

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

SISTEMA PARA HELP-DESK BASEADO EM WORKFLOW
UTILIZANDO SHELL UNIX

FABIO ERNANDES SIMON

BLUMENAU
2007

2007/1-11

FABIO ERNANDES SIMON

**SISTEMA PARA HELP-DESK BASEADO EM WORKFLOW
UTILIZANDO SHELL UNIX**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciências
da Computação — Bacharelado.

Prof. Oscar Dalfovo, Dr - Orientador

**BLUMENAU
2007**

2007/1-11

SISTEMA PARA HELP-DESK BASEADO EM WORKFLOW UTILIZANDO SHELL UNIX

Por

FABIO ERNANDES SIMON

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

Presidente: _____
Prof. Oscar Dalfovo, Doutor – Orientador, FURB

Membro: _____
Prof. Wilson Pedro Carli, Mestre – FURB

Membro: _____
Prof. Everaldo Artur Grahl, Mestre – FURB

Blumenau, 09 de julho de 2007.

Dedico este trabalho a minha esposa Elizângela, que sempre me incentivou com suas palavras de otimismo e amor, além da compreensão na conclusão de mais uma etapa conquistada em minha vida.

AGRADECIMENTOS

Agradeço acima de tudo aos meus pais Conrado e Eliane Simon, que mesmo estando distantes, sempre estiveram presentes em todos os momentos.

A minha esposa Elizângela Simon pela compreensão e companheirismo durante a realização deste trabalho.

Ao meu orientador, Oscar Dalfovo, por ter acreditado na idéia apresentada e dado forças incentivando na conclusão deste trabalho.

O que as grandes e puras afeições têm de bom é que, depois da felicidade de as ter sentido, há ainda a felicidade de recomendá-las.

Alexandre Dumas Filho

RESUMO

O processo de *workflow* pode ser descrito como um fluxo de informação, que envolve uma seqüência de atividades, passos ou tarefas que descrevem um processo de negócio. A informação flui de uma tarefa para outra, com base no conjunto de regras pré-definidas. Este trabalho apresenta a especificação e desenvolvimento de uma aplicação *web* para controle de chamados em um *help-desk* aplicando o conceito de *workflow* em tarefas realizadas durante o atendimento de chamados. As técnicas do PHP com as do AJAX utilizadas neste trabalho permitem uma apresentação de conteúdo ágil em relação às páginas *web* estáticas. A utilização da *shell* permite o monitoramento dos chamados registrados, cujo o usuário ainda não tenha recebido *e-mail* informando dos processos de abertura, fechamento, aguardando retorno e associação de técnico e categoria de *workflow* dos chamados registrados, tornando o processo independente e automatizando o *workflow*.

Palavras-chave: *Help-desk*. *Workflow*. *Shell*. AJAX.

ABSTRACT

The process of *workflow* can be described as an information flow, that involves a sequence of activities, steps or tasks that describe a business process. The information flows of a task for another one, on the basis of the set of daily pay-define rules. This work presents the specification and development of an application *web* for call control in one *help-desk* applying the concept of *workflow* in tasks carried through during the attendance of calls. The techniques of the PHP with the ones of AJAX used in this work allow to a presentation of agile content in relation to the static pages *web*. The use of *shell* allows the management of the registered calls, whose the user not yet has received *email* informing of the processes of opening, closing, waiting return and association of technician and category of *workflow* of the registered calls, becoming the independent process and automatizing the *workflow*.

Key-words: *Help-desk. Workflow. Shell. AJAX.*

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de uma especificação do processo de workflow.....	18
Figura 2 – Processo de atendimento a chamados	20
Figura 3 - O modelo tradicional de aplicações web comparado com o modelo do AJAX	22
Quadro 1 - Principais componentes do cabeçalho.....	23
Quadro 2 – Declaração de versão do XML	24
Quadro 3 - Principais componentes do cabeçalho.....	24
Quadro 4 - Principais componentes do corpo.....	24
Quadro 5 - Métodos do objeto XMLHttpRequest.....	27
Quadro 6 - Propriedades do objeto XMLHttpRequest	27
Quadro 7 – Exemplos de agendamento de shell no crontab.....	29
Figura 4 - Diagrama de caso de uso do administrador	34
Figura 5 - Diagrama de caso de uso do usuário.....	35
Figura 6 - Diagrama de caso de uso do técnico	36
Figura 7 - Diagrama de caso de uso do gerente.....	37
Figura 8 - Diagrama de estados do ciclo de vida do chamado	39
Figura 9 - Diagrama de atividades da abertura de chamado pelo usuário.....	40
Figura 10 - Diagrama de atividades da abertura de chamado pelo técnico	40
Figura 11 - Diagrama de classe do sistema	41
Quadro 08 – Rotina da shell - “Abertura de Chamados”	44
Quadro 09 – Rotina da shell - “Aguardando o Usuário”	45
Quadro 10 – Rotina da shell - “Fechamento de Chamados”	45
Quadro 11 – Rotina da shell - “Gerente associar categoria e técnico”.....	46
Quadro 12 – Rotina de manipulação e alteração da programação	47
Quadro 13 – Rotina de atualização do conteúdo das páginas.....	48
Quadro 14 – Código fonte do gráfico de chamados dentro e fora do padrão de tempo	49
Figura 12 – Tela de login do sistema.....	50
Quadro 15 – Tipos de usuário no sistema de help-desk	50
Figura 13 – Tela principal do administrador do sistema	51
Figura 14 – Tela de edição de um registro de cliente.....	52
Figura 15 – Tela de inclusão de um novo cliente	52
Figura 16 – Tela de inclusão de uma área para um cliente.....	53

Figura 17 – Tela de edição de área de um cliente	53
Figura 18 – Tela de listagem de categorias	54
Figura 19 – Tela de listagem dos nodos da categoria.....	55
Quadro 16 – Nodos do processo de workflow	55
Figura 20 – Tela de edição de nodos da categoria.....	56
Figura 21 – Tela de inclusão de nodo da categoria	56
Figura 22 – Tela de listagem de usuários do sistema	57
Figura 23 – Tela de inclusão de usuário do sistema	57
Figura 24 – Tela de edição de um usuário do sistema.....	58
Quadro 17 – Limitações por tipo de usuário	58
Figura 25 – Tela de agendamento de execução da Shell no Crontab	59
Figura 26 – Tela de Login de Usuário	60
Figura 27 – Nodos/Status dos chamados	61
Figura 28 – Tela de registro de chamados (Usuário).....	61
Figura 29 – E-mail com texto padrão informando encerramento do chamado	62
Figura 30 – Tela pesquisa de satisfação – Encerramento chamado	63
Figura 31 – Tela de Login do Técnico	64
Figura 32 – Tela de registro de chamados (Técnico)	65
Figura 33 – Tela de envio de imagem para servidor	65
Figura 34 – Tela de chamados em modo consulta.....	66
Figura 35 – Tela de cadastro de problemas	67
Figura 36 – Tela de Login do Gerente.....	68
Figura 37 – E-mail com texto padrão solicitando a associação da categoria e técnico.....	68
Figura 38 – Tela de Associação Categoria e Técnico	69
Figura 39 – Tela Relatório dos chamados abertos dentro e fora do padrão	70
Figura 40 – Relatório listagem dos chamados abertos dentro e fora do padrão.....	70
Figura 41 – Relatório gráfico dos chamados abertos dentro e fora do padrão	71
Figura 42 – Tela de geração de relatório de performance por técnico	72
Figura 43 – Relatório gráfico dos chamados abertos dentro e fora do padrão	72
Figura 44 – Tela de chamados excedidos.....	73

LISTA DE SIGLAS

ADH – AD Hoc

ADM – Administrativo

AJAX – Asynchronous Javascript and XML

API – Application Programming Interface

CGI – Common Gateway Interface

CSS – Cascading Style Sheets

DOM – Document Object Model

DTD – Declarações de Tipo de Documento

HTML – Hypertext Markup Language

NEC – Nós de Encaminhamento

NIF – Nós de Início e Fim

NTB – Nós de Trabalho

PHP – Personal Home Page

PRD – Produção

SQL – Structured Query Language

SVG – Scalable Vector Graphics

TF-ORM – Temporal Functionality in Objects with Roles Model

UML – Unified Modeling Language

W3C – World Wide Web Consortium

WFMC – WorkFlow Management Coalition

XHTML – eXtensible Hypertext Markup Language

XML – eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS DO TRABALHO	15
1.2 ESTRUTURA DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 WORKFLOW.....	17
2.2 HELP-DESK	19
2.3 AJAX.....	21
2.3.1 XHTML	23
2.3.2 CASCADING STYLE SHEET.....	24
2.3.3 DOCUMENT OBJECT MODEL	25
2.3.4 JAVASCRIPT	26
2.3.4.1 O objeto XMLHttpRequest	26
2.4 SHELL UNIX.....	28
2.5 TRABALHOS CORRELATOS	29
3 DESENVOLVIMENTO DO SISTEMA.....	31
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	31
3.2 ESPECIFICAÇÃO	33
3.2.1 Diagrama de Caso de Uso	33
3.2.1.1 Diagrama de Caso de Uso: Administrador	33
3.2.1.2 Diagrama de Caso de Uso: Usuário.....	35
3.2.1.3 Diagrama de Caso de Uso: Técnico	36
3.2.1.4 Diagrama de Caso de Uso: Gerente.....	37
3.2.2 Diagrama de Estados.....	38
3.2.3 Diagrama de Atividades	39
3.2.4 Diagrama de Classe.....	41
3.3 IMPLEMENTAÇÃO	42
3.3.1 Técnicas e Ferramentas Utilizadas.....	42
3.3.1.1 PHP.....	42
3.3.1.2 MySQL.....	43
3.3.1.3 VMware Server e Fedora 6.....	43
3.3.1.4 SHELL.....	43

3.3.1.4.1 Manipulação do Contrab.....	46
3.3.1.5 AJAX.....	47
3.3.1.6 Geração dos gráficos com a biblioteca baaChart.....	48
3.3.2 Operacionalidade do Sistema.....	49
3.3.2.1 Login de Administrador	50
3.3.2.2 Login de Usuário	59
3.3.2.3 Login do Técnico.....	63
3.3.2.4 Login do Gerente	67
3.4 RESULTADOS E DISCUSSÃO	73
4 CONCLUSÕES.....	75
4.1 EXTENSÕES	76
REFERÊNCIAS BIBLIOGRÁFICAS	77

1 INTRODUÇÃO

Grandes organizações, em sua maioria, possuem os processos não documentados, o que torna o trabalho ineficiente gerando a perda da competitividade de mercado. Durante anos, as organizações buscam nos processos organizacionais uma oportunidade na redução de custos operacionais e melhoria da qualidade, procurando soluções que permitam automatizar seus fluxos de trabalho. A automação de procedimentos onde documentos, informações ou tarefas possuem um conjunto predefinido de regras, tem como objetivo do negócio atingir ou facilitar o contexto de uma estrutura organizacional, definindo assim papéis funcionais e relacionamentos.

Sistemas de *workflow* monitoram a informação do fluxo de trabalho para gerenciar e controlar o trabalho mais eficientemente, reduzindo o trabalho da coordenação nos processos de negócio (NICOLAU, 1998, p. 10). Sistemas de *help-desk* possuem um fluxo de trabalho padrão, tornando assim, possível implementação do conceito de *workflow* para o registro de chamados. Dessa maneira, este trabalho automatiza um sistema de *help-desk*, aplicando conhecimentos capazes de solucionar esses problemas e implementar os conceitos relacionados à interação entre os processos. O termo *help-desk* designa o serviço de apoio a usuários para suporte e resolução de problemas técnicos em informática, realizado na maioria das vezes através de serviços telefônicos (HELPDESK, 2006).

O desenvolvimento deste trabalho foi um sistema de *help-desk* na *web* utilizando tecnologias como o *Asynchronous JavaScript and XML* (AJAX) para atualização dinâmica das informações apresentadas no conteúdo de uma página *web*. Dependendo de como o AJAX for utilizado, poderão ocorrer atualizações no conteúdo de uma página dinamicamente sem que os usuários notem (ASLESON; SCHUTTA, 2006, p. 17). A maior vantagem das aplicações AJAX é que elas rodam no próprio navegador *web*. Então, para estar hábil a executar aplicações AJAX, basta possuir algum dos navegadores modernos, ou seja, lançados após 2001 (AJAX, 2006).

O fluxo de trabalho, ou seja, o conjunto de procedimentos a serem executados, que compõem um processo, podendo ser de forma seqüencial ou não (NICOLAU, 1998, p. 12), é pré-definido através de um cadastro, mantido em uma base de dados, baseado no conceito de *workflow*. Com o objetivo de analisar o fluxo de trabalho, é desenvolvida uma *shell* agendada através de um recurso administrativo utilizando o utilitário *cron* do sistema UNIX. O termo *shell* é utilizado para descrever programas interagindo com o usuário e o computador,

recebendo, interpretando e executando códigos através de uma linha de comando apresentada na tela por meio de um *prompt* (SHELL, 2006). Já o utilitário *cron* pode ser utilizado para executar funções periódicas através de uma tabela registrada no arquivo “/etc/crontab”, cujo objetivo é ler arquivos fazendo com que os programas sejam executados conforme solicitação (GROFF, 1986, p. 106).

Em suma, o papel da *shell* é apresentar aos usuários informações pertinentes aos chamados, ou seja, a mesma tem o papel de monitorar todos os chamados registrados e repassar retornos aos usuários sobre os nodos de *workflow* pré determinados durante o desenvolvimento do sistema. Esse controle é automatizado e todo retorno ao usuário é encaminhado através de *e-mail*.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um sistema de *help-desk* baseado em *workflow* utilizando *shell* UNIX.

Os objetivos específicos do trabalho são:

- a) proporcionar o controle do tempo gasto no fluxo de trabalho, disponibilizando informações para a gerência responsável sobre os chamados atendidos fora do fluxo normal de atendimento;
- b) analisar performance do tempo médio gasto por técnico nos chamados durante um determinado período;
- c) apresentar relação de informações gráficas sobre o levantamento estatístico de chamados atendidos dentro e fora dos padrões do fluxo de trabalho;
- d) aplicar a tecnologia AJAX para efetuar atualizações de parte das informações de uma página dinamicamente, sem que seja necessário carregá-la novamente por inteiro.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está estruturado em quatro capítulos que estão descritos a seguir.

O primeiro capítulo contextualiza e justifica o desenvolvimento do trabalho.

No segundo capítulo é disponibilizada a fundamentação teórica necessária para um razoável conhecimento nas tecnologias e componentes utilizados no desenvolvimento do trabalho.

O terceiro capítulo tem como foco o desenvolvimento do sistema um sistema de *help-desk* baseado em *workflow* utilizando *shell* UNIX, descrevendo os requisitos principais do problema como também a especificação, implementação e operacionalidade do sistema.

O quarto capítulo apresenta as conclusões finais e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os temas: *workflow*, *help-desk*, AJAX e *shell* UNIX. Na última seção são descritos alguns trabalhos correlatos pesquisados.

2.1 WORKFLOW

Segundo a *WorkFlow Management Coalition* (WFMC), “um processo é um conjunto ordenado de atividades (seqüenciais ou paralelas) que são interligadas com o objetivo de alcançar um meta comum, sendo atividade conceituada como uma descrição de um fragmento de trabalho que contribui para o cumprimento de um processo” (WFMC, 2006).

Workflow é traduzido literalmente como fluxo de trabalho cujo principal objetivo é automatizar processos rotineiros dentro das organizações, ou seja, pode-se definir *workflow* como um sistema de gerenciamento onde o objetivo é rastrear todos os momentos do trabalho realizado controlando e automatizando as atividades de maneira a otimizar os processos de negócio (CRUZ, 2003).

Braggio (2006) diz que o principal objetivo almejado com a automatização dos processos é o aumento de sua eficiência e a efetividade das pessoas que trabalham em conjunto para executá-los, agregando objetivos que podem ser alcançados ou não, tais como:

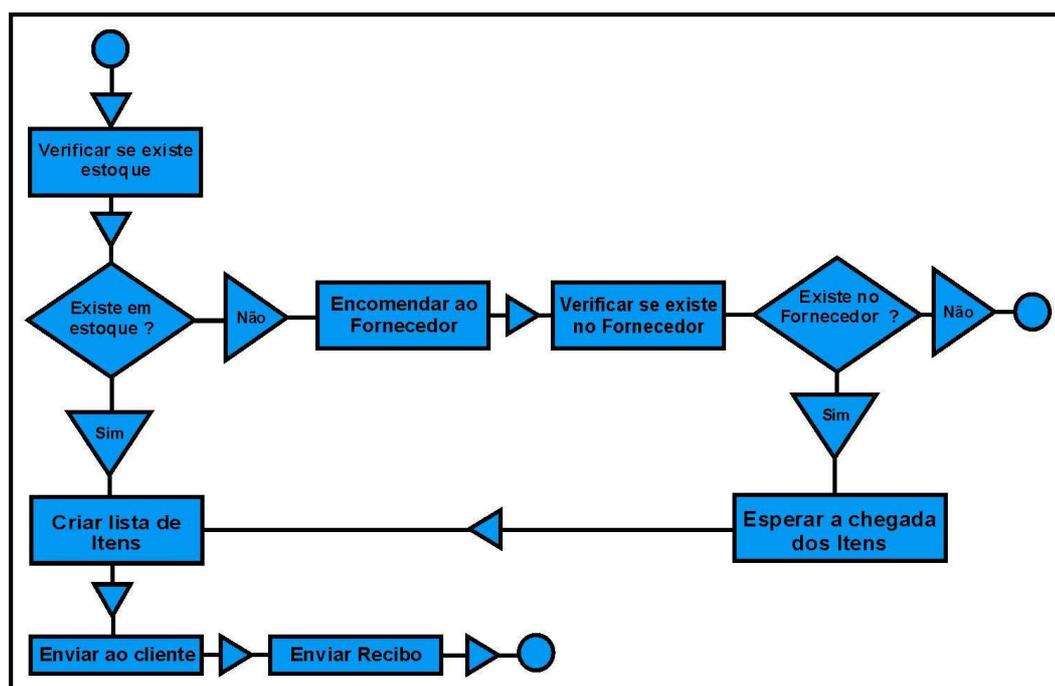
- a) acelerar os processos do negócio reduzindo o tempo nos ciclos de trabalho e, como resultado, prover um serviço mais rápido ao cliente;
- b) diminuir o custo dos processos do negócio através da automação;
- c) rastrear e controlar os processos do negócio à gerenciamento do processo;
- d) melhor fluência de comunicações entre os participantes do processo, reduzindo o tempo nas transferências de informações/documentos e execução de tarefas;
- e) melhor modo de identificar ocorrências de gargalos nos processos;
- f) colaboração mais efetiva entre participantes do processo e conhecimento compartilhado com toda a empresa.

Os termos “*workflow*” e “processo de *workflow*” ou simplesmente “processo” referem-se a uma descrição formal e executável de um processo de negócio que pode ser representado

por um grafo de fluxo que define a ordem de execução entre os vários nós no processo. Estes podem ser de três tipos (SEQUEIRA, 2006, p. 4):

- Nós de Trabalho (NTB): representam itens de trabalho que devem ser executados quer por um recurso humano quer por um recurso automático;
- Nós de Encaminhamento (NEC): definem a ordem pela qual os itens de trabalho devem ser executados e permitir a definição da ativação de nós de trabalho em paralelo ou mediante determinadas condições;
- Nós de Início e Fim (NIF): denotam o início ou o fim de um determinado fluxo de trabalho.

A Figura 1 representa um hipotético grafo de *workflow* que representa os passos que uma encomenda deve passar desde que o cliente executa o pedido até que o produto lhe é finalmente entregue. Na figura, os quadrados representam os NTB, os losangos representam os NEC e os círculos representam os NIF.



Fonte: adaptado de Sequeira (2006, p. 5).

Figura 1 – Exemplo de uma especificação do processo de workflow

A seqüência de eventos representada no grafo é a seguinte (SEQUEIRA, 2006, p. 5):

- quando chega um pedido ao sistema este verifica se o produto existe em estoque;
- caso exista em estoque passa para o passo 6 caso contrário para o passo 3;
- como não existe em estoque, encomenda o produto ao fornecedor;
- verifica se o produto ainda existe no fornecedor. Se existe vai para 5, caso contrário aborta;

- e) espera a chegada dos itens do fornecedor, quando chegarem vai para 6;
- f) cria uma lista de Itens a serem enviados ao cliente;
- g) envia os itens ao cliente;
- h) envia um recibo ao cliente;
- i) termina o processo.

Workflows podem ser caracterizados de três formas distintas (PLESUMS, 2002):

- a) *ad hoc* (ADH): descrevem processos simples onde é difícil encontrar um esquema para a coordenação e cooperação de tarefas, onde não há um padrão fixo para o fluxo de informações entre as pessoas envolvidas. Este tipo é muito conveniente e provê um razoável controle do processo, ou seja, quem fez o quê fez e onde está o trabalho agora. Geralmente usam o *e-mail* como plataforma, quando é informatizado. Os exemplos deste tipo são: processos de escritório, documentação de produtos e propostas de vendas;
- b) produção (PRD): é pré-definido e priorizado, suportando assim um grande volume não existindo negociações sobre quem fará ou como será tratado o trabalho. Pode ser completamente pré-definido ou seguir um procedimento geral, com alguns passos adicionais incluídos quando forem necessários, embora alguns autores não concordem com esta idéia. Os exemplos desse tipo são: processamento de requisição de seguros, processamento de faturas bancárias e de cartão de crédito;
- c) administrativo (ADM): é um meio-termo entre um *workflow* ADH e PRD. Envolve atividades fracamente estruturadas, repetitivas, previsíveis e com regras simples de coordenação de tarefa. Os exemplos deste tipo são: processamento de ordens de compras e autorização de férias e viagens.

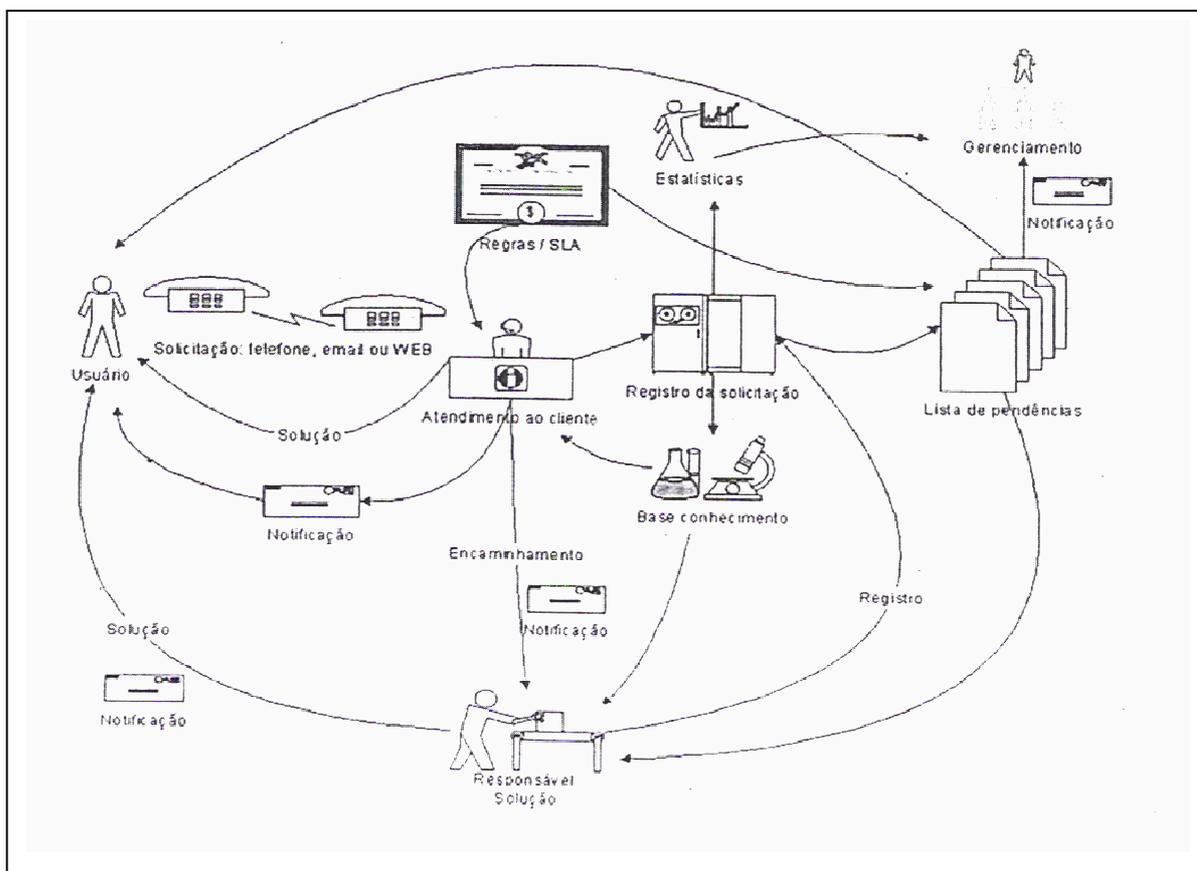
2.2 HELP-DESK

Segundo Statdlober (2006, p.15) diz que a gestão do atendimento em um help-desk procura oferecer uma metodologia para tratar e gerenciar diversas questões.

O processo de atendimento a chamados possui inúmeros elementos que o compõem, mas os oito elementos principais que compõem o processo de atendimento são (STATDLOBER, 2006, p. 15):

- a) usuário final contatar a central de atendimento;
- b) central de atendimento realizar o atendimento em si;
- c) registrar no sistema e encaminhar para solução;
- d) qualificar o chamado e encaminhar para uma fila de atendimento;
- e) responsável pelo atendimento atuar na solução;
- f) gerente possuir acesso a todos os chamados registrados;
- g) solução aplicada ao chamado ficar disponível para futuros chamados e demais técnicos;
- h) comunicar o usuário final sobre eventos durante o atendimento ao chamado.

A Figura 2 representa o modelo do processo de atendimento a chamados, ou seja, os processos envolvidos desde a solicitação de atendimento até a solução aplicada e a notificação de finalização do atendimento.



Fonte: Statdlober (2006, p.17).

Figura 2 – Processo de atendimento a chamados

2.3 AJAX

Segundo Reis e Trindade (2005), “as aplicações que sigam o modelo AJAX dão um grande passo quando se tenta comparar com a complexidade e riqueza das aplicações Desktop tradicionais”.

O AJAX funciona na maioria dos navegadores modernos e não requer nenhum software ou hardware patenteado. Na verdade, uma das vantagens dessa abordagem é que os desenvolvedores não precisam aprender alguma linguagem nova ou descartar o investimento que fizeram do lado do servidor. O AJAX é uma abordagem do lado do cliente e pode interagir com o J2EE, .NET, PHP, Ruby e scripts CGI - realmente não depende do servidor. (ASLESON; SCHUTTA, 2006, p. 13).

O AJAX permite o desenvolvimento de aplicativos *web* com interface “rica”. AJAX não é uma tecnologia. Crane e Pascarello (2005, p. 5) definem que “rica” refere-se ao modelo de interação do aplicativo com o usuário. Um modelo rico de interação com o usuário é aquele que suporta uma variedade de formas de entrada e responde intuitivamente e em tempo hábil a estas entradas.

Conforme Garrett (2005), AJAX na realidade são várias tecnologias, cada uma progredindo de forma independente, e que se juntaram de forma a poder explorar formas de melhorar a interação com os utilizadores em aplicações *web*. Dentre estas tecnologias, o AJAX incorpora:

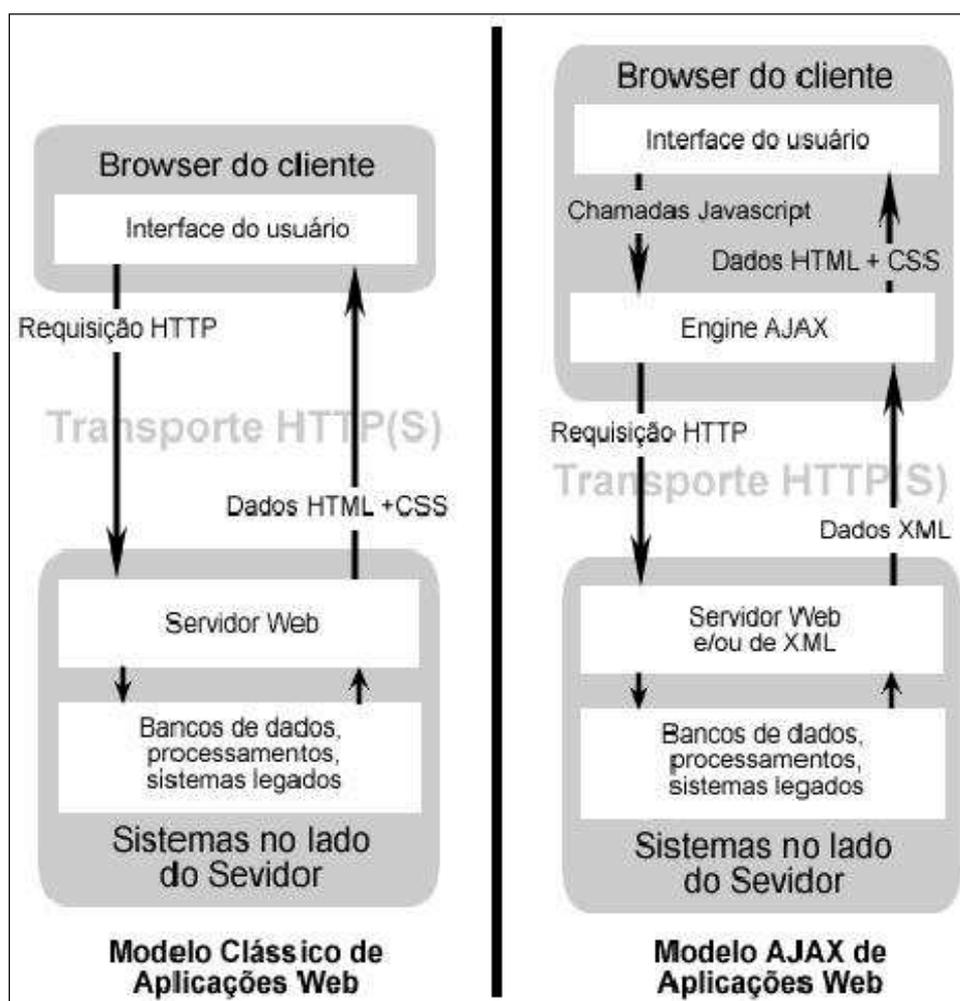
- a) *Cascading Style Sheets (CSS)* e *eXtensible Hypertext Markup Language (XHTML)* para exibição das informações;
- b) *Document Object Model (DOM)* para interagir dinamicamente com as informações apresentadas;
- c) *XMLHttpRequest* para trocar dados de forma assíncrona com o servidor;
- d) *JavaScript* como linguagem de script para suporte às tecnologias acima.

Limeira (2006, p. 16) diz que as principais vantagens da utilização do AJAX são:

- a) menor uso de banda: utilizando AJAX, menos informações irão trafegar na rede, diminuindo sensivelmente o uso de banda e tornando a navegação muito mais rápida;
- b) respostas mais rápidas: com AJAX, os usuários não precisam esperar que toda a página seja processada a cada vez que fazem uma solicitação. O usuário pode interagir com uma parte da página e enquanto aguarda o retorno, pode continuar interagindo com as demais seções da página, tornando a navegação mais eficiente;

- c) mais interatividade: com AJAX, sem precisar usar recursos pesados de programação ou necessidade de softwares proprietários, é possível disponibilizar aplicações com alta capacidade de interatividade e usabilidade. As interfaces são mais amigáveis e divertidas para o usuário final.

A Figura 3 ilustra o modelo clássico de uma aplicação *web* comparando com uma aplicação em AJAX, que é otimizada e reduz o consumo de banda relacionado à comunicação entre a aplicação que está no navegador do cliente com o servidor.



Fonte: Garrett (2005).

Figura 3 - O modelo tradicional de aplicações web comparado com o modelo do AJAX

Segundo Góis (2006, p.21), “aplicativos que utilizam as técnicas do AJAX tentam tratar a maior parte possível das ações do usuário no próprio cliente, sem que seja necessário requisitar informações para o servidor, e, quando isto é inevitável, é feita uma requisição assíncrona e o servidor retorna apenas de dados (sem formatação) que serão formatados e exibidos pelo cliente. A utilização do AJAX é recente e ainda não muito difundida pela internet, mas seu uso vem crescendo, principalmente pela influência do Google, que utiliza

massivamente o AJAX na construção de aplicativos *web* que desafiam cada vez mais os limites da interatividade das páginas *web*. Sua utilização tende a ser cada vez maior já que suas tecnologias base já são maduras e suportadas por todos os navegadores modernos”.

2.3.1 XHTML

A *World Wide Web Consortium* (W3C) define o XHTML como uma reformulação da linguagem de marcação *Hypertext Markup Language* (HTML) baseada em *eXtensible Markup Language* (XML). O XHTML é uma adaptação das *tags* de marcação HTML com regras da XML. O objetivo do XHTML é padronizar a exibição de páginas *web* em diversos dispositivos, como por exemplo televisão, palm, celular, entre outros (W3C, 2002).

XHTML é uma linguagem de marcação bastante familiar para quem conhece HTML e a transformação de um documento existente de HTML para XHTML é uma tarefa bem simples. A junção das duas linguagens resultou no XHTML, uma linguagem quase igual ao HTML original, o que facilita muito aos programadores, que estão acostumados com todas as *tags* e códigos desde que foi criada, mas que é capaz de apresentar a "flexibilidade" da linguagem XML de levar seu conteúdo registrado nela para outras plataformas (GONÇALVES, 2005, p. 2).

Todos os documentos XHTML podem ser divididos em duas partes básicas: o cabeçalho e o corpo; assim como duas especiais: a declaração da versão e a declaração do tipo do documento. O cabeçalho possui informações sobre o documento em si, conforme é demonstrado na Quadro 1 (GONÇALVES, 2005, p. 5).

Elemento	Descrição	Obrigatório
<code><title> </title></code>	Deve ser o primeiro elemento do cabeçalho, informa o texto que aparece na barra de título do navegador.	Sim
<code><meta> <meta></code>	Contém informações sobre o conteúdo do documento	Não
<code><link> </link></code>	Elemento utilizado para realizar a ligação entre os documentos e as páginas contendo as folhas de estilo	Não
<code><style> <style></code>	Informações sobre as folhas de estilo usadas em um determinado documento	Não
<code><object> </object></code> <code><script> <script></code>	Espaço utilizado para inserção de código, por exemplo: Javascript	Não
<code><base> <base></code>	Especifica o endereço do documento XHTML	Não

Fonte: Gonçalves (2005, p. 5).

Quadro 1 - Principais componentes do cabeçalho

A declaração de versão, que informa versão da linguagem XML que será utilizada na descrição do documento, também faz parte do cabeçalho, sua função é determinar com será o processo de codificação do texto (*encoding*) utilizado no documento apresentado, conforme pode ser visto através do Quadro 2 (GONÇALVES, 2005, p. 5).

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

Fonte: Gonçalves (2005, p. 5).

Quadro 2 – Declaração de versão do XML

Já a declaração do tipo do documento (DOCTYPE), que informa a DTD e que deve ser utilizada para validar o documento apresentado é outra declaração que integra o cabeçalho, conforme é demonstrado no Quadro 3 (GONÇALVES, 2005, p. 5).

```
<!DOCTYPE html PUBLIC "-//w3c/DTD XHTML 1.0 Transitional //EN"
"http://www.w3.org/TR/xhtml1-transational.dtd">
```

Fonte: Gonçalves (2005, p. 5).

Quadro 3 - Principais componentes do cabeçalho

A segunda e principal parte do documento é o corpo, nele são inseridos os elementos de marcação, os principais elementos do corpo do documento são descritos no Quadro 4 (GONÇALVES, 2005, p. 6).

Nome	Marcador	Descrição
Parágrafo	<p> </p>	É um dos marcadores mais utilizados, sua representação depende do navegador, mas normalmente é representado por um espaço antes e depois do parágrafo.
Divisão	<div> </div>	Utilizado com folhas de estilo na especificação de blocos e texto.
Título	<hn> </hn>	Introduzem títulos no documento, podem ser de seis níveis (1 até 6), seu tamanho de fonte varia de 24 à 10 pontos.
Blockquete	<bloquete> </loquete>	Utilizados para criação de citações, que inserem uma linha antes e depois do texto, assim como um nível de indentação.
Preserve	<pre> </pre>	Utilizado para preservar a formatação do texto, não interpretando seu conteúdo.

Fonte: Gonçalves (2005, p. 6).

Quadro 4 - Principais componentes do corpo

2.3.2 CASCADING STYLE SHEET

CSS é uma linguagem de folhas de estilo usada para descrever como um documento deve ser exibido (GÓIS, 2006, p. 22). Segundo a *W3schools* (2006), o CSS veio para fazer com que o HTML voltasse a ter o seu propósito inicial. As *tags* HTML originalmente

deveriam dizer o que são os elementos do documento, como por exemplo, “cabeçalho”, “parágrafo” e “tabela” e não como eles deveriam ser exibidos. Com a popularização da internet, os fabricantes de navegadores começaram a adicionar *tags* proprietárias para formatação dentro do documento HTML.

Com CSS é possível definir padrões de aparência e comportamento dos elementos de uma página. As cores, parágrafos, margens, espaçamentos, espessura ou qualquer outro elemento da página, pode ser definido em apenas um arquivo e, assim, a qualquer momento, pode-se mudar totalmente a forma de visualização de todas as páginas de um site, apenas alterando esse arquivo que contém as definições de CSS. Esse recurso torna possível que, em conjunto com AJAX, se possa mudar dinamicamente a aparência do site, para se adaptar ao gosto de cada usuário (LIMEIRA, 2006, p. 17).

2.3.3 DOCUMENT OBJECT MODEL

Document Object Model (DOM) é uma descrição de como documentos XML e HTML podem ser representados em uma estrutura de árvore. O DOM define um conjunto de interfaces de programação orientada a objeto que permitem interpretar estes documentos e realizar operações sobre o seu conteúdo (GÓIS, 2006, p. 24).

O DOM possui alguns componentes específicos para se trabalhar com XML, além de funcionalidades adicionais, para permitir que todos os aspectos de um documento XML possam ser representados. Os componentes XML do DOM adicionam suporte para (WIECZOREK, 2004, p. 26):

- a) declarações de tipo de documento (DTD);
- b) entidades;
- c) seções CDATA;
- d) seções condicionais.

Com a interface de programação de aplicativos (API) DOM é possível criar documentos, navegar pela estrutura de um documento qualquer, incluir, alterar e apagar nós do documento (o próprio nó ou seu conteúdo). Isso é válido tanto para páginas HTML quanto para documentos XML, ou seja, na grande maioria das vezes utilizar-se-á a API DOM para manipular uma página *web* pelo AJAX (SOARES, 2006, p.68).

2.3.4 JAVASCRIPT

De acordo com Crane e Pascarello (2006, p. 34) o Javascript é o jogador central do AJAX. É ele quem controla todo o fluxo de uma aplicação AJAX. O Javascript é o responsável por enviar as requisições ao servidor, receber as respostas de forma assíncrona e determinar como e quando serão exibidos os resultados.

Javascript é uma linguagem de *scripts* voltada para executar operações no lado do cliente. Com ela, é possível fazer validações de campos, abertura de janelas, controle da utilização de botões, mensagens de alertas, confirmações e principalmente para criar uma interatividade maior do usuário com a página utilizada. Com Javascript também é possível alterar o estilo de uma página de forma dinâmica (LIMEIRA, 2006, p. 21).

Javascript não é uma linguagem compilada, ela é interpretada e é inserida junto com o código fonte HTML das páginas. Apesar de não possuir alguns conceitos importantes de orientação a objeto (definição de classe e herança explícita), ainda assim encontram-se alguns conceitos de orientação a objetos no Javascript, como o conceito de objeto, herança implícita, encapsulamento e o polimorfismo (LIMEIRA, 2006, p. 22).

2.3.4.1 O objeto XMLHttpRequest

Crane e Pascarello (2006, p. 56) dizem que antes do objeto XMLHttpRequest já era possível fazer requisições assíncronas ao servidor através da utilização de IFrames ocultos, mas isto é basicamente um truque com um objeto (o IFrame) que foi originalmente criado para exibir conteúdo visível em páginas *web*. A utilização de IFrames também não é mais recomendada pela W3C e este objeto nem mesmo existe no modo Strict (um modo de sintaxe mais rígida) do HTML que é o atualmente recomendado para novas páginas.

Góis (2006, p. 26) diz que o objeto XMLHttpRequest é uma extensão não padronizada do Javascript e por este motivo, existem diferentes nas implementações deste objeto, mas todas compartilham uma interface comum, tornando como única diferença em sua utilização a forma de criação ou instanciação do objeto. Este objeto é utilizado exclusivamente para fazer requisições de forma assíncrona ao servidor.

XMLHttpRequest é o objeto que faz a conexão assíncrona entre a página e o servidor de aplicações *web*, ou seja, é a tecnologia principal do AJAX, sem ele, o mesmo não existiria.

Trata-se de um objeto Javascript que pode ser usado para fazer requisições ao servidor *web*, em segundo plano, sem congelar o navegador ou recarregar a página atual. O objeto XMLHttpRequest é hoje parte da especificação do DOM, ou seja, qualquer navegador que queira oferecer suporte aos padrões precisa implementar o objeto XMLHttpRequest (SOARES, 2006, p.83).

O XMLHttpRequest possui métodos e propriedades, conforme são apresentados nos Quadro 5 e Quadro 6 (ASLESON; SCHUTTA, 2006, p.25):

Métodos	Descrição
Open (método, URL, Assíncrona, usuário, senha).	Esse método relaciona o objeto à página web que se deseja conectar. O argumento de método pode ser GET, POST ou PUT. O endereço da URL pode ser relativo ou absoluto. Os três últimos parâmetros são opcionais.
Send(content)	Envia a solicitação para o Servidor. Caso a conexão tenha sido aberta com o parâmetro “Assinc” igual a “false” (indicando que a conexão não é assíncrona), esse método aguarda a resposta do servidor; caso contrário, não há espera (o que deve ser o padrão para aplicações AJAX).
setRequestHeader	Configura o cabeçalho http especificado com o valor fornecido.
getResponseHeader	Retorna o valor da string do cabeçalho especificado.
getAllResponseHeaders	Retorna uma string com todos os cabeçalhos http especificados.
Abort	Interrompe o processamento atual do objeto XMLHttpRequest.

Fonte: Asleson e Schutta (2006, p. 25).

Quadro 5 - Métodos do objeto XMLHttpRequest

Propriedades	Descrição
Status	Contém o código de status enviado pelo servidor web, isto é, 200 para OK, 404 para Não Encontrado e assim por diante.
statusText	A versão em texto do código de status http, isto é, OK ou Not Found e assim por diante.
readyState	O estado da solicitação. Os cinco valores possíveis são: “0” – “não inicializada”, “1” – “carregando”, “2” – “carregada”, “3” – “interativa” e “4” – “concluída”.
responseText	A resposta do servidor na forma de uma string.
responseXML	A resposta do servidor em formato XML.
Onreadystatechange	O manipulador de eventos que é acionado a cada mudança de estado, normalmente, uma chamada a uma função Javascript.

Fonte: Soares (2006, p. 88).

Quadro 6 - Propriedades do objeto XMLHttpRequest

2.4 SHELL UNIX

Shell é a linha de comando que interpreta os comandos digitados pelos usuários através de um *prompt*, sendo versátil e completo o suficiente para que sejam feitas interfaces interativas, tais como (JARGAS, 1999):

- a) rotinas de cálculos;
- b) instaladores de software;
- c) manipulação de banco de dados;
- d) *common gateway interface* (CGI);
- e) rotinas de *backup*.

O termo *shell* traduzido para português significa concha, porém pode ser definido como meio de interação entre o usuário e o computador, podendo receber, interpretar e executar os comandos de usuário, tendo como facilidades o reaproveitamento de comandos e manipulação da linha de comandos (SHELL, 2006).

O termo *shell* é um conjunto de comandos que podem ou não estar agendados no sistema operacional, sendo apresentados na ordem de sua execução e tendo como comentários o sinal do jogo da velha descrevendo as etapas de sua execução (MICHAEL, 2003, p. 2). O agendamento da execução dos programas pode ser efetuado pelo programa *cron*, executado em tempos específicos, que lê os comandos do arquivo “/usr/lib/crontab”. Cada linha em *crontab* especifica um comando e o horário em que o mesmo deve ser executado (CHRISTIAN, 1985, p. 264). O *crontab* é um programa que edita o arquivo onde são especificados os comandos a serem executados, a hora e dia de execução pelo *cron* (CRONTAB, 2006).

O Quadro 7 apresenta algumas execuções de *shell* no *crontab*, demonstrando um exemplo de como pode ser programada no ambiente UNIX.

```

Todo dia de hora em hora (hora cheia)
00 * * * * /bin/script
A cada cinco minutos todos os dias (divisão por 5 do intervalo 00-59)
00-59/5 * * * * /bin/script
Nas seguintes horas: 10, 12, 16, 18, 22 aos 15 minutos da hora
15 10,12,16,18,22 * * * /bin/script
Nos primeiros cinco dias do mês às 19:25
25 19 01-05 * * /bin/script
De segunda a sexta ao meio-dia e a meia-noite
00 00,12 * * 2-5 /bin/script
Script rodar Segunda,Quarta,Sexta as 2 horas
0 2 * * mon,wed,fri /bin/script
Script para rodar Terça,Quinta as 3 horas
0 3 * * tue,thu /bin/script

```

Fonte: CRONTAB (2006).

Quadro 7 – Exemplos de agendamento de shell no crontab

2.5 TRABALHOS CORRELATOS

Foram analisados quatro trabalhos correlatos descritos a seguir.

O primeiro tem como objetivo implementar um sistema para empresas que prestam serviço de atendimento ao cliente, mais conhecido como *help-desk*, utilizando gestão do conhecimento e técnicas de raciocínio baseado em casos. Possibilita automatizar a busca por soluções similares a novas ocorrências, ou seja, utilizar ocorrências de soluções aplicadas a em atendimentos já realizados (WILVERT, 2004, p. 15).

O segundo trata do desenvolvimento de uma especificação rigorosa de *workflow* em um nível conceitual, utilizando como modelo de dados referencial o *Temporal Functionality in Objects with Roles Model* (TF-ORM), cujo objetivo é formalizar o comportamento interno e seu relacionamento para o ambiente. Apresenta descrições formais do *workflow* que podem ser usadas como entradas para gerar o esquema de dados do *workflow* e código de regras para seu gerenciamento, tal como exceções ou pré-condições de tarefas (NICOLAU, 1998, p. 62).

O penúltimo, denominado 0800net, é um software para gerenciar um sistema de *help-desk* em ambiente *web*, aplicando o conceito de *workflow* com o objetivo de automatizar todas as fases de um chamado aberto no ambiente. Permite abertura de chamados pelo cliente ou por *e-mail*, inventário, controle de projetos, extração de dados para gerenciamento de índices de controle, além de ser implementado para *web*, o que viabiliza o acesso a informação em qualquer lugar do mundo (HBSIS, 2006).

O último tem como objetivo principal o desenvolvimento de uma aplicação *web* para

monitoramento de ambientes físicos utilizando as técnicas do AJAX e a linguagem *Scalable Vector Graphics* (SVG), possibilitando a visualização gráfica da planta-baixa, mapa, terreno, disponibiliza formas típicas de aplicativos *desktop*, tais como menus sensíveis ao contexto. Utilizando-se das técnicas do AJAX, para requisições assíncronas, assim permitindo que a aplicação esteja sempre disponível às ações do usuário, mas possa notificá-lo imediatamente após a ocorrência de um evento (GÓIS, 2006, p. 15).

3 DESENVOLVIMENTO DO SISTEMA

O sistema desenvolvido por este trabalho é um aplicativo para *help-desk* baseado no conceito de *workflow* voltado para registro e acompanhamento de chamados, que tem por finalidade permitir uma visualização gráfica e informações estatísticas de chamados atendidos dentro e fora do padrão estipulado.

Este capítulo aborda o desenvolvimento do sistema, apresentando as especificações em formato de diagramas da *Unified Modeling Language* (UML), tais como diagramas de casos de uso, diagrama de classes, diagrama de atividades e diagrama de estados.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os requisitos, classificados como Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF), descrevem o que o sistema deve fazer e o que não deve fazer. Os RF demonstram às funcionalidades e o comportamento que o sistema deve possuir em determinadas situações, podendo também explicitar o que o sistema não deve realizar. Os RNF demonstram as restrições que o sistema terá sobre alguns serviços ou funções oferecidas como usabilidade, navegabilidade, portabilidade, segurança, *hardware*.

Os requisitos do sistema foram levantados observando as necessidades de um sistema de *help-desk* para controlar as tarefas dentro do atendimento de chamados com a implementação do processo de *workflow*. A seguir, são detalhados os RF e os RNF previstos e implementação durante o desenvolvimento do sistema para *help-desk*:

- a) cadastrar cliente a ser utilizado para o atendimento de chamados (RF);
- b) cadastrar área associada ao cliente para atendimento de chamados (RF);
- c) cadastrar serviço associado à área para atendimento dos chamados (RF);
- d) cadastrar a versão do serviço para atendimento dos chamados (RF);
- e) cadastrar o problema dos serviços para atendimento dos chamados (RF);
- f) cadastrar a categoria de *workflow* para atendimento dos chamados (RF);
- g) cadastrar nodos de origem e destino da categoria de *workflow*, determinando tempo para cada um dos nodos (RF);
- h) cadastrar o gerente responsável pelo serviço (RF);

- i) cadastrar usuários, técnicos e administradores do sistema (RF);
- j) associar áreas a usuários e técnicos para atendimento de chamados (RF);
- k) cadastrar tipo de usuário (RF);
- l) cadastrar nodos do *workflow* (RF);
- m) disponibilizar tela de controle do agendamento de execução da *shell* UNIX, com escolha de execução em minutos, sendo executada nos 7 dias da semana (RF);
- n) gerar relatório e gráfico de chamados dentro e fora do padrão de tempo (RF);
- o) gerar relatório e gráfico estatístico de tempo gasto do atendimento de chamados por técnico ou serviço (RF);
- p) gerar relatório e gráfico de performance por técnico de chamados dentro e fora do padrão de tempo (RF);
- q) gerar relatório de nível de atendimento de chamados por técnico (RF);
- r) ser implementado utilizando PHP com acesso a banco de dados MySQL (RNF);
- s) ser compatível com o Internet Explorer 5.5 ou superior (RNF);
- t) utilizar o AJAX para atualizações de conteúdo em tempo dinâmico, ou seja, sem que seja necessário carregar da página *web* inteira novamente (RNF);
- u) utilizar a biblioteca baaChart para a confecção dos gráficos estatísticos de chamados (RNF);
- v) os perfis de usuário para acesso ao sistema são (RNF):
 - usuário – abre chamados, pesquisa histórico de chamados, visualiza status de chamados,
 - técnico – abre chamados, pesquisa histórico de chamados, atende chamados, cadastra problema dos serviços,
 - gerente – associa técnico e categoria a chamados abertos pelo usuário via sistema, visualiza relatórios e gráficos estatísticos dos chamados registrador no sistema,
 - administrador – cadastra clientes, áreas, serviços, versões dos serviços, problemas, categorias de *workflow*, nodos do *workflow*, usuários, técnicos, gerentes, administradores e controla a execução da *shell* via sistema;
- w) utilizar VMware Server e Fedora 6 para virtualização do sistema operacional LINUX e execução da *shell* de controle do fluxo de trabalho (RNF);

3.2 ESPECIFICAÇÃO

Para a especificação do sistema foi adotado o padrão *Unified Modeling Language* (UML) por atender a todas as necessidades de representação das soluções propostas neste trabalho, bem como apresentar de maneira clara as definições do sistema. Para criação dos diagramas foi utilizada a ferramenta *Enterprise Architect* (SPARX SYSTEMS, 2006) devido ao ótimo suporte que oferece na confecção dos diagramas necessários para a apresentação do sistema.

A seguir serão exibidos os diagramas de caso de uso sistema, diagramas de classes, diagrama de atividades, diagrama de estados e por fim, serão abordadas as ferramentas e técnicas utilizadas na especificação no sistema desenvolvido.

3.2.1 Diagrama de Caso de Uso

Os casos de uso representam as interações entre o sistema e usuários externos ao sistema. A seguir serão apresentados os casos de uso para os usuários externos ao sistema. Juntamente com os diagramas será apresentada uma descrição de cada caso de uso.

3.2.1.1 Diagrama de Caso de Uso: Administrador

O ator administrador é responsável pelo cadastro de clientes, áreas dos clientes, serviços das áreas, versões dos serviços, categorias de *workflow*, nodos das categorias de *workflow*, usuário, técnicos, gerente e os próprios administradores. Na Figura 4 é apresentado o diagrama de casos de uso da área do sistema disponível ao administrador.

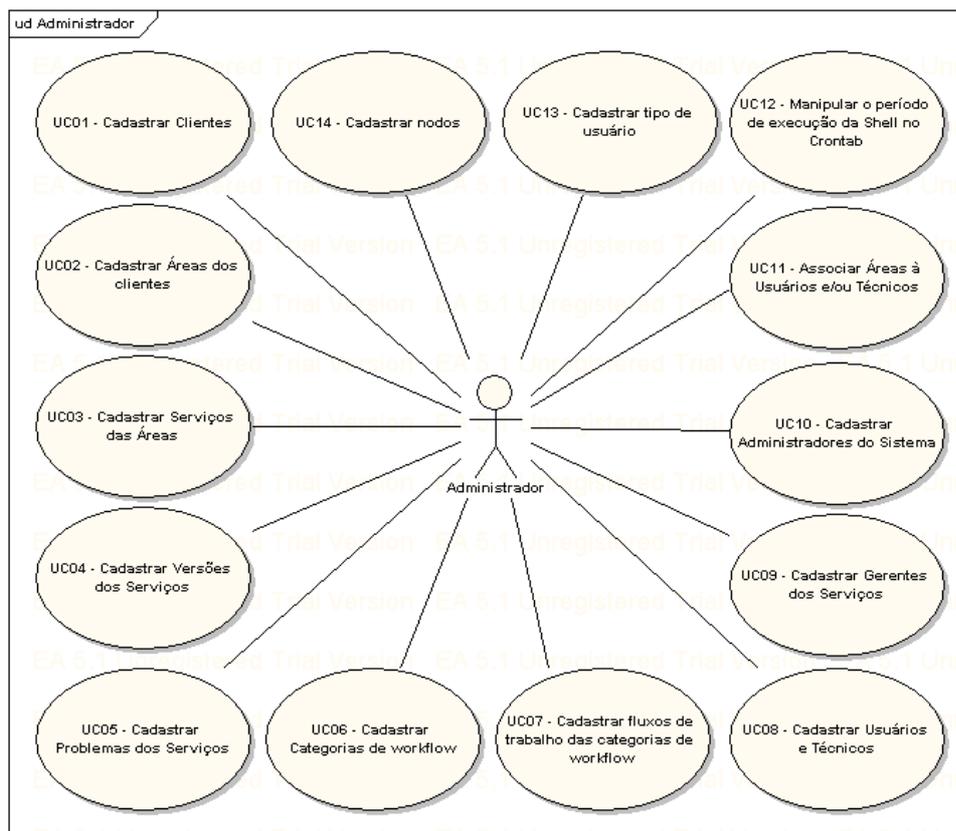


Figura 4 - Diagrama de caso de uso do administrador

A área disponível ao administrador define-se pelos seguintes casos de uso:

- a) cadastrar clientes: permite que o administrador cadastre os clientes que utilizam o sistema;
- b) cadastrar áreas dos clientes: permite que o administrador cadastre as áreas dos clientes que utilizam o sistema;
- c) cadastrar serviços das áreas: permite que o administrador cadastre os serviços das áreas dos clientes que utilizam o sistema;
- d) cadastrar versões dos serviços: permite que o administrador cadastre as versões dos serviços utilizados no sistema;
- e) cadastrar problemas dos serviços: permite que o administrador cadastre os problemas dos serviços a serem utilizados no atendimento de chamados;
- f) cadastrar categorias de *workflow*: permite que o administrador cadastre categorias de *workflow* a ser utilizadas no atendimento de chamados;
- g) cadastrar fluxos de trabalho das categorias de *workflow*: permite que o administrador cadastre o fluxo de trabalho a ser utilizado dentro de cada categoria de *workflow*;
- h) cadastrar usuários e técnicos: permite que o administrador cadastre usuários e

- técnicos de atendimento para utilização do sistema;
- i) cadastrar gerentes dos serviços: permite que o administrador cadastre os gerentes dos serviços, para posterior verificação de relatórios e gráficos estatísticos de chamados registrados no sistema;
 - j) cadastrar administradores do sistema: permite que o administrador cadastre mais administradores para o sistema;
 - k) associar áreas à usuários e técnicos: permite que o administrador associe áreas de clientes para os usuários do sistema;
 - l) manipular o período de execução da *shell* no *crontab*: permite que o administrador defina o período de execução da *shell* agendada no *crontab*;
 - m) cadastrar tipo de usuário: permite que o administrador cadastre tipos de usuários do sistema;
 - n) cadastrar nodos do *workflow*: permite que o administrador cadastre nodos para o processo de *workflow*.

3.2.1.2 Diagrama de Caso de Uso: Usuário

O ator usuário é responsável pela abertura de chamados no sistema, além de acompanhar o andamento destes e poder efetuar verificações históricas de chamados abertos por ele. Na Figura 5 é apresentado o diagrama de casos de uso da área do sistema disponível ao usuário.

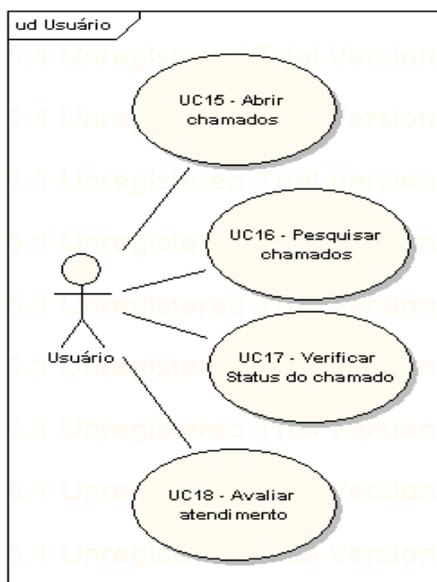


Figura 5 - Diagrama de caso de uso do usuário

A área disponível ao usuário define-se pelos seguintes casos de uso:

- a) abrir chamados: permite que o usuário abra chamados no sistema;
- b) pesquisar chamados: permite que o usuário pesquise seus chamados abertos ou encerrados;
- c) verificar status do chamado: permite que o usuário verifique o status de atendimento de chamados, apresentando em qual passo de processo de *workflow* o chamado encontra-se;
- d) avaliar atendimento: permite que o usuário avalie o atendimento de chamados encerrados, qualificando o atendimento em satisfatório, regular ou insatisfatório.

3.2.1.3 Diagrama de Caso de Uso: Técnico

O ator técnico é responsável pela abertura, edição e atendimento dos chamados abertos em seu nome ou área, além de cadastramento de problemas dos serviços e acompanhamento do andamento de chamados, podendo efetuar verificações históricas de chamados. Na Figura 6 é apresentado o diagrama de casos de uso da área do sistema disponível ao técnico.

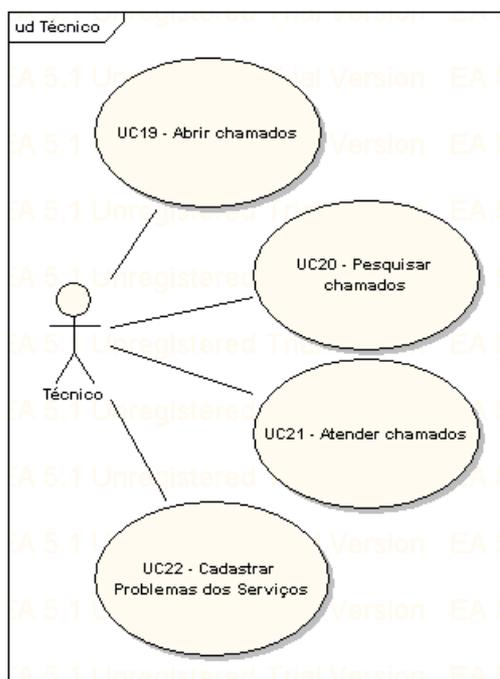


Figura 6 - Diagrama de caso de uso do técnico

A área disponível ao técnico define-se pelos seguintes casos de uso:

- a) abrir chamados: permite que o técnico abra chamados no sistema para as áreas que estiverem liberadas;

- b) pesquisar chamados: permite que o técnico pesquise chamados abertos ou encerrados em seu nome ou em áreas liberadas;
- c) atender chamados: permite que o técnico atenda aos chamados em seu nome ou áreas liberadas;
- d) cadastrar problemas dos serviços: permite que o técnico cadastre os problemas dos serviços a serem utilizados no atendimento de chamados.

3.2.1.4 Diagrama de Caso de Uso: Gerente

O ator gerente é responsável pela associação de chamados que estejam sem categoria de *workflow* ou técnico responsável definido, também tendo uma área disponível para visualização de relatórios e gráficos estatísticos de chamados atendimentos dentro e fora do padrão de tempo estipulado nos nodos do *workflow* correspondente. Na Figura 7 é apresentado o diagrama de casos de uso da área do sistema disponível ao gerente.

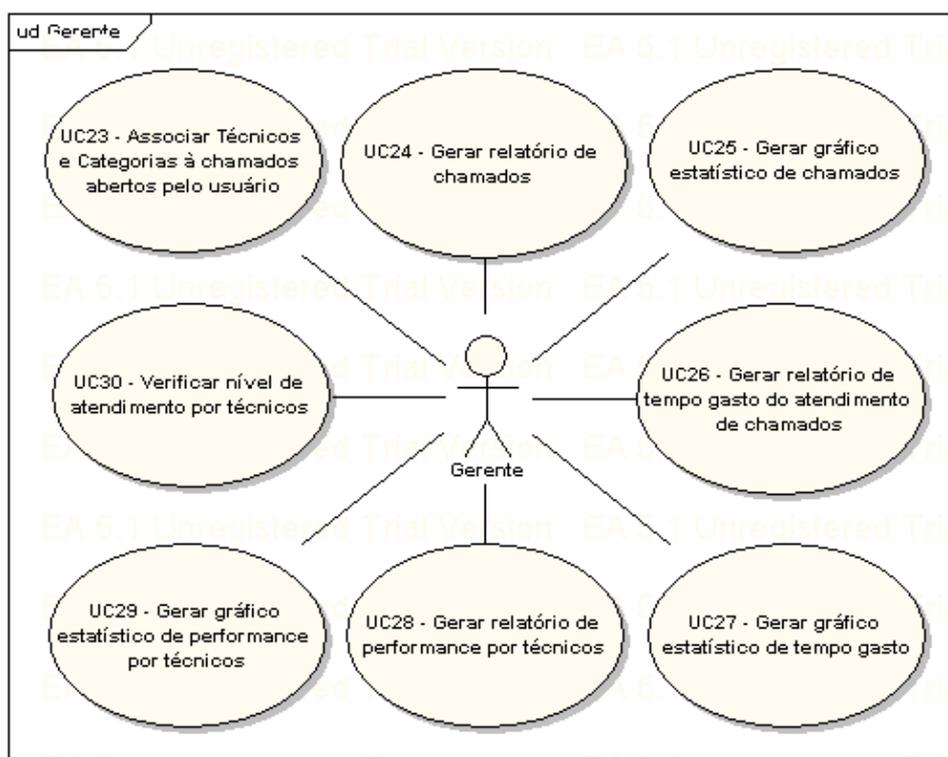


Figura 7 - Diagrama de caso de uso do gerente

A área disponível ao gerente define-se pelos seguintes casos de uso:

- a) associar técnicos e categorias à chamados abertos pelo usuário: permite que o gerente responsável pelo serviço associe um técnico e uma categoria para os chamados abertos pelos usuários;

- b) gerar relatório de chamados: permite que o gerente emita um relatório de chamados dentro e fora do padrão de tempo do *workflow*;
- c) gerar gráfico estatístico de chamados: permite que o gerente verifique através de gráficos os chamados dentro e fora do padrão de tempo do *workflow*;
- d) gerar relatório de tempo gasto do atendimento de chamados: permite que o gerente emita um relatório por técnicos ou serviços de tempo gasto no atendimento de chamados;
- e) gerar gráfico estatístico de tempo gasto: permite que o gerente verifique através de gráficos o tempo gasto no atendimento de chamados;
- f) gerar relatório de performance por técnicos: permite que o gerente emita um relatório de performance por técnico;
- g) gerar gráfico estatístico de performance por técnicos: permite que o gerente verifique através de gráficos a performance dos técnicos;
- h) verificar nível de atendimento por técnicos: permite que o gerente verifique o nível de atendimento classificado pelos usuários de chamados.

3.2.2 Diagrama de Estados

O diagrama de estados representa as mudanças de estado de um objeto, o qual muda de estado a cada vez que acontece um evento interno ou externo do sistema. Durante a mudança de um estado para outro, o objeto realiza determinadas ações dentro do sistema.

Na Figura 8 está sendo representado pelo diagrama de estado do *workflow* que será aplicado no registro e atendimento de chamados abertos no sistema.

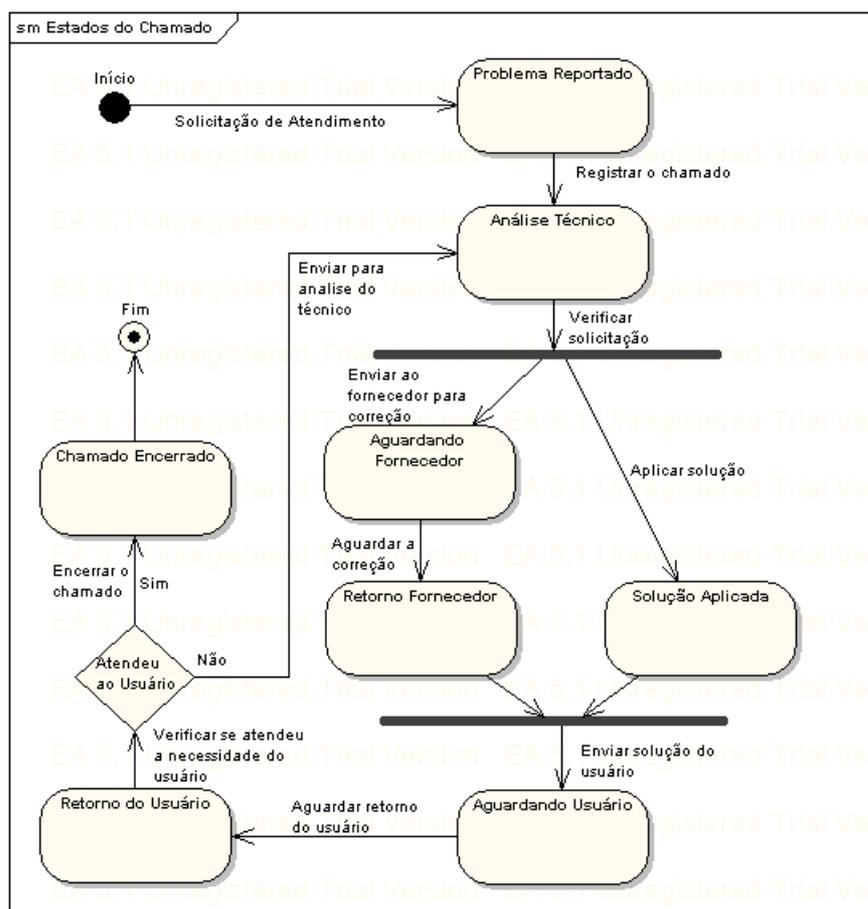


Figura 8 - Diagrama de estados do ciclo de vida do chamado

3.2.3 Diagrama de Atividades

O diagrama de atividades representa a seqüência de atividades, com suporte para comportamento condicional ou paralelo, juntamente com a sincronização. A seguir serão apresentados os diagramas de atividades da abertura de chamados pelo usuário e abertura de chamados pelo técnico.

Na Figura 9 está sendo representado o diagrama de atividades o processo de abertura de chamados pelo usuário. Este processo é efetuado via sistema, com seu *login* e senha de acesso.

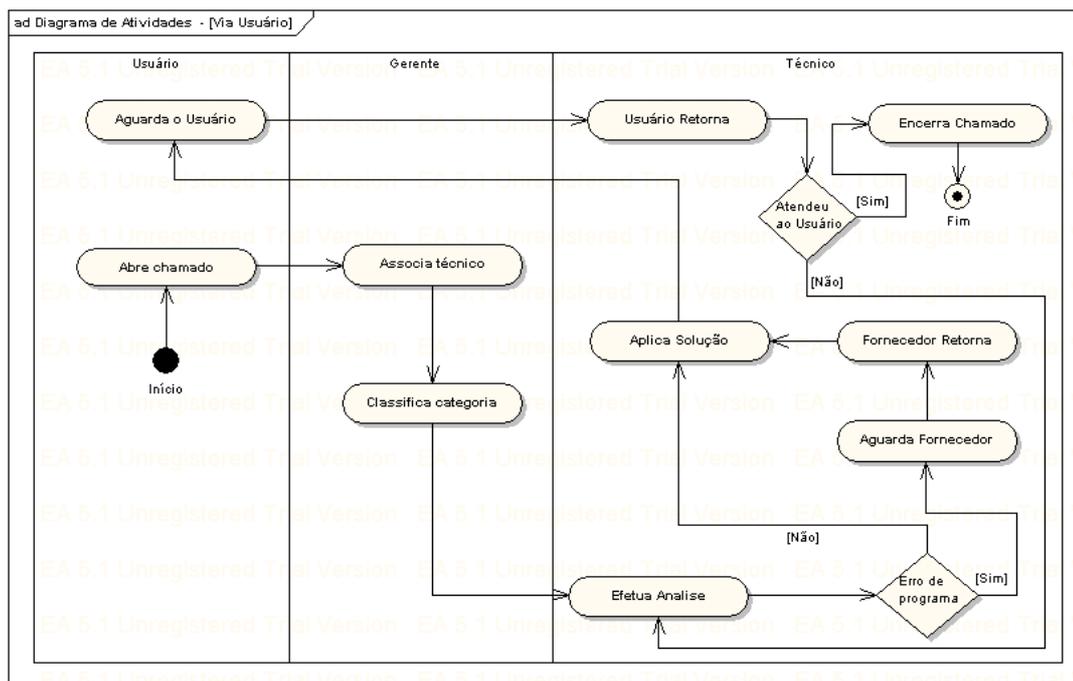


Figura 9 - Diagrama de atividades da abertura de chamado pelo usuário

Na Figura 10 está sendo representado o diagrama de atividades o processo de abertura de chamados pelo técnico. Este processo é efetuado via sistema, onde a solicitação de abertura parte do usuário, mas pelo telefone.

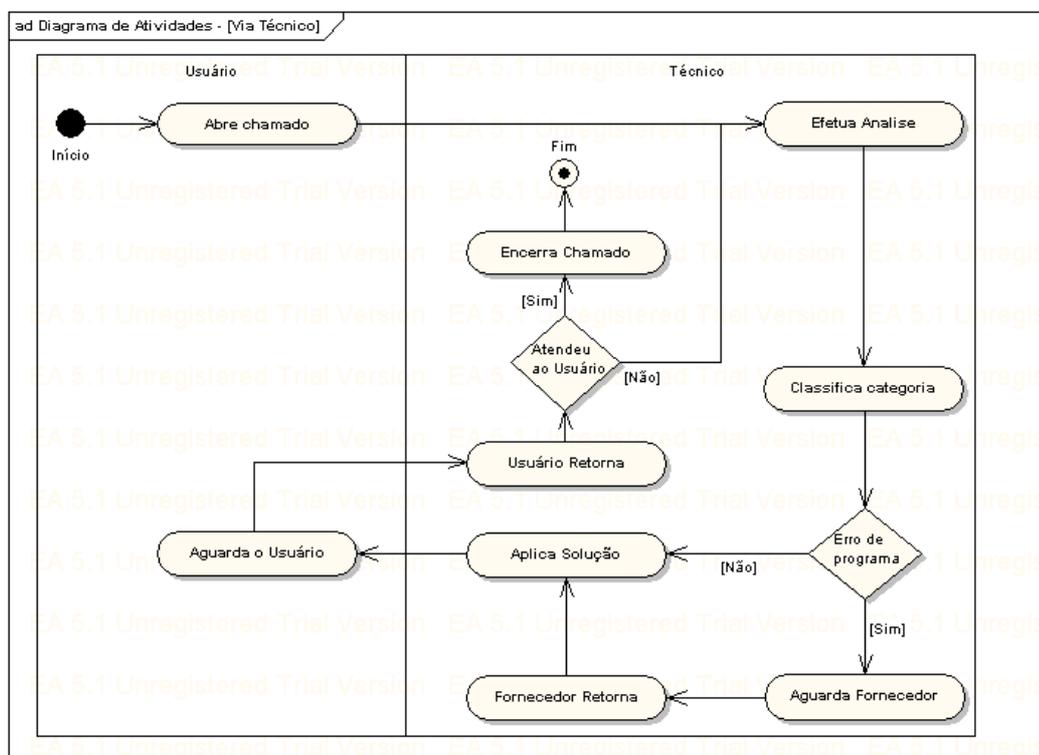


Figura 10 - Diagrama de atividades da abertura de chamado pelo técnico

3.2.4 Diagrama de Classe

Em modelos orientados a objetos, o diagrama de classes é a representação do domínio do negócio. As classes, por sua vez, servem para representar o que será manipulado pelo sistema. São nos diagramas de classes que estão representados os relacionamentos como associações, composições, agregações e especializações entre as classes, com suas multiplicidades, papéis e regras. Este trabalho, por seguir os padrões da UML, utiliza este tipo de diagrama para representar suas estruturas de dados. As classes chamados e ocorrências de chamados são as principais do sistema, que apresenta todos o controle das atividades realizadas no registro de chamados, as classes cliente, área, serviço e versão do serviço compõem o cadastro básico para o registro de chamados e também se pode dizer que estas classes estão compoendo o registro de chamados. As classes de categoria de *workflow* e log de chamados estourados apresentam o controle do processo de *workflow* aplicado nos chamados. A classe usuário e tipo de usuário são as que permitem o controle de usuários que irão utilizar o sistema e por fim, a classe de nível de atendimento está ligada à avaliação do atendimento de chamados. Na Figura 11 está sendo representado o diagrama de classes modelado durante a especificação do sistema.

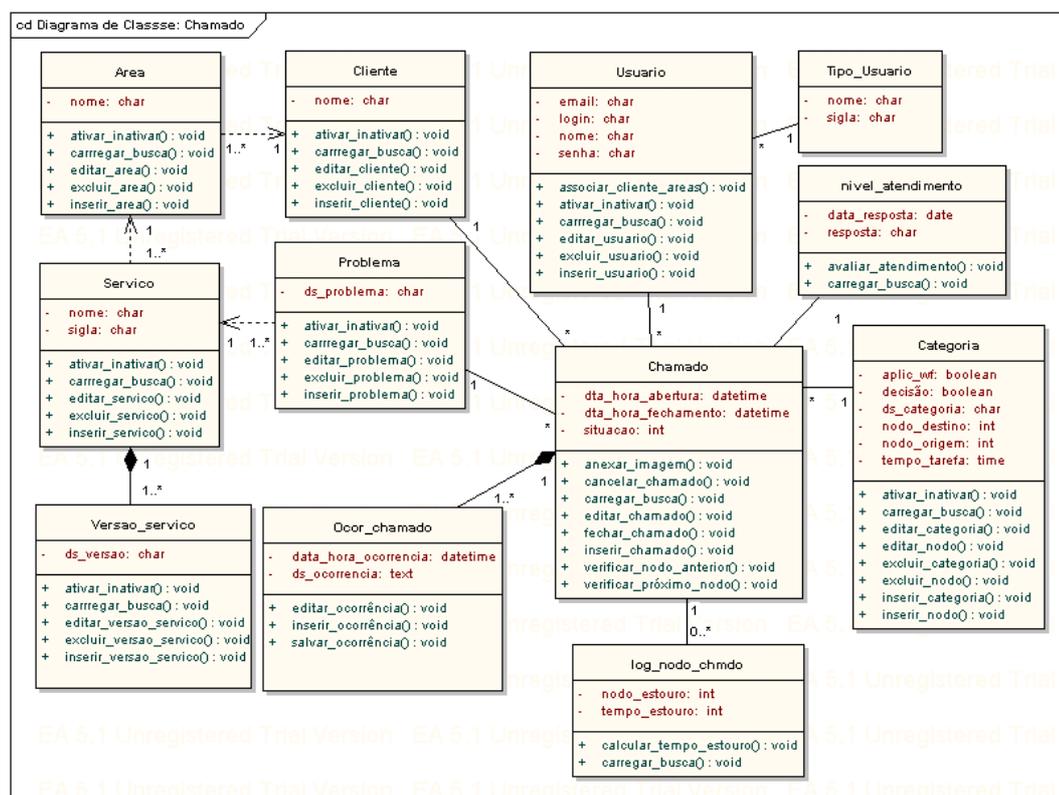


Figura 11 - Diagrama de classe do sistema

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentados os tópicos relacionados à implementação do sistema, apresentando as técnicas e ferramentas utilizadas. Além disso, também são listados alguns códigos fontes do sistema. Por fim é apresentada a operacionalidade do sistema, comentando as telas de interface com os usuários mais relevantes do sistema.

3.3.1 Técnicas e Ferramentas Utilizadas

O sistema foi implementado utilizando a linguagem de programação *Personal Home Page* (PHP), acessando uma base de dados MySQL. O conceito para desenvolvimento de aplicativos *web* AJAX foi utilizado em todo o sistema, o que permite a apresentação do conteúdo do sistema usuários, sem que exista requisições de atualizações durante a utilização. Para a geração dos gráficos estatísticos disponíveis no sistema, utilizou-se da biblioteca *opensource* baaChart. Para a implementação da shell foi utilizado o sistema operacional Linux, o qual foi acessado através de uma VMware. A seguir serão detalhadas as técnicas e ferramentas mais relevantes utilizadas durante a implementação.

3.3.1.1 PHP

Para o desenvolvimento da interface responsável pelo registro chamados no sistema foi utilizada a linguagem PHP. PHP é uma sinergia entre linguagem de programação e servidor de aplicações. É possível programar em PHP definindo variáveis, criando funções, realizando laços como em qualquer outra linguagem de programação, com a vantagem de interagir com o mundo *web*, criando verdadeiras fontes de informação ao invés de páginas estáticas (SOARES, 2000, p. 5).

3.3.1.2 MySQL

Para o armazenamento dos registros do sistema, foi utilizada a ferramenta de gerenciamento de dados MySQL. O MySQL é um gerenciador de banco de dados que utiliza como interface a linguagem *Structured Query Language* (SQL) sendo atualmente um dos mais populares, possuindo cerca de 4 milhões de instalações no mundo (MYSQL, 2006).

3.3.1.3 VMware Server e Fedora 6

Como houve a necessidade de utilizar a plataforma Linux para que a *shell* fosse executada, utilizou-se o aplicativo VMware Server para carregar uma máquina virtual dentro do sistema operacional Windows. O VMware é um software que cria máquinas virtuais que simulam um computador completo dentro de uma janela ou em tela cheia, permitindo instalar praticamente qualquer sistema operacional (MORIMOTO, 2007).

Visando o conceito de máquinas virtuais, foi utilizada uma distribuição gratuita do Linux, cuja versão é o *Fedora 6*. A mesma é utilizada para o desenvolvimento e execução da *shell* que irá controlar o processo de envio de *e-mail* para os atores envolvidos no registro de chamados.

Fedora é um projeto aberto patrocinado pela REDHAT e toda contribuição e desenvolvimento feito pode eventualmente ser agregada a nova versão. O trabalho em conjunto com a comunidade Linux é construir um sistema operacional mais aberto e com uma maior possibilidade de desenvolvimento onde todos podem acompanhar o que está e será feito em um fórum público. A última versão do *Fedora* foi lançada em outubro de 2006, sendo ela a versão 6 (LINUX, 2006).

3.3.1.4 SHELL

Para o monitoramento dos chamados registrados no sistema, com o objetivo de informar via *e-mail* os atores Usuários e Gerentes, necessitou-se do desenvolvimento de uma *shell*. A mesma irá monitorar o posicionamento dos nodos de *workflow*, tais como: "Abertura de chamado", "Encerramento de chamado", "Aguardando retorno de solução aplicada" e

"Associação de categoria e técnico responsável". A *shell* foi desenvolvida em um único arquivo designado "tcc.sh" e está contido dentro da seguinte estrutura do Linux "/etc/tcc.sh". Para melhor entendimento e visualização da *shell* dividiu-se a mesma em quatro blocos, conforme descrito a seguir.

O Quadro 08 apresenta a rotina da *shell* responsável pelo envio das informações de abertura dos chamados registrados no sistema de *help-desk*.

```
#####
# Shell....: tcc.sh #
# Acadêmico: Fabio Ernandes Simon #
# Sistema..: Controle de Help-Desk #
# Função...: Verificar requisições pendentes dos atores #
#           Usuário e Gerente e enviar e-mail informando #
#####
set -x
#####
# Variáveis de conexão com o banco de dados #
#####

host_bd="164.164.164.101"
name_bd="tcc"
user_bd="root2"

#####
# Abertura de chamado #
#####
mysql ${name_bd} -u ${user_bd} -h ${host_bd} -s -e "select
concat(cha.id_chamado,'#',usu.email,'#',usu.nome,'#') from chamados cha,
usuarios usu where cha.id_usuario = usu.id_usuario and cha.situacao = 1
and cha.emLABE = 'N';" > /etc/tmp/usu001.txt
while read -r reg001
do
id_chamado=`echo $reg001 | cut -d"#" -f 1`
email=`echo $reg001 | cut -d"#" -f 2`
nome_usu=`echo $reg001 | cut -d"#" -f 3`
mail -s 'Registro de chamado - [Novo]' $email << EOF

Sr(a) $nome_usu

Foi registrado em nosso sistema o chamado número $id_chamado referente sua
solicitação.

Para visualizar o acompanhamento do chamado, acesse o sistema e verifique
o status
através da listagem de chamados que será apresentada após o devido login.

Favor não responder o e-mail, pois o mesmo é gerado automaticamente pelo
sistema de Help-Desk
EOF
mysql ${name_bd} -u ${user_bd} -h ${host_bd} -s -e "update chamados set
emLABE = 'S' where id_chamado = $id_chamado;"
done < /etc/tmp/usu001.txt
set +x
```

Quadro 08 – Rotina da shell - “Abertura de Chamados”

O Quadro 09 apresenta a rotina da *shell* responsável pelo envio de informações

aguardando o retorno do usuário no sistema de *help-desk*.

```
# Aguardando o usuario #
mysql ${name_bd} -u ${user_bd} -h ${host_bd} -s -e "select
concat(cha.id_chamado,'#',usu.email,'#',usu.nome,'#') from chamados cha,
usuarios usu where cha.id_usuario = usu.id_usuario and cha.situacao = 0
and cha.emLAU = 'N';" > /etc/tmp/usu002.txt
while read -r reg001
do
id_chamado=`echo $reg001 | cut -d"#" -f 1`
email=`echo $reg001 | cut -d"#" -f 2`
nome_usu=`echo $reg001 | cut -d"#" -f 3`
mail -s 'Posicionamento de chamado - [Aguardando]' $email << EOF

Sr(a) $nome_usu

O chamado $id_chamado aberto em seu nome está aguardando um retorno quanto
a solução aplicada.

Entre em contato para retornar um feedback

Favor não responder o e-mail, pois o mesmo é gerado automaticamente pelo
sistema de Help-Desk
EOF
mysql ${name_bd} -u ${user_bd} -h ${host_bd} -s -e "update chamados set
emLAU = 'S' where id_chamado = $id_chamado;"
done < /etc/tmp/usu002.txt
```

Quadro 09 – Rotina da shell - “Aguardando o Usuário”

O Quadro 10 apresenta a rotina da *shell* responsável pelo envio de informações de fechamento de chamados no sistema de *help-desk*.

```
# Fechamento do chamado #
mysql ${name_bd} -u ${user_bd} -h ${host_bd} -s -e "select
concat(cha.id_chamado,'#',usu.email,'#',usu.nome,'#') from chamados cha,
usuarios usu where cha.id_usuario = usu.id_usuario and cha.situacao = 0
and cha.emLENC = 'N';" > /etc/tmp/usu003.txt
while read -r reg001
do
id_chamado=`echo $reg001 | cut -d"#" -f 1`
email=`echo $reg001 | cut -d"#" -f 2`
nome_usu=`echo $reg001 | cut -d"#" -f 3`
mail -s 'Encerramento de chamado' $email << EOF

Sr(a) $nome_usu

O chamado número $id_chamado foi encerrado.

Por favor, acesse o sistema de Help-Desk e classifique o nível de
atendimento.

Favor não responder o e-mail, pois o mesmo é gerado automaticamente pelo
sistema de Help-Desk
EOF
mysql ${name_bd} -u ${user_bd} -h ${host_bd} -s -e "update chamados set
emLENC = 'S' where id_chamado = $id_chamado;"
done < /etc/tmp/usu003.txt
```

Quadro 10 – Rotina da shell - “Fechamento de Chamados”

O Quadro 11 apresenta a rotina da *shell* responsável pelo envio de informações para

que o gerente associe a categoria e o técnico do chamado no sistema de *help-desk*.

```
# Gerente associar categoria e tecnico #
mysql ${name_bd} -u ${user_bd} -h ${host_bd} -s -e "select
concat(cha.id_chamado,'#',usu.email,'#',usu.nome,'#') from chamados cha,
usuarios usu, servicos ser where cha.id_servico = ser.id_servico and
ser.id_gerente = usu.id_usuario and cha.situacao = 1
and cha.emlCTG = 'N';" > /etc/tmp/usu004.txt
while read -r reg001
do
id_chamado=`echo $reg001 | cut -d"#" -f 1`
email=`echo $reg001 | cut -d"#" -f 2`
nome_usu=`echo $reg001 | cut -d"#" -f 3`
mail -s 'Chamado pendente - [Classificar]' $email << EOF

Sr(a) $nome_usu

O chamado número $id_chamado foi registrado em nosso sistema de Help-Desk
e será necessário associar a categoria e o técnico.

Favor não responder o e-mail, pois o mesmo é gerado automaticamente pelo
sistema de Help-Desk
EOF
done < /etc/tmp/usu004.txt
```

Quadro 11 – Rotina da shell - “Gerente associar categoria e técnico”

3.3.1.4.1 Manipulação do Contrab

Para o controle e manipulação de execução da *shell*, foi desenvolvida uma rotina que irá carregar a última programação de execução diretamente da crontab do sistema operacional Linux. A nova escolha feita na tela de agendamento gera um novo arquivo crontab colocando como o dono do arquivo o usuário “root”. Apenas o administrador do sistema pode efetuar a escolha de uma nova programação e a mesma apenas pode ser efetuada para execução em minutos. Para acessar o crontab no sistema operacional Linux foi utilizada a VMware, que apresenta outro sistema operacional virtualmente. No quadro 12 é apresentado o código responsável pela chamada da rotina de manipulação do arquivo e geração da nova programação de execução.

```

<?php
include_once('../././classes/inc.conecta_bancoAJAX.php');
include_once('../././classes/class.paginacao.php');
$divid=$_SESSION['divid'];
$pagina="agendamento";
$msg = "";
//se foi clicado no botão para alterar
if(isset($_REQUEST['acao'])):
    if(isset($_REQUEST['minutos'])&&($_REQUEST['minutos']>0)):
        //exclui o arquivo pra gerar um novo.
        unlink(SYSTEM_CRONTAB_DIR);
        //conteudo a ser gravado no novo arquivo
        $gravar = "00-59/" . $_REQUEST['minutos'] . " * * * *
/etc/tcc.sh";
        $msg = $ado->manipulaarquivo
(SYSTEM_CRONTAB_DIR, 'w', $gravar, false);
    else: //minutos>0
        $msg = "Escolha um intervalo!";
    endif; //minutos>0
endif; //acao>0
/*INICIO recuperando o intervalo de execução atual*/
$minutos = $ado->manipulaarquivo(SYSTEM_CRONTAB_DIR, 'rb', '', false);
$minutos = str_replace('00-59/', '', str_replace(' * * * *
/etc/tcc.sh', '', $minutos));
/*FIM recuperando o intervalo de execução atual*/
?>

```

Quadro 12 – Rotina de manipulação e alteração da programação

3.3.1.5 AJAX

Para carregar o conteúdo das páginas sem a necessidade de fazer a atualização de toda a página, foi criada uma função com o objetivo de carregar o conteúdo passando como parâmetro a URL e o local onde este deve ser aberto, ou seja, o objeto XMLHttpRequest, busca o conteúdo XML ou HTML de uma página qualquer, com base na URL informada e exibe estes dados na *tag div* do HTML informado, conforme é demonstrado no Quadro 13. Com AJAX, em vez de enviar a requisição, a interface lança uma chamada de JavaScript para o mecanismo AJAX que se encontra do lado do cliente. Este por sua vez envia a requisição HTTP para o servidor, através do objeto XMLHttpRequest. Enquanto o servidor processa e retorna os dados da requisição, a interface do usuário fica disponível para que este continue utilizando o sistema normalmente (as requisições são processadas de forma assíncrona). Assim que o servidor retorna o resultado (geralmente em XML), o mecanismo AJAX, utilizando Javascript e DOM, atualiza apenas os elementos do documento HTML que devem ser alterados, não precisando recarregar toda a página novamente (Morujão, A. et al, 2005, p. 2-3).

```

function carregapagina(url,local){
    //declarando o objeto
    objxmlhttp=montar_XMLHttp();
    //Todas as páginas a carregar ficam na pasta OBJETOS/CHAMADOS para
    manter a estrutura de pastas funcionando
    url = "../..objetos/AJAX/"+url;

    /*abrindo o objeto com os parametros Método:GET, caminho:url, */
    url = GeraCRC(url);
    //cancelar.innerHTML=url;

    objxmlhttp.open("POST", url,false);
    objxmlhttp.onreadystatechange=function(){
        if(objxmlhttp.readyState==4){
            eval(local).innerHTML=objxmlhttp.responseText;
            //alert(objxmlhttp.responseText);
        }
    }
    objxmlhttp.send(null);
}

```

Quadro 13 – Rotina de atualização do conteúdo das páginas

3.3.1.6 Geração dos gráficos com a biblioteca baaChart

A biblioteca baaChart foi utilizada para a geração dos gráficos estatísticos utilizados pelo gerente do sistema. BaaChart é uma biblioteca com suporte de orientação a objetos que apresenta seu código totalmente aberto e disponibilizado na *internet*, que utiliza a biblioteca GD2 para imagens e está disponível a partir do PHP 4.3. Por ter sido escrita totalmente em PHP, pode ser utilizada para confeccionar vários tipos de gráficos, tais como gráficos em barras, de pizza, entre outros. Em suma, pode-se dizer que confeccionar gráfico com esta biblioteca requer um mínimo de código e atribui um controle muito simples (BAACHART, 2006). No Quadro 14 é apresentado o código fonte gerado para a confecção do gráfico de chamados dentro e fora do padrão de tempo.

```

$titulo = "Chamados dentro e fora do padrão de tempo";
$ArrNPrazo = array();
$ArrFPrazo = array();

$sql = "select distinct(cha.id_cliente),cli.nome from clientes cli,
chamados cha WHERE cli.id_cliente = cha.id_cliente AND cha.id_chamado in
('.$_REQUEST['noprazo'].','.$_REQUEST['foraprazo'].') order by cli.nome";
$rs = $DB->Execute($sql);
while(!$rs->EOF):
    $ArrNPrazo[$rs->fields['nome']] = 0;
    $ArrFPrazo[$rs->fields['nome']] = 0;
    $rs->MoveNext();
endwhile;
$rs = $DB->Execute($sql);
while(!$rs->EOF):
    //OKS
    $sql = "select count(cha.id_cliente) as total from chamados cha WHERE
cha.id_chamado in ('.$_REQUEST['noprazo'].') AND id_cliente = ".$rs->
fields['id_cliente'];
    $rsOK = $DB->Execute($sql);
    if($rsOK->fields['total'] > 0):
        $ArrNPrazo[$rs->fields['nome']] = $rsOK->fields['total'];
        if($max < $rsOK->fields['total']):
            $max = $rsOK->fields['total'];
        endif;
    endif;
    //NOKS
    $sql = "Select count(cha.id_cliente) as total from chamados cha WHERE
cha.id_chamado in ('.$_REQUEST['foraprazo'].') AND id_cliente = ".$rs->
fields['id_cliente']. """;
    $rsNOK = $DB->Execute($sql);
    if($rsNOK->fields['total'] > 0):
        $ArrFPrazo[$rs->fields['nome']] = $rsNOK->fields['total'];
        if($max < $rsNOK->fields['total']):
            $max = $rsNOK->fields['total'];
        endif;
    endif;
    $rs->MoveNext();
endwhile;
foreach($ArrFPrazo as $cliente => $chamados):
    $Grafico->addDataSeries('C', 0, $ArrNPrazo[$cliente].", $chamados", $cliente);
endforeach;
$Grafico->setXLabels("No prazo, Fora do Prazo");
$Grafico->setYAxis("Qtd - Chamados", 0, floor($max*1.2)+1, floor($max*1.2)+1, 1);
$Grafico->setXAxis("Situação", 1);
break;

```

Quadro 14 – Código fonte do gráfico de chamados dentro e fora do padrão de tempo

3.3.2 Operacionalidade do Sistema

A operacionalidade do sistema será apresentada através da simulação de casos de utilização do sistema, ou seja, serão detalhadas as operações necessárias para a utilização dos recursos no sistema de *help-desk*.

Na Figura 12 é apresentada a tela de *login* do sistema, que encaminhará o usuário para diferentes áreas do sistema dependendo do tipo de usuário.

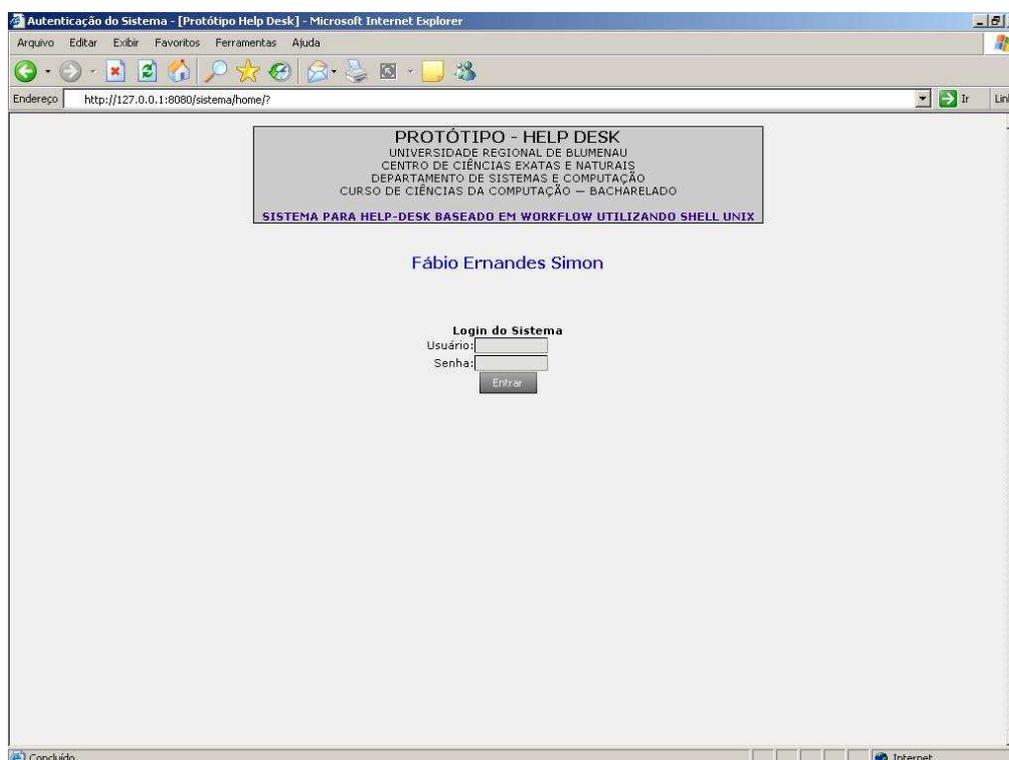


Figura 12 – Tela de login do sistema

Os diferentes tipos de usuários que utilizam o sistema de *help-desk* são apresentados no Quadro 15.

Tipo de Usuário
Usuário de atendimento
Técnico
Gerente
Administrador do sistema

Quadro 15 – Tipos de usuário no sistema de help-desk

3.3.2.1 Login de Administrador

Caso o usuário seja Administrador do sistema, após o *login* ele será encaminhado para a tela apresentada na Figura 13.

Para o usuário administrador é apresentado do lado esquerdo da tela o menu a ele destinado. Do lado direito da tela é exibida a visualização do conteúdo de cada menu selecionado. Neste momento o administrador já está listando os clientes cadastrados.

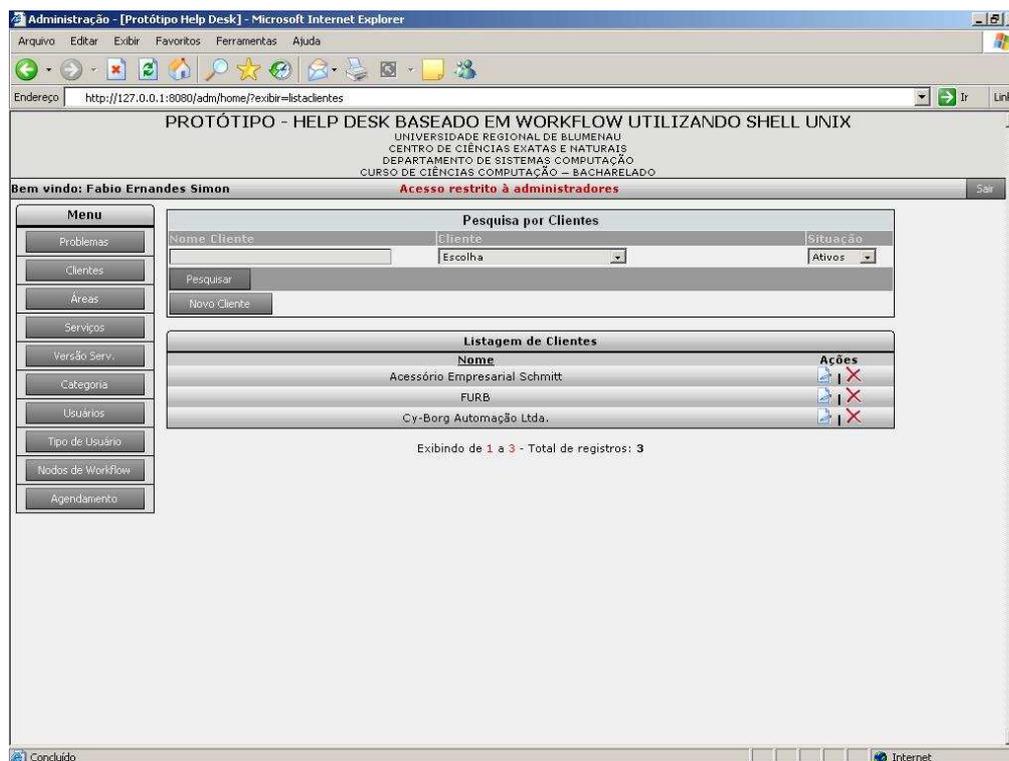


Figura 13 – Tela principal do administrador do sistema

Nas listagens disponíveis ao administrador são apresentadas algumas opções, assim como pesquisa correspondente ao menu seleciona, inclusão de um novo registro, representado pelo botão abaixo da área de pesquisa. Os ícones  e  são comuns em todas as listagens disponíveis ao administrador e servem respectivamente para editar e excluir um registro listado.

Quando ocorrer o evento de edição de um registro da listagem, será apresentada uma nova janela apresentando os dados editados, juntamente com os botões Salvar e Voltar e servem respectivamente para salvar as alterações efetuadas no registro editado e voltar à listagem sem salvar a edição efetuada.

Nos menus Clientes, Áreas, Serviços, Versão do Serviço, Categoria, Problemas e Usuários, quando for selecionada a opção de edição o componente *check* representado com a legenda Ativo, será comum para todas e terá a função de decisão para que o registro fica inativo ou ativo para visualizações em todas as áreas do sistema. Na Figura 14 será apresentada a edição de um registro da listagem de clientes.

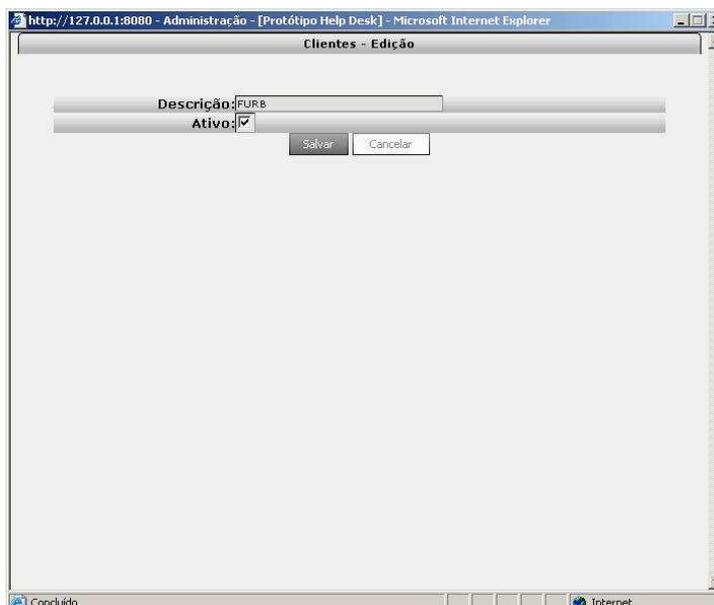


Figura 14 – Tela de edição de um registro de cliente

Quando ocorrer o evento de inclusão de um registro através do botão Novo, será apresentada uma nova janela, contendo os respectivos campos devidamente não preenchidos da opção selecionada, juntamente com os botões Salvar e Voltar e servem respectivamente para salvar as alterações efetuadas no novo registro e voltar à listagem sem salvar a inclusão. Na Figura 15 será apresentada a inclusão de um novo registro.

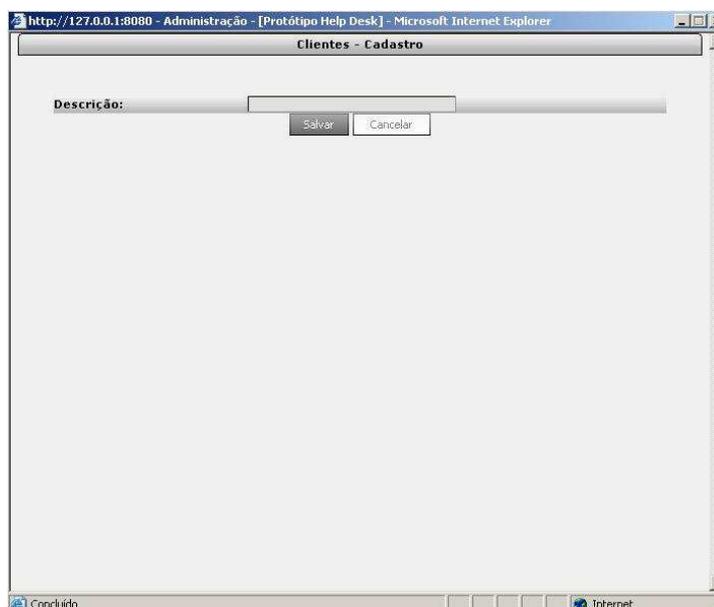


Figura 15 – Tela de inclusão de um novo cliente

Para os menus Áreas, Serviços, Versão de Serviços e Problemas ocorre à vinculação e informações. A vinculação corresponde a um registro ser filho de outro anteriormente incluído em outro menu.

A inclusão de um registro que possui vínculo deverá ser escolhida através do

componente *combobox* o vínculo correspondente. Na Figura 16 será apresentada a inclusão de um novo registro para o menu de Áreas, a qual será devidamente vinculada a um Cliente anteriormente cadastrado.

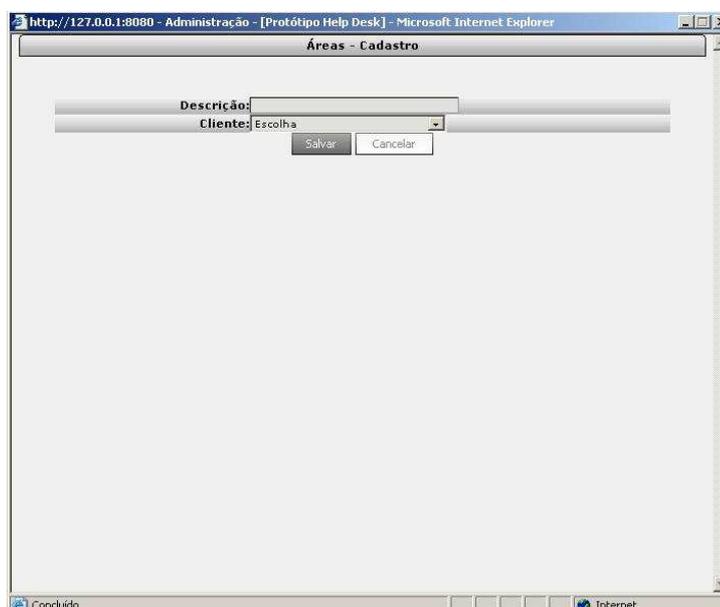


Figura 16 – Tela de inclusão de uma área para um cliente

Quando ocorrer o evento de edição de um registro da listagem, cujo possuir vínculo com outro registro anteriormente incluído, será apresentada uma nova janela apresentando os dados editados podendo ser novamente alterada, o vínculo com outro registro, juntamente com os botões Salvar e Voltar e servem respectivamente para salvar as alterações efetuadas no registro editado e voltar a listagem sem salvar a edição efetuada.

Na Figura 17 será apresentada a edição de um registro do menu áreas da listagem de clientes que possui vínculo com o cliente anteriormente incluído.

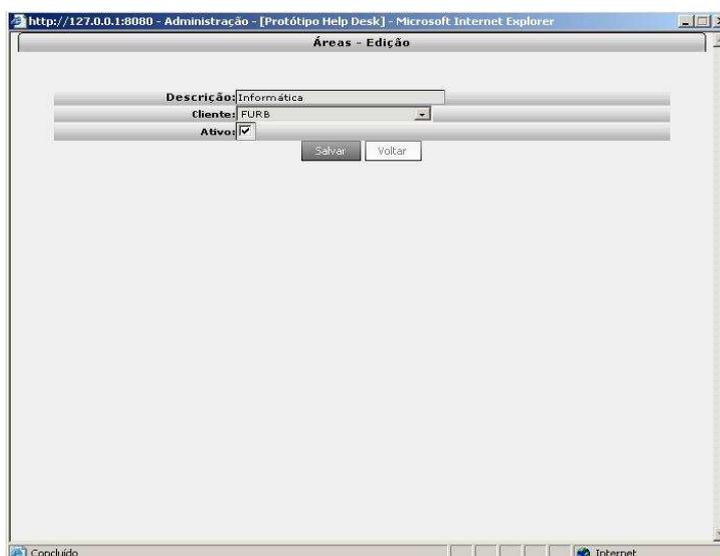
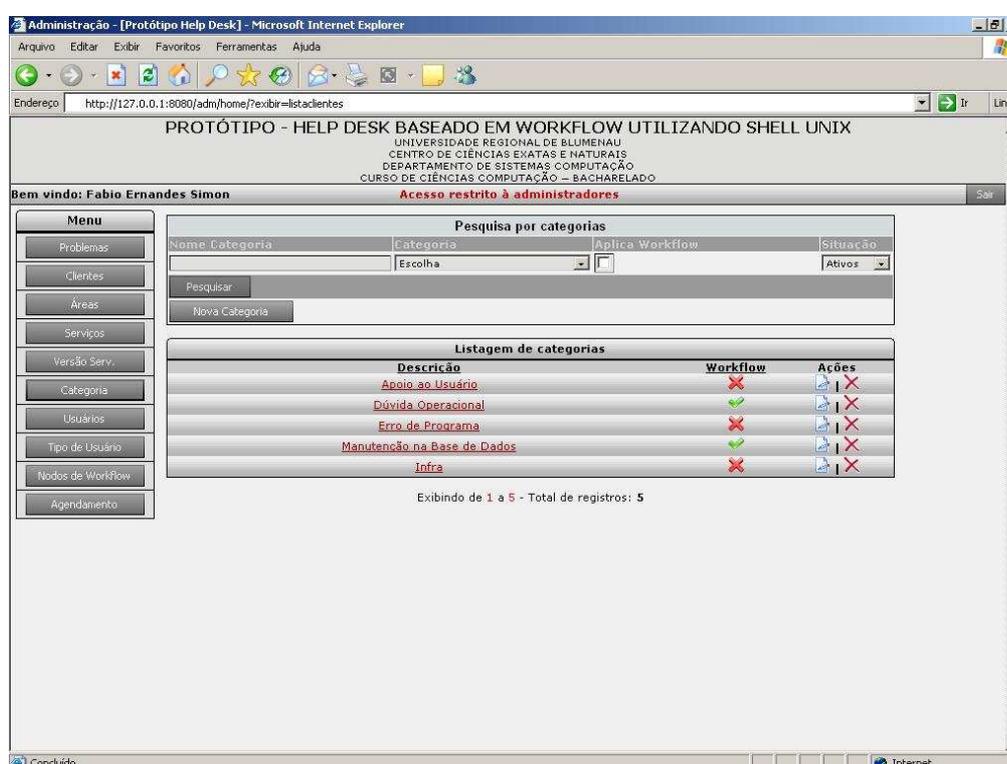


Figura 17 – Tela de edição de área de um cliente

A seguir será apresentado o menu de Categorias, cujo não possuem o mesmo padrão de funcionamento dos demais apresentados. Quando o menu Categoria é selecionado, é apresentada a listagem de categorias e na própria listagem são apresentados os ícones  e  servem respectivamente para informar se a categoria listada irá ter a aplicabilidade do *workflow* em chamados registrados no sistema ou não.

Ainda na mesma tela o usuário terá a opção de editar os nodos da categoria listada, clicando no *link* respectivo a categoria a ser editada. Os nodos referem-se aos passos do *workflow* dos chamados registrados no sistema. Na Figura 18 será apresentada a listagem de categorias.



The screenshot shows a web browser window with the URL `http://127.0.0.1:8080/adm/home/?exibir=listadientes`. The page title is "PROTÓTIPO - HELP DESK BASEADO EM WORKFLOW UTILIZANDO SHELL UNIX". The user is logged in as "Fabio Ernandes Simon" with restricted access to administrators. A sidebar menu on the left includes options like "Problemas", "Clientes", "Áreas", "Serviços", "Versão Serv.", "Categoria", "Usuários", "Tipo de Usuário", "Nodos de Workflow", and "Agendamento". The main content area has a search section for categories and a table titled "Listagem de categorias".

Nome Categoria	Categoria	Aplica Workflow	Situação
	Escolha	<input type="checkbox"/>	Ativos
Pesquisar			
Nova Categoria			

Descrição	Workflow	Ações
Apoio ao Usuário		 
Dúvida Operacional		 
Erro de Programa		 
Manutenção na Base de Dados		 
Infra		 

Exibindo de 1 a 5 - Total de registros: 5

Figura 18 – Tela de listagem de categorias

Quando ocorrer o evento de edição de um registro da listagem de categorias, será aberta uma nova janela apresentando os registros/nodos correspondendo à categoria editada, tendo as opções de pesquisa de nodos correspondentes à categoria, listagem dos nodos da categoria e inclusão de um novo nodo para a categoria.

Existem algumas particularidades com relação à legibilidade de informações que se diferem das demais telas do sistema. Na listagem apresentada, poderão existir registros listados em cor preta e cor vermelha, devidamente correspondente à diferenciação de informações, os registros apresentados em cor vermelha representam que no processo de *workflow* de chamados, pode existir apenas um deles, ou seja, estes registros indicam uma decisão no processo de *workflow*. Juntamente com a listagem dos registros da categoria de

workflow selecionada, são apresentadas as setas  e , que são respectivamente utilizadas para movimentar um registro para cima ou para baixo, sem que haja a necessidade de excluir uma seqüência de registros. Na Figura 19 será apresentada a edição dos nodos de uma categoria editada na listagem de categorias.

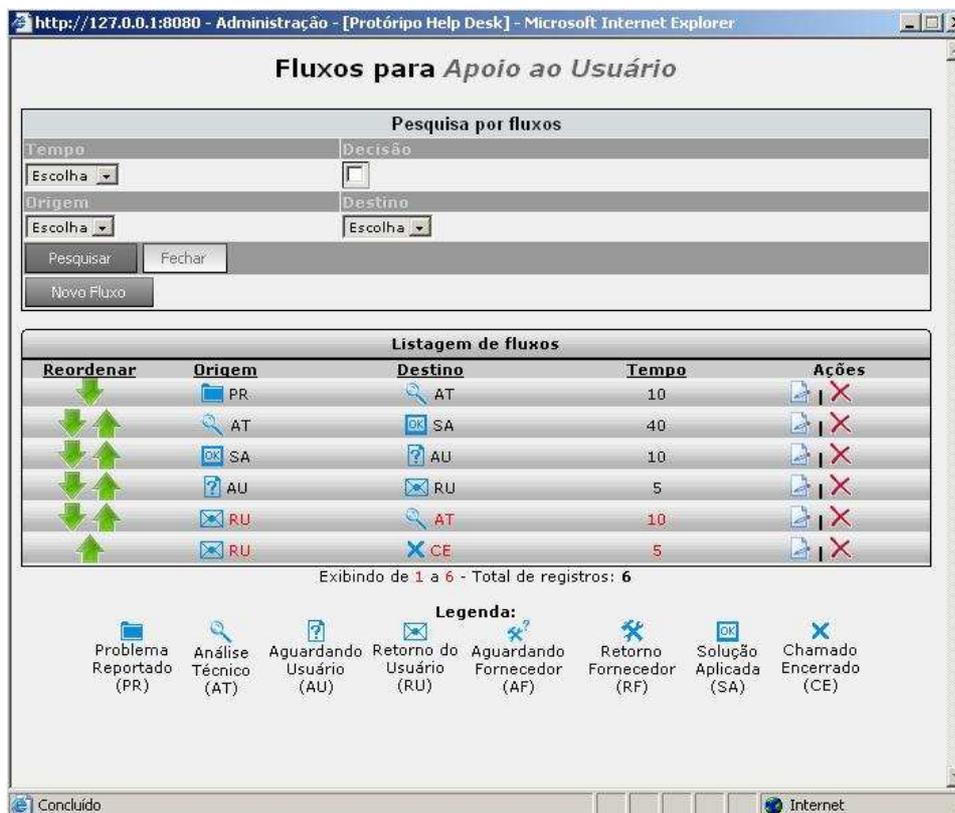


Figura 19 – Tela de listagem dos nodos da categoria

No Quadro 16 são descritos nodos do processo de *workflow*, representados através de ícones.

Ícone	Descrição
	Problema Reportado
	Análise Técnico
	Aguardando Usuário
	Retorno do Usuário
	Aguardando Fornecedor
	Retorno Fornecedor
	Solução Aplicada
	Chamado Encerrado

Quadro 16 – Nodos do processo de workflow

Quando ocorrer o evento de edição de um registro da listagem de um nodo da categoria, será aberta uma nova janela apresentando os dados devidamente preenchidos, com os campos Origem, Destino, Decisão e Tempo disponíveis para alteração, juntamente com os botões Salvar e Voltar e servem respectivamente para salvar as alterações efetuadas no

registro editado e voltar à listagem sem salvar a edição efetuada. Na Figura 20 será apresentada a edição de um nodo de uma categoria editada na listagem de categorias.

Fluxo - Edição

Categoria: Apoio ao Usuário

Ordem: 1

Origem: Problema Reportado

Destino: Análise Técnico

Decisão:

Tempo: 10 Minutos

Salvar Cancelar

Figura 20 – Tela de edição de nodos da categoria

Similar ao processo de edição de um nodo listado será também o processo de inclusão de um novo nodo, acionado através do botão Novo Fluxo, que abrirá respectivamente uma nova janela com os campos Origem, Destino, Decisão e Tempo todos em branco, juntamente com os botões Salvar e Voltar e servem respectivamente para salvar as alterações efetuadas no registro editado e voltar à listagem sem salvar a edição efetuada. Na Figura 21 será apresentada a tela de inclusão de um novo nodo para o fluxo da categoria.

Fluxo - Cadastro

Categoria: Apoio ao Usuário

Ordem:

Origem: Escolha

Destino: Escolha

Decisão:

Tempo: Minutos

Salvar Cancelar

Figura 21 – Tela de inclusão de nodo da categoria

A seguir será apresentado o menu de Usuários do sistema, cujo também não possui o

mesmo padrão de funcionamento dos demais apresentados. Quando o menu Usuário é selecionado, é apresentada a listagem de usuários e na própria listagem é mostrado o tipo de usuário, definido através dos diagramas de caso de uso apresentados anteriormente. Na Figura 22 será apresentada a listagem de usuários.

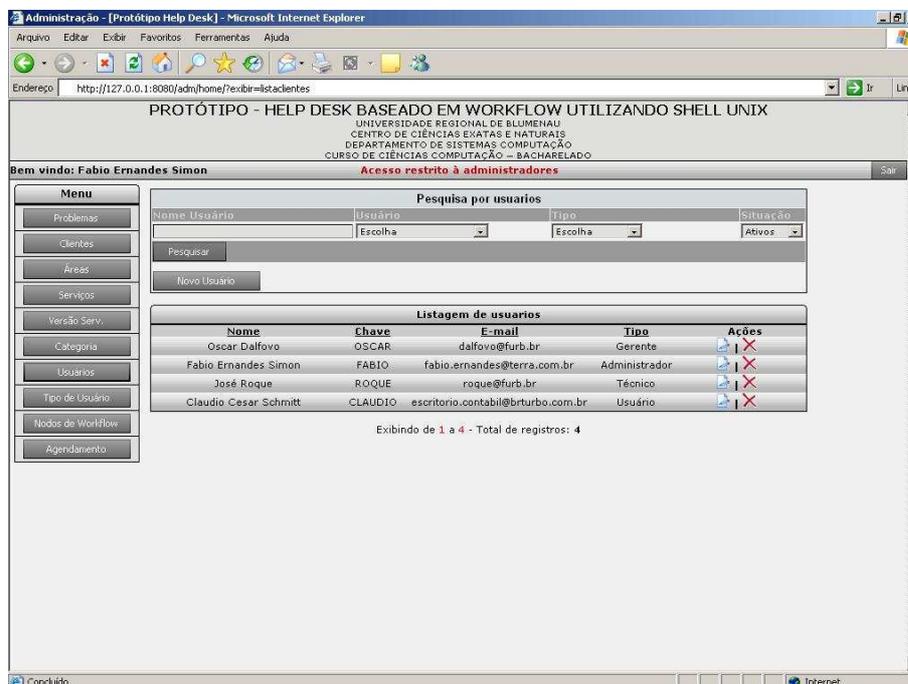


Figura 22 – Tela de listagem de usuários do sistema

Quando ocorrer o evento de solicitação de novo usuário através do botão Novo Usuário, será aberta uma nova janela apresentando os campos usuário, *login*, senha, *email* e tipo de usuário devidamente e branco. Na mesma tela serão carregadas são opções de clientes e áreas dos clientes disponíveis no sistema para associação ao usuário. Na Figura 23 será apresentada a inclusão de um novo usuário do sistema.

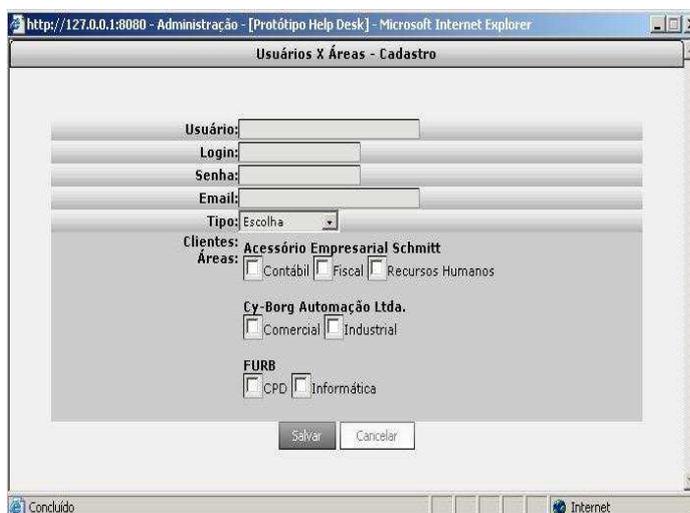


Figura 23 – Tela de inclusão de usuário do sistema

Quando ocorrer o evento de edição de um usuário, será aberta uma nova janela apresentando os campos usuário, *login*, senha, *email* e tipo de usuários devidamente preenchidos e na mesma tela serão carregados são opções de clientes e áreas dos clientes disponíveis para este usuário caso. Na Figura 24 será apresentada a edição de um usuário do sistema.

The screenshot shows a web browser window titled 'http://127.0.0.1:8080 - Administração - [Protótipo Help Desk] - Microsoft Internet Explorer'. The main content area is titled 'Usuários X Áreas - Edição'. It contains a form with the following fields: 'Usuário' with the value 'Oscar Dalfovo', 'Login' with 'OSCAR', 'Senha' (empty), 'E-mail' with 'dalfovo@furb.br', 'Tipo' with a dropdown menu showing 'Gerente', and 'Ativo' with a checked checkbox. Below the form are two buttons: 'Salvar' and 'Cancelar'. The browser's status bar at the bottom shows 'Concluído' and 'Internet'.

Figura 24 – Tela de edição de um usuário do sistema

O controle de visualização dos clientes e áreas dos clientes disponíveis para associação a chave do usuário ocorre através do componente *combobox* com legenda nomeada tipo, a qual irá desabilitar a visualização para os tipos de usuário cujo objetivo principal não é ter clientes e áreas dos clientes associadas a sua chave. No quadro 17 são apresentadas às limitações para o cadastro de usuário do sistema.

Tipo de Usuário	Opções
Administrador	Não habilita a escolha de cliente e áreas dos clientes
Gerente	Não habilita a escolha de cliente e áreas dos clientes
Técnico	Apresenta todos os cliente e áreas dos clientes, não tendo limite de seleção de cliente e áreas dos clientes.
Usuário	Apresenta todos os cliente e áreas dos clientes, mas limitando a seleção de apenas um cliente e duas respectivas áreas.

Quadro 17 – Limitações por tipo de usuário

A seguir será apresentado o menu de Agendamento, cujo também não possui o mesmo

padrão de funcionamento dos demais apresentados. Quando o menu Agendamento é selecionado, é apresentada a última programação de execução efetuada no arquivo Crontab no sistema operacional ativado através da VMware, tendo a opção de alterar a programação de sua execução em minutos através do componente *combobox* com legenda Intervalo de Execução e posteriormente selecionando o botão Alterar. Na Figura 25 será apresentada a tela de agendamento e manipulação do arquivo Crontab.

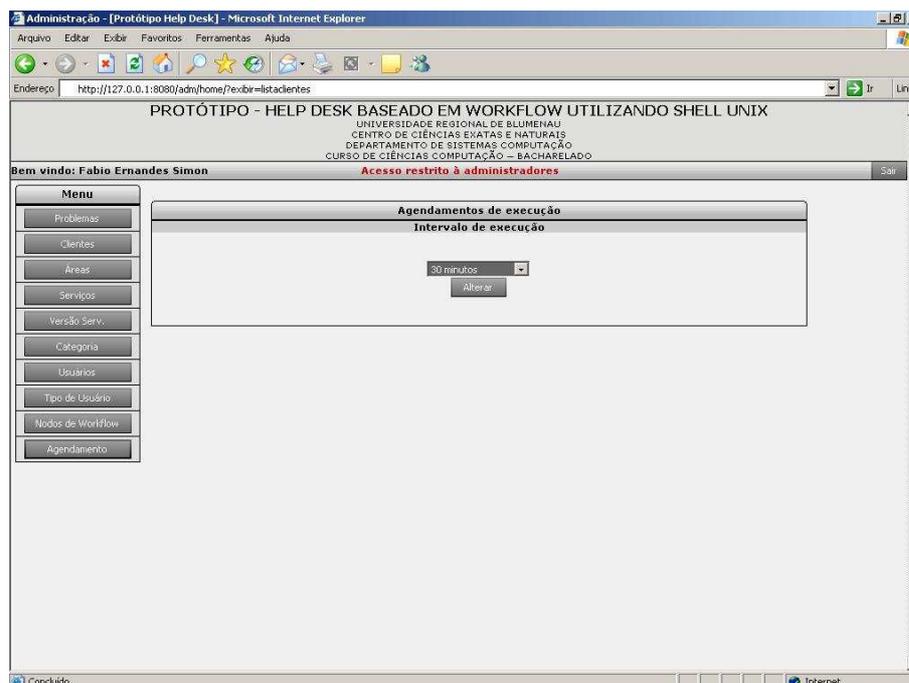
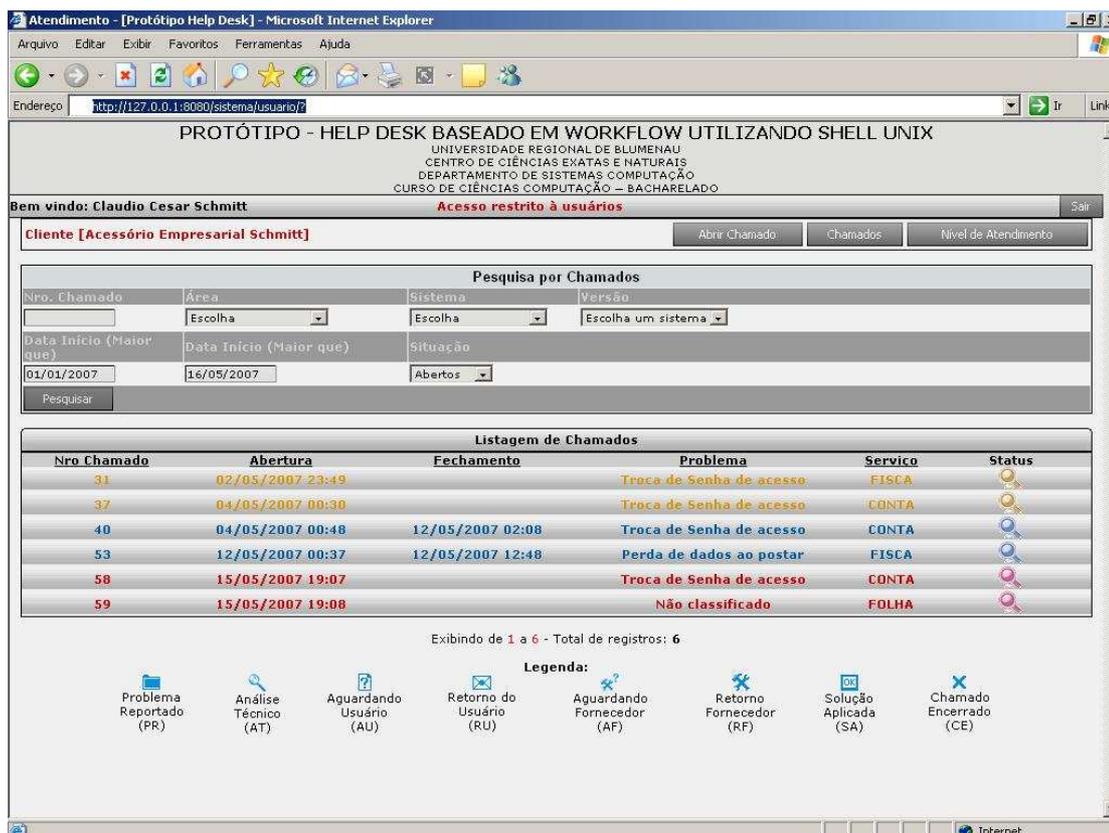


Figura 25 – Tela de agendamento de execução da Shell no Crontab

3.3.2.2 Login de Usuário

Após acesso do usuário, o sistema encaminhará o mesmo para a tela apresentada na Figura 26.

Para o usuário é apresentado na parte superior direita da tela o menu a ele destinado. Na parte inferior da tela é exibida a visualização da listagem dos chamados abertos para sua chave. Por padrão, o sistema carrega inicialmente todos os chamados em se encontram pendentes de solução, ou seja, todos os chamados que ainda não foram fechados.



Atendimento - [Protótipo Help Desk] - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço: http://127.0.0.1:8080/sistema/usuario/z

PROTÓTIPO - HELP DESK BASEADO EM WORKFLOW UTILIZANDO SHELL UNIX
UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
DEPARTAMENTO DE SISTEMAS COMPUTAÇÃO
CURSO DE CIÊNCIAS COMPUTAÇÃO – BACHARELADO

Bem vindo: Claudio Cesar Schmitt Acesso restrito à usuários Sair

Cliente [Acessório Empresarial Schmitt] Abrir Chamado Chamados Nivel de Atendimento

Pesquisa por Chamados

Nro. Chamado	Área	Serviço	Versão
<input type="text"/>	<input type="text" value="Escolha"/>	<input type="text" value="Escolha"/>	<input type="text" value="Escolha um sistema"/>
Data Início (Maior que)	Data Início (Maior que)	Situação	
<input type="text" value="01/01/2007"/>	<input type="text" value="16/05/2007"/>	<input type="text" value="Abertos"/>	

Pesquisar

Listagem de Chamados

Nro Chamado	Abertura	Fechamento	Problema	Serviço	Status
31	02/05/2007 23:49		Troca de Senha de acesso	FISCA	
37	04/05/2007 00:30		Troca de Senha de acesso	CONTA	
40	04/05/2007 00:48	12/05/2007 02:08	Troca de Senha de acesso	CONTA	
53	12/05/2007 00:37	12/05/2007 12:48	Perda de dados ao postar	FISCA	
58	15/05/2007 19:07		Troca de Senha de acesso	CONTA	
59	15/05/2007 19:08		Não classificado	FOLHA	

Exibindo de 1 a 6 - Total de registros: 6

Legenda:

- Problema Reportado (PR)
- Análise Técnico (AT)
- Aguardando Usuário (AU)
- Retorno do Usuário (RU)
- Aguardando Fornecedor (AF)
- Retorno Fornecedor (RF)
- Solução Aplicada (SA)
- Chamado Encerrado (CE)

Figura 26 – Tela de Login de Usuário

Caso o mesmo deseja efetuar uma consulta, a tela disponibiliza uma área de pesquisa de chamados onde é possível efetuar as mais variadas consultas, como por exemplo: consultas pelo número do chamado, área, serviço, versão do serviço, data de abertura e situação do chamado. Na opção de pesquisa pela situação dos chamados, o usuário pode efetuar as consultas pelas seguintes opções: todos, abertos e fechados.

Quando um chamado encontra-se com situação de aberto, a tela disponibilizará os mesmos na cor amarela. Quando a cor é azul, significa que o chamado encontra-se fechado e quando o chamado estiver na cor vermelha significa que o mesmo encontra-se pendente de classificação não sendo possível visualizar o *status*.

Ao clicar no ícone , localizado junto à listagem dos chamados ao lado de serviços, o usuário pode consultar o *status* em que se encontra o chamado, conforme é demonstrado na Figura 27. As legendas de cada *status* são apresentadas na parte inferior da tela de *login* do usuário. O ícone da lupa é apresentado nas cores amarela, azul e vermelha de acordo com o *status* dos chamados, conforme descrito anteriormente.

Status do chamado 1		
Origem	Destino	Status
PR	AT	✓
AT	SA	✓
SA	AU	✓
AU	RU	✗
RU	CE	✗
RU	AT	✗

Figura 27 – Nodos/Status dos chamados

A tela disponibiliza ao usuário a visualização de cada passo executado contendo o nodo de origem e o nodo de destino de cada ação realizada no chamado. Para saber se o mesmo encontra-se finalizado ou não é apresentado os ícones ✓ e ✗, informando respectivamente que o passo está finalizado ou não.

A tela também permite ao usuário efetuar a abertura de chamados. Caso o usuário deseja efetuar a abertura de um chamado, o sistema disponibiliza a opção do botão “Abrir Chamado”. Ao clicar neste, o sistema encaminhará o usuário para a tela conforme é demonstrada na Figura 28.

Figura 28 – Tela de registro de chamados (Usuário)

A tela permite ao usuário efetuar a escolha do setor, serviço, versão e problema. Em seguida o sistema disponibiliza uma área para que o usuário possa efetuar a descrição do

problema. Caso o usuário deseja inserir a imagem de um erro, o sistema disponibiliza a opção através do botão “Enviar imagem para o servidor”. Para que o chamado seja registrado na base de dados, o usuário deve clicar no botão “Cadastrar”.

Quando um chamado é aberto, o usuário recebe um *e-mail* com o um texto padrão disponibilizado ao rodar a *shell*, conforme Figura 29.



Figura 29 – E-mail com texto padrão informando encerramento do chamado

O sistema disponibiliza ainda as opções de envio de *e-mail* para as seguintes situações do chamado: “Aguardando solução aplicada” e “Encerramento do Chamado”.

Ao receber o *e-mail* de encerramento do chamado, o usuário é informado de que necessita acessar o sistema de *help-desk* com o objetivo de responder e classificar o nível de atendimento do chamado recém-fechado, conforme é demonstrado na Figura 30.

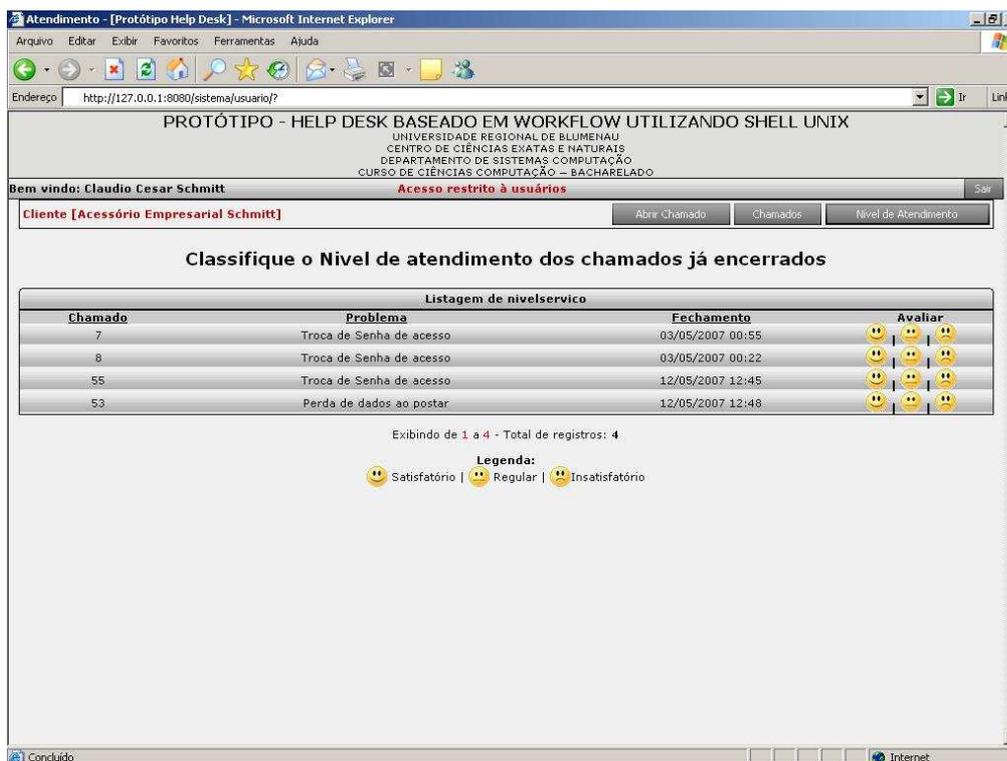


Figura 30 – Tela pesquisa de satisfação – Encerramento chamado

Para avaliar o nível de atendimento do chamado o usuário deve escolher uma das opções disponibilizadas ao lado de cada chamado. O nível de atendimento possui três opções: Satisfatório, Regular e Insatisfatório. A legenda dos ícones está disponibilizada na parte inferior da tela de nível de atendimento.

3.3.2.3 Login do Técnico

Após acesso do técnico, o sistema encaminhará o mesmo para a tela apresentada na Figura 31.

Para o técnico é apresentado ao lado esquerdo da tela o menu a ele destinado. Ao lado direito da tela é exibida uma área de pesquisa e logo abaixo a visualização da listagem de todos os chamados. Por padrão, o sistema carrega inicialmente todos os chamados abertos e que necessitam de uma análise do técnico.

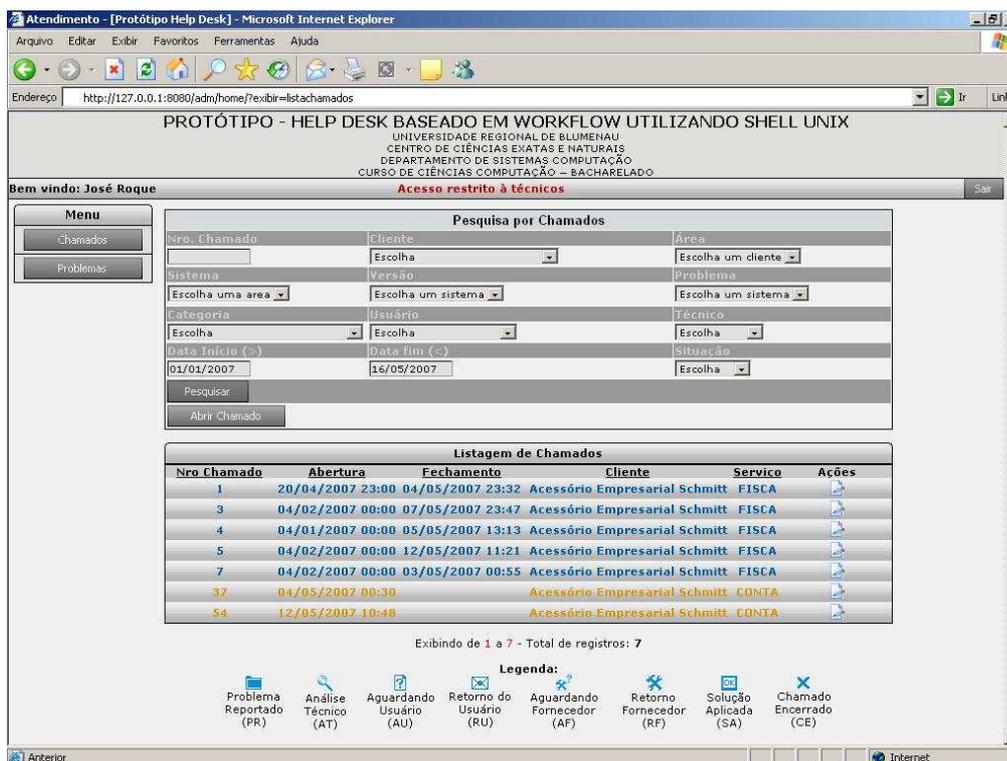


Figura 31 – Tela de Login do Técnico

Caso o mesmo deseja efetuar uma consulta, é possível efetuar variadas consultas, como por exemplo: consultas pelo número do chamado, área, serviço, versão do serviço, data de abertura e situação do chamado. Na opção de pesquisa pela situação dos chamados, o usuário pode efetuar a consultas pelas seguintes opções: todos, abertos e fechados.

Quando um chamado encontra-se com situação de aberto, a tela disponibilizará os mesmos na cor amarela. Caso o usuário desejar editar o chamado com o objetivo digitar os procedimentos de solução o mesmo deve clicar no ícone . Ao clicar neste ícone o sistema irá disponibilizar em uma nova janela a tela de edição do chamado em questão, conforme apresentado na Figura 32.

http://127.0.0.1:8080 - Atendimento - [Protótipo Help Desk] - Microsoft Internet Explorer

Atualizando o chamado Nº:61

Abertura: 15/05/2007 19:27 Fechamento:

Cliente: Acessório Empresarial Schmitt Área: Contábil

Serviço: Domínios Contábil Versão: Versão 1.0

Problema: Troca de Senha de acesso Categoria: Apoio ao Usuário

Usuário: Claudio Cesar Schmitt Responsável: José Roque

Imagens:

PR << 1 de 1 + AT Não encontrado!

Fonte: Tamanho: B I U

Digite aqui o problema reportado!

Concluído Internet

Figura 32 – Tela de registro de chamados (Técnico)

A tela permite ao técnico efetuar a escolha do cliente, área, setor, serviço, versão, problema, categoria e usuário do atendimento. Em seguida o sistema disponibiliza uma área para que o técnico possa efetuar a descrição do problema. Caso o usuário deseje inserir a imagem de um erro, o sistema disponibiliza a opção através do botão “Enviar imagem para o servidor”, conforme é demonstrado na Figura 33. Para que o chamado seja registrado na base de dados, o usuário deve clicar no botão “Cadastrar”.



Figura 33 – Tela de envio de imagem para servidor

A tela disponibiliza ao técnico a opção de inserir no máximo 10 imagens por chamado e será salva no seguinte formato: “ocor_numChamado_numSequencial.jpg”. Para salvar imagens ao chamado, o técnico deverá buscar a imagem na estrutura de diretórios de sua máquina através do botão “Procurar” e em seguida clicar no botão “Enviar”.

Quando a cor é azul, significa que o chamado encontra-se fechado. Caso o usuário desejar editar o chamado com o objetivo visualizar as ocorrências do chamado o mesmo deve clicar no ícone . Ao clicar neste ícone o sistema irá disponibilizar em uma nova janela a tela de edição de chamados, conforme Figura 34.

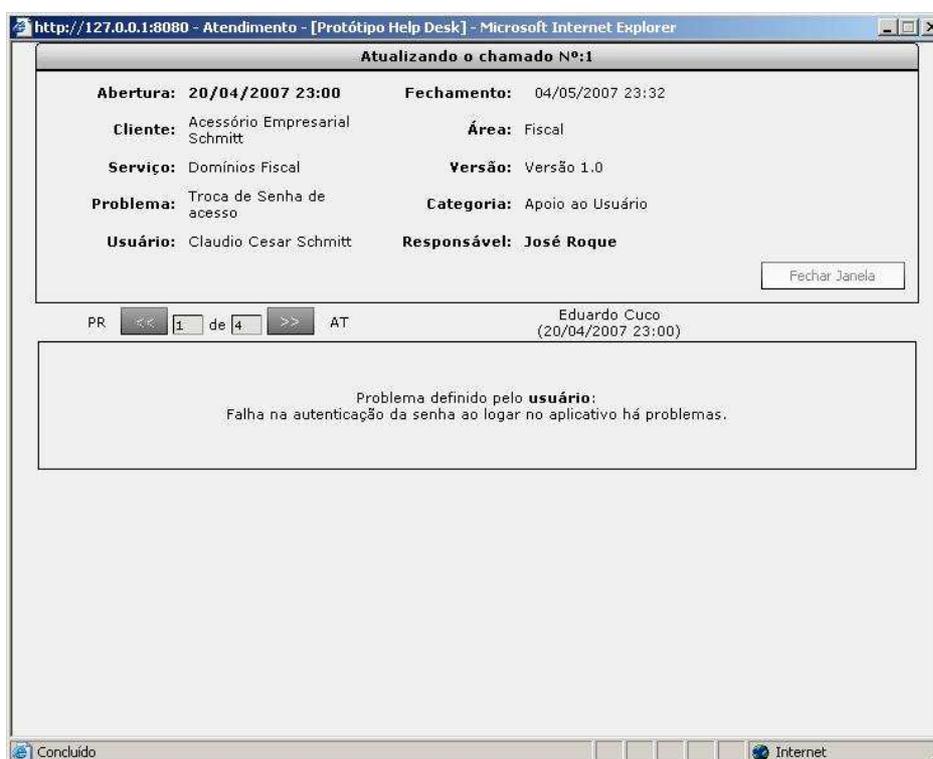


Figura 34 – Tela de chamados em modo consulta

A tela apenas disponibiliza ao técnico a visualização do chamado em modo consulta, ou seja, não é possível efetuar nenhuma alteração no chamado que já se encontra fechado, porém o técnico pode consultar todas as ocorrências do mesmo.

A tela também disponibiliza ao técnico a opção de cadastrar os problemas de cada serviço associado aos chamados, conforme é demonstrado na Figura 35. Na parte superior é apresentada uma área de pesquisa onde o técnico pode consultar os problemas pelas seguintes opções: descrição, serviço e problema além de poder efetuar o filtro dos problemas pelo seu *status*, ou seja, todos, ativos ou inativos. Na parte inferior a tela apresenta a listagem dos problemas de acordo com a consulta realizada. Por padrão, a tela apresenta todos os problemas ativos.

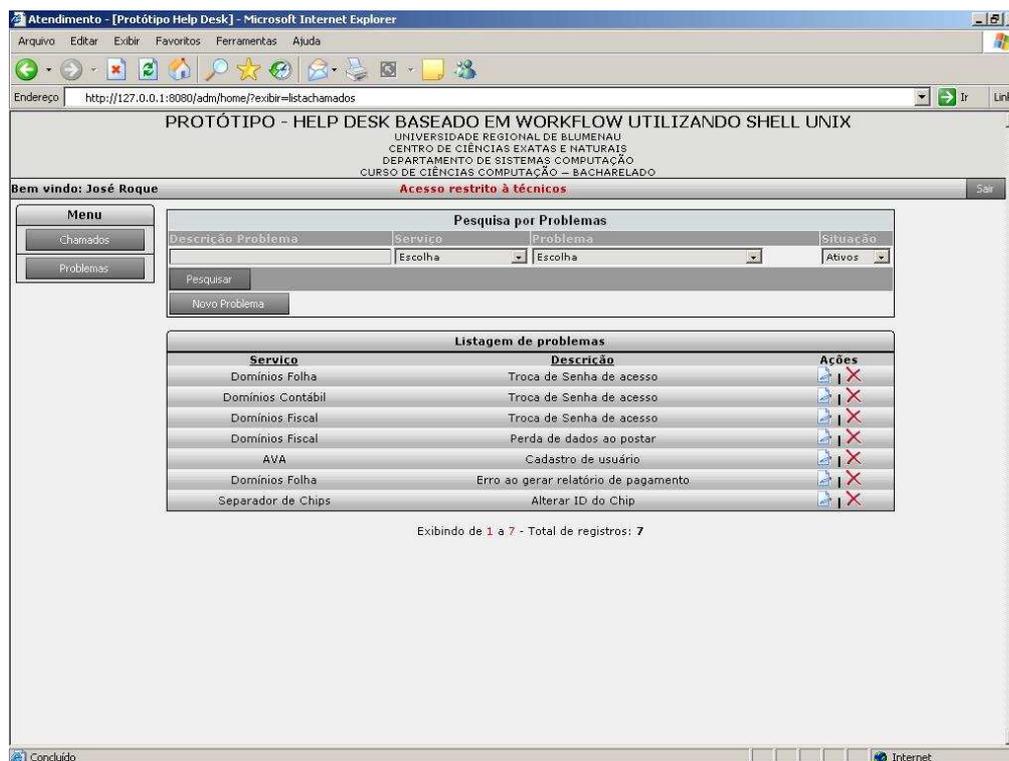


Figura 35 – Tela de cadastro de problemas

3.3.2.4 Login do Gerente

Após acesso do gerente, o sistema encaminhará o mesmo para a tela apresentada na Figura 36.

Para o gerente é apresentado ao lado esquerdo da tela o menu a ele destinado. Ao lado direito da tela é exibida uma área de pesquisa e logo abaixo a visualização da listagem de todos os chamados que necessitam ser associado a uma categoria e um técnico. Essa listagem somente estará disponível ao gerente na cor vermelha e somente quando existirem chamados pendentes desta associação.

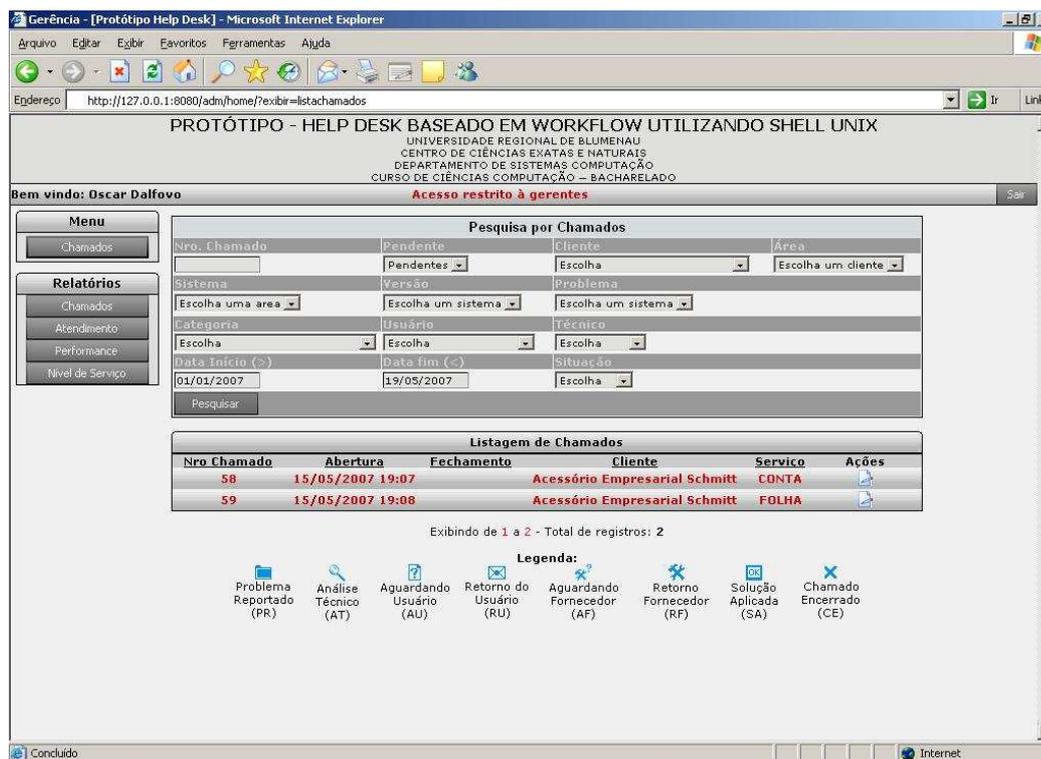


Figura 36 – Tela de Login do Gerente

O papel do gerente é estar efetuando a associação da categoria e técnico do chamado. Enquanto o mesmo não efetuar esta atividade, a *shell* estará monitorando os chamados pendentes desta associação e enviando um *e-mail*, conforme é demonstrado na Figura 37.



Figura 37 – E-mail com texto padrão solicitando a associação da categoria e técnico

Ao receber o *e-mail*, o gerente estará efetuando a associação da categoria do chamado e do técnico responsável pelo atendimento do chamado, conforme demonstra a Figura 38.

Figura 38 – Tela de Associação Categoria e Técnico

Após efetuar a associação da categoria e do técnico, o gerente tem a opção de salvar as informações pré-associadas. Caso o mesmo clique no botão “Salvar”, o sistema irá efetuar o salvamento das informações e fechar a tela. Caso o gerente desejar efetuar a associação em outra oportunidade, o mesmo tem a opção de clicar no botão “Fechar Janela”.

Caso o mesmo deseje efetuar uma pesquisa, é possível efetuar as mais variadas consultas, como por exemplo: consultas pelo número do chamado, situação, cliente, área, sistema, versão do sistema, problema, categoria, usuário, técnico, data de abertura e situação do chamado. Na opção de pesquisa pela situação dos chamados, o usuário pode efetuar a consultas pelas seguintes opções: todos, abertos e fechados.

A tela disponibiliza para o gerente quatro opções de relatórios, tais como: relatório de chamados dentro e fora do padrão, relatório de tempo gasto em atendimento de chamados, relatório de performance por técnico e relatório de nível de atendimento de chamados. Os mesmos relatórios podem ser gerados nas versões de impressão em formato de listagem ou impressão em formato de gráfico estatístico, exceto o relatório de nível de atendimento de chamados, que não possui a versão de impressão em formato de gráfico estatístico.

A Figura 39 apresenta a tela de geração do relatório de chamados dentro e fora do padrão.

Gerência - [Protótipo Help Desk] - Microsoft Internet Explorer

http://127.0.0.1:8080/adm/home/?exibir=listachamados

PROTÓTIPO - HELP DESK BASEADO EM WORKFLOW UTILIZANDO SHELL UNIX
UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
DEPARTAMENTO DE SISTEMAS COMPUTACIONAIS
CURSO DE CIÊNCIAS COMPUTAÇÃO - BACHARELADO

Bem vindo: Oscar Dalfovo **Acesso restrito à gerentes**

Menu
Chamados

Relatórios
Chamados
Atendimento
Performance
Nível de Serviço

Filtros do Relatório

Data início (Maior que) 01/01/2007 Data início (Menor que) 19/05/2007 Cliente Escolha Categoria Apoio ao Usuário

Imprimir

Listagem de Chamados no prazo

Nro Chamado	Cliente	Serviço	Tempo gasto
1	Acessório Empresarial Schmitt	Domínios Fiscal	32 Min
6	FURB	AVA	49 Min
55	FURB	AVA	43 Min

Exibindo de 1 a 3 - Total de registros: 3

Listagem de Chamados Fora do prazo

Nro Chamado	Cliente	Serviço	Tempo gasto
4	Acessório Empresarial Schmitt	Domínios Fiscal	12831 Min
5	Acessório Empresarial Schmitt	Domínios Fiscal	21681 Min
22	FURB	AVA	16794 Min
40	FURB	AVA	11599 Min
53	FURB	AVA	731 Min

Exibindo de 1 a 5 - Total de registros: 5

Figura 39 – Tela Relatório dos chamados abertos dentro e fora do padrão

Na parte superior da tela de Relatório de chamados dentro e fora do padrão de tempo é disponibilizada ao gerente uma área de pesquisa onde o mesmo pode consultar os chamados pelas seguintes opções: Data de início (“maior que” e “menor que”), cliente e categoria. Após pesquisa efetuada o gerente poderá gerar o relatório através dos ícones e .

Caso o gerente desejar gerar o relatório em formato de listagem o mesmo deve utilizar o ícone . A listagem é gerada exatamente igual à pesquisa realizada, porém são abertos uma nova janela com o botão de imprimir somente com os resultados necessários para a impressão do relatório, conforme é demonstrada Figura 40.

http://127.0.0.1:8080 - Gerência - [Protótipo Help Desk] - Microsoft Internet Explorer

Relatório de Chamados dentro e fora do padrão de tempo

Listagem de Chamados no prazo

Nro Chamado	Cliente	Serviço	Tempo gasto
1	Acessório Empresarial Schmitt	Domínios Fiscal	32 Min
6	FURB	AVA	49 Min
55	FURB	AVA	43 Min

Listagem de Chamados Fora do prazo

Nro Chamado	Cliente	Serviço	Tempo gasto
4	Acessório Empresarial Schmitt	Domínios Fiscal	12831 Min
5	Acessório Empresarial Schmitt	Domínios Fiscal	21681 Min
22	FURB	AVA	16794 Min
40	FURB	AVA	11599 Min
53	FURB	AVA	731 Min

Imprimir

Figura 40 – Relatório listagem dos chamados abertos dentro e fora do padrão

O mesmo conceito de geração de relatórios é aplicado para as telas de relatórios de tempo gasto em atendimento de chamados e tela de relatório de nível de atendimento de chamados.

Caso o gerente desejar gerar o relatório em formato de gráfico estatístico o mesmo deve utilizar o ícone . Após execução do relatório, o sistema abrirá uma nova janela com o gráfico no formato de barras demonstrando o percentual de performance dos chamados atendidos dentro do tempo do processo de *workflow* comparando com os chamados que não foram atendidos dentro do tempo do processo de *workflow*. A Figura 41 demonstra a imagem gerada após execução do relatório.

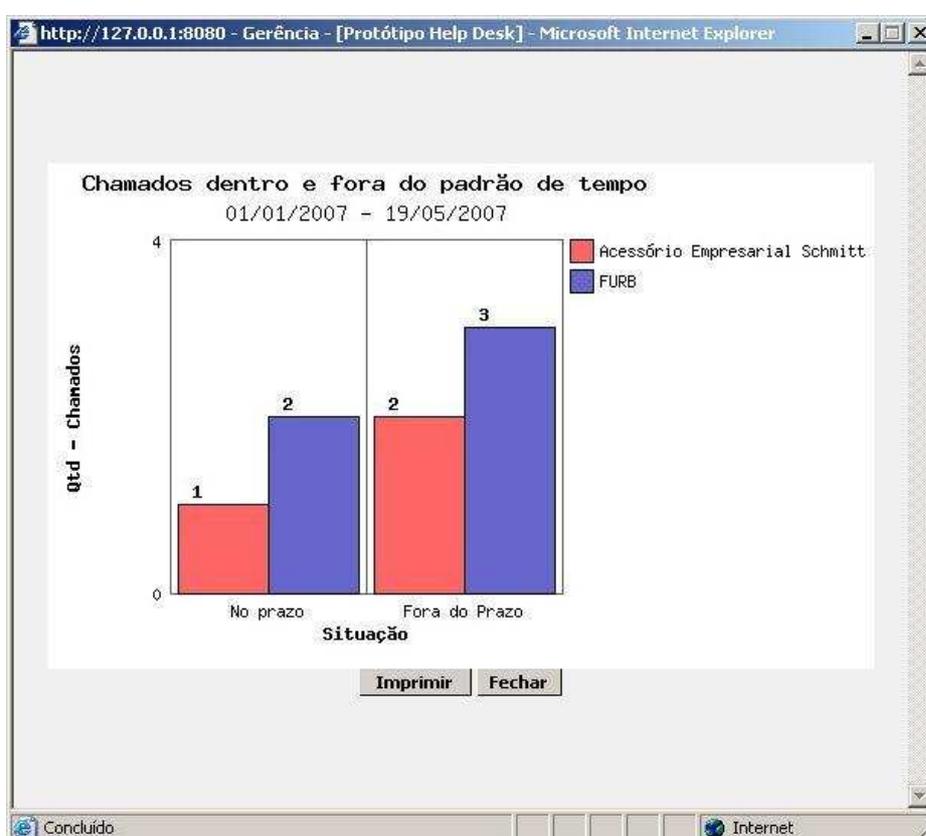


Figura 41 – Relatório gráfico dos chamados abertos dentro e fora do padrão

A Figura 42 apresenta a tela de geração de relatório de performance por técnico. Na parte superior da tela de Relatório de performance por técnico é disponibilizada ao gerente uma área de pesquisa onde o mesmo pode consultar os chamados pelas seguintes opções: Data de início (“maior que” e “menor que”) e técnico. Após pesquisa efetuada o gerente poderá gerar o relatório através dos ícones  e .

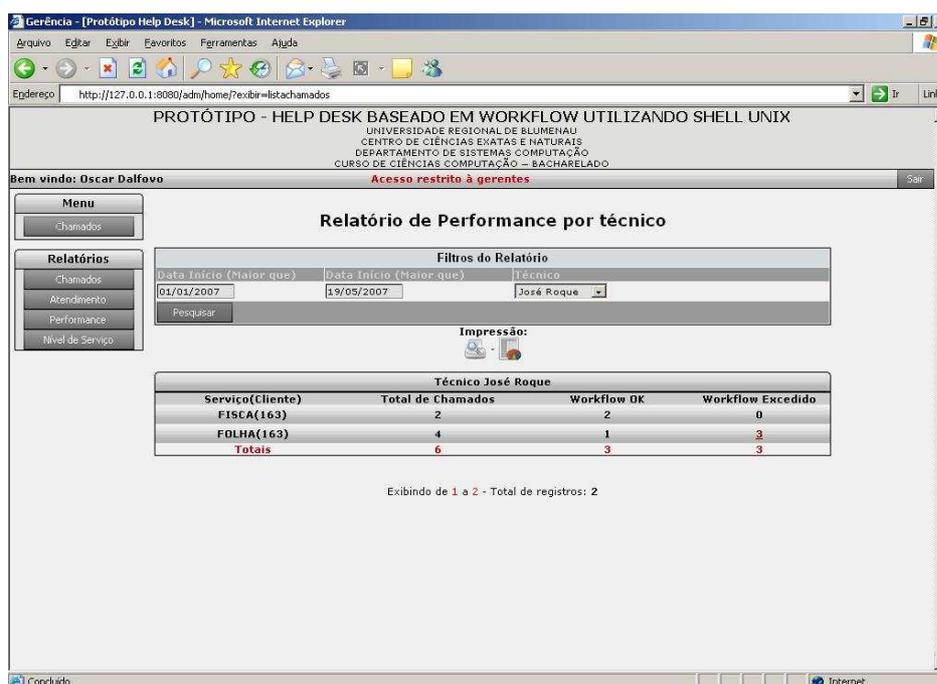


Figura 42 – Tela de geração de relatório de performance por técnico

O conceito do relatório em formato de listagem segue a mesma linha de raciocínio do relatório dos chamados abertos dentro e fora do padrão, ou seja, a mesma é gerada exatamente igual à pesquisa realizada.

Caso o gerente desejar gerar o relatório em formato de gráfico estatístico o mesmo deve utilizar o ícone . Após execução do relatório, o sistema abrirá uma nova janela com o relatório gráfico no formato pizza demonstrando o percentual de performance do técnico. A Figura 43 demonstra a imagem gerada após execução do relatório.

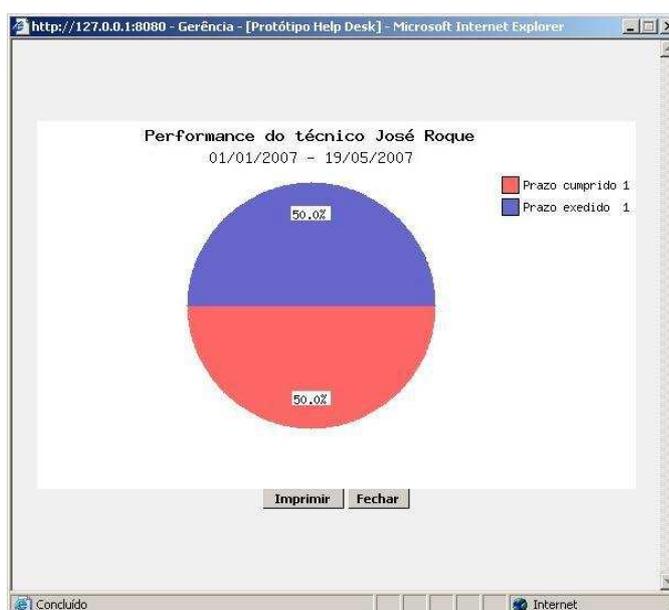
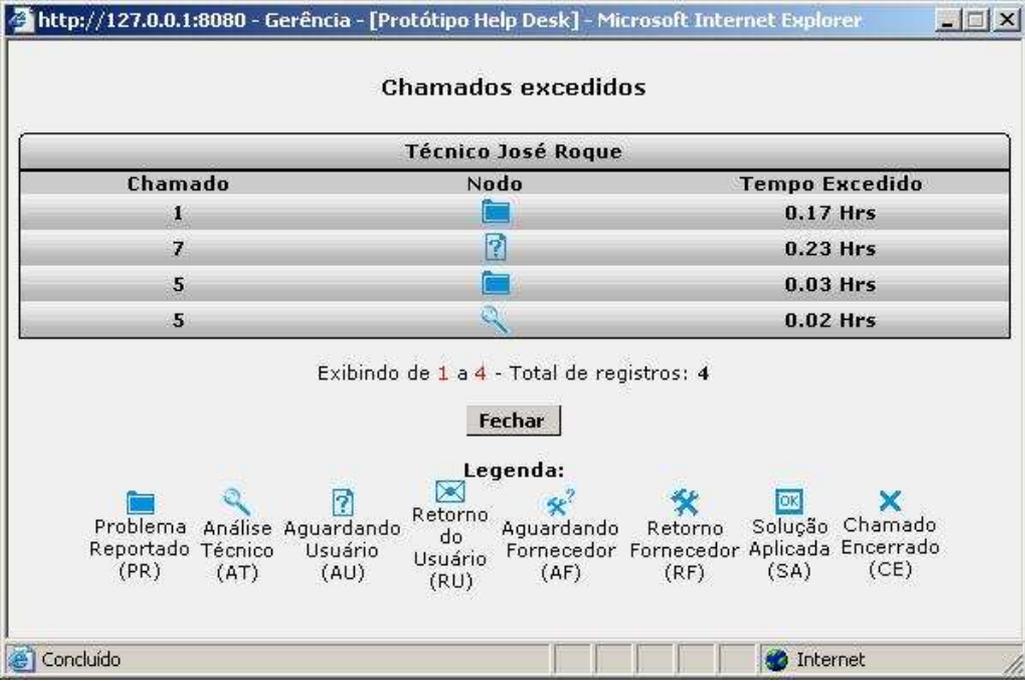


Figura 43 – Relatório gráfico dos chamados abertos dentro e fora do padrão

Caso o técnico não tenha cumprido o prazo de acordo com as regras do processo de

workflow, a tela também disponibiliza ao gerente em qual ponto do atendimento do chamado ocorreu o estouro do tempo. Para isso, o mesmo deve clicar no *link* do chamado onde existe o conceito de *workflow* estendido. Após clicar no *link*, o sistema abrirá uma nova janela demonstrando em qual ponto houve estouro do tempo no atendimento, conforme é demonstrada na Figura 44.



The screenshot shows a web browser window with the URL `http://127.0.0.1:8080 - Gerência - [Protótipo Help Desk] - Microsoft Internet Explorer`. The main content area is titled "Chamados excedidos" and displays a table for "Técnico José Roque".

Chamado	Nodo	Tempo Excedido
1		0.17 Hrs
7		0.23 Hrs
5		0.03 Hrs
5		0.02 Hrs

Below the table, it says "Exibindo de 1 a 4 - Total de registros: 4" and a "Fechar" button. A legend titled "Legenda:" defines the icons used in the table:

- Problema Reportado (PR)
- Análise Técnico (AT)
- Aguardando Usuário (AU)
- Retorno do Usuário (RU)
- Aguardando Fornecedor (AF)
- Retorno Fornecedor (RF)
- Solução Aplicada (SA)
- Chamado Encerrado (CE)

Figura 44 – Tela de chamados excedidos

A tela informa ao gerente as seguintes informações em relação ao tempo excedido em cada nodo do processo de *workflow*: número do chamado, nodo e o tempo excedido. Na parte inferior da tela é apresentada a legenda de cada nodo pertencente ao processo de *workflow*.

3.4 RESULTADOS E DISCUSSÃO

Grande parte dos sistemas de *help-desk* desenvolvidos, não possui o processo de *workflow* implementado, assim como o controle do tempo das tarefas realizadas dentro deste processo. O sistema desenvolvido tem o objetivo de atender essa necessidade, apresentado um formato dinâmico para o controle de tempo das atividades no processo de *workflow* nos chamados registrados em um *help-desk*.

As técnicas do AJAX mostraram-se muito eficientes na manipulação das requisições de atualização de informações e no formato padronizado de intercâmbio de dados. Por possuir

grande utilização do Javascript, notou-se que ainda falta a criação de ferramentas que facilitem a sua organização e manipulação das informações. Outrossim, notou-se que à medida onde aumentava a implementação do código Javascript, mais complicada torna-se a organização do código fontes, assim como também a correção de *bugs* no sistema.

Um dos fatores alterados durante o processo de definição do trabalho foi a utilização do sistema operacional Linux, sendo apresentado através da distribuição gratuita do Fedora Core 6 para o desenvolvimento da *shell*, em virtude de não ter-se encontrado uma distribuição gratuita do sistema operacional UNIX, fato este que não influenciou no cumprimento do requisito especificado, que era de desenvolver uma *shell* para controle monitoramento de chamados registrados no sistema de *help-desk*, com o objetivo de encaminhar através de e-mail informações aos usuários pertinentes a nodos do processo de *workflow* definido neste trabalho.

Com relação aos demais trabalhos pesquisados, este trabalho utilizou apenas os conceitos de *workflow* e AJAX para a definição de desenvolvimento do sistema de *help-desk*. No que diz respeito ao processo de *workflow*, implementou-se neste trabalho a utilização dos conceitos de nós de trabalho, nós de início e fim e nós de encaminhamento, os quais estão diretamente ligados ao processo de registro de ocorrências dos chamados, apresentando para o técnico quais são os passos do *workflow* já realizados, assim como os próximos passos a serem realizados. O sistema diferencia-se dos demais não apenas pelo controle das atividades, mas também pelo controle do tempo de execução das mesmas, o que abrange um controle de tempo das tarefas realizadas durante o atendimento aos chamados e apresentando informações estatísticas sobre a performance de atendimento dos técnicos. Em relação à utilização do AJAX, o sistema desenvolvido utiliza seus conceitos e técnicas durante todo o desenvolvimento deste trabalho, o que apresenta uma molhara em *performance* comparado com as demais ferramentas encontradas durante o desenvolvimento deste trabalho.

Por fim, pode-se afirmar que todos os RF foram implementados atingindo o resultado final proposto. Em relação aos RNF pode-se afirmar que todos estão contemplados no sistema e foram atingidos sem maiores dificuldades, devido à característica assíncrona das requisições do AJAX, que apresentam os resultados de pesquisas realizadas em tela, após a consulta ser finalizada. Com isso conclui-se que não é necessário transferir blocos muito grandes de informações de uma só vez do servidor para o navegador, permitindo assim que os mesmos executem suas tarefas simultaneamente e com maior velocidade.

4 CONCLUSÕES

Levando-se em consideração que grandes organizações, em sua maioria possuem os processos não documentados, o que torna o trabalho ineficiente gerando a perda da competitividade de mercado, para isto foi desenvolvido de um sistema para *help-desk* em *web*, utilizando o conceito de *workflow*, juntamente com as técnicas do AJAX para maior ganho de *performance* do sistema quando for necessário efetuar atualizações de informações.

O desenvolvimento do sistema utiliza o processo de gerenciamento do fluxo de trabalho em chamados, aplicando e implementando os conceitos de *workflow* para os relacionamentos e interações entre os processos.

Com relação aos objetivos, pode-se afirmar que todos foram atingidos. Os objetivos específicos definidos no início deste trabalho gerenciam o tempo gasto no fluxo de trabalho, analisando a *performance* do tempo médio gasto por técnico no atendimento de chamados. Estas informações são disponibilizadas através de listagens em tela e gráficos estatísticos, com o objetivo de apresentar a gerência os resultados obtidos pelos técnicos no registro de chamados. Outro objetivo foi aplicar as técnicas do AJAX em apenas algumas áreas do sistema. No entanto, durante a realização deste trabalho decidiu-se utilizar as técnicas em todo o sistema, devido à eficiência na manipulação das requisições assíncronas e na padronização do formato de intercâmbio de dados, além do seu ganho de *performance* em aplicações *web*.

O papel da *shell* é monitorar paralelamente ao sistema de *help-desk* todos os chamados registrados, informando via *e-mail* os atores Usuário e Gerente o posicionamento sobre os nodos de *workflow*, representados com "Abertura de chamado", "Encerramento de chamado", "Aguardando retorno de solução aplicada" e "Associação de categoria e técnico responsável".

Ao concluir este trabalho, notou-se que a utilização do *workflow* em sistemas apresenta um ganho considerável em controle de tarefas, além de automatizar todo o processo envolvido no fluxo de trabalho. Para tanto, é necessário que sejam definidas quais serão as tarefas envolvidas, bem como os nós de trabalho por onde o *workflow* será aplicado, evitando que ocorram falhas durante o controle destes processos.

Pode-se dizer ainda que o sistema desenvolvido neste trabalho possui algumas limitações, como o de não possuir uma diferenciação dos técnicos de acordo com o seu conhecimento, ou seja, todos os técnicos que se conectam no sistema serão avaliados igualmente no *workflow* do chamado atendido, independente de seu nível de conhecimento.

Outra limitação do sistema é quando o chamado encontra-se com o *status* de “Aguardando o usuário”, ou seja, o *workflow* aplicado não paralisa a sua contagem do tempo, penalizando assim o técnico no que se refere ao fluxo de trabalho do chamado atendido.

O envio de *e-mail* também se torna uma limitação do sistema quando o usuário não estiver verificando seus *e-mails* em tempo integral durante o seu horário de trabalho.

A seção extensões sugere possíveis implementações que complementaríamos as funcionalidades e limitações apontadas após a conclusão deste trabalho.

4.1 EXTENSÕES

Este trabalho contempla o controle de tempo nas atividades executadas no atendimento a chamados registros no sistema. O *workflow* em atendimento a chamados é categorizado por níveis de dificuldade e se podem criar novas categorias de *workflow* através de um recurso administrativo. Como sugestão para extensão deste trabalho poderiam ser implementadas algumas outras funcionalidades como:

- a) envio de mensagens SMS para os usuários com informativos parciais sobre o andamento dos chamados;
- b) criação de regras para controle de tempo diferenciado por níveis, ou seja, diferenciar o controle de tempo quando os técnicos têm conhecimento diferenciado;
- c) controle paralelo para chamados que entram em status de aguardando o usuário, visto que no sistema desenvolvido o técnico será penalizado quando este tempo for ultrapassado;
- d) criação de um recurso para sinalização das atividades, verificando o tempo utilizado e o tempo estante, informando o técnico através de mensagens de alerta no próprio sistema;
- e) implementação de um recurso administrativo diferenciado para a execução da shell, ou seja, no sistema desenvolvido apenas podemos escolher o intervalo de execução da shell, mas não podemos escolher o tempo de execução diferenciado para os tipos de e-mail enviados.

REFERÊNCIAS BIBLIOGRÁFICAS

AJAX (web). In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, [2006?]. Disponível em: <http://pt.wikipedia.org/wiki/AJAX_%28Web%29>. Acesso em: 26 ago. 2006.

ASLESON, Ryan; SCHUTTA, Nathaniel. **Fundamentos do AJAX**. Rio de Janeiro: Alta Books, 2006.

BAACHART. **baaChart**. [S.l.], 2006. Disponível em: <<http://www.phpclasses.org/browse/package/1193.html>>. Acesso em: 07 maio 2007.

BRAGGIO, Simone. **Workflow**: uma tecnologia para agilizar o desenvolvimento de novos produtos e serviços. [S.l.], 2006. Disponível em: <<http://www.path.com.br/noticia26.html>>. Acesso em: 26 out. 2006.

CRANE, David; PASCARELLO, Eric. **Ajax in action**. Greenwich: Manning, 2005.

CHRISTHIAN, Kaare. **Sistema operacional UNIX**. Rio de Janeiro: Campus, 1985.

CRONTAB (web). In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, [2006?]. Disponível em: <<http://pt.wikipedia.org/wiki/Crontab>>. Acesso em: 25 out. 2006.

CRUZ, Tadeu. **Workflow**: a tecnologia que vai revolucionar processos de negócio. [S.l.], 2003. Disponível em: <http://www.trcr.com.br/ver_opiniao.asp?codigo=44>. Acesso em: 25 ago. 2006.

GARRETT, Jesse. **AJAX**: a new approach to web applications, adaptive path publications. [S.l.], 2005. Disponível em: <<http://www.adaptivepath.com/publications/essays/archives/000385.php>>. Acesso em: 01 set. 2006.

GOIS, Marcus. **AJAX na construção de uma aplicação web para monitoramento de ambientes em plantas 2d**. 2006. 72 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

GONÇALVES, Luis. **Apostila de XHTML**: curso introdutório. [S.l.], 2005. Disponível em: <<http://grade01.lncc.br/~lrodrigo/arquivos/xhtml.basico.pdf>> Acesso em: 27 mar. 2007.

GROFF, James; WEINBERG, Paul. **Sistema operacional UNIX**: um guia conceitual. Rio de Janeiro: Campus, 1986.

HBSIS. **0800NET**: visão geral. Blumenau. [2006?]. Disponível em:
<<http://www.0800net.com.br/0800net/default.asp>>. Acesso em: 02 set. 2006.

HELPDESK. In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, [2006?]. Disponível em: <http://pt.wikipedia.org/wiki/Help_Desk>. Acesso em: 22 out. 2006.

JARGAS, Aurélio. **Shell script**. [S.l.], [1999?]. Disponível em:
<<http://www.aurelio.net/shell/>>. Acesso em: 05 set. 2006.

LIMEIRA, José. **Utilização de AJAX no desenvolvimento de sistemas web**. Porto Alegre, 2006. Disponível em: <http://www.limeira.eti.br/monografia_ajax.pdf>. Acesso em: 25 mar. 2007.

LINUX. **Fedora Project**: Fedora Core 6. [S.l.], [2006]. Disponível em:
<http://www.linuxnarede.com.br/conceitos/distros/linux_fedora.php> Acesso em: 05 abr. 2007.

MICHAEL, Randal. **Dominando Unix**: shell scripting. Rio de Janeiro: Campus, 2003.

MORIMOTO, Carlos. **VMware Server**. [S.l.], 2007. Disponível em:
<<http://www.guiadohardware.net/tutoriais/vmware-server/>>. Acesso em: 12 maio. 2002.

MORUJÃO, A. et al. **AJAX**: uma nova visão da web. Porto, 2005. Disponível em:
<<http://gnomo.fe.up.pt/~arestivo/page/artigos/seminarios/2005/grupo21.pdf>>. Acesso em: 23 abr. 2007.

MYSQL. In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, [2006?]. Disponível em: <<http://pt.wikipedia.org/wiki/MySQL>>. Acesso em: 04 set. 2006.

NICOLAU, Mariano. **Modelagem de workflow utilizando um modelo de dados temporal orientado a objetos com papéis**. 1998. 99 f. Dissertação (Mestrado em Ciências da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em: <<http://www.inf.ufrgs.br/~nina/Dissertacao/MarianoNicolao.pdf>>. Acesso em: 02 set. 2006.

PLESUMS, Charles. **Introduction to workflow**. Texas, 2002. Disponível em:
<<http://plesums.com/image/introworkflow.html>>. Acesso em: 23 ago. 2006.

REIS, Ricardo; TRINDADE, Luis. **AJAX**: introdução. [S.l.], 2005. Disponível em:
<<http://pwp.net.ipl.pt/alunos.isel/24138/AJAX/IntroducaoAJAX.pdf>>. Acesso em: 01 set. 2006.

SEQUEIRA, João. **Workflow management systems**. [S.l.], [2006?]. Disponível em:
<<http://www.di.fc.ul.pt/~ler/docencia/tm0405/WorkflowManagementSystems.pdf>>. Acesso em: 25 out. 2006.

SHELL. In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, [2006?]. Disponível em: <<http://pt.wikipedia.org/wiki/Shell>>. Acesso em: 27 ago. 2006.

SOARES, Wallace. **Programando em PHP: conceitos e aplicações**. 2. ed. São Paulo: Érica Ltda, 2000.

_____. **AJAX: guia prático para windows**. São Paulo: Érica Ltda, 2006.

SPARX SYSTEMS. **Enterprise Architect**. [S.l.], 2006. Disponível em: <<http://www.sparxsystems.com.au/products/ea.html>>. Acesso em 23 mar. 2007.

STATDLOBER, Juliano. **Help-Desk e SAC com qualidade**. Rio de Janeiro: Brasport, 2006.

W3C. **The extensible hypertext markup language**. [S.l.], 2002. Disponível em: <<http://www.w3.org/TR/xhtml1/#xhtml>>. Acesso em: 23 mar. 2007.

W3SCHOOLS. **Introduction to CSS**. [S.l.], 2006. Disponível em: <http://www.w3schools.com/css/css_intro.asp>. Acesso em: 23 mar. 2007.

WIECZOREK, Emilio. **Sistema de gerenciamento de documentos jurídicos utilizando XML, DOM e a Plataforma .NET**. Palmas, 2004. Disponível em: <http://www.ulbrato.br/ensino/43020/artigos/relatorios2004-2/Arquivos/Emilio_TCC.pdf>. Acesso em: 24 mar. 2007.

WFMC. **The workflow management coalition**. [S.l.], [2006?]. Disponível em: <<http://www.wfmc.org>>. Acesso em: 25 mar. 2007.

WILVERT, Carla. **Aplicação de sistemas em help-desk utilizando gestão do conhecimento e técnicas de raciocínio baseado em casos**. 2004. 72 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.