

# Modelagem e Desenvolvimento de um Sistema *Help-Desk* Para a Prefeitura Municipal de Lavras – MG

GABRIEL OTÁVIO TREVISANI CAVALARI  
HEITOR AUGUSTUS XAVIER COSTA

UFLA – Universidade Federal de Lavras  
DCC – Departamento de Ciência da Computação  
Cx. Postal 37 – CEP 37.200-000 Lavras (MG)  
{cavalari, heitor}@comp.ufla.br

**Resumo:** Sistemas *Help-Desk* podem ser definidos como sendo um setor da empresa ao qual são endereçadas questões e onde são resolvidos problemas, tendo como principal característica a de ser um sistema facilitador de informações. A proposta deste trabalho é realizar a modelagem e o desenvolvimento de um sistema *Help-Desk* para a Prefeitura Municipal de Lavras, para tal são discutidas a arquitetura e as tecnologias deste tipo de sistema, tal como Gestão do Conhecimento, Raciocínio Baseado em Casos e Baseado em Regras.

**Palavras Chaves:** Sistemas *Help-Desk*, Gestão do Conhecimento, Raciocínio Baseado em Casos, Raciocínio Baseado em Regras.

## 1. Introdução

Sistemas *Help-Desk* podem ser definidos como sendo um setor da empresa ao qual são endereçadas questões e onde são resolvidos problemas. Entretanto a concepção de *Help-Desk* vai muito além do conceito de suporte técnico, tendo hoje como principal característica o fato de ser um sistema facilitador de informações ao usuário, não importando se esta facilidade é ou não de natureza técnica computacional.

Um sistema *Help-Desk* é composto por três componentes básicos: i) *software* (ferramenta); ii) equipe; e iii) metodologia de serviço [G&P, 2004].

O termo *Help-Desk* pode ser utilizado tanto para o sistema em si quanto para a ferramenta de *software* utilizada por este [G&P, 2004].

O *software* controla o inventário tecnológico da empresa, revelando a repetição de problemas, os tempos médios de atendimento e as soluções para as áreas mais demandantes.

A equipe atua em dois níveis: o campo e a retaguarda. A equipe de campo faz o atendimento aos problemas, solucionando-os quando possível. A equipe de retaguarda é acionada quando um problema supera a capacidade de resolução da equipe de campo [G&P, 2004].

A metodologia de serviço contempla o posicionamento do sistema *Help-Desk* junto aos usuários, definindo quais são as estratégias de ação diante de determinado problema [G&P, 2004].

Um sistema *Help-Desk* é utilizado para melhorar o gerenciamento das soluções de atendimento. Domínios atraentes para *Help-Desk* são relativos a: i) suporte à informática; ii) sac - serviço de atendimento ao consumidor (interno/externo); iii) controle de serviços/manutenção; e iv) centro de informações.

### 1.1. Motivação

A Prefeitura Municipal de Lavras (PML) ciente da importância que a informática adquiriu nos últimos anos tem se esforçado para modernizar-se, tendo iniciado o seu processo de informatização recentemente. Porém, como em todo processo de mudança, a prefeitura tem enfrentado problemas.

Há diversos motivos que levaram a PML a estudar a implantação de um sistema *Help-Desk*, a saber: i) chamados, reclamações e sugestões não são anotados e quando são não há controle sobre esses processos; ii) o sistema de atendimento utilizado não atende todas as necessidades; iii) não existe documentação das soluções de problemas já resolvidos anteriormente; e iv) necessidade de diminuir os custos operacionais.

### 1.2. Objetivos

O presente trabalho tem como objetivos a modelagem do sistema utilizando a UML (Unified Modeling Language) [OMG's UML, 2004], a modelagem de dados utilizando o modelo de entidade relacionamento com notação UML [Gornik, 2004] e o desenvolvimento de produto de *software Help-Desk* para a PML. Esse *software* visa melhorar o suporte aos funcionários, que

são os principais usuários da infra-estrutura tecnológica da prefeitura.

### 1.3. Metodologia do Desenvolvimento

O presente trabalho dividiu-se em duas fases. A primeira diz respeito à busca de informações e à análise das principais tecnologias e ferramentas de software empregadas em sistemas *Help-Desk*.

A segunda etapa do trabalho consistiu na modelagem e no desenvolvimento de um produto de software para o *Help-Desk* da PML.

Durante a fase de implementação do sistema ocorreu a implementação do produto de software *Help-Desk* PML.

A implementação da ferramenta *Help-Desk* foi dividida em três partes: banco de dados, interface do usuário e interface do suporte.

O banco de dados é responsável por armazenar todas as informações relevantes do sistema. Essa parte da ferramenta foi implementada utilizando o sistema gerenciador de banco de dados (SGBD) MySQL.

A interface do usuário é o meio pelo qual o funcionário da prefeitura interage com o sistema. É por essa interface que o usuário realiza as solicitações de suporte ao sistema e verifica o status de uma solicitação aberta. Essa parte da ferramenta foi implementada utilizando a tecnologia PHP (*Personal Home Page Tools*).

A interface do suporte, é por onde a equipe interage com o sistema, essa interface do sistema também foi desenvolvida utilizando a linguagem PHP.

No desenvolvimento da ferramenta de software, foram utilizadas técnicas de engenharia de software objetivando conduzir o projeto dentro de um cronograma viável e adaptável. Para isso, foi utilizado a técnica de prototipação [Pressman, 2002]. Alguns motivos que levaram a sua escolha foram [Pressman, 2002]: i) possibilita que o cliente possa acompanhar o desenvolvimento do software; ii) mudanças durante a fase de implementação do projeto, não prejudicam a sua construção; e iii) o software a ser desenvolvido é simples, não necessitando de fazer análise de riscos [Pressman, 2002].

## 2. Ferramentas *Help-Desk*

Pode-se dizer que a fórmula para o sucesso de um sistema *Help-Desk*, consiste em encontrar o equilíbrio ideal entre equipe, metodologia e ferramenta de *software*.

O uso da ferramenta certa pode melhorar as operações de apoio ao usuário através da redução de

custos na manipulação de incidentes, melhora no fluxo de trabalho, na divisão do conhecimento e na colaboração entre funcionários.

Para uma melhor compreensão das ferramentas *Help-Desk* é necessário conhecer alguns conceitos envolvidos na utilização destes sistemas. Entre eles, pode-se destacar [Automidia, 2004]:

- **Help-Desk Interno:** São definidos como sendo um departamento da empresa dedicado a realizar suporte técnico de seus próprios funcionários;
- **Help-Desk Externo:** Também chamados serviço de atendimento ao cliente, são definidos como um grupo ao qual são endereçadas questões e onde são resolvidos problemas, sendo seu objetivo final manter os usuários produtivos e satisfeitos;
- **Chamados:** Chamados representam as solicitações dos usuários para os quais o *Help-Desk* oferece atendimento, pode representar um problema na execução de uma determinada tarefa ou uma solicitação de serviço, por exemplo;
- **Cliente:** Usuário que contata os analistas para a abertura de um chamado;
- **Analistas:** São os responsáveis pelo atendimento a um chamado;
- **Eventos:** Eventos são rotinas associadas aos chamados;
- **Tempo de Resposta:** É o intervalo de tempo decorrido entre a notificação de um problema (abertura do chamado) e o primeiro contato do *Help-Desk* com o usuário;
- **Tempo de Resolução:** É o tempo necessário para a execução dos serviços de atendimento;
- **SLA (*Service Level Agreement*):** SLAs representam classes de usuários, equipamentos e situações que devem ser considerados de uma maneira diferenciada.

Atualmente existem no mercado várias empresas que desenvolvem ferramentas *Help-Desk* comerciais. As ferramentas variam das mais simples que oferecem apenas funções de administração e rastreamento de chamados até as mais complexas que possuem suporte a gestão do conhecimento, acesso remoto, além de recursos de CRM (*Customer Relationship Management*).

Para uma melhor compreensão das características e do funcionamento desse tipo de *software*, foram analisadas três ferramentas *Help-Desk*: *Footprints*, *Fireman* e *Qualitor Help*.

Todas as informações utilizadas na análise das ferramentas foram obtidas no site dos respectivos desenvolvedores.

As ferramentas *Help-Desk* analisadas são semelhantes no que se refere às funções básicas, como rastreamento de chamados, gestão de inventário, uso de bases de conhecimento e geração de relatórios. A principal diferença entre as três ferramentas reside na arquitetura e nas funções auxiliares que refletem o enfoque dado pelo desenvolvedor.

Enquanto a Unipress, desenvolvedora do *FootPrints*, criou uma ferramenta mais completa capaz de gerenciar grande variedade de processos dentro de uma organização, a Constat, *Qualitor Help*, se preocupou em criar um sistema que privilegiasse o atendimento ao cliente, dando grande destaque ao gerenciamento de SLAs. O *Fireman Help-Desk*, desenvolvido pela Sial, é mais voltado para o uso em sistemas *Help-Desk* internos, não apresentando assim muitas ferramentas de apoio.

### 3. Tecnologias Utilizadas

Essa seção pretende mostrar a integração dos sistemas *Help-Desk* ao contexto da área de Gestão do Conhecimento. Será apresentado, também, o conceito de Raciocínio Baseado em Regras, o primeiro modelo a ser utilizado na construção de sistemas *Help-Desk* Inteligentes. Em seguida, será mostrada a evolução desse conceito, o Raciocínio Baseado em Casos (RBC). E, por último, será apresentado um modelo de arquitetura para sistemas *Web Help-Desk* utilizando Raciocínio Baseado em Casos e Regras.

#### 3.1 Gestão do Conhecimento

Sistemas *Help-Desk* necessitam de uma integração contextual para que o seu papel possa ser devidamente valorizado. A necessidade de se relacionar sistemas *Help-Desk* à Gestão do Conhecimento decorre natureza aglutinadora de informações de ambos.

Para entender o porquê dessa integração, é necessário que se defina Gestão do Conhecimento e para tal é preciso antes compreender o que é conhecimento do ponto de vista computacional.

Conhecimento pode ser definido como a combinação de experiência, valores, informação contextual e intuições significativas que propiciam uma abordagem para novas informações e experiências [Coelho et al., 2004].

Nas organizações, o conhecimento pode tomar forma, não apenas de repositório, mas também de rotinas organizacionais, processos, práticas e normas.

Este conhecimento pode ser classificado em dois tipos: explícito ou tácito [Coelho et al., 2004].

O conhecimento explícito é aquele que pode ser expresso em uma linguagem formal, em afirmações gramaticais, expressões matemáticas, especificações e manuais [Moresi, 2001].

O conhecimento tácito por sua vez é aquele que está relacionado a habilidades e competências pessoais e de difícil especificação.

A Gestão do Conhecimento pode ser vista como a formalização de conhecimentos para que estes se tornem acessíveis à organização, contribuindo para a melhoria do desempenho individual ou organizacional [Junior & Shimabukuro, 2003].

O sistema de Gestão do Conhecimento é uma ferramenta administrativa para representar, armazenar, compartilhar e distribuir conhecimentos com o objetivo de que a informação seja levada à pessoa certa no tempo certo.

A Gestão do Conhecimento significa adotar uma diversidade de fontes do conhecimento com a ajuda de um banco de dados corporativo e ambientes baseados em troca de informações via Intranet e Internet, onde funcionários e demais colaboradores poderão utilizar os recursos para descrever os modelos mentais que utilizam na resolução de problemas [Junior & Shimabukuro, 2003].

Antes de correlacionar *Help-Desk* como uma tecnologia de Gestão do Conhecimento, é importante enfatizar que sistemas facilitadores de informação podem ser modelados com o emprego de metodologias e ferramentas as mais variadas. Têm-se interesse, particularmente, os sistemas construídos com base no conhecimento sobre Problemas e sobre as suas Soluções.

Esta representação computacional sobre Problemas e Soluções dá origem a um importante repositório denominado base de conhecimento.

Base de conhecimento pode ser definida como sendo uma estrutura organizada de informação que facilita o armazenamento da inteligência do sistema com a finalidade de ser resgatada ou recuperada em apoio a uma demanda feita ao sistema *Help-Desk* [Coelho et al., 2004].

O papel do sistema *Help-Desk* como um processo automático da Gestão do Conhecimento, no interior de uma organização, fica deste modo evidenciado. Pois, ambos: i) lidam com processos envolvendo conhecimento; ii) requerem a criação de bases de conhecimento nas organizações; iii) dedicam-se à inteligência e à aprendizagem das organizações; e iv)

estão totalmente a serviço da moderna economia baseada em conhecimento [Coelho et al., 2004].

### 3.2 Raciocínio Baseado Regras

As funções de um sistema *Help-Desk*, fundamentalmente, envolvem tarefas de classificação.

Ao realizar classificações, os sistemas *Help-Desk* respondem indagações que a todo o momento podem vir a ser colocadas pelos usuários destes sistemas. Nos primeiros sistemas *Help-Desk*, todas as indagações eram respondidas pela equipe, nos sistemas atuais, a própria ferramenta de *software* é capaz de responder às dúvidas dos usuários. Os sistemas *Help-Desk* dotados dessa característica são chamados de *Help-Desk* inteligente [Coelho et al., 2004].

A construção de um sistema *Help-Desk* inteligente consiste em captar o conhecimento de um analista, representar este conhecimento em uma base e transmiti-lo ao usuário, permitindo-lhe obter respostas a perguntas relacionadas à base de conhecimento do sistema [Hall & Kandel, 1986].

O uso de regras de produção é uma das maneiras mais utilizadas para representação do conhecimento. Em sistemas baseados em regras, os conhecimentos são representados através de pares condição-ação, onde as regras possuem duas partes: uma antecedente (“*IF*”) e outra conseqüente (“*THEN*”), a esse tipo de regra dá-se o nome de *IF-THEN* [DCC/UA, 2004].

A capacidade para definir as regras no formato *IF-THEN* tem diversas vantagens [DCC/UA, 2004]: i) as regras são entendidas pelos programadores e pelos peritos de forma idêntica; ii) as regras podem conter “pedaços” de conhecimento, que coletivamente podem modelar um problema bastante complexo; iii) as regras são independentes entre si; e iv) a estrutura de controle é relativamente simples.

Nos sistemas baseados em regras, o conhecimento é representado por: fatos, relações entre fatos e regras para a manipulação desses fatos.

Os sistemas baseados em regras têm limitações significativas. Na maioria das situações, é extremamente difícil obter um conjunto correto de regras, isto ocorre pois, a abordagem baseada em regras assume que existe um corpo de conhecimento que a maioria dos especialistas na área usa e aprova. No entanto, em muitas áreas não existem modelos nem princípios gerais aceitos pela maioria dos especialistas. Deste modo torna-se não trivial o desenvolvimento de sistemas baseados em regras surgindo então o raciocínio baseado em casos [Simões, 1999].

### 3.3 Raciocínio Baseado Em Casos

Os sistemas *Help-Desk* centrados no RBC oferecem respostas às perguntas procurando em uma base de casos (devidamente indexada), casos passados que se aplicam ao problema atual. Uma indexação coerente e o modo como os casos são representados facilitam sua recuperação correta.

É possível simplificar o processo mental de um sistema RBC como sendo cíclico e composto por quatro “R” [Aamodt, 1994] (figura 1): i) recuperar o(s) caso(s) mais similar(es); ii) reusar o(s) caso(s) para resolver o problema; iii) revisar a solução proposta, se necessário; e iv) reter a nova solução incluindo-a na base de casos.

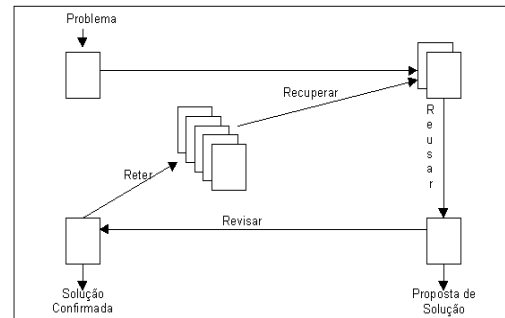


Figura 1: O Ciclo RBC [Moresi, 2001]

A principal parte do conhecimento nos sistemas RBC é representada através de seus casos. Um caso pode ser entendido como a abstração de uma experiência descrita em termos de seu conteúdo e contexto. Todo caso é composto por um problema e uma solução. É importante salientar que um caso não é uma regra. A representação dos casos é uma tarefa complexa e importante para o sucesso do sistema RBC [Coelho et al., 2004].

Na representação dos casos, cada característica tem uma certa importância. Baseado nisso, é feita a indexação dos casos, ou seja, associa-se rótulos de forma a caracterizá-los, para posteriormente recuperá-los na base de casos. A indexação determina o que comparar entre os casos para determinar a sua similaridade. Bons índices são abstratos o suficiente para fornecerem cobertura, mas concreto suficiente para serem reconhecíveis [Coelho et al., 2004].

Pelo fato de nenhum problema passado ser exatamente igual a um problema atual, soluções passadas usualmente são adaptadas para solucionar novos problemas. A adaptação pode ser uma simples substituição de um atributo da solução por outro ou uma complexa e total modificação na estrutura da solução.

A aprendizagem em um sistema de RBC acontece principalmente pelo acúmulo de novas experiências em sua memória e pela correta indexação dos problemas. Um sistema de RBC somente se tornará eficiente quando estiver preparado para, a partir das experiências

passadas e da correta indexação dos problemas, aprender [Coelho et al., 2004].

### 3.4 Arquitetura Help-Desk Baseada Em Casos e Regras

Grande parte dos sistemas *Help-Desk* inteligentes, utiliza o paradigma baseado em regras ou o baseado em casos para implementá-lo. Essa abordagem, onde apenas uma metodologia é utilizada, não é suficientemente boa, pois ambas possuem deficiências [Coelho et al., 2004].

Mostra-se então, como alternativa, uma arquitetura implementada através do emprego do Raciocínio Baseado em Casos acoplado ao emprego de Regras. Um exemplo de um sistema *Help-Desk* inteligente é a arquitetura mostrada na figura 2 [Gorgônio, 1999].

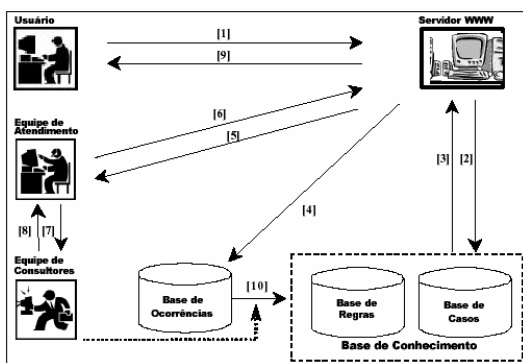


Figura 2: Arquitetura Web *Help-Desk* Utilizando RCB e Regras [Gorgônio, 1999].

A arquitetura compreende uma base de conhecimento, uma base de ocorrências, um servidor *Web*, uma equipe de atendimento e uma equipe de consultores no domínio de aplicação. A idéia central da arquitetura é fazer com que somente problemas de extrema complexidade no domínio cheguem às mãos dos consultores.

A base de conhecimento é composta por uma base de regras e uma base de casos, sendo essa a principal diferença dessa arquitetura híbrida para as arquiteturas convencionais. Junto com a base de conhecimento, a base de ocorrências forma a memória do sistema. A base de ocorrências é responsável por registrar todas as solicitações que foram feitas, porém não foram resolvidas de forma automática. É a partir da base de ocorrências que a base de conhecimento é atualizada, seja pela inserção de novos casos, ou pela criação de novas regras.

Nessa arquitetura, o servidor *Web* é responsável por ser a interface entre o usuário e o suporte. O sistema funciona da seguinte forma, ao necessitar de suporte, o usuário acessa o suporte (1). Em uma primeira instância, o sistema irá tentar responder diretamente à indagação do usuário através de busca à base de regras (2), que

soluciona problemas que ocorrem com maior frequência (3, 9). Quando o sistema não consegue uma resposta para o problema do usuário o passo seguinte será tentar obter da base de casos um caso que seja semelhante ao problema trazido e cuja solução possa ser utilizada para a situação desse usuário (3). Os casos resgatados são então apresentados ao usuário, dispostos em ordem de semelhança (9). Se, porém, o problema trazido também não estiver previsto na base de casos, este problema é inserido da base de ocorrências (4) e repassado para a equipe de atendimento (5). Ao resolver o problema, uma resposta é enviada ao usuário (6, 9). Só quando a equipe de atendimento não puder resolver o problema, ela poderá acionar a equipe de consultores (7,8). Cabe ainda à equipe de consultores a responsabilidade de inclusão de novos casos a povoarem a base de casos (10) de modo a permitir a aprendizagem do sistema. Estes casos podem ser extraídos a partir de uma base de ocorrência (casos reais) ou hipoteticamente criados (casos abstratos).

### 4. Desenvolvimento do sistema help-desk

O produto de software desenvolvido tem como objetivos gerenciar e controlar a qualidade dos serviços oferecidos pelo setor de suporte em informática da PML. A base desse controle consiste na gestão dos equipamentos que compõe a infra-estrutura tecnológica da prefeitura e no rastreamento dos chamados abertos pelos usuários.

Com o intuito de oferecer informações gerenciais, o software armazena os dados cadastrais dos usuários do sistema, bem como o registro dos equipamentos de informática e as suas características. Caso o equipamento esteja na garantia, ou coberto por um contrato de assistência técnica, terá associado a ele o código da empresa responsável pela sua manutenção, uma vez que o sistema também armazena informações relativas às empresas de assistência com as quais a PML trabalha.

Se o equipamento for um microcomputador ou uma antena de internet a rádio, alguns dados adicionais sobre sua configuração também são registrados. Além disso, o sistema armazena informações referentes aos produtos de software que a PML utiliza.

Quando um usuário necessita de assistência para a utilização de algum equipamento, ou mesmo de algum serviço do setor de informática, este deve acessar o site do *Help-Desk* e abrir um chamado, informando qual o tipo e a categoria do chamado.

Cada chamado de suporte está obrigatoriamente associado ao usuário que originou o chamado e será atendido de acordo com a prioridade associada ao setor que o usuário pertence. Dependendo da categoria do

chamado, o usuário deverá informar também qual é o equipamento ou software que apresenta problemas. Cada chamado aberto é automaticamente associado a um analista que fica responsável pela resolução do problema.

Durante o atendimento de um chamado, podem ocorrer vários eventos tais como: i) um equipamento pode ser enviado ou recebido da assistência técnica; ii) um chamado pode ser transferido de um analista para outro; iii) o atendimento pode ser temporariamente suspenso devido à falta de peças de reposição, etc. O sistema mantém registro dos eventos ocorridos para cada chamado.

Através desses dados, é possível um maior controle sobre o serviço de suporte da PML, uma vez que o sistema fornece informações mais precisas e confiáveis sobre os chamados.

O sistema possui ainda possibilidade de geração de relatórios gerenciais que auxiliam a avaliação do desempenho dos técnicos, a tomada de decisões e a elaboração de material para treinamento dos usuários e do quadro técnico.

Além disso, as estatísticas que poderão ser extraídas do sistema tornarão mais precisas as estimativas dos prazos necessários para a solução dos problemas que surgem com maior frequência.

#### 4.1. Modelagem de Dados

A abordagem escolhida para a representação do banco de dados do sistema é o paradigma relacional, para tal, será utilizado o modelo entidade relacionamento com notação UML [Gornik, 2004].

O poder de modelagem da linguagem UML [Booch et al., 1999] não se limita apenas ao desenvolvimento de software orientado a objetos. Um exemplo disso é o Diagrama de Modelagem de Dados, que é uma extensão da linguagem UML para que esta possa suportar a modelagem de bancos de dados relacionais. Essa ampliação inclui extensões especiais para tabelas, *schema* de banco de dados, chaves de tabelas, *triggers* e *constraints* [Gornik, 2004].

O Diagrama de Modelagem de Dados UML pode ser usado para descrever o desenvolvimento completo de um banco de dados relacional ou objeto relacional, desde as exigências empresariais até o modelo de dados físico.

#### 4.2. Modelagem do Produto de Software

A finalidade da modelagem é apresentar uma visão lógica do sistema, mostrando suas características e a sua funcionalidade, facilitando a sua compreensão e o seu

entendimento. Para a modelagem do sistema *Help-Desk* foram utilizados um subconjunto dos artefatos de *software* (diagramas e conceitos) presentes na UWE [Koch, 2001].

Os artefatos de software utilizados na modelagem foram: modelo de casos de uso, modelo conceitual e modelo navegacional.

O modelo de casos de uso é utilizado para descrever a funcionalidade da aplicação e a sua interação com os usuários.

Os casos de uso são apresentados utilizando o diagrama de casos de uso proposto pela UML, sendo escrito em termos de atores, casos de uso e o sistema a ser modelado.

O modelo de casos de uso é central, pois seu conteúdo é base do desenvolvimento de outros diagramas e da implementação da aplicação.

O objetivo do design conceitual é construir um modelo do domínio da aplicação levando em conta as exigências observadas nos casos de uso. Nesse modelo, as classes e os objetos participantes do sistema, bem como as relações entre eles, são modelados utilizando-se técnicas tradicionais de orientação a objeto. O modelo conceitual é representado por um diagrama de classes UML.

A modelagem de navegação de aplicações Web compreende a construção de dois modelos, o modelo do espaço de navegação e o modelo de estrutura de navegação [Koch, 2001].

O modelo do espaço de navegação é baseado no modelo conceitual e nos requisitos definidos nos casos de uso. A função desta modelagem é especificar quais classes do modelo conceitual serão visíveis ao usuário e quais serão os caminhos para se chegar a essas classes [Koch, 2001].

O modelo de estrutura de navegação é construído a partir do refinamento do espaço de navegação. Neste modelo, são adicionados elementos de acesso como índices, excursões, consultas e menus [Koch, 2001].

### 5. Conclusão

Este trabalho de pesquisa teve por objetivos realizar a modelagem e a implementação de um sistema *Help-Desk* para a PML.

Uma vez terminada a implementação do sistema, pôde-se notar a importância da modelagem no processo de confecção de um produto de software, pois além de servir de base para construção do sistema, a modelagem tem o intuito de servir também como documentação do software caso seja necessário realizar alguma alteração em seu código.

Como o sistema *Help-Desk* ainda está em fase de implantação, não é possível dizer com segurança que a ferramenta desenvolvida irá cumprir todos os objetivos propostos. Porém, pode-se concluir que, para o sistema funcionar com sucesso, é necessário alguns ajustes para ele se enquadrar totalmente na metodologia de serviços adotada pela PML.

Além dos ajustes no sistema, é necessário que a equipe de suporte habitue-se ao uso da ferramenta, pois quando os analistas estiverem totalmente familiarizados, o sistema *Help-Desk* será capaz de cumprir todos benefícios propostos.

Atualmente qualquer empresa, independente de seu porte, precisa garantir a disponibilidade de seus recursos de informática, gerenciar o pessoal de suporte responsável por esta tarefa e acompanhar o atendimento de solicitações dos usuários, solucionando o mais rapidamente possível as dificuldades encontradas.

A ausência de tal controle, ou um controle deficiente, implica em maiores despesas operacionais, desperdício de tempo e insatisfação dos usuários que dependem do funcionamento dos recursos de informática. Tudo isso se traduz em ineficiência e alto custo operacional, que precisam ser evitados pela organização.

Diante desse contexto, o sistema *Help-Desk* desenvolvido é de grande utilidade, uma vez que este oferece soluções gerenciais que auxiliam na prestação de um serviço técnico de informática com qualidade e possibilita um melhor monitoramento das ocorrências diárias da PML.

Além do sistema desenvolvido, são contribuições deste projeto as pesquisas bibliográficas e documentais realizadas para este trabalho, uma vez que estas podem servir como fonte de pesquisa sobre: Gestão do Conhecimento, Sistemas *Help-Desk*, Raciocínio Baseado em Casos e Raciocínio Baseado em Regras.

Além disso, toda a documentação sobre o desenvolvimento do sistema poderá servir de base à construção de outras ferramentas *Help-Desk* baseadas na arquitetura *thin client*.

O sistema *Help-Desk* gerado por esse projeto de pesquisa foi desenvolvido especificamente para a realidade e os problemas da PML. Sendo assim, várias funções que são desejáveis a um sistema *Help-Desk* comercial não foram implementadas, pois a prefeitura tinha necessidade por um sistema mais simples. Entre as características que poderiam ser incluídas no sistema destacam-se: i) suporte a SLA; ii) respostas Automáticas; iii) base de Conhecimento; iv) FAQ; e v) fórum de discussão.

## 6. Bibliografia

- [Aamodt, 1994] Aamodt, A. & Plaza, E., *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*, AI Communications, 7(i), pp.39-59, 1994.
- [Automidia, 2004] *Automidia Web Site*, Disponível em <http://www.automidia.com.br/>, Consultado em: 17/03/2004.
- [Booch et al., 1999] Booch, G., Rumbaugh, J. & Jacobson, I, *The Unified Modeling Language User Guide*, Addison-Wesley, 1999.
- [Coelho et al., 2004] Coelho, Á.V. S.; Ferneda, E.; M., A.; Barros, M. A. & Gorgônio, F. L. "*Help Desk Inteligente em Gestão do Conhecimento: Um Tratamento Integrador de Paradigmas*", Disponível em <http://www.exercito.gov.br/06OMs/gabcmtex/PEG-EB/Noticias/artigo.htm>, Consultado em 17/03/2004.
- [G&P, 2004] *G&P – Help-Desk*, Disponível em: [http://www.gpnet.com.br/help\\_desk/help\\_desk.shtml](http://www.gpnet.com.br/help_desk/help_desk.shtml), Consultado em 17/03/2004.
- [Gorgônio, 1999] Gorgônio, F. L. *Uma Arquitetura de Sistemas Inteligentes de Apoio ao Usuário*, Dissertação de Mestrado em Informática, UFPB, Campina Grande, 1999.
- [Gornik, 2004] Gornik, D. *UML Data Modeling Profile*, Disponível em <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/tp162.pdf>, Consultado em 17/03/2004.
- [Hall & Kandel, 1986] Hall, L. O'Higgins & Kandel, A., *Designing Fuzzy Expert Systems*,. Ed. Verlag, Rheinland, 1986.
- [OMG's UML, 2004] *Introduction to OMG's Unified Modeling Language™ (UML®)*, Disponível em [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm), Consultado em 17/03/2004.
- [Junior & Shimabukuro, 2003] Junior, K. S.; Shimabukuro, M.H., *Introdução a Sistemas de Gestão do Conhecimento Corporativo*, XXII JAI, Campinas, 2003.
- [Koch, 2001] Koch, N. P. *Software Engineering for Adaptive Hypermedia Systems*, Ludwig-Maximilians-Universität München, 2001.
- [Moresi, 2001] Moresi, E. A. D., *Gestão da informação e do conhecimento*, Inteligência organizacional e competitiva, Editora UnB, Brasília, 2001.
- [Pressman, 2002] Pressman, R. S., *Engenharia de software*. Ed. Makron Books, São Paulo, 2002.
- [Simões, 1999] Simões, C. *CBR – Raciocínio Baseado em Regras*, Notas de Aula, 1999.
- [DCC/UA, 2004] *Site de Suporte ao curso de Inteligência Artificial do DCC/UA*, Disponível em [http://www.dcc.fua.br/~dcc\\_ia/](http://www.dcc.fua.br/~dcc_ia/), Consultado em 17/03/2004.