type: none"> - Nome equipamento; - Descrição; - Categoria (Trivial, Importante, Fundamental); - Valor (R\$); - Versão;

	7) O técnico seleciona a opção “Cadastrar”; 8) O sistema registra os dados do equipamento.
Cenário alternativo	No passo 6, após informar o nome do equipamento, o técnico opta por consultar os dados do equipamento. 6.1) O técnico opta por consultar o equipamento; 6.2) O técnico seleciona a opção “Consultar”; 6.3) O sistema mostra os dados do equipamento consultado.
Cenário alternativo	Após o passo 6.3, o técnico opta por editar os dados do equipamento. 6.3.1) O técnico opta por editar os dados do equipamento; 6.3.2) O técnico realiza as alterações nos dados do equipamento; 6.3.3) O técnico seleciona a opção “Editar”; 6.3.4) Retorne para o item 8.
Cenário alternativo	Após o passo 6.3, o técnico opta por excluir o equipamento. 6.3.1) O técnico opta por excluir o equipamento. 6.3.2) O técnico seleciona a opção “Excluir”; 8.1) O sistema exclui o equipamento.
Cenário alternativo	No passo 6.3.1, o técnico opta por editar o nome do equipamento. 6.3.1.1) O técnico seleciona a opção “Alterar Nome”; 6.3.1.2) O técnico informa o novo nome do equipamento; 6.3.1.3) Retorne para o item 8.
Cenário de exceção	Obrigatoriedade no preenchimento do campo de nome da empresa, setor e equipamento. O sistema exibe uma mensagem informando a situação.
Cenário de exceção	Obrigatoriedade de que não existe nenhum software vinculado ao equipamento na base de dados para realizar a exclusão. O sistema exibe uma mensagem informando a situação.
Cenário de exceção	Obrigatoriedade de que não exista outra empresa com o mesmo nome. E que não exista um setor com o mesmo nome na empresa. E que por fim não exista um equipamento com a mesma identificação no setor. O sistema exibe uma mensagem informando a situação.
Pós-condições	Um equipamento foi cadastrado, editado ou excluído de um setor.

Quadro 23 - Detalhamento do caso de uso Cadastrar Software

UC 04 – Cadastrar Software	
Ator	Técnico
Objetivo	Permitir ao técnico cadastrar, consultar, editar e excluir softwares.
Pré-condições	A empresa, setor e o equipamento associado devem ser previamente cadastrados. Em caso de edição, consulta e exclusão é necessário que a empresa, o setor, o equipamento e o respectivo software já tenham sido cadastrados previamente.
Cenário principal	1) O técnico seleciona a opção “Cadastros”; 2) O técnico seleciona a opção “Software”; 3) O técnico opta por cadastrar um novo software; 4) O técnico informa os dados básicos da empresa: - Nome empresa. 5) O técnico informa os dados básicos do setor: - Nome setor. 6) O técnico informa os dados básicos do equipamento: - Nome equipamento. 7) O técnico informa os dados básicos do software: - Nome software; - Descrição; - Categoria (Trivial, Importante, Fundamental); - Valor (R\$);

	- Versão; 8) O técnico seleciona a opção “Cadastrar”; 9) O sistema registra os dados do software.
Cenário alternativo	No passo 7, após informar o nome do equipamento, o técnico opta por consultar os dados do software. 7.1) O técnico opta por consultar o software; 7.2) O técnico seleciona a opção “Consultar”; 7.3) O sistema mostra os dados do software consultado.
Cenário alternativo	Após o passo 7.3, o técnico opta por editar os dados do software. 7.3.1) O técnico opta por editar os dados do software; 7.3.2) O técnico realiza as alterações nos dados do software; 7.3.3) O técnico seleciona a opção “Editar”; 7.3.4) Retorne para o item 9.
Cenário alternativo	Após o passo 7.3, o técnico opta por excluir o software. 7.3.1) O técnico opta por excluir o software. 7.3.2) O técnico seleciona a opção “Excluir”; 9.1) O sistema exclui o software.
Cenário alternativo	No passo 7.3.1, o técnico opta por editar o nome do software. 7.3.1.1) O técnico seleciona a opção “Alterar Nome”; 7.3.1.2) O técnico informa o novo nome do software; 7.3.1.3) Retorne para o item 9.
Cenário de exceção	Obrigatoriedade no preenchimento do campo de nome da empresa, setor, equipamento e software. O sistema exibe uma mensagem informando a situação.
Cenário de exceção	Obrigatoriedade de que não exista um software com a mesma identificação. O sistema exibe uma mensagem informando a situação.
Pós-condições	Um software foi cadastrado, editado ou excluído de um equipamento.

Quadro 24 - Detalhamento do caso de uso Consultar Incidentes

UC 06 – Consultar Incidentes	
Ator	Técnico
Objetivo	Permitir ao técnico encontrar os incidentes desejados.
Pré-condições	Ter previamente cadastrados incidentes para que os mesmos possam ser consultados.
Cenário principal	1) O técnico seleciona o item “Incidentes” no menu “Consultas”; 2) O sistema exibe a lista de todos os incidentes cadastrados no sistema; 3) O técnico opta por consultar os dados de abertura de um incidente visualizado; 4) O técnico informa o código do incidente; 5) O técnico seleciona a opção “Consultar Dados de Abertura”; 6) O sistema exibe a tela de abertura do incidente constando os dados do incidente requisitado.
Cenário alternativo	No passo 3, o técnico opta por consultar os dados de encerramento de um incidente visualizado. 3.1) O técnico informa o código do incidente; 3.2) O técnico seleciona a opção “Consultar Dados de Encerramento”; 3.3) O sistema exibe a tela de encerramento do incidente constando os dados do incidente requisitado.
Cenário alternativo	Após o passo 2, o técnico opta por atualizar a tabela para verificar se não foi inserido novos incidentes. 2.1) O sistema atualiza a lista exibida na tabela com base nos incidentes cadastrados no sistema.
Cenário	Após o passo 2, o técnico opta por ordenar a lista por um dos filtros da tabela

alternativo	da tela de consulta de incidentes. 2.1) O técnico clica sobre a coluna desejada; 2.2) O sistema realiza a ordenação dos dados da tabela;
Cenário de exceção	Obrigatoriedade no preenchimento do campo código do incidente para realizar a consulta. O sistema exibe uma mensagem informando a situação.

Quadro 25 - Detalhamento do caso de uso Cadastrar Técnico

UC 08 – Cadastrar Técnico	
Ator	Técnico
Objetivo	Permitir ao técnico cadastrar, consultar, editar e excluir técnicos.
Pré-condições	Em caso de edição, consulta e exclusão é necessário que o técnico já tenha sido cadastrada previamente.
Cenário principal	1) O técnico seleciona a opção “Cadastros”; 2) O técnico seleciona a opção “Técnico”; 3) O técnico opta por cadastrar um novo técnico 4) O técnico informa os dados básicos do técnico: - Nome Técnico; - Função (Eletrotécnica, Rede, Manutenção, Análise de Software). 5) O técnico seleciona a opção “Cadastrar”; 6) O sistema registra os dados do técnico.
Cenário alternativo	No passo 4, após informar o nome do técnico, o técnico opta por consultar os dados do técnico. 4.1) O técnico opta por consultar o técnico; 4.2) O técnico seleciona a opção “Consultar”; 4.3) O sistema mostra os dados do técnico consultado.
Cenário alternativo	Após o passo 4.3, o técnico opta por editar os dados do técnico. 4.3.1) O técnico opta por editar os dados do técnico; 4.3.2) O técnico realiza as alterações nos dados do técnico; 4.3.3) O técnico seleciona a opção “Editar”; 4.3.4) Retorne para o item 6.
Cenário alternativo	Após o passo 4.3, o técnico opta por excluir um técnico. 4.3.1) O técnico opta por excluir o técnico. 4.3.2) O técnico seleciona a opção “Excluir”; 6.1) O sistema exclui o técnico.
Cenário alternativo	No passo 4.3.1, o técnico opta por editar o nome do técnico. 4.3.1.1) O técnico seleciona a opção “Alterar Nome”; 4.3.1.2) O técnico informa o novo nome do técnico; 4.3.1.3) Retorne para o item 6.
Cenário de exceção	Obrigatoriedade no preenchimento do campo de nome do técnico. O sistema exibe uma mensagem informando a situação.
Cenário de exceção	Obrigatoriedade de que não existe nenhum incidente vinculado ao técnico na base de dados para realizar a exclusão. O sistema exibe uma mensagem informando a situação.
Cenário de exceção	Obrigatoriedade de que não exista outro técnico com a mesma identificação. O sistema exibe uma mensagem informando a situação.
Pós-condições	Um técnico foi cadastrado, editado ou excluído.

APÊNDICE B – Exemplos de consulta a incidentes semelhantes

Neste apêndice, são apresentados outros exemplos de consulta a incidentes semelhantes aos que não foram descritos na seção 3.4.2.

Observa-se na Figura 20, uma consulta de um problema no navegador, especificamente no Google Chrome, onde o mesmo fica abrindo janelas indesejadas conforme a definição indica. Então, têm-se os incidentes de código 11 e 31 que possuem 57.971 % e 57.907 % respectivamente. No entanto, o motivo de ambos, mesmo tendo uma definição muito semelhante ao que o técnico definiu, possuem uma complexidade e risco baixos, bem como custos e tempo para resolução menores ao que o técnico definiu.

Figura 20 - Consulta 3 para demonstrar a análise RBC

The screenshot shows a web application window titled 'Sistema de Gestão de Incidentes'. The main area is 'Consulta Incidentes Semelhantes'. The search criteria are as follows:

- Categoria: Aplicação específicas(Navegador)
- Definição: abrindo janelas indesejadas
- Descrição: (empty)
- Tipo de incidente: Software
- Nome Hardware/Software: Google Chrome
- Nº de Ocorrências: 1
- Custo Solução(R\$): 100
- Tempo p/ Resolução(hrs): 1
- Complexidade de Aplicação da Solução: Média
- Risco de Aplicação da Solução: Médio

Buttons include 'Consultar', 'Consultar Dados de Abertura', and 'Consultar Dados de Encerramento'. Below the search criteria is a table of results:

Sim. (%)	Cód.	Definição	Categoria	Técnico
57,971	11	Navegador apresentando janelas indesejadas durante a na...	Aplicação específicas(Navegador)	Jewiton
57,907	31	Navegador apresentando janelas indesejadas durante a na...	Aplicação específicas(Navegador)	Jewiton
51,874	4	Problema com um plugin para navegador	Aplicação específicas(Navegador)	Jewiton
42,425	9	O navegador Google Chrome não está carregando páginas	Aplicação específicas(Navegador)	Jewiton
42,383	12	O navegador Google Chrome apresentando uma lentidão e...	Aplicação específicas(Navegador)	Jewiton
42,319	7	Navegador fechando eventualmente	Aplicação específicas(Navegador)	Jewiton

Na Figura 21, pode-se observar a uma consulta do mesmo problema, com o ajuste do custo da solução para R\$ 50, o tempo de resolução para 15 minutos e risco e a complexidade para baixos. Com isso, os incidentes de código 11 e 31 têm os seus graus de similaridades alterados para 75.544 % e 75.589 % respectivamente. Assim, tornado o incidente de código 31 o mais semelhante. Isso ocorre devido aos custos de resolução e tempo exatos ao informado pelo técnico, já o incidente de código 11 possui um tempo de 30 minutos e custo de R\$ 80.

Figura 21 - Consulta 4 para demonstrar a análise RBC

Sistema de Gestão de Incidentes

Cadastros Consultas Configurações Ajuda

Consulta Incidentes Semelhantes

Categoria:

Definição:

Descrição:

Tipo de incidente: Nome Hardware/Software:

Nº de Ocorrências:

Custo Solução(R\$):

Tempo p/ Resolução(hrs):

Complexidade de Aplicação da Solução:

Risco de Aplicação da Solução:

Incidentes Semelhantes

Código:

Sim. (%)	Cód.	Definição	Categoria	Técnico
75,589	31	Navegador apresentando janelas indesejadas durante a na...	Aplicação específicas(Navegador)	Jewiton
75,544	11	Navegador apresentando janelas indesejadas durante a na...	Aplicação específicas(Navegador)	Jewiton
60	7	Navegador fechando eventualmente	Aplicação específicas(Navegador)	Jewiton
59,955	12	O navegador Google Chrome apresentando uma lentidão e...	Aplicação específicas(Navegador)	Jewiton
59,925	9	O navegador Google Chrome não está carregando páginas	Aplicação específicas(Navegador)	Jewiton
51,961	4	Problema com um plugin para navegador	Aplicação específicas(Navegador)	Jewiton

A Figura 22, representa uma consulta a um incidente da categoria Atualização(SO), onde o Windows 7 apresenta lentidão ao realizar certas operações. O técnico busca uma solução com custo de R\$ 200, com tempo de resolução de 30 minutos e com complexidade e risco altos. Com isso, o incidente de código 17 aparece com 60.868 % de similaridade, no entanto, a solução proposta por este incidente possui um risco médio, uma complexidade baixa, com tempo de resolução de 2 horas e com custo R\$ 160.

Figura 22 - Consulta 5 para demonstrar a análise RBC

Consulta Incidentes Semelhantes

Categoria:

Definição:

Descrição:

Tipo de incidente: Nome Hardware/Software:

Nº de Ocorrências:

Custo Solução(R\$):

Tempo p/ Resolução(hrs):

Complexidade de Aplicação da Solução:

Risco de Aplicação da Solução:

Incidentes Semelhantes

Código:

Sim. (%)	Cód.	Definição	Categoria	Técnico
60,868	17	Atraso ao desligar, reiniciar ou fazer logoff do Windows 7	Atualização(SO)	Jewiton
46,054	23	Problema ao trocar o idioma do sistema operacional Windo...	Atualização(SO)	Jewiton
45,741	18	Erro 0x8024200D ao instalar o Service Pack 1 do Windows 7	Atualização(SO)	Jewiton
43,372	25	Problemas para realizar o upgrade do Windows 8.1 para o ...	Atualização(SO)	Jewiton
43,065	29	Mensagem de tela azul Windows 8 que não especifica o pro...	Atualização(SO)	Jewiton
42,937	27	Não é possível configurar as opções de rede no Windows 8	Atualização(SO)	Jewiton
42,565	30	Sistema Operacional Windows 8 fica pedindo para atualizar...	Atualização(SO)	Jewiton

Na Figura 23, o técnico insere na consulta um custo de R\$ 80, tempo de 2 horas, com uma complexidade baixa e com risco médio. Portanto, a consulta realizada retorna um grau de similaridade de 78.41 %.

Figura 23 - Consulta 6 para demonstrar a análise RBC

Consulta Incidentes Semelhantes

Categoria:

Definição:

Descrição:

Tipo de incidente: Nome Hardware/Software:

Nº de Ocorrências:

Custo Solução(R\$):

Tempo p/ Resolução(hrs):

Complexidade de Aplicação da Solução:

Risco de Aplicação da Solução:

Incidentes Semelhantes

Código:

Sim. (%)	Cód.	Definição	Categoria	Técnico
78,41	17	Atraso ao desligar, reiniciar ou fazer logoff do Windows 7	Atualização(SO)	Jewiton
63,496	18	Erro 0x8024200D ao instalar o Service Pack 1 do Windows 7	Atualização(SO)	Jewiton
61,127	25	Problemas para realizar o upgrade do Windows 8.1 para o ...	Atualização(SO)	Jewiton
60,782	27	Não é possível configurar as opções de rede no Windows 8	Atualização(SO)	Jewiton
60,749	29	Mensagem de tela azul Windows 8 que não especifica o pro...	Atualização(SO)	Jewiton
60,408	30	Sistema Operacional Windows 8 fica pedindo para atualizar...	Atualização(SO)	Jewiton
60,211	22	Erro 0xc000007b ao abrir aplicativo	Atualização(SO)	Jewiton

Na Figura 24, pode-se observar uma consulta realizada para demonstrar a análise de texto, onde é inserida apenas a expressão `plugin` na definição da consulta, e é feito um filtro por software, no caso o navegador Mozilla Firefox. Na Tabela de

Incidentes são retornados cinco incidentes que se encaixam na categoria e no filtro por software, no entanto, os incidentes de código 14 e 15 recebem um grau de similaridade maior de 59.91 % e 66.272 % respectivamente.

Figura 24 - Consulta 2 para demonstrar análise de texto

The screenshot shows a web application window titled 'Sistema de Gestão de Incidentes'. The main section is 'Consulta Incidentes Semelhantes'. The search criteria are:

- Categoria: Aplicação específicas(Navegador)
- Definição: plugin
- Descrição: (empty)
- Tipo de incidente: Software
- Nome Hardware/Software: Mozilla Firefox
- Nº de Ocorrências: 1
- Custo Solução(R\$): 100
- Tempo p/ Resolução(hrs): 1
- Complexidade de Aplicação da Solução: Média
- Risco de Aplicação da Solução: Médio

 Below the search criteria is a table titled 'Incidentes Semelhantes' with columns: Sim. (%), Cód., Definição, Categoria, and Técnico. The table contains five rows of results.

Sim. (%)	Cód.	Definição	Categoria	Técnico
66,272	15	O plugin do Flash não está funcionando corretamente no Fir...	Aplicação específicas(Navegador)	Jewiton
59,91	14	Erro com plugins no navegador Mozilla Firefox	Aplicação específicas(Navegador)	Jewiton
51,962	13	O navegador Firefox trava ou não está respondendo	Aplicação específicas(Navegador)	Jewiton
42,319	16	O navegador Firefox esta travando frequentemente e tem ap...	Aplicação específicas(Navegador)	Jewiton
42,319	32	O navegador Firefox trava ou não está respondendo	Aplicação específicas(Navegador)	Jewiton

Isso ocorre, pois ambos possuem a expressão *plugin* na definição, mas na Figura 25, o técnico insere na descrição as expressões Java e Adobe Reader. Assim, o incidente de código 14, que antes tinha apenas 59.91 % de similaridade, passou a ter um grau de similaridade de 72.466 %, pois o mesmo é o único incidente analisado que possui as expressões citadas acima.

Figura 25 - Consulta 3 para demonstrar análise de texto

Sistema de Gestão de Incidentes

Cadastros Consultas Configurações Ajuda

Consulta Incidentes Semelhantes

Categoria: Aplicação específicas(Navegador)

Definição: plugin

Descrição: Java, Adobe Reader

Tipo de incidente: Software Nome Hardware/Software: Mozilla Firefox

Nº de Ocorrências: 1

Custo Solução(R\$): 100

Tempo p/ Resolução(hrs): 1

Complexidade de Aplicação da Solução: Média

Risco de Aplicação da Solução: Médio

Consultar

Incidentes Semelhantes

Código: Consultar Dados de Abertura Consultar Dados de Encerramento

Sim. (%)	Cód.	Definição	Categoria	Técnico
72,466	14	Erro com plugins no navegador Mozilla Firefox	Aplicação específicas(Navegador)	Jewiton
54,295	15	O plugin do Flash não está funcionando corretamente no Fir...	Aplicação específicas(Navegador)	Jewiton
51,962	13	O navegador Firefox trava ou não está respondendo	Aplicação específicas(Navegador)	Jewiton
46,296	16	O navegador Firefox esta travando frequentemente e tem ap...	Aplicação específicas(Navegador)	Jewiton
42,319	32	O navegador Firefox trava ou não está respondendo	Aplicação específicas(Navegador)	Jewiton

Na Figura 26, observa-se um exemplo de consulta de um problema da categoria de atualização do sistema operacional, com a definição contendo as expressões `tela azul windows 8`. Então, como o incidente de código 29 é o único a possuir na definição as expressões `tela azul`, a expressão tem um grande peso, levando o grau de similaridade do incidente para 90.897% .

Figura 26 - Consulta 4 para demonstrar análise de texto

Sistema de Gestão de Incidentes

Cadastros Consultas Configurações Ajuda

Consulta Incidentes Semelhantes

Categoria: Atualização(SO)

Definição: tela azul Windows 8

Descrição:

Tipo de incidente: Nenhum Nome Hardware/Software:

Nº de Ocorrências: 1

Custo Solução(R\$): 50

Tempo p/ Resolução(hrs): 0.3

Complexidade de Aplicação da Solução: Baixa

Risco de Aplicação da Solução: Baixo

Consultar

Incidentes Semelhantes

Código: Consultar Dados de Abertura Consultar Dados de Encerramento

Sim. (%)	Cód.	Definição	Categoria	Técnico
90,897	29	Mensagem de tela azul Windows 8 que não especifica o pro...	Atualização(SO)	Jewiton
60,595	23	Problema ao trocar o idioma do sistema operacional Windo...	Atualização(SO)	Jewiton
58,148	27	Não é possível configurar as opções de rede no Windows 8	Atualização(SO)	Jewiton
56,911	30	Sistema Operacional Windows 8 fica pedindo para atualizar...	Atualização(SO)	Jewiton
52,779	25	Problemas para realizar o upgrade do Windows 8.1 para o ...	Atualização(SO)	Jewiton
52,569	17	Atraso ao desligar, reiniciar ou fazer logoff do Windows 7	Atualização(SO)	Jewiton
52,514	18	Erro 0x8024200D ao instalar o Service Pack 1 do Windows 7	Atualização(SO)	Jewiton