

# Service Desk Móvel com Retenção de Conhecimento e Sensível ao Contexto

Taciano Balardin de Oliveira  
PPGI – UFSM  
tacianobalardin@gmail.com

Roseclea Duarte Medina  
PPGI – UFSM  
roseclea.medina@gmail.com

**Resumo**—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, bases de conhecimento, computação móvel podem incrementar estes aplicativos. Portanto, o objetivo deste trabalho é projetar e desenvolver um sistema de *Service Desk Mobile*, que agrega funcionalidades de sensibilidade ao contexto, tais como localização do usuário, experiência do técnico e tempo. Além disso, a ferramenta possui uma base de conhecimento para que interações passadas fiquem armazenadas e sejam sugeridas como possível solução para novos problemas.

## I. INTRODUÇÃO

O aumento crescente 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 [8]. Na ocorrência de problema em algum equipamento de informática presente nestes locais gerenciados (e.g. computador, impressora, *software* 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 [9].

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, de acordo com [7], foi inicialmente denominado de *Help Desk*. Entretanto, em consonância com [9][12], 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.

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. 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 [5], 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 [18].

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 e prática) 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 [8][9].

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 [6], 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 apontam que existe uma tendência de que, até 2013, os celulares ultrapassem os computadores pessoais como dispositivos mais comuns para acesso à *web*.

Uma possível solução para o problema de rotatividade de pessoal é a agregação de uma base de conhecimento em um sistema de *Service Desk*, pois, dessa forma, o conhecimento empregado pelos técnicos em situações anteriores pode ficar retido no sistema e ser apresentado como possível solução para problemas com características semelhantes. 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.

Este trabalho é continuidade de pesquisas realizadas pelo GRECA (Grupo de Redes de Computadores e Computação Aplicada) da UFSM (Universidade Federal de Santa Maria) e consiste em parte de uma Dissertação para obtenção do título de Mestre em Computação. Tem como principal objetivo projetar e desenvolver uma ferramenta de *Service Desk* móvel, sensível ao contexto e que possua detecção de expertise necessária para alocação do técnico que atenderá os chamados de suporte. Além disso, munir esta ferramenta com uma base de conhecimento, para que o capital intelectual, gerado através dos atendimentos realizados pela equipe de suporte, fique retido no sistema e seja apresentado como possível solução para novos chamados.

O artigo está organizado da seguinte forma, na seção II é apresentada a revisão bibliográfica, enquanto a seção III aborda o desenvolvimento do aplicativo e suas características. Na seção IV constam os resultados obtidos até o momento e, por fim, a seção V traz as conclusões parciais do trabalho.

## II. REVISÃO BIBLIOGRÁFICA

### A. 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 ponto único de contato vital para uma comunicação efetiva entre os usuários e as equipes que gerenciam a TI em uma organização. 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 [15].

O funcionamento de um sistema de *Service Desk* se dá 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*).

### B. Computação Ubíqua e Ciente ao Contexto

O termo Computação Ubíqua foi definido pelo cientista Mark Weiser [14] como um novo paradigma onde a computação deve estar invisível para o usuário. 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 [9][14].

A computação ciente de contexto é uma área de pesquisa da Computação Ubíqua, que tem o objetivo de prover o acesso de sistemas computacionais às informações de contexto, ampliando a comunicação entre ser humano e sistemas computacionais, assim permitindo o desenvolvimento de serviços computacionais mais úteis e adaptáveis [1].

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 [1][2]. 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 [2][4].

### C. Base de Conhecimento

Uma Base de Conhecimento (BC) é um tipo especial de banco de dados para gestão do conhecimento, uma BC fornece um meio das informações serem coletadas, organizadas, compartilhadas, conhecidas e utilizadas.

Um dos objetivos da implantação de uma arquitetura baseada em conhecimento utilizada como suporte nos ambientes de gerenciamento de TI é encurtar a curva de aprendizado de um membro ingressante na equipe. Esta aceleração da absorção do conhecimento permitirá que o colaborador ingressante assuma de forma plena as tarefas do antigo, minimizando assim o impacto da rotatividade, bem como utilizar o conhecimento aplicado a resolução

de um problema para auxiliar na resolução de outro que tenha características semelhantes [18].

Assim, muitas empresas tentam criar sistemas inteligentes de suporte técnico para melhorar a qualidade de serviço ao cliente. A partir de um novo chamado do usuário, o objetivo de um sistema de *Service Desk* integrado a uma base de conhecimento é descobrir se chamados semelhantes foram processados antes. 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) [18].

### D. Mineração de Texto e Similaridade entre Strings

Na literatura existem diversas técnicas para calcular similaridade entre *strings*, tais como a remoção de *stopwords*, *stemming*, modelo de índice invertido, algoritmo *Levenshtein Distance*, processamento de linguagem natural.

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 [10].

2) *Stemming*: Seu objetivo é a redução de cada palavra até que seja obtida sua raiz 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” [17]. Segundo pesquisa apresentada em [16], a utilização de *stemming* melhorou em 35% a recuperação de informações em alguns conjuntos de dados.

3) *Algoritmo Levenshtein Distance*: 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*) [10].

Outras informações sobre métodos para mineração de texto e cálculos de similaridade entre *strings* podem ser encontradas em [3], [10], [11] e [17].

## III. SERVICE DESK MÓVEL COM RETENÇÃO DO CONHECIMENTO E SENSÍVEL AO CONTEXTO

Desenvolvido na linguagem de programação PHP, que é responsável por gerar o conteúdo dinâmico do sistema e

executado em uma arquitetura cliente-servidor. 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.

O sistema possui uma interface de usuário unificada, ou seja, aplicação única para todos os dispositivos móveis suportados e sistemas operacionais. Para proporcionar essa interface unificada, é 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 [13].

O HTML 5 também é 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 *Service Desk*.

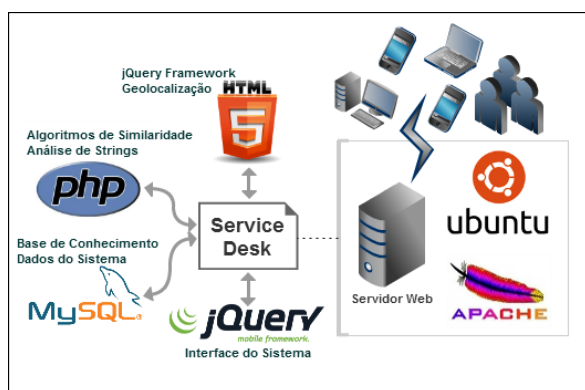


Figura 1. Arquitetura do Sistema de *Service Desk*.

A Figura 1 apresenta a arquitetura do *Service Desk*, dando uma ideia geral sobre as tecnologias empregadas em seu desenvolvimento e a forma como os usuários acessam o ambiente.

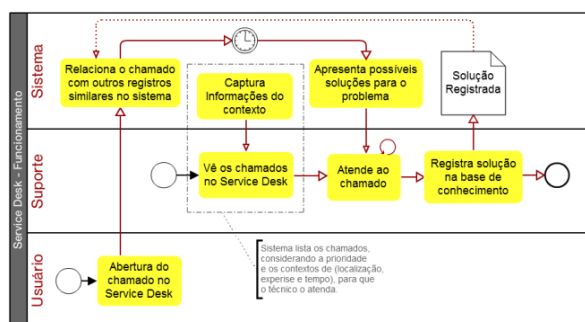


Figura 2. Funcionamento do Sistema de *Service Desk*.

Por sua vez, a Figura 2 mostra 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 são utilizados algoritmos de similaridade e também mineração de texto.

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 na base de conhecimento do sistema, para que a mesma possa servir como solução para outros problemas.

Não apresentado no esquema da Figura 2, mas de vital importância para o funcionamento do sistema, está o Gestor, que 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.

#### IV. RESULTADOS PARCIAIS

A aplicação está instalada em um servidor com o sistema operacional Ubuntu, na versão 12.04 LTS, utilizando arquitetura de 32 bits. Neste está 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.

Como o *Service Desk* é executado através da internet e, atualmente, encontra-se em fase de desenvolvimento e testes, o escopo de abrangência foi definido para o acesso da rede interna da UFSM e externamente através de uma rede privada virtual, utilizando o aplicativo OpenVPN.



Figura 3. Tela de Login.

Figura 4. Abertura de Chamado.

As capturas de telas apresentadas na Figura 3 e Figura 4 foram realizadas a partir de um *smartphone* com sistema operacional Android 2.3.

Na tela apresentada pela Figura 3, o usuário informa seus dados de acesso para entrar no sistema. Caso ainda

não seja cadastrado, tem a possibilidade de cadastrar-se, a partir disso, assim que o Gestor confirmar seu registro, também poderá acessar o ambiente.

A Figura 4 mostra parte do formulário para registrar um novo chamado no *Service Desk*. O campo “Local” é identificado automaticamente de acordo com a posição geográfica do usuário, desde que o dispositivo disponha de tal funcionalidade e o utilizador autorize o acesso do aplicativo a essa informação. Ainda assim, mesmo que identifique a localização, é possível informar outro local onde o problema ocorreu. O campo “Contato” é o nome do responsável pelo chamado, já o campo “Descrição do Problema” serve para escrever sobre o incidente a ser resolvido pela equipe de suporte. Por fim, o campo “Categoria” categoriza os problemas de acordo com algumas características, como por exemplo, se o problema é em um equipamento ou então em um *software*.

Até o momento, o sistema foi executado nos dispositivos apresentados na Tabela 1, que mostra o nome do dispositivo, o Sistema Operacional que o mesmo executa e a aplicação em que o sistema foi testado (S.O / App). Além disso, informa se a interface (Int.) carregou corretamente e também se as funcionalidades (Func.) da ferramenta foram executadas de forma satisfatória, ou seja, sem a ocorrência de erros ou algum problema que impeça a conclusão de alguma atividade no *Service Desk*.

Tabela 1  
DISPOSITIVOS TESTADOS

Dispositivo	S.O / App	Int.	Func.
Notebook	Ubuntu - Firefox	Ok	Ok
Notebook	Win7 - Chrome	Ok	Ok
Notebook	Win7 - IE	Ok	Ok
Smartphone	Android 2.3 - Browser Nativo	Ok	Ok
Smartphone	iOs3 - Browser Nativo	Ok	Ok
Smartphone	Symbian - Opera Mini	Ok	Ok
Tablet	Android 2.2 - Browser Nativo	Ok	Ok

De acordo com a Tabela 1, em todos os dispositivos testados a aplicação se comportou de forma correta, executando as funcionalidades e a interface do *Service Desk* de forma plena.

## V. CONCLUSÃO

Sistemas de *Service Desk* centralizam os problemas que acontecem em ambientes gerenciados por equipes de TI. A utilização cada vez maior de dispositivos móveis e o avanço das tecnologias abrem novas possibilidades para expansão deste tipo de serviço, sempre visando à melhoria da qualidade e rapidez nos atendimentos prestados pelos técnicos de suporte.

Até o momento, de acordo com os testes apresentados na Tabela 1, o *framework* de desenvolvimento utilizado tem suportado todos os dispositivos testados, além disso, não é necessário criar um aplicativo para cada sistema operacional, seja ele de um dispositivo móvel ou não, todos executam a mesma aplicação.

De acordo com os estudos realizados, as técnicas de mineração de textos apresentam-se como uma possível solução para a identificação de chamados na base de conhecimento, a fim de avaliar a similaridade entre estes e o chamado a ser atendido pela equipe de suporte. Também podem solucionar o problema de identificar o contexto de

expertise necessária para atender certo chamado, baseando-se em experiências anteriores.

Além disso, obter a localização, tanto do técnico quanto do problema a ser solucionado, pode tornar o processo de resolução mais rápido, pois prioriza os técnicos mais próximos do incidente para atendê-lo, desde que estejam em horário de expediente e possuam a expertise necessária para realizar a tarefa.

Como trabalhos futuros, os autores pretendem que o sistema sugira possíveis soluções para problemas simples e recorrentes, sem que seja necessária a intervenção de um técnico, bem como um estudo para melhoria no algoritmo de análise de similaridade entre chamados. Também é possível tornar a ferramenta escalável, para que funcione em um ambiente de *cloud computing*.

## REFERÊNCIAS

- [1] A. Dey, “Understanding and Using Context. Personal and Ubiquitous Computing” In Journal Personal and Ubiquitous Computing 5(1):4-7, 2001.
- [2] C. Jardim “Usando Serviços Web para integrar aplicações cientes de contexto”, Dissertação (Ciências da Computação e Matemática Computacional) – Universidade de São Paulo, São Paulo, SP, 2006.
- [3] C. N. Aranha and E. P. L. Passos “A Tecnologia de Mineração de Textos” RESL Revista Eletrônica de Sistemas de Informação, v. 2, p. 2, 2006.
- [4] D. Siewiorek “New frontiers of application design” In: Communications of the ACM, 45(12):79-82, 2002.
- [5] Gartner “CIO Alert: U.S. IT Staff Turnover Trends and Analyses” In: <http://www.gartner.com/id=771513>, 2011.
- [6] Gartner “Top 10 Strategic Technologies for 2012 - Analysts Examine Latest Industry Trends During Gartner Symposium” In <http://www.gartner.com/it/page.jsp?id=1826214>, 2011.
- [7] G. Cavallari, H. Costa “Modelagem e Desenvolvimento de um Sistema Help-Desk para a Prefeitura Municipal de Lavras - MG” In: Revista Eletrônica de Sistemas de Informação, 2005.
- [8] I. Magalhães, W. Pinheiro “Gerenciamento de Serviços de TI na Prática: Uma abordagem com base na ITIL”. São Paulo: Novatec, 2007.
- [9] J. Lobo “Contexto de Expertise e Localização Influenciando a Gerência de TI”, Dissertação (Mestrado em Computação) - Universidade Federal de Santa Maria, Santa Maria, RS, 2011.
- [10] J. R. C. Junior “Desenvolvimento de uma Metodologia para Mineração de Textos” Dissertação (Mestrado em Engenharia Elétrica), Rio de Janeiro/RJ PUC-Rio, 2007.
- [11] M. Bendersky and W. B. Croft “Finding Text Reuse on the Web” In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, Barcelona, 2009, pp. 262-271.
- [12] M. Jäntti, J. Kalliokoski “Identifying Knowledge Management Challenges in a Service Desk: A Case Study” In: eKNOW '10 - Second International Conference on Information, Process, and Knowledge Management, p. 100-105, 2010.
- [13] M. S. Silva “jQuery Mobile - Desenvolva aplicações web para dispositivos móveis com HTML5, CSS3, AJAX, jQuery e jQuery UI” Novatec, 2011.
- [14] M. Weiser “The Computer for the Twenty-First Century”. In: Scientific American, pages 94-10, 1991.
- [15] R. Cohen “Gestão de Help Desk e Service Desk” 1 ed. São Paulo: Novatec, 2011. 296 p. ISBN 978-85-7522-276-8.
- [16] R. Krovetz “Viewing morphology as an inference process” In: ACM SIGIR Conference on Research and Development in Information Retrieval, 1993, pp. 191-202.
- [17] V.N. Orengo “A stemming algorithm for the portuguese language” In: String Processing and Information Retrieval, 2001, pp. 186-193
- [18] W. Dingding, T. Li, S. Zhu, Y. Gong “iHelp: An Intelligent Online Helpdesk System”. In: IEEE transactions on systems, man, and cybernetics, vol. 41, issue 1, p. 173-182, 2011.