

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**GERENCIADOR DE INSTALAÇÃO DE SOFTWARES POR**  
**REDE**

**ANDRÉ BONATTI**

**BLUMENAU**  
**2013**

**2013/1-02**

**ANDRÉ BONATTI**

**GERENCIADOR DE INSTALAÇÃO DE SOFTWARES POR  
REDE**

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

Prof. Francisco Adell Péricas, Mestre - Orientador

**BLUMENAU  
2013**

**2013/1-02**

# **GERENCIADOR DE INSTALAÇÃO DE SOFTWARES POR REDE**

Por

**ANDRÉ BONATTI**

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

Presidente: \_\_\_\_\_  
Prof. Francisco Adell Péricas, Me. – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Roberto Heinzle, Dr. – FURB

Membro: \_\_\_\_\_  
Prof. Antonio Carlos Tavares, Me. – FURB

Blumenau, 05 de julho de 2013.

Dedico este trabalho a minha família, aos amigos, e todos que me ajudaram diretamente na realização deste.

## **AGRADECIMENTOS**

À minha família, que teve toda a compreensão para os momentos difíceis passados.

Aos meus amigos, que me ajudaram e apoiaram de alguma maneira para a conclusão do trabalho.

Ao meu orientador, professor Francisco Adell Péricas, por ter acreditado na conclusão deste trabalho.

Um raciocínio lógico leva você de A a B. A imaginação leva você a qualquer lugar que quiser.

Albert Einstein

## RESUMO

Este trabalho apresenta uma forma de automatização via gerenciamento de rede, da tarefa de padronizar e gerenciar os softwares instalados nos computadores em uma rede. O trabalho foi desenvolvido para utilização em ambientes de rede Windows, em plataforma *web*, para utilização de modo administrativo pelo gerenciador da rede, e em *desktop*, para os computadores poderem executar as ações solicitadas em modo autônomo.

Palavras-chave: Gerência de rede. Sistema de instalação. Padronização de sistemas. Sistemas *web*.

## **ABSTRACT**

This paper presents a form of automation via network management, of the task of standardize and manage the software installed on computers in a network. The work was developed for use in Windows network environments, web platform administrative mode for the network manager, and desktop computers, can perform the requested actions in standalone mode.

Key-words: Network management. Systems installer. Patterning systems. Web systems.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Esquema de gerência de rede .....	16
Figura 2 – Tela de impressão de saída de informações para inventário de softwares .....	21
Figura 3 – Tela de gerência do computador .....	22
Figura 4 – Diagrama de atividades software gerenciador .....	24
Figura 5 – Diagrama de atividades Phantom.....	25
Quadro 1 – Requisitos funcionais.....	25
Quadro 2 – Requisitos não funcionais.....	26
Figura 6 – Diagrama de casos de uso .....	27
Figura 7 – Modelo de entidade relacionamento .....	28
Figura 8 – Software Phantom software de apoio ao Gerenciamento .....	31
Figura 9 – Software de Gerenciamento de software, desenvolvido na plataforma JEE.....	32
Figura 10 – Página inicial. Exemplo de utilização da biblioteca Primefaces.....	33
Figura 11 – Arquivo de configuração XML do Spring framework.....	34
Figura 12 – Classe DAO.....	35
Figura 13 – Classe ComputadorDao.....	35
Figura 14 – Arquivo de configuração XML de pool de conexão com banco de dados .....	36
Figura 15 – Opções de operacionalidade software phantom.....	37
Figura 16 – Configuração geral no software phantom .....	38
Figura 17 – Configuração de proxy no software phantom .....	39
Figura 18 – Código fonte da identificação do computador .....	40
Figura 19 – Tela de <i>login</i> software de gerenciamento .....	41
Figura 20 – Tela inicial do software de gerenciamento.....	41
Figura 21 – Tela <i>pop up</i> da lista de operações a serem executadas.....	42
Figura 22 – Tela <i>pop up</i> da lista de softwares instalados no computador.....	42
Figura 23 – Tela <i>pop up</i> trazendo mais informações do computador .....	43
Figura 24 – Classe de controle de protocolo .....	44
Figura 25 – Tela Ação de instalação de software.....	45
Figura 26 – Tela de “Scan”.....	45
Quadro 3 – Dicionário de dados da tabela “sistema” .....	53
Quadro 4 – Dicionário de dados da tabela “usuario” .....	53
Quadro 5 – Dicionário de dados da tabela “historico_operacoes” .....	53

Quadro 6 – Dicionário de dados da tabela “Computador” .....	54
Quadro 7 – Dicionário de dados da tabela “fila_operacoes” .....	54
Quadro 8– Dicionário de dados da tabela “fila_operacoes” .....	55

## LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

AD - *Active Directory*

API – *Application Programming Interface*

ARP - *Address Resolution Protocol*

DML - *Data Manipulation Language*

DNS - *Domain Name System*

DSC – Departamento de Sistemas e Computação

EI - Empreendedores Individuais

HTML - *Hypertext Markup Language*

ICMP - *Internet Control Message Protocol*

IDE - *Integrated Development Environment*

JCP - *Java Community Process*

JEE - *Java Enterprise Edition*

JSE - *Java Standard Edition*

JSF - *Java Server Faces*

JVM - *Java Virtual Machine*

LAN - *Local Area Network*

MER – Modelo entidade/relacionamento

MVC - *Model View Controller*

OO - Orientação a Objeto

PC - *Personal Computer*

PIB - Produto Interno Bruto

SGBD - Sistema de Gerenciamento de Banco de Dados

SIS – Bacharelado em sistemas de informação

SQL - *Structured Query Language*

TCC - Trabalho de Conclusão de Curso

TI – Tecnologia da Informação

TTL - *Time To Live*

WAN - *Wide Area Network*

XML - *Extensible Markup Language*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 PROBLEMA .....	13
1.2 OBJETIVOS DO TRABALHO .....	13
1.3 ESTRUTURA DO TRABALHO .....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
2.1 GERÊNCIA DE REDES .....	15
2.2 PROTOCOLO ICMP.....	16
2.3 REDES LAN E WAN .....	17
2.4 SISTEMAS WEB .....	18
2.5 JAVA SERVER FACES .....	19
2.6 TRABALHOS CORRELATOS .....	20
<b>3 DESENVOLVIMENTO .....</b>	<b>23</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES .....	23
3.2 ESPECIFICAÇÃO .....	25
3.2.1 REQUISITOS FUNCIONAIS.....	25
3.2.2 REQUISITOS NÃO FUNCIONAIS .....	25
3.2.3 DIAGRAMA DE CASOS DE USO.....	25
3.2.4 MODELO DE ENTIDADE RELACIONAMENTO .....	25
3.3 IMPLEMENTAÇÃO .....	29
3.3.1 Técnicas e ferramentas utilizadas .....	29
3.3.1.1 Java.....	30
3.3.1.2 Bibliotecas Java .....	32
3.3.1.3 Framework Java.....	33
3.3.1.4 API Java.....	34
3.3.1.5 Banco de dados MySQL.....	36
3.3.1.6 Servidor web Glassfish.....	36
3.3.2 Operacionalidade da implementação .....	37
3.4 RESULTADOS E DISCUSSÃO .....	46
<b>4 CONCLUSÕES.....</b>	<b>47</b>
4.1 EXTENSÕES .....	47
<b>REFERÊNCIAS .....</b>	<b>49</b>

<b>APÊNDICE A – Descrição dos Casos de Uso .....</b>	<b>51</b>
<b>APÊNDICE B – Dicionário de dados.....</b>	<b>53</b>

## 1 INTRODUÇÃO

No Brasil são criados anualmente mais de 1,2 milhões de novos empreendimentos formais. Desse total, mais de 99% são micro e pequenas empresas e Empreendedores Individuais (EI).

A sobrevivência desses empreendimentos é condição indispensável para o desenvolvimento econômico do País. Todos os estudos no Brasil e no mundo mostram que os dois primeiros anos de atividade de uma nova empresa são os mais difíceis, o que torna esse período o mais importante em termos de monitoramento da sobrevivência. Os dados mais recentes mostram que a cada 100 empreendimentos criados, 73 sobrevivem aos primeiros dois anos de atividade. A taxa supera a de muitos outros países, como por exemplo, a Itália (BARETTO, 2011).

A área de TI cresce 10% ao ano, em média, e a previsão é que chegue aos 12% em 2015, ante uma expectativa de avanço do Produto Interno Bruto (PIB) nacional de 4,5% em 2011. Só no ano passado, as companhias locais investiram cerca de 90 bilhões de reais em infraestrutura de TI, valor que deverá ser superado neste ano, uma vez que mais da metade delas pretende elevar seus orçamentos até dezembro, de acordo com a IDC Brasil – consultoria especializada na análise de mercados (DELLA VALLE, 2011).

Diante das boas perspectivas do crescimento do mercado brasileiro, e do grande potencial computacional que vem diante desse constante crescimento, há uma necessidade cada vez maior de automatização e melhoria nos recursos computacionais que atendam os problemas que vem com as atuais e futuras demandas. A eficiência e até a sobrevivência de empresas, desde as grandes corporações até as empresas de médio e pequeno porte, pode ser determinada pelo bom funcionamento de seus recursos de informática. O constante aumento do tamanho das redes de computadores provoca um aumento da complexidade e diversidade de tecnologias usadas.

Dentro deste contexto, ferramentas que ajudem os administradores de redes no controle de seus equipamentos, seja na coleta de informações, no monitoramento ou na execução de tarefas automatizadas, são de vital importância para que os administradores de rede desempenhem um bom trabalho.

Frente às mudanças constantes na área da Tecnologia da Informação (TI) e ao aumento na concorrência desta área, surge a necessidade de que as organizações sejam inteligentes e que se modifiquem, o que requer um melhor planejamento de suas informações, de seu

conhecimento e de sua tecnologia da informação. Levando isso em consideração pode-se chegar à conclusão de que as empresas lidam a cada dia com mais dados, os quais precisam ser devidamente trabalhados para que possam agregar algum valor à organização a fim de gerar conhecimento sempre que necessário (REZENDE, 2003, p. 39).

## 1.1 PROBLEMA

Com o constante aumento das redes de computadores, fica imprescindível a sua administração, pois acaba sendo inviável qualquer controle manual.

Um dos grandes problemas encontrados nas empresas, seja em grandes corporações ou médias e pequenas empresas, é a falta de padronização e facilidade de obtenção de informação sobre a rede. A instalação e desinstalação de softwares, que ocorrem nos equipamentos das empresas, podendo até mesmo ser executado por pessoas autorizadas a utilizar o PC, por vários meios (*Pen drive*, CD, DVD, Internet, etc.) instalarem softwares maliciosos, ou mesmo softwares ilegais, que podem causar danos diretamente, como roubo de informações, mau funcionamento do equipamento, ou indiretamente, ocasionando eventualmente um processo judicial por pirataria além do custo, denegrindo a imagem da empresa.

O software “gerenciador de instalação de software por rede” proposto no trabalho tem como seu propósito facilitar a padronização dos softwares para administra-los por meio de um sistema automatizado. Com ele se ganha o controle sobre os softwares instalados, em todos os PCs da rede, economiza tempo, e tem menos chances de ocorrer erros por ser feito por um processo automatizado e normalizado. Para que não ocorra falha retirando a responsabilidade, de um técnico ficar instalando ou desinstalando software de PC em PC, para ser realizado em vários PCs simultaneamente, por um processo autônomo.

## 1.2 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é o desenvolvimento de um sistema para gerenciamento de instalação de software por rede.

Os objetivos específicos do trabalho são:

- a) manter o registro dos softwares de cada PC da rede;
- b) permitir as ações de instalação, desinstalação e identificação dos PCs da rede;
- c) registrar o histórico das ações realizadas.

### 1.3 ESTRUTURA DO TRABALHO

Este trabalho está organizado em quatro capítulos, sendo que, no primeiro, é apresentada a introdução, os objetivos específicos e como o trabalho está estruturado.

No segundo capítulo apresenta-se a fundamentação teórica pesquisada sobre gerência de redes, protocolo ICMP, redes LAN e WAN, sistemas *web*, Java Server faces, e trabalhos correlatos.

O terceiro capítulo apresenta o desenvolvimento do sistema iniciando-se com o levantamento de informações, tendo na sequência as técnicas e ferramentas utilizadas bem como a elaboração de alguns diagramas para auxiliar na compreensão do sistema, a operacionalidade do mesmo e resultados e discussões.

No quarto capítulo tem-se as conclusões e extensões deste trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos a serem apresentados nas seções a seguir, que fundamentam o desenvolvimento do trabalho, tais como o conceito de gerenciamento de rede, protocolo *Internet Control Message Protocol (ICMP)*, redes *Local Area Network (LAN)* e *Wide Area Network (WAN)*, *Java Server Faces*, e trabalhos correlatos.

### 2.1 GERÊNCIA DE REDES

Kurose e Ross (2005, p. 572) diz que, nos primórdios das redes de computadores, quando elas eram artefatos de pesquisa e não uma infraestrutura usada por milhões de pessoas diariamente, ‘gerenciamento de rede’ era algo que nunca se tinha ouvido falar. Se alguém descobrisse um problema de rede, poderia realizar alguns testes, como o *ping*, para localizar a fonte do problema. Com a internet pública e as intranets privadas crescendo e se transformando de pequenas redes em grandes infraestruturas globais, a necessidade de gerenciar mais sistematicamente a enorme quantidade de componentes de hardware e software dentro dessas redes também se tornou muito importante.

De acordo com Lopes, Sauv e, Nicoletti (2003, p. 4), o objetivo da Ger ncia de Redes   monitorar e controlar os elementos da rede (sejam eles f sicos ou l gicos), assegurando certo n vel de qualidade de servi o. Para realizar esta tarefa, os administradores de redes s o geralmente auxiliados por um sistema de ger ncia de redes. Um sistema de ger ncia de rede pode ser definido como uma cole o de ferramentas integradas para a monitora o e o controle da rede.

Segundo Kurose e Ross (2005, p. 575), gerenciamento de rede exige a capacidade de “monitorar, testar, consultar, configurar e controlar” os componentes de hardware e software de uma rede. Como os dispositivos da rede s o distribu dos, fazer isso exigir , no m nimo, que o administrador possa coletar dados de uma entidade remota e efetuar mudan as nessa entidade.

De acordo Lopes, Sauv e, Nicoletti (2003, p. 6), um dos objetivos da ger ncia de redes   prevenir e solucionar problemas na rede. N o existe uma regra r gida sobre os profissionais que fazem parte desta equipe, por m,   comum que nesta equipe existam profissionais que

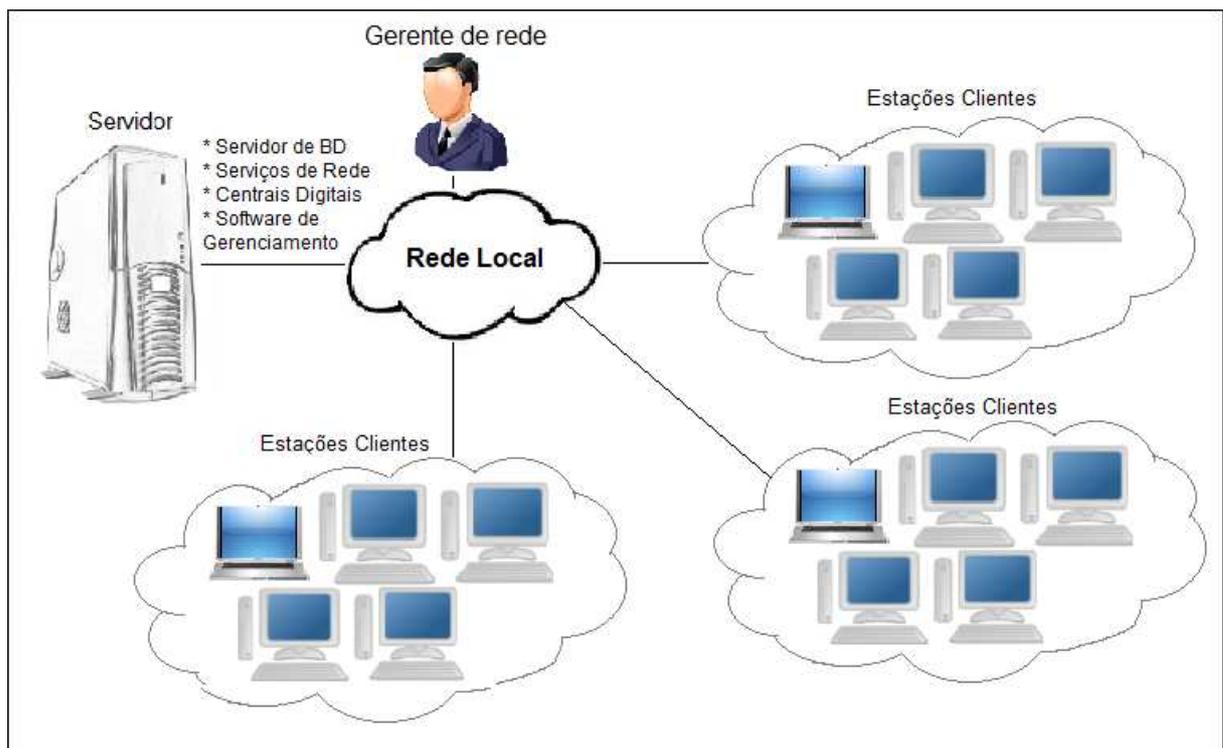
executam quatro tarefas distintas: o pessoal do *help desk*, o operador da rede, a equipe de suporte técnico e o gerente da equipe.

O *help desk* atua no auxílio direto ao usuário, tirando dúvidas, auxiliando a resolver problemas, por telefone ou remotamente. Já a equipe de suporte técnico é quem soluciona os problemas, responsável pela configuração, operação e manutenção da rede.

O operador de sistema é o profissional encarregado de acompanhar os alarmes gerados pelo sistema de gerência, e o gerente da equipe administra os recursos, avaliando o desempenho de sua equipe, solicitando a compra de equipamentos, aplicações e outros recursos necessários, solicitando treinamentos adequados para a equipe. (BAMBINETI, 2008).

Na Figura 1 vê-se um esquema representando a gerência de redes, onde os mais diversos tipos de equipamentos são gerenciados por um ou vários sistemas gerentes de redes.

Figura 1 – Esquema de gerência de rede



## 2.2 PROTOCOLO ICMP

Comer (1998) indica que o *Internet Control Message Protocol* (ICMP) é um protocolo de mensagens de controle usado basicamente para três funcionalidades: avisar quando o fluxo de mensagens é maior que a capacidade de processamento de um dispositivo; checar o número de roteadores que o pacote já percorreu (parâmetro *Time To Live* (TTL)); e mensagens de redirecionamento.

Eventualmente um roteador pode receber mais informação do que pode processar, sendo assim ele passa a contar com controle de fluxo, enviando uma mensagem *source quench* para o dispositivo de origem para que ele pare ou diminua o fluxo de dados.

O segundo caso envolve o parâmetro TTL do cabeçalho do pacote IP que basicamente é o número de roteadores, ou *hops*, total que uma informação pode percorrer. Ele é decrementado a cada *hop* e quando chega a zero, o roteador descarta o datagrama e envia uma mensagem à fonte informando que a informação não chegou ao seu destino, utilizando o ICMP.

O terceiro caso é a mensagem de redirecionamento ICMP, que é utilizada quando o roteador determina que um caminho melhor existe para o pacote que acabou de ser enviado. Neste caso a implementação do protocolo de roteamento pode definir um novo caminho.

### 2.3 REDES LAN E WAN

As redes LAN emergiram para possibilitar o compartilhamento de recursos (hardware e software) e de informações, bem como a troca de informações entre vários computadores, mantendo a independência entre as diversas estações de trabalho conectadas à rede e possibilitando a integração destas estações e seus recursos em ambientes de trabalho corporativos (SOARES; LEMOS; COLCHER, 1995).

Uma rede LAN é limitada em uma região, onde sua principal característica é não necessitar do uso de telecomunicações como meio de transmissão de dados entre seus nós, que correspondem aos próprios computadores dos usuários. Entre as características básicas das redes LANs, segundo Soares, Lemos e Colcher (1995), estão:

- a) abrangência geográfica limitada;
- b) altas taxas de transmissão;
- c) baixas taxas de erro;

- d) pequenos atrasos de transmissão;
- e) geralmente redes privadas;
- f) facilidade de interconexão entre redes distintas.

Segundo Soares, Lemos e Colcher (1995), as redes WAN são as pioneiras entre as redes de computadores. São constituídas de uma diversidade de aplicações e usos, destacando-se as redes públicas, isto é, “o sistema de comunicação é mantido, gerenciado e de propriedade de grandes operadoras (públicas ou privadas) e seu acesso é público”. As redes WAN possuem um custo elevado de comunicação, por isto a taxa de transmissão é baixa, mesmo que hoje em dia alguns *links* alcancem a velocidade de megabits por segundo (Mbps).

## 2.4 SISTEMAS WEB

Nas últimas duas décadas a *web* passou por diversas transformações, desde o princípio da utilização do *Hipertext Markup Language* (HTML) para apresentação de conteúdo estático. Com a expansão do acesso e a presença no dia a dia das pessoas, foram criadas necessidades e oportunidades para tornar a web mais interativa e atrativa, mantendo o princípio do HTML para independência de navegador e plataforma (GOODMAN; MORRISON; NOVITSKI; RAYL, 2010).

Conforme Goodman, Morrison, Novitski e Rayl (2010), novas técnicas e padrões foram introduzidos com a finalidade de melhorar a experiência de navegação e possibilitar um maior nível de interatividade entre a página e o usuário. As seguintes tecnologias podem ser mencionadas como fundamentais:

- a) *Cascading Style Sheets* (CSS), um padrão estruturado pelo *World Wide Web Consortium* (W3C) para a definição de estilo (fontes, cores e posicionamento) de elementos de uma página HTML;
- b) JavaScript, uma linguagem orientada a objetos, compilada e executada no navegador, que permite a execução de algumas operações, como validações de campos, usando a capacidade de processamento do cliente (e, portanto, liberando o servidor), manipulação da estrutura das páginas sem necessidade de recarregamento e controle da navegação;

- c) *Asynchronous JavaScript And XML (AJAX)*, uma técnica de utilização do JavaScript para efetuar requisições ao servidor em segundo plano. Ele permite que uma página seja modificada, parcialmente, sem necessidade de recarregamento ou redirecionamento;
- d) programação no lado servidor, possibilitado por tecnologias como o *Java Server Faces (JSF)*, Microsoft .NET e PHP Hypertext Processor (PHP), permitiram a introdução de conteúdo dinâmico, com acesso a bases de dados e demais recursos do servidor;
- e) *plugins*, extensões do navegador fornecidas por terceiros que ampliam sua funcionalidade. Um dos *plugins* mais utilizados é o Adobe Flash, o qual permitiu um nível mais elevado de interatividade com aplicativos embutidos no navegador que podem, por exemplo, executar vídeos, jogos e outros programas interativos.

Com a evolução das tecnologias empregadas na web, esta se tornou uma plataforma muito mais adequada para finalidades mais complexas. A introdução das tecnologias do lado servidor e do AJAX tornaram possível aos sites modernos oferecer um conjunto de recursos maior com um nível de usabilidade digno de uma aplicação nativa de desktop. A diferença cada vez mais tênue entre algumas páginas e aplicativos de desktop.

## 2.5 JAVA SERVER FACES

De acordo com Luckow e Melo (2010), o *Java Server Faces (JSF)* é a especificação para um *framework* de componentes para desenvolvimento web em Java. Essa especificação foi definida por meio do *Java Community Process (JCP)*, que é a entidade que tem como objetivo especificar a evolução da linguagem Java de acordo com o interesse do mercado e não apenas da Sun, criadora da linguagem Java. O fato de ser uma especificação do JCP significa que grandes empresas como Apache, Bea System, EDS, EMC, Fujitsu, Hewlett-Packard, IBM, Macromedia, Novell, Oracle, Siemens e Sun participaram na definição do *Java Server Faces* e aprovaram sua especificação. Isso torna o JSF imediatamente um padrão de mercado. O fato de ser especificação formal e segura permite essas empresas e outras que se interessam que invistam na tecnologia e desenvolvam ferramentas e componentes para o desenvolvimento *web* com JSF.

A atual versão da especificação do JSF é a 2.1, e conta com algumas bibliotecas de componentes prontos para utilização. As mais populares bibliotecas utilizadas pelas comunidades de desenvolvimento são:

- a) Tomahawk, Trinidad, Tobago, da Apache MyFaces;
- b) OpenFaces, da TeamDev;
- c) RichFaces, da JBoss;
- d) ICEFaces, da ICESoft;
- e) PrimeFaces, da Primefaces.

## 2.6 TRABALHOS CORRELATOS

Pode-se citar como trabalho correlato a monografia realizada pelo aluno Charles Bambineti, para conclusão do curso de Ciências da Computação na Universidade Regional de Blumenau.

O trabalho de Bambineti (2008) foi desenvolver um Sistema de *Web services* para inventário de estações em rede. O foco do trabalho foi buscar nos computadores da rede, informações dos softwares instalados Figura 2, com intenção de obter o inventário da rede.

Figura 2 - Tela de impressão de saída de informações para inventário de softwares

```

C:\WINDOWS\system32\command.com
C:\TCCSRC>wscmd.py 192.168.1.200:80 pacote lista trinity win
Microsoft Office Excel MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Office Enterprise 2007 12.0.4518.1014
VMware Tools 3.1.0000
Microsoft Office InfoPath MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Software Update for Web Folders <Portuguese <Brazil>> 12 12.0.451
8.1019
Microsoft Office Access MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Office Groove MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Office Publisher MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Office Proof <Spanish> 2007 12.0.4518.1014
Enterprise Architect 7.1 7.0.829
Microsoft Office Shared MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Office Word MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Office Proof <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Office Proof <English> 2007 12.0.4518.1014
Macromedia Dreamweaver 8 8.0.2
ActiveState Komodo Edit 4.3.2 4.3.2
Microsoft Office Proofing <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Office PowerPoint MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
Macromedia Extension Manager 1.7.240
WebFldrs XP 9.50.7523
Python 2.5.2 2.5.2150
Microsoft Office Outlook MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
Microsoft Office OneNote MUI <Portuguese <Brazil>> 2007 12.0.4518.1019
C:\TCCSRC>

```

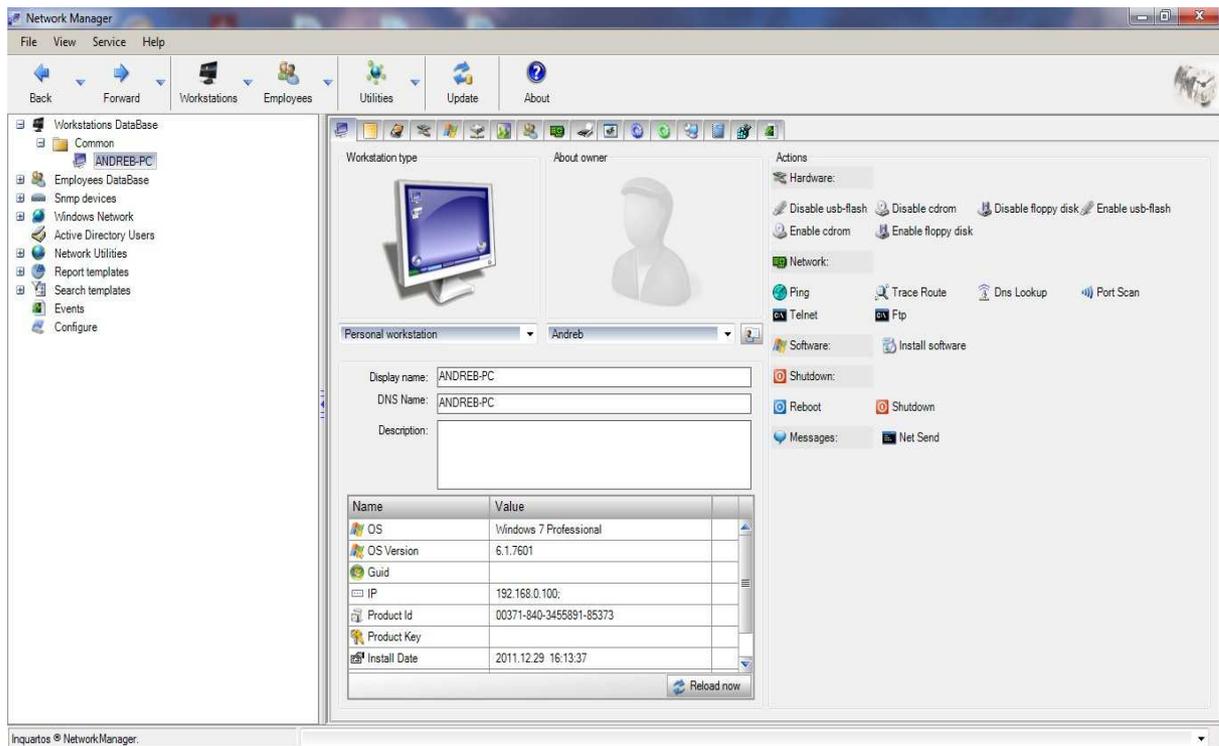
Fonte: Bambineti (2008).

Outro trabalho é o Inquartos® NetworkManager (2012), um software de gerenciamento de rede destinado a simplificar a tarefa de monitoramento, inventário e administração de computadores, com suporte Windows NT/2000/XP/2003. Não requer a instalação de qualquer software cliente em computadores remotos, Figura 3. Suas características são:

- a) obtenção de informações de computadores remotos: software instalado e componentes de hardware, as sessões, log de eventos, recursos compartilhados e assim por diante;
- b) monitoramento automático de registro e alterações de hardware e componentes de software que ocorreram em PCs remotos e administradores notificando sobre isso muda;
- c) administração remota conveniente (gestão das contas de usuários locais, serviços, processos em execução, instalação de edição de registro e desinstalação do software gestor de clientes, terminais);
- d) suporta MS SQL Server 2000/2005 o mesmo banco de dados integrado de *Extensible Markup Language* (XML) para armazenar informações de inventário;
- e) permite que os usuários facilmente criem relatórios detalhados de inventário personalizáveis e informações de inventário rápido;
- f) capacidade de aplicar ações para o grupo de computadores (instalação do software

- em modo silencioso, desligar e reiniciar computadores, gerando relatórios de inventário, envio de mensagens);
- g) contém funções de administração do Windows *Active Directory* (AD);
  - h) utilitários de rede (*ping*, rota de rastreamento, procurando anfitriões por ICMP *ping* ou *scan Address Resolution Protocol* ,(ARP), a pesquisa de *Domain Name System* (DNS), NetSend).

Figura 3 - Tela de gerência do computador



Fonte: Inquartos® NetworkManager (2012).

### 3 DESENVOLVIMENTO

Neste capítulo são apresentados o levantamento de informações, os requisitos, a especificação e a implementação do trabalho. Também são mostrados diagramas de casos de uso, diagramas de atividades e o modelo de entidade/relacionamento. Para finalizar é apresentada uma validação do trabalho utilizando as telas juntamente com códigos fonte de exemplos. Com os resultados obtidos é realizada uma comparação com os trabalhos correlatos.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

O sistema desenvolvido é uma ferramenta para gerenciamento de instalação de software pela rede, voltada a administradores de rede, visando à diminuição do tempo gasto com a tarefa e facilitando a padronização dos softwares nos computadores da rede que estão interligados a mesma. Para a viabilização da instalação, desinstalação e identificação remota, foi desenvolvido um aplicativo chamado Phantom, um software de apoio que deverá ser instalado nos computadores que serão gerenciados.

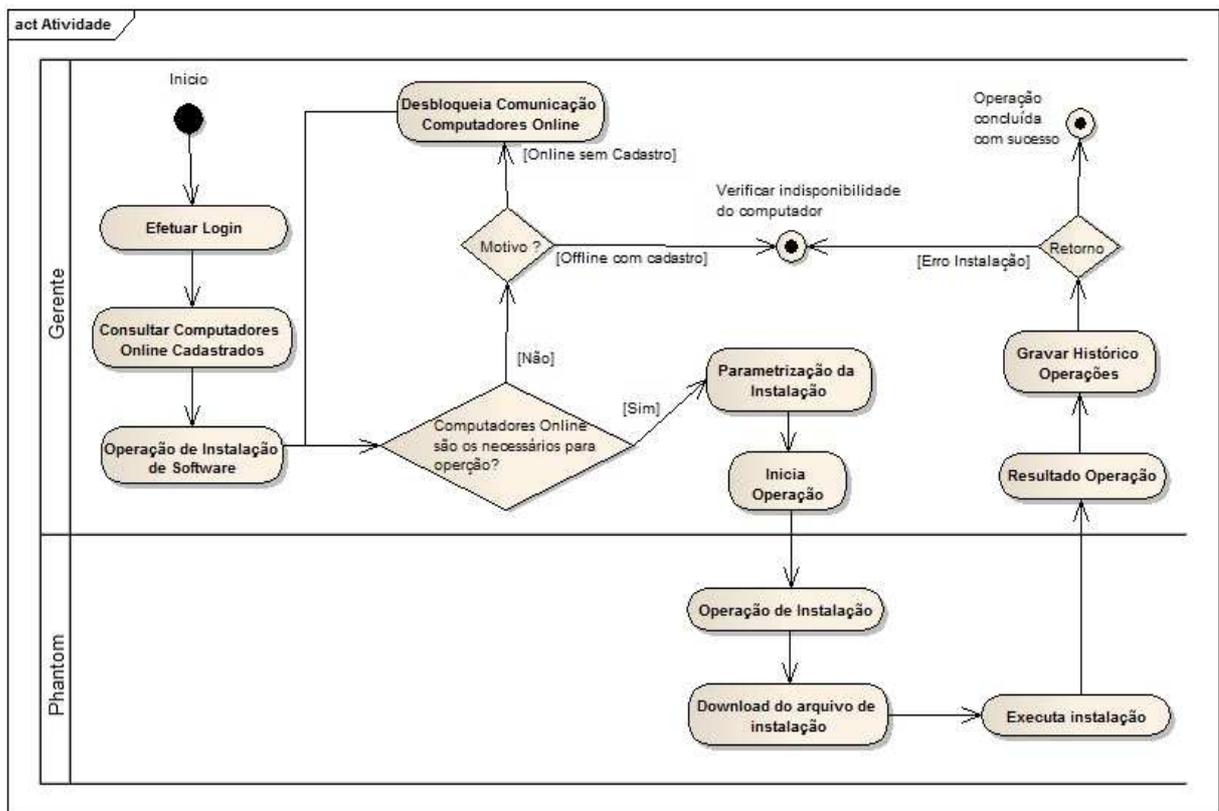
As ações são comandadas pelo gerente de rede através do software de gerenciamento que têm como objetivo distribuir as ações realizadas através do gerente para as máquinas afetadas. O software de apoio denominado de Phantom terá de ser um aplicativo que estará executando em *background* nas máquinas pela rede, que simulará um usuário fazendo ações administrativas na máquina.

O sistema de gerenciamento precisa de uma rotina autônoma de verificação de PCs na rede, ao qual o administrador de rede deverá informar a faixa inicial e faixa final de IP a ser percorrida. Para funcionar corretamente, o protocolo ICMP (*ping*) deverá estar habilitado nas máquinas e liberado para mandar repostas.

A comunicação entre os sistemas de gerenciamento e o sistema de apoio (Phantom) deverá ser efetuada através de *socket* comandado através de protocolo próprio. O sistema de gerenciamento tem o papel de Servidor, e os PCs que tiverem o software Phantom instalados se conectarão e se identificarão.

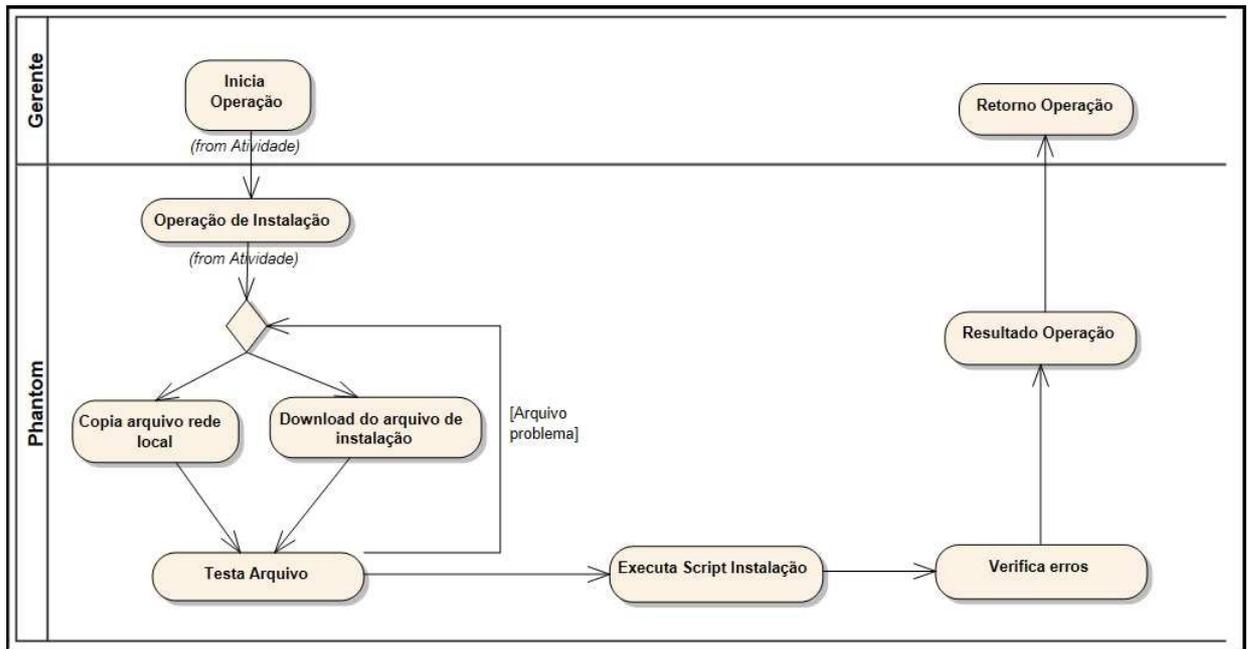
As ações (instalação, desinstalação, identificação) só poderão ser realizadas uma de cada vez no PC de destino. O sistema de gerenciamento controlará ações para vários PCs, de forma que ao constatar mais de uma ação para o mesmo PC, a ação será inserida numa lista de espera para ficar aguardando o retorno da última ação, liberando assim para a próxima ação, e assim sucessivamente. Este funcionamento está representado na Figura 4.

Figura 4 - Diagrama de atividades: função principal de instalação de software



Na Figura 5, podemos observar mais detalhado o processo de instalação pelo software phantom, desde o momento de iniciar operação, até finalizar e retornar o resultado da operação ao software gerente.

Figura 5 - Diagrama de atividades Phantom: ação de instalação software



## 3.2 ESPECIFICAÇÃO

Nesta seção serão apresentados os principais requisitos funcionais e não funcionais, sua rastreabilidade com casos de uso e a entidade relacionamento do sistema (MER).

### 3.2.1 Requisitos funcionais

O Quadro 1 apresenta os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

Quadro 1 - Requisitos funcionais

Requisitos Funcionais	Caso de Uso
RF01: O sistema gerenciamento deverá permitir o usuário verificar PCs da rede	UC01
RF02: O sistema gerenciamento deverá permitir o usuário cadastrar PC	UC02

da rede	
RF03: O sistema gerenciamento deverá permitir o usuário executar uma ação (Instalar, Desinstalar, Identificar)	UC03, UC04, UC05, UC06.
RF04: O sistema gerenciamento deverá permitir o usuário manter ações realizadas	UC07
RF05: O sistema de apoio deverá permitir instalação autônoma do software	UC08
RF06: O sistema de apoio deverá permitir desinstalação autônoma do software	UC09
RF07: O sistema de apoio deverá permitir que o usuário identifique-se para sistema de gerenciamento	UC10

### 3.2.2 Requisitos não funcionais

O Quadro 2 lista os requisitos não funcionais previstos para o sistema. O sistema foi desenvolvido com linguagens, bibliotecas e ferramentas livres.

Quadro 2 - Requisitos não funcionais

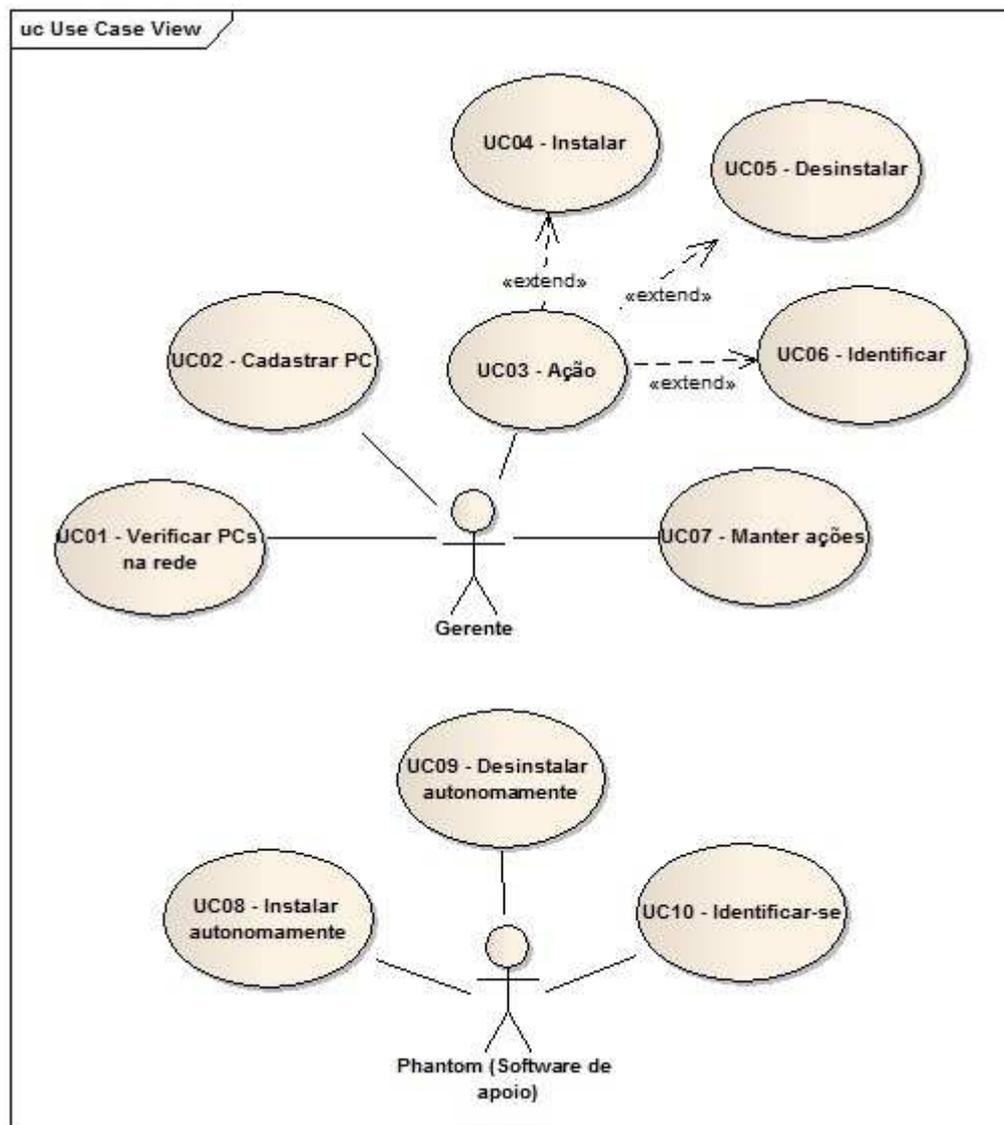
<b>Requisitos Não Funcionais</b>
RNF01: O sistema deverá utilizar banco de dados MySQL
RNF02: O sistema será implementado em Java 1.6
RNF03: O sistema deverá ser executado nos sistemas operacionais Windows XP, Vista, Seven
RNF04: O software de apoio Phantom, tem que ter privilégios de administrador da máquina
RNF05: O sistema deverá utilizar os protocolos TCP/IP, através de <i>socket</i> , para

comunicação entre o software de gerenciamento e o software de apoio
RNF06: O sistema web será desenvolvido com o framework JSF

### 3.2.3 Diagrama de casos de uso

A Figura 6 contempla o funcionamento geral do sistema bem como o papel de cada ator envolvido no processo através de um diagrama de casos de uso. O ator principal do sistema é o gerente, que gera as ações que repercutem nos usuários Phantom, que é o sistema de apoio para execução da ação. A descrição detalhada dos casos de uso está apresentada no Apêndice A.

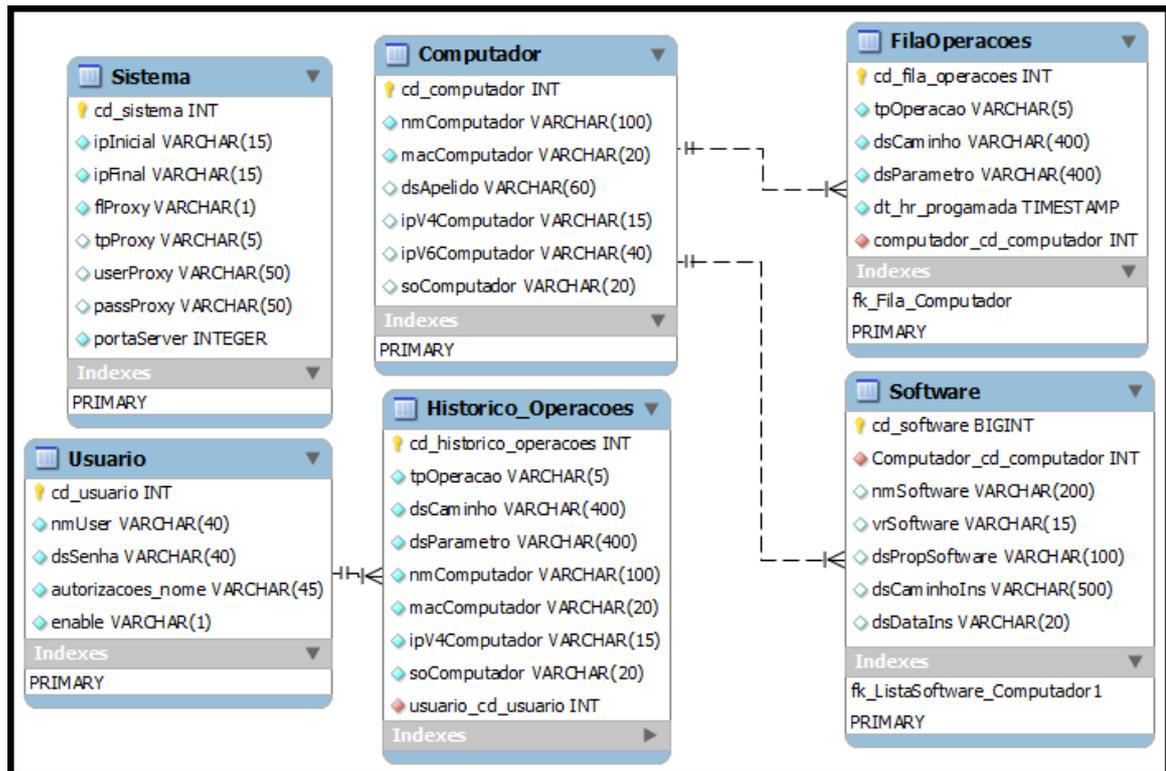
Figura 6 - Diagrama de casos de uso



### 3.2.4 Modelo de Entidade Relacionamento

A Figura 7 contempla o Modelo Entidade Relacionamento (MER) do sistema. O dicionário de dados está apresentado no Apêndice B.

Figura 7 - Modelo de entidade relacionamento



A tabela “sistema” contém as configurações gerais do software de gerenciamento, para fazê-lo funcionar. A tabela “usuario” contém as informações do usuário do software de gerenciamento, para poder ter acesso ao software. A tabela “histórico\_operações” contém as ações efetuadas pelo usuário em um determinado computador “Log”. A tabela “computador” contém as informações chaves dos computadores em rede, para o software de gerenciamento. A tabela “software” contém as informações dos softwares instalados nos computadores. A tabela “filaOperacoes” contém as informações das ações a serem realizadas nos computadores.

### 3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

#### 3.3.1 Técnicas e ferramentas utilizadas

O sistema foi desenvolvido utilizando a linguagem Java que é uma linguagem de código aberto (*Open Source*) sob os termos *GNU General Public License* (GPL). O seu código fonte é compilado para um *bytecode* que é executado por uma máquina virtual, em tempo de execução.

Foram utilizadas também bibliotecas, *frameworks* e API Java para facilitar o desenvolvimento do trabalho. Bibliotecas são conjuntos de código criados para facilitar a implementação de tarefas comuns como componentes de interface, gerador de relatório. *Framework* é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. O *framework* pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação. Ao contrário das bibliotecas, é o *framework* quem dita o fluxo de controle da aplicação, chamado de Inversão de Controle, utilizado no sistema para controle de sessão, segurança da aplicação e manipulação e gerenciamento de objetos. API (Application Programming Interface ou Interface de Programação de Aplicativos) é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por programas aplicativos que não querem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços, utilizado no sistema para camada de persistência de dados.

O servidor *web* utilizado é o Glassfish, lançado em 2006 pela Sun Microsystems. É um software livre, sendo duplamente licenciado sob duas licenças de software livre: *Common Development and Distribution License* (CDDL) e *GNU General Public License* (GPL).

Os dados do sistema são armazenados em um Sistema Gerenciador de Banco de Dados (SGBD) chamado MySQL, escolhido pela facilidade e simplicidade do uso do mesmo, e por ter licença de software livre *GNU General Public License* (GPL).

### 3.3.1.1 Java

Para o desenvolvimento do trabalho foram utilizadas duas plataformas Java, Java Platform Standard Edition (JSE) e Java Platform Enterprise Edition (JEE). Na Figura 8 pode ser observada a estrutura de projeto do software Phantom desenvolvido em JSE. Na Figura 9 pode ser observado a estrutura de projeto do software de gerenciamento, desenvolvido em JEE.

Java SE é uma ferramenta de desenvolvimento para a plataforma Java, que contém todo o ambiente necessário para a criação e execução de aplicações Java, incluindo a máquina virtual Java (JVM), o compilador Java, as APIs do Java.

Java EE é uma plataforma de programação para servidores na linguagem de programação Java. A plataforma fornece uma API e um ambiente de tempo de execução para o desenvolvimento e execução de softwares corporativos, incluindo serviços de rede e web, e outras aplicações de rede de larga escala, multicamadas, escaláveis, confiáveis e seguras. Java EE estende a Java Platform, Standard Edition (JSE), fornecendo uma API para mapeamento objeto-relacional, arquiteturas multicamada e distribuídas e web services.

Figura 8 - Software Phantom software de apoio ao Gerenciamento, desenvolvido na plataforma JSE

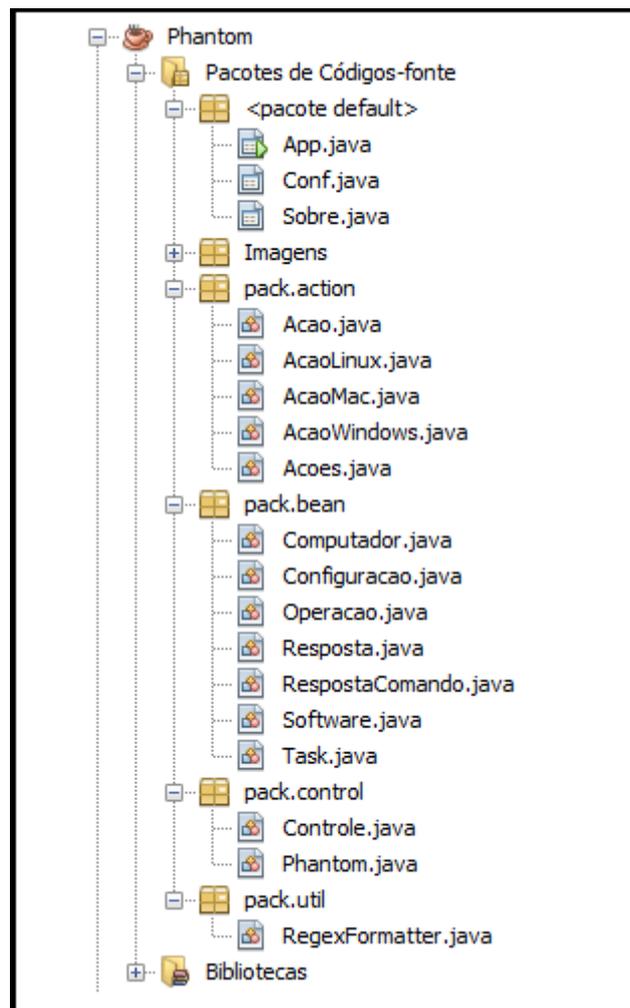
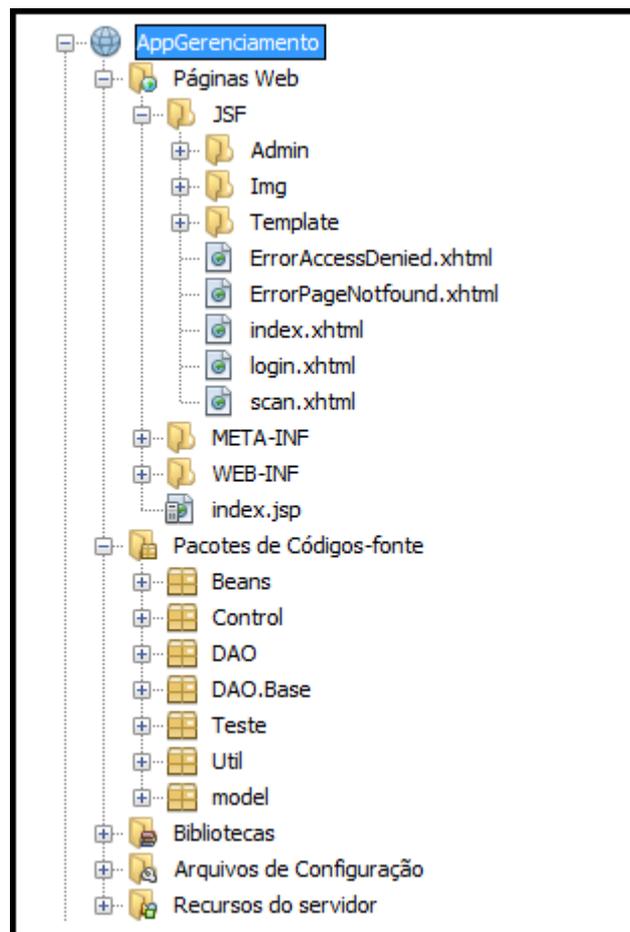


Figura 9 - Software de Gerenciamento de software, desenvolvido na plataforma JEE



### 3.3.1.2 Bibliotecas Java

Primefaces é uma biblioteca livre de código aberto distribuído com a licença Apache compatível com GLP, biblioteca de componentes para o *framework* Java Server Faces, desenvolvido pela Prime Teknoloji. Ele fornece um conjunto de componentes principalmente visuais. Estes podem ser usados por programadores de JSF, além do pequeno conjunto de componentes básicos que vêm com a plataforma central JSF para compor a interface para uma aplicação web (JEE). Na Figura 10 demonstra-se a utilização da biblioteca Primefaces diretamente no fonte do software de gerenciamento.

Figura 10 - Página inicial. Exemplo de utilização da biblioteca Primefaces

```

1 <ui:composition template="Template/geral.xhtml"
2     xmlns="http://www.w3.org/1999/xhtml"
3     xmlns:f="http://java.sun.com/jsf/core"
4     xmlns:h="http://java.sun.com/jsf/html"
5     xmlns:ui="http://java.sun.com/jsf/facelets"
6     xmlns:p="http://primefaces.org/ui"
7
8     xmlns:c="http://java.sun.com/jsp/jstl/core">
9
10 <ui:define name="content">
11     <h:body>
12         <div style="size: auto;width: 100%;height: 100%;">
13
14             <h1> Lista de Computadores conectados </h1>
15
16             <c:if test="#{index.listaComputadores.size()}>0" >
17
18                 <h:form id="form" >
19                     <p:dataTable id="tabela" var="computador" value="#{index.listaComputadores}">
20                         <p:column>
21                             <f:facet name="header">
22                                 <h:outputText value="Nome Pc"/>
23                             </f:facet>
24                             <p:commandLink update=":form:infComp" oncomplete="dlginf.show();" >
25                                 <h:outputText value="#{computador.nmComputador}" title="Mais informações" />
26                                 <f:setPropertyActionListener value="#{computador}" target="#{index.computador}" />
27                             </p:commandLink>
28                         </p:column>
29                         <p:column>
30                             <f:facet name="header">
31                                 <h:outputText value="IpV4"/>
32                             </f:facet>
33                             <h:outputText value="#{computador.ipV4computador}"/>
34                         </p:column>
35                     </p:dataTable>
36                 </h:form>
37             </c:if>
38         </div>
39     </h:body>
40 </ui:define>

```

### 3.3.1.3 Framework Java

O Spring *framework* fornece uma programação abrangente e modelo de configuração para aplicações empresariais modernas baseadas em Java - em qualquer tipo de plataforma de implementação. Um elemento-chave do Spring é o suporte de infraestrutura no nível do aplicativo: Spring enfoca o "encanamento" de aplicações empresariais para que as equipes possam se concentrar na lógica de negócios em nível de aplicativo, sem vínculos desnecessários para ambientes específicos de implantação (GOPIVOTAL, 2013). Na Figura 11 pode ser observado o arquivo XML de configuração do Spring *framework*, para definir as atribuições utilizadas do Spring.

Figura 11 - Arquivo de configuração XML do Spring *framework*

```

24
25
26 <b:bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean" >
27   <b:property name="persistenceUnitName" value="AppGerenciamentoPU" />
28 </b:bean>
29
30 <b:bean class="org.springframework.orm.jpa.JpaTransactionManager" id="transactionManager">
31   <b:property name="entityManagerFactory" ref="entityManagerFactory" />
32 </b:bean>
33
34 <tx:annotation-driven />
35 <context:component-scan base-package="*" />
36 <context:annotation-config />
37 <b:bean class="org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor" />
38 <b:bean class="org.springframework.dao.annotation.PersistenceExceptionTranslationPostProcessor"/>
39
40 <http auto-config="true" use-expressions="true">
41   <intercept-url pattern="/JSF/index.jsf" access="hasRole('ROLE_USER')"/>
42   <form-login login-page="/JSF/login.jsf" authentication-failure-url="/JSF/login.jsf?erro=true"/>
43 </http>
44
45 <authentication-manager>
46   <authentication-provider>
47     <jdbc-user-service data-source-ref="dataSource"
48       users-by-username-query="SELECT nmUser, dsSenha, enable FROM usuario WHERE nmUser=?"
49       authorities-by-username-query="SELECT nmUser as username, autorizacoes_nome as authority FROM usuario WHERE nmUser=?"
50     />
51   </authentication-provider>
52 </authentication-manager>
53
54 <b:bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource" >
55   <b:property name="url" value="jdbc:mysql://localhost:3306/tcc" />
56   <b:property name="driverClassName" value="com.mysql.jdbc.Driver" />
57   <b:property name="username" value="root" />
58   <b:property name="password" value="root" />
59 </b:bean>

```

O Hibernate é um *framework* para o mapeamento objeto relacional escrito na linguagem Java, que também é disponível no .Net com o nome NHibernate. Este *framework* facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (XML) ou anotações Java. Hibernate é um software livre de código aberto distribuído com a licença LGPL. O Hibernate é utilizado no sistema como base de objeto relacional, fazendo também a implementação da interface JPA utilizando métodos padronizados por especificação do Java.

### 3.3.1.4 API Java

*Java Persistence API* ou JPA, é uma API padrão do java para persistência que deve ser implementada por frameworks que queiram seguir o padrão. A JPA define um meio de mapeamento objeto relacional para objetos Java simples e comuns, denominados beans de entidade. Diversos *frameworks* de mapeamento objeto relacional como o Hibernate implementam a JPA. Também gerencia o desenvolvimento de entidades do Modelo

Relacional usando as plataformas nativa Java SE e Java EE. Nas Figuras 12 e 13 pode-se observar duas classes utilizando anotações e métodos providos da interface JPA, e utilizando a herança da orientação a objeto (OO), pode-se replicar as implementações feitas em uma classe nas demais classes que fazem acesso ao banco de dados.

Figura 12 - Classe DAO. Implementa métodos comuns de utilização dos comandos de DML entre as entidades

```

19 public abstract class DAOImpl<T, PK extends Serializable> extends JpaDaoSupport implements DAO<T, PK> {
20
21     @Autowired
22     public void setEntityManagerFactoryAuto(EntityManagerFactory emf) {
23         super.setEntityManagerFactory(emf);
24         super.getJpaTemplate().setEntityManagerFactory(emf);
25     }
26
27     @Transactional(rollbackFor = Exception.class)
28     @Override
29     public Object persist(T object) {
30         getJpaTemplate().persist(object);
31         return object;
32     }
33
34     @Transactional
35     @Override
36     public T merge(T object) {
37         return getJpaTemplate().merge(object);
38     }
39
40     @Transactional
41     @Override
42     public void remove(PK primaryKey) {
43         T excluir = find(primaryKey);
44         getJpaTemplate().remove(excluir);
45     }
46
47     @Transactional
48     @Override
49     public void removeObject(T objeto) {
50         getJpaTemplate().remove(objeto);
51     }
52
53 }

```

Figura 13 - Classe ComputadorDao. Herda a classe DAO e implementa seus métodos

```

15 @Repository
16 public class ComputadorDaoImpl extends DAOImpl<Computador, Integer> implements ComputadorDao {
17
18     @Override
19     public Class<Computador> getObjectClass() {
20         return Computador.class;
21     }
22 }
23

```

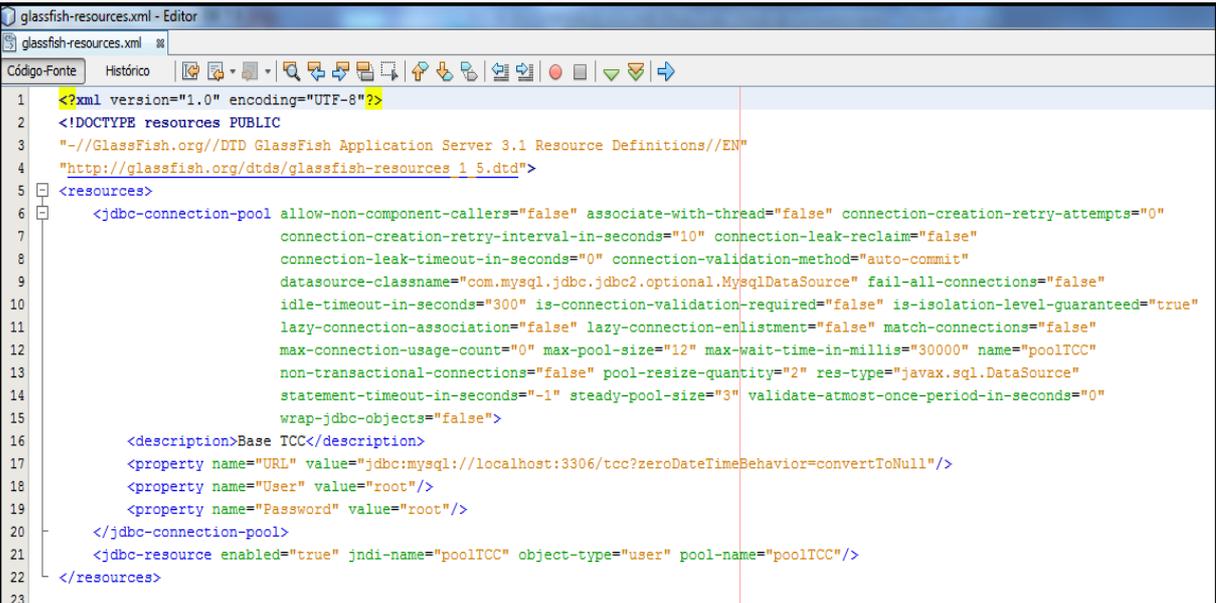
### 3.3.1.5 Banco de dados MySQL

O MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD) desenvolvido pela Oracle Corporation. Ele é utilizado para armazenar todos os dados do sistema. São utilizados comandos para que o sistema possa comunicar-se com o banco. Através destes comandos DML, o SGBD pode recuperar, gravar ou alterar registros do banco (ORACLE, 2013).

### 3.3.1.6 Servidor web Glassfish

GlassFish é um servidor de aplicação *web*, *open source* mantida pela Sun Microsystems para a plataforma Java EE. Sua versão proprietária é chamada Sun GlassFish Enterprise Server, é um software livre, sendo duplamente licenciado sob duas licenças de software livre: *Common Development and Distribution License (CDDL)* e *GNU General Public License (GPL)*. O servidor Glassfish é utilizado para implantar o sistema de gerenciamento e fazer o controle se fluxo de conexões com o banco de dados, conforme mostrado na Figura 14.

Figura 14 - Arquivo de configuração XML de *pool* de conexão com banco de dados



```

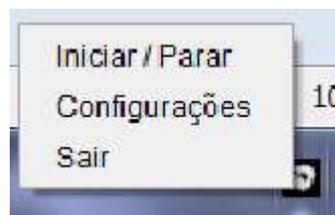
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE resources PUBLIC
3  "http://GlassFish.org/DTD GlassFish Application Server 3.1 Resource Definitions//EN"
4  "http://glassfish.org/dtds/glassfish-resources_1_5.dtd">
5  <resources>
6  <jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false" connection-creation-retry-attempts="0"
7  connection-creation-retry-interval-in-seconds="10" connection-leak-reclaim="false"
8  connection-leak-timeout-in-seconds="0" connection-validation-method="auto-commit"
9  datasource-classname="com.mysql.jdbc.jdbc2.optional.MysqlDataSource" fail-all-connections="false"
10 idle-timeout-in-seconds="300" is-connection-validation-required="false" is-isolation-level-guaranteed="true"
11 lazy-connection-association="false" lazy-connection-enlistment="false" match-connections="false"
12 max-connection-usage-count="0" max-pool-size="12" max-wait-time-in-millis="30000" name="poolTCC"
13 non-transactional-connections="false" pool-resize-quantity="2" res-type="javax.sql.DataSource"
14 statement-timeout-in-seconds="-1" steady-pool-size="3" validate-atmost-once-period-in-seconds="0"
15 wrap-jdbc-objects="false">
16   <description>Base TCC</description>
17   <property name="URL" value="jdbc:mysql://localhost:3306/tcc?zeroDateTimeBehavior=convertToNull"/>
18   <property name="User" value="root"/>
19   <property name="Password" value="root"/>
20 </jdbc-connection-pool>
21 <jdbc-resource enabled="true" jndi-name="poolTCC" object-type="user" pool-name="poolTCC"/>
22 </resources>
23

```

### 3.3.2 Operacionalidade da implementação

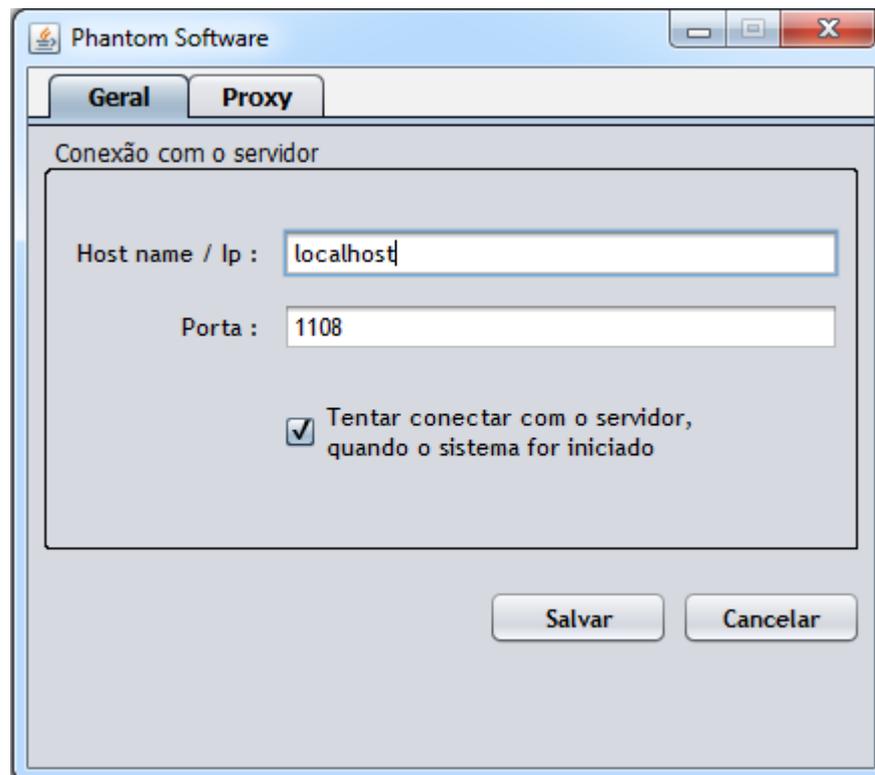
Para habilitar o sistema de gerenciamento de rede, deve-se antes instalar o sistema de apoio Phantom desenvolvido para executar ações administrativas, em *background* no computador. Após executar ele aparecerá na barra de tarefas do Windows, como ícone de tarefa. Na Figura 15 demonstra-se as opções de operacionalidade do Phantom que se resume a parar ou iniciar comunicação com o servidor, configurações gerais e sair do software ou fechá-lo.

Figura 15 – Opções de operacionalidade software phantom

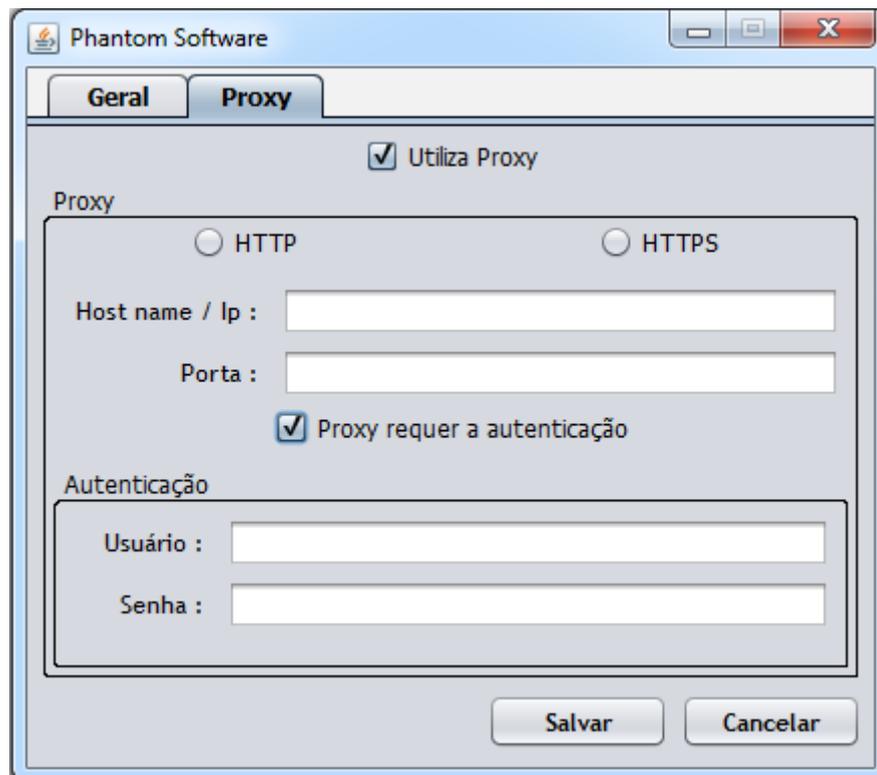


Na Figura 16 é apresentada a configuração geral no software Phantom. Na aba de configuração geral estão os dados necessários para comunicação com o software de gerenciamento, onde deve-se configurar o IP ou *Host name*, e porta de comunicação. Também há a opção de tentativa automática de conexão a partir dos dados cadastrados.

Figura 16 – Configuração geral no software Phantom



Na Figura 17 é apresentada a configuração de *proxy* no software Phantom. Na aba de configuração *proxy* tem-se os dados necessários para autenticação com o proxy, caso a rede assim o necessite, pois a instalação de software pode ocorrer tanto de um arquivo local de rede quanto uma chamada extranet (HTTP, FTP), necessitando da autenticação antes da chamada.

Figura 17 – Configuração de *proxy* no software phantom

As telas de interface do software Phantom são somente essas, que são configurações de operacionalidade, por isso não foi utilizado qualquer tipo de biblioteca, API, ou *Framework*, que não fosse nativo do Java para a construção do mesmo. Na Figura 18 mostra-se o código fonte de identificação, que é a autenticação do computador no software de gerenciamento.

Figura 18 – Código fonte da identificação do computador

```

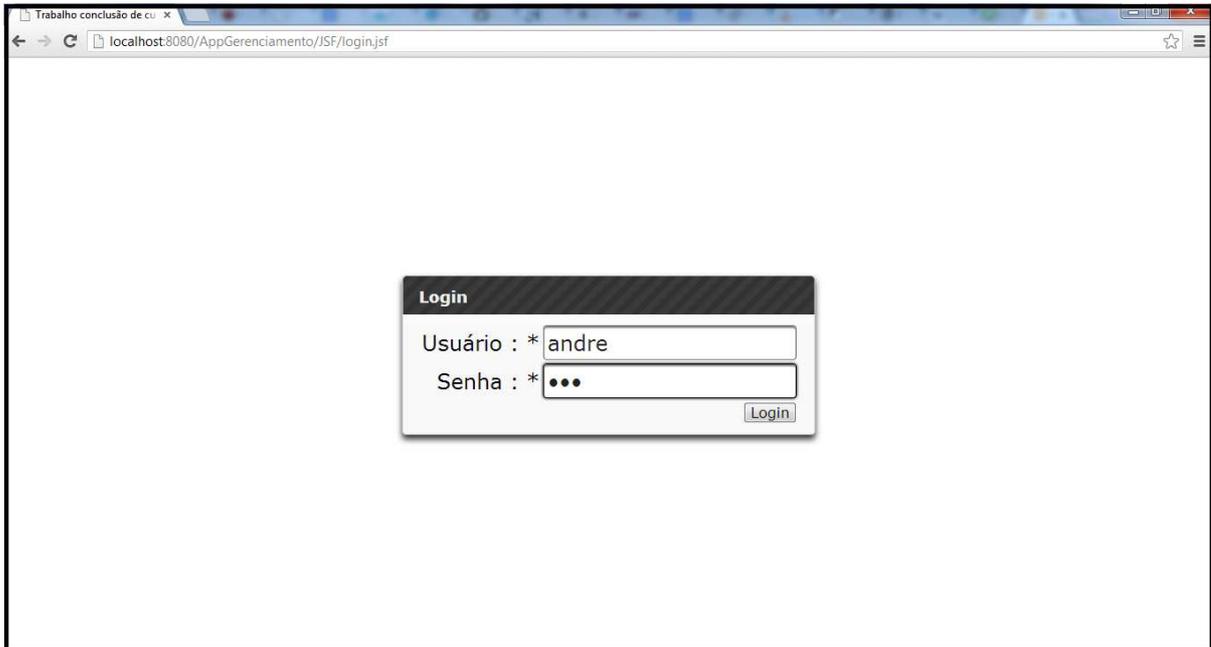
20 public class Acao implements Acoes {
21     protected ArrayList listaSoftware = new ArrayList();
22
23     @Override
24     public String sO() {
25         return System.getProperty("os.name");
26     }
27
28     @Override
29     @SuppressWarnings("CallToThreadDumpStack")
30     public Computador identifica(Operacao op) {
31         Computador comp = new Computador();
32         try {
33             InetAddress address = InetAddress.getLocalHost();
34             comp.setNmComputador(address.getHostName());
35             comp.setIpv4Computador(address.getHostAddress());
36             NetworkInterface ni = NetworkInterface.getByInetAddress(address);
37             byte[] mac = ni.getHardwareAddress();
38             for (int i = 0; i < mac.length; i++) {
39                 comp.setMacComputador(String.format("%02X%s", mac[i], (i < mac.length - 1) ? "-" : ""));
40             }
41             comp.setSoComputador(this.sO());
42         } catch (Exception e) {
43             e.printStackTrace();
44             comp = null;
45         }
46         return comp;
47     }

```

O software de gerenciamento do sistema é um software *web* que foi desenvolvido em JEE, com a tecnologia JSF para interface, JPA para persistência dos dados, Spring para gerenciamento de objetos e escalonamento do projeto. A aplicação é executada sobre o servidor Glassfish, onde também faz o controle de conexão com o banco de dados, e o banco de dados utilizado é o MySQL. O software de gerenciamento se localiza dentro da mesma rede onde os computadores.

Inicialmente o usuário faz o *login* no software de gerenciamento de rede, conforme a Figura 19, com usuário e senha passada pelo administrador do sistema.

Figura 19 – Tela de *login* software de gerenciamento



Após estar logado, o sistema apresenta a tela conforme mostra a Figura 20, onde o usuário visualiza os computadores conectados ao software de gerenciamento. Aqueles que já estão cadastrados apresentam um botão de “Edit”, senão há um botão de “Add” para cadastrar o computador no sistema.

Figura 20 – Tela inicial do software de gerenciamento. Lista de computadores conectados



Na coluna ao lado, há a opção “Softwares”, que lista os softwares instalados na máquina e dá a opção de desinstalação remota do software colocando na fila de operação, apresentado na Figura 21. Na outra coluna “Operações”, se visualizam as operações que estão na fila do computador selecionado, podendo ser uma instalação ou desinstalação de software, apresentado na Figura 22.

Figura 21 – Tela *pop up* da lista de operações a serem executadas

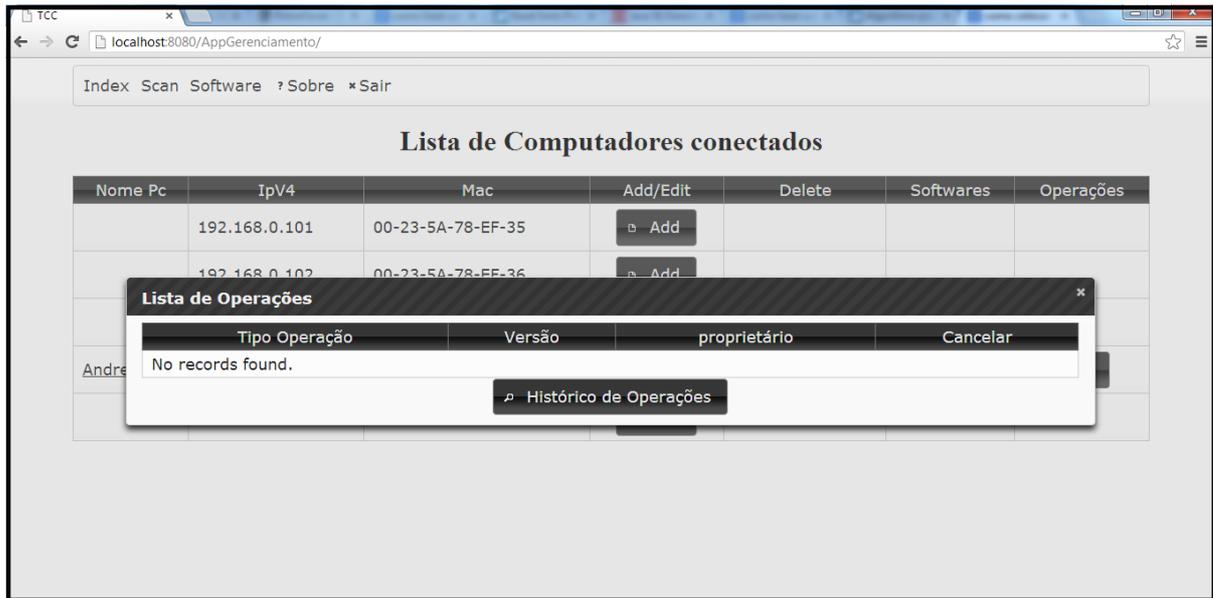
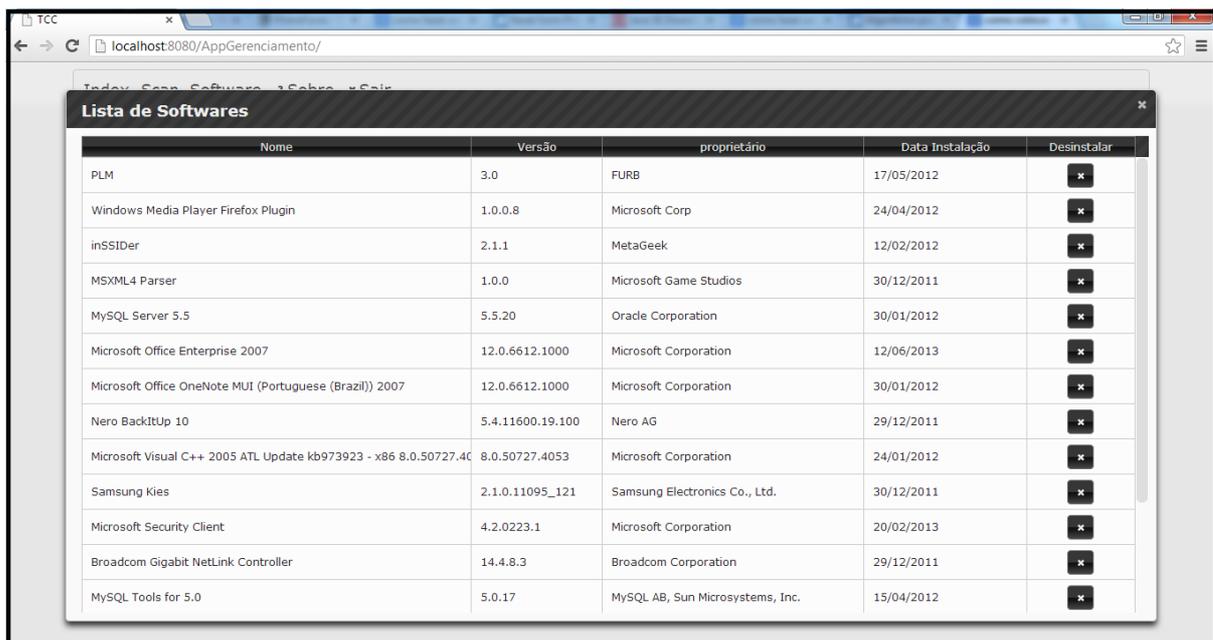


Figura 22 – Tela *pop up* da lista de softwares instalados no computador



Na coluna Nome PC pode-se obter mais detalhes do computador como apresentado na Figura 23.

Figura 23 – Tela *pop up* trazendo mais informações do computador



Na Figura 24 pode-se observar a classe de controle de protocolo. Nela são serializados objetos para serem comunicados entre o programa cliente "Phantom", e o programa servidor (o software de gerenciamento).

Figura 24 – Classe de controle de protocolo

```

38     InputStream in = this.phantom.getSocket().getInputStream();
39     objOut = new ObjectOutputStream(this.phantom.getSocket().getOutputStream());
40
41     while (this.phantom.isConectado() == true) {
42
43         if (this.phantom.getSocket().isClosed()) {
44             throw new Exception("finalizou");
45         }
46         if (in.available() > 0) {
47             Operacao op = (Operacao) new ObjectInputStream(in).readObject();
48             respostaCom = new RespostaComando(Resposta.ERRO, op);
49
50             switch (op.getTpOperacao().toUpperCase()) {
51                 case "IDENT":
52                     ArrayList lista = new ArrayList();
53                     lista.add(acao.identifica(op));
54                     respostaCom.setLista(lista);
55                     respostaCom.setResposta(Resposta.OK);
56                     break;
57                 case "INSTA":
58                     respostaCom.setResposta(acao.instalacaoSilent(op));
59                     break;
60                 case "DESIN":
61                     respostaCom.setResposta(acao.desinstalaSoftware(op));
62                     break;
63                 case "LSINS":
64                     respostaCom = acao.listaSoftwareInstalados(respostaCom);
65                     break;
66                 case "LSMEM":
67                     respostaCom = acao.listaSoftwareMemoria(respostaCom);
68                     break;
69                 default:
70                     respostaCom.setResposta(Resposta.SEMSUPORTE);
71             }
72         }

```

A tela de parametrização para instalação de software está apresentada na Figura 25. Nela pode-se parametrizar o caminho onde buscar o arquivo, também podendo colocar *links* de internet (HTTP/FTP) para o mesmo. Os parâmetros são para fazer a instalação do software em *background* ou instalação silenciosa, e há a opção de agendamento de dia e hora a ser feito a instalação. Após ser parametrizado, escolhem-se os computadores afetados com essa ação de instalação e se salvar para geração da ação e ser mandado para fila de operações e efetuar o “Log” de ação no sistema.

Figura 25 – Tela Ação de instalação de software

Index Scan Software ? Sobre \* Sair

Caminho : \* E:\jdk-7u25-windows-x64.exe

Parâmetro : \* /s

Data Hora : \* 27/06/2013 07:35

Cancelar Salvar

**Lista de Computadores conectados e cadastrados**

	Nome Pc	IpV4	Mac	Apelido
<input checked="" type="checkbox"/>	Andreb-PC	201.54.194.51	94-39-E5-F6-6C-5C	

Por fim a tela de “Scan” está apresentada na Figura 26. Com esta tela é possível parametrizar o IP inicial e IP final e *scanear* a rede local para encontrar os computadores que estão respondendo na rede, através do protocolo ICMP (o Ping) testando a faixa entre os IPs, para detectar possíveis problemas de não comunicação do computador.

Figura 26 – Tela de “Scan”

Index Scan Software ? Sobre \* Sair

Ip inicial : \* 192.168.0.100

Ip final : \* 192.168.0.150

Scan Limpar

**Ips Encontrados**

Ip
No records found.

### 3.4 RESULTADOS E DISCUSSÃO

A proposta inicial de um software para gerenciar a instalação de softwares por rede foi atendida. O resultado final deste projeto fornece uma base adequada para comunicação entre vários PCs sobre a rede, podendo ser gerenciado dentro de uma rede interna ou externa dependendo da necessidade, recomendando a utilização interna, pois não foi feita nenhuma segurança de comunicação.

Nos testes efetuados sobre as ações de instalação e desinstalação de software em rede, o software de gerenciamento mandando uma das ações citadas anteriormente, ele tem apenas 3 possíveis respostas de resultado de operação, são elas “OK” sinalizando sucesso na ação executada e removendo a ação da lista de operações, “ERRO” sinalizando um erro acontecido durante a ação executada, não removendo ação da lista de operações, fazendo o sistema mandar repetidas vezes a ação até obter sucesso ou ser excluída pelo administrador do sistema, e “SEMSUPORTE” sinaliza que ação não é suportada naquele sistema operacional, removendo a operação da lista.

Em relação aos trabalhos correlatos, o trabalho desenvolvido por Bambineti (2008), tem como objetivo somente extrair informações dos softwares instalados na máquina, através de *web services*, para fins de utilização de inventário, sendo que este software vai mais além de um inventário de estações, com ele é possível executar ações e que afetam as próprias estações.

Com relação ao software da Inquartos® NetworkManager (2012), o software tem mais funções administrativas de redes, como apresentado no item 2.6 do trabalho, mas não abrange a parte de desinstalação de software em computadores remotos e instalação somente de arquivos de instalação padronizados Windows (MSI).

## 4 CONCLUSÕES

O sistema produto deste trabalho atendeu aos objetivos definidos no item 1.1 deste documento. Os softwares desenvolvidos trabalhando conjuntamente conseguem desempenhar os papéis propostos inicialmente e fazer suas funções administrativas. Ademais mostrou-se bastante intuitivo e prático para ser utilizado.

Foi possível coletar os dados dos computadores na rede através de software Phantom instalado em cada computador, com uma configuração básica para o software de gerenciamento de rede.

As ferramentas utilizadas para o desenvolvimento do trabalho foram adequadas. As utilizações de bibliotecas, *frameworks* e API, se mostraram bastante eficientes para agilização das implementações. O SGBD MySQL mais do que atendeu as necessidades de recuperação e armazenamento de dados e o servidor *web* Glassfish se demonstrou adequado pelos seus recursos, como o suporte nativo de JSF 2.0 e pool de conexão com integração ao sistema através do JPA.

O maior desafio do trabalho foi em compreender o funcionamento do Windows para executar *scripts* de desinstalação de software e recuperação dos softwares instalados, o comportamento e normalização dos mesmos. E ainda desenvolver um protocolo de comunicação, passando objetos serializados para o mesmo, e juntar todas as tecnologias configurando os arquivos XML necessários, para formarem os dados a serem exibidos.

### 4.1 EXTENSÕES

Este sistema pode ser melhorado e ampliado ou incrementado em trabalhos futuros agregando novas funcionalidades ou utilizando outras técnicas e ferramentas para a execução de algumas tarefas, entre elas:

- a) compatibilizar as classes de comandos com outros sistemas operacionais (Linux e Mac OS);
- b) introduzir técnicas de criptografia e segurança na comunicação entre cliente e servidor;
- c) introduzir técnicas de elevação de usuário para instalação e desinstalação,

- tirando a obrigatoriedade do programa Phantom ser administrador da máquina;
- d) fazer instalador para o programa Phantom;
  - e) fazer um atualizador para o Sistema de gerenciamento e o programa Phantom;
  - f) mudar o sistema de protocolo utilizando RMI ou *web service*.

Estas melhorias tornariam o sistema mais próximo de se tornar um produto comercial.

## REFERÊNCIAS

- BAMBINETI, Charles. **Sistema de web services para inventário de estações em rede**. Trabalho de Conclusão de Curso - Universidade Regional de Blumenau, Curso de Ciências da Computação, Blumenau, 2008. Disponível em: <<http://campeche.inf.furb.br/tccs/2008-I/2008-1-07-vf-charlesbambineti.pdf>>. Acesso em: 14 abr. 2012.
- BARETTO, Luiz. **Taxa de Sobrevivência das Empresas no Brasil**, [S.l.], 2011. Disponível em: <[http://www.biblioteca.sebrae.com.br/bds/bds.nsf/45465B1C66A6772D832579300051816C/\\$File/NT00046582.pdf](http://www.biblioteca.sebrae.com.br/bds/bds.nsf/45465B1C66A6772D832579300051816C/$File/NT00046582.pdf)>. Acesso em: 11 abr. 2012.
- COMER, Douglas E. **Interligação em rede com TCP/IP**. Tradução de ARX Publicações. Rio de Janeiro: Campus, 1998.
- DELLA VALLE, James. **Faltam profissionais e sobram oportunidades em TI**. [S.l.], 2011. Disponível em: <<http://veja.abril.com.br/noticia/vida-digital/faltam-profissionais-e-sobram-oportunidades-em-ti>>. Acesso em: 12 abr. 2012.
- GOODMAN, Danny; MORRISON, Michael; NOVITSY, Paul; RAYL, Tia G. **JavaScript Bible**. 7th ed. Indianapolis: Wiley, 2010.
- GOPIVOTAL. **Spring Framework**. [S.l.], 2013. Disponível em: <<http://www.springsource.org/spring-framework>>. Acesso em: 16 jun. 2013.
- INQUARTOS® NETWORKMANAGER. **Sistema de Gerenciamento de rede**. [S.l.], 2012. Disponível em: <<http://www.inquartos.com>>. Acesso em: 14 abr. 2012.
- KUROSE, James F; ROSS, Keith W. **Redes de computadores e a Internet: uma abordagem top-down**. 3. ed. São Paulo: Pearson Addison Wesley, 2005.
- LOPES, Raquel V.; SAUVÉ, Jacques P.; NICOLLETTI, Pedro S. **Melhores práticas para gerência de redes de computadores**. Rio de Janeiro: Campus, 2003.
- LUCKOW, Décio H.; MELO, Alexandre A. **Programação Java para a web**. São Paulo: Novatec, 2010.
- ORACLE. **MySQL Data Sheet**. [S.l.], 2013. Disponível em: <<http://www.oracle.com/us/products/mysql/mysql-enterprise-ds-067312.pdf>>. Acessado em: 17 jun. 2013.
- REZENDE, Denis A. **Planejamento de sistemas de informação e informática: guia prático para planejar a tecnologia da informação integrada ao planejamento estratégico das organizações**. São Paulo: Atlas, 2003.

SOARES, Luis F. G.; LEMOS, Guido; COLCHER, Sergio. **Redes de computadores:** das LANs MANs e WANs as redes ATM. Rio de Janeiro: Campus, 1995.

## APÊNDICE A – Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos casos de uso conforme previstos no(s) diagrama(s) apresentado(s) na subseção 3.3.1.

### **UC01 Verificar PCs na rede**

Permite fazer uma verificação na rede em busca de PCs, para visualização na rede como um todo através do protocolo ICMP com o comando ping, destacando as máquinas gerenciáveis.

#### **Cenário**

##### **Verificar rede {Principal}.**

1. O gerente informa ip inicial e final para efetuar a busca, na rede;
2. Sistema valida as informações digitadas pelo gerente;
3. Gerente clica no botão “Scan”;
4. Sistema grava as informações;
5. Sistema gera log automático da ação realizada;
6. Sistema executa uma busca na rede;
7. Sistema exhibe os resultados.

##### **Serviço não localizado {Exceção}**

No passo 2, caso as informações estejam incoerentes, apresenta mensagem “As faixas de ip estão incoerentes. Informe-as novamente !”

### **UC02 Cadastrar PC**

Permite ao gerente visualizar, incluir, alterar ou excluir os PCs cadastrados.

### **UC03 Ação**

Permite ao gerente executar uma ação em um ou vários computadores sendo ação, (instalação, desinstalação e identificação).

#### **Cenário**

##### **Cadastrar ação {Principal}.**

1. O gerente escolhe entre uma das ações possíveis (instalação, desinstalação e identificação);
2. Preenche dados solicitados para a ação;
3. Sistema valida as informações digitadas pelo gerente;
4. Gerente clica no botão “Executar”;
5. Sistema grava as informações;
6. Sistema gera log automático da ação realizada;
7. Sistema fica esperando retorno da rotina automática, para finalização da ação;

### **UC04 Instalar**

Permite ao gerente executar uma ação instalação remota.

### **UC05 Desinstalar**

Permite ao gerente executar uma ação desinstalação remota.

### **UC06 Identificar**

Permite ao gerente executar uma ação identificação remota.

### **UC07 Manter ações**

Permite ao gerente visualizar, e emitir relatórios inventário e de uma listagem completa contendo todas as ações efetuadas, instalação e desinstalação.

#### **Cenários**

##### **Consulta ações {Principal}.**

1. Gerente consulta uma ação
2. Gerente opta por localizar uma ação
3. Gerente seleciona a opção de pesquisa (computador e, ou a data da ação) e informa os dados para pesquisa.
4. Sistema apresenta o(s) dado(s) da(s) ação(ões) relacionados com a consulta.

##### **Emitir relatório de inventário {Alternativo}.**

No passo 1, o gerente opta por emitir um relatório de inventário

- 1.1. Gerente informa os computadores, que deverão estar no relatório.
- 1.2. Sistema valida as informações.
- 1.3. Sistema gera o relatório.
- 1.4. Sistema gera log automático da ação realizada.

##### **Emitir relatório listagem completa de ações {Alternativo}.**

No passo 1, o gerente opta por emitir um relatório listagem completa de ações

- 1.1. Gerente informa os computadores e a data das ações, que deverão estar no relatório.
- 1.2. Sistema valida as informações.
- 1.3. Sistema gera o relatório.
- 1.4. Sistema gera log automático da ação realizada.

### **UC08 Instalar autonomamente**

Permite que o software de gerenciamento efetue a ação de instalação no computador (PC) especificado pelo gerente.

### **UC09 Desinstalar autonomamente**

Permite que o software de gerenciamento efetue a ação de desinstalação no computador (PC) especificado pelo gerente.

### **UC10 Identifica-se**

Permite que o software de gerenciamento efetue a ação de identificação do computador (PC) remoto, podendo ser efetuado a ação, pelo gerente, ou pela rotina automática do software de gerenciamento.

## APÊNDICE B – Dicionário de dados

Este Apêndice apresenta o dicionário de dados das tabelas do sistema e fornece um breve comentário dos seus campos. Os tipos de dados de cada campo são descritos a seguir:

- a) Varchar: armazena caracteres alfanuméricos até 255 caracteres;
- b) Int: armazena numéricos inteiros de 32 bits;
- c) Bigint: armazena numéricos inteiros de 64 bits;
- d) Timestamp: armazena data e hora.

No Quadro 3 tem-se o dicionário de dados da tabela “sistema”.

Quadro 3 – Dicionário de dados da tabela “sistema”

<b>Entidade:</b> Sistema		
<b>Descrição:</b> Configurações gerais do sistema		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_sistema	INT	Chave primária
ipInicial	VARCHAR(15)	
ipFinal	VARCHAR(15)	
flProxy	VARCHAR(1)	Status pra uso de proxy
tpProxy	VARCHAR(5)	Tipo de proxy
userProxy	VARCHAR(50)	Usuário proxy
passProxy	VARCHAR(50)	Senha proxy

No Quadro 4 tem-se o dicionário de dados da tabela “usuario”.

Quadro 4 – Dicionário de dados da tabela “usuario”

<b>Entidade:</b> Usuario		
<b>Descrição:</b> Armazena usuários do sistema		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_usuario	INT	Chave primária
nmUser	VARCHAR(40)	Nome do usuário
dsSenha	VARCHAR(40)	Senha do usuário
autorizacoes_nome	VARCHAR(45)	Tipo de autorização do usuário
enable	VARCHAR(1)	Status usuário (0,1) desativo ou ativo

No Quadro 5 tem-se o dicionário de dados da tabela “historico\_operacoes”.

Quadro 5 – Dicionário de dados da tabela “historico\_operacoes”

<b>Entidade:</b> Historico_operacoes		
<b>Descrição:</b> Armazena histórico de operações realizadas pelo usuário		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_historico_operacoes	INT	Chave primária
tpOperacao	VARCHAR(5)	Tipo de Operação
dsCaminho	VARCHAR(400)	Caminho a ser instalado
dsParametro	VARCHAR(400)	Parâmetros da operação
nmComputador	VARCHAR(100)	Nome do computador
macComputador	VARCHAR(20)	Mac do computador
ipV4Computador	VARCHAR(15)	IPV4 do computador
soComputador	VARCHAR(20)	Sistema operacional do computador
usuario_cd_usuario	INT	Chave estrangeira, entidade usuário

No Quadro 6 tem-se o dicionário de dados da tabela “computador”.

Quadro 6 – Dicionário de dados da tabela “computador”

<b>Entidade:</b> Computador		
<b>Descrição:</b> Informações pertinentes aos computadores conectados na rede		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_computador	INT	Chave primária
nmComputador	VARCHAR(100)	Nome do computador
macComputador	VARCHAR(20)	Mac do computador
dsApelido	VARCHAR(60)	Apelido para identificação do computador
ipV4Computador	VARCHAR(15)	IPV4 do computador
ipV6Computador	VARCHAR(40)	IPV6 do computador
soComputador	VARCHAR(20)	Sistema operacional do computador

No Quadro 7 tem-se o dicionário de dados da tabela “fila\_operacoes”.

Quadro 7 – Dicionário de dados da tabela “fila\_operacoes”

<b>Entidade:</b> Fila_operacoes		
<b>Descrição:</b> Armazena as operações a serem realizadas no computador		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>

cd_fila_operacoes	INT	Chave primária
tpOperacao	VARCHAR(5)	Tipo de Operação
dsCaminho	VARCHAR(400)	Caminho a ser instalado
dsParametro	VARCHAR(400)	Parâmetros da operação
dt_hr_progamada	TIMESTAMP	Agendamento operação
computador_cd_computador	INT	Chave estrangeira, entidade computador

No Quadro 8 tem-se o dicionário de dados da tabela “software”.

Quadro 8 – Dicionário de dados da tabela “software”

<b>Entidade:</b> Software		
<b>Descrição:</b> Armazena os softwares instalados no computador		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_software	BIGINT	Chave Primária
nmSoftware	VARCHAR(200)	Nome do software
vrSoftware	VARCHAR(15)	Versão do software
dsPropSoftware	VARCHAR(100)	Proprietário do software
dsCaminhoIns	VARCHAR(500)	Caminho do software instalado
dsDataIns	VARCHAR(20)	Data de instalação software
Computador_cd_computador	INT	Chave estrangeira, entidade computador