UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MOBILE SERVICE DESK: INTEGRANDO SENSIBILIDADE AO CONTEXTO E SISTEMA DE RECOMENDAÇÃO

DISSERTAÇÃO DE MESTRADO

Taciano Balardin de Oliveira

Santa Maria, RS, Brasil

2013

MOBILE SERVICE DESK: INTEGRANDO SENSIBILIDADE AO CONTEXTO E SISTEMA DE RECOMENDAÇÃO

Taciano Balardin de Oliveira

Dissertação apresentada ao Curso de Mestrado Programa de Pós-Graduação em Informática (PPGI), Área de Concentração em Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de
Mestre em Ciência da Computação

Orientadora: Prof. Drª. Roseclea Duarte Medina

Santa Maria, RS, Brasil

2013

Balardin de Oliveira, Taciano

Mobile Service Desk: Integrando Sensibilidade ao Contexto e Sistema de Recomendação / por Taciano Balardin de Oliveira. – 2013.
100 f.: il.; 30 cm.

Orientadora: Roseclea Duarte Medina
Dissertação (Mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS, 2013.

1. Service Desk. 2. Dispositivos Móveis. 3. Sensível ao Contexto. 4. Sistema de Recomendação. I. Duarte Medina, Roseclea. II. Título.

© 2013
Todos os direitos autorais reservados a Taciano Balardin de Oliveira. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.
E-mail: tacianobalardin@gmail.com

Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Informática

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

MOBILE SERVICE DESK: INTEGRANDO SENSIBILIDADE AO CONTEXTO E SISTEMA DE RECOMENDAÇÃO

elaborada por
Taciano Balardin de Oliveira

como requisito parcial para obtenção do grau de
Mestre em Ciência da Computação

COMISSÃO EXAMINADORA:

Roseclea Duarte Medina, Drª.
(Presidente/Orientadora)

Rodrigo dos Santos Keller, Dr. (ULBRA)

Raul Ceretta Nunes, Dr. (UFSM)

Santa Maria, 27 de Março de 2013.

Aos meus familiares e amigos... companheiros de todas as horas...

AGRADECIMENTOS

Esta dissertação marca o fim de uma etapa muito importante na minha vida, desta forma, gostaria de agradecer a todos que de alguma maneira contribuíram para que eu chegasse até aqui. Começando pelos meus pais, que sempre incentivaram a dedicação aos estudos e apoiaram todas minhas decisões. Agradeço as minhas irmãs e minha família que, apesar da distância, sempre torceram e rezaram muito por mim. Amo vocês!

Agradeço a todos os membros do GRECA, em especial aos colegas Érico Amaral, muito dessa conquista devo ao teu incentivo na graduação; Henrique Pedroso, pela parceria e publicações antes e durante o mestrado; Priscila Arenhardt e Camila Possobom, por terem passado pelas mesmas angústias e alegrias, tanto nas aulas quanto no laboratório; Ao Ricardo Gomes, pelo apoio no decorrer deste trabalho; Aos amigos Gleizer Voss, Giani Petri, Felipe Nunes, Victor Orozco e Andreia Mühlbeier agradeço por todos momentos de trabalho e descontração dentro e fora da universidade, vocês são DEZ! A minha orientadora Rose, que teve um papel fundamental para que este trabalho fosse concluído, uma pessoa muito especial que trata todos seus orientandos de uma forma muito respeitosa, amigável e sempre com aquela alegria. Muito obrigado!

Aqueles "malucos", até então desconhecidos, que tornaram-se grandes amigos: Guinho Goulart, Diego Neves, Guilherme Fusieger e todos aqueles que em algum momento estiveram comigo durante os anos em que estive em Santa Maria, valeu! O agradecimento e a saudade daqueles que viram o início desta jornada, mas infelizmente, não puderam acompanhar de perto este fim.

Minha gratidão aos professores, por todo esforço realizado para dividirem suas sabedorias nessa caminhada rumo ao aprendizado, em especial a Raul Ceretta, Rodrigo Keller e Giovani Librelotto, que compõem a comissão examinadora e também contribuíram na prévia deste trabalho; A UFSM pela oportunidade do estudo gratuito e de qualidade e ao CPD pela colaboração neste trabalho; Por fim, agradeço àqueles grandes amigos de Cachoeira que sempre esforçaram-se para que eu saísse da rotina.

Muito obrigado!

"Não existe triunfo sem perda, não há vitória sem sofrimento, não há liberdade sem sacrifício."

— SENHOR DOS ANÉIS, O RETORNO DO REI

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria

MOBILE SERVICE DESK: INTEGRANDO SENSIBILIDADE AO CONTEXTO E SISTEMA DE RECOMENDAÇÃO
AUTOR: TACIANO BALARDIN DE OLIVEIRA
ORIENTADORA: ROSECLEA DUARTE MEDINA
Local da Defesa e Data: Santa Maria, 27 de Março de 2013.

A gerência dos problemas ocorridos em ambientes que fazem uso da Tecnologia da Informação (TI), aliada a necessidade de uma resposta rápida das equipes de suporte, faz com que organizações necessitem de sistemas para gerenciamento desses incidentes. O *Service Desk* apresenta-se como uma boa solução para centralizar estes registros. Conceitos de computação baseada em contexto, sistemas de recomendação, computação móvel podem incrementar estes aplicativos. Portanto, o objetivo deste trabalho é projetar e implementar um sistema de *Service Desk* móvel, denominado Mobile Service Desk (MSD), que agrega funcionalidades de sensibilidade ao contexto, tais como localização do usuário, experiência do técnico e contexto temporal. Além disso, está integrado à ferramenta um sistema de recomendação, que armazena interações passadas e as sugere como possível solução para novos incidentes similares ocorridos no ambiente gerenciado. Como contribuições deste trabalho, além do projeto do sistema que visa redução de tempo com deslocamentos desnecessários e otimização do alocamento de técnicos, foram comparados algoritmos para análise de similaridade e aplicado a norma NBR 9241-11 para avaliação de usabilidade de alguns produtos de Service Desk.

Palavras-chave: Service Desk. Dispositivos Móveis. Sensível ao Contexto. Sistema de Recomendação.

ABSTRACT

Master's Dissertation
Post-Graduate Program in Informatics
Federal University of Santa Maria

MOBILE SERVICE DESK: INTEGRATING CONTEXT AWARENESS AND SYSTEM RECOMMENDATION

AUTHOR: TACIANO BALARDIN DE OLIVEIRA
ADVISOR: ROSECLEA DUARTE MEDINA
Defense Place and Date: Santa Maria, March 27st, 2013.

The management of problems occurred in environments that make use of Information Technology (IT), coupled with the need for a rapid response support teams, makes organizations require systems to manage these incidents. The Service Desk presents itself as a good solution to centralize these records. Concepts based computing context, recommender systems, mobile computing can enhance these applications. Therefore, the aim of this work is to design and implement a system of Service Desk Mobile, called Mobile Service Desk (MSD), which adds context awareness features such as user location, technical experience and temporal context. Moreover, the tool is integrated into a recommendation system, which stores past interactions and suggests as a possible solution for new similar incidents occurring in the managed environment. As contributions of this work, in addition to system design that aims to reduce unnecessary time-shift and optimize the allocation of technical, algorithms were compared for similarity analysis and applied to NBR 9241-11 for usability evaluation of some products Service Desk.

Keywords: Service Desk, Mobile Device, Context Awareness, Recommendation System.

LISTA DE FIGURAS

Figura 2.1 – Segmentos do sistema de GPS (PAZ, 1997) 22
Figura 2.2 – Exemplo de *script* para geolocalização adaptado de (W3CSCHOOLS, 2012) 23
Figura 2.3 – Modelo de Método Híbrido para Sistema de Recomendação (BALABANOVIć; SHOHAM, 1997) 27
Figura 4.1 – Simulação de incidentes em três prédios diferentes (LOBO, 2011) 39
Figura 4.2 – Sistema Operacional de Dispositivos Móveis no Brasil (GOOGLE, 2012) . . . 40
Figura 4.3 – Sistema Operacional de Dispositivos Móveis nos EUA (GOOGLE, 2012) . . . 40
Figura 4.4 – Fluxograma para Recomendação de Soluções 44
Figura 4.5 – Caso de uso do sistema *Mobile Service Desk* 45
Figura 4.6 – Funcionamento do *Mobile Service Desk* 47
Figura 4.7 – Modelo Entidade Relacionamento do *Mobile Service Desk* 48
Figura 5.1 – Síntese do *Mobile Service Desk* 49
Figura 5.2 – Cadastro de Técnicos no *Mobile Service Desk* 51
Figura 5.3 – Cadastro de Prédios e Salas no *Mobile Service Desk* 53
Figura 5.4 – Código para detectar a localização do usuário 54
Figura 5.5 – Solicitação de Localização no *Mobile Service Desk* 55
Figura 5.6 – Cálculo de distância baseado na Fórmula de Haversine 56
Figura 5.7 – Código de pré-processamento de texto 57
Figura 5.8 – Função que passa o texto para minúsculo e remove alguns caracteres 58
Figura 5.9 – Função para remoção de *stopwords* 58
Figura 5.10 – *Script* Python para técnica de *stemming* 58
Figura 5.11 – Função para leitura e gravação do algoritmo de *stemming* 59
Figura 6.1 – Número mensal de chamados abertos no sistema da CAU em 2012 (CPD, 2012). 62
Figura 6.2 – Prazos para atendimento de chamados na CAU em 2012 (CPD, 2012). 62
Figura 6.3 – Mapa do Campus da UFSM, adaptado de (COPERVES, 2012) 64
Figura 6.4 – Classificação dos chamados abertos para teste no *Mobile Service Desk* 66
Figura 6.5 – Lista de Chamados para o técnico Atendimento 1 67
Figura 6.6 – Lista de Chamados para o técnico Atendimento 2 68
Figura 6.7 – Lista de Chamados para o técnico Atendimento 3 69
Figura 6.8 – Desempenho dos Algoritmos de Análise de Similaridade Testados (em segundos) 75
Figura 6.9 – Facilidade de Aprender: Foram Aprendidas Todas as Tarefas? 78
Figura 6.10 – Facilidade de Aprender: Qual a Escala de Satisfação em Relação à Facilidade de Aprender a Utilizar o Sistema? 78
Figura 6.11 – Facilidade de Aprender: Média de Tempo Gasto para Realizar as Tarefas . . . 79
Figura 6.12 – Usabilidade Global: Qual a Escala de Satisfação em Relação a Usabilidade Global do Sistema? 80
Figura 6.13 – Usabilidade Global: Média de Tempo Gasto para Realizar as Tarefas 81
Figura 6.14 – Legibilidade: É Possível Ler os Textos e Navegar entre as Áreas do Sistema? 82
Figura 6.15 – Legibilidade: Qual a Escala de Satisfação em Relação à Legibilidade e Navegabilidade no Sistema 82
Figura 6.16 – Legibilidade: Média de Tempo Gasto para Realizar as Tarefas 83
Figura C.1 – Medidas Para as Propriedades Desejáveis do Produto 99

LISTA DE TABELAS

Tabela 2.1 – Etapas do removedor de sufixos, adaptado de ORENGO; HUYCK (2001) . . . 29
Tabela 6.1 – Descrição do problema e parecer final de chamados da CAU 61
Tabela 6.2 – Relação de Usuários do *Mobile Service Desk* 65
Tabela 6.3 – Relação de Chamados Inseridos no *Mobile Service Desk* 65
Tabela 6.4 – Textos para teste dos algoritmos de Análise de Similaridade 70
Tabela 6.5 – Resultado do pré-processamento dos Textos 70
Tabela 6.6 – Teste do Chamado 1 para o Algoritmo Jaro-Winkler 71
Tabela 6.7 – Teste do Chamado 1 para o Algoritmo Levenshtein Distance 71
Tabela 6.8 – Teste do Chamado 1 para o Algoritmo String Similarity 71
Tabela 6.9 – Teste do Chamado 1 para o Algoritmo *similar_text()* 71
Tabela 6.10 – Teste do Chamado 2 para o Algoritmo Jaro-Winkler 72
Tabela 6.11 – Teste do Chamado 2 para o Algoritmo Levenshtein Distance 72
Tabela 6.12 – Teste do Chamado 2 para o Algoritmo String Similarity 72
Tabela 6.13 – Teste do Chamado 2 para o Algoritmo *similar_text()* 72
Tabela 6.14 – Teste do Chamado 3 para o Algoritmo Jaro-Winkler 73
Tabela 6.15 – Teste do Chamado 3 para o Algoritmo Levenshtein Distance 73
Tabela 6.16 – Teste do Chamado 3 para o Algoritmo String Similarity 74
Tabela 6.17 – Teste do Chamado 3 para o Algoritmo *similar_text()* 74
Tabela 6.18 – Comparativo entre ferramentas de *Service Desk* 84
Tabela B.1 – *Stoplist* do Mobile Service Desk 97

LISTA DE APÊNDICES

APÊNDICE A – Questionário para avaliação seguindo a NBR 9241-11 96
APÊNDICE B – *Stoplist* utilizada no *Mobile Service Desk* 97
APÊNDICE C – Especificações e Propriedades Desejáveis do Produto - NBR 9241-11 98
APÊNDICE D – Dispositivos suportados pelo *framework* jQuery Mobile. 100

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
CAU	Central de Atendimento ao Usuário
GNU	*GNU Operating System*
GPS	*Global Positioning System*
GRECA	Grupo de Redes e Computação Aplicada
HTML	*HyperText Markup Language*
IDE	*Integrated Development Environment*
ITIL	*Information Technology Infrastructure Library*
MSD	Mobile Service Desk
MSN	*Messenger*
PHP	PHP *Hypertext Preprocessor*
SGBD	Sistema de Gerenciamento de Banco de Dados
SPOC	*Single Point of Contact*
TI	Tecnologia da Informação
UFSM	Universidade Federal de Santa Maria
UML	*Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO 16
1.1 Motivação 18
1.2 Problema de Pesquisa 18
1.3 Objetivo Geral 19
1.4 Objetivos Específicos 19
1.5 Principais Contribuições 19
1.6 Organização da Dissertação 20
2 REVISÃO BIBLIOGRÁFICA 21
2.1 Sistemas de *Service Desk* 21
2.2 Sistema de posicionamento global 21
2.2.1 HTML 5 e a API de geolocalização 22
2.3 Computação Ubíqua e Sensível ao Contexto 23
2.4 Gestão do Conhecimento 24
2.5 Sistemas de Recomendação 25
2.5.1 Método Baseado em Conteúdo 26
2.5.2 Método de Filtragem Colaborativa 26
2.5.3 Método Híbrido 27
2.6 Pré-Processamento de Texto 27
2.6.1 Remoção de *Stopwords* 28
2.6.2 *Stemming* 28
2.6.2.1 Algoritmo de Orengo 29
2.6.2.2 Algoritmo de Porter 29
2.7 Algoritmos de Similaridade entre *Strings* 30
2.7.1 Algoritmo Levenshtein Distance 30
2.7.2 Algoritmo Boyer-Moore 31
2.7.3 Algoritmo Rabin-Karp 31
2.7.4 Algoritmo Jaro-Winkler 31
2.8 Trabalhos Relacionados 31
3 METODOLOGIA DO TRABALHO 35
4 *MOBILE SERVICE DESK*: PROPOSTA 38
4.1 Funcionamento em Diversos Dispositivos 39
4.2 Sensibilidade ao Contexto 41
4.3 Ordenação de chamados 41
4.3.1 Contexto de *Expertise* 42
4.3.2 Prioridade de Chamados 42
4.3.3 Contexto de Localização 42
4.4 Relacionamento de Chamados 43
4.5 Recomendação de Soluções 44
4.6 Modelagem do MSD 45
4.6.1 Nível Usuário 45
4.6.2 Nível Classificador 46
4.6.3 Nível Suporte 46
4.6.4 Nível Gestor 46

5 *MOBILE SERVICE DESK: IMPLEMENTAÇÃO* 49
5.1 Dispositivos Móveis 50
5.2 Sensibilidade ao Contexto 50
5.2.1 Cadastro de Técnicos 50
5.2.1.1 Definição da *Expertise* 51
5.2.1.2 Definição de Jornada de Trabalho 52
5.2.2 Cadastro de Prédios e Salas 52
5.2.3 Contexto de Localização 53
5.2.3.1 Calculando a Distância entre Técnico e Atendimentos 55
5.2.4 Classificação de Chamados 56
5.3 Atendimento de Chamados 56
5.4 Relacionamento de Chamados 57
5.4.1 Pré-Processamento de Texto 57
5.4.2 Análise de Similaridade 60
6 *MOBILE SERVICE DESK*: RESULTADOS 61
6.1 Validação do Sistema 61
6.1.1 Metodologia 63
6.1.2 Plano de testes 64
6.1.3 Execução do plano de testes 66
6.1.3.1 Classificação dos chamados 66
6.1.3.2 Ordenação dos chamados 66
6.1.3.2.1 Técnico Atendimento 1 67
6.1.3.2.2 Técnico Atendimento 2 68
6.1.3.2.3 Técnico Atendimento 3 69
6.1.4 Análise de Similaridade 69
6.2 Avaliação de Interfaces Utilizando a NBR 9241-11 75
6.2.1 Facilidade de Aprender 77
6.2.2 Usabilidade Global do Sistema 79
6.2.3 Legibilidade 81
6.3 Comparativo entre ferramentas 84
7 CONCLUSÕES E TRABALHOS FUTUROS 86
7.1 Trabalhos Futuros 88
REFERÊNCIAS 89
APÊNDICES 95
D.1 Suporte Integral 100
D.2 Suporte sem Ajax 100
D.3 HTML Básico 100

1 INTRODUÇÃO

O aumento da dependência das organizações na utilização da tecnologia de informação (TI) vem tornando o gerenciamento de serviços de TI dentro desses ambientes uma atividade cada vez mais importante (MAGALHAES; PINHEIRO, 2007). Na ocorrência de problema em algum sistema informatizado presente nestes locais gerenciados (e.g, computador, impressora, *software*, redes de comunicação ou qualquer dispositivo que ocasione o funcionamento anormal dos serviços de TI), a expectativa é que o usuário tenha uma resposta rápida da equipe de suporte para que os prejuízos causados possam ser minimizados (LOBO, 2011).

Para que os problemas pudessem ser centralizados e posteriormente solucionados por técnicos responsáveis por estas tarefas, criou-se o conceito de serviço de atendimento aos usuários de TI, que foi inicialmente denominado de *Help Desk* (CAVALARI; COSTA, 2005). Entretanto, atualmente esta área absorveu outros serviços e passou a chamar-se de *Service Desk*, tratando-se de uma versão estendida do *Help Desk* e oferecendo uma quantidade maior de serviços (JANTTI; KALLIOKOSKI, 2010).

Um desafio que atinge os responsáveis por estes ambientes de gerenciamento é que em diversas organizações existe uma alta rotatividade dos recursos humanos de TI, também conhecida por *turnover*. Em 2010, a rotatividade do pessoal de TI em todo o mundo era de apenas 3%. Em 2011, de acordo com o instituto de pesquisas Gartner (GARTNER, 2011a), saltou para 5%. Logo, a saída de um funcionário representa uma perda de capital humano e gera um custo de substituição (i.e, recrutamento, seleção, contratação e treinamento) que pode ser alto. Além disso, existe a dificuldade de transferir o conhecimento e experiência entre os colaboradores da área (WANG et al., 2011).

Outro fator que deve ser considerado em um *Service Desk*, diz respeito à alocação de recurso humano que tenha o perfil adequado para resolução dos diferentes tipos de problemas. Assim, um *Service Desk* onde os técnicos trabalham sem planejamento algum, atendendo aos chamados desordenadamente, ou sem ter a *expertise* (i.e, experiência, prática e conhecimento técnico) necessária para resolver o problema relatado, pode ter sérios problemas com a perda de tempo, com deslocamento desnecessário ou com a alocação errada de membros da equipe (LOBO, 2011).

Uma possível solução para o problema de *turnover* é a integração de uma ferramenta de *Service Desk* à um sistema de recomendação, onde as soluções aplicadas pelos técnicos em

situações anteriores ficam retidas no banco de dados do sistema e, a partir de um novo chamado com características semelhantes, são apresentadas como possível solução para o incidente.

A computação baseada em contexto usa informações sobre um usuário ou ambiente para melhorar a qualidade de interação com o sistema que, munido destes dados, age de forma proativa, antecipando as necessidades do utilizador e provendo uma maior adaptação às suas atividades em tempo de execução. De acordo com pesquisa de (GARTNER, 2011b), até o ano de 2013 as aplicações sensíveis ao contexto aparecerão em áreas como: serviços baseados em localização, realidade aumentada em dispositivos móveis e comércio móvel.

Além disso, os mesmos estudos de (GARTNER, 2011b) apontam que existe uma tendência de que, até 2013, os dispositivos móveis ultrapassem os computadores pessoais como meios mais comuns para acesso à *web*. Tendência esta que, segundo (ACCENTURE, 2012), vem se confirmando, pois esses aparelhos tem se tornado o meio primário para acesso à rede, principalmente devido ao crescimento do uso de *smartphones* como meio de conexão.

Por sua vez, a computação baseada em contexto pode auxiliar na alocação de chamados, para que sejam solucionados utilizando informações do ambiente como: localização, à *expertise* e também o horário de expediente de cada técnico que realiza atendimento. Além disso, a computação móvel possibilita a utilização do sistema a partir de qualquer dispositivo compatível, em qualquer lugar e a qualquer momento.

Esta dissertação é continuidade de pesquisas realizadas pelo GRECA (Grupo de Redes e Computação Aplicada) da UFSM (Universidade Federal de Santa Maria) considerando os trabalhos de (LOBO, 2011), que adiciona características de computação sensível ao contexto em uma ferramenta de *Service Desk* integrada a um Portal Corporativo e (PEDROSO; OLIVEIRA; MEDINA, 2012), que tem como objetivo construir um módulo destinado a reter uma parte da expertise gerencial, criando uma base de informação (i.e, agregando às causas do incidente a descrição textual de como o gestor o solucionou), em uma ferramenta de monitoramento de ataques, instabilidades e indisponibilidades da infraestrutura de rede.

Este trabalho tem como principal objetivo projetar e implementar uma ferramenta de *Service Desk* móvel, denominada *Mobile Service Desk*, que utiliza características de contexto para auxiliar na alocação de técnicos aos chamados. Além disso, munir esta ferramenta com um sistema de recomendação, para que o capital intelectual, gerado através dos atendimentos realizados pela equipe de suporte, fique retido no sistema e seja recomendado como possível solução para novos chamados similares.

1.1 Motivação

A necessidade de elevar o nível de gerenciamento nos ambientes de TI e os problemas da rotatividade pessoal (*turnover*) nestes locais fomentam a pesquisa de soluções que auxiliem nestas situações e também sirvam como base para atingir um gerenciamento proativo. Através disso, acredita-se que agregando sensibilidade ao contexto e sistema de recomendação em uma ferramenta de *Service Desk* irá qualificar e agilizar o atendimento dos incidentes, além de valorizar o capital intelectual por meio da retenção do conhecimento empregado pelos colaboradores na resolução de incidentes ocorridos nestes ambientes e funcionar de acordo com as preferências, características e atividades do usuário.

A relevância científica desta pesquisa recai sobre o estudo e possibilidade de adaptar as soluções propostas pela computação ciente ao contexto em um sistema de *Service Desk*, visando a redução do desperdício de tempo e uma melhor alocação dos recursos humanos disponíveis. Além disso, a aplicação de técnicas de pré-processamento de texto e análise de similaridade no texto de abertura de cada chamado, aliado ao armazenamento da descrição do método utilizado pelo técnico para solucionar um incidente, da suporte a funcionalidade de Sugestão de Soluções e poderá servir como auxílio para técnicos solucionarem problemas similares.

1.2 Problema de Pesquisa

Perante a alta demanda de serviços no ambiente tecnológico, aliada a complexidade para manter tais locais em pleno funcionamento e levando em consideração as equipes que, além de reduzidas e com problemas de rotatividade pessoal, trabalham no limite de suas capacidades de produção, este trabalho pretende apontar uma possível solução para o seguinte problema de pesquisa:

É possível otimizar o trabalho das equipes de *Service Desk*, fazendo uso das informações de contexto, a fim de reduzir o tempo com deslocamentos desnecessários e elaborar um sistema de recomendação que indique possíveis soluções para problemas similares, a fim de auxiliar os técnicos na resolução de incidentes, contribuindo para a disponibilidade da infraestrutura tecnológica e com a competitividade das organizações?

1.3 Objetivo Geral

O objetivo principal deste trabalho consiste em provar que a hipótese da integração de características de sensibilidade ao contexto (e.g, localização do técnico ou usuário, expertise do técnico, tempo) à um sistema de *Service Desk*, de forma que seja possível otimizar o número de atendimentos realizados através de uma melhor alocação dos chamados com base nestas informações de contexto. Além disso, demonstrar que através de um sistema de recomendação é possível indicar aos técnicos possíveis soluções para novos incidentes similares.

1.4 Objetivos Específicos

Este projeto tem como objetivos específicos um conjunto de atividades, que permitem alcançar o objetivo geral. Dentre elas destacam-se:

- Levantar e comparar as funcionalidades das principais ferramentas de *Help Desk*/*Service Desk* disponíveis atualmente com a que é proposta neste trabalho.
- Comparar algoritmos de análise de similaridade, medindo a qualidade dos resultados e seus desempenhos.
- Modelar o sistema proposto, de forma que agregue características de contexto às suas funcionalidades, possibilite o acesso tanto por meio de dispositivos móveis quanto por computadores pessoais, além de propor técnicas para que seja possível a recomendação de soluções à problemas semelhantes;
- Validar o sistema proposto através de um plano de testes e também utilizando uma norma NBR baseada em ISO.

1.5 Principais Contribuições

Este trabalho se propõe as seguintes contribuições:

- Elaborar mecanismos válidos para utilização de informações de contexto em uma ferramenta de *Service Desk*, de forma que seja possível uma melhor alocação de recursos humanos;

- Descrever algoritmos para análise de similaridade e compará-los no que diz respeito a desempenho e qualidade de resultados, para que este seja utilizado em um sistema de recomendação que também é uma contribuição deste trabalho;
- Aplicar uma metodologia para avaliação de interfaces entre sistemas de *Service Desk* e discutir os resultados obtidos.

1.6 Organização da Dissertação

A dissertação está organizada da seguinte forma. No Capítulo 2 consta a Revisão Bibliográfica, que aborda temas como Sistemas de *Service Desk*, Sistema de Posicionamento Global, HTML 5 e API de Geolocalização, Computação Ubíqua e Sensível ao Contexto, Gestão do Conhecimento, Sistemas de Recomendação, Pré-Processamento de Texto e Análise de Similaridade, além dos trabalhos relacionados à esta pesquisa. No Capítulo 3 é apresentada a metodologia do trabalho. Já no Capítulo 4 consta a proposta do *Mobile Service Desk*, com suas funcionalidades, características e a modelagem do sistema. O Capítulo 5 apresenta a implementação da proposta e no Capítulo 6 estão os resultados obtidos através desta pesquisa, ou seja, a validação do sistema, avaliação da interface, seguindo a norma NBR 9241-11 e um comparativo de funcionalidades entre ferramentas de *Service Desk*. Por fim, o Capítulo 7 apresenta a conclusão do trabalho e os trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo faz uma revisão literária de termos relacionados a esta pesquisa e que também servem como base para o desenvolvimento do trabalho, tais como: sistemas de *Service Desk*, sistema de posicionamento global e sua utilização em HTML, computação sensível ao contexto, que é uma área de pesquisa da computação ubíqua, técnicas de pré-processamento de texto e algoritmos de análise de similaridade que dão suporte ao sistema que recomenda chamados similares. Estes itens estão descritos nas próximas seções.

2.1 Sistemas de *Service Desk*

Com o aumento da demanda empresarial e da globalização, cada vez mais as organizações precisam se certificar da qualidade dos serviços executados para obterem maiores chances no mercado. Com isso, o objetivo de um *Service Desk* é prover aos usuários de TI um SPOC (*Single Point of Contact*), ou ponto único de contato, vital para uma comunicação efetiva entre os usuários e as equipes que gerenciam a TI em uma organização (OCG, 2007). Sua principal missão é o restabelecimento da operação normal dos serviços dos usuários o mais rápido possível, minimizando o impacto nos negócios causados por falhas de TI (COHEN, 2011). Além disso, o serviço de atendimento mantém os usuários informados sobre o progresso na resolução de incidentes, mudanças e eventos relacionados (JANTTI; KALLIOKOSKI, 2010).

O funcionamento de um sistema de *Service Desk* se da através da abertura de chamados ou *tickets*. A partir deste momento, sempre que houver um chamado em aberto, o mesmo é gerenciado para que seja atendido. Além disso, esses sistemas também podem ser baseados em práticas de alguma metodologia para manutenção de serviços de TI, como, por exemplo, o ITIL (*Information Technology Infrastructure Library*) (ITIL, 2007).

Uma lista com trabalhos relacionados a *Service Desk* e algumas ferramentas gratuitas disponíveis na atualidade estão descritas na seção 2.8.

2.2 Sistema de posicionamento global

O sistema de posicionamento global, ou GPS (*Global Positioning System*), é um sistema de navegação por satélite que fornece a um dispositivo receptor a sua posição, a qualquer momento e em qualquer lugar na Terra, desde que o receptor se encontre no campo de visão de três

satélites GPS. Esse sistema foi elaborado com recursos do Departamento de Defesa dos Estados Unidos para fornecer a posição instantânea e a velocidade de um ponto sobre a superfície terrestre, ou próximo a ela. Inicialmente para fins militares, o GPS é hoje utilizado gratuitamente por diversos segmentos da sociedade civil e seu uso tende a se popularizar cada vez mais (PAZ, 1997).

De acordo com (PAZ, 1997), esse sistema é composto por três segmentos apresentados na Figura 2.1: o segmento espacial, composto por satélites artificiais da Terra que emitem sinais eletromagnéticos; o segmento de controle, composto pelas estações terrestres que mantêm os satélites em funcionamento; e o segmento dos usuários, composto pelos receptores, que captam os sinais enviados pelos satélites e calculam sua posição.

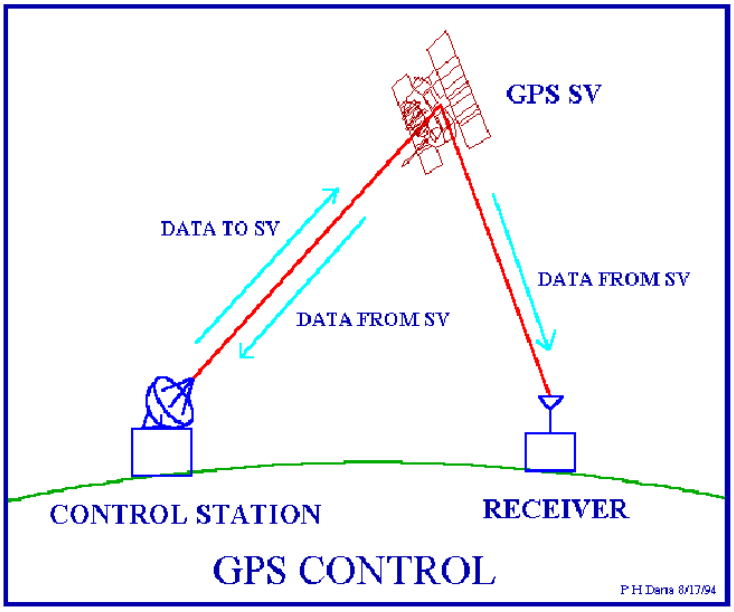

Figura 2.1 – Segmentos do sistema de GPS (PAZ, 1997)

2.2.1 HTML 5 e a API de geolocalização

O HTML5 representa o novo padrão HTML, como seus antecessores (HTML 4.01 e XHTML 1.1) é um padrão para estruturação e representação de conteúdo na *web*. Atualmente encontra-se em desenvolvimento, mas já existem sites e aplicações que são implementados utilizando suas novas funcionalidades (PEJIC; PEJIC; COVIC, 2010). Uma das principais características do HTML5 é a introdução de novos elementos e atributos para utilização. Isso estende, melhora e racionaliza a marcação disponível para documentos, introduz marcações e interfaces de programação de aplicativos (APIs) para aplicações *web* complexas. Além disso,

muitos recursos do HTML5 tem sido construídos com a consideração de serem capazes de executar em dispositivos de menor potência, como *smartphones* e *tablets* (W3C, 2012). Neste grupo de novas APIs adicionado ao HTML5 encontra-se a Geolocation API (PEJIC; PEJIC; COVIC, 2010).

A Figura 2.2 apresenta um exemplo simples de *script* Geolocation, sem tratamento de erro. Os passos executados pelo *script* são os seguintes:

- Verifica se existe suporte à geolocalização;
- Se suportado, executa o método *getCurrentPosition()*. Se não, exibe uma mensagem para o utilizador;
- Se o *getCurrentPosition()* é bem sucedido, ele retorna um objeto contendo as coordenadas para a função especificada no parâmetro *showPosition*;
- A função *showPosition()* obtém e mostra a latitude e longitude;

```
1  <script>
2      var x=document.getElementById("demo");
3      function getLocation()
4      {
5          if (navigator.geolocation)
6          {
7              navigator.geolocation.getCurrentPosition(showPosition);
8          }
9          else{x.innerHTML="Geolocation is not supported by this browser.";}
10     }
11     function showPosition(position)
12     {
13         x.innerHTML="Latitude: " + position.coords.latitude +
14             "<br>Longitude: " + position.coords.longitude;
15     }
16 </script>
```

Figura 2.2 – Exemplo de *script* para geolocalização adaptado de (W3CSCHOOLS, 2012)

O sistema de posicionamento global e a API de geolocalização serão aplicados a esta pesquisa para dar suporte ao contexto de localização, a fim de otimizar a alocação de chamados aos técnicos evitando deslocamentos desnecessários.

2.3 Computação Ubíqua e Sensível ao Contexto

O termo computação ubíqua, definido pelo cientista Mark Weiser em 1991 (WEISER, 1991) como um novo paradigma onde a computação deve estar invisível para o usuário. Em consonância com LOBO (2011), "a computação ubíqua prevê a integração e comunicação de diversos dispositivos e recursos (i.e, *software* e *hardware*) em um ambiente real de forma que o

usuário possa realizar alguma atividade sem ter a consciência da utilização dos recursos computacionais".

De acordo com DEY (2001), contexto é qualquer informação que caracteriza a situação de uma entidade, que pode ser pessoa, lugar ou objeto, considerada relevante em uma interação entre usuário e aplicação. A computação ciente de contexto é uma área de pesquisa da computação ubíqua, que tem o objetivo de prover o acesso de sistemas computacionais a essas informações de contexto, ampliando a comunicação entre ser humano e sistemas computacionais, assim permitindo o desenvolvimento de serviços computacionais mais úteis e adaptáveis (DEY, 2001).

Um sistema é considerado ciente de contexto se utilizar elementos do contexto para fornecer informações e/ou serviços relevantes para o usuário, sendo a relevância dependente das tarefas a serem realizadas pelo mesmo (DEY, 2001) (JARDIM, 2006). Assim, espera-se que sistemas computacionais cientes de contexto não somente respondam quanto ao estado social e cognitivo do usuário, mas também antecipem suas necessidades (JARDIM, 2006) (SIEWIOREK, 2002).

Além do contexto de localização, relatado na seção anterior, a utilização de outras informações de contexto foram aplicadas neste trabalho, tais como contexto de *expertise* que é utilizado para alocar chamados aos técnicos que possuam a *expertise* (i.e, experiência, prática e conhecimento técnico) necessária para solucionar o problema, evitando perca de tempo com deslocamentos desnecessários e contexto de tempo que indica se o técnico está em horário de trabalho, para que não sejam alocados chamados a ele nos momentos em que está fora de expediente.

2.4 Gestão do Conhecimento

O conhecimento é uma mistura de experiência estruturada, valores, informação contextual e discernimento que fornece uma estrutura para avaliar e incorporar novas experiências e informações. A gestão do conhecimento é aplicada nas organizações com o objetivo de capturar, analisar, interpretar, organizar, mapear e difundir informações, para que ela seja útil e esteja disponível como conhecimento (DAVENPORT; PRUSAK, 1998).

Sistemas de *Service Desk* geralmente usam bancos de dados para armazenar interações passadas (e.g, descrições de um problema e as soluções recomendadas para sua resolução) (WANG et al., 2011). Logo, a partir de um novo chamado do usuário, um sistema de *Service*

Desk pode descobrir se chamados semelhantes foram processados anteriormente e sugerir a solução aplicada no item similar para este que ainda não foi solucionado.

Neste trabalho, é proposta a utilização do texto que descreve a solução aplicada pelo técnico à resolução de um problema para auxiliar na solução de outro que tenha características semelhantes, para isso é integrado um sistema de recomendação ao sistema de *Service Desk*. O objetivo disto é manter um registro das ações realizadas pelos técnicos para solucionar um incidente, a fim de que estas informações possam ajudar um membro ingressante na equipe, auxiliando-o para que possa assumir de forma plena as tarefas do antigo, tentando minimizar o impacto da rotatividade, ou até mesmo para auxiliar técnicos mais experientes a solucionarem novos problemas similares.

2.5 Sistemas de Recomendação

Um sistema de recomendação combina técnicas computacionais para selecionar itens personalizados com base nos interesses dos usuários e conforme o contexto no qual estão inseridos. De acordo com ADOMAVICIUS; TUZHILIN (2005), esse tema tornou-se uma importante área de pesquisa a partir dos primeiros trabalhos surgidos sobre filtragem colaborativa na década de 90. Ainda segundo o autor, o interesse por esta área se mantém em alta porque possui um grande número de problemas de pesquisa e também pela abundância de aplicações que auxiliam os usuários a lidar com a sobrecarga de informações e fornecem recomendações, conteúdos e serviços personalizados para eles.

O problema de um sistema de recomendação é formalizado a partir de um conjunto C de todos os usuários de um determinado sistema, S é o conjunto de todos os possíveis itens que podem ser recomendados como livros, filmes, restaurantes etc. Seja u a função utilidade que mede o quão útil é um determinado item s para um determinado usuário c, onde *R* é um conjunto totalmente ordenado (i.e., $u : C \times S \rightarrow R$). Então, para cada usuário $c \in C$, procura-se um item $s' \in S$ que maximiza a utilidade do usuário (ADOMAVICIUS; TUZHILIN, 2005). Isto pode ser expressado pela equação a seguir:

$$\forall c \in C, s'_c = \arg\max_{s \in S} u(c, s)$$

Com base na forma como são realizadas as recomendações, sistemas de recomendação são geralmente classificados da seguinte forma: (i) baseado em conteúdo (*content-based*), onde se busca recomendar itens semelhantes aqueles que já foram de interesse do usuário; (ii) fil-

tragem colaborativa (*collaborative filtering*), que atua identificando usuários com preferências semelhantes ao atual e recomenda itens que foram de interesse daquele usuário similar; (iii) abordagem híbrida (*hybrid-approaches*), esse método resulta da combinação entre o método colaborativo e o baseado em conteúdo; (BALABANOVIć; SHOHAM, 1997).

2.5.1 Método Baseado em Conteúdo

O método baseado em conteúdo calcula a utilidade de um item s para um usuário c, com base nas utilidades de itens "similares" a s para o mesmo usuário c. O cálculo da similaridade entre itens é realizado a partir do uso de um conjunto de atributos que caracterizam cada item (ADOMAVICIUS; TUZHILIN, 2005). Para possibilitar este tipo de recomendação é necessário que se encontrem associações entre esses itens (REATEGUI; CAZELLA, 2005). Este método tem origem na recuperação da informação e filtragem de informação (ADOMAVICIUS; TUZHILIN, 2005).

Esta técnica possui alguns problemas, como: (i) a análise limitada de conteúdo, visto que se está limitado às características explicitamente associadas aos objetos recomendados; (ii) excesso de especificação, em casos onde o sistema recomenda apenas itens com similaridade alta, serão sempre recomendados itens altamente parecidos, sem que haja uma diversificação; (iii) presença de novos usuários, os quais deverão avaliar determinado número de itens antes de receber uma recomendação útil (ADOMAVICIUS; TUZHILIN, 2005).

2.5.2 Método de Filtragem Colaborativa

O objetivo da filtragem colaborativa é recomendar novos itens ou predizer a utilidade de um item para um determinado usuário, tendo como base os dados dos usuários similares à ele. Desta forma, o usuário receberá recomendações de itens que pessoas com gostos similares aos dele preferiram no passado. Este método é subdividido em duas categorias: a primeira chamada de *memory-based*, e a segunda chamada de *model-based*. O cálculo de utilidade de um item s em relação a um usuário c é feita a partir da utilidade do mesmo item s para outros usuários c, similares ao usuário c (ADOMAVICIUS; TUZHILIN, 2005).

Os problemas de análise limitada de conteúdo e excesso de especificação, visualizados no método baseado em conteúdo, não estão presentes na filtragem colaborativa, uma vez que são utilizadas as preferências de outros usuários, independentemente do conteúdo dos itens. Entretanto, são problemas desta abordagem: (i) novos itens começarão a ser recomendados

apenas após avaliação de usuários; (ii) conjuntos pequenos de usuários em grandes volumes de dados tornará a recomendação esparsa, diminuindo o conjunto de itens recomendáveis; (iii) usuários com preferências não usuais, comparados ao resto do grupo, não receberão boas recomendações, pois não terão muitos vizinhos próximos (BALABANOVIć; SHOHAM, 1997).

2.5.3 Método Híbrido

O híbrido é definido como um método que combina tanto estratégias de recomendação baseadas em conteúdo quanto estratégias baseadas em colaboração (BALABANOVIć; SHOHAM, 1997). A vantagem de uma abordagem que unifique as outras é ampliar significativamente as chances de obter acertos em suas recomendações e eliminar as limitações de ambas abordagens. A Figura 2.3 mostra um modelo de método híbrido proposto por BALABANOVIć; SHOHAM (1997).

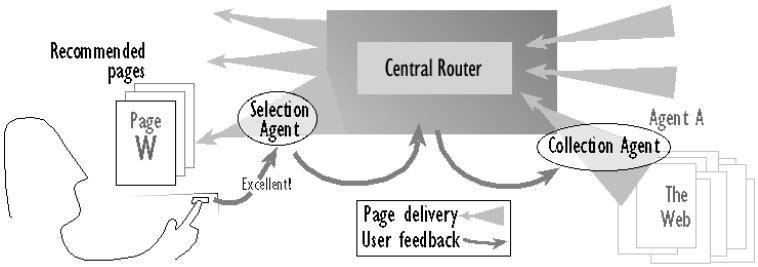

Figura 2.3 – Modelo de Método Híbrido para Sistema de Recomendação (BALABANOVIć; SHOHAM, 1997)

Na Figura 2.3 é apresentado um modelo de método híbrido, onde o sistema de recomendação apresenta sugestões considerando o conteúdo dos objetos e também a recomendação do usuário para aquele conteúdo que lhe foi sugerido.

2.6 Pré-Processamento de Texto

Técnicas de mineração de textos, que também podem ser encontrada na literatura como mineração de dados textuais ou então descoberta de conhecimento de bases de dados textuais, de forma geral, se referem ao processo de descobrir conhecimento em documentos de texto não-estruturados. Podem ser vistas como uma extensão da mineração de dados ou da descoberta de conhecimento em bases de dados estruturadas (ARANHA; PASSOS, 2006).

Em um sistema de *Service Desk*, técnicas de mineração de texto podem ser aplicadas à descrição do chamado aberto pelo usuário, a fim de realizar um pré-processamento de textos

para posteriormente analisar a similaridade entre eles e identificar casos semelhantes de interações passadas armazenadas na base de dados, através da utilização de algumas técnicas de mineração de texto como, por exemplo, a remoção de *stopwords*, o *stemming* e algoritmos para análise de similaridade.

2.6.1 Remoção de *Stopwords*

Um conjunto de *strings* que compõem um documento é formado por algumas palavras (*tokens*) que não possuem valor semântico, sendo útil apenas para que o texto possa ser compreendido de forma geral. Em um sistema de mineração de dados, tais palavras são consideradas as *stopwords* e pertencem a uma *stoplist*. Com uma *stoplist* bem organizada é possível eliminar até 50% do total de palavras de um texto (JUNIOR, 2007).

Um exemplo pode ser visto através da frase "*Estou com problema na minha impressora.*", que ao aplicar a técnica de remoção de *stopwords* resulta em "*problema impressora*", removendo cerca de 66% das palavras que compõem a frase.

2.6.2 *Stemming*

O termo *stemming* em inglês ou simplesmente "radicalização", em português, tem o objetivo de reduzir cada palavra até que seja obtido seu radical, através da eliminação de sufixos que indicam variação na forma da palavra, como plural, tempo verbal, locução adverbial, aumentativo, gênero, acentuação. Por exemplo, as palavras "duvido, dúvida, duvidamos, duvidem" se submetidas a uma técnica de *stemming* tornam-se "duvid" (ORENGO; HUYCK, 2001). Segundo pesquisa apresentada em (KROVETZ, 1993), a utilização de *stemming* melhorou em 35% a recuperação de informações em alguns conjuntos de dados.

Infelizmente a radicalização é um processo que implica algoritmos diferentes de acordo com o idioma em questão, pois existem diferenças na forma como as palavras são formadas, de forma que a aplicação de uma técnica específica pode produzir resultados desiguais de acordo com a linguagem dos textos (VIERA; VIRGIL, 2007).

De acordo com VIERA; VIRGIL (2007), a proposta mais conhecida para a língua portuguesa é a de ORENGO; HUYCK (2001), também existe o algoritmo de PORTER (1997) para o português, seguindo as mesmas regras do desenvolvido pelo mesmo autor para a língua inglesa.

2.6.2.1 Algoritmo de Orengo

O algortimo de Orengo ORENGO; HUYCK (2001) é desenvolvido especificamente para o português. Esse algoritmo consiste de uma série de oito passos, executados de acordo com uma ordem pré-definida pelo algoritmo, de tal maneira que os sufixos mais extensos devem ser removidos primeiro. Esse algoritmo foi desenvolvido com base nos sufixos mais comuns encontrados no português (SOARES; PRATI; MONARD, 2009). O algoritmo de Orengo apresenta oito etapas que são apresentadas na Tabela 2.1.

Tabela 2.1 – Etapas do removedor de sufixos, adaptado de ORENGO; HUYCK (2001)

Etapa	**Descrição**
1. Redução do plural	Remove-se o final -s indicativo de plural de palavras que não se constituem em exceções à regra, realizando modificações, quando necessário.
2. Redução do feminino	Remove-se o final -a de palavras femininas com base nos sufixos mais comuns.
3. Redução adverbial	Remove-se o final -mente de palavras que não se constituem em exceção.
4. Redução de grau	Removem-se os indicadores de aumentativo e diminutivo mais comuns.
5. Redução nominal	Removem-se 61 sufixos possíveis para substantivos e adjetivos.
6. Redução verbal	Reduzem-se as formas verbais aos seus radicais.
7. Remoção de vogais	Removem-se as vogais a, e e o das palavras que não foram tratadas pelos dois passos anteriores.
8. Remoção de acentos	Removem-se os sinais diacríticos das palavras.

Algumas etapas são executadas apenas se determinada característica é encontrada, isso ocorre no caso na redução do plural e na redução do feminino. Em outras duas situações, determinados procedimentos deixam de ser executados se uma etapa anterior foi realizada. É o caso da redução verbal, que só ocorre se a redução nominal não ocorreu e o caso da remoção de vogais, que ocorre somente quando a redução nominal e a redução verbal não foram feitas (SOARES; PRATI; MONARD, 2009).

2.6.2.2 Algoritmo de Porter

O algoritmo de PORTER (1997), originalmente proposto para o inglês, está baseado na ideia de que os sufixos em inglês são principalmente compostos por uma combinação de sufixos menores e simples. O algoritmo é composto por uma série de cinco passos, no qual algumas

regras são aplicadas para remover os sufixos em cada um dos passos. Se um sufixo casa com a palavra, o sufixo é removido se as regras definidas para aquele passo são aplicáveis (SOARES; PRATI; MONARD, 2009).

Em consonância com VIERA; VIRGIL (2007), o algoritmo de Porter adaptado à língua portuguesa também está baseado em regras. Cinco são os passos realizados por ele:

- remoção dos sufixos;
- remoção dos sufixos verbais, se o primeiro passo não realizou nenhuma alteração;
- remoção do sufixo i, se precedido de c;
- remoção dos sufixos residuais os, a, i, o, á, í, ó;
- remoção dos sufixos e, é, ê e tratamento da cedilha.

Ao final, devem-se voltar as vogais nasalizadas à sua forma original (VIERA; VIRGIL, 2007).

2.7 Algoritmos de Similaridade entre *Strings*

Na literatura existem diversas técnicas para cálculo de similaridade entre *strings*, tais como o modelo de índice invertido, algoritmo *Levenshtein Distance*, processamento de linguagem natural, algoritmos de Boyer-Moore, Rabin-Karp, Jaro-Winkler, entre outras técnicas que podem ser vistas em (ARANHA; PASSOS, 2006) (JUNIOR, 2007) (BENDERSKY; CROFT, 2009) (ORENGO; HUYCK, 2001). Nas próximas seções são apresentados alguns destes algoritmos.

2.7.1 Algoritmo Levenshtein Distance

O Algoritmo de LEVENSHTEIN (1966) define a distância de diferença entre duas *strings*, é baseado no número mínimo de operações necessárias para transformar uma *string* em outra. Para isso, existem três operações que transformam uma sequência de caracteres em outra: inserção (i.e, insere um novo caractere na *string* "destino"), eliminação (i.e, elimina um caractere na *string* "origem") e substituição (i.e, substitui um caractere na *string* "origem", com o objetivo de transformar na *string* "destino"). Um exemplo pode ser os passos necessários para transformar "casas" em "massa", que define a distância de edição em 3 (substituição de "c" por "m"; eliminação do segundo "a"; inserção de "a" no final da *string*) (JUNIOR, 2007).

2.7.2 Algoritmo Boyer-Moore

A partir de um padrão *P* relativamente grande e um alfabeto *A* também com a mesma característica, como normalmente acontece na maioria dos textos comuns, o algoritmo proposto por BOYER; MOORE (1977) realiza suas comparações da direita para a esquerda fazendo o uso de duas variáveis que são usadas para calcular o deslocamento. Essas duas novas variáreis são conhecidas como heurística do mau-caractere e a heurística do bom-sufixo (AVILA; SOARES, 2012).

2.7.3 Algoritmo Rabin-Karp

O algoritmo de KARP; RABIN (1987) tem característica probabilística utilizando como parâmetro duas *strings* ao qual se deseja comparar. Caso a *string* seja encontrada, a função retorna o número de caracteres iguais, caso contrário o valor *0* será retornado. O algoritmo surgiu com a ideia de comparar o *hash* da *string* padrão com o *hash* de *sub-strings* do texto. Em geral, a ideia é aparentemente simples, porém é necessário utilizar uma função *hash* que gere um *hash* diferente para cada *sub-string* (AVILA; SOARES, 2012).

2.7.4 Algoritmo Jaro-Winkler

O Algoritmo Jaro-Winkler, WINKLER (1990), calcula a medida de similaridade entre duas *strings*. Foi criado com base no algoritmo de JARO (1989). Este algoritmo estabelece que dadas duas *strings* s_1 e s_2, sua distância d_j é calculada através da seguinte fórmula:

$$d_j = \frac{m}{3a} + \frac{m}{3b} + \frac{m-t}{3m}$$

Onde, m é o número de correlações entre caracteres; a e b correspondem aos tamanhos de s_1 e s_2, respectivamente; t é o número de transposições (WINKLER, 1990).

2.8 Trabalhos Relacionados

Existem vários estudos que abordam aspectos de como melhorar o controle da gestão dos incidentes em ambientes de TI. Nesta seção são apresentados os trabalhos correlatos na área de sistemas de *Service Desk* e também *Help Desk*.

Em WANG et al. (2011) é proposto um sistema de *Help Desk online* denominado iHelp.

Neste trabalho é utilizado um algoritmo de agrupamento, baseado em SNMF (*Symmetric Non-negative Matrix Factorization*) para agrupar os chamados em diferentes categorias. Posteriormente, o iHelp realiza um resumo com uma solução de referência para cada grupo de casos identificados, levando em consideração as interações da equipe de suporte para solucionar o problema. Desta forma, a partir de um novo incidente registrado pelo usuário, o sistema pesquisa e o aloca em um dos grupos que tem conhecimento e sugere uma solução de referência. Para isso, faz uso de um sistema de recomendação e técnicas de pré-processamento de texto. No entanto, esta ferramenta não faz o gerenciamento dos chamados, pois atua apenas com a alocação de cada problema requisitado ao sistema em um dos grupos que tem conhecimento.

O estudo de caso apresentado por JANTTI; KALLIOKOSKI (2010) teve como objetivo responder quais os tipos de desafios de gestão do conhecimento estão relacionados à um *Service Desk* e em seus processos de suporte e serviços. A principal contribuição deste estudo é apresentar os desafios identificados e classificá-los de acordo com cinco atividades principais da área de gestão do conhecimento (e.g, identificação, criação, armazenamento, compartilhamento e uso). De acordo com os autores, os desafios mais importantes foram relacionados à classificação de incidentes (i.e, grande número de categorias de incidentes), automatização de tratamento de incidentes e gerenciamento de chamados.

No trabalho de SAMARAKOON; KUMARAWADU; PULASINGHE (2011) é descrito um método para um sistema de perguntas e respostas que, segundo os autores, pode ser utilizado como primeiro passo para automatização de respostas à incidentes em um sistema de *Service Desk*. Na abordagem são utilizadas técnicas de pré-processamento de texto, como a remoção de *stopwords*, técnica de *stemming* e normalização de tamanho. Na pesquisa foram avaliadas a recuperação de informação aplicando cada técnica separadamente, depois uma técnica mais a outra e, por fim, onde obtiveram um melhor retorno dos dados, aplicando todas as técnicas em conjunto. Assim como no trabalho de WANG et al. (2011), o trabalho de SAMARAKOON; KUMARAWADU; PULASINGHE (2011) não faz o gerenciamento dos chamados, pois se propõe a ser um primeiro passo para um sistema automatizado de respostas à incidentes que também deve ser agregado a outro sistema de *Service Desk*.

Existem alguns *softwares* no mercado que incorporam algum mecanismo para características da *expertise* através do direcionamento dos chamados para setores cadastrados como áreas no sistema. Como exemplo, cita-se o Trellis Desk, o OcoMon e o eTicket ACCORD5 (2012) OCOMON (2012) ETICKET (2012), *softwares* livres que podem ser utilizados sem

custos por qualquer empresa. Os *softwares* utilizam um cadastro de departamentos, áreas ou setores para direcionar a equipe que realizará o atendimento do chamado. Apesar de apresentar esta funcionalidade, na abertura do chamado é o usuário que deve determinar para qual área ele será direcionado e qual a prioridade que este chamado terá.

De acordo com Lobo,

> Neste caso dois problemas podem ocorrer: O primeiro ligado à prioridade, pois o sistema poderá ser inundado com chamados de prioridade críticas daqueles usuários que acreditam ser o seu problema o mais importante a ser resolvido; e o segundo está ligado ao direcionamento do chamado, pois nem sempre o usuário tem o real conhecimento do problema ou quais são os problemas que todos os setores estão aptos a resolver LOBO (2011).

Outros *softwares* disponíveis são o SPICEWORKS (2012), que é uma ferramenta de gerenciamento de redes que incorpora um sistema de *Help Desk*, e o ProManager PROMANAGER (2012). Estas ferramentas, diferente dos outros três *softwares* citados anteriormente, não possibilita a atribuição de um chamado para um determinado setor ou área de conhecimento. A atribuição permitida nestes *softwares* é o direcionamento do chamado a um técnico específico, que é realizada pelo usuário na abertura do *ticket*. A permissão para o usuário definir o técnico que irá prestar o atendimento pode ocasionar a concentração de chamados para um técnico em detrimento de outros que possam ficar totalmente ociosos. Além disso, chamados podem ser direcionadas para um técnico que não tenha a *expertise* necessária para resolvê-la. Outra característica presente no Spiceworks é que este possui um Fórum que serve para que os técnicos busquem soluções para alguns dos problemas a serem solucionados, no entanto um Fórum é uma discussão aberta e não uma recomendação direta de solução.

Além dos sistemas de *Service Desk* citados, também existe o DotProject DOTPROJECT (2012), que é uma ferramenta para gerenciamento de projetos que possui integrado um sistema de gestão de *tickets*. Assim como no Spiceworks e ProManager, nesta ferramenta não é possível atribuir chamado para um setor, o que a ferramenta possui é um método de priorizar os chamados, que é realizado pelo usuário no momento em que o chamado é inserido no sistema. Com a possibilidade do usuário definir a prioridade de seu chamado podem ocorrer casos onde a necessidade de solução é maior, entretanto não foram resolvido antes, pois outros usuários acharam que seu problema era o de maior urgência. A atribuição de prioridade pelo usuário que abre o chamado também está presente nas ferramentas eTicket, ProManager, Ocomon, Spiceworks e Trellis Desk.

De forma geral, nos trabalhos apresentados por WANG et al. (2011) e SAMARAKOON; KUMARAWADU; PULASINGHE (2011) são apresentadas propostas que dão suporte à uma

outra ferramenta de *Service Desk* já existente. Além disso, estes trabalhos não consideram características de contexto. Já nos *softwares* gratuitos, em alguns casos são utilizadas informações de contexto, no entanto nenhum deles possui um sistema de recomendação integrado que indique possíveis soluções para incidentes baseados em casos similares já solucionados. Logo, nenhuma das soluções reúne as características de sensibilidade ao contexto e sistema de recomendação em um único sistema de *Service Desk*, que é a proposta e o diferencial deste trabalho.

3 METODOLOGIA DO TRABALHO

As pesquisas relacionadas à sensibilidade ao contexto, sistemas de recomendação e utilização de dispositivos móveis, a fim de propor um sistema de *Service Desk* foram estimuladas pela observação do funcionamento da Central de Atendimento ao Usuário (CAU) da UFSM. Durante o período de observação (i.e, entre Outubro e Dezembro de 2012), foi realizada uma entrevista informal a cada mês, com o supervisor e alguns técnicos da CAU.

Através das entrevistas foram levantadas questões como: (i) funcionamento do sistema atual da CAU quanto a abertura do chamado; (ii) existência de algum tipo de priorização e classificação dos chamados; (iii) existência de uma divisão dos técnicos quanto ao seu conhecimento; (iv) como se dá a distribuição de chamados quanto a distância que os técnicos precisam percorrer para atender; (v) se as soluções dos incidentes ficam armazenadas; (vi) quantidade de técnicos que atuam.

Além destas questões, que serviram para auxiliar na proposta do sistema elaborado neste trabalho, foi buscado junto a CAU um conjunto de dados contendo chamados reais do sistema, a fim de utilizá-los para comparar os algoritmos de análise de similaridade e avaliar estes quanto ao tempo gasto para processar uma comparação de similaridade com os demais registros e avaliar a qualidade dos resultados obtidos por cada algoritmo. Para a comparação foram utilizados quatro algoritmos que calculam similaridade, são eles: (i) Algoritmo Jaro-Winkler; (WINKLER, 1990) (ii) Algoritmo Levenshtein Distance; LEVENSHTEIN (1966) (iii) Algoritmo String Similarity, que é uma classe usada para calcular a semelhança entre duas cadeias de texto, criada a partir do algoritmo *diff* que compara a diferença entre arquivos no GNU (GNU Operating System) (GNU, 2012); (iv) A função similar_text(), nativa do PHP para cálculo de similaridade (PHP, 2012).

O *Mobile Service Desk* foi instalado em um servidor com o sistema operacional Ubuntu versão 12.04 LTS, utilizando arquitetura de 32 bits. Neste, foi instalado e configurado o banco de dados MySQL 5.5 e o servidor *web* Apache 2.2 com suporte a linguagem de programação PHP 5.3. As capturas de telas apresentadas neste trabalho foram realizadas a partir de *smartphones* e *tablets* com os seguintes sistemas operacionais: Android 2.3, 4.0 e 4.1; iOs 5.1 e 6.0; Windows Phone.

Para a modelagem do sistema foi utilizada a Linguagem de Modelagem Unificada - UML (*Unified Modeling Language*) que permite representar os objetos da aplicação através

de uma notação gráfica e diagramas padronizados (WAZLAWICK, 2010). Para a criação dos diagramas UML utilizou-se o *software* Astah Community (ASTAH, 2012), versão 6.6, que é uma ferramenta de modelagem UML gratuita.

O banco de dados foi modelado com o apoio do *software* DbDesigner versão 4, que é um programa gratuito que integra criação e modelagem gráfica de dados (DBDESIGNER, 2012). O banco de dados utilizado para a persistência dos dados foi o MySQL (MYSQL, 2012) e a ferramenta utilizada para acesso e manutenção ao banco de dados foi a HeidiSQL (HEIDISQL, 2012) que também é gratuita.

O *Mobile Service Desk* foi implementado no NetBeans, *software* gratuito, que é um IDE (*Integrated Development Environment*) e pode ser utilizado com diferentes linguagens de programação, como por exemplo: Java, PHP, HTML, JavaScript, C/C++, entre outras (NETBEANS, 2012).

Para validar a proposta e desenvolver o sistema buscou-se uma linguagem de programação gratuita que possua características como facilidade de tratamento de cadeias de caracteres e programação para *web*. Além dos requisitos iniciais, a linguagem de programação PHP (PHP *Hypertext Preprocessor*) foi escolhida por também possuir as seguintes características: (i) linguagem *server-side*, que executa multi-funções; (ii) compatível com MySQL; (iii) multiplataforma; (iv) funções disponíveis para utilização; (v) documentação.

Assim o PHP atendeu as necessidades de projeto para o desenvolvimento do *Mobile Service Desk*, sendo adotado para a implementação. Para que possam ser armazenados todos os dados dos chamados, técnicos, usuários, prédios, salas e também reter o conhecimento gerado pelos atendimentos aos usuários, utiliza o SGBD (Sistema de Gerenciamento de Banco de Dados) MySQL.

Para funcionar em diversos dispositivos móveis, o sistema possui uma interface de usuário unificada, ou seja, aplicação única para todos os dispositivos móveis e sistemas operacionais suportados. Desta forma, para proporcionar essa interface unificada, foi utilizado o *framework* jQuery Mobile, que baseia-se no HTML 5 (*HyperText Markup Language*), nas bibliotecas jQuery e jQuery UI e tem como característica ser otimizado para interações por toque (SILVA, 2011). Além disso, o HTML 5 também foi aplicado no sistema para dar suporte ao contexto de localização geográfica, pois é através dele que a latitude, longitude e altitude dos usuários são requisitadas pelo sistema *Mobile Service Desk*.

A avaliação da interface do sistema proposto, juntamente com a de outros *softwares* de

Service Desk, foi realizada por meio da elaboração e aplicação de um questionário, norteado pela norma NBR 9241-11 da (ABNT, 2002), que tem como base a norma ISO 9241-11. Esta norma brasileira inclui orientações sobre como a usabilidade de um produto pode ser especificada e avaliada, ela se aplica tanto a produtos de uso geral quanto a produtos sendo adquiridos ou sendo desenvolvidos. A orientação inclui procedimentos para medir usabilidade, mas, não detalha todas as atividades a serem realizadas. A especificação de métodos de medidas detalhados baseados no usuário está além do objetivo da norma (ABNT, 2002). Desta forma, seguindo orientações da NBR 9241-11, foi criado o questionário com doze questões objetivas e dissertativas, divididas em três quesitos a serem avaliados: (i) Facilidade de Aprender; (ii) Usabilidade Global do Sistema; (iii) Legibilidade. O questionário que foi aplicado está no Apêndice A e também disponível no endereço <http://goog.gl/GQg9p>.

O questionário foi respondido por quatro Mestrandos do Programa de Pós-Graduação em Informática da UFSM, com conhecimentos avançados em informática, onde apenas um deles possuía prévio conhecimento de todas as ferramentas testadas, os demais tiveram o primeiro contato com as ferramentas durante a aplicação da pesquisa. Para a realização dos testes foram utilizados *smartphones* e *tablets* rodando tanto o sistema operacional Android quanto iOS e com tamanhos de tela variando de 4"a 10".

4 *MOBILE SERVICE DESK*: PROPOSTA

De acordo com as entrevistas informais realizadas ao supervisor e alguns técnicos da CAU, existem duas maneiras para reportar incidentes à central de atendimentos, a primeira é o contato telefônico e a segunda através do site da CAU. Quando o contato é feito por telefone o próprio atendente realiza o cadastro do *ticket* no sistema. Atualmente a CAU conta com 16 estagiários/bolsistas do Curso Técnico em Informática do Colégio Politécnico da UFSM. Os estagiários são divididos em dois grupos, o grupo externo, que realiza o atendimento ao usuário, e o grupo interno, que recebe as ligações na central de atendimento. A distribuição dos chamados para o grupo externo é realizada pelo grupo interno, no entanto, esta distribuição não obedece a um critério bem definido.

Segundo os técnicos de atendimento interno, os bolsistas externos são encaminhados para realizar suas atividades aleatoriamente. Ao concluir um atendimento, o técnico externo liga para o CAU e o técnico interno irá avisá-lo sobre o próximo atendimento a ser realizado, sem fazer nenhuma correlação entre o problema e o conhecimento do técnico de suporte. A única priorização existente entre chamados é para os casos em que o problema envolve algum setor no prédio da reitoria, como por exemplo alguma secretaria da reitoria.

Ainda que seja feita a distinção da prioridade pelo técnico interno, percebe-se que o critério de prioridade é abstrato, pois para um estagiário o usuário "A"pode ter maior prioridade que o usuário "B", e para outro esta ordem de prioridade pode ser invertida. Ainda com relação à localização, a depender do tamanho da região, que no caso da UFSM é uma área territorial de 1.837,36 hectares, sendo 273.150,92 m^2 de área construída no Campus (UFSM, 2012), e da distribuição dos *tickets*, um técnico poderá perder tempo em deslocamentos desnecessários. Por exemplo, assumindo-se três chamados de mesma prioridade ocorridos em prédios diferentes, como mostra a Figura 4.1, se o técnico atender na ordem 1, 2 e 3 dos *tickets*, terá percorrido 400m. No entanto se o técnico for encaminhado ao *ticket* 1, em seguida ao 3 e posteriormente ao *ticket* 2, terá, ao final, percorrido 600m. Esta distribuição displicente de chamados pode impactar na dinâmica do atendimento, diminuindo sua rapidez e gerando insatisfação ou prejuízos para o usuário.

O sistema de *Service Desk* proposto nessa pesquisa possui três características principais: funcionar em dispositivos móveis e em outros dispositivos como computadores de mesa e *notebooks*; tratar características de sensibilidade ao contexto; armazenar a solução empregada pelos

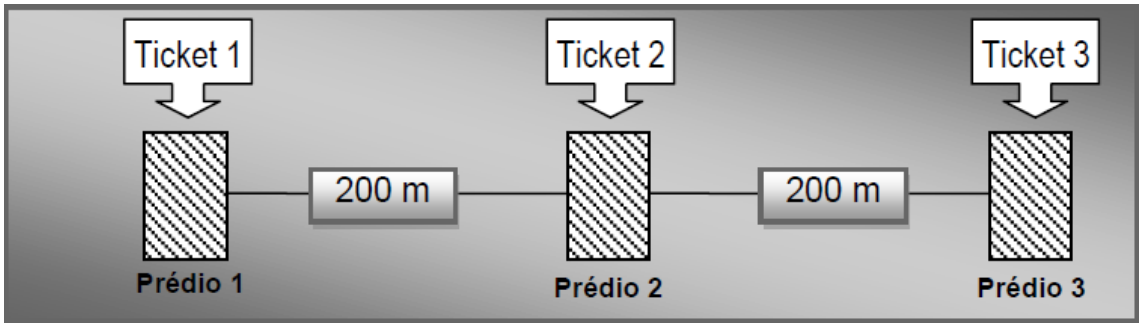

Figura 4.1 – Simulação de incidentes em três prédios diferentes (LOBO, 2011)

técnicos nos incidentes do ambiente de TI e sugeri-las através de um sistema de recomendação a novos casos similares.

4.1 Funcionamento em Diversos Dispositivos

Motivado pelo aumento na utilização de dispositivos móveis e considerando que os técnicos de atendimento estão em constante movimentação realizando atendimentos aos usuários, uma das características desejáveis à ferramenta de *Service Desk* é que funcione na grande maioria de dispositivos móveis disponíveis na atualidade, como *Smartphones*, *Tablets*, *E-Readers*, *Netbooks* de diferentes fabricantes e Sistemas Operacionais (Android, iOS, Windows Phone, Blackberry, Palm OS, Samsung Bada, Kindle, Opera Mobile, Firefox Mobile, Chrome for Android, Meego, Symbian, entre outros), além de outros dispositivos de acesso, como os Computadores Pessoais e *Notebooks* de diversos Sistemas Operacionais e navegadores (Windows XP, Windows 7, Linux, Mac, Safari, Internet Explorer, Firefox, Opera, Chrome, entre outros).

Desta forma, o técnico de suporte carregará consigo um dispositivo móvel para iniciar e encerrar chamados, saber qual o próximo atendimento a ser realizado e também utilizar as demais funcionalidades propostas para este sistema.

Através da Figura 4.2, que apresenta o percentual de sistemas operacionais em dispositivos móveis no Brasil, é possível observar que a grande maioria dos dispositivos utiliza sistema operacional Android.

Outro ponto relevante é que a grande maioria de dispositivos (69%) utiliza um dos sistema operacionais (Android, Blackberry, iOS, Windows Mobile ou Symbian). Através dessas informações é possível concluir que para elaboração de um *Service Desk* móvel um dos atributos que este deve possuir é que funcione, se não na totalidade, na grande maioria dos dispositivos utilizados atualmente.

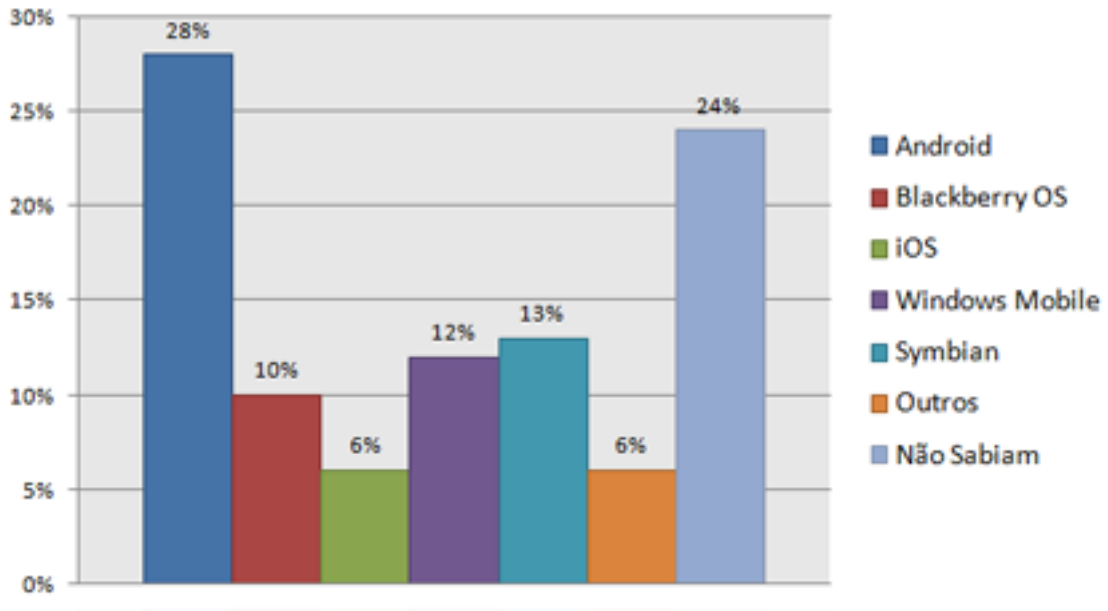

Figura 4.2 – Sistema Operacional de Dispositivos Móveis no Brasil (GOOGLE, 2012)

Por sua vez, a Figura 4.3 apresenta os mesmos dados da Figura 4.2, porém com informações dos sistemas operacionais utilizados em dispositivos móveis nos Estados Unidos. Nesse caso, o predomínio dos sistemas Android, Blackberry, iOS, Windows Mobile e Symbian é ainda maior, abrangendo 89% dos casos.

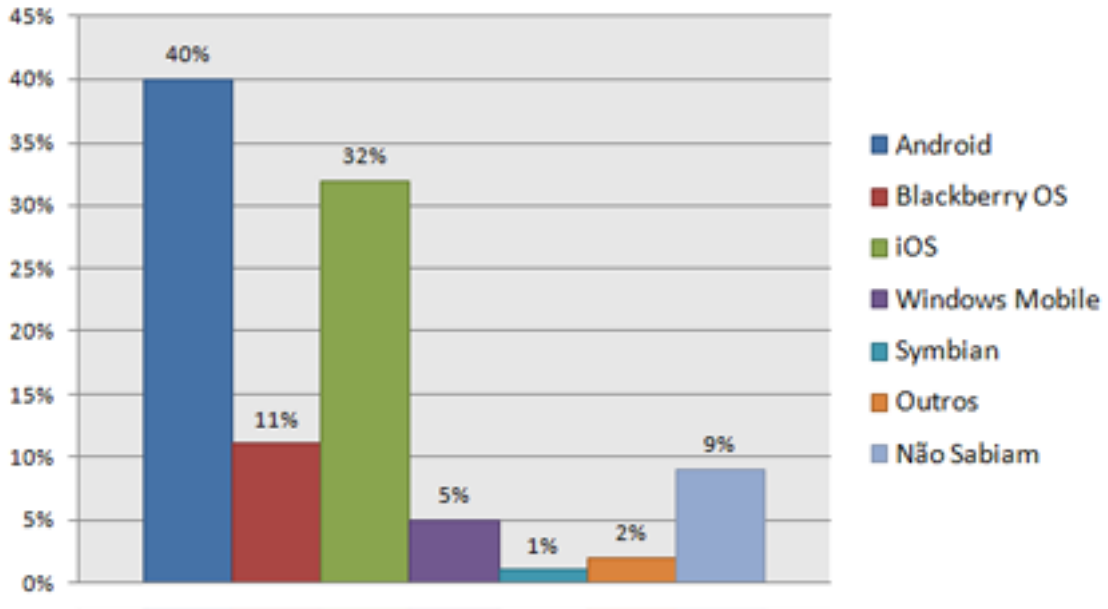

Figura 4.3 – Sistema Operacional de Dispositivos Móveis nos EUA (GOOGLE, 2012)

No desenvolvimento deste trabalho foi utilizado o *framework* jQuery Mobile JQUERY-MOBILE (2012) que monta uma interface de acesso única para todos os dispositivos suportados e também tem a característica de ser otimizado para interações por toque. A lista de dispositivos

suportados por este *framework* estão no apêndice D.

4.2 Sensibilidade ao Contexto

Neste trabalho, o contexto de localização foi utilizado com o objetivo de otimizar o número de atendimentos realizados através de uma melhor alocação dos chamados, com base em informações de latitude, longitude e altitude em que o técnico de suporte se encontra, a fim de priorizar atendimentos mais próximos, evitando perca de tempo com deslocamentos desnecessários.

Outra informação de contexto a ser utilizada é a *expertise* (i.e, experiência, prática e conhecimento técnico) do técnico, para que os chamados abertos no sistema sejam classificados em níveis de *expertise* necessária para atendê-lo, direcionando o técnico com a *expertise* correta para solucionar o incidente. Evitando casos onde um técnico sem conhecimento suficiente vá atender a um chamado e tenha que retornar para que outro possa resolver.

Por fim, o contexto temporal traz a informação do horário de serviço de cada técnico, para que não sejam alocados chamados para algum técnico no momento em que o mesmo não se encontra em horário de expediente.

Assim, o sistema contempla os contextos de localização (i.e, auxiliar a distribuição de chamados para os técnicos mais próximos), o contexto de expertise (i,e, atrelar chamados apenas para um técnico que possui a expertise necessária) e o contexto de tempo (i.e, não relacionar um chamado para um técnico fora de expediente).

4.3 Ordenação de chamados

Para que os *tickets* abertos no sistema sejam atendidos de forma ordenada, é proposto que o sistema considere três condições: a primeira é que o técnico de suporte possua o contexto de *expertise* necessário para realizar o atendimento, a segunda é que cada chamado deverá receber do Classificador um nível de prioridade (i.e, Baixa, Média ou Alta) e a terceira condição de ordenação é o contexto de localização, que leva em consideração a proximidade do chamado em relação ao técnico que irá atender.

4.3.1 Contexto de *Expertise*

Em ambientes como o da CAU, onde técnicos de suporte são alocados para resolver chamados sem que seja levado em consideração se eles possuem a *expertise* (i.e, experiência, prática e conhecimento técnico) necessária para realizar tal tarefa, podem ocorrer casos onde o técnico se desloca até o local de atendimento e acaba retornando sem que solucione o problema do usuário. Desta forma, o sistema proposto nesse trabalho leva em consideração o contexto de *expertise* de cada técnico de suporte no momento de alocá-lo para resolução de um incidente. No *Mobile Service Desk*, quem define a *expertise* necessária para atendimento de um chamado é o usuário com perfil de Classificador.

4.3.2 Prioridade de Chamados

Neste trabalho, a proposta é que a priorização de chamados seja definida pelo usuário Classificador e não pelo usuário que realiza a abertura do *ticket* (e.g., como ocorre em outras ferramentas apresentadas na seção 2.8). Sendo assim, quando o Classificador encaminhar um chamado para solução ele terá que definir, além da *expertise* necessária para resolução do problema, também a prioridade deste chamado de acordo com três níveis: (i) Alta; (ii) Média; (iii) Baixa, a fim de que estas informações sejam levadas em consideração na hora de alocar e priorizar chamados aos técnicos.

4.3.3 Contexto de Localização

O terceiro item considerado na ordenação de chamados para atendimento é o contexto de localização, para que seja possível priorizar chamados de acordo com a proximidade do técnico em relação ao local de ocorrência do incidente, economizando tempo com deslocamentos desnecessários. Para isso, é necessário a aplicação de uma fórmula que calcule a distância entre dois pontos terrestres, baseado na informação da latitude e longitude de cada um desses pontos. Assim, a fórmula de Haversine apresentada a seguir e elaborada por ROBUSTO (1957), pode ser utilizada no *Mobile Service Desk* para realizar esta tarefa.

$$haversin\left(\frac{d}{r}\right) = haversin(\phi_1 - \phi_2) + \cos(\phi_1)\cos(\phi_2)haversin(\lambda_2 - \lambda_1)$$

$$d = 2R\arcsin\left(\sqrt{h}\right)$$

Onde d é a distância entre dois pontos; r é o raio da Terra; ϕ_1 é a latitude do ponto 1 e

ϕ_2 é a latitude do ponto 2; λ_1 é a longitude do ponto 1 e λ_2 é a longitude do ponto 2. A distância finalmente é obtida pelo uso da função arcoseno, onde h denota de $haversin\left(\frac{d}{r}\right)$. Para definir o valor de r (i.e, raio da Terra) é utilizado um número aproximado, pois a mesma não é uma esfera perfeita. O raio da Terra varia entre 6356,78km nos pólos e 6.378,14km na linha do equador.

4.4 Relacionamento de Chamados

Uma das características do *Service Desk* proposto neste trabalho é de que ele apresente soluções de problemas passados para os novos incidentes que o técnico de Suporte terá que atender, a fim de buscar um meio de resolver este novo incidente aplicando um método já utilizado em outro chamado similar.

Desta forma, técnicos da equipe de suporte podem ter ao seu alcance uma possível solução para os problemas que precisam atender. Para que isso seja possível, no momento em que um Usuário abre um chamado no sistema de *Service Desk* o sistema deve automaticamente relacioná-lo com outros registros anteriores no sistema que já tenham sido solucionados.

Neste trabalho são aplicadas cinco etapas que compõem o pré-processamento de texto, propostas por (AVILA; SOARES, 2012), e servem como etapa inicial para posterior análise de similaridade entre *strings*:

- remoção de caracteres inválidos: caracteres como aspas, colchetes, parênteses, dentre outros, devem ser retirados;
- substituição de caracteres acentuados: caracteres acentuados foram substituídos por seu respectivo caractere não acentuado;
- exclusão de palavras repetidas: para evitar a comparação desnecessária de uma palavra duplicada várias vezes;
- aplicação de *lowercase*: para impedir que palavras com o mesmo significado iniciadas com caracteres em maiúsculo venham a ser diferenciadas de uma palavra semelhante iniciada com caracteres em minúsculo;
- remoção de *stopwords*: para prevenir que palavras como artigos, advérbios, pronomes, preposições, dentre outras que não possuem valor semântico, sendo útil apenas para que o texto possa ser compreendido de forma geral;

- algoritmo de *stemming*: aplicar o algoritmo de Orengo ORENGO; HUYCK (2001), que foi escolhido devido a ter sido desenvolvido especialmente para o idioma Português, para reduzir as variantes morfológicas das palavras, como formas singulares, plural e conjugações verbais, para a sua raiz ou radical, retirando sufixos e prefixos.

Após execução das etapas de pré-processamento no texto de abertura do chamado, restarão um conjunto de *strings* que serão analisadas com casos anteriores a fim de determinar a similaridade entre chamados, para isso foi utilizado o algoritmo *similar_text()*, a justificativa do uso deste algoritmo está na Seção 6.1.4 que apresenta o comparativo entre alguns algoritmos para análise de similaridade entre *strings*.

4.5 Recomendação de Soluções

Após determinada a similaridade entre chamados, é proposto que o sistema recomende possíveis soluções para o novo incidente. Desta forma, no momento em que o técnico for realizar seu atendimento, o sistema deve ter conhecimento de casos solucionados onde o texto de abertura do chamado tenha similaridade com o incidente atual.

Neste trabalho é proposto um método para sistema de recomendação baseado em conteúdo. Onde o conteúdo e as características dos chamados são analisados a fim de definir qual o nível de similaridade que eles possuem e, posteriormente, recomendar casos já solucionados para um semelhante que ainda não foi atendido.

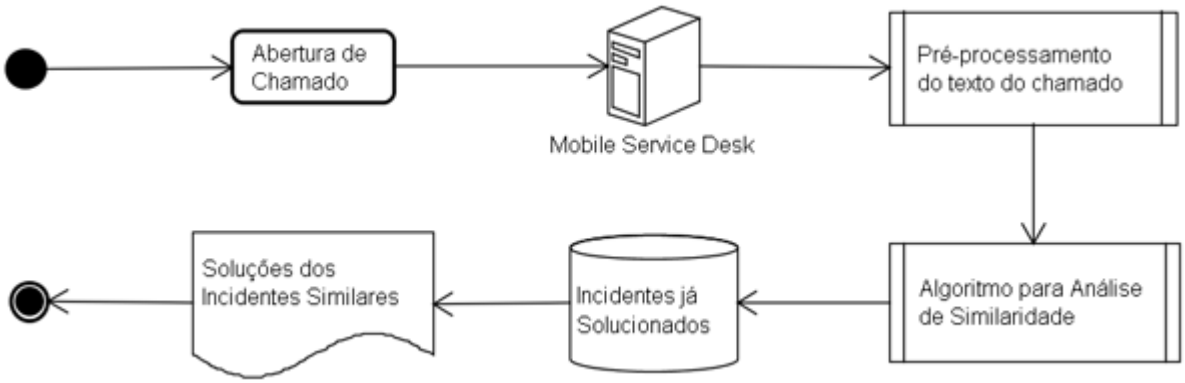

Figura 4.4 – Fluxograma para Recomendação de Soluções

A Figura 4.4 apresenta em forma de fluxograma o funcionamento do sistema de recomendação de soluções proposto neste trabalho, que a partir do texto de abertura do chamado aplica as técnicas de pré-processamento de texto abordadas na Seção 4.4, posteriormente ana-

lisa a similaridade do texto resultante com casos já solucionados, a fim de obter uma possível solução para o incidente e apresentá-la ao técnico no momento em que o mesmo for atendê-lo.

4.6 Modelagem do MSD

Como apresentado no Caso de Uso da Figura 4.5, o sistema proposto é dividido em quatro níveis de usuários (Usuário, Classificador, Suporte e Gestor), suas características e funções estão descritas nas subseções a seguir.

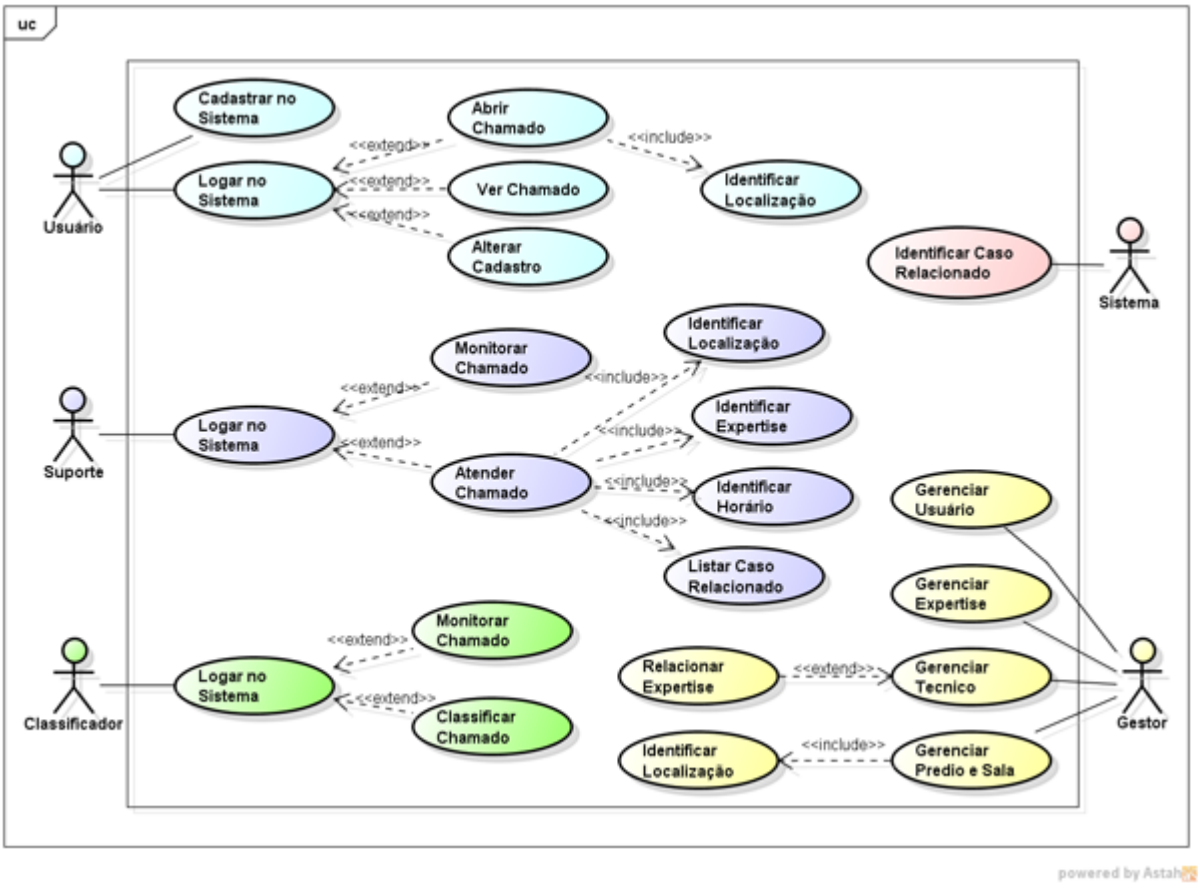

Figura 4.5 – Caso de uso do sistema *Mobile Service Desk*

4.6.1 Nível Usuário

Os utilizadores de nível "Usuário" são os que fazem uso de tecnologias no ambiente gerenciado, serão cadastrados no sistema com informações como nome, ramal, sala, *login* e senha. Sua tarefa é inserir no sistema (i.e, abrir chamado ou *ticket*) cada incidente que ocorre no ambiente de trabalho, para que a equipe de TI possa solucioná-lo. A informação da sala é solicitada para que, no momento em que o usuário for abrir um chamado, já venha selecionada

a sua sala como padrão.

4.6.2 Nível Classificador

Para que os *tickets* abertos pelos Usuários possam ser atendidos, é necessário que o usuário "Classificador" defina qual o nível de *expertise* necessário para resolver o incidente. Com isso, o sistema poderá direcionar o chamado para os técnicos de suporte que tenham habilidade para resolver tal situação. Além disso, o Classificador também é responsável por atribuir o nível de prioridade ao chamado, a fim de priorizar chamados de incidentes que causam maior impacto na organização ou que possuam uma urgência maior em relação aos demais.

4.6.3 Nível Suporte

A partir do momento em que um usuário de nível "Suporte" acessa o sistema, são listados os chamados em aberto, por ordem de prioridade, considerando informações do contexto, como por exemplo, localização atual do técnico, seu horário de expediente e também se possui a *expertise* necessária para atendimento do problema.

Ao iniciar um atendimento, são apresentadas sugestões de problemas relacionados, pois a solução aplicada em algum destes problemas talvez possa ser reutilizada para resolver o novo. Após concluir o atendimento, o técnico vai registrar a solução utilizada no banco de dados do sistema, para que a mesma possa servir como solução para outros problemas similares.

4.6.4 Nível Gestor

O nível "Gestor" é responsável por gerenciar o sistema de *Service Desk*, cadastrando a equipe de suporte, bem como a *expertise* que cada membro possui. Além disso, controla os Usuários que tem acesso ao sistema para abertura de chamados e também os prédios/salas que, ao serem cadastrados, tem o contexto de localização detectado pelo sistema. Com isso, na abertura do chamado para resolver um problema em algum prédio ou sala é possível identificar o técnico mais próximo do local e que atenda a *expertise* necessária para solucioná-lo.

A Figura 4.6 mostra a proposta de como acontece o funcionamento do sistema. A partir da abertura de um chamado pelo usuário, o sistema automaticamente relaciona este com chamados anteriores, com isso, é possível sugerir à equipe de Suporte possíveis soluções relacionadas ao atendimento em aberto. Para relacionar os chamados foi proposta a utilização de algoritmos

de pré-processamento de texto e análise de similaridade, como apresentados na seção 4.4.

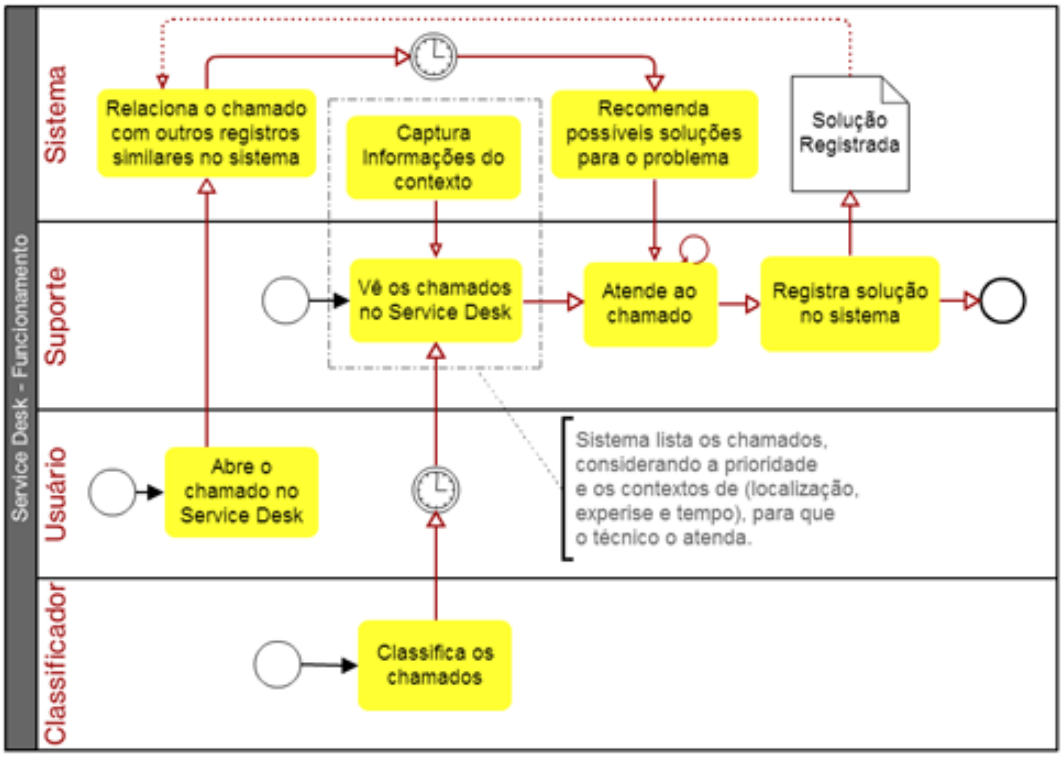

Figura 4.6 – Funcionamento do *Mobile Service Desk*

A Figura 4.7 mostra a proposta de modelo entidade relacionamento do *Mobile Service Desk*, que é um modelo abstrato cuja finalidade é descrever, de maneira conceitual, os dados a serem utilizados no sistema e a forma como eles se relacionam. Nela estão apresentadas as tabelas responsáveis por armazenar os usuários do sistema (i.e, usuario), níveis de expertises (i.e, expertise), relacionamento entre usuários e expertises (i.e, usuarioxp), tipos de usuários (i.e, usuariotipo), chamados (i.e, chamado), locais (i.e, local), prioridades (i.e, prioridade), situações possíveis para os chamados (i.e, situacao) e os chamados que são similares àquele caso (i.e, chamadorel).

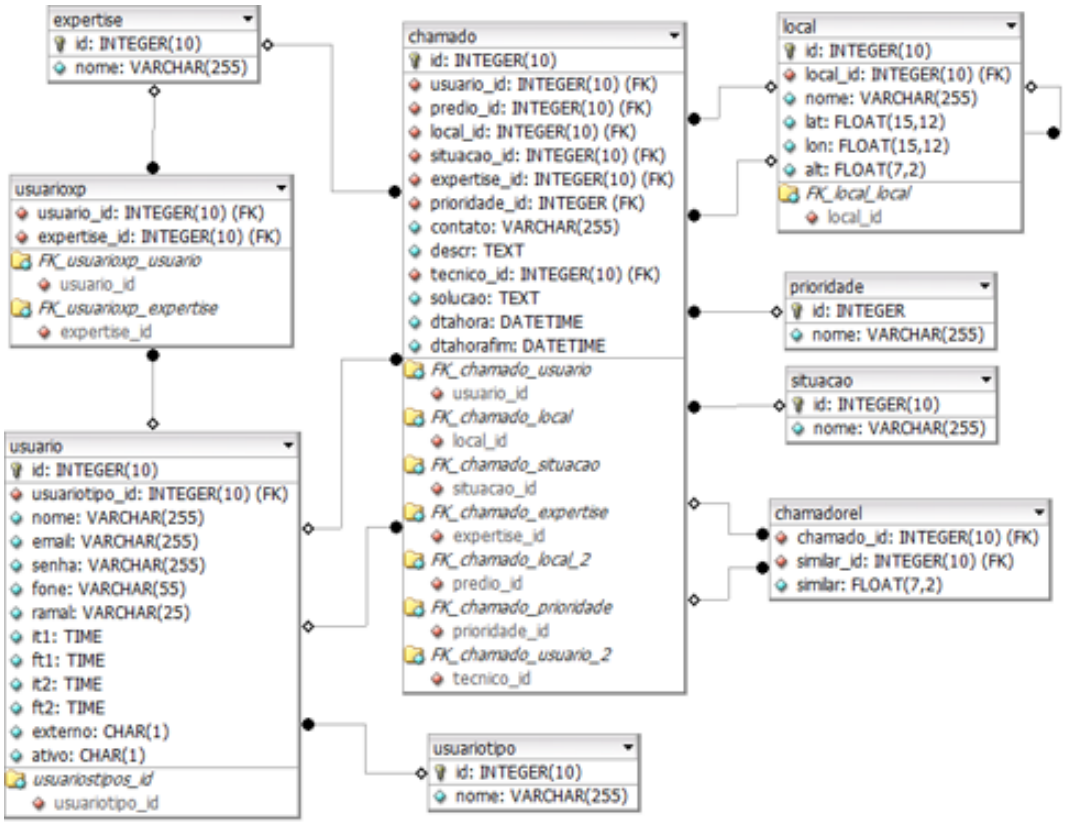

Figura 4.7 – Modelo Entidade Relacionamento do *Mobile Service Desk*

5 *MOBILE SERVICE DESK: IMPLEMENTAÇÃO*

Diante da alta utilização de dispositivos móveis como *smartphones* e *tablets*, decidiu-se estudar formas de utilizar essas tecnologias para aplicar na gerência de TI. A ideia de utilizar a informação de localização do técnico para agilizar o atendimento dos chamados já está expressa no trabalho de LOBO (2011), que também considerava que não só a localização era importante, mas também a definição de um mecanismo que utilizasse a experiência e prática que os técnicos possuíam para diminuir a ocorrência de um segundo atendimento.

Nessa dissertação, estas práticas foram mantidas e aprimoradas no sentido de considerar, além da latitude e longitude do técnico, sua altitude na hora de definir uma prioridade de atendimento, pois pode se tratar de um ambiente onde o prédio é composto por vários andares. Então a questão da altitude é um fator fundamental para definir a distância entre um ponto e outro. Além disso, outros problemas que são encontrados nestes ambientes de TI, como a questão da rotatividade pessoal (*turnover*), levaram ao estudo de técnicas para armazenar as soluções e posteriormente o tratamento desses dados, a fim de que novos membros ingressantes na equipe possam obter sugestões de possíveis soluções para problemas similares que já tenham sido resolvidos. Para que as sugestões de soluções similares pudesse ser agregada à ferramenta, foi necessário o estudo de técnicas na área de sistemas de recomendação.

A Figura 5.1 apresenta a síntese do *Mobile Service Desk* implementada para validar a proposta deste trabalho, mostrando as tecnologias empregadas em seu desenvolvimento e a forma como os usuários interagem com o ambiente.

Figura 5.1 – Síntese do *Mobile Service Desk*

Desta forma, o *Mobile Service Desk* é um sistema que possui funcionalidades específicas como, executar em diversos dispositivos móveis, tratar sensibilidade ao contexto, manter um histórico das soluções dos problemas para que posteriormente esta solução possa ser sugerida para algum outro *ticket* similar por meio de um sistema de recomendação.

5.1 Dispositivos Móveis

Com o objetivo de diminuir a perca de tempo com deslocamentos desnecessários o sistema pode ser acessado pelos técnicos de suporte por meio de um dispositivo móvel. Para isso, em seu desenvolvimento foi utilizado o *framework* jQuery Mobile que monta uma interface de acesso única para todos os dispositivos suportados e também tem a característica de ser otimizado para interações por toque.

Para avaliar este sistema e comparar os resultados obtidos com a avaliação de outras ferramentas de *Service Desk*, visando medir sua usabilidade em dispositivos móveis e computadores, foram seguidas orientações da norma NBR 9241-11. Esta avaliação está descrita na seção 6.2 do capítulo de Resultados.

5.2 Sensibilidade ao Contexto

As características de sensibilidade ao contexto implementadas no *Mobile Service Desk* foram: (i) contexto de expertise (i.e, experiência, prática e conhecimento técnico), que é aplicada na alocação dos técnicos que possuem expertise compatível para realizar o atendimento à um incidente; (ii) contexto de localização, que trata da distância entre o local em que o usuário que necessita atendimento está em relação ao local em que o técnico de suporte se encontra; (iii) contexto temporal, utilizado para gerenciar a jornada de trabalho do técnico, a fim de que não sejam alocados chamados para um técnico que não está no horário de trabalho.

5.2.1 Cadastro de Técnicos

Para ser cadastrado como um técnico da central de serviços, é preciso que o usuário nível Gestor realize o cadastro. Para inclusão das características de contexto, na interface de cadastro existe um item denominado "Contexto". É nesta área que devem ser inseridas as informações relativas ao contexto do técnico (i.e, expertise e jornada de trabalho). A Figura 5.2 mostra a interface administrativa para cadastro de técnicos.

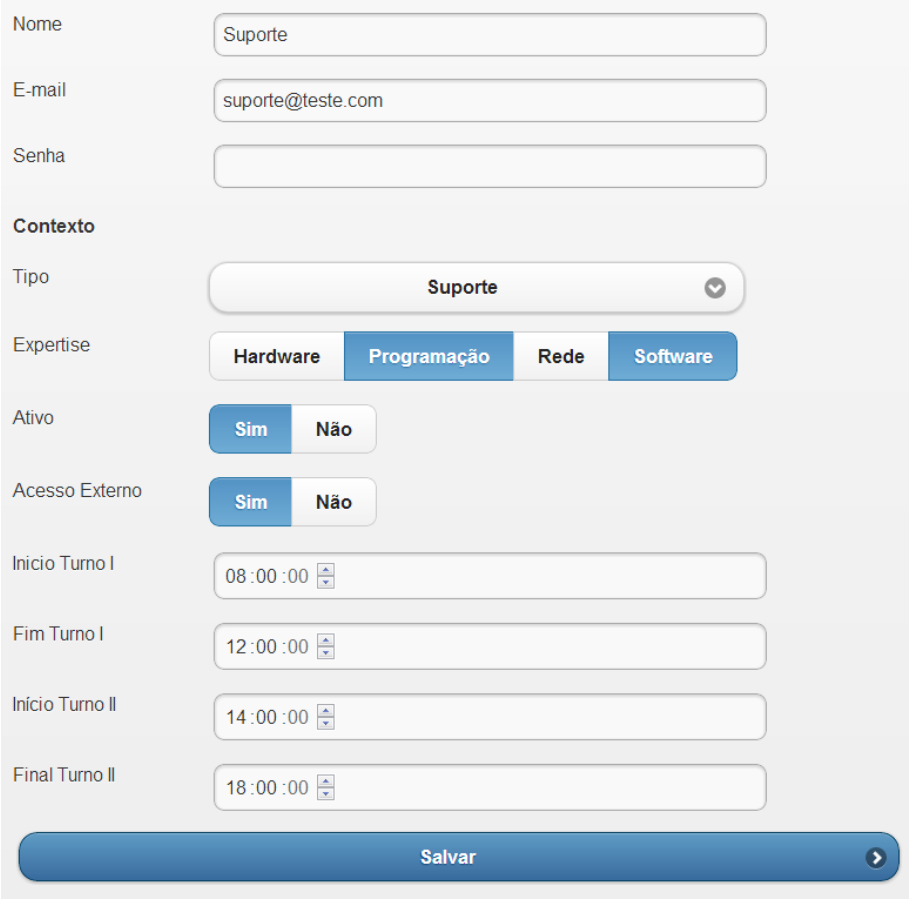

Figura 5.2 – Cadastro de Técnicos no *Mobile Service Desk*

Conforme pode ser visto na Figura 5.2, as informações apresentadas referem-se ao técnico cadastrado com o nome "Suporte". No que tange à contexto, o primeiro campo que aparece é referente ao tipo de perfil do usuário, onde está definido como "Suporte" para que tenha as permissões de um técnico. As demais informações de contexto estão detalhadas nas próximas seções 5.2.1.1 e 5.2.1.2.

5.2.1.1 Definição da *Expertise*

A definição de expertise do técnico também é apresentada na Figura 5.2, neste caso o técnico de nome "Suporte" possui expertise em "Programação" e "*Software*". O cadastro de tipos de expertises (e.g., *Hardware*, Programação, Rede, *Software*), assim como o cadastro de técnicos são gerenciados por usuários tipo Gestor. Este grupo é quem define quais os tipos de expertises que os técnicos podem possuir.

5.2.1.2 Definição de Jornada de Trabalho

A próxima informação relativa à contexto é se o técnico terá ou não acesso ao sistema fora do horário de expediente. Caso não tenha acesso fora do expediente, é considerada a informação sobre a jornada de trabalho do técnico. Para esta informação, o Gestor deverá informar os horários de entrada e saída do técnico que poderá ter até dois turnos.

O cálculo que o sistema utiliza para restringir o acesso quando o técnico não tiver permissão para utilizar o sistema de uma área externa à central de serviços é dado a partir da detecção do contexto temporal (i.e, buscando o horário atual no servidor em que a aplicação está instalada) e comparando se o técnico está dentro do horário de expediente ou não. Caso não esteja em seu horário de expediente e não possua acesso externo, o sistema bloqueia seu *login* e emite uma mensagem de aviso ao técnico. Isso evita com que algum técnico acesse o sistema e acidentalmente inicie atendimento à um chamado em horário que não faz parte de seu expediente, fazendo com que este chamado leve mais tempo para ser resolvido.

5.2.2 Cadastro de Prédios e Salas

Conforme proposto, os chamados devem ser ordenados considerando a distância entre o local onde o atendimento deve ser realizado e o técnico. Para que isso fosse possível, era preciso conhecer as coordenadas do técnico e de cada *ticket*. Considerando-se que os equipamentos de TI estão agrupados em salas e estas por sua vez fazem partem de um prédio, optou-se por criar um cadastro de prédios e salas com as suas respectivas coordenadas e associar cada chamado a uma das salas cadastradas. O cadastro de prédios e salas é realizado pelo Gestor e, assim como todo o sistema, pode ser realizado em uma estação *desktop* ou através do dispositivo móvel. A vantagem do dispositivo móvel é que ao acessar o sistema no prédio ou sala que se deseja cadastrar, a localização (latitude, longitude e altitude) será fornecida automaticamente. Caso as coordenadas dos prédios já sejam conhecidas, o administrador poderá utilizar uma estação *desktop* e cadastrar manualmente essas informações.

A Figura 5.3 mostra um exemplo de cadastro para a Sala 384 do Centro de Tecnologia da UFSM. As coordenadas (latitude e longitude) do local foram detectadas pelo dispositivo móvel após a autorização para que a localização fosse compartilhada.

Figura 5.3 – Cadastro de Prédios e Salas no *Mobile Service Desk*

5.2.3 Contexto de Localização

Durante os estudos identificou-se em (W3C, 2012), que existe um padrão para se obter a identificação do local geográfico de um usuário ou dispositivo de computação chamado de Geolocalização que está incorporado à versão 5 do HTML. A ideia proposta pela W3C é que a obtenção das coordenadas (i.e, latitude, longitude e altitude) ocorra no momento em que o usuário acessar um determinado site. Esta ideia atende justamente a expectativa do projeto, pois a localização será fornecida ao sistema sem a necessidade de se instalar qualquer aplicação nos dispositivos.

Para obter as coordenadas geográficas do usuário, executa-se o código JavaScript:

navigator.geolocation.getCurrentPosition(sucesso);

Onde, $sucesso$ é uma função $callback$, que receberá um objeto de posicionamento.

A Figura 5.4 mostra o *script* utilizado pelo *Mobile Service Desk* para obter a localização dos usuários e realizar o tratamento de erros retornados pelo método. Na linha 2 é verificado

```
1  <script type="text/javascript">
2      if (navigator.geolocation) {
3          navigator.geolocation.getCurrentPosition(sucesso,erro)
4      }
5      else {
6          alert("Este navegador não suporta a funcionalidade Geolocation.");
7      }
8
9      function sucesso(position) {
10         var latitude = position.coords.latitude;
11         var longitude = position.coords.longitude;
12         var altitude = position.coords.altitude;
13
14         $.post("ajax.php", { f: "location", latitude: latitude, longitude:
               longitude, altitude: altitude },
15         function(returned_data) {
16             $("#retorno").empty().html(returned_data);
17         });
18     }
19
20     function erro(num) {
21         switch(num.code) {
22             case num.PERMISSION_DENIED:
23                 alert("O aplicativo não tem permissão para utilizar a API
                       de Geolocalização.");
24                 break;
25             case num.POSITION_UNAVALIABLE:
26                 alert("A posição do dispositivo não pode ser determinada.")
                       ;
27                 break;
28             case num.TIMEOUT:
29                 alert("Esgotado o tempo limite para requisição da
                       localização.");
30                 break;
31     }
32 </script>
```

Figura 5.4 – Código para detectar a localização do usuário

se o navegador suporta o método W3C Geolocation; na linha 3 é solicitada a localização do dispositivo, se obtiver sucesso executa a função $sucesso$, caso contrário a $erro$. Quando o técnico se autentica no sistema uma mensagem é enviada para o dispositivo informando que o site deseja usar a localização atual e solicita permissão. O técnico neste momento pode então escolher se deseja ou não compartilhar sua posição com o site. Outros erros ainda podem acontecer neste processo de autorização. Dentre os erros estão: (i) permissão negada, quando o usuário não compartilhou a sua localização; (ii) posição indisponível, quando os satélites de GPS não puderam ser alcançados ou algum erro semelhante ocorreu; (iii) erro de *timeout* ou de tempo esgotado na tentativa de se obter uma posição.

Nos testes realizados com recursos de API Geolocalização do HTML5 em *smartphones* e *tablets* foram obtidos bons resultados, pois o método de Geolocalização capturou a localização

do dispositivo em todos os testes onde o usuário autorizou o compartilhamento da posição. Já as exceções fixadas entre as linhas 20 e 31 da Figura 5.4 trataram as exceções geradas quando foi negada a permissão para acesso a localização e também quando o dispositivo não obteve sinal de GPS para definir a posição geográfica. A Figura 5.5, apresenta o sistema *Mobile Service Desk* e a tela de solicitação de permissão para acessar a localização do dispositivo.

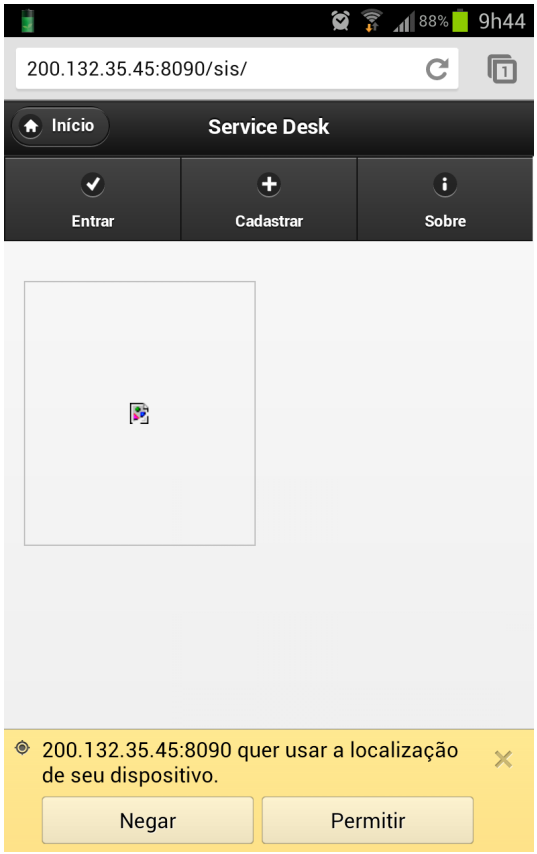

Figura 5.5 – Solicitação de Localização no *Mobile Service Desk*

A permissão de acesso a localização pode ser configurada para sempre liberar acesso ao *Mobile Service Desk*, tornando a utilização do sistema mais dinâmica, pois não é necessário autorizar o recurso de Geolocalização cada vez que for utilizar o sistema.

5.2.3.1 Calculando a Distância entre Técnico e Atendimentos

A implementação da função para realizar o cálculo da distância entre duas coordenadas geográficas, apresentada na Figura 5.6, foi baseada na fórmula de Haversine (ROBUSTO, 1957):

$$haversin\left(\frac{d}{r}\right) = haversin(\phi_1 - \phi_2) + \cos(\phi_1)\cos(\phi_2)haversin(\lambda_2 - \lambda_1)$$

$$d = 2R \arcsin\left(\sqrt{h}\right)$$

```
function distance($latitude1, $longitude1, $latitude2, $longitude2)
{
        $earth_radius = 6371;

        $dLat = deg2rad($latitude2 - $latitude1);
        $dLon = deg2rad($longitude2 - $longitude1);

        $a = sin($dLat/2) * sin($dLat/2) + cos(deg2rad($latitude1)) * cos(
            deg2rad($latitude2)) * sin ($dLon/2) * sin($dLon/2);
        $c = 2 * asin(sqrt($a));
        $d = $earth_radius * $c;

        return $d;
}
```

Figura 5.6 – Cálculo de distância baseado na Fórmula de Haversine

Na função $distance()$ são passados como parâmetros as latitudes e longitudes de dois pontos que deseja-se saber a distância. No caso do *Mobile Service Desk*, o primeiro ponto é a localização atual do técnico e o segundo é a localização do chamado a ser atendido. O valor do raio da Terra está definido em 6371km. Após o cálculo, a função retorna a distância entre os dois pontos utilizando uma escala em metros e a diferença de altura entre o técnico e o local de atendimento também é considerada neste cálculo.

5.2.4 Classificação de Chamados

No sistema da CAU, atualmente, existem dois técnicos que trabalham internamente encerrando os chamados e delegando quem deve atender qual chamado. No *Mobile Service Desk* o técnico com perfil de Classificador é o responsável por informar ao sistema qual é o nível de *expertise* necessário para solucionar cada chamado. Além disso, classifica os chamados quanto a sua prioridade (i.e, Alta, Média ou Baixa). Após essa classificação o chamado é liberado para que possa ser atendido por algum técnico que esteja inserido no contexto necessário para solucionar o incidente. O atendimento de chamados é abordado na seção 5.3.

5.3 Atendimento de Chamados

Para realizar o atendimento de chamados, a partir do momento em que um usuário nível Suporte acessa o sistema, são listados para ele os chamados abertos ordenados de acordo com as regras propostas na seção 4.3. No momento em que o técnico informar que irá iniciar o

atendimento ao incidente, o sistema irá listar as soluções de casos similares ao problema atual, a fim de que possa ser reaproveitada alguma solução. Ainda no momento em que o técnico vai atender algum chamado, são apresentados casos relacionados ao atendimento atual, para isso foi implementado o relacionamento de chamados, que é explicado nas seções 5.4.

5.4 Relacionamento de Chamados

A implementação do relacionamento de chamados, para posteriormente ser sugerido como possível solução de um problema a ser atendido, está dividida em duas etapas: (i) pré-processamento de texto; (ii) análise de similaridade.

5.4.1 Pré-Processamento de Texto

Nesta etapa, é aplicado a cada texto de abertura de chamado as seguintes ações: (i) remoção de caracteres inválidos; (ii) aplicação de *lowercase*; (iii) remoção de *stopwords*; (iii) exclusão de palavras repetidas; (v) aplicação do algoritmo de *stemming*.

A Figura 5.7 apresenta o código de pré-processamento no qual chamados abertos são submetidos antes de realizar a análise de similaridade.

```
1  <?
2      $string = $_POST['descr'];
3      $string = limpaTexto($string);
4      $stpw = explode(' ',$string);
5      $retorno = removeStopwords($stpw);
6      $retorno = array_unique($retorno);
7      arquivoStemming($retorno);
8      $temp = exec('python stemmer\stm.py');
9      $preprocessado = arquivoStemming('','ler');
10 ?>
```

Figura 5.7 – Código de pré-processamento de texto

Na segunda linha da Figura 5.7 a variável "string" recebe via método POST o valor do texto a ser processado, na função $limpaTexto()$, apresentada na Figura 5.8, o texto é passado para minúsculo e os acentos e outros caracteres de pontuação, traços, aspas, parenteses, entre outros são removidos. A quarta linha é responsável por montar um $array$ com todas as palavras que compõem o texto.

A função $removeStopwords()$, Figura 5.9, é responsável por retirar as *stopwords* do texto, por sua vez, a função $array_unique()$ tem a tarefa de retirar os termos duplicados. Na linha 7, a função $arquivoStemming()$, Figura 5.11, gera o arquivo texto com as palavras chaves

```
1  <?
2  function limpaTexto($s)
3  {
4   $s = strtolower($s);
5   $s = preg_replace(array('/á/','/à/','/â/','/ã/','/ª/'),"a",$s);
6   $s = preg_replace(array('/Á/','/À/','/Â/','/Ã/'),"A",$s);
7   $s = preg_replace(array('/é/','/è/','/ê/'),"e",$s);
8   $s = preg_replace(array('/É/','/È/','/Ê/'),"E",$s);
9   $s = preg_replace(array('/í/','/ï/','/ì/','/î/'),"i",$s);
10  $s = preg_replace(array('/Í/','/Ï/','/Ì/','/Î/'),"I",$s);
11  $s = preg_replace(array('/ó/','/ò/','/ô/','/õ/','/°/'),"o",$s);
12  $s = preg_replace(array('/Ó/','/Ò/','/Ô/','/Õ/'),"O",$s);
13  $s = preg_replace(array('/ú/','/ù/','/û/','/ü/'),"u",$s);
14  $s = preg_replace(array('/Ú/','/Ù/','/Û/'),"U",$s);
15  $s = preg_replace(array('0','1','2','3','4','5','6','7','8','9'),"",$s);
16  $s = str_replace("ç","c",$s);
17  $s = str_replace("Ç","C",$s);
18  $s = str_replace(array('.',':',';',',','|','\\','/','<','>','(',')','[',']
    ','{','}','#','$','%','&','*','"','?','!','ª','°','+','=','-','_','~','
    ^','´','`'),"",$s);
19  return $s;
20 }
21 ?>
```

Figura 5.8 – Função que passa o texto para minúsculo e remove alguns caracteres

```
1 <?
2     function removeStopwords($kw)
3     {
4         $kw = array_map('strtolower',array_diff($kw,array('')));
5         $sw = explode("\r\n",file_get_contents('stopwords.txt'));
6         return array_values(array_diff($kw,$sw));
7     }
8 ?>
```

Figura 5.9 – Função para remoção de *stopwords*

```
1  #!/usr/bin/env python
2
3  from ptstemmer.implementations.OrengoStemmer import OrengoStemmer
4
5  stemmer = OrengoStemmer()
6  texto = []
7
8  f = open('stemmer\input.txt','r')
9  for linha in f:
10     texto.append(stemmer.getWordStem(linha.rstrip()))
11     texto.append('\n')
12 f.close()
13
14 s = open('stemmer\output.txt','w')
15 s.writelines(texto)
16 s.close()
```

Figura 5.10 – *Script* Python para técnica de *stemming*

do texto, para que na linha 8 seja executado o *script* em Python, Figura 5.10, que aplica a técnica de *stemming* em português em cima do arquivo gerado anteriormente. O resultado disto, na linha 9, é armazenado na variável "preprocessado".

Quanto a elaboração da *stoplist* para ser aplicada a técnica de remoção de *stopwords*,

```
<?
    function arquivoStemming($array,$acao)
    {
        if($acao=='gravar')
        {
            foreach($array as $key => $texto)
            {
                $text .= $texto."\r\n";
            }

            $arq = fopen($base_dir.'stemmer/input.txt','w');
            fwrite($arq,$text);
            fclose($arq);
        }
        elseif($acao=='ler')
        {
            $arq = fopen($base_dir.'stemmer/output.txt','r');
                        while (($data = fgetcsv($arq, 1000, "\n")) !==
                            FALSE)
                        {
                            $num = count ($data);
                            $row++;
                            for ($c=0; $c < $num; $c++)
                            {
                                $texto .= $data[$c].' ';
                            }
                        }
            fclose($arq);

            $texto = str_replace("\r", " ", $texto);

            return $texto;
        }
    }
?>
```

Figura 5.11 – Função para leitura e gravação do algoritmo de *stemming*

foram encontrados diversos modelos em wikis, google code e também em uma ferramenta de *stemming* já implementada. Desta forma, para este trabalho, foi feito uma mescla das *stoplists* encontradas nas fontes: (i) (RANKSNL, 2012); (ii) (WIKIMEDIA, 2012); (iii) (JOJABACODE, 2012); (iv) (PORTER, 2012); a partir da intersecção destas listas foi criada uma *stoplist* com cerca de 410 palavras que é apresentada no Apêndice B.

Outra informação a ser destacada da etapa de pré-processamento de texto é quanto ao *stemming*, foi utilizado o *script* PTSTEMMER (2012) que possui tanto o algoritmo de PORTER (1997) quanto o de ORENGO; HUYCK (2001) implementados para o idioma Português. Neste trabalho, foi optado por utilizar o algoritmo de Orengo, pois suas regras foram especificamente criadas para o idioma Português (SOARES; PRATI; MONARD, 2009), já o algoritmo de Porter trata de uma adaptação de outro idioma para o Português. Caso a abertura de chamados seja realizada em um idioma diferente do Português, uma técnica de *stemming* específica para o idioma deve ser aplicada ao sistema.

5.4.2 Análise de Similaridade

Para verificar a similaridade entre o texto do chamado a ser atendido e os textos armazenados no sistema, referentes aos atendimentos já realizados, foi aplicado o algoritmo *similar_text()* que compara as *strings* e retorna o percentual de similaridade entre elas. Este algoritmo foi escolhido devido ao seu melhor desempenho em relação aos demais algoritmos testados. A comparação completa entre os algoritmos de similaridade é apresentada na Seção 6.1.4.

6 *MOBILE SERVICE DESK*: RESULTADOS

Neste capítulo está descrito a validação do *Mobile Service Desk*, com seu plano de testes e os resultados obtidos. Além disso, traz a avaliação dos algoritmos para análise de similaridade. Por fim, são apresentados os resultados do questionário que mede a usabilidade dos produtos de *Service Desk*, com base na norma NBR 9241-11.

6.1 Validação do Sistema

Para auxiliar a elaboração deste trabalho, foram utilizadas informações obtidas através de entrevistas informais com os responsáveis pelo CAU da UFSM, juntamente com registros do banco de dados do atual sistema utilizado pela CAU contendo dados do chamado como: (i) texto de abertura; (ii) data de abertura e data de fechamento; (iii) parecer final do técnico; Foram preservados dados como o nome do usuário que abriu o chamado. Alguns dos dados recebidos da CAU que foram importados para o banco de dados do sistema e utilizados na validação do módulo de recomendação de soluções, podem ser vistos na Tabela 6.1.

Tabela 6.1 – Descrição do problema e parecer final de chamados da CAU

Descrição	**Parecer Técnico**
Configuração de PC para acessar serpro - SIAPE	Foi reinstalado o Java e o Serpro.
O software Statistica foi instalado e não funcionou	Foi reeditado o arquivo stat.lic.
Computador não liga - aparece tela preta cheia de códigos	A tela preta estava relacionada um erro na BIOS
Dificuldade de instalação de impressora	Foi reinstalado o driver e compartilhado a impressora HP Laserjet P2055dn
Computador não liga	Foi feito CheckDisk e restauração do sistema para uma data anterior a ultima atualização do windows.
Registrar 80 Computadores na Rede, já estão cabeados e ligados	Foram enviados 80 MACs para que façam o cadastramento no CPD

Foram mais de trinta e cinco mil registros de chamados importados para o sistema, que compreendem aos chamados abertos no sistema durante o período entre 13 de Março de 2009 e 20 de Novembro de 2012. Além da utilização para validação das recomendações, foi possível realizar uma análise em cima destes dados, obtendo informações como número total de chamados e prazos de atendimento. A Figura 6.1 apresenta o total de chamados abertos por mês no período de 01 de Janeiro de 2012 até 20 de Novembro de 2012 que foi a data em que os dados foram solicitados ao CPD.

Analisando o gráfico é possível concluir que em Março/2012 (i.e, mês em que iniciou o

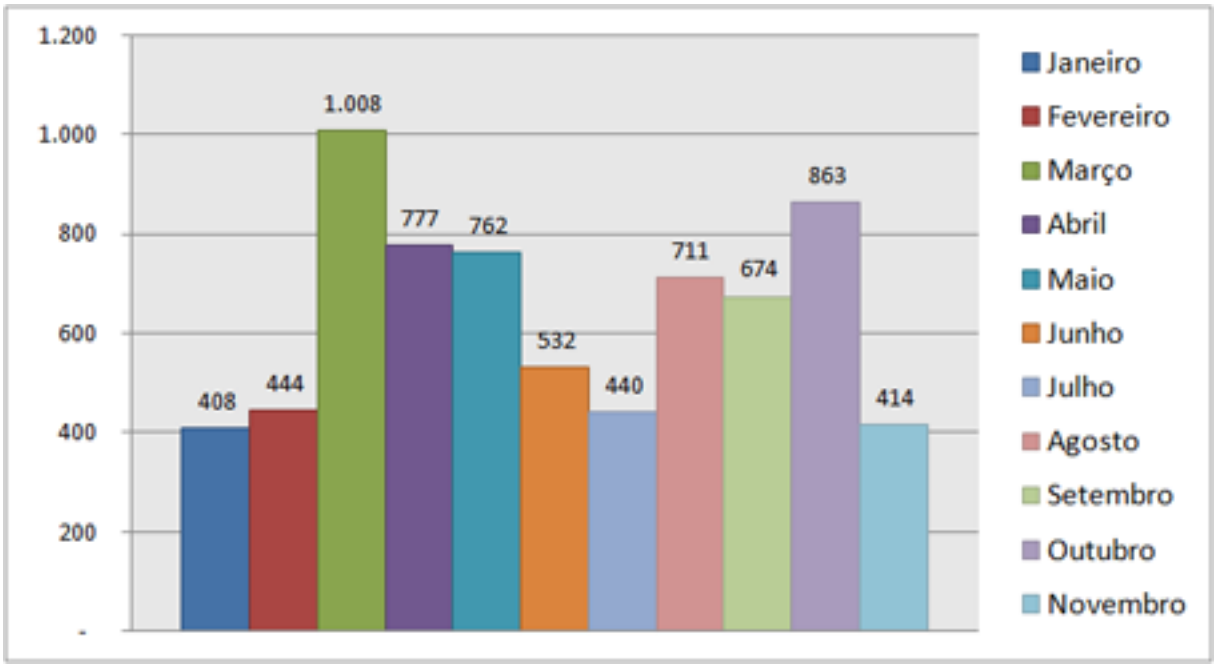

Figura 6.1 – Número mensal de chamados abertos no sistema da CAU em 2012 (CPD, 2012).

semestre letivo) foi o mês que os técnicos da CAU mais receberam chamados, num total de 1008 registros. Menos chamados foram registrados no período de férias entre Janeiro e Fevereiro, além dos meses de Junho e Julho quando ocorreu greve dos professores e funcionários.

Outras informações referentes aos dados importados estão apresentadas na Figura 6.2, que apresenta o prazo de atendimento aos chamados no ano de 2012.

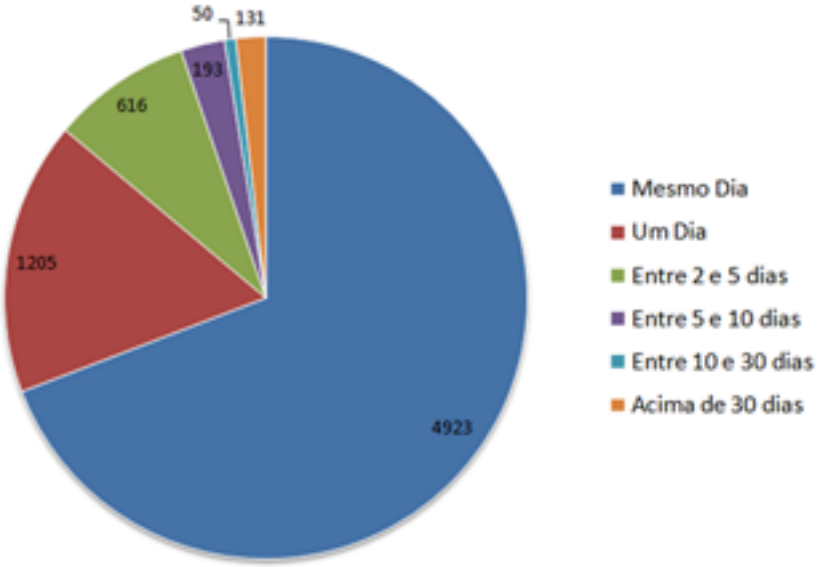

Figura 6.2 – Prazos para atendimento de chamados na CAU em 2012 (CPD, 2012).

Apesar desta proximidade com a CAU, não foi possível validar o sistema no seu ambiente, visto que o número de atendimentos, segundo os coordenadores entrevistados no CPD,

é relativamente alto para o número de técnicos que podem atender e utilizar dois sistemas ao mesmo tempo demandaria maior dedicação dos técnicos com os sistemas. Além disso, por ser uma instituição pública, seria necessário a autorização por parte da direção e abertura de um edital para compra dos dispositivos móveis que seriam utilizados na validação. Sendo assim, optou-se por validar o funcionamento do sistema através de uma simulação.

Este item apresentará a metodologia utilizada para a validação do sistema, onde serão detalhados o planejamento e a execução dos testes.

De acordo com MOREIRA et al. (2009):

> Para sistemas de comunicação em geral, e para sistemas sem fio, em particular, a experiência mostra que os resultados de simulação nem sempre correspondem aos obtidos em implementações reais. A simulação normalmente baseia-se em modelos simplificados, que não consideram aspectos importantes que surgem quando se implementa a proposta em um ambiente real. Há certo consenso que as propostas devem também ser testadas em ambientes o mais próximo possível de um ambiente real. (MOREIRA et al., 2009)

Neste trabalho ainda que se tenha utilizada uma simulação, a validação do sistema ocorreu em um ambiente real e com aparelhos do tipo *smartphones* e *tablets* próprios que possuíam GPS integrado. Os testes foram realizados no campus da UFSM em Santa Maria (RS) pelo período de 30 dias.

6.1.1 Metodologia

Os testes iniciais visaram validar o mecanismo de obtenção da localização do técnico através do dispositivo móvel. A segunda bateria de testes objetivou validar a fórmula de cálculo da distância entre o técnico e os prédios com chamados para atendimento. A terceira etapa teve como meta validar o relacionamento de chamados e as sugestões de possível solução. A fase final de testes serviu para validar o funcionamento e comportamento do sistema. Nesta última fase foram reportados incidentes no sistema com localização em diferentes prédios do campus.

Para a simulação foi montado um plano de testes com cenários que atingissem às diversas funcionalidades previstas para o sistema. Uma das vantagens de se utilizar o plano de testes elaborado é que se tornaria possível identificar facilmente erros do sistema, pois já se teria prévio conhecimento dos resultados esperados. Além da utilização de *smartphones* e *tablets* que foram utilizados nos testes, para a criação do *Mobile Service Desk*, foram usados computadores *desktop* e *notebooks*, na confecção dos modelos e na implementação do código e *softwares* para auxílio a modelagem e desenvolvimento.

Os itens a seguir apresentarão o plano de testes elaborado, o funcionamento do sistema

e telas que foram capturadas no momento da realização dos testes.

6.1.2 Plano de testes

Para a realização dos testes foram cadastradas no sistema quatro expertises: Suporte Básico, *Hardware*, *Software* e Rede. Cada *expertise* representa uma classificação que será utilizada para denominar o conhecimento que o técnico deve possuir para prestar o atendimento. É importante salientar que estas expertises não estão associadas ao ambiente da CAU e que foram utilizadas apenas como exemplo. Outro ponto importante de chamar a atenção é que o sistema não possui qualquer cadastro prévio de expertises, e que cada utilizador deverá manter o cadastro da forma que melhor se adaptar a sua necessidade.

Para distribuir o reporte de incidentes em diferentes localizações, foram cadastrados seis prédios: Centro de Ciências Naturais e Exatas, Centro de Ciências Rurais, Centro de Ciências da Saúde, Centro de Tecnologia, Colégio Politécnico e Reitoria, cujas localizações podem ser visualizadas na Figura 6.3.

Figura 6.3 – Mapa do Campus da UFSM, adaptado de (COPERVES, 2012)

A Figura 6.3 mostra o mapa do Campus Universitário da Universidade Federal de Santa Maria, local onde foram aplicados os testes. Na imagem é apresentada uma visão geral da localização dos prédios que foram utilizados no plano de testes.

Com relação aos usuários cadastrados, foram inseridos três técnicos com perfil para realizar atendimentos, um com perfil para classificar chamados, um gestor para administrar o sistema e seis usuários para abertura de chamados. Para os técnicos que realizam atendimentos de chamados, foram associadas *expertises* conforme demonstrado na Tabela 6.2 que apresenta um resumo dos usuários do sistema e técnicos, com suas respectivas *expertises*, cadastrados no *Mobile Service Desk*.

Tabela 6.2 – Relação de Usuários do *Mobile Service Desk*

Nome do Usuário	**Tipo de Usuário**	***Expertise***
Atendimento 1	Suporte	Suporte Básico, *Software*, *Hardware* e Rede.
Atendimento 2	Suporte	Suporte Básico, *Software*.
Atendimento 3	Suporte	Suporte Básico.
Classificador 1	Classificador	-
Gestor 1	Gestor	-

Foram registrados no sistema seis incidentes para diferentes locais, os quais foram classificados com diferentes *expertises*. A Tabela 6.3 mostra a relação dos incidentes cadastrados e as prioridades e expertises que estes devem receber.

Tabela 6.3 – Relação de Chamados Inseridos no *Mobile Service Desk*

Usuário	**Local**	***Descrição***	**Prioridade**	***Expertise***
Roseclea	CT Sala 386	Instalar o SIE nas máquinas da sala.	Média	*Software*
Gleizer	Colégio Politécnico	Configurar máquinas do laboratório 2 na rede.	Média	Suporte Básico
Felipe	CCNE Sala 223	Computador não está ligando.	Baixa	Suporte Básico
Andreia	CCS Sala 115	Trocar o HD de uma máquina na sala.	Baixa	*Hardware*
Henrique	Reitoria - 7º Andar	SIAPE não funciona nas máquinas da PRPGP	Alta	*Software*
Ricardo	Reitoria - 1º Andar	Configurar roteador sem fio.	Alta	Rede

Como pode ser visto na Tabela 6.3 foram classificadas chamados com prioridade Alta, Média e Baixa. O objetivo destas classificações é mostrar que o sistema sempre lançara no topo dos chamados a atender aqueles que possuem mais alta prioridade, independente da existência de chamado com localização mais próxima da posição do técnico. É possível observar ainda que foram cadastrados dois chamados de prioridade Alta, dois com prioridade Média e dois com prioridade Baixa. Desta forma pretendeu-se mostrar que o sistema apresentará os chamados de mesma prioridade ordenadas de acordo com a menor distância entre o técnico e o chamado.

6.1.3 Execução do plano de testes

Durante a fase de execução do plano de testes, foram realizadas observações de campo no ambiente da UFSM, analisando o comportamento do sistema no trajeto compreendido entre os locais de atendimento dos chamados. Este item apresentará como ocorreu a execução do plano de testes.

6.1.3.1 Classificação dos chamados

Ao ser aberto no sistema, os chamados ficaram disponíveis para o Classificador informar qual a prioridade e qual a *expertise* adequada para solução do incidente. A Figura 6.4 mostra capturas de tela realizadas no momento da classificação de chamados.

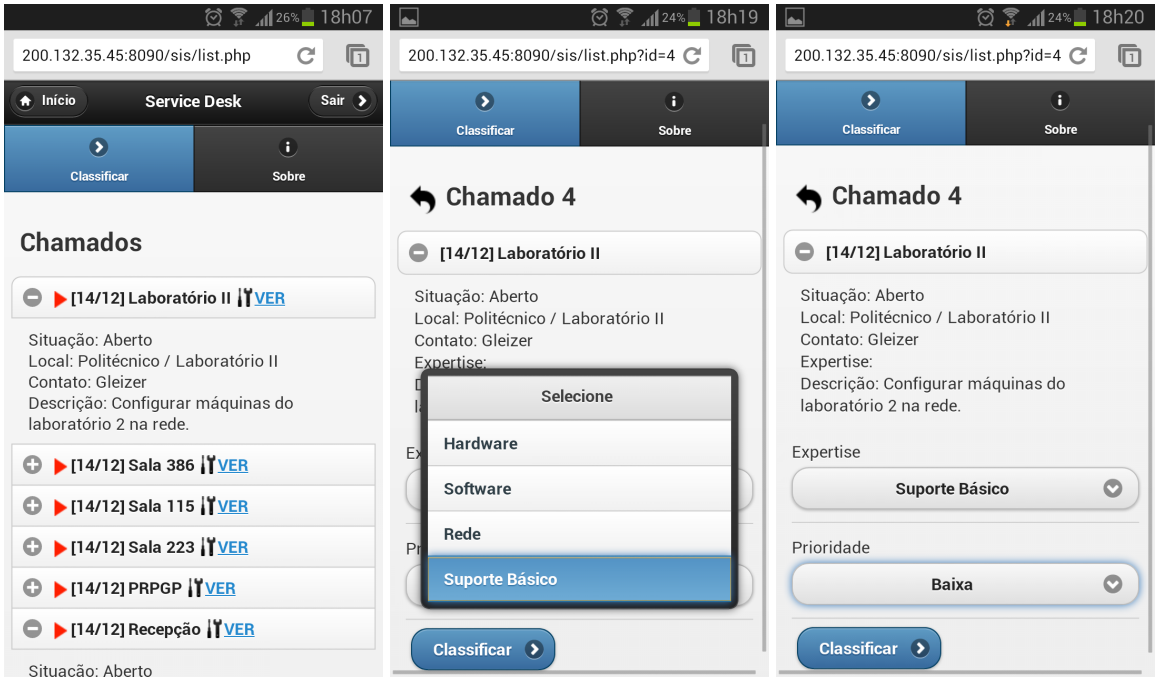

Figura 6.4 – Classificação dos chamados abertos para teste no *Mobile Service Desk*

6.1.3.2 Ordenação dos chamados

Após validar a classificação de chamados, o próximo passo foi a validação quanto à ordenação dos chamados, tanto com relação à prioridade quanto a posição geográfica do técnico. Foram feitos acessos ao sistema com os técnicos Atendimento 1, Atendimento 2 e Atendimento 3, que tinham o perfil para atendimento, com os técnicos Atendimento 1 e Atendimento 2 partindo em frente ao CPD e o técnico Atendimento 3 partindo em frente ao CT. Além dos

acessos registrados, durante o deslocamento entre o CPD, CT, e locais de atendimento, outros acessos foram realizados para verificar o comportamento do sistema. Os testes desta etapa mostraram os seguintes resultados:

6.1.3.2.1 Técnico Atendimento 1

O técnico Atendimento 1 que, de acordo com o plano de testes, possui as *expertises* "*Hardware*", "*Software*", "Rede" e "Suporte Básico", ao acessar o sistema foram listados os chamados em aberto que ele possui *expertise* para atender, ordenando-os levando em consideração os fatores de prioridade e localização de cada *ticket*.

No plano de testes foram inseridos seis chamados e classificados dois em cada prioridade (i.e, Alta, Média e Baixa). Conforme a proposta do sistema, os chamados devem ser ordenados de acordo com a sequência prioridade e localização. Levando em consideração que dois *tickets* possuam a mesma prioridade, o primeiro a ser exibido é aquele que está localizado mais próximo ao técnico. A Figura 6.5 mostra a tela que foi capturada durante o acesso com o técnico Atendimento 1.

Figura 6.5 – Lista de Chamados para o técnico Atendimento 1

A Figura 6.5 mostra que no momento da consulta existem dois chamados com prioridade Alta e localizados no prédio da Reitoria. Neste caso, além da distância até o prédio, é considerado a altura e priorizado o atendimento na Recepção (i.e, 1º andar) antes do atendimento na PRPGP (i.e, 7º andar), pois do local em que se encontrava o técnico era mais próximo a Recepção do que a PRPGP.

6.1.3.2.2 Técnico Atendimento 2

De acordo com o plano de testes, o técnico Atendimento 2 recebeu as *expertises* "*Software*" e "Suporte Básico", supondo que este técnico possui conhecimento sobre os programas que são utilizados no ambiente da Universidade e também pode prestar atendimento básico para os usuários. Desta forma, quando o técnico Atendimento 2 acessou o sistema, foram apresentados apenas os chamados que o técnico poderia atender. A Figura 6.6 mostra a tela que foi capturada durante o acesso do técnico Atendimento 2.

Figura 6.6 – Lista de Chamados para o técnico Atendimento 2

Considerando que o ponto de partida do técnico Atendimento 2 era no prédio do CPD,

os chamados apresentados na Figura 6.6 foram ordenados primeiramente de acordo com a prioridade. Como haviam dois incidentes de prioridade Média, o sistema priorizou de acordo com a distância, pois, de acordo com a Figura 6.3 que apresenta o mapa da UFSM, o Politécnico é mais próximo ao CPD em relação ao CT.

6.1.3.2.3 Técnico Atendimento 3

Por fim, o técnico Atendimento 3 que possuía a *expertise* "Suporte Básico", ao acessar o sistema teve listado apenas dois chamados: um no CCNE e outro no Politécnico.

Figura 6.7 – Lista de Chamados para o técnico Atendimento 3

No plano de testes, foi definido que este técnico teria como ponto de partida o CT. Neste caso, como apresentado na Figura 6.7, mesmo que o CCNE seja mais próximo do local onde o técnico estava em relação ao Politécnico, o chamado para o Politécnico é colocado no topo da lista de atendimento, pois possui prioridade maior que o chamado do CCNE.

6.1.4 Análise de Similaridade

Os testes nos algoritmos de análise de similaridade foram realizados utilizando dados reais, capturados do banco de dados da CAU, contendo os chamados abertos entre Janeiro e Novembro de 2012, num total 7033 chamados. Para testar o tempo de execução dos algoritmos,

bem como a qualidade dos resultados, foram abertos no sistema os chamados apresentados na Tabela 6.4.

Tabela 6.4 – Textos para teste dos algoritmos de Análise de Similaridade

Nº	Texto do Chamado
1	Computador com mensagem de falsificação de software
2	Meu computador está extremamente lento. Desconfio que seja algum vírus
3	Ocorre erro de spooler na impressão!!!

Após o pré-processamento dos textos de abertura de chamados foi obtido o resultado apresentado na Tabela 6.5. Este texto resultante foi utilizado para comparar a similaridade entre eles e os chamados da CAU importados para o sistema.

Tabela 6.5 – Resultado do pré-processamento dos Textos

Nº	Texto pré-processado
1	comput mens falsificaca softw
2	comput extrem lent desconfi viru
3	ocorr err spool impressa

Para o teste foram utilizados quatro algoritmos que calculam similaridade, são eles: (i) Algoritmo Jaro-Winkler (WINKLER, 1990); (ii) Algoritmo Levenshtein Distance (LEVENSHTEIN, 1966); (iii) Algoritmo String Similarity, que é uma classe usada para calcular a semelhança entre duas cadeias de texto, criada a partir do algoritmo *diff* que compara a diferença entre arquivos no GNU (GNU Operating System) (GNU, 2012); (iv) A função *similar_text()*, nativa do PHP para cálculo de similaridade (PHP, 2012). As implementações dos algoritmos de Rabin-Karp KARP; RABIN (1987) e Boyer-Moore BOYER; MOORE (1977) localizadas não retornavam o percentual de similaridade entre as *strings*, mas sim a distância entre uma *string* em relação a outra. Desta forma, foram descartados dos testes.

Ao executar o teste referente ao primeiro chamado aberto no sistema "*Computador com mensagem de falsificação de software*", o resultado contendo os cinco casos com maior similaridade, obtidos utilizando os algoritmos, estão expressos nas Tabelas 6.6, 6.7, 6.8 e 6.9.

Nestas tabelas, a primeira coluna traz o texto de abertura para o chamado similar, a segunda coluna mostra o mesmo chamado após serem aplicadas as técnicas de pré-processamento de texto. O resultado deste pré-processamento é comparado ao texto do chamado que também foi pré-processado, no primeiro caso testado foi "*comput mens falsificaca softw*". Por fim, na terceira coluna está apresentado o percentual de similaridade entre estes registros e o caso que foi verificado.

Tabela 6.6 – Teste do Chamado 1 para o Algoritmo Jaro-Winkler

Texto do Chamado	**Texto Pré-Processado**	**Similar**
computador infectado e instalação do SIE.	comput infect instalaca sie	87,94 %
Computador com processamento muito lento e com mensagem de falsificação de software.	comput process lent mens falsificaca softw	87,28 %
Computador com som falhando!	comput som falh	85,72 %
Computador não emitindo sinal	comput emit sinal	85,31 %
O computador não está acesando o MSN.	comput aces msn	85,09 %

Tabela 6.7 – Teste do Chamado 1 para o Algoritmo Levenshtein Distance

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Computador com processamento muito lento e com mensagem de falsificação de software.	comput process lent mens falsificaca softw	95,39 %
estou voltando das ferias e o pc está com essa mensagem. "voce pode ter sido vitima de falsificaçao de software"	volt fer pc mens vit falsificaca softw	83 %
O Windows esta com mensagem de falsificação de software.	window mens falsificaca softw	81,44 %
Bom dia! Gostaria de solicitar serviços, pois meu PC está há vários dias com a seguinte mensagem "você pode ter sido vítima de falsificação de software. Obrigada, Cleusa	gost solic serv pc vari seguint mens vit falsificaca softw obrig cleus	81,32 %
O SIE não está funcionando há 3 dias, desde que começou a aparecer a mensagem Windows detectou um problema de falsificação de software.	sie funcion comec aparec mens window detect problem falsificaca softw	81,08 %

Tabela 6.8 – Teste do Chamado 1 para o Algoritmo String Similarity

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Mensagem de falsificação de software.	mens falsificaca softw	84,61 %
mensagem de falsificação de software	mens falsificaca softw	84,61 %
O Windows esta com mensagem de falsificação de software.	window mens falsificaca softw	81,35 %
Computador com processamento muito lento e com mensagem de falsificação de software.	comput process lent mens falsificaca softw	80,55 %
estou voltando das ferias e o pc está com essa mensagem. "voce pode ter sido vitima de falsificaçao de software"	volt fer pc mens vit falsificaca softw	70,58 %

Tabela 6.9 – Teste do Chamado 1 para o Algoritmo *similar_text()*

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Mensagem de falsificação de software.	mens falsificaca softw	84,61 %
mensagem de falsificação de software	mens falsificaca softw	84,61 %
O Windows esta com mensagem de falsificação de software.	window mens falsificaca softw	81,35 %
Computador com processamento muito lento e com mensagem de falsificação de software.	comput process lent mens falsificaca softw	80,55 %
estou voltando das ferias e o pc está com essa mensagem. "voce pode ter sido vitima de falsificaçao de software"	volt fer pc mens vit falsificaca softw	73,52 %

Analisando os resultados obtidos no primeiro teste, é possível concluir que o algoritmo de Jaro-Winkler não trouxe bons resultados, pois relacionou como similar casos onde o usuário relata problemas com vírus, problemas com falhas no som, falha no acesso ao MSN (Messenger) e computador sem sinal ao problema de computador lento. Já nos demais algoritmos, os casos mais similares retornados por eles realmente são relativos ao problema de computador lento, com destaque para o algoritmo de Levenshtein Distance que obteve o percentual de si-

milaridade mais alto em relação aos algoritmos de String Similarity e *similar_text()*, o que dá mais confiança para o resultado obtido.

O teste do segundo chamado aberto no sistema "*Meu computador está extremamente lento. Desconfio que seja algum vírus*", retornou os resultados expressos nas Tabelas 6.10, 6.11, 6.12 e 6.13.

Tabela 6.10 – Teste do Chamado 2 para o Algoritmo Jaro-Winkler

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Computador extremamente lento e, eventualmente travando.	comput extrem lent event trav	89,75 %
Computador necessita ser formatado, pois está com excesso de vírus.	comput necessit format excess viru	88,83 %
Computador não liga normalmente, infectado por virus.	comput lig norm infect viru	88,28 %
Computador extremamente lento e trancando. Veio um técnico mas parece que ficou pior.	comput extrem lent tranc vei tecn fic pi	87,94 %
Computador lento e configurar a impressora	comput lent configur impress	87,25 %

Tabela 6.11 – Teste do Chamado 2 para o Algoritmo Levenshtein Distance

Texto do Chamado	**Texto Pré-Processado**	**Similar**
O computador está muito lento, e alguns arquivos da área de trabalho sumiram. Desconfiamos que esteja com vírus.	comput lent arqu are trabalh sum desconfi estej viru	79,17 %
Computador extremamente lento e trancando. Veio um técnico mas parece que ficou pior.	comput extrem lent tranc vei tecn fic pi	78,41 %
Computador extremamente lento para entrar na internet. As vezes tem que ser reiniciado e mesmo assim não consegue entrar na internet. Demora para abrir os e-mails e não consegue enviar a resposta do e-mail.	comput extrem lent entr internet vez reinici assim conse dem abr email envi respost email	76,71 %
Computador não está entrando na internet e nem no SIE. Está sem rede. Após configurar impressora em rede.	comput entr internet sie red configur impress	73,65 %
Computador extremamente lento e, eventualmente travando.	comput extrem lent event trav	71,25 %

Tabela 6.12 – Teste do Chamado 2 para o Algoritmo String Similarity

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Computador extremamente lento e, eventualmente travando.	comput extrem lent event trav	77,96 %
Computador lento e escaner.	comput lent escan	72,34 %
Computador extremamente lento e trancando. Veio um técnico mas parece que ficou pior.	comput extrem lent tranc vei tecn fic pi	71,42 %
Computador e retro-projetor desconfigurado	comput retroproje desconfigur	71,18 %
Computadores com sistema muito lento	comput sistem lent	66,66 %

Tabela 6.13 – Teste do Chamado 2 para o Algoritmo *similar_text()*

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Computador extremamente lento e, eventualmente travando.	comput extrem lent event trav	74,19 %
Computador extremamente lento e trancando. Veio um técnico mas parece que ficou pior.	comput extrem lent tranc vei tecn fic pi	71,23 %
computador lento, possível vírus	comput lent possi viru	68 %
Computador muito lento...deve ter virus....	comput lentodev viru	67,92 %

Ao analisar os dados do segundo teste dos algoritmos de similaridade, é possível concluir que os cinco registros com maior similaridade capturados pelo algoritmo Jaro-Winkler estão ao menos relacionado com o problema de "*computador lento*". Já no algoritmo de Levenshtein Distance, dois dos resultados não tem relação com o problema, pois um se trata de uma lentidão apenas no navegador e outro está sem acesso a rede. No String Similarity, o resultado "*Computador e retro-projetor desconfigurado*" não tem relação com o problema. Por fim, no *similar_text()* os cinco melhores resultados são relativos ao problema.

Desta forma, é possível concluir que, neste segundo teste, os algoritmos Levenshtein Distance e String Similarity não retornaram um bom resultado, visto que definiram como similar casos que não tinham relação com o problema.

Por fim, o terceiro teste realizado através do chamado "*Ocorre erro de spooler na impressão!!!*", após a execução dos algoritmos de similaridade obteve como resultados o que está apresentado nas Tabelas 6.14, 6.15, 6.16 e 6.17.

Tabela 6.14 – Teste do Chamado 3 para o Algoritmo Jaro-Winkler

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Ocorreu um problema na galeria da página do centro de Educação, solicito pessoal especializado em Páginas. OBR. Aguardo.	ocorr problem gal pag centr educaca solicit especi pag obr aguard	75,45 %
Esta ocorrendo enceramento dos programas, tanto do oficce quanto de páginas de internet. Em algumas vezes aparece a mensagem de memória insuficiente ou que ocorreu falha no sistema e em outras vezes os programas são simplesmente finalizados.	ocorr encer programas tant oficc pag internet vez aparec mens memor insufici ocorr falh sistem simples final	73,85 %
Não ocorre a inicialização. A fonte está funcionando normalmente.	ocorr inicializaca font funcion norm	73,15 %
Está ocorrendo um erro na inicialização do computador. Já ocorreu a visita do técnico do CPD o qual informou que está ocorrendo um erro em virtude da própria atualização do computador e que provavelmente terá que ser formatado.	ocorr err inicializaca comput ocorr visit tecn cpd inform virtud atualizaca prova format	72,81 %
Não consigo imprimir, erro na impressão.	consig imprim err impressa	72,65 %

Tabela 6.15 – Teste do Chamado 3 para o Algoritmo Levenshtein Distance

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Impressora configurada e não responde às solicitações de impressão.	impress configur respond solicitaco impressa	75,47 %
Computador formatado ontem precisa ser colocado em rede. Inatalar impressora.	comput format ont precis coloc red inatal impress	74,06 %
Reinstalar e configurar na rede serpro uma impressora matricial Epson fx 2180 .Obs. para impressão de relatórios contínuos.	reinstal configur red serpr impress matric epson fx ob impressa relato continu	73,33 %
Instalar serviço de spooler, impressoras desativadas	instal serv spool impress desativ	72 %
Peço para retirar de uma sala e instalar em outra um computador e colocar em rede o mesmo com a impressora.	pec retir sal instal comput coloc red impress	70,99 %

Tabela 6.16 – Teste do Chamado 3 para o Algoritmo String Similarity

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Instalar serviço de spooler, impressoras desativadas	instal serv spool impress desativ	57,14 %
Não consigo imprimir, erro na impressão.	consig imprim err impressa	57,14 %
Solucionar problema com impressora	soluc problem impress	54,90 %
Solucionar problema com impressora	soluc problem impress	54,90 %
Demora p/ impressão.	dem p impressa	54,54 %

Tabela 6.17 – Teste do Chamado 3 para o Algoritmo *similar_text()*

Texto do Chamado	**Texto Pré-Processado**	**Similar**
Não consigo imprimir, erro na impressão.	consig imprim err impressa	62,74 %
Instalar serviço de spooler, impressoras desativadas	instal serv spool impress desativ	62,06 %
porblemas de impressão.	porblemas impressa	60,46 %
Não dá a ordem para impressão	ord impressa	59,45 %
Reinstalar impressora..	reinstal impress	58,53 %

Analisando os dados do terceiro teste dos algoritmos de similaridade, é possível concluir que o algoritmo Jaro-Winkler não obteve um bom resultado, visto que os quatro primeiros registros mais similares que ele encontrou não tem relação com o problema. Nos algoritmos Levenshtein Distance, String Similarity e *similar_text()*, todos os resultados tem relação com problemas de impressão, além de trazerem um resultado relacionado ao problema de *spooler*.

Ao comparar cada um dos três textos de abertura dos chamados com cada registro no banco do sistema, os algoritmos de análise de similaridade tiveram desempenhos bastante distintos. O que obteve a menor média de tempo para realizar todos os cálculos, conforme o gráfico apresentado na Figura 6.8, foi o algoritmo da função *similar_text()* com uma média de 0,16 segundos. O algoritmo Levenshtein Distance foi o que apresentou o pior desempenho, levando em média 125,85 segundos para realizar a tarefa. Já os algoritmos Jaro-Winkler (8,45 segundos) e String Similarity (26,59 segundos) tiveram desempenho intermediário.

Este comportamento distinto pode ser devido ao fato de que o algoritmo *similar_text()* é uma função nativa do PHP, portanto tem sua execução otimizada em relação a outros algoritmos. Já o Levenshtein Distance, que obteve o pior desempenho, não é um bom algoritmo para análise de um grande conjunto de palavras, visto que realiza suas comparações de similaridade pelo cálculo do número de ações necessárias para transformar uma *string* em outra. O tempo de execução vai impactar no custo de processamento que um algoritmo vai ter para relacionar um novo caso inserido no sistema com os demais já conhecidos e vai aumentando na medida em que novos casos são inseridos no sistema, desta forma, o algoritmo Levenshtein Distance não se mostra como uma boa opção para estes testes.

Para medir o tempo de execução de cada algoritmo, no início do processamento foi

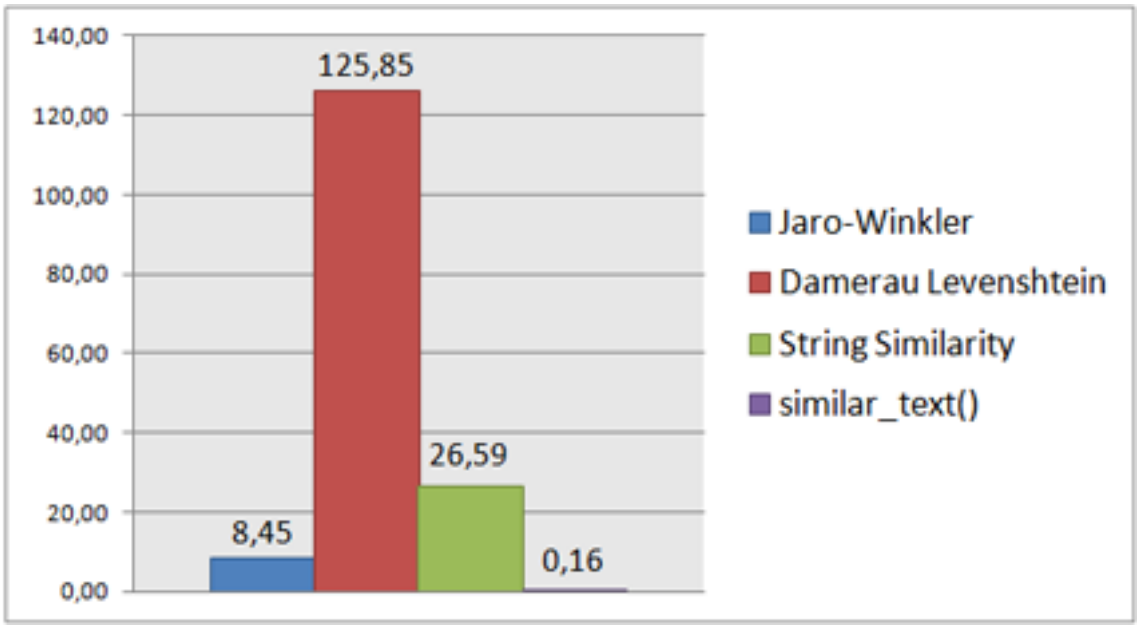

Figura 6.8 – Desempenho dos Algoritmos de Análise de Similaridade Testados (em segundos)

detectado o *timestamp* do servidor e ao término este valor foi subtraído do *timestamp* atual, desta forma obteve-se o tempo de execução em microsegundos que foi convertido para o valor em segundos e apresentado em forma de média entre todos os testes na Figura 6.8.

6.2 Avaliação de Interfaces Utilizando a NBR 9241-11

A avaliação do *Mobile Service Desk* e outras ferramentas de *Service Desk*, visando a usabilidade, é necessária para que seja garantido que os usuários possam alcançar seus objetivos e satisfaçam suas necessidades em um contexto particular de uso. Para essa avaliação, existe a norma NBR 9241-11, que foi elaborada pela (ABNT, 2002) tendo como base a ISO 9241-11. Esta norma esclarece os benefícios de medir usabilidade em termos de desempenho e satisfação do usuário, que são medidos pela extensão na qual os objetivos pretendidos de uso são alcançados, pelos recursos gastos para alcançar os objetivos pretendidos e pela extensão na qual o usuário considera aceitável o uso do produto (ABNT, 2002).

A NBR 9241-11 inclui orientações sobre como a usabilidade de um produto pode ser especificada e avaliada, ela se aplica tanto a produtos de uso geral quanto a produtos sendo adquiridos ou sendo desenvolvidos. A orientação inclui procedimentos para medir usabilidade, mas, não detalha todas as atividades a serem realizadas. A especificação de métodos de medidas detalhados baseados no usuário está além do objetivo da norma (ABNT, 2002).

Desta forma, seguindo orientações da NBR 9241-11, para realizar uma pesquisa de le-

vantamento foi elaborado um questionário contendo doze questões divididas em três quesitos a serem avaliados: (i) Facilidade de Aprender; (ii) Usabilidade Global do Sistema; (iii) Legibilidade. O questionário aplicado está no Apêndice A e também disponível no endereço <http://goog.gl/GQg9p>.

No Apêndice C estão as propriedades desejáveis do produto e as especificações, que, de acordo com a NBR 9241-11, são necessários para que norteiem à execução das avaliações. Para avaliar a facilidade de aprender e a usabilidade global do sistema, os avaliadores precisavam realizar as seguintes tarefas: Cadastrar um usuário; Efetuar *login* no sistema com o usuário cadastrado; Abrir um chamado; Atender um chamado (i.e, finalizar). Já para a avaliação da legibilidade as tarefas foram: Efetuar *login* no sistema; Navegar entre os cadastros de usuários e chamados; Ler texto dos chamados.

Quanto à facilidade de aprender, os avaliadores analisaram a compreensibilidade com que cada um aprende a realizar as tarefas nos sistemas testados, se o tempo decorrido desta atividade foi aceitável e se houveram muitas dificuldades para aprender a executar as atividades. Com relação à usabilidade global do sistema, foi avaliado se as tarefas foram realizadas, qual o tempo gasto para a sua execução e se a forma de funcionamento e disposição das informações nas ferramentas auxilia no processo de realização das atividades. Na legibilidade foram avaliadas a leitura e navegação entre as áreas do sistema, além do tempo decorrido para estas atividades e o nível de satisfação dos usuários quanto ao modo como a interface foi desenvolvida.

O questionário foi respondido por quatro Mestrandos do Programa de Pós-Graduação em Informática da UFSM, com conhecimentos avançados em informática, onde apenas um deles possuía prévio conhecimento de todas as ferramentas testadas, os demais tiveram o primeiro contato com as ferramentas durante a aplicação da pesquisa. Foram coletadas respostas referentes a sete ferramentas de *Service Desk*: (i) DotProject; (ii) eTicket; (iii) *Mobile Service Desk*; (iv) OcoMon; (v) ProManager; (vi) Spicework; (vii) Trellis Desk. Para a realização dos testes foram utilizados *smartphones* e *tablets* rodando tanto o sistema operacional Android quanto o iOS e com tamanhos de tela variando de 4 a 10 polegadas.

Quanto à avaliação dos testes realizados, foi feita seguindo os conceitos da escala Likert. Os questionamentos deste tipo de escala servem para indicar o quanto alguém concorda ou discorda, aprova ou reprova, acredita que seja verdadeiro ou falso uma determinada afirmação (LIKERT, 1932). Não existe nenhuma abordagem perfeita para seguir a escala de Likert, no en-

tanto para se ter uma visão equilibrada da situação. Uma escala de Likert acrescenta respostas a afirmações de alguma atitude particular, o que permite a um participante fornecer um *feedback* ligeiramente mais extenso que uma simples questão fechada mas que é mais facilmente quantificada do que uma resposta completamente em aberto. A escala enumera um conjunto de declarações e prevê um número determinado de respostas onde cada uma das alternativas recebe um valor específico diferente que serve para quantificar o resultado final do questionamento (LIKERT, 1932).

Para mensurar as respostas dos avaliadores, alternativas como "Sim", "Parcialmente" e "Não" foram quantificadas, respectivamente, com "10", "5" e "0" pontos. Da mesma forma, as alternativas "Satisfeito" (10 pontos), "Pouco Satisfeito" (5 pontos) e "Insatisfeito" (0 ponto). Já a verificação dos tempos se deu com a utilização de aplicativos que permitiram a cronometragem do tempo tanto nos *smartphones* quanto nos *tablets* e *desktop*. Os resultados das avaliações, bem como uma discussão em torno dos mesmos são apresentados nas próximas seções.

6.2.1 Facilidade de Aprender

Nesta avaliação, foi medida pelos avaliadores a facilidade com que cada um aprende a realizar as tarefas nos sistemas testados.

Com relação a facilidade de aprender as tarefas a serem realizadas nas ferramentas os avaliadores tinham três respostas possíveis: (i) "Sim", quando realizou todas as tarefas sem necessidade de ajuda; (ii) "Parcialmente", quando realizou as tarefas, mas precisou de alguma ajuda para realizar; (iii) "Não", quando alguma tarefa não pode ser executada.

Após a quantificação e tabulação dos dados, conforme ilustra a Figura 6.9, é possível observar que não houve uma grande predominância entre as ferramentas, sendo que todas as atividades solicitadas pelos usuários na etapa de testes foram aprendidas. A eTicket, *Mobile Service Desk* e Trellis Desk obtiveram pontuação máxima (40), o que indica que todas as tarefas foram aprendidas com sucesso pelos avaliadores. Nas ferramentas DotProjetc e Spiceworks apenas um avaliador realizou parcialmente as tarefas, alcançando uma pontuação intermediária (35). As ferramentas OcoMon e ProManager formam classificadas inferiormente (30), pois mais de um avaliador teve dificuldade em aprender as tarefas avaliadas.

Na escala de satisfação em relação à facilidade de aprender a utilizar o sistema os avaliadores tinham três respostas possíveis: (i) "Satisfeito", quando ficou plenamente satisfeito com a facilidade em aprender a utilizar as ferramentas; (ii) "Parcialmente Satisfeito", quando sua

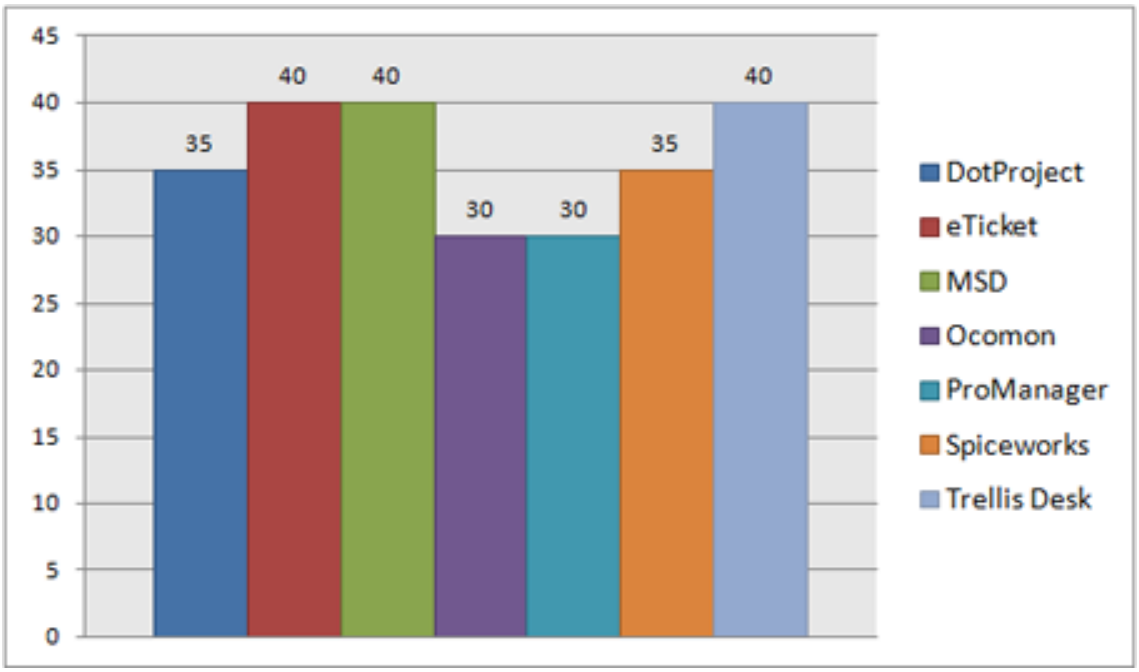

Figura 6.9 – Facilidade de Aprender: Foram Aprendidas Todas as Tarefas?

satisfação em relação a facilidade de aprender não foi plena; (iii) "Insatisfeito", quando sentiu muitas dificuldades em aprender a utilizar o sistema.

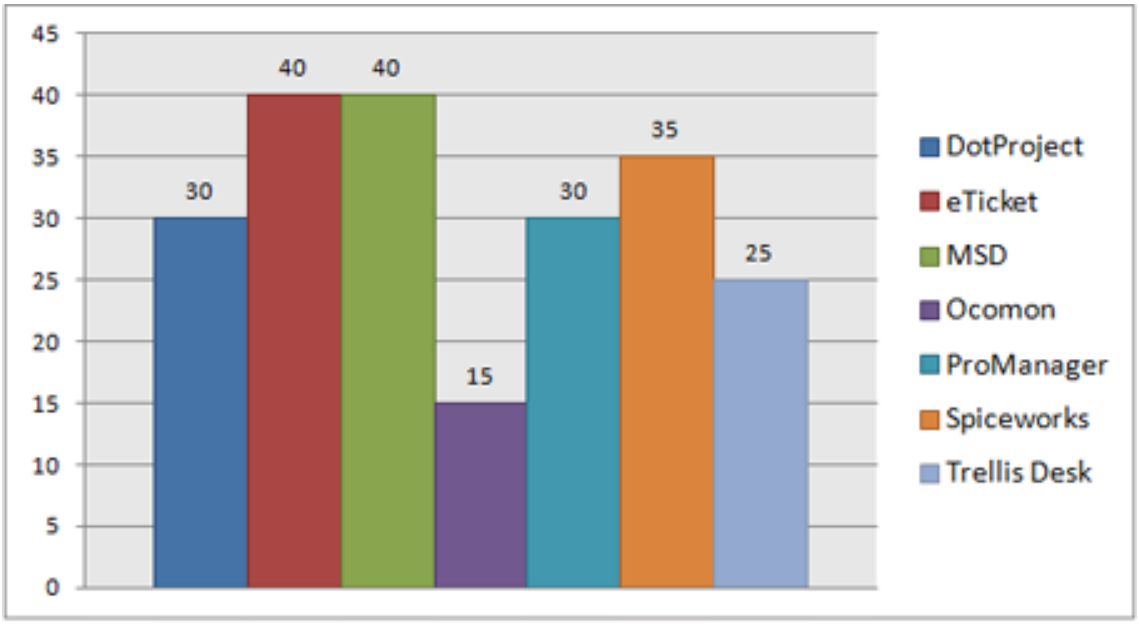

Figura 6.10 – Facilidade de Aprender: Qual a Escala de Satisfação em Relação à Facilidade de Aprender a Utilizar o Sistema?

Ao analisar os resultados da avaliação, expressos na Figura 6.10, é possível notar que as ferramentas eTicket e *Mobile Service Desk* obtiveram máxima satisfação dos avaliadores (40), por serem objetivas, seguidas pela Spiceworks (35). As ferramentas Dotproject (30),

ProManager (30) e Trellis Desk (25) ficaram em desvantagem, pois, segundo os avaliadores, não existem muitas informações que auxiliem o usuário quanto conclusão e criação de uma tarefa. A OcoMon foi a ferramenta de menor satisfação (15), em relação à facilidade de aprendizagem, pois sua interface complexa trouxe dificuldade para o aprendizado.

A Figura 6.11 apresenta a média de tempo gasto para realizar as tarefas referentes ao tópico de facilidade de aprendizado. A ferramenta com o pior desempenho em relação ao tempo gasto foi a OcoMon (20m42s), esse tempo elevado provavelmente deve-se a apresentação de formulários extensos e campos que para serem acessados necessitam que o usuário role a tela tanto verticalmente quanto horizontalmente. As ferramentas Trellis Desk (13m32s), eTicket (12m02s), Pro Manager (11m18s) e DotProjetc (09m48s) possuem um tempo em escala decrescente e aproximado por apresentarem a interface objetiva em relação a OcoMon. Os menores tempos para aprendizado das tarefas foram obtidos nas ferramentas Spiceworks (08m38s) e *Mobile Service Desk* (06m33s), um dos fatores que podem ter contribuído para estes resultados é que, segundo os avaliadores, estas ferramentas possuem uma interface mais objetiva que facilita o aprendizado.

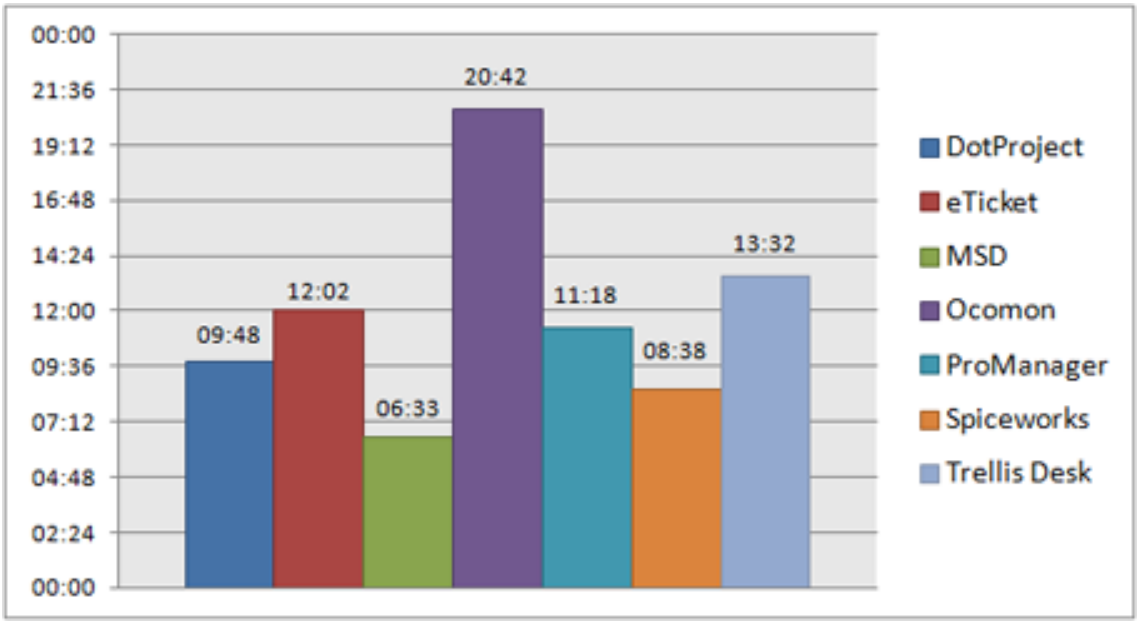

Figura 6.11 – Facilidade de Aprender: Média de Tempo Gasto para Realizar as Tarefas

6.2.2 Usabilidade Global do Sistema

Os dados apresentados na Figura 6.12 apresentam a satisfação dos usuários no que diz respeito à usabilidade do sistema. A ferramenta OcoMon obteve a classificação mais baixa

dentre as analisadas (15), na qual foram apontados como principais dificuldades a interface complicada de se utilizar, não intuitiva, formulários extensos para preenchimento, travamento da aplicação durante sua utilização, além do problema de necessitar rolar a tela verticalmente e horizontalmente que foi mencionado na avaliação anterior.

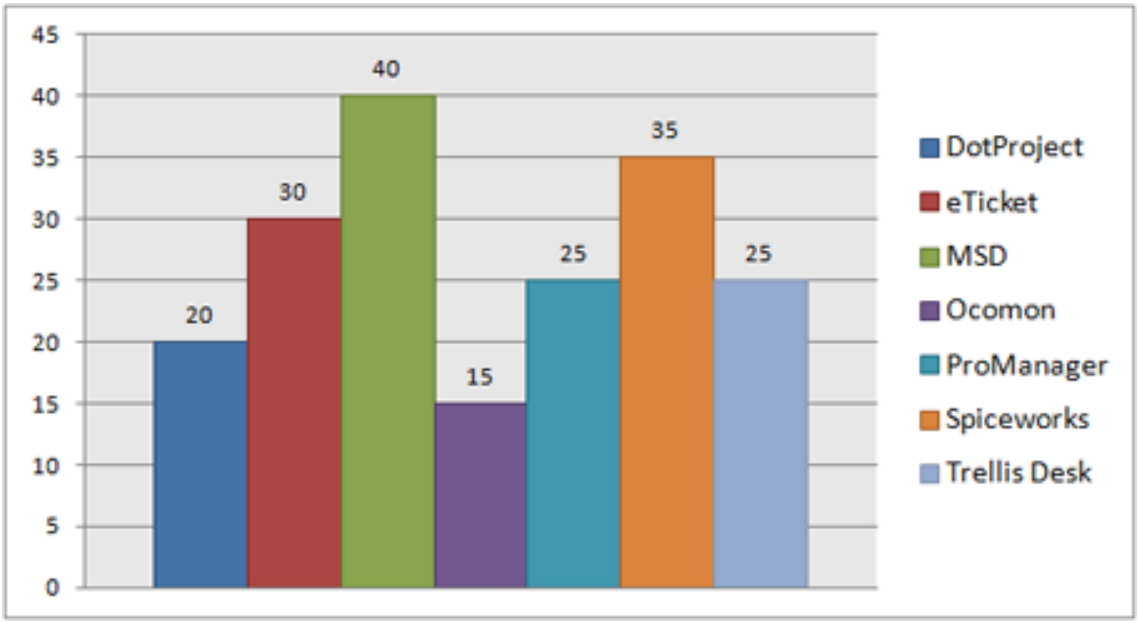

Figura 6.12 – Usabilidade Global: Qual a Escala de Satisfação em Relação a Usabilidade Global do Sistema?

O *Mobile Service Desk* e o Spiceworks atingiram as melhores classificações (40 e 35, respectivamente), o que os avaliadores destacaram destas ferramentas foram suas interfaces intuitivas, contendo somente as informações necessárias, de forma clara e visíveis. As demais ferramentas obtiveram classificações medianas (entre 30 e 20), tendo como aspectos relevantes as caixas de texto e botões de tamanho pequeno, assim como o preenchimento de formulários extensos, dificultando a usabilidade, principalmente por parte dos usuários de dispositivos móveis.

Quanto ao tempo decorrido para a realização das tarefas, apresentado na Figura 6.13, a ferramenta OcoMon atingiu a média de tempo mais elevada (19m44s). A *Mobile Service Desk*, eTicket e Spiceworks tiveram o menor tempo de utilização (06m22s, 07m21s e 07m33s, respectivamente).

Ao agregar os resultados obtidos nos três gráficos é possível observar que a ferramenta OcoMon obteve o menor índice (15) e o tempo mais elevado (19m44s), obtendo a classificação mais baixa dentre as ferramentas analisadas no quesito de usabilidade global. Em contrapartida a *Mobile Service Desk* obteve a classificação mais alta (40) e o menor tempo (06m22s), fi-

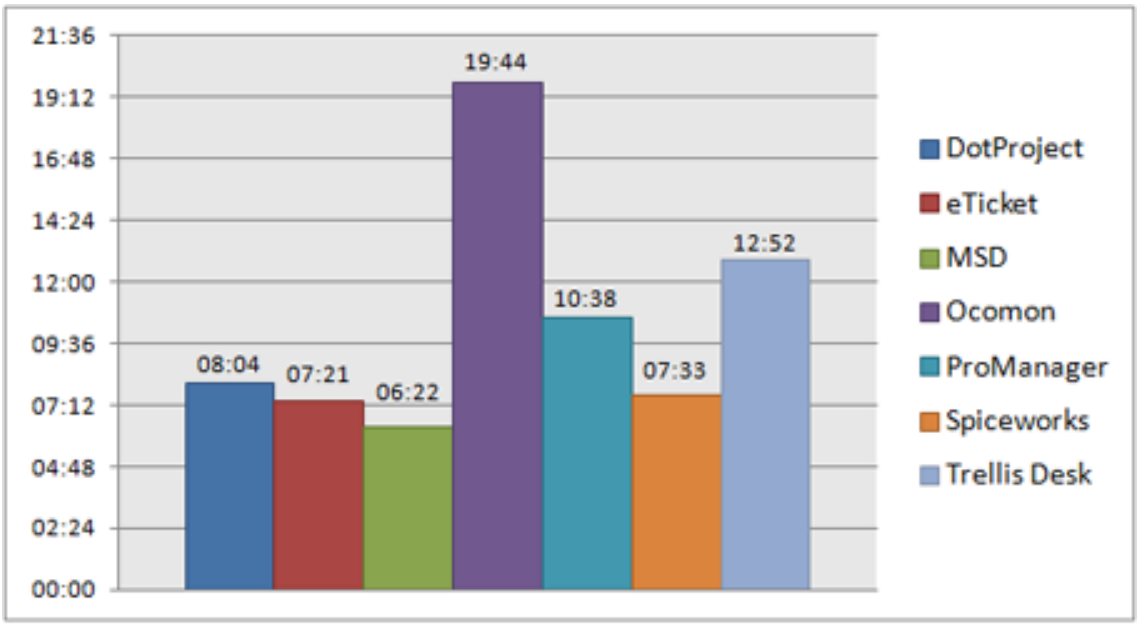

Figura 6.13 – Usabilidade Global: Média de Tempo Gasto para Realizar as Tarefas

cando com a melhorar avaliação de usabilidade global. As ferramentas eTicket e Spiceworks obtiveram bons resultados (30 e 35 pontos), ficando, respectivamente, com a segunda e terceira melhor avaliação de usabilidade global.

Esse resultado demonstra que a ferramenta OcoMon não dispõe de uma boa usabilidade, pois apresenta problemas de interface que atrapalham a utilização e influenciam diretamente no tempo consumido para realizar as tarefas propostas pela avaliação. Através desse resultado é possível relacionar uma boa usabilidade auxilia à otimização do tempo que o usuário leva para executar as tarefas, tornando-a um ponto importante para um sistema de *Service Desk*.

6.2.3 Legibilidade

Com relação à leitura e navegação entre as áreas do sistema, a Figura 6.14 mostra que não houve grande disparidade entre as ferramentas, ou seja, os usuários conseguiram realizar todas as tarefas, no entanto a ferramenta ProManager ficou um pouco aquém das demais (30), isso justifica-se, conforme os avaliadores, pelo fato da interface da ferramenta ser composta de menus com textos pequenos, dificultando a leitura e a navegação entre as áreas do sistema, o que levou alguns usuários a realizarem as tarefas apenas parcialmente.

Por outro lado, entre as que obtiveram as melhores pontuações nem todas possuíam interfaces objetivas e com tamanhos de fontes adequadas, nos comentários e respeito da ferramenta eTicket, por exemplo, os usuários descreveram que a interface possui botões e links

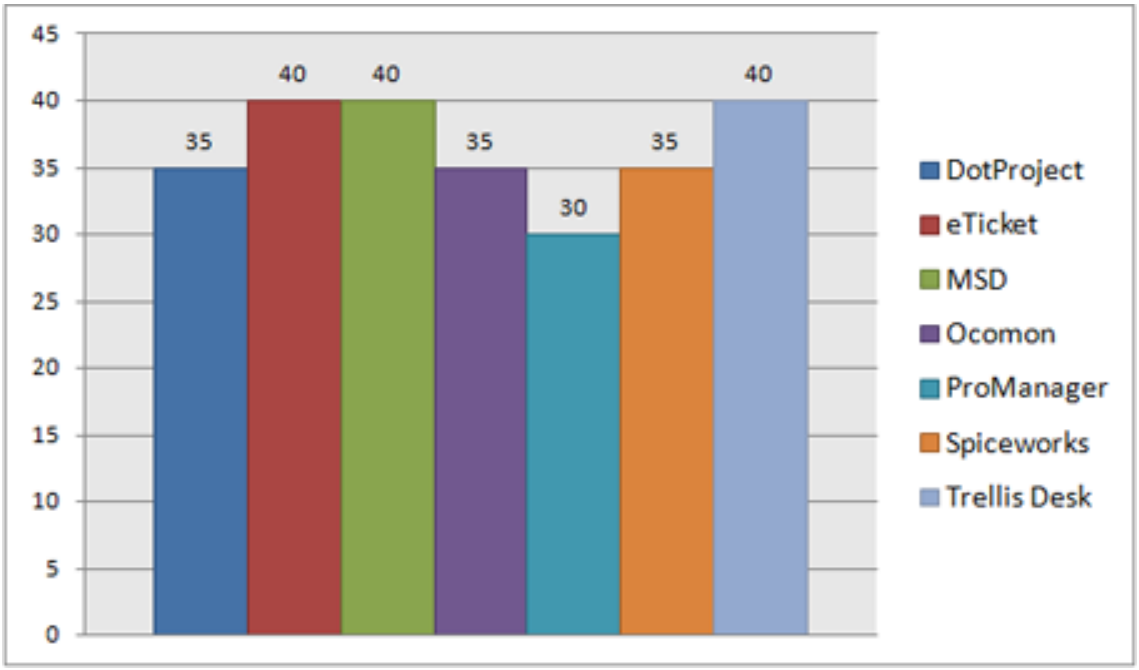

Figura 6.14 – Legibilidade: É Possível Ler os Textos e Navegar entre as Áreas do Sistema?

pequenos que dificultam a navegação, quando utilizada em dispositivos com telas pequenas. Porém, esse aspecto não impediu a realização das tarefas, o que justificou a pontuação máxima (40).

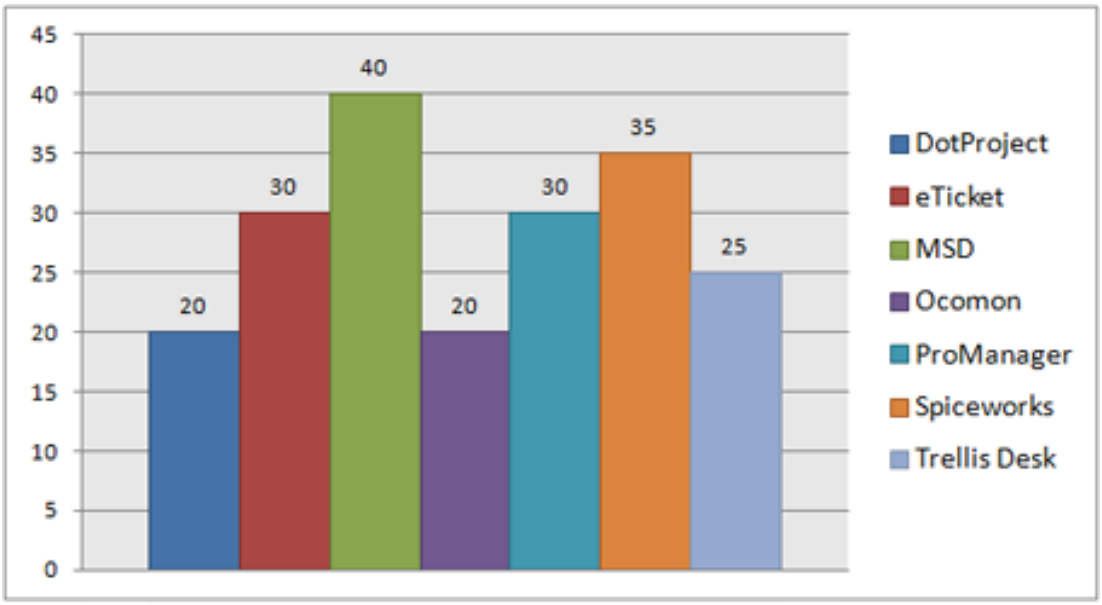

Figura 6.15 – Legibilidade: Qual a Escala de Satisfação em Relação à Legibilidade e Navegabilidade no Sistema

Analisando a escala de satisfação dos usuários em relação à legibilidade das informações, Figura 6.15, nota-se uma diferença significativa entre a que obteve a melhor pontuação (*Mobile Service Desk*, 40) e aquelas que obtiveram as piores pontuações (DotProject e

Ocomon, 20). A alta pontuação obtida pela ferramenta *Mobile Service Desk*, é justificada pelos avaliadores por sua boa legibilidade e navegabilidade, onde as caixas de textos e botões apresentam um tamanho maior, adequando-se melhor aos diferentes dispositivos utilizados nos testes. Já as principais razões apontadas para justificar as baixas pontuações das ferramentas DotProject e OcoMon são principalmente o tamanho pequeno das letras dos links e menus dificultando a leitura e realização de tarefas na ferramenta DotProject e a disposição do menu "em árvore"da ferramenta OcoMon, que acaba deixando apenas uma pequena parte da interface visível, forçando o usuário utilizar a barra de rolagem com frequência para poder visualizar os textos e menus.

A interface adaptada ao dispositivo móvel influencia diretamente na legibilidade do usuário, pois, com exceção das ferramentas Spiceworks e *Mobile Service Desk* que obtiveram as melhores avaliações de satisfação neste quesito, todas as demais ferramentas avaliadas não foram desenvolvidas para serem adaptáveis à dispositivos móveis. Desta forma, características destas ferramentas que obtiveram maior satisfação, tais como, botões, menus e textos maiores, facilitam a navegação entre as áreas do sistema e colaboram para a legibilidade, principalmente em dispositivos móveis com telas menores.

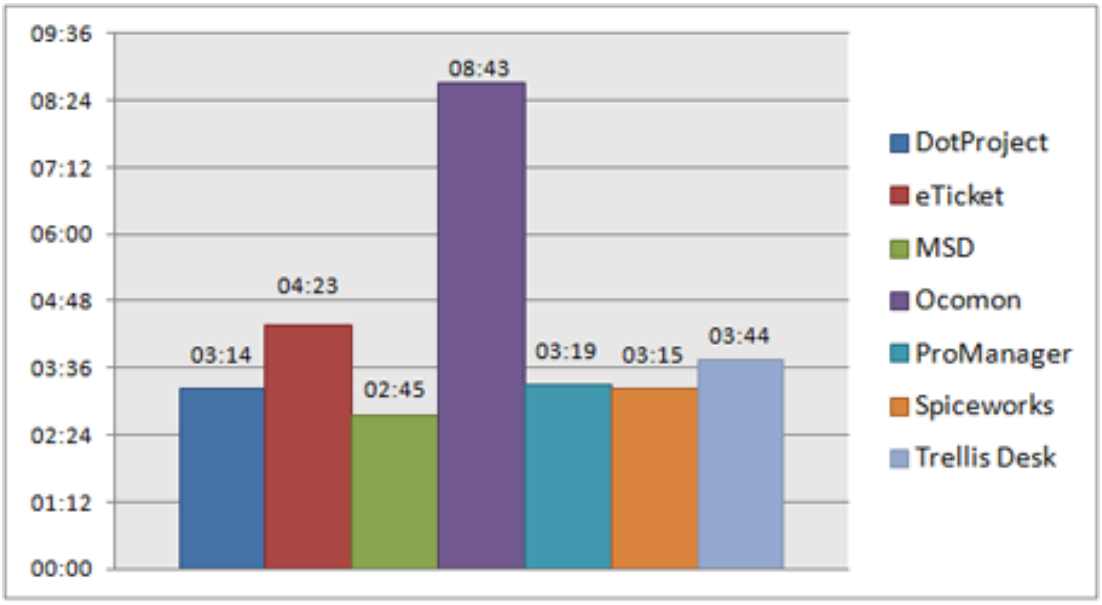

Figura 6.16 – Legibilidade: Média de Tempo Gasto para Realizar as Tarefas

A média do tempo gasto para a execução das tarefas, Figura 6.16, apresenta uma diferença considerável entre a ferramenta *Mobile Service Desk* (02m45s) que obteve a melhor média e a ferramenta OcoMon que apresentou o pior resultado (08m43s). Este resultado, aliado à avaliação de satisfação da usabilidade demonstram que uma interface objetiva e com um

tamanho de letra adequado, além de contribuir para uma melhor satisfação dos usuários, influencia diretamente no tempo despendido para realização das tarefas nas ferramentas.

6.3 Comparativo entre ferramentas

Foi realizado um comparativo com as principais ferramentas gratuitas disponíveis, de acordo com diversos requisitos, tais como: (i) funcionamento em dispositivos móveis; (ii) funcionalidades de sensibilidade ao contexto; (iii) se trata a questão de priorização de chamados; (iv) se possui um sistema de sugestão de soluções integrado a ferramenta. O resultado deste comparativo é apresentado na Tabela 6.18.

Tabela 6.18 – Comparativo entre ferramentas de *Service Desk*

	OcoMon	**Trellis Desk**	**Spiceworks**	**DotProject**	**eTicket**	**proManager**	**MSD**
Dispositivos Móveis	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Sensível ao Contexto	Sim	Sim	Não	Não	Sim	Não	Sim
Prioridade de Chamados	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Sugestão de Solução	Não	Não	Sim	Não	Não	Não	Sim

Todas as ferramentas funcionam em dispositivos móveis, visto que, com exceção da Spiceworks que possui um aplicativo específico para o sistema operacional iOS e para o Android, as demais são ferramentas *web* que apenas com o navegador do dispositivo móvel e conexão à *internet* é possível acessá-las. No entanto, conforme avaliação realizada com base na norma NBR 9241-11, apresentada na Seção 6.2, algumas destas ferramentas não são totalmente adaptadas para este tipo de dispositivos, o que acarreta em uma série de dificuldades na utilização.

Quanto a sensibilidade ao contexto, as ferramentas Trellis Desk, OcoMon e Eticket possibilitam que no momento da abertura do chamado o mesmo seja direcionado à uma área de atendimento específica. No entanto, como esta classificação é feita pelo mesmo usuário que abre o chamado, podem ocorrer casos onde essa classificação não seja atribuída de forma correta. A priorização de chamados na *Mobile Service Desk*, assim como o direcionamento de chamados para um nível de *expetise* específico, são realizados por um técnico com nível de classificador, já nas demais ferramentas o próprio usuário que abre o chamado define a prioridade da sua solicitação, o que pode ocasionar casos onde uma solicitação mais urgente esteja definida com menor prioridade que outra menos urgente.

Por fim, a sugestão de solução está presente em forma de busca em fóruns na ferramenta Spiceworks, no entanto um fórum é uma discussão ampla e não traz uma sugestão direta de

solução, já no sistema proposto nessa dissertação a sugestão é realizada através de um sistema de recomendação, com base em incidentes anteriores que sejam similares ao problema a ser solucionado, indicando uma sugestão direta para auxilar o técnico.

7 CONCLUSÕES E TRABALHOS FUTUROS

A função de um *Service Desk* é realizar o processo de gerenciamento de incidentes, lidando com todos os incidentes de uma organização relacionados a área de TI, tais como falhas de *software* e *hardware*, redes de comunicação ou qualquer dispositivo que ocasione o funcionamento anormal dos serviços de TI. O principal objetivo da gestão de incidentes é restaurar o funcionamento do serviço o mais rápido possível. Além disso o *Service Desk* provê aos usuários de TI um SPOC (*Single Point of Contact*), vital para uma comunicação efetiva entre os usuários e as equipes que gerenciam a TI em uma organização (OCG, 2007); isto auxilia os gestores, pois não é necessário visualizar inúmeras ferramentas para acessar informações relativas aos incidentes do ambiente de TI.

A utilização cada vez maior de dispositivos móveis e o avanço das tecnologias abrem novas possibilidades para expansão deste tipo de serviço, visando a melhoria da qualidade e rapidez nos atendimentos prestados pelos técnicos de suporte.

Desta forma, este trabalho teve como objetivo responder à seguinte questão de pesquisa: é possível otimizar o trabalho das equipes de *Service Desk*, fazendo uso das informações de contexto, a fim de reduzir o tempo com deslocamentos desnecessários e elaborar um sistema de recomendação que sugira possíveis soluções para problemas similares, a fim de auxiliar novos técnicos na resolução de incidentes, contribuindo para a disponibilidade da infraestrutura tecnológica e com a competitividade das organizações?

Os resultados obtidos por meio desta pesquisa e apresentados no Capítulo 6, apontam que a utilização de informações do contexto podem auxiliar para a racionalização de tempo de deslocamento entre prédios, principalmente se o processo não possui nenhum tipo de priorização entre chamados ou então se essa priorização é realizada pelo próprio usuário que solicita a resolução do incidente.

No que diz respeito aos algoritmos para análise de similaridade e pré-processamento de texto que compõem o sistema de recomendação proposto neste trabalho, é possível concluir que estes podem fazer parte de um sistema de *Service Desk* e que, na maioria das vezes, retornam resultados relativos ao problema em questão, conforme visto na Seção 6.1.4, que compara os algoritmos para análise de similaridade. No entanto, não é possível confiar totalmente neste resultado, visto que em alguns casos mais específicos ou para novos problemas (i.e, que não possuem casos parecidos no histórico de chamados já solucionados) o sistema trará soluções que

não são aplicáveis integralmente ao incidente, nestes casos será exigido um maior conhecimento e experiência do técnico para resolução do problema.

Quanto à avaliação de usabilidade dos sistemas de *Service Desk*, foi utilizada a norma brasileira NBR 9241-11 que foi criada a partir da ISO 9241-11. Através dela foi possível elaborar um questionário, que serve tanto para aplicações de *Service Desk* quanto para outras aplicações que necessitem medir usabilidade em termos de desempenho e satisfação do usuário, que foram avaliados pela extensão na qual os objetivos pretendidos de uso são alcançados, pelos recursos gastos para alcançar os objetivos pretendidos e pela extensão na qual o usuário considera aceitável o uso do produto. Através desta avaliação, foi possível concluir que os *softwares* disponíveis atualmente não contemplam de forma satisfatória as necessidades dos usuários, principalmente daqueles que acessam por meio de dispositivos móveis, com exceção do Spiceworks que obteve boa avaliação em relação aos demais. A ferramenta *Mobile Service Desk*, desenvolvida a partir da proposta deste trabalho, foi a que obteve o melhor resultado na avaliação baseada na NBR 9241-11.

Desta forma, as principais contribuições deste estudo são: (i) modelagem de um sistema de *Service Desk* denominado *Mobile Service Desk*, que agrega características de sensibilidade ao contexto à ferramenta, a fim de que estas informações contribuam com um melhor alocamento da equipe técnica na resolução de incidentes; (ii) um sistema de recomendação agregado a ferramenta que, fazendo uso de técnicas de pré-processamento de texto e análise de similaridade entre *strings*, sugere a técnicos possíveis soluções de problemas que ainda não foram solucionados, baseando-se em outros casos similares que já tenham sido resolvidos; (iii) avaliação deste sistema proposto e também de outros já existentes, através de uma norma que mede a usabilidade em termos de desempenho e satisfação do usuário.

Diante dos resultados encontrados, considera-se que a ordenação de chamados de uma central de serviços, utilizando geolocalização e a expertise dos técnicos, pode melhorar a eficácia do atendimento, reduzindo o tempo dos técnicos com deslocamentos desnecessários e o tempo de inatividade de equipamentos ou ferramentas de TI. Além disso, é levado em conta que a utilização de recomendação para possíveis soluções pode auxiliar os técnicos das equipes de atendimento, principalmente aqueles que ingressaram recentemente à equipe. Considera-se ainda que estas melhorias ajudem a reduzir o custo de determinação e resolução de incidentes que para SONG; SAILER; SHAIKH (2009) representam mais da metade dos custos operacionais de TI.

7.1 Trabalhos Futuros

Como trabalhos futuros é desejável que sejam testados outros algoritmos de análise de similaridade, a fim de avaliar se existe algum algoritmo que retorne um resultado melhor que os avaliados neste trabalho. Bem como, estudar e definir um percentual de confiança para o algoritmo, a fim de que ele ignore casos que não tenham muita similaridade com o problema, evitando que traga resultados com baixo percentual de similaridade e um técnico tente aplicar a solução indicada sem obter sucesso.

Além disso, é possível modificar o Sistema de Recomendação que, neste trabalho, utiliza um método baseado em conteúdo, em um sistema que utilize um método híbrido. Isso pode ser feito através da interação do técnico de suporte, que pode avaliar a qualidade de cada solução que lhe é apresentada e esta avaliação poderá influenciar em novas indicações daquela solução para outros problemas similares.

Outra possibilidade de trabalho futuro é que o sistema sugira possíveis soluções para problemas simples e recorrentes, sem que seja necessária a intervenção de um técnico. Apenas utilizando as soluções dos chamados similares e indicando-as para os usuários logo após abrirem o chamado, agindo como um sistema inteligente.

REFERÊNCIAS

ABNT. **NBR 9241-11**. 2002.

ACCENTURE. **Mobile Internet - Spawning New Growth Opportunities in the Convergence Era**. Disponivel em: <http://www.accenture.com/us-en/Pages/insight-mobile-web-watch-2012-mobile-internet.aspx >. Acesso em: Outubro 2012.

ACCORD5. **Trellis Desk**. Disponivel em: <http://www.accord5.com/trellis >. Acesso em: Maio 2012.

ADOMAVICIUS, G.; TUZHILIN, A. Toward the Next Generation of Recommender Systems: a survey of the state-of-the-art and possible extensions. **IEEE Trans. on Knowl. and Data Eng.**, Piscataway, NJ, USA, v.17, n.6, p.734–749, jun 2005.

ARANHA, C.; PASSOS, E. Tecnologia de Mineração de Textos. **RESI - Revista Eletrônica de Sistemas de Informação**, [S.l.], p.p.v. 2, p. 2, 2006.

ASTAH. **Astah Community**. Disponivel em: <http://www.astah.net/ >. Acesso em: Dezembro 2012.

AVILA, R. de; SOARES, J. Concepção de ferramenta de apoio à correção de questões dissertativas com base na adaptação de algoritmos de comparação e busca textual combinados com técnicas de pré-processamento de textos. **RENOTE - Revista Novas Tecnologias**, [S.l.], v.10, n.3, dezembro 2012.

BALABANOVIć, M.; SHOHAM, Y. Fab: content-based, collaborative recommendation. **Commun. ACM**, New York, NY, USA, v.40, n.3, p.66–72, mar 1997.

BENDERSKY, M.; CROFT, B. Finding text reuse on the web. In: SECOND ACM INTERNATIONAL CONFERENCE ON WEB SEARCH AND DATA MINING, New York, NY, USA. **Proceedings...** ACM, 2009. p.262–271. (WSDM '09).

BOYER, R. S.; MOORE, J. S. A fast string searching algorithm. **Commun. ACM**, New York, NY, USA, v.20, n.10, p.762–772, oct 1977.

CAVALARI, G.; COSTA, H. Modelagem e Desenvolvimento de um Sistema Help-Desk para a Prefeitura Municipal de Lavras. **RESI - Revista Eletrônica de Sistemas de Informação**, [S.l.], p.p.v. 2, p. 2, 2005.

COHEN, R. **Gestão de Help Desk e Service Desk**. [S.l.]: NOVATEC, 2011.

COPERVES. **Comissão Permanente do Vestibular**. Disponivel em: <http://www.coperves.ufsm.br/ >. Acesso em: Dezembro 2012.

CPD. **Centro de Processamento de Dados**. 2012.

DAVENPORT, T.; PRUSAK, L. **Conhecimento Empresarial**: como as organizações gerenciam seu capital. [S.l.]: Campus, 1998.

DBDESIGNER. **DbDesigner**. Disponivel em: <http://fabforce.net/dbdesigner4/ >. Acesso em: Dezembro 2012.

DEY, A. K. Understanding and Using Context. **Personal Ubiquitous Comput.**, London, UK, UK, v.5, n.1, p.4–7, jan 2001.

DOTPROJECT. **DotProject**. Disponivel em: <http://www.eticketsupport.com/ >. Acesso em: Maio 2012.

ETICKET. **eTicket Support**. Disponivel em: <http://www.eticketsupport.com/ >. Acesso em: Maio 2012.

GARTNER. **CIO Alert**: u.s. it staff turnover trends and analyses. Disponivel em: <http://www.gartner.com/id=771513 >. Acesso em: Maio 2012.

GARTNER. **Top 10 Strategic Technologies for 2012**. Disponivel em: <http://www.gartner.com/it/page.jsp?id=1826214 >. Acesso em: Julho 2012.

GNU. **Binary Files and Forcing Text Comparisons**. Disponivel em: <http://www.heidisql.com/ >. Acesso em: Outubro 2012.

GOOGLE. **Our Mobile Planet**. Disponivel em: <http://www.thinkwithgoogle.com/mobile planet/pt-br/ >. Acesso em: Outubro 2012.

HEIDISQL. **HeidiSQL**. Disponivel em: <http://www.heidisql.com/ >. Acesso em: Dezembro 2012.

ITIL. **Information Technology Infrastructure Library**. v3.

JANTTI, M.; KALLIOKOSKI, J. Identifying Knowledge Management Challenges in a Service Desk: a case study. In: SECOND INTERNATIONAL CONFERENCE ON INFORMATION, PROCESS, AND KNOWLEDGE MANAGEMENT, 2010., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2010. p.100–105. (EKNOW '10).

JARDIM, C. H. **Usando Serviços Web para integrar aplicações cientes de contexto**. 2006. Dissertação (Mestrado em Ciência da Computação) — Universidade de São Paulo - USP.

JARO, M. A. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. **Journal of the American Statistical Association**, [S.l.], v.84, n.406, p.414–420, 1989.

JOJABACODE. **Stop word list pt**. Disponivel em: <http://code.google.com/p/jojaba-code/ >. Acesso em: Outubro 2012.

JQUERYMOBILE. **Documentação jQuery Mobile**. Disponivel em: <http://jquerymobile.com/demos/1.2.0/docs/about/platforms.html >. Acesso em: Novembro 2012.

JUNIOR, J. R. **Desenvolvimento de uma Metodologia**. 2007. Dissertação (Mestrado em Ciência da Computação) — PUC-Rio.

KARP, R. M.; RABIN, M. O. Efficient randomized pattern-matching algorithms. **IBM J. Res. Dev.**, Riverton, NJ, USA, v.31, n.2, p.249–260, mar 1987.

KROVETZ, R. Viewing morphology as an inference process. In: ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 16., New York, NY, USA. **Proceedings...** ACM, 1993. p.191–202. (SIGIR '93).

LEVENSHTEIN, V. I. **Binary codes capable of correcting deletions, insertions, and reversals**. [S.l.: s.n.], 1966. (8).

LIKERT, R. A technique for the measurement of attitudes. **Archives of Psychology**, [S.l.], v.22, n.140, p.1–55, 1932.

LOBO, J. **Contexto de Expertise e Localização Influenciando a Gerência de TI**. 2011. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Santa Maria (UFSM).

MAGALHAES, I. L.; PINHEIRO, W. B. **Gerenciamento de Serviços de TI na Prática**: uma abordagem com base na itil. São Paulo: Novatec, 2007.

MOREIRA, M. et al. Internet do Futuro: um novo horizonte. In: SBRC 2009. **Anais...** [S.l.: s.n.], 2009.

MYSQL. **MySQL**. Disponivel em: <http://www.mysql.com/ >. Acesso em: Dezembro 2012.

NETBEANS. **NetBeans**. Disponivel em: <http://netbeans.org/ >. Acesso em: Dezembro 2012.

OCG. **Office of Government Commerce, ITIL Service Transition.** UK: The Stationary Office, 2007.

OCOMON. **OcoMon - Monitor de Ocorrências e Inventário de equipamentos de informática**. Disponivel em: <http://ocomonphp.sourceforge.net/ >. Acesso em: Maio 2012.

ORENGO, V.; HUYCK, C. A stemming algorithm for the portuguese language. In: STRING PROCESSING AND INFORMATION RETRIEVAL, 2001. SPIRE 2001. PROCEEDINGS.EIGHTH INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2001. p.186 – 193.

PAZ, S. M. **Uma ferramenta para desenvolvimento de equipamentos que utilizem um receptor**. 1997. Dissertação (Mestrado em Ciência da Computação) — Escola Politécnica da Universidade de São Paulo.

PEDROSO, H. S.; OLIVEIRA, T. B. de; MEDINA, R. D. **Implementação do Módulo de Retenção da Expertise Gerencial (MREG) Através de Trouble Ticket no Sistema Integrado de Monitoramento (S.I.M)**. 27ª JAI - Jornada Acadêmia Integrada.

PEJIC, B.; PEJIC, C.; COVIC, Z. Uses of W3C's Geolocation API. In: COMPUTATIONAL INTELLIGENCE AND INFORMATICS (CINTI), 2010 11TH INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2010. p.319 –322.

PHP. **PHP similar_text Manual**. Disponivel em: <http://php.net/manual/pt_BR/function.similar-text.php >. Acesso em: Agosto 2012.

SPARCK JONES, K.; WILLETT, P. (Ed.). **Readings in information retrieval**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997. p.313–316.

PORTER, M. F. **The Porter Stemming Algorithm**. Disponivel em: <http://snowball.tartarus.org/ >. Acesso em: Outubro 2012.

PROMANAGER. **ProManager**. Disponivel em: <http://www.eticketsupport.com/ >. Acesso em: Maio 2012.

PTSTEMMER. **PTStemmer - A Stemming toolkit for the Portuguese language**. Disponivel em: <http://code.google.com/p/ptstemmer/ >. Acesso em: Outubro 2012.

RANKSNL. **Portuguese Stopwords**. Disponivel em: <http://www.ranks.nl/stopwords/ portugese.html >. Acesso em: Outubro 2012.

REATEGUI, E. B.; CAZELLA, S. C. Sistemas de Recomendação. **XXV Congresso da Sociedade Brasileira da Computação - V ENIA**, [S.l.], v.17, p.306–348, 2005.

ROBUSTO, C. C. The cosine-haversine formula. **The American Mathematical Monthly**, [S.l.], v.64, n.1, p.38–40, 1957.

SAMARAKOON, L.; KUMARAWADU, S.; PULASINGHE, K. Automated question answering for customer helpdesk applications. In: INDUSTRIAL AND INFORMATION SYSTEMS (ICIIS), 2011 6TH IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2011. p.328 –333.

SIEWIOREK, D. New frontiers of application design. **Commun. ACM**, New York, NY, USA, v.45, n.12, p.79–82, dec 2002.

SILVA, M. S. **jQuery Mobile - Desenvolva aplicações web para dispositivos móveis com HTML5, CSS3, AJAX, jQuery e jQuery UI**. [S.l.]: Novatec, 2011.

SOARES, M. V. B.; PRATI, R.; MONARD, C. WCI 02 Improvements on the Porter's Stemming Algorithm for Portuguese. **Latin America Transactions, IEEE (Revista IEEE America Latina)**, [S.l.], v.7, n.4, p.472 –477, aug. 2009.

SONG, Y.; SAILER, A.; SHAIKH, H. Problem classification method to enhance the ITIL incident and problem. In: INTEGRATED NETWORK MANAGEMENT, 2009. IM '09. IFIP/IEEE INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2009. p.295 –298.

SPICEWORKS. **Spiceworks Free Network Management Software**. Disponivel em: <http://www.spiceworks.com/ >. Acesso em: Maio 2012.

UFSM. **Universidade Federal de Santa Maria**. Disponivel em: <http://ufsm.br/ >. Acesso em: Dezembro 2012.

VIERA, A. F. G.; VIRGIL, J. **Uma revisão dos algoritmos de radicalização em língua portuguesa**. Information Research, 12(3) paper 315. Disponivel em: <http://InformationR.net/ir/12-3/paper315.html >. Acesso em: Setembro 2012.

W3C. **HTML5 differences from HTML4**. Disponivel em: <http://www.w3.org/TR/html5-diff/ >. Acesso em: Novembro 2012.

W3CSCHOOLS. **HTML5 Geolocation**. Disponivel em: <http://www.w3schools.com/html/html5_ geolocation.asp >. Acesso em: Dezembro 2012.

WANG, D. et al. iHelp: an intelligent online helpdesk system. **Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on**, [S.l.], v.41, n.1, p.173 –182, feb. 2011.

WAZLAWICK, R. **Análise e Projeto de Sistemas de Informação Orientados a Objetos, 2E**. [S.l.]: Elsevier, 2010.

WEISER, M. The Computer for the 21st Century. **Scientific American**, [S.l.], v.265, n.3, p.94–104, September 1991.

WIKIMEDIA. **Stop word list pt**. Disponivel em: <http://wikimedia.org/ >. Acesso em: Outubro 2012.

WINKLER, W. E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In: SECTION ON SURVEY RESEARCH. **Proceedings...** [S.l.: s.n.], 1990. p.354–359.

APÊNDICES

APÊNDICE A – Questionário para avaliação seguindo a NBR 9241-11

Para cada objetivo (Facilidade de aprender; Usabilidade Global e Legibilidade) cada usuário deverá realizar as tarefas específicas para cada item, conforme descrito em cada seção.

Facilidade de aprender

Tarefas a serem realizadas:
Cadastrar usuário; Abrir um chamado;
Logar no Sistema; Atender um chamado (finalizar);

01.Foram aprendidas as tarefas? *
Cadastrar usuário; Logar no Sistema; Abrir um chamado; Atender um chamado (finalizar);

- Sim
- Parcialmente
- Não

01.Tempo gasto para aprender as tarefas: *
em minutos

01.Qual a escala de satisfação em relação à facilidade de aprender a utilizar o sistema? *

- Satisfeito
- Pouco Satisfeito
- Insatisfeito

Observações sobre a Facilidade de Utilização
Informar o motivo das avaliações, bem como possíveis problemas encontrados durante a execução dos testes.

Usabilidade Global

Tarefas a serem realizadas:
Cadastrar usuário; Abrir um chamado;
Logar no Sistema; Atender um chamado (finalizar);

02.As tarefas foram realizadas? *

- Sim
- Parcialmente
- Não

02.O tempo gasto para realizar as tarefas: *
em minutos

02.Qual a escala de satisfação em relação à usabilidade do sistema: *

- Satisfeito
- Pouco Satisfeito
- Insatisfeito

Observações sobre a Usabilidade do Sistema
Informar o motivo das avaliações, bem como possíveis problemas encontrados durante a execução dos testes.

Legibilidade

Tarefas a serem realizadas:
Logar no Sistema; Navegar entre os cadastros de usuários e chamados;
Ler texto dos chamados;

É possível ler textos e navegar entre as áreas do sistema? *

- Sim
- Parcialmente
- Não

Tempo gasto para realizar as tarefas: *
em minutos

Qual a escala de satisfação em relação à legibilidade das informações no sistema: *

- Satisfeito
- Pouco Satisfeito
- Insatisfeito

Observações sobre a Legibilidade
Informar o motivo das avaliações, bem como possíveis problemas encontrados durante a execução dos testes.

Submit

APÊNDICE B – *Stoplist* utilizada no *Mobile Service Desk*

Tabela B.1 – *Stoplist* do Mobile Service Desk

Palavras	Palavras	Palavras	Palavras	Palavras	Palavras	Palavras	Palavras
a	acerca	agora	ainda	alguem	algum	alguma	algumas
alguns	ali	ampla	amplas	amplo	amplos	ante	antes
ao	aos	apos	apontar	aquela	aquelas	aquele	aqueles
aqui	aquilo	as	ate	atras	atraves	bem	bom
cada	caminho	causar	cima	coisa	coisas	com	coma
como	comprido	conhecido	contra	contudo	corrente	da	daquele
daqueles	das	de	debaixo	dela	delas	dele	deles
dentro	depois	desde	desligado	dessa	dessas	desse	desses
desta	destas	deste	deste	destes	deve	devem	devendo
dever	devera	deverao	deveria	deveriam	devia	deviam	dia
dias	disse	disso	disto	dito	diz	dizem	dizendo
dizer	dizia	do	dos	e	ela	elas	ele
eles	em	enquanto	entao	entre	era	essa	essas
esse	esses	esta	estado	estamos	estao	estas	estava
estavam	estavamos	este	estes	estou	eu	falar	fazendo
fazer	feita	feitas	feito	feitos	ficar	foi	fomos
for	fora	foram	foramos	forem	formos	fosse	fossem
fui	grande	grandes	ha	haja	hajam	hajamos	haver
haveria	havido	hei	houve	houvemos	houver	houvera	houveram
houveramos	houverei	houverem	houveremos	houvesse	houvessem	houvessemos	ia
iamos	ida	ir	is	isso	isto	ja	la
lhe	lhes	lo	mas	mais	me	mesma	mesmas
mesmo	mesmos	meu	meus	minha	minhas	muita	muitas
muito	muitos	na	nao	nas	nem	nenhum	nessa
nessas	nesta	nestas	net	ninguem	no	nos	nossa
nossas	nosso	nossos	num	numa	nunca	o	os
ou	outra	outras	outro	outros	paga	pagando	pagar
page	pagou	para	parece	parecia	parecendo	pareceu	parecer
pareceram	parecem	pareceria	pela	pelas	pelo	pelos	pensar
pensam	pensando	pensado	pensaria	pensava	pense	pensei	pensou
pequena	pequenas	pequeno	pequenos	per	perante	perguntar	perguntando
perguntaram	perguntou	perguntei	pergunte	pessoa	pessoas	pessoal	pode
podendo	poder	poderia	poderiam	podia	podiam	pois	por
porem	porque	possa	posso	pouca	poucas	pouco	poucos
povo	primeiro	primeiros	propria	proprias	proprio	proprios	quais
qual	qualquer	quando	quanto	quantos	que	quem	quieto
sabe	sabem	sabera	sabemos	saber	saberia	sabia	sabiamos
sabido	sabendo	sao	se	sei	seja	sejam	sem
sempre	sendo	ser	sera	serao	serei	serem	seremos
seria	seriam	seriamos	seu	seus	si	sido	so
sob	sobre	sou	somente	somos	sua	suas	ta
tal	talvez	tambem	tampouco	te	tem	temos	tempo
tendo	tenha	tenham	tenhamos	tenho	tentar	tentaram	tente
tentei	ter	tera	terao	terei	teremos	teria	teriam
teriamos	tes	teu	teus	teve	ti	tido	tinha
tinham	tio	tios	tive	tivemos	tiver	tivera	tiveram
tiveramos	tiverem	tivermos	tivesse	tivessem	tivessemos	to	toda
todas	todavia	todo	todos	tu	tua	tuas	tudo
ultima	ultimas	ultimo	ultimos	um	uma	umas	uns
usa	usar	uso	usos	valor	veja	ver	verdade
verdadeiro	vendo	vez	vindo	vir	voce	vos	vos

APÊNDICE C – Especificações e Propriedades Desejáveis do Produto - NBR 9241-11

Para medir a usabilidade das ferramentas, seguindo as orientações da norma NBR 9241-11, foi preciso especificar os usuários que realizariam a avaliação, o ambiente e equipamentos necessários para que ela pudesse ser realizada, bem como as tarefas a serem executadas.

C.1 Especificação de usuários

Assume-se que os usuários estejam familiarizados com as características mais comuns de utilização de sistemas web. Alguns usuários potenciais terão que adquirir o conhecimento antes de usar com sucesso as ferramentas que serão avaliadas.

C.2 Especificação de ambientes

Para acessar as ferramentas é necessário um dispositivo com acesso à internet. Neste caso, serão testados com dispositivos móveis e computadores pessoais.

C.3 Especificação de equipamentos

Os equipamentos necessários para realizar as tarefas são dispositivos com acesso à internet. Além deste requisito, na ferramenta que trabalha com Geolocalização é desejável que o dispositivo possua GPS integrado.

C.4 Especificação de tarefas

Sistemas de Service Desk possuem diversas funcionalidades. Entretanto, tais sistemas pretendem ajudar o usuário a atingir certos objetivos particulares e estes estão listados aqui. Várias tarefas específicas são mencionadas onde a usabilidade do Service Desk é crítica, isto é: abertura e acompanhamento de chamados. Em um nível mais detalhado, a legibilidade da interface é mencionada como sendo uma condição para a usabilidade.

C.5 Propriedades Desejáveis do Produto

De acordo com a NBR 9241-11, usabilidade é a medida na qual um produto pode ser usado por usuários específicos, para alcançar objetivos específicos com eficácia, eficiência e satisfação, em um contexto específico de uso. Eficácia é acurácia e completude com as quais usuários alcançam objetivos específicos. A eficiência, são recursos gastos em relação à acurácia e abrangência com as quais usuários atingem objetivos. Satisfação é a ausência do desconforto e presença de atitudes positivas para com o uso de um produto. As propriedades desejáveis do produto, bem como as medidas de eficiência, eficácia e satisfação consideradas neste trabalho estão expressas na Figura C.1.

Objetivo	Eficácia	Eficiência	Satisfação
Usabilidade global	Porcentagem de objetivos alcançados; Porcentagem de usuários completando a tarefa com sucesso; Média da acurácia de tarefas completadas.	Tempo para completar uma tarefa; Tarefas completadas por unidade de tempo;	Escala de satisfação; Freqüência de uso; Freqüência de reclamações.
Facilidade de Aprender	Número de funções aprendidas; Porcentagem de usuários que conseguem aprender por critério.	Tempo para aprender por critério; Tempo para reaprender por critério; Eficiência relativa durante o aprendizado.	Escala para facilidade de aprendizado.
Legibilidade	Porcentagem de palavras lidas corretamente em ambiente com alta luminosidade.	Tempo para ler corretamente um número especificado de caracteres.	Escala para legibilidade e facilidade de navegação.

Figura C.1 – Medidas Para as Propriedades Desejáveis do Produto

APÊNDICE D – Dispositivos suportados pelo *framework* jQuery Mobile

De acordo com a documentação do jQuery Mobile, o *framework* tem suporte para a maioria dos *desktops* modernos, *smartphones*, *tablets* e plataformas *e-readers*. Além disso, suporta alguns telefones e navegadores mais antigos. Na documentação o suporte é dividido em 3 níveis: (i) Suporte Integral; (ii) Total, sem funcionalidade de Ajax; (iii) HTML básico. A seguir é apresentada a lista de dispositivos suportados em cada nível (JQUERYMOBILE, 2012).

D.1 Suporte Integral

- Apple iOS 3.2*-6.0; Android 2.1-2.3; Android 3.2 (Honeycomb); Android 4.0 (ICS); Android 4.1 (Jelly Bean); Windows Phone 7-7.5; Blackberry 6.0; Blackberry 7; Blackberry Playbook (1.0-2.0); Palm WebOS (1.4-2.0); Palm WebOS 3.0; Firefox Mobile 15; Chrome for Android 18; Skyfire 4.1; Opera Mobile 11.5-12; Meego 1.2; Tizen (pre-release); Samsung Bada 2.0; UC Browser; Kindle 3 and Fire; Nook Color 1.4.1; Chrome Desktop 11-21; Safari Desktop 4-5; Firefox Desktop 4-15; Internet Explorer 7-10; Opera Desktop 10-12.

D.2 Suporte sem Ajax

- Blackberry 5.0*; Opera Mini 7; Nokia Symbian3.

D.3 HTML Básico

- Blackberry 4.x; Windows Mobile.