

UNIVERSIDADE DA REGIÃO DE JOINVILLE - UNIVILLE
DEPARTAMENTO DE INFORMÁTICA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

DESENVOLVIMENTO E IMPLANTAÇÃO DE UM HELPDESK PARA A EMPRESA WIEST
S/A.

RICARDO MONDINI

Joinville
2007

RICARDO MONDINI

DESENVOLVIMENTO E IMPLANTAÇÃO DE UM HELPDESK PARA A EMPRESA WIEST
S/A.

Trabalho de conclusão de Estágio
apresentado ao Curso de Bacharelado em
Sistemas de Informação na Universidade da
Região de Joinville – UNIVILLE – como
requisito parcial para a obtenção de Grau em
Sistemas de Informação.
Orientador Específico: Paulo Marcondes
Bousfield

Joinville – SC
2007

TERMO DE APROVAÇÃO

O aluno **Ricardo Mondini**, regularmente matriculado na 4ª série do curso de Sistemas de informação, apresentou e defendeu o Presente trabalho de Conclusão de Estágio obtendo da Banca Examinadora a média final ____ (_____), tendo sido considerado aprovado.

Joinville, ____ de _____ de _____.

Prof.

Prof.

Paulo M. Bousfield

Página contendo a avaliação do estágio.

A minha família que sempre me apoiou a frequentar a universidade, mesmo nos momentos difíceis, a minha namorada Daiani que me deu força nos momentos em que me faltaram, a meus amigos que sempre estiveram presentes em mais essa jornada da minha vida.

Em especial a Sérgio Mondini, meu querido falecido pai, que nos deixou em abril deste ano, que sempre ajudou a conseguir o que eu busquei, principalmente a iniciar e concluir este curso.

A Deus que deu forças para seguir, fazendo com que não caísse mesmo no mais duro momento de minha vida.

RESUMO

MONDINI, Ricardo. Desenvolvimento e implantação de um Helpdesk para a empresa Wiest S/A. Trabalho apresentado à disciplina de estágio curricular supervisionado para obtenção de Bacharel em Sistemas de Informação. Univille. 2007. Este trabalho tem como objetivo a criação de um software para documentação e resolução de problemas relacionados à área de Informática da empresa Wiest S/A. Através deste sistema, os colaboradores que utilizam de alguma forma os equipamentos de informática da empresa, serão cadastrados para poderem utilizar do Helpdesk, informando o problema que está ocorrendo para que um dos analistas avalie e corrija-o. Atualmente os problemas são informados aos analistas através de ligações ao setor. Além desta abertura de chamados, o usuário também poderá pesquisar por problemas ocorridos anteriormente, fazendo com que a atuação dos analistas não seja necessária. Com todos os chamados e resoluções de problemas cadastrados no sistema, será possível criar uma base de conhecimento, fazendo com que se saiba exatamente qual setor e/ou área necessita de novos equipamentos ou treinamentos. O embasamento teórico foi realizado através de livros sobre o tema em questão, e para a elaboração do trabalho utilizou-se toda a informação relevante a respeito da empresa. O levantamento das informações foi feito através da análise do projeto, onde foi empregada a técnica de UML para elaborar o modelo lógico do sistema. Para o desenvolvimento do software, foram utilizadas as linguagens PHP e HTML, e o banco de dados MySQL. O sistema será disponibilizado aos usuários através da intranet da empresa.

Palavras-chave: análise do projeto, UML, PHP, HTML, MySQL.

LISTA DE FIGURAS

Figura 1 – Processos de desenvolvimento de software	22
Figura 2 – UML – Representação de uma Classe.	27
Figura 3 – UML – Representação de um Objeto.....	28
Figura 4 – UML – Representação dos relacionamentos.....	28
Figura 5 – UML – Casos de Uso	28
Figura 6 – UML – Representação de um Ator	28
Figura 7 – Site Institucional da empresa Wiest S/A.....	53
Figura 8 – Diagrama de Classes	57
Figura 9 – Priorização dos Casos de Uso do Sistema HelpDesk.....	61
Figura 10 – Modelo de Caso de Uso: Gestão de Departamentos	67
Figura 11 – Modelo de Caso de Uso: Gestão de Problemas	68
Figura 12 – Modelo de Caso de Uso: Gestão de Usuários	68
Figura 13 – Modelo de Caso de Uso: Gestão de Perguntas Frequentes	68
Figura 14 – Modelo de Caso de Uso: Abertura de Chamados	69
Figura 15 – Modelo de Caso de Uso: Assumir Chamado	69
Figura 16 – Modelo de Caso de Uso: Atendimento Chamado	69
Figura 17 – Modelo de Caso de Uso: Chamado em Diligência	70
Figura 18 – Modelo de Caso de Uso: Finalizar Chamado.....	70
Figura 19 – Modelo de Caso de Uso: Cancelar Chamado.....	70
Figura 20 – Modelo de Caso de Uso: Emissão de Relatórios	71
Figura 21 – Modelo de Caso de Uso de Contexto do Sistema HelpDesk	72
Figura 22 – Diagrama de Classe detalhado do Sistema Helpdesk	73
Figura 23 – Normalização do Banco de Dados Sistema Helpdesk	74
Figura 24 – Tela da Intranet da empresa	78
Figura 25 – Tela de Login do Sistema Helpdesk.....	79
Figura 26 – Tela inicial e de FAQ - Administrador	80
Figura 27 – Tela de Abertura de Chamados - Administrador	81
Figura 28 – Tela de Todos os Chamados - Administrador.....	82
Figura 29 – Tela de Lista de Chamados - Administrador.....	83
Figura 30 – Tela de Meus Chamados - Administrador	84
Figura 31 – Tela de Lista de Meus Chamados - Administrador.....	85
Figura 32 – Tela de Adição de Usuário – Administrador	86
Figura 33 – Tela de Editar Usuários – Administrador.....	87
Figura 34 – Tela de Editar Usuário – Popup – Administrador.....	88
Figura 35 – Tela de Adicionar Departamentos – Administrador	89
Figura 36 - Tela de Editar Departamentos – Administrador	90
Figura 37 – Tela de Editar Departamentos – Popup – Administrador	90
Figura 38 – Tela de Adicionar FAQ – Administrador	91
Figura 39 – Tela de Editar FAQ – Administrador.....	92

Figura 40 – Tela de Editar Faq – Popup – Administrador.....	93
Figura 41 – Tela de Relatórios	94
Figura 42 – Tela de Exibição dos dados do Relatório – Administrador.....	95
Figura 43 – Tela inicial e de FAQ – Usuário	96
Figura 44 – Tela de Abertura de Chamado – Usuário.....	97
Figura 45 – Tela de Meus Chamados – Usuário	98
Figura 46 – Tela de Lista Meus Chamados – Usuário	99

LISTA DE TABELAS

Tabela 1 – Atores do Sistema Helpdesk	58
Tabela 2 – Relacionamento de Casos de Uso X Atores	59
Tabela 3 – Detalhamento do Caso de Uso: Gestão de Departamentos	62
Tabela 4 – Detalhamento do Caso de Uso: Gestão de Usuários	62
Tabela 5 – Detalhamento do Caso de Uso: Gestão de Problemas	63
Tabela 6 – Detalhamento do Caso de Uso: Perguntas Frequentes	63
Tabela 7 – Detalhamento do Caso de Uso: Abertura de Chamados	64
Tabela 8 – Detalhamento do Caso de Uso: Assumir Chamados	64
Tabela 9 – Detalhamento do Caso de Uso: Atendimento de Chamados	65
Tabela 10 – Detalhamento do Caso de Uso: Chamado em Diligência	65
Tabela 11 – Detalhamento do Caso de Uso: Finalizar Chamado	66
Tabela 12 – Detalhamento do Caso de Uso: Cancelar Chamado	66
Tabela 13 – Detalhamento do Caso de Uso: Emissão de Relatórios	67
Tabela 14 – Matriz de Associação do Sistema Helpdesk	75
Tabela 15 – Dicionário de Dados: HELP_CHAMADOS	75
Tabela 16 – Dicionário de Dados: HELP_CHAMADOS_STATUS	76
Tabela 17 – Dicionário de Dados: HELP_DEPARTAMENTOS	76
Tabela 18 – Dicionário de Dados: HELP_PROBLEMAS	76
Tabela 19 – Dicionário de Dados: HELP_FAQ	76
Tabela 20 – Dicionário de Dados: HELP_USUÁRIO	77

SUMÁRIO

RESUMO.....	6
LISTA DE FIGURAS.....	7
LISTA DE TABELAS.....	9
SUMÁRIO.....	10
INTRODUÇÃO	12
1 PROJETO DE ESTÁGIO	13
1.1 Identificação do Estágio	13
1.1.1 Dados de Identificação do Aluno	13
1.1.2 Dados de Identificação da Empresa.....	13
1.1.3 Dados dos Responsáveis pelo Estágio.....	13
1.2 Tema	14
1.3 Assunto	14
1.4 Problema.....	14
1.5 Objetivos.....	15
1.5.1 Objetivo Geral	15
1.5.2 Objetivos Específicos	15
1.6 Justificativa.....	16
1.7 Metodologia.....	17
1.8 Cronograma	18
2 Fundamentação Teórica	19
2.1 Sistemas de Informação.....	19
2.2 Construindo Sistemas de Informação.....	20
2.3 Metodologia de Construção de Sistemas.....	23
2.4 UML	27
2.4.1 Diagramas de Classe.....	29
2.4.2 Diagrama de Caso de Uso	30
2.4.3 Diagramas de Interação	31
2.4.4 Diagramas de Seqüência.....	32
2.5 Levantamento de Requisitos.....	32
2.5.1 Conceito de Requisitos	33
2.5.2 Atividades do Levantamento de Requisitos	33
2.5.2.1 Elaborar Modelo de Negócio.....	34
2.5.2.2 Descobrir Atores e Casos de Uso	34
2.5.2.3 Prototipar a Interface com o Usuário.....	35
2.5.2.4 Detalhar Casos de Uso.....	35
2.6 Análise	35
2.6.1 Análise arquitetônica	36
2.6.2 Análise de Casos de Uso	37
2.6.3 Análise de Classes	37

2.7 Banco de dados	37
2.7.1 Sistema Gerenciador de Banco de Dados.....	38
2.7.2 <i>Design</i> do Banco de Dados.....	39
2.7.2.1 Normalização dos Dados	39
2.7.2.2 MER – Modelo Entidade-Relacionamento.....	40
2.8 Linguagens de Programação.....	41
2.8.1 Primeira Geração – Linguagem de máquina	42
2.8.2 Segunda Geração – Linguagem <i>Assembly</i>	42
2.8.3 Terceira Geração – Linguagem de alto nível.....	42
2.8.4 Quarta geração	43
2.8.5 Linguagem Orientada a Objetos	44
2.9 Ferramentas de Desenvolvimento	44
2.9.1 Servidores para internet	45
2.9.2 Servidor Apache	46
2.9.3 PHP	46
2.9.4 JavaScript	47
2.9.5 MySQL	48
2.10 Intranet.....	49
2.11 Internet.....	50
2.12 Fornecendo acesso para usuário baseado na Web	51
3 DESCRIÇÃO DA PRÁTICA	52
3.1 Local do Desenvolvimento do Helpdesk Wiest S/A	52
3.2 O Helpdesk	53
3.3 Levantamento de requisitos	55
3.3.1 Elaborar modelo de negócio	56
3.3.2 Encontrar Atores e Casos de Uso	57
3.3.2.1 Encontrar Atores.....	58
3.3.2.2 Encontrar Casos de Uso.....	59
3.3.3 Priorizar Casos de Uso	60
3.3.4 Detalhar Casos de Uso.....	62
3.3.5 Estruturar Modelo de Caso de Uso.....	67
3.3.6 Elaborar Modelo de Caso de Uso de Contexto.....	71
3.4 Analise do Sistema Helpdesk	72
3.5 Design do Sistema Helpdesk	74
3.5.1 Projetar Banco de Dados	74
3.5.2 Matriz de Associação.....	75
3.5.3 Dicionário de Dados	75
3.5.4 Projetar Interface	77
3.5.4.1 Área do administrador	79
3.5.4.2 Área do usuário.....	96
CONSIDERAÇÕES FINAIS.....	100
BIBLIOGRAFIA	102

INTRODUÇÃO

A empresa Wiest S/A, possui profissionais no Departamento de TI que resolvem com rapidez e eficiência os problemas gerados pelos equipamentos e pelo software da empresa, porém muitas vezes os problemas ocorridos, são repetidos, fazendo com que haja um re-trabalho muito grande para os analistas. Além deste problema, a outro que atualmente não há como ocorrer mais dentro de empresas deste porte, que é a falta de documentação dos trabalhos feitos. Esta falta de documentação trás problemas freqüentes aos analistas e ao gerente da área, pois não há como confirmar os trabalhos concluídos, e não existem relatórios sobre problemas ocorridos, dificultando a tomada de decisão, principalmente no momento de compra de novos equipamentos e de treinamentos. Com isso, surgiu a necessidade de uma ferramenta que automatize toda a área de suporte a informática dentro da empresa e que gere relatórios para documentar todos os trabalhos concluídos e facilitem a tomada de decisão dos gerentes e diretores da empresa.

Este trabalho de conclusão de estágio tem por objetivo a criação de um Helpdesk para a solução de problemas relacionados à área de informática da empresa Wiest S/A. O sistema contará com uma base de conhecimentos, para os usuários pesquisarem sobre os problemas ocorridos e encontrar a solução apropriada para seu problema. Caso não encontre a solução adequada, o usuário poderá abrir um chamado para a informática e informar o problema que está ocorrendo. Desta forma o analista responsável irá atender ao pedido.

Com o objetivo de proporcionar um melhor entendimento das definições relacionadas ao desenvolvimento do Sistema Helpdesk, este trabalho será estruturado em três capítulos: projeto, fundamentação teórica e descrição da prática.

No projeto, que estará disposto no primeiro capítulo, será apresentado a identificação do estágio, o tema, assunto, problema, objetivos, justificativa, metodologia e um cronograma. O segundo capítulo descreve os conceitos relacionados a sistemas de informação, a metodologia, tecnologias e ferramentas utilizadas no desenvolvimento de um sistema baseado em tecnologias da *Internet*. Por fim, o terceiro capítulo conta com a descrição da prática realizada durante o desenvolvimento da aplicação.

1 PROJETO DE ESTÁGIO

1.1 Identificação do Estágio

1.1.1 Dados de Identificação do Aluno

Nome: Ricardo Mondini

Curso: Bacharelado em Sistemas de Informação

Endereço: Rua Cerro Verde, 585 - Iriú - Joinville - SC.

E-mail: mondini@gmail.com

1.1.2 Dados de Identificação da Empresa

Denominação: Wiest S/A.

Ramo de Atividade: Industrial. Fabricação de Tubos de Aço e Escapamentos.

Endereço: Rua Erwino Menegotti, 588 – Água Verde – Jaraguá do Sul - SC.

Fone: (47) 3372-5150 e-mail: ricardo.mondini@wiest.com.br

1.1.3 Dados dos Responsáveis pelo Estágio

Orientador de Classe: Paulo Marcondes Bousfield.

Orientador Específico: Paulo Marcondes Bousfield.

Supervisor no Campo de Estágio: Jeferson Sesering.

1.2 Tema

Desenvolvimento de um Helpdesk para a empresa Wiest S/A, como uma ferramenta de suporte a microinformática e ao sistema LOGIX, a fim de facilitar a resolução de problemas referentes à informática na empresa.

1.3 Assunto

Este trabalho visa desenvolver um Helpdesk para a empresa, fazendo assim com que o usuário tenha a possibilidade de abrir chamados para a informática, informando seu problema para a área de suporte e manutenção de hardware, microinformática e também para a área de suporte ao sistema da empresa (logix). As respostas permanecerão no sistema, e haverá uma área de pesquisa para os usuários buscarem erros que já aconteceram e possam resolver sem abrir um novo chamado ou efetuar uma ligação ao setor.

O sistema Helpdesk irá gerar também um relatório para o gerente da área analisar o trabalho dos colaboradores do setor, analisando a carga de trabalho. Também será possível, analisar quais os principais problemas que ocorrem com os computadores da empresa, e com isso minimizá-los através de modificações, atualizações e/ou treinamento dos usuários.

1.4 Problema

O principal problema encontrado na empresa na área de TI a ser resolvido, é o alto número de ligações para o setor de informática referente a problemas ocorridos com os computadores e com o sistema ERP que a organização utiliza.

Existem cerca de 110 equipamentos de informática na empresa, dentre eles, 100 computadores, 100 usuários e 60 licenças do ERP. Com esta estatística é possível imaginar o número de problemas diários que acontecem com os equipamentos e com o sistema. Desta forma, o departamento de TI recebe muitas ligações referentes a problemas já resolvidos anteriormente, fazendo com que a equipe perca um precioso tempo, resolvendo problemas que poderiam ser resolvidos facilmente por um FAQ que permitisse ser acessado pelos usuários.

Isto gera também outro grave problema, que é a falta de documentação referente a erros ocorridos, pois desta forma, os processos de melhorias são dificultados, e principalmente, o reconhecimento de todos os trabalhos realizados pelos colaboradores da área de informática, não são conhecidos pelo gerente da área.

1.5 Objetivos

1.5.1 Objetivo Geral

Desenvolver um sistema Helpdesk para o Departamento de Informática para a empresa conseguir controlar seu trabalho e melhorar o atendimento ao usuário.

1.5.2 Objetivos Específicos

- i. Implementar o Sistema Helpdesk na empresa Wiest S/A;
- ii. Desenvolver níveis de acesso ao sistema, nível Administrador e nível Usuário;
- iii. Criar uma área onde apenas o administrador terá acesso, para gerenciar o Helpdesk; (Adicionar, Editar e Remover Usuários, FAQ e Departamentos além de acesso a relatórios.);
- iv. Desenvolver uma área de abertura de chamados, área de exibição de todos os chamados abertos e área de meus chamados para o administrador assumir e resolver os chamados dos usuários;
- v. Desenvolver uma área de abertura de chamados, e outra de acompanhamento do chamado aberto para o usuário;
- vi. Área de acesso administrador para a geração de relatórios;
- vii. Incluir o Sistema Helpdesk à Intranet da empresa para atingir todos os usuários.
- viii. Prover a melhor organização das tarefas no Departamento de Informática da empresa.

- ix. Adquirir documentação sobre os problemas mais freqüentes e levá-los ao gerente da área, para análise e a efetuação de possíveis melhorias.
- x. Permitir atualização dos problemas ou soluções, de acordo com experiências ao longo do tempo.

1.6 Justificativa

Atualmente as empresas precisam sempre estar à frente do seu tempo para que possam manter um alto nível de competitividade, produtos de qualidade, atenção e suporte total ao cliente e garantir o bem estar dos colaboradores.

Para garantir estes fatores, essas empresas utilizam de recursos tecnológicos. Aplicativos para Internet estão cada vez mais em alta, pois faz com que a troca de informações seja mais eficaz e rápida.

A Wiest S/A possui uma unidade em Jaraguá do Sul, somando em média, 100 colaboradores que utilizam de alguma forma o sistema, e um computador, que está propício a erros e má utilização.

Devido ao grande número de usuários, em determinados casos, o atendimento torna-se desorganizado e desordenado, pois é feito somente via telefone, e em alguns casos via e-mail. Com uma empresa deste porte, é praticamente obrigatória a utilização de um sistema que controle e disponibilize esses chamados, fazendo assim com que a qualidade dos serviços melhore significativamente.

Este, não irá somente beneficiar os usuários do sistema e de microinformática, o Helpdesk Wiest S/A fará também com que os analistas de suporte da empresa possam se programar melhor para atender o chamado e trazer inovações para a empresa.

1.7 Metodologia

O projeto será elaborado por meio de pesquisa bibliográfica e pesquisa na internet, buscando livros, manuais, artigos e palestras referente à linguagem PHP e JavaScript, além do banco de dados MySQL. Será feito também, um *benchmarking*, conhecendo como o sistema Helpdesk funciona em outras empresas do mesmo porte para adquirir informações que possam ser úteis ao projeto.

Após a fase de pesquisas sobre a linguagem, inicia-se a análise do projeto, adquirindo o máximo de informação possível dos colaboradores da empresa Wiest S/A, aliado ao conhecimento e experiência dos analistas de suporte que já atendem estes usuários. A tecnologia utilizada será a UML, que servirá para criar um modelo lógico do sistema, utilizando diagramas de visualização.

No desenvolvimento do sistema, será utilizado a linguagem de programação PHP, JavaScript, HTML. O armazenamento de dados será realizado pelo banco MySQL.

Com a conclusão do desenvolvimento, o Helpdesk será testado e comprovado a funcionalidade total do sistema, será iniciada a etapa de treinamento aos usuários.

1.8 Cronograma

ATIVIDADES	01	02	03	04	05	06	07	08	09	10	11	12
Levantamento das informações para a elaboração do projeto	X	X	X									
Organização dos processos que o sistema irá prover para os usuários e gerentes			X	X	X							
Análise do problema utilizando diagramas da UML				X	X	X						
Aprendizado da linguagem e da técnica de criação de um sistema para web.			X	X	X	X	X	X	X	X		
Desenvolvimento do projeto, do Helpdesk da empresa.							X	X	X	X		
Elaboração do relatório final.			X	X	X	X	X	X	X	X		
Conclusões e considerações finais do projeto, testes e treinamento ao pessoal.											X	

Fonte: Ricardo Mondini (2007).

2 FUNDAMENTAÇÃO TEÓRICA

O objetivo da fundamentação teórica é buscar fundamentos que possam levar a resolução do problema apresentado no projeto de estágio. Os assuntos abordados serão: Sistemas de Informação, Desenvolvimento de sistemas, Metodologias de Desenvolvimento, Intranet, entre outros.

2.1 Sistemas de Informação

Um dos grandes desafios dos Sistemas de Informação é assegurar a qualidade e agilidade da informação, imprescindível para as corporações e seus gestores.

Um sistema de informação pode ser definido como um conjunto de homem, equipamentos, estrutura da organização e procedimentos. Tem como propósito transmitir informações para controle das operações. Ele tem um papel muito importante em organizações, pois sendo utilizado com sucesso, auxilia nos processos de tomada de decisão organizacional.

Laudon e Laudon (2001, p.5), define sistema de informação como:

Um conjunto de componentes inter-relacionados que coleta (ou recupera), processa, armazena e distribui informação para dar suporte à tomada de decisão e ao controle da organização. Além de apoiar, coordenar e controlar a tomada de decisão, os sistemas de informação também podem ajudar os gerentes e trabalhadores a analisar problemas, visualizar assuntos complexos e criar novos produtos.

O principal objetivo de um sistema de informação é disponibilizar para a organização as informações necessárias para que ela atue da melhor forma possível no ambiente que for necessário.

Para o objetivo dos sistemas de informação ser alcançado, as pessoas envolvidas (recursos humanos) é o elemento mais importante, os profissionais da área, como os analistas, programadores, operadores e usuários finais, todos os envolvidos na utilização dos sistemas de informação.

Os sistemas de informação utilizam recursos humanos, *hardware*, *software*, dados e redes de comunicações para coletar, transformar e espalhar as informações em uma organização.

Além dos recursos humanos citados acima, influem também o *hardware*, que são os computadores, o qual é executado processos de entrada de dados, processamento e saída de dados, o *software* que são os programas e procedimentos passados ao computador e aos usuários para que possa existir o processamento de dados. Estes são armazenados em bancos para que promova um conjunto organizado de informações, possibilitando assim, relatórios, e diversas consultas ao sistema. As redes de comunicações podem ligar as organizações e suas diversas filiais, num mesmo local ou em outros países, oferecendo assim informação *on-line* e atualizada para todos os envolvidos. (O'BRIEN, 2002)

Os sistemas de informações devem ajudar as organizações em suas estratégias e em tomadas de decisão, pois o emprego estratégico dos sistemas de informação está pautado na utilização desses sistemas como elemento que dá suporte a estratégias empresariais para enfrentar as forças competitivas atuantes no mercado (AUDY, ANDRADE E CIDRAL, 2005). Um sistema de informação pode tornar as organizações mais eficientes e competitivas.

2.2 Construindo Sistemas de Informação

O processo de desenvolvimento dos sistemas de informação é um momento crítico, pois além de envolver alta tecnologia, este processo envolve todos os funcionários da organização, inclusive gerentes que utilizam de algum tipo de sistema para auxiliar suas tarefas dentro da organização.

Para (LAUDON E LAUDON, 2001), o desenvolvimento de sistemas refere-se a todas as atividades que levem a resolução de algum problema, ou ao atendimento de alguma oportunidade de negócio para a organização. É essencialmente um tipo estruturado de resolução de problemas. Porém, decidir que novos sistemas construir, deve ser um componente essencial de um processo de planejamento organizacional. Comenta ainda que, deve-se desenvolver um plano de sistemas de informação que de suporte ao seu plano empresarial geral.

As organizações constroem diversos tipos de sistemas, de acordo com o que necessitam, pois a maioria possui valores que incentivam qualquer espécie de inovação, independentemente do benefício econômico direto para a empresa. (LAUDON E LAUDON, 2001)

De um ponto de vista econômico, a tecnologia de sistema de informação pode ser vista como um fator de produção que pode ser um livre substituto para capital e, se o custo da tecnologia cai, ela é substituta para o trabalho, que historicamente tem sido um gasto ascendente.

Para resolver isso, a tecnologia da informação poderia trazer informações diretamente de unidades operacionais para os gerentes, assim faria com que fosse eliminado, desse modo, os gerentes médios e os funcionários de apoio, tendo assim informações precisas e com um tempo de envio mais rápido. Novos sistemas de informação podem ser poderosos instrumentos para a mudança organizacional da organização. (LAUDON E LAUDON, 2001).

O desenvolvimento de sistemas envolve, em geral, um processo contínuo e constituído de diversas etapas, que são denominadas por muitos autores como *ciclo de vida do desenvolvimento de sistemas*. Este ciclo envolve cinco etapas:

- Avaliação: É a identificação de problemas e oportunidades;
- Análise: Período em que se determinarão quais os requisitos dos sistemas, “o que” o sistema deve fazer. A fase de análise tem por objetivo interpretar e definir uma estrutura para um problema ainda não estruturado. Diz respeito à eficácia do sistema;
- Projeto: Fase em que se determinará “como” o sistema funcionará para atender aos requisitos especificados na fase de análise. Diz respeito à eficiência do sistema, já com preocupações de “desempenho”. O objetivo desta fase é modelar o sistema de modo a implementar a solução idealizada na fase de análise, mas de acordo com os recursos tecnológicos existentes na empresa.
- Implementação é a fase em que será efetuada a construção do sistema de acordo com o modelo de funcionamento especificado na fase de projeto. Faz uso dos recursos tecnológicos existentes na empresa para atividades como a programação de computadores.
- Manutenção: É o monitoramento do sistema, a execução reparos e melhorias.

Já (LAUDON E LAUDON, 2001) ilustra o processo de desenvolvimento que pode ser visualizado na Figura 1, da seguinte forma:

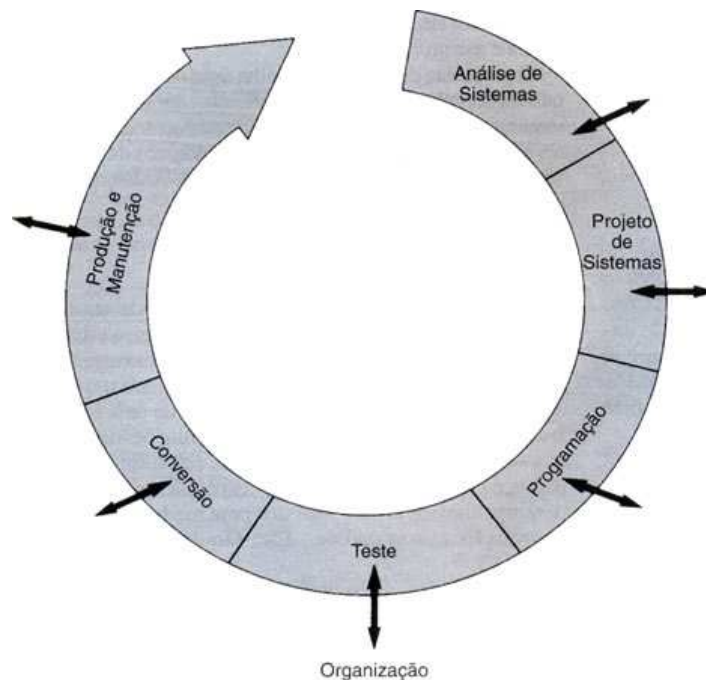


Figura 1 – Processos de desenvolvimento de software
Fonte: Laudon e Laudon (2001)

As atividades aqui, ocorrem em uma ordem seqüencial, algumas sendo repetidas, outras rodando simultaneamente, ou algumas podem precisar ser repetidas, dependendo de como será empregada. Deste modo, cada atividade envolve a participação de algum membro da organização. (LAUDON E LAUDON, 2001)

A Análise consiste em definir o problema que deve ser resolvido, e envolve todos os funcionários que utilizam algum tipo de sistema para auxílio de trabalho na organização; O Projeto de Sistemas mostra como o sistema cumprirá o objetivo, ele é um plano geral, ou modelo deste sistema. A Programação é o processo de tradução das especificações do projeto em *software* para o computador. A fase de Teste como o próprio nome já sugere, é o momento em que são efetuados testes completos para saber se o sistema produz os resultados esperados, corretamente. A conversão é o processo de mudança do sistema antigo para o novo sistema. E concluindo, o processo de Produção e Manutenção, que após o processo de conversão estar completo, o sistema entra em produção, onde será revisado pelos usuários do sistema para confirmar se os objetivos foram atendidos. Por fim, a manutenção, que proporciona mudanças de *software*, *hardware*, documentação ou procedimentos, para corrigir erros, melhorar a eficiência ou atender novas exigências. (LAUDON E LAUDON, 2001)

Além do ciclo de vida dos sistemas existem também outras abordagens tais como:

- Pacotes de *software* de aplicativos, que são programas vendidos comercialmente;
- Terceirização, onde os processos são realizados por pessoal externo a empresa organização;
- Desenvolvimento do próprio usuário, onde ele cria o sistema de informação necessário para tornar seu trabalho mais eficaz.

Independentemente do ciclo de vida do desenvolvimento, deve ser tomadas providências para que todos os processos sejam realizados corretamente, pois uma grande porcentagem de sistemas de informação falha no fornecimento de benefícios ou na solução de problemas para os quais foram programados porque o processo de mudança organizacional associado com a construção de sistemas não foi tratado corretamente, pois para um sistema ser bem-sucedido, é necessário um ótimo planejamento e gerenciamento do mesmo. (LAUDON E LAUDON, 2001).

2.3 Metodologia de Construção de Sistemas

Para se desenvolver sistemas de informação, é imprescindível a utilização de uma metodologia, ou seja, estipular o modo de fazer. Várias ferramentas e metodologias de desenvolvimento têm sido empregadas para facilitar a criação de sistemas, segundo Laudon e Laudon (2001, p.275) “Uma metodologia de desenvolvimento é uma coleção de métodos, um ou mais para cada atividade dentro de cada fase de um projeto de desenvolvimento de sistemas”.

As metodologias e ferramentas mais utilizadas incluem metodologias tradicionais estruturadas e o desenvolvimento de *software* orientado a objeto. A diferença entre os dois modelos de metodologia, a principal diferença está nas etapas de desenvolvimento. Este trabalho irá utilizar a metodologia de O.O.

Para Oliveira (1997) os principais objetivos da metodologia para sistemas de informação são:

- Criar uma ferramenta que possibilite o desenvolvimento de projetos na empresa, em harmonia com os princípios elementares da administração, tais como: planejamento, previsão, organização, decisão, comando, coordenação e controle;

- Promover o cumprimento de prazos, eficiência e qualidade do serviço, visando uma maior produtividade através da padronização das atividades de desenvolvimento e da racionalização dos controles e dos itens de documentação;
- Servir de apoio ao desenvolvimento de projetos e suas etapas, orientando a execução das atividades requeridas em todos os níveis e setores envolvidos, de uma forma padronizada e integrada;
- E por fim, estabelecer uma estrutura de documentação padronizada e compatível com a organização das fases e necessidades operacionais.

As ferramentas de análise estruturada utilizam representações de ferramentas gráficas para documentação. As principais ferramentas de documentação da análise estruturada, ainda segundo Oliveira (1997), incluem:

- Diagrama de fluxo de dados (DFD): que fornece um modo gráfico de representação de fluxo de dados de um sistema;
- Dicionário de dados: que é um conjunto organizado das definições lógicas de todos os nomes de dados mostrados no DFD;
- Especificações de processo: onde permite que o analista descreva a direção de negócios, representada por cada um dos processos de nível mais baixo dos diagramas detalhados do fluxo de dados;
- Diagrama de entidades relacionadas (E-R): que enfatiza os principais objetos ou entidades de dados armazenados, com os quais o sistema deve lidar, bem como a relação entre esses objetos.

As principais características da análise estruturada é que ela é particionada, gráfica, e lógica.

O desenvolvimento de *software* orientado a objeto difere das metodologias tradicionais pela substituição do foco de modelos de processos empresariais separadamente e dados para combinar dados e procedimentos em objetos unificados. (LAUDON E LAUDON, 2001). Neste método o sistema visto como um conjunto de classes e objetos relacionados. Os objetos são criados, normalmente, e salvos para serem utilizados em outras aplicações.

Estes objetos já criados são de fácil reutilização, fazendo assim com que o tempo e o custo de criação de *software* se tornem menor, obviamente só terá reduções após já existir uma biblioteca de objetos criados.

Segundo SICA (2006), pode-se definir alguns termos da programação orientada a objeto como:

Classe: Representa um conjunto de objetos com características afins. Uma classe define o comportamento dos objetos, através de métodos, e quais estados ele é capaz de manter, através de atributos. Exemplo de classe: Os seres humanos.

Objeto: é uma instância de uma classe. Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos. Exemplo de objetos da classe Humanos: João, José, Maria.

Atributos: são características de um objeto. Basicamente a estrutura de dados que vai representar a classe. Exemplos: Funcionário: nome, endereço, telefone, CPF,; Carro: nome, marca, ano, cor, ...; Livro: autor, editora, ano. Por sua vez, os atributos possuem valores. Por exemplo, o atributo cor pode conter o valor azul.

Métodos: definem as habilidades dos objetos.

Mensagem: é uma chamada a um objeto para invocar um de seus métodos, ativando um comportamento descrito por sua classe. Também pode ser direcionada diretamente a uma classe.

Herança: (ou generalização) é o mecanismo pelo qual uma classe (sub-classe) pode estender outra classe (super-classe), aproveitando seus comportamentos (métodos) e estados possíveis (atributos). Há Herança múltipla quando uma sub-classe possui mais de uma super-classe. Essa relação é normalmente chamada de relação "é um". Um exemplo de herança: Mamífero é super-classe de Humano. Ou seja, um Humano é um mamífero.

Associação: é o mecanismo pelo qual um objeto utiliza os recursos de outro. Pode tratar-se de uma associação simples "usa um" ou de um acoplamento "parte de". Por exemplo: Um humano usa um telefone. A tecla "1" é parte de um telefone.

Encapsulamento: consiste na separação de aspectos internos e externos de um objeto. Este mecanismo é utilizado amplamente para impedir o acesso direto ao estado de um objeto (seus atributos), disponibilizando externamente apenas os métodos que alteram estes estados. Exemplo: você não precisa conhecer os detalhes dos circuitos de um telefone para utilizá-lo. A carcaça do telefone encapsula esses detalhes, provendo a você uma interface mais amigável (os botões, o monofone e os sinais de tom).

Abstração: é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais. Em modelagem orientada a objetos, uma classe é uma abstração de entidades existentes no domínio do sistema de software.

Este trabalho irá utilizar a uma metodologia de desenvolvimento baseada nos princípios da engenharia de sistemas e principalmente o Processo Unificado (*Unified Process*), utilizando como linguagem de modelagem orientada em objetos, a UML (*Unified Modeling Language*).

O Processo Unificado é um processo que define o ciclo de vida do projeto, e visa desenvolver um produto de *software* que atenda as necessidades do usuário. Segundo Scott (2003) a vida de um sistema de *software* é representado com uma série de ciclos. Cada ciclo possui quatro fases, a de Concepção, cujo objetivo é estabelecer a viabilidade do sistema proposto, ou seja, definir o escopo do sistema, identificar riscos e inicia a análise econômica do projeto, para saber se vale a pena fazer o projeto; a de Elaboração, cujo objetivo é de definir qual a capacidade para o desenvolvimento do novo sistema, sabendo das restrições financeiras, tempo, entre outros tipos de restrições.

As principais tarefas executadas nesta fase são a de adquirir a maioria dos requisitos funcionais válidos que restaram abordar riscos significativos e elaborar um plano que contenha detalhes suficientes para orientar sua próxima fase, que é a de Construção, cujo objetivo é construir um sistema capaz de funcionar bem em ambientes de testes dos clientes. Durante esta construção, a equipe tende executar tarefas de modo a ter sempre certeza que a viabilidade do sistema estará sempre evidente; e por fim, a fase de Transição cujo principal objetivo é entregar o sistema completo, e totalmente funcional para o cliente.

Passando pelas quatro fases, está sendo feito o ciclo de desenvolvimento e resulta na criação do software. Neste projeto será integrado o ciclo de vida de sistemas e a UML, pois assim pode-se garantir o comprimento dos cronogramas e da qualidade do site.

O Processo Unificado especifica que a arquitetura do sistema em construção como alicerce fundamental sobre o qual ele se erguerá deve ser uma das principais preocupações da equipe de projeto e também de arquitetura, levando isso em consideração, este projeto será desenvolvido integrando o ciclo de vida de sistemas e a visão técnica da UML, com um processo de cinco etapas, denominado por Scott (2003) como cinco *workflows*, são eles:

1. Requisitos
2. Análise

3. Projeto
4. Implementação
5. Teste

Cada um destes *workflow's* é um conjunto de atividades que vários membros do projeto executam.

2.4 UML

A UML (*Unified Modelin Language*) foi desenvolvida para ajudar a desenvolvedores de software a criar modelos que permitam visualizar o sistema, especificar a estrutura e o comportamento, construir e documentar as decisões tomadas durante o projeto.

Para Booch, Rumbaugh e Jacobson (2000, p.13),

A UML é uma linguagem-padrão para elaboração da estrutura de projetos de software. A UML poderá ser empregada para visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software. [...] é adequada para a modelagem de sistemas, cuja abrangência poderá incluir sistemas de informação corporativos a serem distribuídos a aplicações baseados em Web e até sistemas complexos embutidos de tempo real. É uma linguagem muito expressiva, abrangendo todas as visões necessárias ao desenvolvimento e implantação desses sistemas.

Os elementos estruturais da UML podem ser representados da seguinte forma:

A representação de uma classe pode ser vista na Figura 2:



Figura 2 – UML – Representação de uma Classe.
Fonte – Ricardo Mondini (2007)

Um objeto é representado da seguinte forma, na Figura 3:

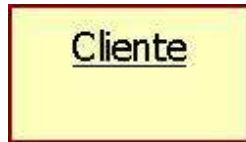


Figura 3 – UML – Representação de um Objeto.
Fonte – Ricardo Mondini (2007)

A Figura 4 mostra a representação dos relacionamentos.

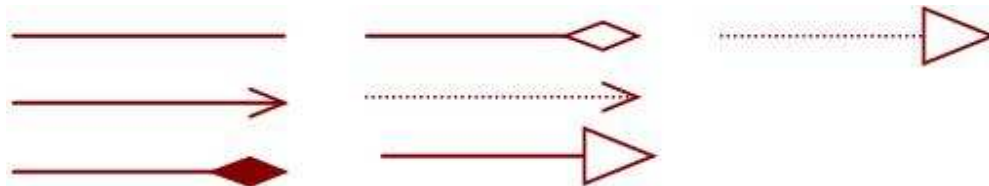


Figura 4 – UML – Representação dos relacionamentos.
Fonte – Ricardo Mondini (2007)

Os elementos comportamentais da UML podem ser representados da seguinte forma:

Os casos de uso são representados na Figura 5 da seguinte forma:

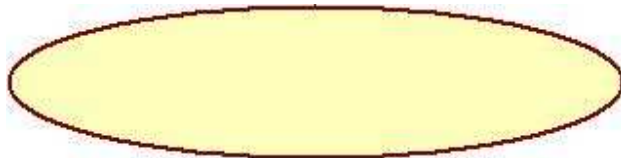


Figura 5 – UML – Casos de Uso
Fonte – Ricardo Mondini (2007)

Os atores são representados na Figura 6 da seguinte forma:

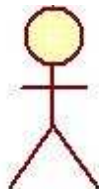


Figura 6 – UML – Representação de um Ator
Fonte – Ricardo Mondini (2007)

A UML é uma linguagem de modelagem, esta modelagem permite que o sistema seja mais bem compreendido, e visualizado. Para modelar partes mais específicas do sistema, o UML utiliza diagramas, que permite visualizar o programa de diferentes perspectivas. Conforme Booch, Rumbaugh e Jacobson (2000, p.25) “Um diagrama é a apresentação gráfica de um conjunto de elementos, geralmente representados como gráficos de vértices (itens) e arcos (relacionamentos)”.

Entre muitos diagramas oferecidos pelo UML, os mais importantes são:

- a. Diagramas de Classe: Mostra o conjunto de classes, interfaces e colaborações e seus relacionamentos, para documentar a arquitetura do sistema;
- b. Diagramas de Casos de Uso: Exibe um conjunto de casos de uso e atores e seus relacionamentos, para documentar os requisitos funcionais;
- c. Diagramas de Seqüências: É um digrama de interação entre objetos, que dá ênfase na ordenação temporal das mensagens (dentro de uma linha de tempo);
- d. Diagramas de Componentes: Focaliza os componentes do sistema, bancos de dados, código-fonte etc.

Este projeto irá focalizar e utilizar os Diagramas de Classe, Diagramas de Casos de Uso, de Interação e de Seqüência.

2.4.1 Diagramas de Classe

Os Diagramas de Classe é utilizado para expor os objetos e seus relacionamentos dentro do sistema desenvolvido. Segundo Booch, Rumbaugh e Jacobson (2000, p.104) “Um diagrama de classe é um diagrama que mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos”.

As classes definem os tipos de objetos que existem no sistema. Classes podem ter atributos que são geralmente membros de dados primitivos de objetos, e operações definidoras de métodos que podem ser aplicados sobre objetos. (LEE E TEPFENHART, 2001).

O diagrama de classe compartilha as mesmas propriedades de todos os outros diagramas, e nunca pode ser utilizado sozinho. Cada uma dessas classes trabalha em conjunto, em colaboração umas com as outras, para realizar algo que somente todas juntas pode prover. Uma classe é representada com um retângulo subdividido em três partes que armazenam o Nome da classe, a Lista de atributos e a Lista de operações. Esta é a noção básica dos diagramas de classe.

Quanto aos seus relacionamentos, são responsáveis por mostrar os vínculos existentes entre objetos ou entre classes. Existem basicamente três tipos de relacionamentos nos diagramas de classe:

- Generalização/Especificação: indica o relacionamento entre um elemento geral, e um específico;
- Associação: um relacionamento que descreve um conjunto de vínculos, onde um objeto tem conhecimento de um outro objeto para serviços específicos;
- Associação Todo-Parte: são divididos em dois tipos, a agregação, onde o objeto que representa o todo é associado a um conjunto de objetos que representam suas partes, e a composição, onde o objeto parte pode pertencer a somente um todo.

Para representar os relacionamentos são utilizadas setas entre as classes, cada qual especificando o relacionamento.

2.4.2 Diagrama de Caso de Uso

Um caso de uso envolve a interação dos atores com o sistema. Para Booch, Rumbaugh e Jacobson (2000, p.218) “Um caso de uso descreve um conjunto de seqüências, cada uma representando a interação de itens externos ao sistema (seus atores) com o próprio sistema (e suas principais abstrações)”. Já Larman (2000, p.68), define caso de uso como “um documento narrativo que descreve a seqüência de eventos de um ator (agente externo) que usa um sistema para completar um processo”. Em um modelo de caso de uso, existem os atores e casos de uso.

Um ator, conforme Booch, Rumbaugh e Jacobson (2000, p.219) “representa um conjunto coerente de papéis que os usuários dos casos de uso, desempenham quando interagem com esses

casos“. Para Larman (2000, p.71) “Os atores são usualmente os papéis desempenhados por seres humanos, mas podem ser qualquer tipo de sistema [...]”.

A modelagem de um diagrama de casos de uso deve representar o relacionamento entre atores e os casos de uso. (LARMAN, 2000)

2.4.3 Diagramas de Interação

De acordo com Furlan (1999), o diagrama de interação mostra uma seqüência de troca de mensagens entre objetos. Geralmente eles são utilizados quando é necessário visualizar o comportamento de objetos dentro do caso de uso. Isto é necessário para obter informações sobre o comportamento dos objetos do sistema nos diversos casos de uso. Booch, Rumbaugh e Jacobson (2000, p.205) citam que “uma interação é um comportamento que compreende um conjunto de mensagens trocadas entre um conjunto de objetos em determinado contexto para realização de um propósito”.

Para Larman, (2000), os diagramas de interação, são os mais importantes, pois os dados que constam neles, são de fundamental importância na fase de desenvolvimento, pois nele se encontram os aspectos dinâmicos do sistema.

A modelagem dos aspectos dinâmicos do sistema é feita pela utilização das interações. Uma interação define estaticamente o conjunto de estados para o seu comportamento pela inclusão de todos os objetos que trabalham em conjunto para execução de determinada ação. (BOOCH, RUMBAUGH E JACOBSON, 2000):

Um diagrama de interação é utilizado para estabelecer a realização de um caso de uso através de uma seqüência de interação.

2.4.4 Diagramas de Seqüência

Um diagrama de seqüência mostra a colaboração dinâmica de vários objetos de um sistema. Para Booch, Rumbaugh e Jacobson (2000, p.96), “Um diagrama de seqüência é um diagrama de interação que dá ênfase a ordenação temporal de mensagens. Um diagrama de seqüência mostra um conjunto de objetos e as mensagens enviadas e recebidas pelos objetos”. Desta forma é possível fazer a documentação dos detalhes de um caso de uso.

Os diagramas de seqüências são representados graficamente da seguinte forma: Uma tabela que mostra os objetos distribuídos no eixo X e as mensagens, no eixo Y, em ordem crescente no tempo. (RUMBAUGH E JACOBSON, 2000).

Os diagramas de seqüência podem mostrar todas as interações que ocorrem através da fronteira do sistema e entre objetos dentro dele para satisfazer os requisitos definidos para um ou mais caminhos por meio de um caso de uso. Desta forma, é possível concluir que este diagrama mostra uma seqüência de mensagens e deve ser usado para especificações de tempo real para cenários complexos. (LEE E TEPFENHART, 2001).

2.5 Levantamento de Requisitos

Esta é a primeira etapa de uma metodologia de desenvolvimento de sistemas de informação, e também uma das mais importantes para o projeto, pois com um levantamento de requisitos correto, é possível garantir um desenvolvimento eficaz, atendendo as expectativas solicitadas e evidenciando um processo de desenvolvimento organizado.

Para Larman(2000, p.60):

Requisitos são uma descrição das necessidades ou dos desejos para um produto. O objetivo básico da fase de requisitos é identificar e documentar o que é realmente necessário em uma forma que comunica claramente essa informação ao cliente e aos membros da equipe de desenvolvimento.

Segundo Scoot (2003) essa etapa é geralmente difícil, pois o cliente e o desenvolvedor falam uma “língua” diferente em relação aos propósitos práticos, além de ser difícil definir

exatamente quais os objetivos, pois na maioria das vezes nem o cliente sabe o que necessita realmente, fazendo assim com que entre frequentemente em contradição.

2.5.1 Conceito de Requisitos

Segundo Martins (2002) os requisitos expressam características, propriedades ou comportamentos a serem atendidos por um produto, processo ou serviço. No processo de desenvolvimento de sistemas de informação, os requisitos desempenham o mesmo papel, ou seja, ele atende as especificações passadas pelo usuário ou processo. Os requisitos definem as características técnicas, funcionais e estruturais básicas do sistema conforme observadas pelo cliente, apontando o desenvolvimento na direção na direção correta. O levantamento destes requisitos deve ser feito pela equipe de projetos em conjunto com os usuários da área que será desenvolvido o projeto.

Os requisitos funcionais descrevem as funções de tratamento de informação que o sistema deve oferecer. Os requisitos não funcionais podem ser classificados nas seguintes categorias: confiabilidade, precisão dos resultados, desempenho, restrições operacionais, restrições físicas, portabilidade e outras. (SCHMITZ E SILVEIRA, 2000),

2.5.2 Atividades do Levantamento de Requisitos

Para o que foi exigido, ser entendido de forma mais clara, são criados modelos, o problema é dividido em partições e são desenvolvidas representações que apresentem os requisitos e os detalhes da implementação.

O ponto mais importante desta fase é a localização dos casos de uso e os atores, utilizando diagramas de casos de uso para representar seus relacionamentos.

Para Fowler e Scott(2000) uma das coisas mais importantes a ser feita na fase de elaboração é descobrir todos os casos de uso potenciais para o sistema que você esta construindo.

Na prática, certamente não será possível descobrir todos. Entretanto tem que se descobrir a maioria particularmente os mais importantes e mais arriscados.

O modo de representação dos relacionamentos casos de uso (funções do sistema) e dos atores (usuário e outros tipos de sistemas que interagem com o sistema) são feitos através do diagrama de caso de uso.

2.5.2.1 Elaborar Modelo de Negócio

Normalmente, quando os casos de uso empregados na fase de levantamento de requisitos são muito complexos, o modelo de negócio é utilizado para possibilitar um melhor entendimento do contexto de negócio no qual está sendo colocado o sistema. Esta atividade envolve construir um modelo de casos de uso e um de objetos de negócio para que se possa entender o funcionamento do negócio que esta sendo modelado. (SCOTT, 2003)

O principal objetivo do modelo de negócio é capturar a relação entre conceitos, assim como a estrutura das informações contidas nos conceitos. Normalmente o modelo de negócio é representado através de um MER (Modelo Entidade Relacionamento). (MARTINS, 2002)

2.5.2.2 Descobrir Atores e Casos de Uso

Esta atividade envolve descobrir os atores humanos e não-humanos que estarão interagindo com o sistema e reunir um conjunto de casos de uso que reflitam o comportamento que estes atores terão em conjunção com o sistema. (SCOTT, 2003)

Para muitos autores, o processo de modelagem pode ser iniciado a partir da descoberta de uma lista de atores ou de casos de uso, sendo que, se é encontrado uma lista de atores, pode-se chegar à lista de casos de uso, e vice-versa. Há a possibilidade de se encontrar atores, se forem seguidos alguns critérios, entre eles:

- Quem está interessado em certo requisito;
- Quem usa, instala, carrega, mantém e desliga o sistema;
- Quem fornece informações ao sistema;
- Quem irá utilizar as informações do sistema.

Os modelos de negócio são uma ótima fonte de atores e casos de uso.

2.5.2.3 Prototipar a Interface com o Usuário

A finalidade desta etapa é fazer um protótipo de interface entre usuário e sistema. Para Scott (2003) os protótipos geralmente ajudam a encontrar atores e casos de uso e a escrever o texto para casos de uso. Eles também servem como uma excelente fonte de classes e atributos pertencentes a estas classes.

2.5.2.4 Detalhar Casos de Uso

Esta etapa, como o nome sugere, detalha todos os casos de uso, desde como ele inicia, termina e interage com os atores. Conforme Scott (2003), esta atividade envolve escrever o texto para o curso básico e para os cursos alternativos de ação para um caso de uso.

Os casos de uso possuem muitos cenários, deve-se ser escolhido o cenário mais simples para descrever o caso de uso. O detalhamento irá apresentar um fluxo normal indicando a situação mais comum, e fluxos alternativos que indicam desvios ou exceções.

2.6 Análise

O principal objetivo desta fase é saber o que os clientes irão precisar, e utilizar este conhecimento para efetuar o desenvolvimento do projeto e da implementação.

Larman (2000, p.29):

A análise enfatiza uma investigação do problema, de como uma solução é definida. [...] se o desejo é ter um novo sistema de informação de bibliotecas computadorizados, quais são os processos de negócio relacionados com seu uso? Para desenvolver uma aplicação também é necessário ter descrições de alto nível e descrições detalhadas da solução lógica e de como ela atende os requisitos e as restrições.

Para (SCOTT, 2003), o objetivo fundamental é entender realmente os requisitos e usar este entendimento para conduzir o desenvolvimento do projeto e da implementação, pois o resultado desta fase é a geração do modelo de análise, que ajuda a achar outros objetivos para o sistema que está sendo desenvolvido.

O modelo de análise faz abstrações e evita resolver problemas que deveriam ser abordados na fase de *design*. Estes problemas geralmente estão relacionados à tecnologia, como linguagem, plataforma, banco de dados e outros.

Nesta fase, só será efetuada a modelagem de classe que pertençam ao escopo principal do problema do sistema, ou seja, as classes técnicas que gerenciam banco de dados, interface, comunicação, concorrência, já as outras, não estarão presentes neste diagrama.

2.6.1 Análise arquitetônica

Consiste em analisar os elementos da modelagem, para saber se será necessário organizar seus itens em pacotes, ou conforme a linguagem de alguns autores, subsistemas.

Conforme Scott (2003), esta atividade envolve criar esboços do modelo de análise e da arquitetura como um todo. O modelo de análise é composto por pacotes de análise e cada um destes pacotes deve possuir uma breve descrição elaborada, indicando seus principais objetivos. Inicialmente é necessário identificar os primeiros pacotes que irão possuir as realizações de análise de casos de uso e as classes de análise, que serão tratadas nos próximos dois tópicos.

2.6.2 Análise de Casos de Uso

Nesta atividade geralmente é elaborado um diagrama de seqüência para modelar como os objetos do sistema irão interagir. Neste diagrama irá existir apenas a troca de mensagens entre as classes da análise.

Scott (2003), afirma que esta atividade envolve construir uma realização de análise para um caso de uso. O diagrama de robustez resultante identifica objetos que pertencem as classes de análise, e os relacionamentos entre análises.

2.6.3 Análise de Classes

Neste processo são analisadas as classes do diagrama de classes, podendo ser identificadas novas classes, atributos, operações e/ou relacionamentos. O resultado dessa análise implicará um diagrama de classes detalhado.

Conforme Scott (2003, p.54) “Esta atividade envolve expandir a definição de uma classe de análise que participa de uma ou mais realizações de análise de casos de uso”. Para que se possam construir classes coesas e fracamente acopladas, é necessário, identificar as responsabilidades que a classe tem por meio de um conjunto de realizações de casos de uso e tomar decisões baseadas em qual classe tem quais responsabilidades.

2.7 Banco de dados

Banco de dados pode ser definido como um conjunto de dados relacionados entre si, referentes a um assunto em comum e organizados de maneira útil, servindo como base para todas as informações necessárias para que se tirem conclusões e tome decisões.

Para Kroenke (1999, p.12) “Um bando de dados é uma coleção auto-descritiva de registro integrados.” Ele é auto-descritivo pois contém uma definição de sua própria estrutura, chamado

dicionário de dados. A finalidade de um banco de dados é auxiliar as pessoas a armazenarem de forma organizada informações sobre as coisas.

Pode-se afirmar ainda que: Um banco de dados é composto de uma série de registro. Cada registro é um conjunto de campos e cada campo contém informações ou dados pertinentes. Um único registro pode ser composto de qualquer número de campos de diferentes tipos e que uma série de registros em um banco de dados, todos com os mesmos campos é chamado de tabela.

2.7.1 Sistema Gerenciador de Banco de Dados

Um Sistema Gerenciador de Banco de Dados ou SGDB é um conjunto de programas utilizados para criar e manter um banco de dados. Pode-se citar como exemplos de SGDB:

WIKIPEDIA (2007)

- MySQL, gratuito;
- PostgreSQL, gratuito;
- Firebird, gratuito;
- Oracle, proprietário;
- DB2, proprietário;
- SQLServer, proprietário;

As vantagens de se utilizar um SGDB são: WIKIPEDIA (2007)

- Consistência: minimiza incoerências;
- Integração: evita redundância;
- Independência de Dados: facilita modificações;
- Sistemas Distribuídos: organização;
- Compartilhamento: controle de concorrência;
- Representação de Relacionamentos Complexos;
- Segurança: restrições específicas de acesso;

2.7.2 Design do Banco de Dados

Ao criar um banco de dados sem muito planejamento, pode suprir as necessidades de pequenas aplicações, porém se o banco de dados tiver a necessidade de ser mais robusto, planejamento e *design* serão necessários, principalmente para poupar tempo e dor de cabeça.

Pode-se também entender *design* de banco de dados como modelagem de dados, que nada mais é do que criar uma estrutura de dados que irá representar um conjunto de informações para que se possam analisar os dados com facilidade.

Para Yarger, Reese E King (2000) um dos principais objetivos do *design* do banco de dados é eliminar a redundância. Para executar esta tarefa, é utilizada a técnica de normalização de dados.

2.7.2.1 Normalização dos Dados

O objetivo da normalização dos dados é remover certos tipos de redundância de dados e então evitar erros na atualização. São dificuldades com as operações de inserir, atualizar e excluir em um banco de dados devido à estrutura. (YARGER, REESE E KING, 2000).

Para Maslakowski (2002, p.49):

Normalização é um conjunto de regras para ajudar projetistas de banco de dados a desenvolver um esquema que minimiza problemas de lógica. Cada regra baseia-se na regra anterior. A normalização foi adaptada porque o velho estilo de colocar todos os dados em um lugar, como arquivo ou tabela de banco de dados, era ineficiente e levava a erros de lógica ao tentar manipular os dados contidos.

O processo de normalização atua no conceito de formas normais, quais serão dispostas em tópicos: (YARGER, REESE E KING, 2000).

- Primeira forma normal (1FN): É dito que uma entidade está na primeira forma normal quando todos os atributos têm um valor único. Todas as colunas que se repetem devem ser eliminadas e colocadas em tabelas separadas.
- Segunda forma normal (2FN): É dito que uma entidade está na segunda forma normal se ela já estiver na 1FN e todos os atributos não identificados forem dependentes do identificador único da entidade, ou seja, todos os atributos não chave, tem que ser totalmente dependente da chave primária.

- Terceira forma normal (3FN): É dito que uma entidade está na terceira forma normal se ela já estiver na 2FN e nenhum atributo não identificado forem dependentes de nenhum outro atributo não identificado, ou seja, todos os atributos não chave tem de ser dependentes não transitivos da chave primária.

Utilizando-se da normalização o resultado serão as entidades que recebem o nome de Modelo Entidade e Relacionamento

2.7.2.2 MER – Modelo Entidade-Relacionamento

O MER é um modelo de dados conceitual de alto nível e seus conceitos foram projetados para estar o mais próximo possível da visão do usuário sobre os dados, sem se preocupar como estes dados estão sendo realmente armazenados.

Uma entidade é algo que pode ser identificado no ambiente de trabalho do usuário, alguma coisa que ele quer localizar. Ex: Funcionário ou um cliente. E um relacionamento é a associação de entidades entre si. (KROENKE, 1999).

Alguns conceitos importantes:

- Entidade: qualquer objeto real ou abstrato, onde foram especificados os dados que são de interesse para armazenamento. São coisas identificáveis e de importância para o usuário;
- Atributo: descrição de uma entidade;
- Relacionamento: as associações entre entidades;
- Chave primária: chave utilizada para identificar as diferentes ocorrências da entidade, possuindo valor único;
- Chave estrangeira: uma chave que estabelece o relacionamento entre duas tabelas.

A maioria dos relacionamentos é binária. Existem três tipos binários que são:

1:1 (Uma-para-Um): Cada elemento de uma entidade relaciona-se apenas com um elemento de outra entidade;

1:N (Um-para-Muitos): Um elemento de uma entidade relaciona-se com vários elementos de outras entidades;

N:N(Muitos-para-Muitos): Vários elementos de uma entidades se relacionam com vários elementos de outra entidade.

Utilizando-se do MER, é possível conseguir um esquema que mostra a lógica global do banco de dados.

2.8 Linguagens de Programação

Todos os software de sistemas e aplicativos são escritos em códigos chamados linguagens de programação, e os sistemas para *web* não são diferentes. A principal função de uma linguagem de programação é permitir que o programador desenvolva as instruções que constituem o programa do computador, para que possa ser executado o processamento, seja para atingir objetivo ou solucionar um problema. Para Norton (1994, p.478), a linguagem de programação: “instrui o computador para que este execute as ações desejadas, como a apresentação de informações na tela, a leitura de dados no disco, a execução de cálculos numéricos e a aceitação de dados a partir do teclado”.

Antigamente, o ato de programar era extremamente complexo, fazendo com que os programadores não tirassem proveito das vantagens de processamento.

As linguagens de programação usam um conjunto de símbolos que tem significado especial, e também cada uma tem sua própria série de regras, chamada de sintaxe da linguagem (STAR, 1998).

Assim como os computadores as linguagens de programação evoluíram com o tempo, tornando mais acessível à programação e a criação de programas e sistemas. Esta evolução foi impulsionada pelo desejo de aplicar eficientemente o poder de processamento da informação a mais ampla variedade de atividades de solução de problemas. (STAIR, 1998). Cada linguagem foi projetada para resolver um determinado tipo de problema, com isso, serão citados as principais gerações das linguagens.

2.8.1 Primeira Geração – Linguagem de máquina

No início do desenvolvimento da computação, na década de 40, surgem a linguagem de máquina, que também são conhecidas como linguagens de primeira geração, o nível mais básico das linguagens de programação. A linguagem de máquina é considerada de baixo nível pois não existe um esquema de codificação do programa menos sofisticado do que símbolos binários 1 e 0. Esta programação é extremamente complexa, pois exige que as instruções sejam escritas na forma de sequência de dígitos binários ou outros sistemas numérico, além de exigir um conhecimento detalhado das operações internas de cada tipo específico de CPU. (O'BREIN, 2002).

Apesar de sua complexidade, esta é a única forma de enviar instruções diretamente para a CPU.

2.8.2 Segunda Geração – Linguagem *Assembly*

Devido à complexidade da linguagem de máquina, surgiu a segunda geração de linguagens, que foram desenvolvidas para diminuir a dificuldade na gravação de programas em linguagem de máquina.

A linguagem *Assembly* trouxe símbolos que são melhor compreendidos que o código binário. Esta segunda geração recebeu o nome de linguagem *Assembly* por causa dos programas de sistema usados para traduzi-los para código de máquina, chamados de programas assembler ou montadores. Estes montadores traduzem as instruções na linguagem *Assembly* para linguagem de máquina. O'Brein (2002).

2.8.3 Terceira Geração – Linguagem de alto nível

Da mesma forma que a segunda geração, a linguagem de alto nível, continuou a tendência do uso de códigos simbólicos e deixou de dar ao computador instruções específicas de como completar uma operação. Segundo STAIR (2002), uma linguagem de programação de alto nível

usa instruções e comandos semelhantes à língua inglesa. As linguagens desta geração se tornaram ainda mais fáceis de aprender e de usar do que a linguagem *Assembly*. Pode-se citar algumas linguagens que apareceram nesta geração:

- BASIC;
- COBOL;
- FORTRAN;
- PASCAL.

À medida que as linguagens distanciam-se mais a linguagem de máquina, mais processamento e memória são necessários, pois exigem uma maior tempo do computador para traduzir a programação em linguagem de máquina. O’Brein (2002).

2.8.4 Quarta geração

Assim como as linguagens de alto nível, as linguagens de quarta geração são parecidas com o inglês, e ainda mais, tornando o aprendizado ainda mais fácil e chamado à atenção de mais pessoas para tornarem-se programadores. Podem-se citar algumas linguagens que apareceram nesta geração:

- DELPHI;
- PROGRESS;
- VISUAL BASIC;
- C++.

Estas linguagens enfatizam os resultados desejados em vez de como as declarações dos programas são escritos, visto que são menos procedurais e mais convencionais que as anteriores. Alguns dos principais aspectos das linguagens desta geração são sua capacidade de consultas e manipulação de banco de dados, capacidade de geração de código e capacidades gráficas. Laudon & Laudon (2001).

2.8.5 Linguagem Orientada a Objetos

A linguagem orientada a objeto divide o programa em objetos que contém os dados e também as instruções e, diferentemente dos métodos de desenvolvimento tradicionais, estas linguagens não tratam os dados e os procedimentos de forma diferente.

Segundo Laudon e Laudon (2001), o principal benefício deste modo de programação é a reutilização de códigos, ou seja, o código de um objeto pode ser utilizado em outros programas, tornando a programação mais rápida e padronizada.

Para David (2007):

"O aprendizado do paradigma de programação orientada a objetos é bem mais complicado no início do que os velhos sistemas procedurais. Para começar a programar é necessário ter estabelecido uma série de conceitos bastante complexos. Já na programação procedural tradicional, basta decorar meia dúzia de comandos e você já consegue fazer um programa simples".

David (2007) comenta também que outro ponto fraco da POO é o desempenho obtido, pois consome mais memória e processamento, tornando praticamente impossível ter um desempenho em tempo de execução superior a linguagens não orientadas a objetos.

As principais características da orientação a objetos são:

- Herança: Uma classe pode estender todas as características de outra e adicionar algumas coisas a mais. Uma classe pode sempre ter vários filhos, mas normalmente as linguagens de programação orientadas a objetos exigem que cada classe filha tenha apenas uma classe pai. (David, 2007)
- Polimorfismo: O conceito de polimorfismo pode ser explicado como, objetos diferentes, respondem de forma diferente a uma mesma mensagem. (David, 2007)

2.9 Ferramentas de Desenvolvimento

Para se construir um sistema utilizando os métodos descritos, é necessário transformá-los em linguagem de máquina através de ferramentas de desenvolvimento já existentes no mercado.

Atualmente, existem varias linguagens que podem suprir as necessidades de criação do Helpdesk. Neste projeto serão utilizadas como linguagem de programação, o PHP, o Javascript e o HTML, como sistema de gerenciamento de banco de dados, o MySQL, e como serviço de servidor o Apache. Todas as ferramentas utilizadas são livres.

2.9.1 Servidores para internet

Um servidor web é um computador, ou programa de computador responsável por aceitar pedidos HTTP de clientes.

HTTP é a sigla em português, de Protocolo de Transferência de Hipertexto, um protocolo de Aplicação do Modelo OSI utilizado para transferência de dados na rede mundial de computadores, a World Wide Web. Também é possível transferir dados de hiper-mídia, como imagens, sons e textos. Wikipedia (2007).

O processo se inicia com a conexão entre o computador onde está instalado o servidor web e o computador do cliente. Como na web não é possível prever a que hora se dará essa conexão, os servidores web precisam estar disponíveis dia e noite. A partir daí é processado o pedido do cliente, e conforme as restrições de segurança e a existência da informação solicitada, o servidor devolve os dados. Wikipedia (2007).

Os servidores web também são capazes de executar programas e scripts, desta forma, obtendo uma interação maior entre, web e usuário.

São exemplos de alguns WebServers:

- Covalent Enterprise Ready Server, baseado no Apache HTTP Server;
- Oracle HTTP Server, baseado no Apache HTTP Server;
- Internet Information Services (IIS) da Microsoft, incluso no Windows XP;
- Sun Java System Web Server da Sun Microsystems, antigo Sun ONE Web Server, iPlanet Web Server, and Netscape Enterprise Server;
- Fnord;
- IBM HTTP Server (baseado no Apache HTTP Server), antigo Domino Go Webserver;

- Light HTTP Server (lighttpd);

2.9.2 Servidor Apache

O servidor Apache é o mais bem sucedido servidor web livre. Foi criado em 1995 por Rob McCool. Numa pesquisa realizada em dezembro de 2005, foi constatado que a utilização do Apache supera 60% nos servidores ativos no mundo. Wikipedia (2007).

O servidor é compatível com o protocolo HTTP e suas funcionalidades são mantidas através de uma estrutura de módulos, podendo inclusive o usuário escrever seus próprios módulos — utilizando a API do software. Wikipedia (2007).

O Apache é um servidor extremamente seguro, e para garantir segurança nas transações HTTP, o servidor dispõe de um módulo chamado `mod_ssl`, o qual adiciona a capacidade do servidor atender requisições utilizando o protocolo HTTPS que utiliza uma camada SSL para criptografar todos os dados transferidos entre o cliente e o servidor, provendo maior grau de segurança, confidencialidade e confiabilidade dos dados. A camada SSL é compatível com certificados X.509, que são os certificados digitais fornecidos e assinados por grandes entidades certificadoras no mundo. Wikipedia (2007). Internet Banking's de todo o mundo utilizam-se deste protocolo para manter os dados de seus clientes em sigilo.

2.9.3 PHP

O PHP é uma linguagem de programação de computadores interpretada, o que a torna ideal para instalação e uso em servidores web.

Segundo Niederauer (2005) suas principais características são:

- Gratuito e com código aberto: o arquivo de instalação pode ser obtido gratuitamente no site <http://www.php.net>. Além disso, o PHP é um software com código-fonte aberto.

- Embutido no HTML: o HTML e o PHP podem ser misturados. Pode-se começar a escrever em PHP, de repente escrever um trecho em HTML, depois voltar para o PHP, e assim por diante.
- Baseado no servidor: quando página em PHP é acessada através do navegador, todo o código PHP é executado no servidor, e somente o resultado final é mostrado para o usuário. Fazendo assim, com que o navegador mostre a página já processada, sem consumir recursos de processamento do computador local.
- Bancos de dados: diversos bancos de dados são suportados pelo PHP, ou seja, a linguagem possui códigos que executam funções de vários bancos, entre eles, o MySQL, PostgreSQL, Interbase, Oracle, SQL Server, entre muitos outros disponíveis que oferecem suporte a linguagem SQL.
- Portabilidade: pode-se executar o PHP no Linux, Unix ou Windows.

Um programa em PHP pode ser escrito em qualquer editor de textos, como o Bloco de Notas (do Windows) ou Vi (Linux). Neste trabalho será utilizado o software DreamweaverMX para o desenvolvimento em PHP.

2.9.4 JavaScript

O JavaScript é uma linguagem interpretada pelo navegador criada para incorporar códigos executáveis diretamente em um documento HTML. Conforme Cummins (2002) é utilizado um código no nível do documento para operações de inicialização e para definir funções chamadas de outras partes do documento. Estas funções podem chamar sub-rotinas para dar suporte a um código mais estruturado.

Existem vários recursos importantes oferecidos pelo JavaScript, Cummins (2002) cita alguns:

- Caixa de diálogo: Servem principalmente para validar os valores inseridos em formulários, porém também tem utilidade para alertar o usuário de algo;
- Validação de entradas de dados;

- Cálculos: Pode-se executar vários cálculos matemáticos com o JavaScript para as mais diversas finalidades;
- Eventos de ação: Ações básicas são orientadas por eventos especificados em elementos individuais, como por exemplo, *onmouseover* (Ao passar o mouse sobre a superfície x, execute a função y), *onkeydown* (Ao pressionar tecla x, execute função y) entre vários outros eventos disponíveis;
- Alterações no conteúdo dinâmico: Pode-se alterar dinamicamente o conteúdo de determinado documento;

Acesso as características do navegador: Descobrir qual navegador o usuário esta utilizando para personalizar a exibição da página, ou definir restrições a determinadas funções não aceitas por determinado navegador (Ex: o Firefox possui regras de funções diferentes do Internet Explorer);

- Atualização de propriedades do navegador: Modificar atributos do navegador, como cores das barras de rolagem.
- Cookies: Cookies são registros armazenados pelo navegador que contêm dados associados a interação com determinada página ou determinado servidor web. Com o JavaScript é possível acessar esses dados para delegar mais funcionalidades.

2.9.5 MySQL

O MySQL é um sistema de gerenciamento de banco de dados multiencadeado (SGBD), de código fonte livre e de nível corporativo. O MySQL foi desenvolvido pois houve a necessidade de um banco de dados que fosse extremamente rápido e flexível, atributos que são consideráveis e contam muito neste SGDB. O MySQL é vagamente baseado em outro SGDB, chamado MSQL. (MASLAKOWSKI, 2002). Este se utiliza da linguagem SQL para realizar tarefas dentro do banco de dados, como mostrar, criar e manipular dados.

O MySQL é um SGDB de muito sucesso utilizado em todo mundo, pois possui uma fácil integração com o PHP, tornando o uso muito interessante. Atualmente milhares de serviços de hospedagem de sites possuem o MySQL como serviço oferecido.

É possível citar seis principais características, que são (1) Portabilidade, suportando praticamente qualquer plataforma atual; (2) Compatibilidade, vários módulos de interface para diversas linguagens de programação; (3) Desempenho e Estabilidade; (4) Não necessita de altos recursos de hardware; (5) De fácil utilização e manutenção e (6) é um software livre, com código-fonte aberto para comunidades efetuarem alterações.

Conforme Niederauer (2005) o MySQL tem como principais vantagens:

- Número ilimitado de utilização por usuários simultâneos;
- Capacidade de manipulação de tabelas com mais de 50.000.000 de registros;
- Alta velocidade de execução de comandos;
- Fácil e eficiente controle de privilégios de usuários.

O MySQL possui várias ferramentas para auxílio do desenho banco de dados como o MySQLAdmin, PHP MyAdmin entre outros. Neste projeto será utilizado o PHP MyAdmin por já ser de conhecimento do autor.

2.10 Intranet

Intranet é uma rede de computadores particular, dentro de uma organização que tem como princípio básico, disseminar informações de forma rápida dentro da empresa.

A Intranet utiliza-se dos padrões de comunicação de dados comuns, os mesmo da Internet que são o TCP/IP, que endereça e envia dados de um computador a outro de modo confiável; o FTP, que transmite um arquivo de um computador a outro; o SMTP, que é a base de correio eletrônico, entre outros.

Para Benett (1997, p.4) “uma intranet é um meio privativo que permite a troca de informações e oferece vantagens inigualáveis em termos de custo e recursos através da integração de serviços de redes tradicionais”.

As decisões dentro de uma organização se convertem diretamente no sucesso ou fracasso do material. Com a tecnologia web interna, o acesso às informações se torna mais eficaz. De acordo com Benett (1997) é possível citar três vantagens ao se utilizar desta tecnologia:

- É uma plataforma universal, pois as webs fornecem uma plataforma comum para localizar, recuperar, exibir e atualizar uma variedade de informações;
- É um modo de exibição unificado, pois assim é possível organizar as informações através da apresentação de diversos tipos de dados em estilo padrão, fazendo assim com que se reduza a curva de aprendizagem dos novos aplicativos;
- É uma língua franca, pois esta tecnologia de baseia em padrões flexíveis e universalmente aceitos.

2.11 Internet

Para Laudon e Laudon (2001) a Internet é uma rede mundial de computadores que se originou como uma rede do Departamento de Defesa dos Estados Unidos que pudesse resistir a um ataque estrangeiro, interligando cientistas e pesquisadores. Todo o computador tem igual capacidade para se comunicar como outros computadores na Internet. A Internet não tem gerenciamento central.

A internet é formada por um conjunto de redes de computadores que são interligadas entre si, sendo distribuídas pelo mundo inteiro. Cada vez mais procurada e utilizada por pessoas e organizações, a Internet é procurada para atender propósitos diferentes, para se fazer compras, consultas ou apenas para diversão.

A internet traz muitos benefícios às empresas, podendo interligar sistemas através de acessos *on-line* aos seus bancos de dados, obtendo informações com maior rapidez e precisão, e mesmo, possibilitando a consulta por vendedores aos dados de clientes e pedidos.

Grande parte de sua tecnologia é nova. A segurança é fundamental porque a internet é potencialmente aberta a espões, conhecidos como *crackers* ou *hackers*, que podem causar danos a dados.

Todos os serviços disponíveis na Internet são padronizados, utilizando os mesmo conjunto de protocolos, TCP/IP, o que permite aos usuários a ela conectados, independentemente de sua localização, usufruir dos serviços de informação e comunicação com a certeza de que eles são compatíveis.

2.12 Fornecendo acesso para usuário baseado na Web

A web esta se tornando um meio universal de os usuários acessarem sistemas de negócio. O acesso baseado na web permite que os usuários utilizem o sistema de qualquer parte do mundo, basta executar um navegador de internet comum.

Para Cummins (2002) o acesso para usuários baseado na web proporciona quatro vantagens que devem ser consideradas:

- Os usuários podem ter diversas plataformas de computação;
- Os usuários não precisam ser membros de uma comunidade definida;
- Os usuários podem acessar sistemas de qualquer parte do mundo;
- Os custos de acesso dos usuários são reduzidos.

Por todos estes motivos, é de se esperar que em um futuro breve, todos os aplicativos forneçam um acesso Web.

Normalmente o acesso aos dados da empresa estava contido apenas em uma instalação dentro do escritório, mais com esta nova geração de sistemas é possível consultar, ou inserir determinada informação de qualquer parte, tornando assim o acesso às informações mais fáceis e rápidas.

3 DESCRIÇÃO DA PRÁTICA

Nesta etapa será descrito as atividades práticas realizadas durante o período de estágio na Wiest S/A, tais como análises de elaboração e desenvolvimentos do Helpdesk.

Esta primeira versão foca somente o Departamento de Informática, visando futuramente atender todos os setores da empresa.

3.1 Local do Desenvolvimento do Helpdesk Wiest S/A

O Helpdesk está sendo desenvolvido na empresa Wiest S/A, um dos mais expressivos grupos empresariais brasileiros do setor metal-mecânico. Com unidades fabris instaladas em alguns dos principais pólos industriais do país, produz e fornece excelência em qualidade, estendendo sua atuação por todo o Brasil e também no mercado internacional.

Organização dinâmica, em permanente expansão vem ampliando continuamente a participação em diversos segmentos, consolidando sua posição de vanguarda e liderança.

Como a maioria das empresas, a Wiest também iniciou suas atividades de forma modesta, em 1947, produzindo sistemas de exaustão para veículos em uma pequena fábrica na cidade de Jaraguá do Sul, região norte de Santa Catarina.

A iniciativa se revelaria de grande visão empresarial. Nos anos seguintes, a empresa experimentou um formidável crescimento, impulsionado pelo acelerado desenvolvimento da indústria automobilística brasileira.

A Wiest Escapamentos se desenvolveu, incorporou novas tecnologias, assumiu a liderança no mercado e expandiu fronteiras.

Este é o retrato da Wiest S/A, um grupo de empresas genuinamente brasileiro, construído com trabalho, visão estratégica e movido pela constante busca de novos caminhos para o futuro.

Na Figura 7 pode-se visualizar o site institucional da organização.

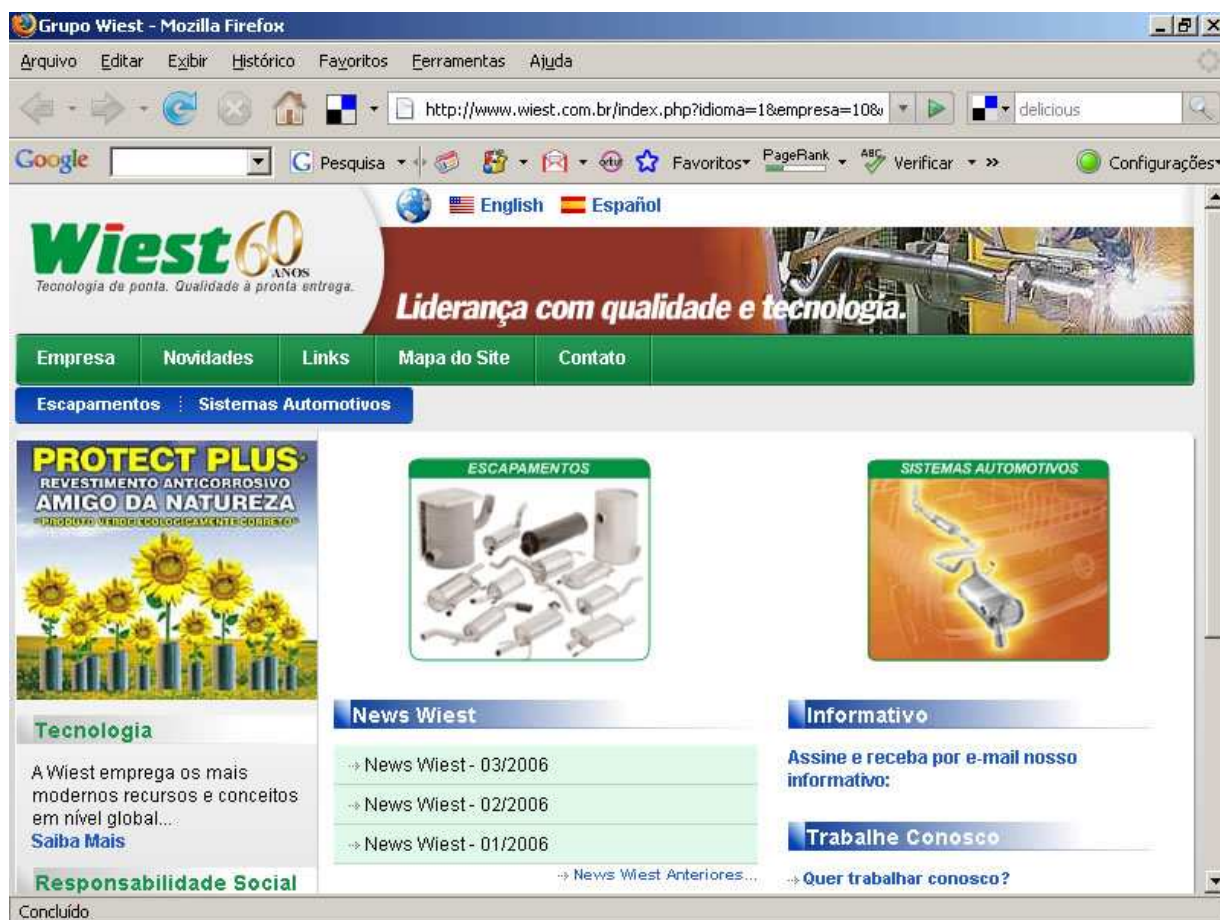


Figura 7 – Site Institucional da empresa Wiest S/A.
Fonte – Ricardo Mondini (2007)

3.2 O Helpdesk

HelpDesk é um termo em inglês que designa o serviço de apoio à usuários para suporte e resolução de problemas técnicos em informática, telefonia e tecnologias de informação. Este apoio pode ser tanto dentro de uma empresa, quanto externamente. Wikipedia (2007).

O HelpDesk é uma ferramenta de fundamental importância para empresas e indústrias de qualquer tamanho e o número de ações necessárias será determinado pela complexidade e dimensão do ambiente envolvido.

Uma definição bem clara do serviço de HelpDesk é descrevê-lo como a manutenção e o suporte técnico-operacional em um ambiente corporativo, envolvendo uma série de atividades distintas mas, ao mesmo tempo, fortemente interdependentes.

Hoje em dia é necessário, cada vez mais, manter o controle de tarefas realizadas e serviços prestados, visando reconhecimento dos colaboradores e satisfação dos clientes ou funcionários. Conhecendo este cenário, as empresas estão adquirindo sempre novos softwares de controle para todas as suas tarefas.

Sabendo dos problemas que ocorrem com o Departamento de Informática da Wiest S/A, tais como o alto número de ligações para o setor; a falta de documentação referente a erros ocorridos; entre outros, é que o Helpdesk foi idealizado.

Este sistema atuará com aberturas de chamado, ou seja, qualquer problema relacionado a informática, o usuário irá abrir um chamado através do sistema para que o técnico de informática ou analista responsável possa atendê-lo e resolver seu problema. Para que a abertura de chamado seja feita, o usuário irá através da Intranet da empresa, acessar o sistema Helpdesk com um login e senha disponibilizado pelo setor, preenchendo um formulário com as seguintes informações:

- Nome;
- Setor;
- Ramal;
- Tipo de problema;
- Descrição do Problema;

Observações. (referentes ao problema).

Quando o chamado for aberto, os funcionários da informática serão alertados pelo software, e o mais habilitado a resolver o problema, assume o chamado. Por exemplo: Se o problema for referente ao Sistema Logix utilizado pela empresa, o Analista de Sistemas irá assumi-lo, caso o problema seja o mau funcionamento de algum hardware da empresa, o técnico de informática irá assumi-lo. Após o responsável assumir ao chamado, poderá ser definido quatro tipos de status ao chamado:

- Em aberto: Quando o chamado ainda esta na lista de espera para ser atendido;
- Em diligência: Quando o chamado esta esperando para ser atendido ou por algo, por exemplo, peças;
- Cancelado: Quando o usuário ou o responsável técnico cancela o chamado;
- Concluído: Quando o chamado foi atendido e resolvido o problema.

Poderão ser feitos a qualquer hora, a criação de relatórios para obter importantes informações sobre problemas ocorridos com equipamentos ou sistema, por setor, por usuário, por data, e por qualquer forma de pesquisa necessária. Sendo assim, torna-se possível tomar decisões como, compra de novos equipamentos ou treinamento para determinada área da empresa.

3.3 Levantamento de requisitos

Para que um sistema possa ser bem desenvolvido e faça o que foi proposto como idéia inicial, um levantamento de requisitos correto e bem elaborado é essencial.

Este sistema atuará em conjunto com os usuários da empresa que utilizam equipamentos de informática. O sistema apresentará como requisito funcional o controle de uso, qualquer pessoa que acessar o sistema terá que informar login e senha de acesso. Com estes dados será possível saber qual o nível de acesso do usuário que esta acessando sistema no momento. Caso o usuário seja administrador, será possível ver os chamados abertos, criar relatórios, atender o chamado e adicionar perguntas frequentes. Caso for usuário comum, somente abrem chamados e tem acesso a área de perguntas frequentes. O sistema irá ser executar em PHP com banco de dados MySQL que será colocado no servidor da intranet da empresa. Assim, qualquer usuário com acesso a intranet, que abrange todos os funcionários que utilizam equipamentos de informática na empresa, terá acesso ao Sistema Helpdesk.

Através da técnica de levantamento de requisitos é possível entender e documentar todas as técnicas e funcionalidades que existirão no sistema.

3.3.1 Elaborar modelo de negócio

Com o levantamento de requisitos, foi possível discriminar as classes que se relacionam entre si para representar a estrutura do sistema. Este sistema foi elaborado num modelo de negócio com base numa primeira pesquisa que trouxe algumas informações para a elaboração do projeto. Na segunda análise foi possível constatar o funcionamento da rotina diária do analista de suporte e do analista de sistemas, descobrindo que as principais classes estão relacionadas como, Usuário, Setor, Problema e Relatórios.

Com estas identificações foi possível construir e caracterizar os relacionamentos existentes entre as classes, sendo criado um diagrama de classe do Sistema Helpdesk.

Deste modo, foi possível encontrar seis classes no sistema, sendo que uma é apenas de consulta, não tendo relação com o restante das classes. O diagrama encontra-se na Figura 8.

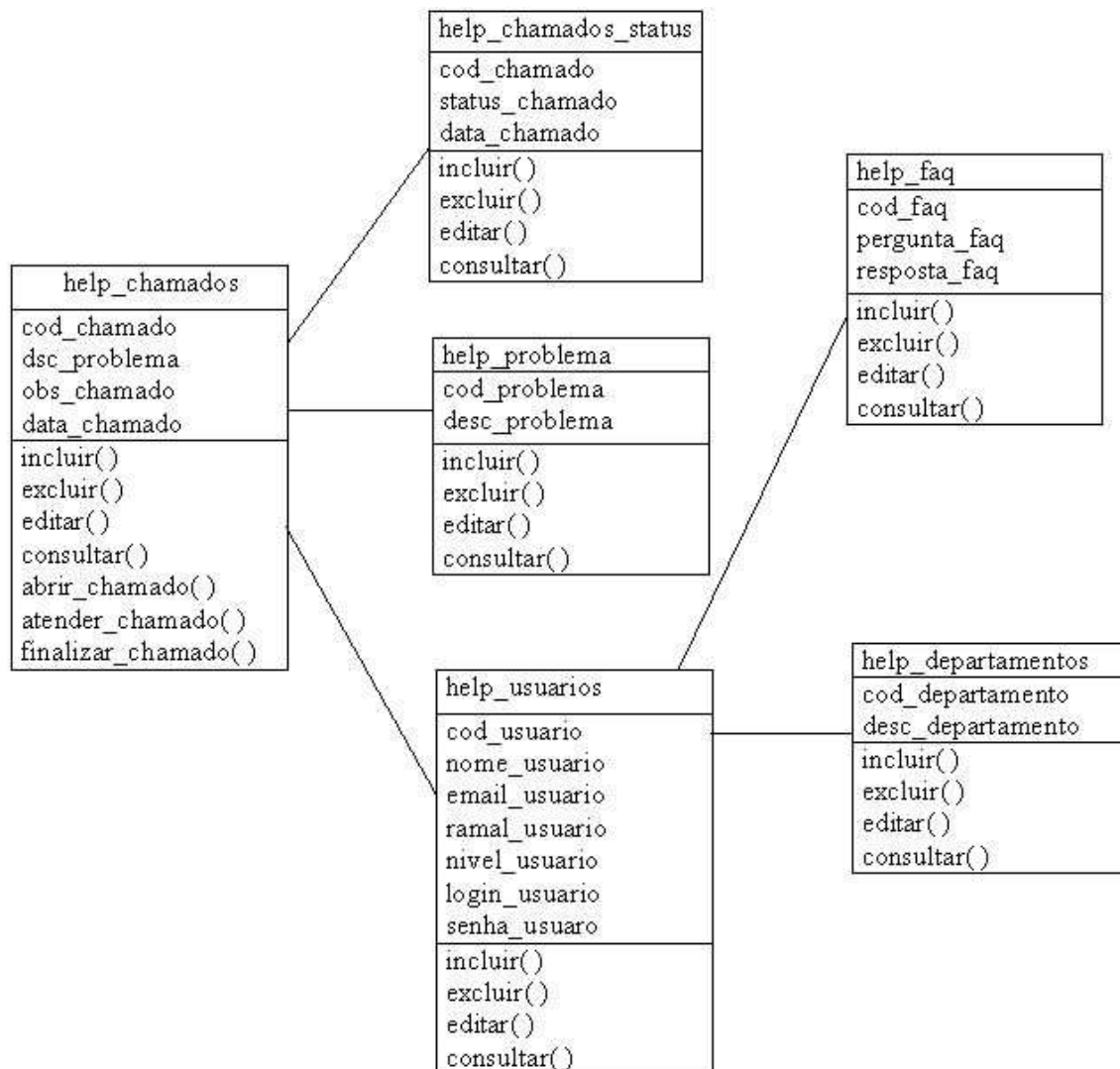


Figura 8 – Diagrama de Classes
Fonte – Ricardo Mondini (2007)

O diagrama acima está num nível de detalhamento avançado, fazendo com que seja possível visualizar os objetos que estão dentro da classe que lhe convém, e estas estão relacionadas entre si.

3.3.2 Encontrar Atores e Casos de Uso

As atividades do sistema dependerão de totalmente dos atores e casos de uso tornando esta etapa a mais importante do levantamento de requisitos.

Os atores são os agentes que interagem com o sistema, podendo ser as pessoas, usuários e outros sistemas externos. A forma clássica citada por muitos autores, é se perguntar: Quem utilizará o sistema? Existirá algum elemento externo que colaborará com o sistema? Já os casos de uso são seqüências de ações do sistema desempenhadas por um ator na sua interação com o sistema, para facilitar, a pergunta para ser feita é: O que o sistema fará?

Após os conceitos estarem completamente claros, foi possível coletar os dados necessários com os nomes dos atores e dos casos de uso e elaborar um relacionamento entre eles. Com estes relacionamentos é que consegue-se apreciar a especificação das interfaces dos usuários. Na Tabela 1, pode-se conferir todos os atores do sistema Helpdesk e algumas informações necessárias, e na Tabela 2 os casos de uso relacionados com seus atores e com suas descrições.

3.3.2.1 Encontrar Atores

Tabela 1 – Atores do Sistema Helpdesk

Ator	Definição	Permissão de acesso	Frequência de uso
Gerente de informática	Relatórios	Total	Sempre que houver necessidade de obtenção de informações.
Analistas de sistemas/Analista de Suporte	Relatórios, Atendimentos de chamados, inclusão de perguntas frequentes.	Total	Diariamente
Usuários finais	Cadastro de chamados, pesquisa de Perguntas Frequentes.	Aplicação	Diariamente

Fonte – Ricardo Mondini (2007)

Na Tabela 1 se pode constatar o detalhamento dos atores do sistema Helpdesk, mostrando seus nomes, permissões no sistema e frequências de utilização.

3.3.2.2 Encontrar Casos de Uso

Os casos de uso encontrados no sistema Helpdesk foram:

- Emissão de relatórios;
- Gestão de usuários;
- Gestão de departamentos;
- Gestão de problemas;
- Gestão de perguntas frequentes;
- Abertura de chamados;
- Assumir chamado;
- Atendimento de chamado;
- Chamado em diligência;
- Finalizar chamado;
- Cancelar chamado.

Após a identificação dos casos de uso e atores do sistema Helpdesk, será apresentado, como dito anteriormente, na Tabela 2 o fluxo que cada ator terá nos casos de uso.

Tabela 2 – Relacionamento de Casos de Uso X Atores

Nome do Caso de Uso	Atores	Descrição
Emissão de relatórios	Gerente de informática/Analista de Sistemas / Analista de Suporte	Emissão de relatórios com informações sobre setores com maiores problema, usuários com maiores problemas, problemas frequentes, filtro de usuário, setor, status e total geral de chamados do período, atendimentos dos analistas também pelos mesmos filtros.
Gestão de usuários	Analista de Sistemas / Analista de Suporte	Processamento de inclusão, alteração, exclusão e consulta de dados do usuário.
Gestão de departamentos	Analista de Sistemas / Analista de Suporte	Processamento de inclusão, alteração, exclusão e consulta de departamentos da empresa.
Gestão de problemas	Analista de Sistemas / Analista de Suporte	Processamento de inclusão, alteração, exclusão e consulta de problemas.
Gestão de Perguntas Frequentes.	Analista de Sistemas / Analista de Suporte / Usuário	Processamento de inclusão, alteração, exclusão e consulta de problemas frequentes, para tipo de usuário administrador e consulta para tipo de usuário normal.
Abertura de chamados	Analista de Sistemas / Analista de Suporte / Usuário	Operação de cadastro de chamado para solicitação de um analista/técnico para a resolução do mesmo.
Assumir chamado	Analista de Sistemas / Analista de Suporte	Momento onde o profissional que melhor se encaixa para resolver o chamado, assumi o chamado para si para então poder resolve-lo.
Atendimento de chamado	Analista de Sistemas / Analista de Suporte	Operação onde o analista/técnico entra no sistema e altera o status para “Em diligência” e se dirige ao local do problema.

Chamado em diligência	Analista de Sistemas / Analista de Suporte	Operação onde o analista/técnico entra no sistema e altera o status para “Em diligência” e informa o motivo.
Finalizar chamado	Analista de Sistemas / Analista de Suporte	Operação onde o analista/técnico entra no sistema e finaliza o chamado informando o problema e a solução.
Cancelar chamado	Analista de Sistemas / Analista de Suporte / Usuário	Operação onde o administrador pode cancelar qualquer chamado informando o motivo do cancelamento e o usuário normal pode cancelar apenas os seus chamados, informando o motivo do cancelamento.

Fonte – Ricardo Mondini (2007)

Na Tabela 2, constam às ações que o sistema irá proporcionar para os colaboradores da empresa envolvidos com o sistema. Os casos de uso estão nomeados, mostrando qual ator irá interagir com ele e sua definição.

3.3.3 Priorizar Casos de Uso

Nesta fase, é necessário destacar, os casos de uso mais críticos para Sistema Helpdesk, que serão a base para o funcionamento. Após a realização de análises, foi possível identificar os casos de uso mais críticos do sistema no Sistema Helpdesk, o resultado da seleção aparece na Figura 9:

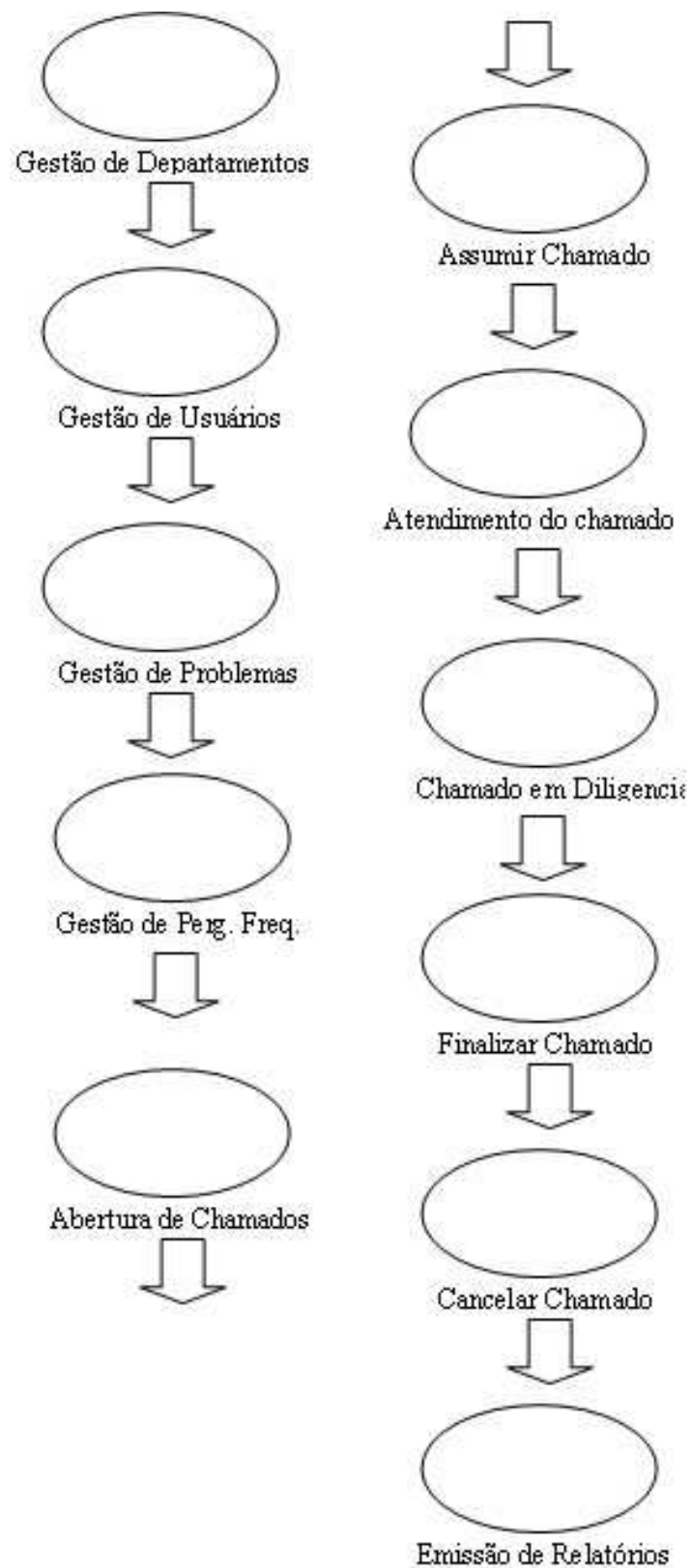


Figura 9 – Priorização dos Casos de Uso do Sistema HelpDesk
Fonte – Ricardo Mondini (2007)

3.3.4 Detalhar Casos de Uso

O detalhamento dos casos de uso é uma documentação do Sistema Helpdesk. Esta documentação deve ser suficiente para a compreensão do seu funcionamento. Para que isso possa ocorrer serão montadas as Tabelas 3 á 13 com os detalhamentos.

Tabela 3 – Detalhamento do Caso de Uso: Gestão de Departamentos

Caso de Uso	Gestão de Departamentos
Objetivo	Armazenar informações dos departamentos no sistema
Atores	Analista de Sistemas / Analista de Suporte
Pré-Condições	Existir setor para cadastrar
Pós-Condições	O sistema armazena o cadastro de setores
Ativação	Chamar a tela de cadastro de setores
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso a esta tela; 2. Acessa o <i>link</i> “Adicionar Departamentos” ou “Editar Departamentos”; 3. Informa o nome do novo departamento; 4. Clica em Incluir.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. Se o administrador não digitar o departamento, o sistema irá alertá-lo e não gravará os dados.

Fonte – Ricardo Mondini (2007)

Tabela 4 – Detalhamento do Caso de Uso: Gestão de Usuários

Caso de Uso	Gestão de Usuários
Objetivo	Armazenar informações dos usuários do sistema
Atores	Analista de Sistemas/ Analista de Suporte
Pré-Condições	Existir usuários para cadastrar
Pós-Condições	O sistema armazena o cadastro de usuários
Ativação	Chamar a tela de cadastro de usuários
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso a esta tela; 2. Acessa o <i>link</i> “Adicionar Usuário” ou “Editar Usuário”; 3. Informa o nome, e-mail, ramal, setor, nível de acesso, <i>login</i> e senha do novo usuário; 4. Clica em Incluir.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. Se o administrador não digitar qualquer um dos campos, o sistema irá alertá-lo e não gravará os dados; 2. Se o administrador inserir dados que já existem no banco, no campo login ou e-mail o sistema irá alertá-lo e não gravará os dados avisando que não é possível incluir dois usuários com o mesmo login ou e-mail.

Fonte – Ricardo Mondini (2007)

Tabela 5 – Detalhamento do Caso de Uso: Gestão de Problemas

Caso de Uso	Gestão de Problemas
Objetivo	Armazenar informações de possíveis problemas no sistema
Atores	Analista de Sistemas / Analista de Suporte
Pré-Condições	Existir problemas para cadastrar
Pós-Condições	O sistema armazena o problema cadastrado
Ativação	Chamar a tela de cadastro de problemas
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso a esta tela; 2. Informa o problema; 3. Clica em Incluir.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. Se o administrador não digitar o problema, o sistema irá alertá-lo e não gravará os dados.

Fonte – Ricardo Mondini (2007)

Tabela 6 – Detalhamento do Caso de Uso: Perguntas Frequentes

Caso de Uso	Gestão de Perguntas Frequentes
Objetivo	Armazenar Perguntas e Respostas frequentes
Atores	Analista de Sistemas / Analista de Suporte
Pré-Condições	Existir perguntas e respostas para cadastrar
Pós-Condições	O sistema armazena a pergunta e resposta cadastrada
Ativação	Chamar a tela de cadastro de Perguntas e Respostas
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso a esta tela; 2. Acessa o <i>link</i> “Adicionar FAQ” ou “Editar FAQ”; 3. Informa pergunta; 4. Informa resposta; 5. Clica em Incluir.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. Se o administrador não digitar o pergunta e/ou resposta, o sistema irá alertá-lo e não gravará os dados; 2. Se o administrador inserir dados que já existem no banco, no campo pergunta e/ou e-mail o sistema irá alertá-lo e não gravará os dados avisando que não é possível incluir duas perguntas ou respostas com o mesmo conteúdo.

Fonte – Ricardo Mondini (2007).

Tabela 7 – Detalhamento do Caso de Uso: Abertura de Chamados

Caso de Uso	Abertura de Chamados
Objetivo	Armazenar informações sobre os chamados ocorridos
Atores	Analista de Sistemas / Analista de Suporte / Usuário
Pré-Condições	É necessário existir, usuário, setor e problema para cadastrar
Pós-Condições	O sistema armazena a abertura de chamado
Ativação	Chamar a tela de abertura de chamados
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema; 2. Acessa o <i>link</i> “Abrir Chamado”; 3. Informa o problema, a descrição do problema e se necessário, uma observação; 4. Clica em Abrir chamado.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. Se o usuário não digitar o problema e a descrição o sistema irá alertá-lo e não gravará os dados.

Fonte – Ricardo Mondini (2007)

Tabela 8 – Detalhamento do Caso de Uso: Assumir Chamados

Caso de Uso	Assumir Chamados
Objetivo	Determinar para qual profissional o chamado será direcionado
Atores	Analista de Sistemas / Analista de Suporte
Pré-Condições	É necessário existir chamados em aberto e que nenhum outro profissional tenha assumido
Pós-Condições	O sistema passa o chamado para o administrador responsável
Ativação	Chamar a tela de chamados abertos
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso a esta tela; 2. Acessa o <i>link</i> “Todos os Chamados”; 3. Na tela, o administrador responsável clica no código do chamado aberto; 4. Dentro da nova janela aberta na tela, ele clica em “Assumir”.
Fluxo Alternativo	

Fonte – Ricardo Mondini (2007)

Tabela 9 – Detalhamento do Caso de Uso: Atendimento de Chamados

Caso de Uso	Atendimento do Chamado
Objetivo	Alterar o status do chamado no sistema para “Em diligência”
Atores	Analista de Sistemas / Analista de Suporte
Pré-Condições	Precisa existir chamado em aberto
Pós-Condições	O sistema altera o status do chamado para “Em diligência”
Ativação	Chamar a tela de atendimento de chamados
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso a esta tela; 2. Acessa o <i>link</i> “Meus Chamados”; 3. Clica no código do chamado assumido anteriormente; 4. Modifica o status do chamado e descreve o motivo como “Em atendimento”; 5. Clica em Alterar.
Fluxo Alternativo	

Fonte – Ricardo Mondini (2007)

Tabela 10 – Detalhamento do Caso de Uso: Chamado em Diligência

Caso de Uso	Chamado em Diligência
Objetivo	Alterar o status do chamado no sistema para “Em diligência”
Atores	Analista de Sistemas / Analista de Suporte
Pré-Condições	Precisa existir chamado em aberto
Pós-Condições	O sistema altera o status do chamado para “Em diligência”
Ativação	Chamar a tela de atendimento de chamados
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso a esta tela; 2. Acessa o <i>link</i> “Meus Chamados”; 3. Clica no código do chamado; 4. Modifica o status do chamado e descreve um motivo para este status; 5. Clica em Alterar.
Fluxo Alternativo	

Fonte – Ricardo Mondini (2007)

Tabela 11 – Detalhamento do Caso de Uso: Finalizar Chamado

Caso de Uso	Finalizar Chamado
Objetivo	Alterar o status do chamado no sistema para “Concluído”
Atores	Analista de Sistemas / Analista de Suporte
Pré-Condições	Precisa existir chamado em aberto
Pós-Condições	O sistema altera o status do chamado para “Concluído”
Ativação	Chamar a tela de atendimento de chamados
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso a esta tela; 2. Acessa o <i>link</i> “Meus Chamados”; 3. Clica no código do chamado; 4. Modifica o status do chamado e descreve a solução para o problema; 5. Clica em Alterar.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. Se o administrador não descrever a solução, o sistema irá alertá-lo e não gravará os dados.

Fonte – Ricardo Mondini (2007)

Tabela 12 – Detalhamento do Caso de Uso: Cancelar Chamado

Caso de Uso	Cancelar Chamado
Objetivo	Alterar o status do chamado no sistema para “Cancelado”
Atores	Analista de Sistemas / Analista de Suporte / Usuário
Pré-Condições	Precisa existir chamado em aberto
Pós-Condições	O sistema altera o status do chamado para “Cancelado”
Ativação	Chamar a tela de atendimento de chamados
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso a todos os chamados, se for usuário, somente aos próprios chamados; 2. Acessa o <i>link</i> “Meus Chamados”; 3. Clica no código do chamado; 4. Modifica o status do chamado e descreve o motivo do cancelamento; 5. Clica em Alterar.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. Se o administrador ou usuário não descrever o motivo do cancelamento, o sistema irá alertá-lo e não gravará os dados.

Fonte – Ricardo Mondini (2007)

Tabela 13 – Detalhamento do Caso de Uso: Emissão de Relatórios

Caso de Uso	Emissão de Relatórios
Objetivo	Emitir relatórios para acompanhamento
Atores	Gerente de informática / Analista de Sistemas / Analista de Suporte
Pré-Condições	Precisam existir chamados no sistema
Pós-Condições	Haver chamados no período decorrente ao relatório escolhido
Ativação	Chamar a tela de relatório necessária
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário acessa o sistema, a partir do seu nível, ele determina se é ou não administrador, se administrador tem acesso; 2. Clica no <i>link</i> “Relatórios”; 3. Seleciona o relatório por: Filtro por datas, e depois ordenado por Usuário; Setor; ou Problema; 4. Clica em Gerar relatório; 5. O relatório é gerado e aparece a opção de impressão.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. Se a data estiver digitada errada o sistema alertará e não fará o relatório; <p>Se não houver chamados no período selecionado, o sistema irá mostrar o relatório em branco.</p>

Fonte – Ricardo Mondini (2007)

3.3.5 Estruturar Modelo de Caso de Uso

Abaixo segue as estruturas do Modelo de Caso de Uso detalhadas anteriormente na seção 3.3.4.

Gestão de departamentos:



Figura 10 – Modelo de Caso de Uso: Gestão de Departamentos

Fonte – Ricardo Mondini (2007)

Gestão de problemas:

Figura 11 – Modelo de Caso de Uso: Gestão de Problemas
Fonte – Ricardo Mondini (2007)

Gestão de usuários:

Figura 12 – Modelo de Caso de Uso: Gestão de Usuários
Fonte – Ricardo Mondini (2007)

Gestão de perguntas freqüentes:

Figura 13 – Modelo de Caso de Uso: Gestão de Perguntas Freqüentes
Fonte – Ricardo Mondini (2007)

Abertura de chamados:

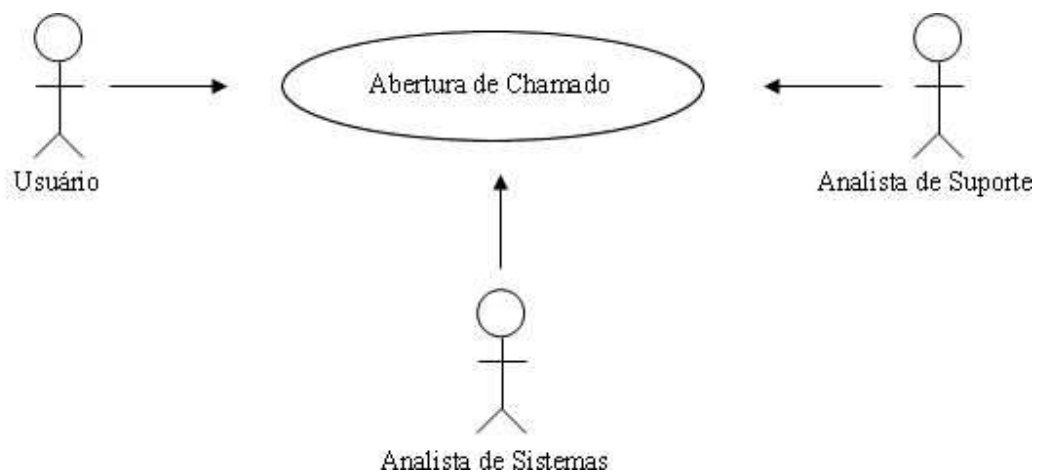


Figura 14 – Modelo de Caso de Uso: Abertura de Chamados
Fonte – Ricardo Mondini (2007)

Assumir chamado:

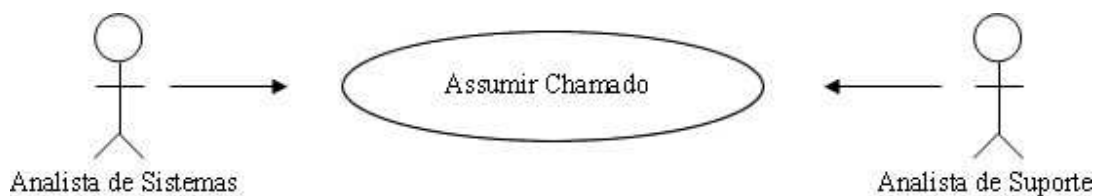


Figura 15 – Modelo de Caso de Uso: Assumir Chamado
Fonte – Ricardo Mondini (2007)

Atendimento do Chamado:



Figura 16 – Modelo de Caso de Uso: Atendimento Chamado
Fonte – Ricardo Mondini (2007)

Chamado em diligência:



Figura 17 – Modelo de Caso de Uso: Chamado em Diligência
Fonte – Ricardo Mondini (2007)

Finalizar chamado:



Figura 18 – Modelo de Caso de Uso: Finalizar Chamado
Fonte – Ricardo Mondini (2007)

Cancelar chamado:

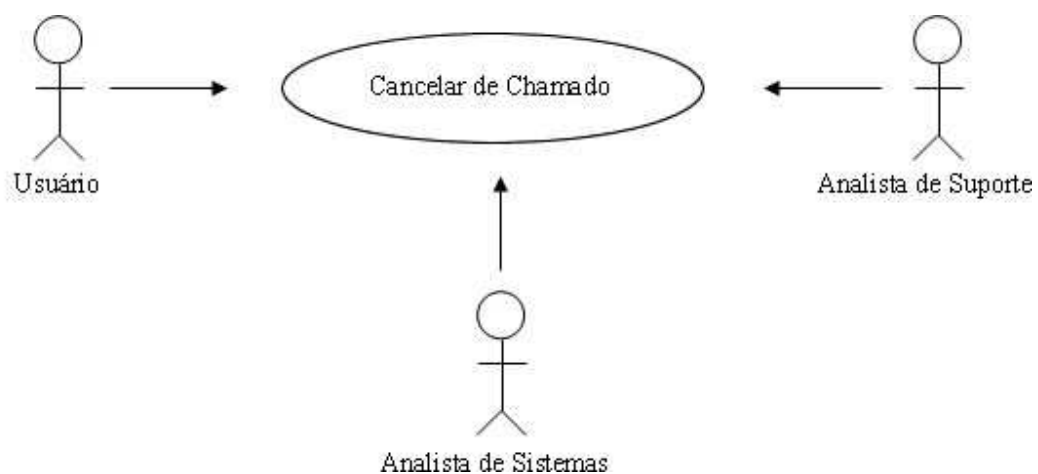


Figura 19 – Modelo de Caso de Uso: Cancelar Chamado
Fonte – Ricardo Mondini (2007)

Emissão de relatórios:

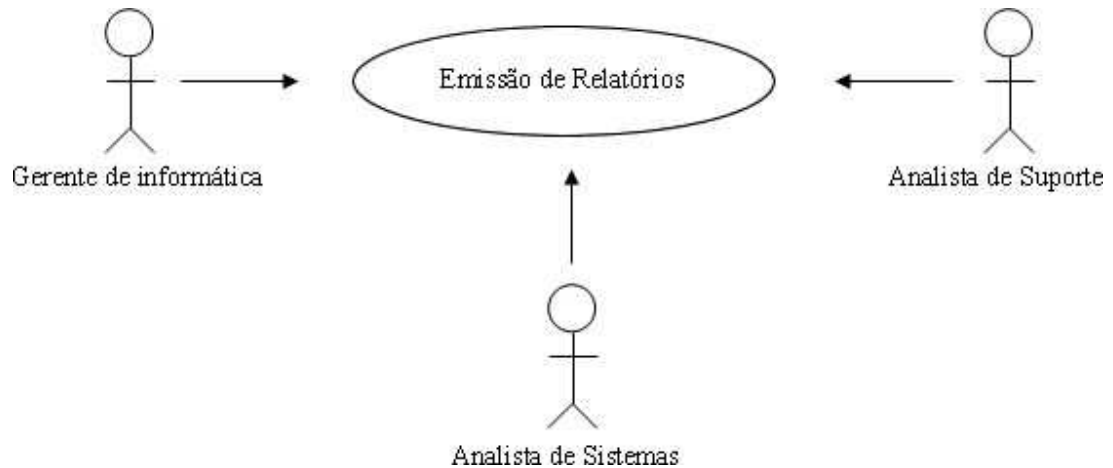


Figura 20 – Modelo de Caso de Uso: Emissão de Relatórios
 Fonte – Ricardo Mondini (2007)

3.3.6 Elaborar Modelo de Caso de Uso de Contexto.

O Modelo de Caso de Uso de Contexto é considerado o mais importante, pois permite uma visão das interfaces do sistema com os diversos tipos de usuários e outros sistema que poderiam ser representados caso fosse necessário.

No modelo de caso de uso de contexto abaixo, os usuários continuam sendo representados como atores e os casos de uso representam as possíveis formas de interação do produto com os atores.

Na Figura 21, é possível visualizar o modelo de caso de uso de contexto do Sistema Helpdesk:

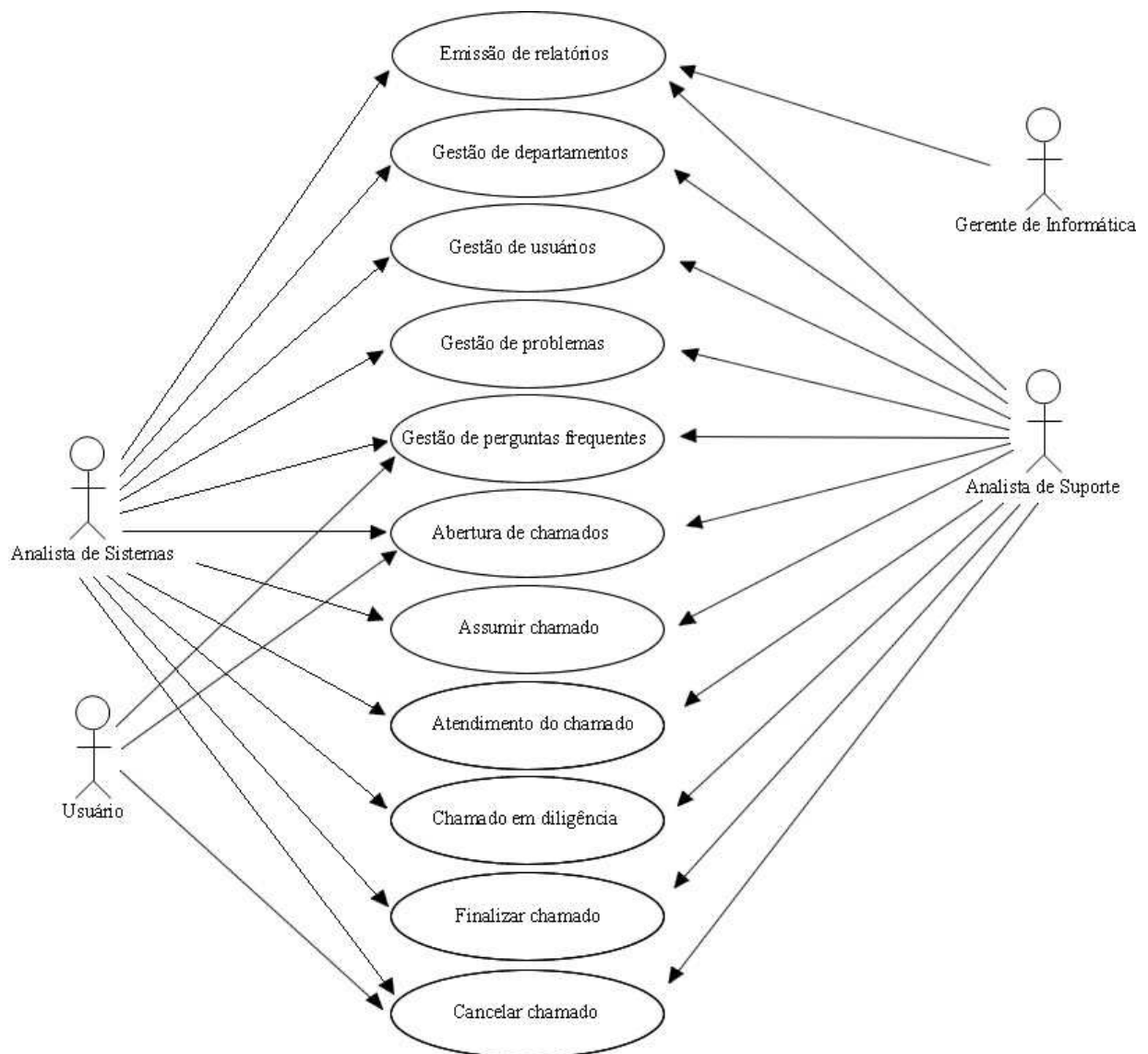


Figura 21 – Modelo de Caso de Uso de Contexto do Sistema HelpDesk
Fonte – Ricardo Mondini (2007)

3.4 Análise do Sistema Helpdesk

Os diagramas de classe mostram as operações e os relacionamentos de dependência que se considera necessário mostrar. Este processo consiste em analisar e detalhar o diagrama de

classe do levantamento de requisitos, se necessário, novas classe no diagrama resultando num novo diagrama de classes com mais relacionamentos.

Na Figura 16 é possível visualizar o novo Diagrama de classes do Sistema Helpdesk, desta vez mostrando os relacionamentos (um para um, muitos para muitos, um para muitos ou ao contrário), colocando também as chaves estrangeiras para que se possa tomar cuidado com ações futuras.

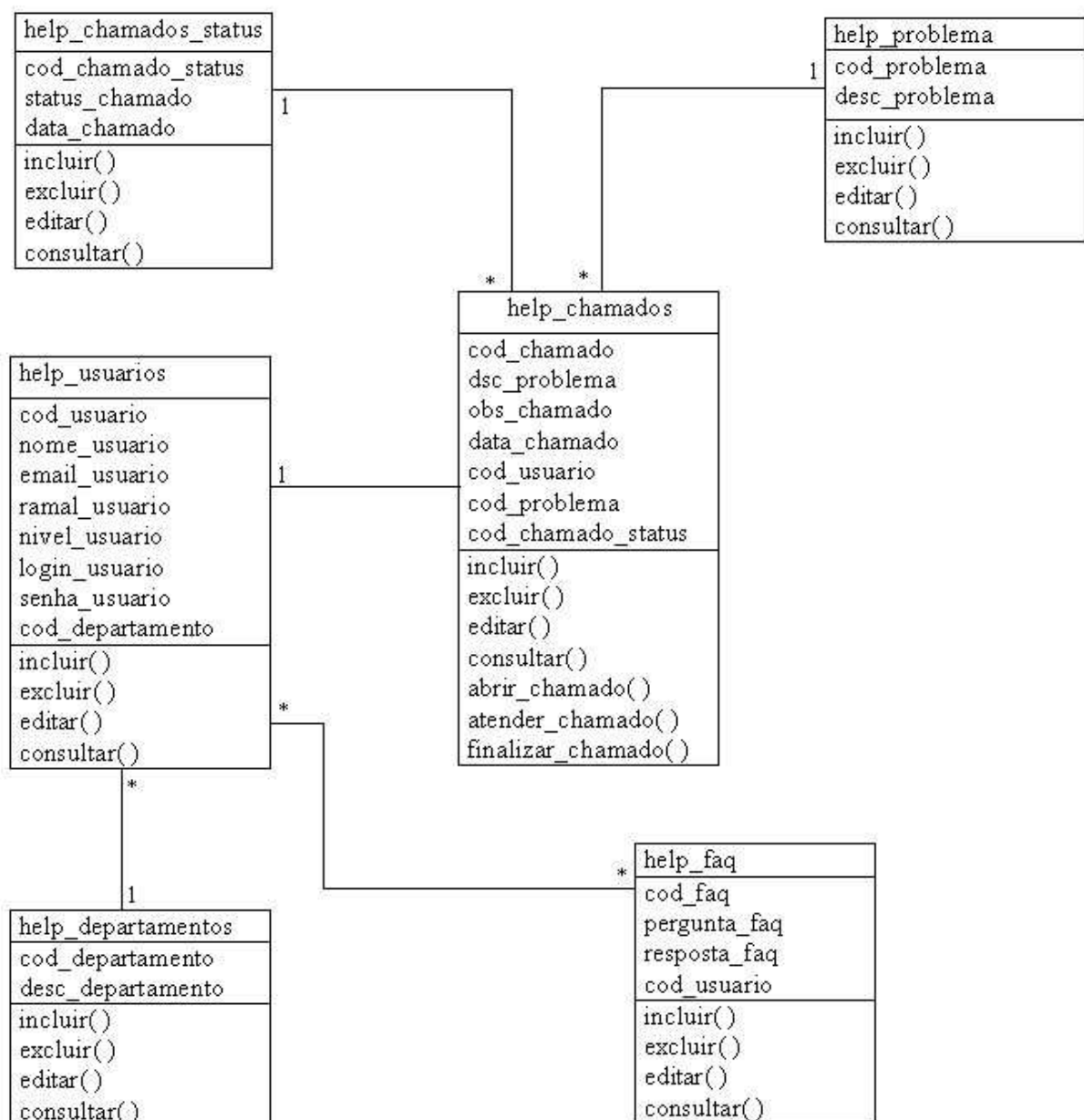


Figura 22 – Diagrama de Classe detalhado do Sistema Helpdesk
Fonte – Ricardo Mondini (2007)

3.5 Design do Sistema Helpdesk

Nesta fase todos os modelos foram especificados detalhadamente para destacar o armazenamento dos dados e a interface do sistema com o usuário, considerando a tecnologia que será utilizada.

3.5.1 Projetar Banco de Dados

Esta fase é muito importante para o bom funcionamento do sistema, pois o banco de dados deve estar normalizado para que não ocorra nenhum erro no sistema.

A Figura 23 mostra que o banco de dados do Sistema Helpdesk está na terceira forma normal, pois não há nenhum atributo que possui dependência transitiva em relação a outro atributo da entidade que não participe da chave primária, além de não conter tabelas que propiciem a redundância de dados, tornando assim o banco legível evitando a ocupação de espaço desnecessário e deixando-o mais rápido.

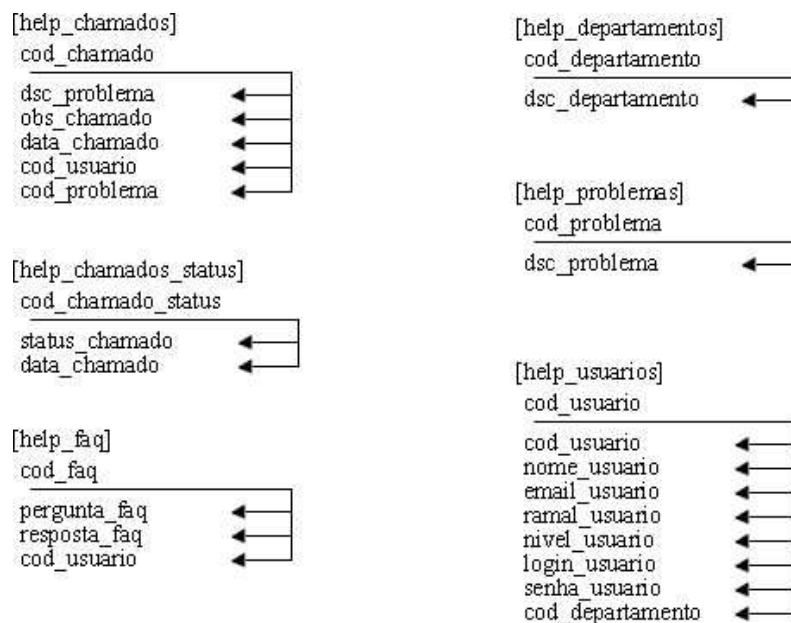


Figura 23 – Normalização do Banco de Dados Sistema Helpdesk
Fonte – Ricardo Mondini (2007)

3.5.2 Matriz de Associação

Nesta etapa, a Tabela 14 demonstra a matriz de associação onde é possível visualizar facilmente o relacionamento entre as tabelas do banco de dados.

Tabela 14 – Matriz de Associação do Sistema Helpdesk

	Tabelas	1	2	3	4	5	6
1	help_chamados			N		N	N
2	help_departamentos						1
3	help_chamados_status	1					
4	help_faq						N
5	help_problemas	1					
6	help_usuario	1	N		N		

Fonte – Ricardo Mondini (2007)

3.5.3 Dicionário de Dados

Nesta fase serão apresentadas todas as tabelas do banco de dados do Sistema Helpdesk com os atributos, formatos, descrição e observações.

Tabela 15 – Dicionário de Dados: HELP_CHAMADOS

Tabela: HELP_CHAMADOS			
Descrição: Armazena dados referentes aos chamados efetuados pelo usuário.			
Atributo	Descrição	Formato	Observação
COD_CHAMADO	Código do chamado	9999	Chave primária
DSC_PROBLEMA	Descrição do problema	X(150)	Obrigatório
OBS_CHAMADO	Observações	X(150)	Livre
DATA_CHAMADO	Data de inserção do chamado no sistema	dd/mm/aaaa hh:mm:ss	Obrigatório
COD_USUARIO	Código do usuário	9999	Chave estrangeira
COD_PROBLEMA	Código do Problema	9999	Chave estrangeira

Fonte – Ricardo Mondini (2007)

Tabela 16 – Dicionário de Dados: HELP_CHAMADOS_STATUS

Tabela: HELP_CHAMADOS_STATUS			
Descrição: Armazena dados referentes aos status do sistema.			
Atributo	Descrição	Formato	Observação
COD_CHAMADO_STATUS	Código do status	9999	Chave primária
STATUS_CHAMADO	Tipos de status	X(150)	Obrigatório
DATA_CHAMADO	DATA?	dd/mm/aaaa hh:mm:ss	Obrigatório

Fonte – Ricardo Mondini (2007)

Tabela 17 – Dicionário de Dados: HELP_DEPARTAMENTOS

Tabela: HELP_DEPARTAMENTOS			
Descrição: Armazena dados referentes aos departamentos da empresa.			
Atributo	Descrição	Formato	Observação
COD_DEPARTAMENTO	Código do depto	9999	Chave primária
DSC_DEPARTAMENTO	Nome do depto	X(150)	Obrigatório

Fonte – Ricardo Mondini (2007)

Tabela 18 – Dicionário de Dados: HELP_PROBLEMAS

Tabela: HELP_PROBLEMAS			
Descrição: Armazena dados referentes aos problemas ocorridos.			
Atributo	Descrição	Formato	Observação
COD_PROBLEMA	Código do problema	9999	Chave primária
DSC_PROBLEMA	Descrição do problema	X(150)	Obrigatório

Fonte – Ricardo Mondini (2007)

Tabela 19 – Dicionário de Dados: HELP_FAQ

Tabela: HELP_FAQ			
Descrição: Armazena dados referentes perguntas frequentes do sistema.			
Atributo	Descrição	Formato	Observação
COD_FAQ	Código do faq	9999	Chave primária
PERGUNTA_FAQ	Pergunta	X(150)	Obrigatório
RESPOSTA_FAQ	Resposta	X(150)	Obrigatório
COD_USUARIO	Código do usuário	9999	Chave estrangeira

Fonte – Ricardo Mondini (2007)

Tabela 20 – Dicionário de Dados: HELP_USUÁRIO

Tabela: HELP_USUARIOS			
Descrição: Armazena dados referentes aos usuários do sistema.			
Atributo	Descrição	Formato	Observação
COD_USUARIO	Código do usuário	9999	Chave primária
NOME_USUARIO	Nome do usuário	X(150)	Obrigatório
EMAIL_USUARIO	E-mail do usuário	X(150)	Obrigatório
RAMAL_USUARIO	Ramal do usuário	X(150)	Obrigatório
NIVEL_USUARIO	Nível de acesso, adm ou normal	X(1)	Obrigatório
LOGIN_USUARIO	Login do usuário	X(20)	Obrigatório
SENHA_USUARIO	Senha do usuário	X(32)	Obrigatório
COD_DEPARTAMENTO	Código do departamento	9999	Chave estrangeira

Fonte – Ricardo Mondini (2007)

3.5.4 Projetar Interface

O desenho do projeto necessita de muita atenção, pois será com ele que o usuário irá interagir para utilizar o sistema. Um projeto de interface bem elaborado pode definir se o sistema será utilizado ou não.

Na empresa Wiest S/A, o Sistema Helpdesk será colocado para acesso aos usuários, em forma de um *link* na intranet da empresa, desenvolvida pelo autor. Pode-se visualizar a atual Intranet da empresa na Figura 24.



Figura 24 – Tela da Intranet da empresa
Fonte – Ricardo Mondini (2007)

Todas as telas foram desenvolvidas utilizando-se do PHP e HTML como linguagem de programação e MySQL como banco de dados.

Neste tópico serão observadas todas as telas do sistema e toda a área de acesso dos administradores e dos usuários comuns do sistema. A partir do momento em que o usuário acessa o link **http://intranet.wiest.lan/helpdesk** ou acessa o link na intranet no seu navegador (*Mozilla Firefox* ou *IE6.0+*), será solicitado um *login* e senha, conforme pode ser visualizado abaixo.

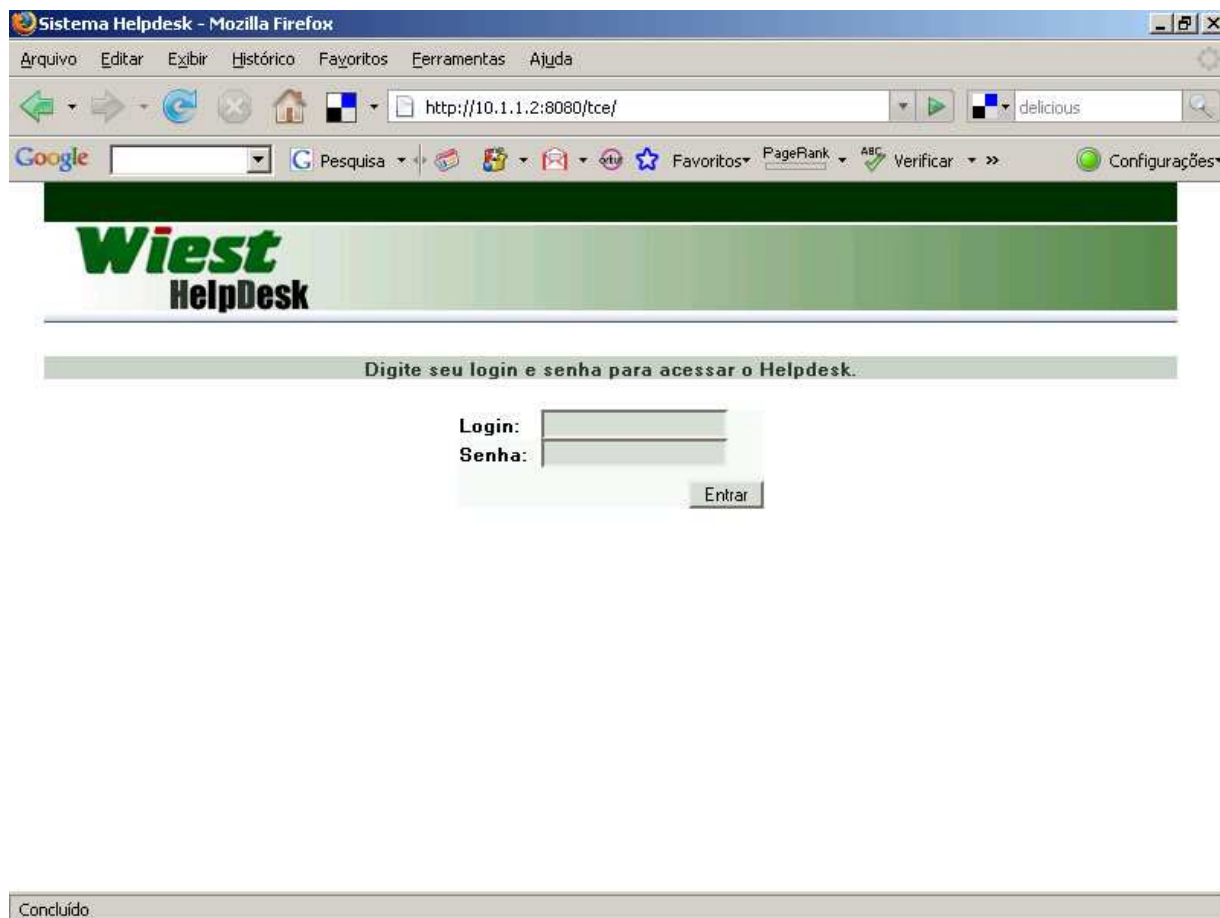
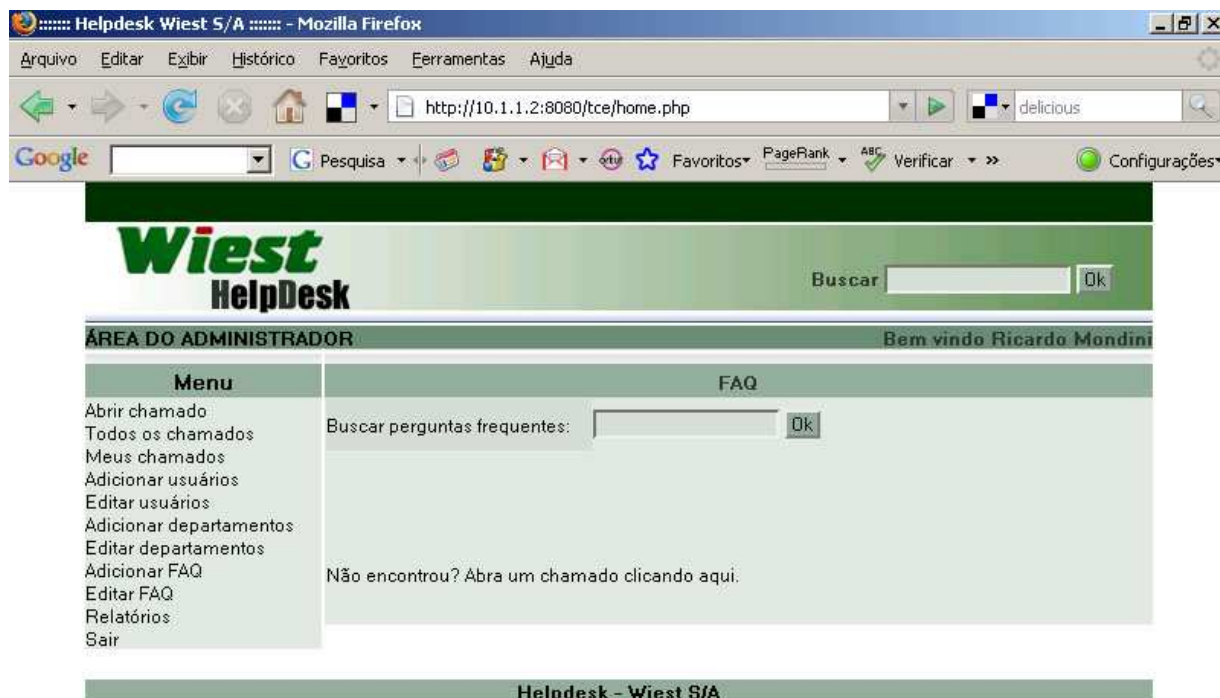


Figura 25 – Tela de Login do Sistema Helpdesk
Fonte – Ricardo Mondini (2007)

Na A partir do momento que o usuário acessa o sistema, automaticamente é verificado se ele possui acesso Administrador, ou acesso Usuário. No tópico 3.5.4.1 será mostrado as telas referentes ao acesso administrador, e no tópico 3.5.4.2 poderá ser visualizado as telas do usuário.

3.5.4.1 Área do administrador

Nesta área, o analista de suporte e o analista de sistema, que é pré-cadastrado com a condição administrador, podem efetuar toda a gerencia dos chamados que existem no sistema. Abaixo seguem as telas.



Concluído.

Figura 26 – Tela inicial e de FAQ - Administrador
Fonte – Ricardo Mondini (2007)

Na Figura 26 é mostrado a página inicial do Helpdesk, nela está disposto o menu de acesso a todo o sistema. É também a área de FAQ para o administrador pesquisar possíveis problemas que já ocorreram e também para o caso de novos analistas entrarem na empresa, possa ser efetuada uma busca pelo banco para realizar o suporte da melhor e mais rápida forma possível.

Wiest HelpDesk

ÁREA DO ADMINISTRADOR Bem vindo Ricardo Mondini

Menu	Abrir chamado		
Abrir chamado	Operador:	Ricardo Mondini	Ramal: 5152
Todos os chamados	Data de abertura:	05/11/2007 - 16:39:47	Setor: Informática
Meus chamados	Tipo do problema:	Selecione área...	Status: Em aberto
Adicionar usuários	Descrição do problema:		
Editar usuários	Observações:		
Adicionar departamentos			
Editar departamentos			
Adicionar FAQ			
Editar FAQ			
Relatórios			
Sair			

Limpar campos Abrir chamado

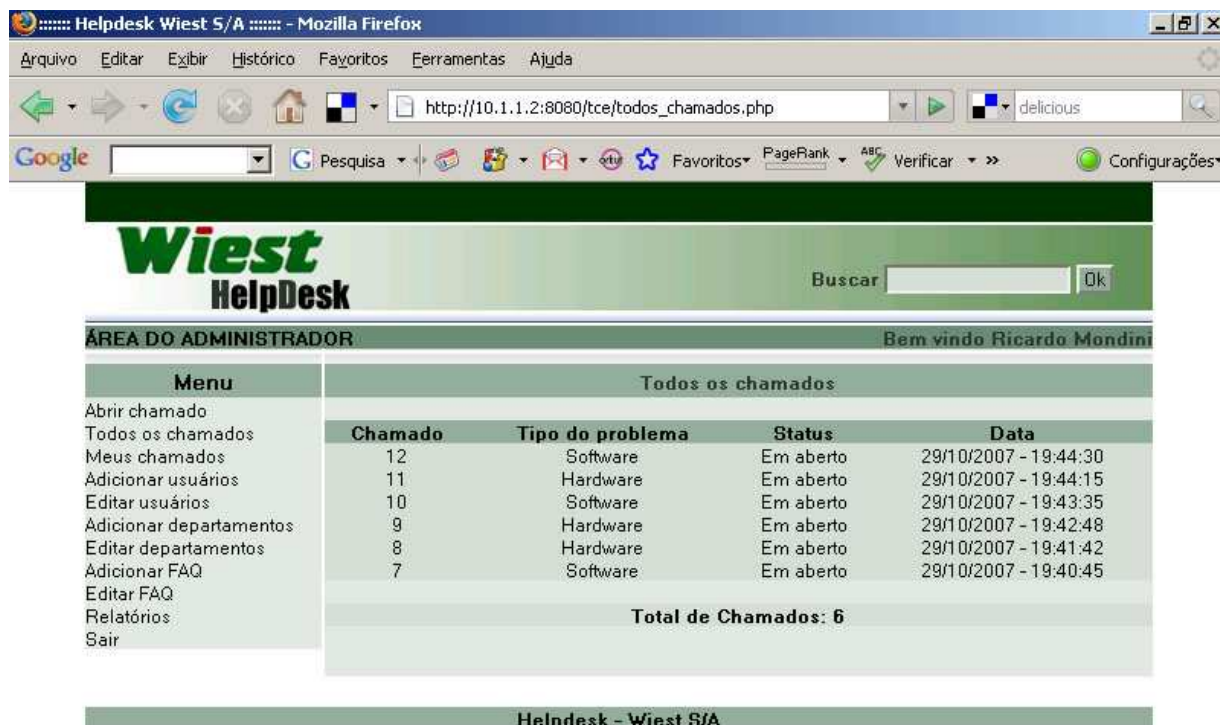
Helpdesk - Wiest S/A

Concluído

Figura 27 – Tela de Abertura de Chamados - Administrador
 Fonte – Ricardo Mondini (2007)

Na Figura 27, o administrador tem a sua disposição a abertura de chamado. Para os casos urgentes ou quando um computador de usuário não ligar, impossibilitando o próprio usuário abrir o chamado.

Nesta tela o administrador insere o tipo de problema, “HARDWARE” ou “SOFTWARE”, logo após descreve o problema que está ocorrendo no campo “DESCRIÇÃO DO PROBLEMA” e adiciona possíveis comentários no campo “OBSERVAÇÃO”.



Concluído.

Figura 28 – Tela de Todos os Chamados - Administrador
Fonte – Ricardo Mondini (2007)

Na Figura 28 o administrador acessa todos os chamados abertos por outro administrador e por todos os usuários. Ao clicar no número do chamado, o administrador visualiza uma popup, Figura 29, e visualiza os dados do chamado aberto para assumi-lo caso seja de sua área.

http://10.1.1.2:8080 - Chamados Listados - Mozilla Firefox

Lista de chamados

Operador:	Angelita Simas	Ramal:	5122
		Setor:	Financeiro
Data de abertura:	29/10/2007 - 19:40:45	Status:	Em aberto
Tipo do problema:	Software		
Descricao do problema:	Liberar acesso para MAN0369		
Observacoes:			

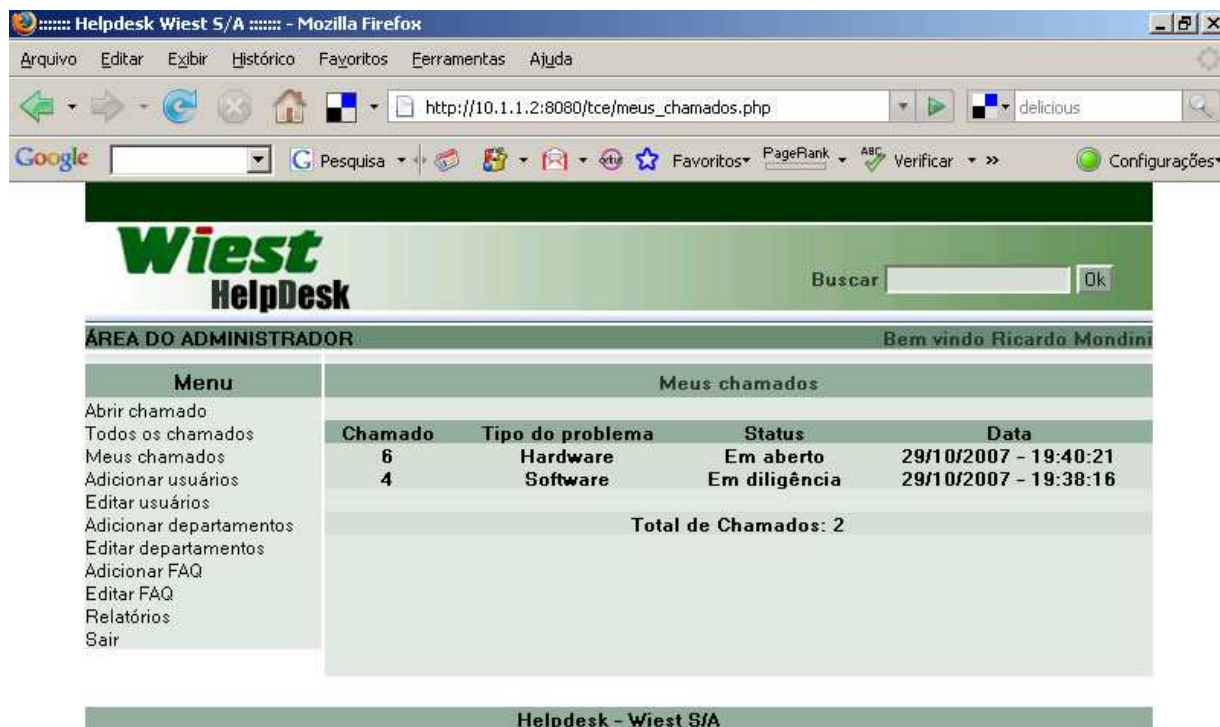
Assumir chamado

Concluído

Figura 29 – Tela de Lista de Chamados - Administrador
Fonte – Ricardo Mondini (2007)

A tela de Lista de Chamados, disposto na Figura 29, é aberta assim que o administrador clica no código do chamado, podendo assim, assumir o chamado para si.

A partir do momento que o chamado foi assumido, ele vai para a área de Meus chamados do administrador que assumiu. Pode ser visualizada na Figura 30.



http://10.1.1.2:8080/tce/editar_usuario.php

Figura 30 – Tela de Meus Chamados - Administrador

Fonte – Ricardo Mondini (2007)

Na Figura 30 o administrador visualiza todos os chamados que assumiu anteriormente. Clicando no código do chamado, é possível visualizar detalhes do chamado para efetuar a mudança de status para “Em diligência, Cancelado ou Concluído”. Pode-se visualizar esta tela na Figura 31.

http://10.1.1.2:8080 - :::::: Meus Chamados Listados :::::: - Mozilla Firefox

Meus chamados

Operador: Regiane Gil Espinha Ramal: 5132
Setor: Financeiro

Data de abertura: 29/10/2007 - 19:38:16
Status: Em diligência ▾

Obs. Status: Esperando liberação.

Tipo do problema: Software
Descricao do problema: Liberar MAN0351
Observacoes: A Aline já solicitou via logix.

Modificar chamado

Concluído

Figura 31 – Tela de Lista de Meus Chamados - Administrador
Fonte – Ricardo Mondini (2007)

A tela de Meus Chamados, disposto na Figura 31 acima, é aberta assim que o administrador clica no código do chamado, podendo assim, atender ao chamado modificando o status e adicionando a informações sobre a mudança no campo “OBS. STATUS”.

Assim que o chamado for modificado, o usuário já pode verificar o novo status.

Helpdesk Wiest S/A - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://10.1.1.2:8080/tce/adicionar_usuario.php

Google Pesquisa Pesquisar Favoritos PageRank Verificar Configurações

Wiest HelpDesk Buscar [] Ok

ÁREA DO ADMINISTRADOR Bem vindo Ricardo Mondini

Menu	Adicionar usuário	
Abrir chamado	Nome: []	Ramal: []
Todos os chamados	E-mail: []	Setor: [Seleccione o setor...]
Meus chamados	Nível: [Seleccione o nível...]	
Adicionar usuários	Login: []	
Editar usuários	Senha: []	
Adicionar departamentos		
Editar departamentos		
Adicionar FAQ		
Editar FAQ		
Relatórios		
Sair		

Limpar campos Adicionar usuário

Helpdesk - Wiest S/A

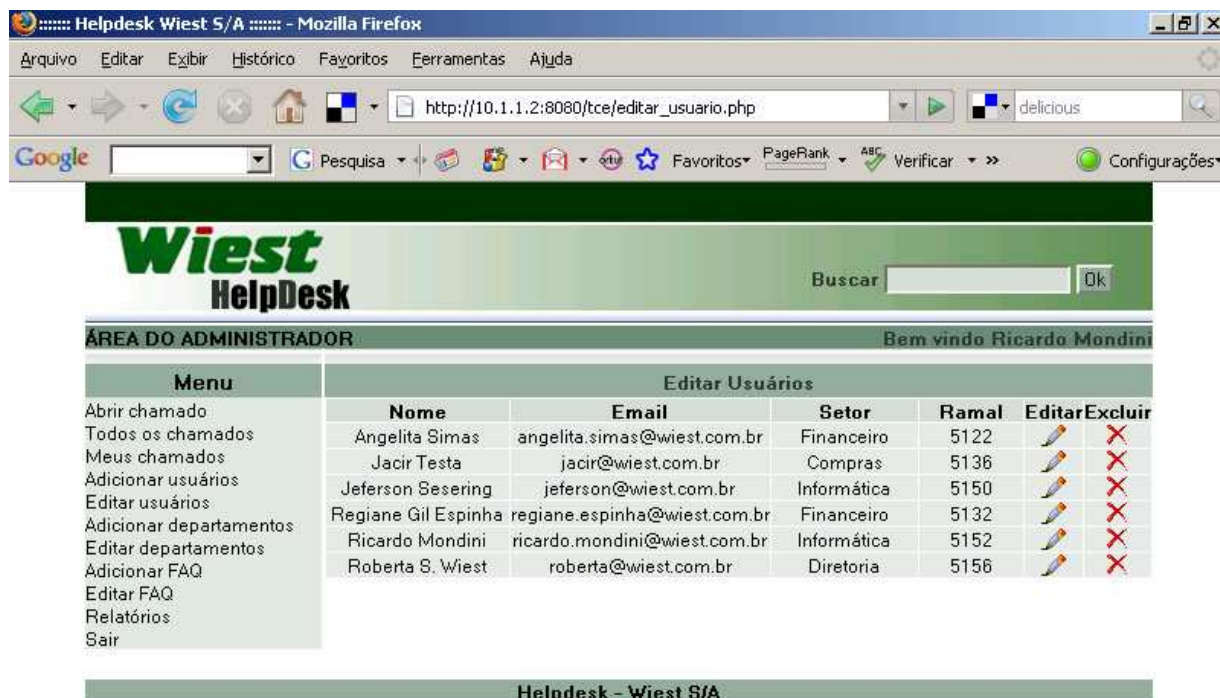
Concluido

Figura 32 – Tela de Adição de Usuário – Administrador

Fonte – Ricardo Mondini (2007)

Na Figura 32, o administrador pode adicionar usuários ou administradores no sistema.

São dispostos campos de informação sobre o novo usuário, selecionando o nível de acesso. Não são possíveis criar dois *login's* com o mesmo nome.



Concluído

Figura 33 – Tela de Editar Usuários – Administrador
 Fonte – Ricardo Mondini (2007)

Na Figura 33 o administrador edita os usuários adicionados no sistema. Ao clicar no ícone de um lápis, é possível fazer as modificações, conforme a Figura 34 demonstra.

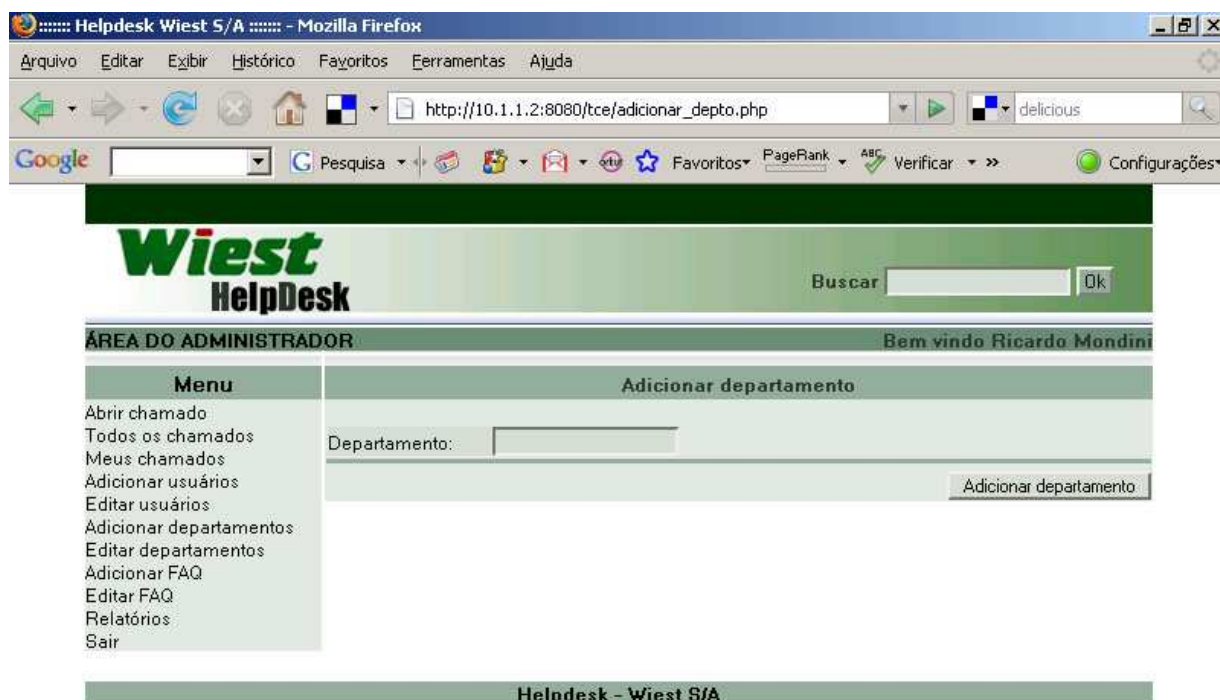
The screenshot shows a web browser window with the address bar displaying 'http://10.1.1.2:8080 - Modificar Usuario - Mozilla Firefox'. The main content area is titled 'Modificar usuário' and contains a form with the following fields:

Nome:	Ricardo Mondini	Ramal:	5152
E-mail:	ricardo.mondini@wiest.co	Setor:	Informática
Nível:	Administrador		
Login:	ricardo		
Senha:			

At the bottom right of the form is a button labeled 'Modificar usuário'. Below the form is a status bar with the text 'Concluído'.

Figura 34 – Tela de Editar Usuário – Popup – Administrador
Fonte – Ricardo Mondini (2007)

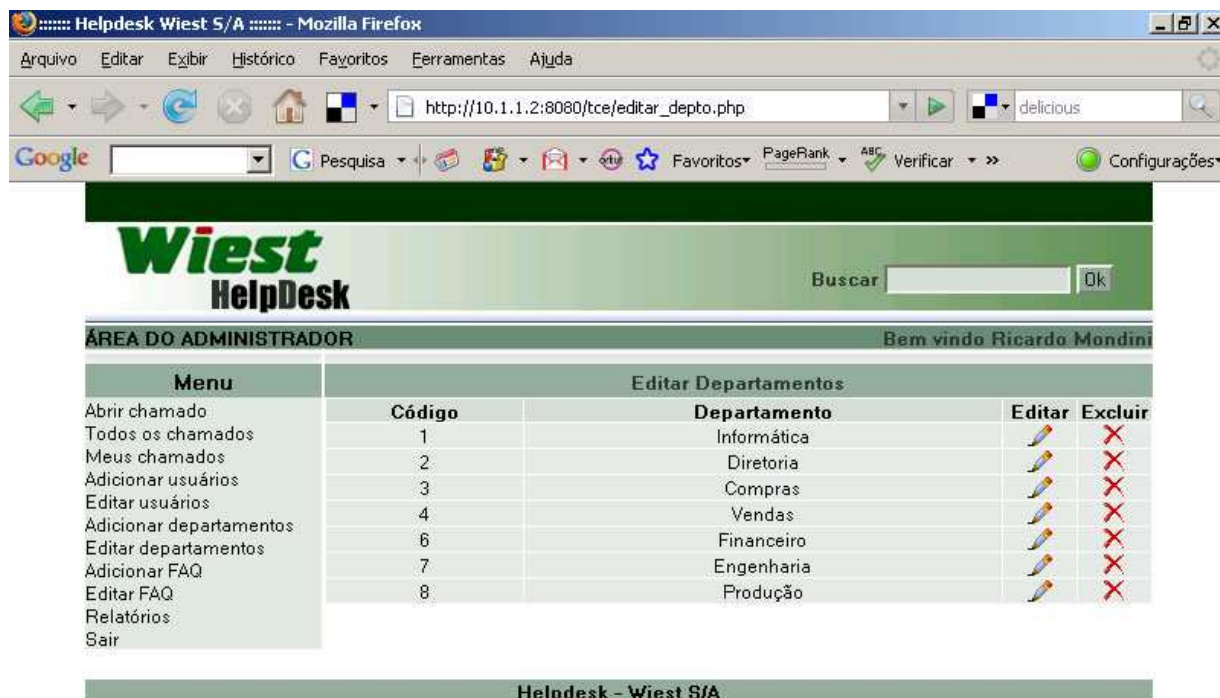
Na Figura 34 é possível efetuar a troca das informações dos usuários do sistema.



Concluido

Figura 35 – Tela de Adicionar Departamentos – Administrador
Fonte – Ricardo Mondini (2007)

Na Figura 35, o administrador pode adicionar novos departamentos no sistema.



Concluído

Figura 36 – Tela de Editar Departamentos – Administrador
Fonte – Ricardo Mondini (2007)

Na Figura 36, o administrador pode editar os departamentos do sistema. Ao clicar no ícone de um lápis, é possível fazer as modificações, conforme a Figura 37 demonstra.

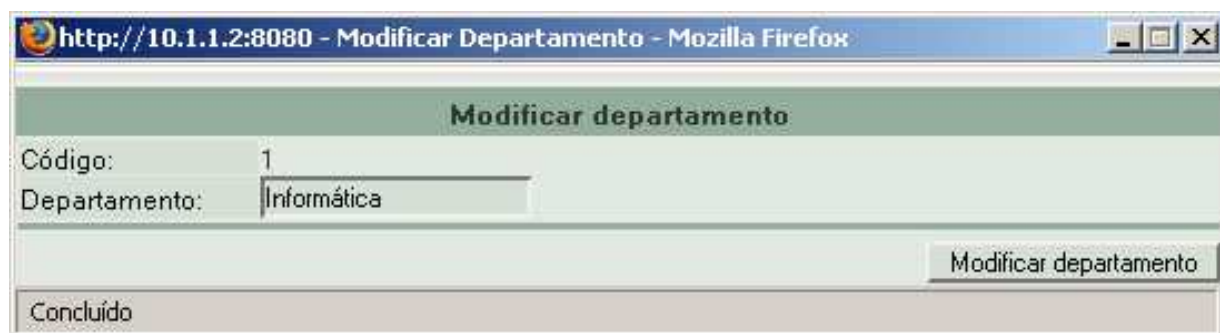
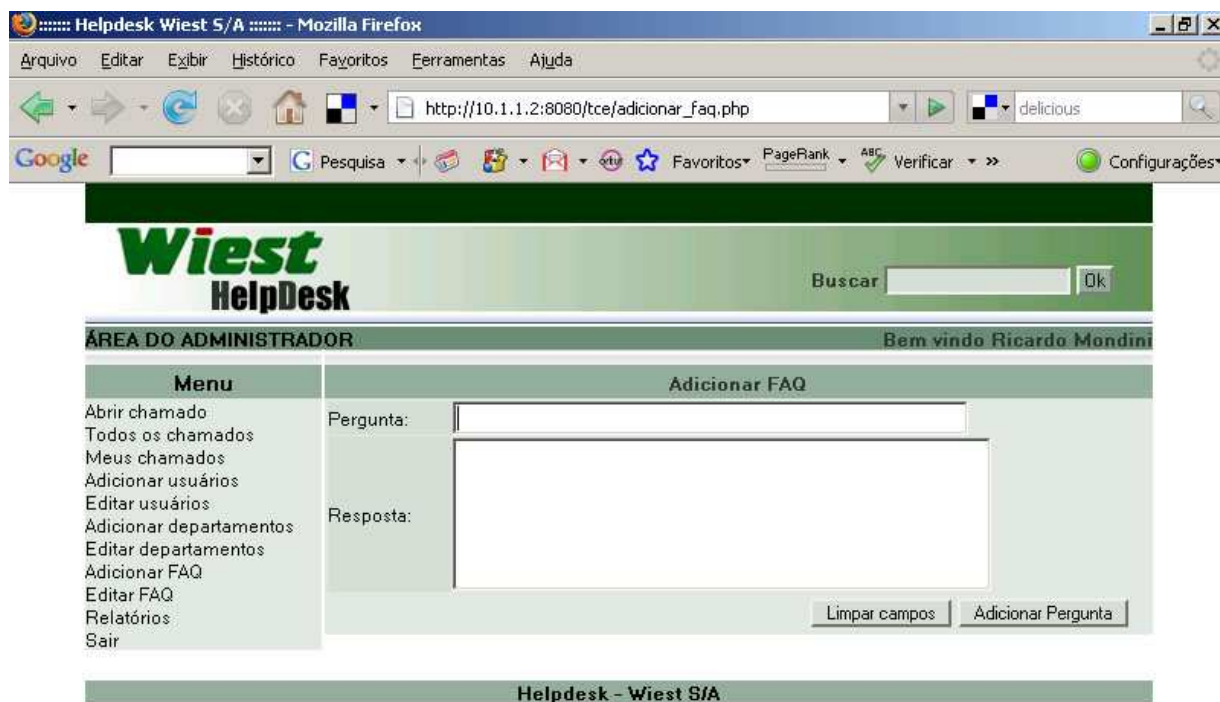


Figura 37 – Tela de Editar Departamentos – Popup – Administrador
Fonte – Ricardo Mondini (2007)

Na Figura 37 é possível visualizar a popup que permite o administrador alterar o nome do departamento.



Concluído.

Figura 38 – Tela de Adicionar FAQ – Administrador
Fonte – Ricardo Mondini (2007)

Na Figura 38, é possível visualizar a tela onde o administrador pode adicionar perguntas frequentes ao sistema.

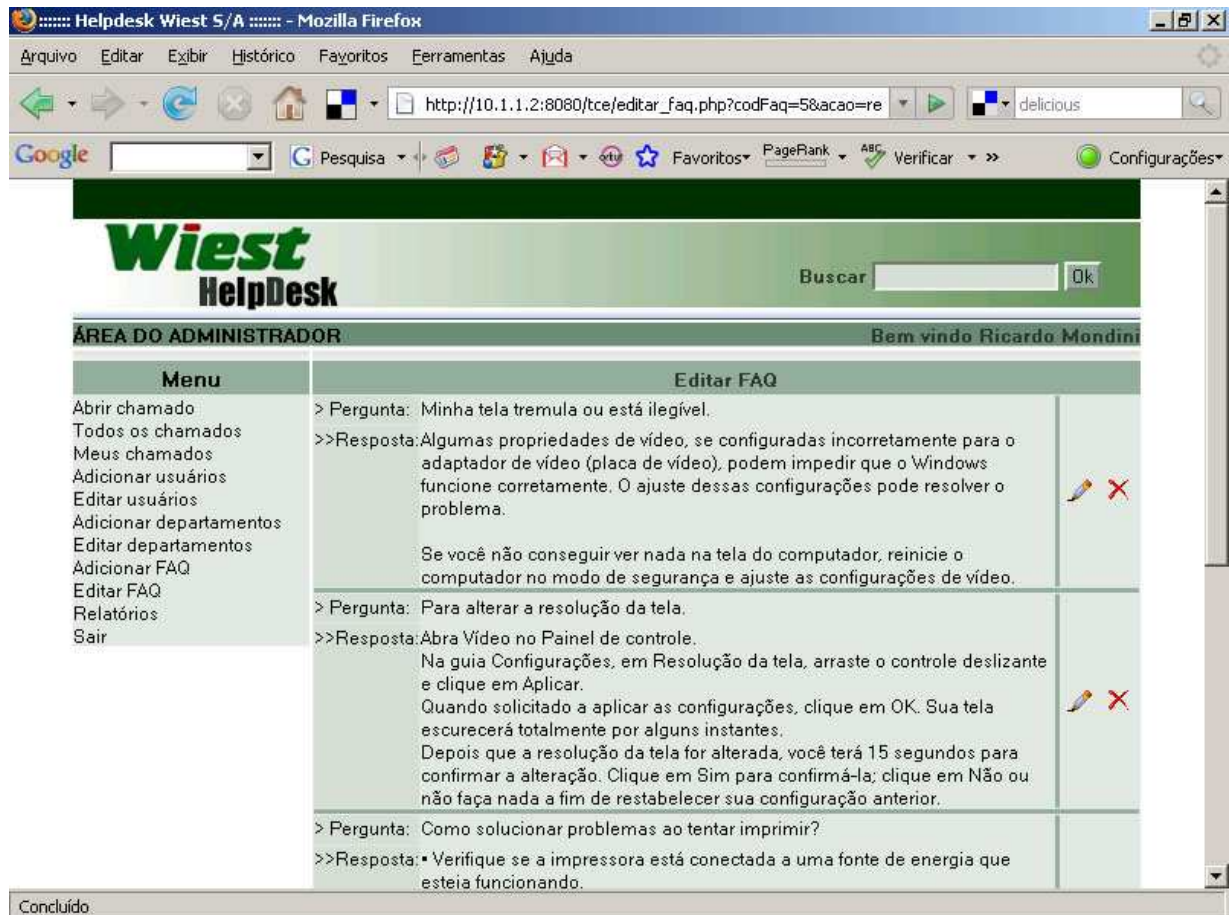


Figura 39 – Tela de Editar FAQ – Administrador
Fonte – Ricardo Mondini (2007)

Na Figura 39, o administrador pode editar as perguntas frequentes do sistema. Ao clicar no ícone de um lápis, é possível fazer as modificações, conforme a Figura 40 demonstra.

http://10.1.1.2:8080 - Modificar FAQ - Mozilla Firefox

Editar FAQ

> Código: 7

> Pergunta: Minha tela tremula ou está ilegível..

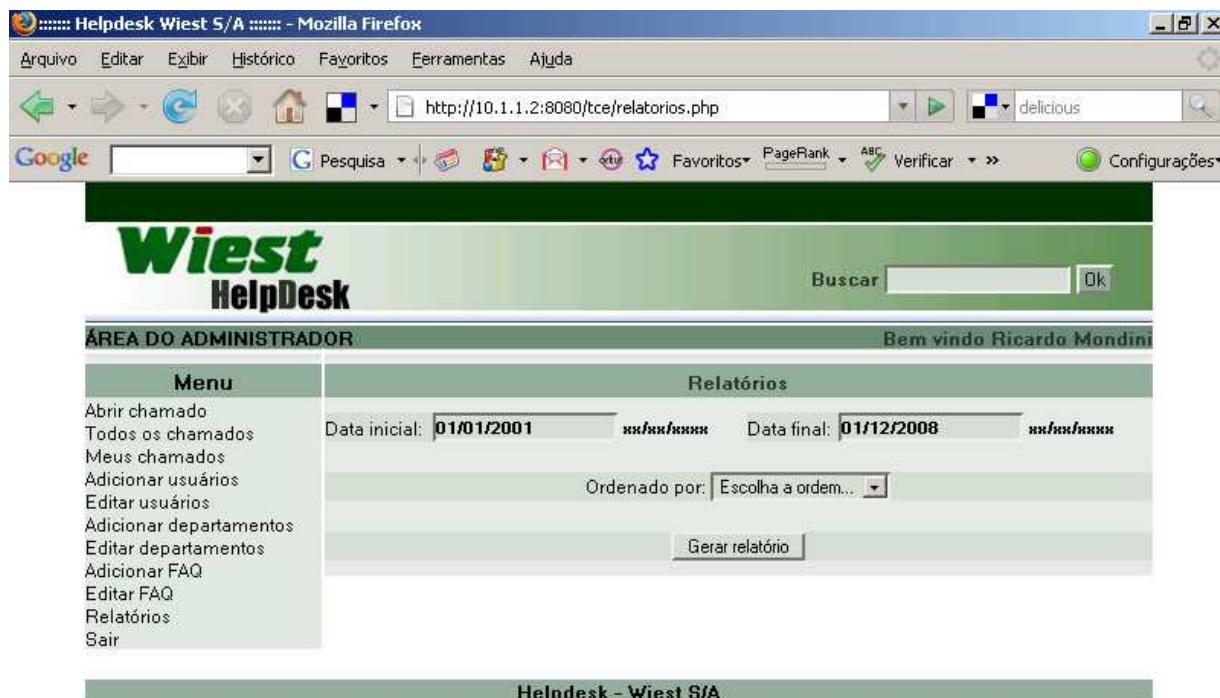
>>Resposta: Algumas propriedades de vídeo, se configuradas incorretamente para o adaptador de vídeo (placa de vídeo), podem impedir que o Windows funcione corretamente. O ajuste dessas configurações pode resolver o

Editar Faq

Concluído

Figura 40 – Tela de Editar Faq – Popup – Administrador
Fonte – Ricardo Mondini (2007)

Na Figura 40 é possível visualizar a popup que permite o administrador alterar os dados da pergunta frequente.



Concluído.

Figura 41 – Tela de Relatórios
Fonte – Ricardo Mondini (2007)

Na Figura 41 é possível visualizar a tela onde o administrador cria o relatório na ordem e na data que ele especificar.

Após a escolha, é aberta uma nova janela, Figura 42, com os dados do relatório, e possibilitando a impressão.

Relatórios - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://10.1.1.2:8080/tce/imprimir_relatorio.php delicious

Google Pesquisa Pesquisar Favoritos PageRank Verificar Configurações

Helpdesk Wiest S/A

Relatórios

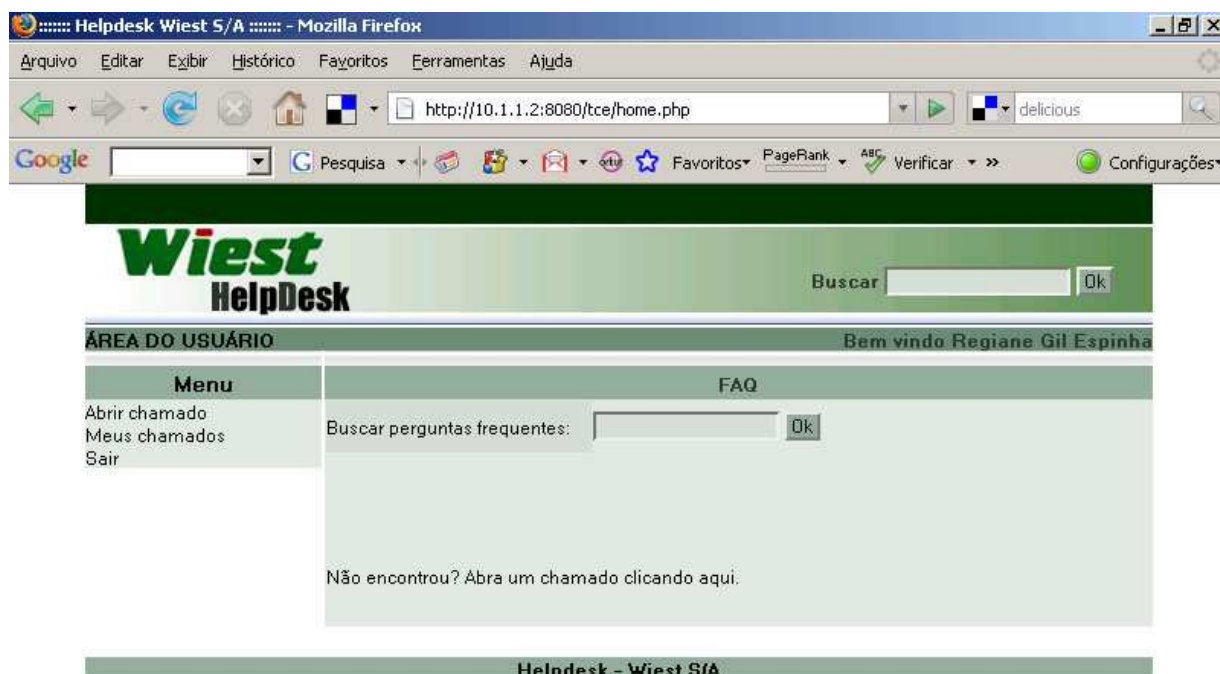
Relatório de chamados					
Nr	Usuário	Problema	Status	Setor	Data
8	Jacir Testa	Hardware	Em aberto	Compras	29/10/2007
9	Jacir Testa	Hardware	Em aberto	Compras	29/10/2007
10	Jacir Testa	Software	Em aberto	Compras	29/10/2007
11	Roberta S. Wiest	Hardware	Em aberto	Diretoria	29/10/2007
12	Roberta S. Wiest	Software	Em aberto	Diretoria	29/10/2007
3	Regiane Gil Espinha	Hardware	Concluído	Financeiro	29/10/2007
4	Regiane Gil Espinha	Software	Em diligência	Financeiro	29/10/2007
6	Angelita Simas	Hardware	Em aberto	Financeiro	29/10/2007
7	Angelita Simas	Software	Em aberto	Financeiro	29/10/2007

imprimir

Concluído

Figura 42 – Tela de Exibição dos dados do Relatório – Administrador
 Fonte – Ricardo Mondini (2007)

3.5.4.2 Área do usuário



Concluído

Figura 43 – Tela inicial e de FAQ – Usuário

Fonte – Ricardo Mondini (2007)

Na Figura 43, é possível visualizar a área onde o usuário pesquisa por problemas que já ocorreram para que possam resolver algum possível problema sem a necessidade de abrir um novo chamado para um responsável no Departamento de Informática, fazendo assim com que não haja a repetição de chamados para o mesmo problema.

A pesquisa irá abranger todas as perguntas frequentes adicionadas ao sistema pelo administrador.

Wiest HelpDesk

ÁREA DO USUÁRIO Bem vindo Regiane Gil Espinha

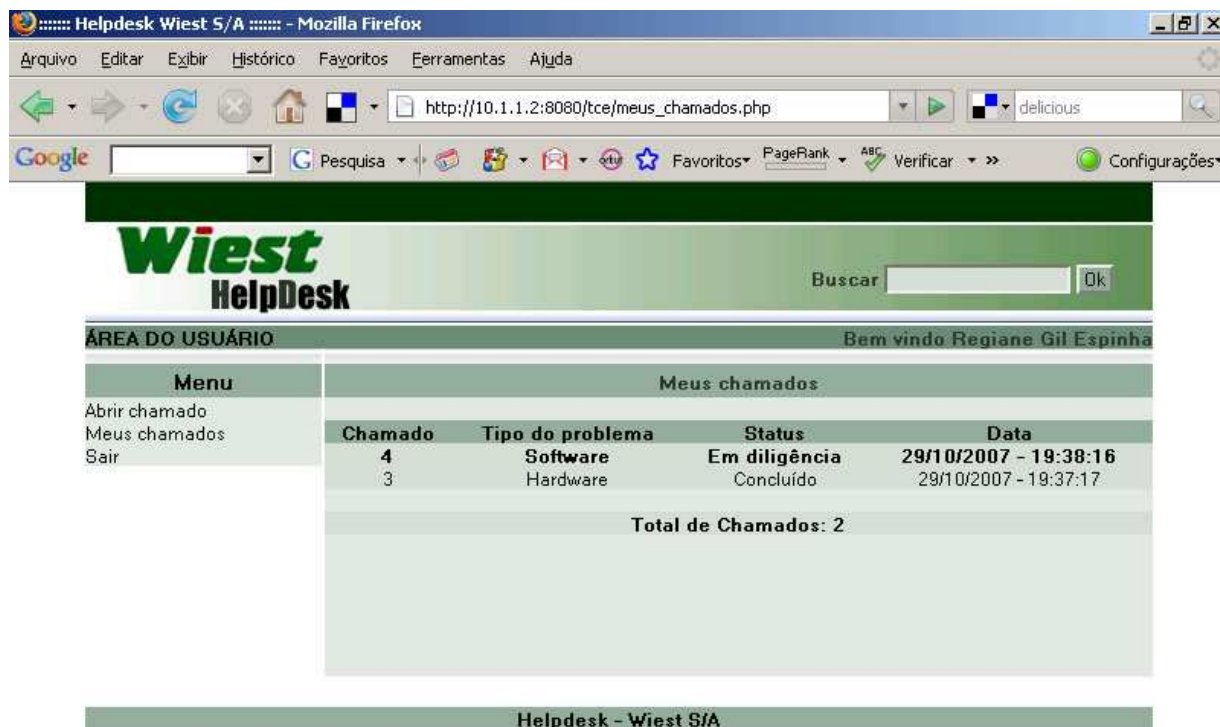
Menu	Abrir chamado		
Abrir chamado	Operador:	Regiane Gil Espinha	Ramal: 5132
Meus chamados	Data de abertura:	05/11/2007 - 16:55:11	Setor: Financeiro
Sair	Tipo do problema:	Selecione área...	Status: Em aberto
	Descrição do problema:		
	Observações:		

Limpar campos Abrir chamado

Helpdesk - Wiest S/A

Figura 44 – Tela de Abertura de Chamado – Usuário
 Fonte – Ricardo Mondini (2007)

Na Figura 44 é visualizado a tela onde o usuário informa os dados para abrir o chamado ao suporte. Nela, o usuário insere o tipo de problema, “HARDWARE” ou “SOFTWARE”, logo após descreve o problema que está ocorrendo no campo “DESCRIÇÃO DO PROBLEMA” e adiciona possíveis comentários no campo “OBSERVAÇÃO”. Então clica no botão “Abrir chamado” para efetuar a abertura.



Concluído

Figura 45 – Tela de Meus Chamados – Usuário
 Fonte – Ricardo Mondini (2007)

Na Figura 45 o usuário visualiza seus todos os seus chamados, podendo cancelar algum chamado aberto por ele caso necessário. É possível efetuar as modificações clicando sobre o código do chamado, conforme Figura 46:

http://10.1.1.2:8080 - :::::: Meus Chamados Listados :::::: - Mozilla Firefox

Meus chamados

Operador: Regiane Gil Espinha Ramal: 5132
Setor: Financeiro

Data de abertura: 29/10/2007 - 19:37:17
Status: Concluído

Obs. Status:

Efetuando compra de um novo mouse.
{-- Editado --}
Compra efetuada

Tipo do problema: Hardware
Descricao do problema: Meu mouse não está funcionando.
Observacoes:

Modificar chamado

Concluído

Figura 46 – Tela de Lista Meus Chamados – Usuário
Fonte – Ricardo Mondini (2007)

Após o chamado ser concluído não será possível alterar o status.

CONSIDERAÇÕES FINAIS

O presente trabalho de conclusão de estágio, teve por objetivo a apresentação dos resultados obtidos durante a realização do estágio curricular supervisionado, cujo projeto foi o desenvolvimento de um Sistema Helpdesk para controle dos problemas ocorridos na empresa Wiest S/A, propondo um sistema que possibilite aos colaboradores abrir um chamado sem a necessidade de uso telefônico e fazendo com que haja a interação com o andamento da resolução do problema e o acesso aos problemas ocorridos anteriormente.

O primeiro passo do projeto foi realizar uma fundamentação teórica, revisando conceitos. Nesta fase foram decididas quais as ferramentas seriam utilizadas para análise e desenvolvimento. A segunda etapa do projeto foi colocar em prática todos os conceitos levantados anteriormente. Para análise foram utilizados diagramas de representação baseados na UML, e para o desenvolvimento foram utilizadas as linguagens para *web*, PHP, Javascript e HTML e banco de dados MySQL. O terceiro passo foi o desenvolvimento do sistema através da metodologia estudada na fundamentação teórica.

Com a conclusão da aplicação do Sistema Helpdesk, será possível visualizar com clareza quais os principais problemas ocorridos, por setor, fazendo com que a decisão do gerente da área tenha mais fundamentação e credibilidade para junto aos diretores da empresa. Será possível também, solicitar treinamentos específicos a módulos no sistema atual (Logix) com mais facilidade através da análise dos chamados abertos referentes ao sistema Logix.

O projeto proporcionou grande conhecimento ao aluno deste trabalho, pois foi colocada em prática a programação, a escrita deste documento e a utilização de novas ferramentas, utilizando-se do conhecimento adquirido dentre as várias disciplinas vistas em sala no curso de Sistemas de Informação, principalmente, metodologia científica, banco de dados, programação, engenharia de software etc.

Este projeto irá continuar a crescer dentro da empresa Wiest S/A, para futuramente existir um filtro de perguntas frequentes automático, de acordo com a frequência do erro e a criação de novos relatórios com gráficos para dar ainda mais facilidade para a análise.

Assim, o projeto do Sistema Helpdesk, proporcionará agilidade aos analistas, devolvendo-lhes o tempo que era perdido na resolução de problemas já ocorridos, facilidade na decisão de compras de novos equipamentos e treinamentos além de um maior reconhecimento aos profissionais do Departamento de TI.

BIBLIOGRAFIA

ANSELMO, Fernando. **PHP e MySQL para Windows**. Florianópolis: VisualBooks: 2000.

AUDY, Jorge L. N.; ANDRADE, GilbertoK.; CIDRAL, Alexandre. **Fundamentos de Sistemas de informação**. Porto Alegre: Bookman, 2005.

BENETT, Gordon. **Intranets. Como implantar com sucesso na sua empresa**. Rio de Janeiro: Campus, 1997.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML, guia do usuário**. Rio de Janeiro: Campus, 2000.

CUMMINS, Fred A. **Integração de Sistemas**. Rio de Janeiro: Campus, 2002.

DAVID, Marcio Frayze. **Programação Orientada a Objetos: uma introdução**. Disponível em: < <http://www.guiadohardware.net/artigos/programacao-orientada-objetos/>>. Acesso em: 26 nov. 2007.

FOWLER, Martin; SCOTT, Kendal. **UML essencial**. Porto Alegre: Bookman, 2000.

FURLAN, J. D. **Modelagem de objetos através de UML: análise e desenhos orientados a objeto**. São Paulo: Markron, 1999.

KROENKE, David M. **Banco de dados: Fundamentos, Projeto e Implementação**. Rio de Janeiro: LTC, 1999.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Gerenciamento de Sistemas de Informação**. Rio de Janeiro: LTC, 2001.

LARMAN, Craig. **Utilizando UML e Padrões**. Porto Alegre: Bookman, 2000.

LEE, Richard C.; TEPFENHART, William M. **UML e C++: guia pratico de desenvolvimento orientado a objeto**. São Paulo: MAKRON Books, 2001.

MASLAKOWSKI, Mark. **Aprenda em 21 dias. MySQL**. Rio de Janeiro: Campus, 2000.

MARTINS, J. C. C. **Gestão de projetos de desenvolvimento de software**. Rio de Janeiro: Brasport, 2002.

NORTON, Peter **Introdução aos computadores**. New York: Mcgraw-Hill, 1994.

NIEDERAUER, Juliano. **Integrando PHP 5 com MySQL**. São Paulo: Novatec, 2005.

O'BRIEN, James A. **Sistemas de Informação e as decisões gerenciais na era da Internet**. São Paulo: Saraiva, 2002.

OLIVEIRA, Jayr Figueiredo de. **Metodologia para Desenvolvimento de Projetos de Sistemas**. São Paulo: Érica, 1997.

SHMITZ, E. A.; SILVEIRA, D. S. **Desenvolvimento de software orientado a objetos utilizando UML e Delphi 5**. Rio de Janeiro: Brasport, 2000.

SCOTT, Kendall. **O Processo Unificado Explicado**. Porto Alegre: Bookman, 2003.

SICA, Carlos. **PHP Orientado a Objetos - Fale a Linguagem da Internet**. Rio de Janeiro: Ciência Moderna, 2006.

STAIR, Ralph M. **Princípios de sistemas de informação: uma abordagem gerencial**. Rio de Janeiro: LTC, 2002.

YARGER, Randy Jay; REESE, George; KING, Tim. **MySQL e mSQL**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2000.

WIKIPEDIA, **HelpDesk**. Disponível em:

< http://pt.wikipedia.org/wiki/Help_desk >. Acesso em: 6 nov. 2007.

WIKIPEDIA, **Servidor Apache**. Disponível em:

< http://pt.wikipedia.org/wiki/Servidor_Apache >. Acesso em: 15 dez. 2007.

WIKIPEDIA, **Servidor Web**. Disponível em:

< http://pt.wikipedia.org/wiki/Servidor_web >. Acesso em: 15 dez. 2007.

WIKIPEDIA, **SGDB**. Disponível em:

< <http://pt.wikipedia.org/wiki/SGDB> >. Acesso em: 15 dez. 2007.