

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO
DO RIO GRANDE DO NORTE
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**ANÁLISE DE UM PROCESSO DE SOFTWARE
APLICADA A UM GERENTE SNMP PELA WEB**

Raphael Juliano de Medeiros

**ANÁLISE DE UM PROCESSO DE SOFTWARE
APLICADA A UM GERENTE SNMP PELA WEB**

Orientador:

Almirão Ferreira da Rocha Neto, M.Sc.

NATAL/RN

2006

ANÁLISE DE UM PROCESSO DE SOFTWARE APLICADA A UM GERENTE SNMP PELA WEB

Aluizio Ferreira da Rocha Neto, M.Sc.

2006

RB. 29.304

FARN
BIBLIOTECA
Nº Reg. 1000644
28/05/07

004.7
M438a
2006
Ex. 01

Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Medeiros, Raphael Juliano

Análise de um processo de software aplicada a um gerente SNMP pela Web /
Raphael Juliano Medeiros. – Natal: FARN, 2006.
64f.

Orientador: Aluizio Ferreira da Rocha Neto (M.Sc.)

Monografia (Bacharelado em Sistemas de Informação) – Faculdade Natalense
para o Desenvolvimento do Rio Grande do Norte.

1. Redes de Computadores – Monografia. 2. Gerência de Redes – Monografia
3. Desenvolvimento de Softwares – Monografia I. Título.

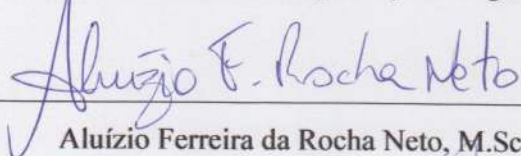
RN/FARN/BC

CDU 004.7

RAPHAEL JULIANO DE MEDEIROS

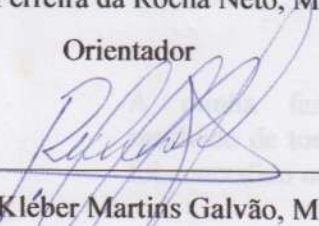
**ANÁLISE DE UM PROCESSO DE SOFTWARE
APLICADA A UM GERENTE SNMP PELA WEB**

Monografia apresentada ao curso de Bacharelado em Sistemas de Informação da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte, aprovada em 19 de dezembro de 2006, pela banca examinadora composta pelos seguintes membros:

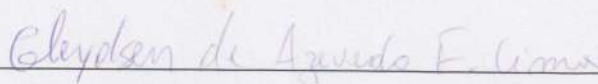


Aluizio Ferreira da Rocha Neto, M.Sc.

Orientador



Ricardo Kléber Martins Galvão, M.Sc.
Membro Examinador



Gleydson A. Ferreira de Lima, Esp.
Membro Examinador

RESUMO

Este trabalho apresenta uma análise de um processo de software aplicada a um gerente SNMP pela web (baseado em browser), o qual, se baseará em consultas utilizando funções específicas da extensão SNMP do PHP para obtenção de informações sobre os agentes que utilizam o protocolo SNMP e suas MIBs. A solução escalada permite uma utilização mais simples e estável da interface de consultas a agentes, independentemente de plataforma e baseada em componentes de software, descreve também um modelo de processo para o projeto que irá tornar o desenvolvimento de mesmo mais rápido.

Palavras-chave: Redes de Computadores, Ciência de Redes, SNMP, MIB, Software.

A minha família dedico este trabalho,
resultado de todo o apoio e dedicação que me
foi concedido nessa jornada.

RESUMO

Este trabalho apresenta uma análise de um processo de software aplicada a um gerente SNMP pela web (baseada em internet), o qual, se baseará em consultas utilizando funções específicas da extensão SNMP do PHP para obtenção de informações sobre os agentes que utilizam o protocolo SNMP e suas MIBs. A solução analisada permite uma utilização mais simples e estável da interface de consultas a agentes, independente de plataforma e baseada em componentes de software, descreve também um modelo de processo para o projeto que irá tornar o desenvolvimento do mesmo mais rápido.

Keywords: Computer Networks, Network Management, SNMP, MIB, Software.

Palavras-chave: Redes de Computadores, Gerência de Redes, SNMP, MIB, Software.

LISTA DE ILUSTRAÇÕES

ABSTRACT

This work presents an software process analysis applied to a SNMP manager for web (Internet based), which, will be based on consultations using specific functions of extension SNMP of the PHP for information attainment of the agents who use protocol SNMP and its MIBs. The analyzed solution allows a simpler use and steady of the interface of consultations the agents, platform independent and software components based, also describe a model of process for the project that will go to become the development of the same faster.

Keywords: Computer Networks. Networks Management. SNMP. MIB. Software.

Ilustração 5: Estrutura geral do módulo do framework.....	35
Ilustração 6: Manutenção de usuário do sistema.....	38
Ilustração 7: Listando os usuários do sistema.....	39
Ilustração 8: Manutenção de usuário do sistema.....	39
Ilustração 9: Página principal de acesso para usuários não root do sistema.....	39
Ilustração 10: Listando os agentes do sistema.....	40
Ilustração 11: Manutenção de agentes do sistema.....	40
Ilustração 12: Listando plataformas do sistema.....	41
Ilustração 13: Manutenção de plataformas do sistema.....	41
Ilustração 14: Listando MIBs do sistema.....	41
Ilustração 15: Manutenção de MIB do sistema.....	42
Ilustração 16: Listando objetos MIB do sistema.....	43
Ilustração 17: Manutenção de objetos MIB do sistema.....	43
Ilustração 18: Visualizando histórico de consultas do sistema.....	43
Ilustração 19: Consultando um agente cadastrado no sistema.....	44
Ilustração 20: Diagrama de casos de uso.....	45
Ilustração 21: Modelo físico do banco de dados.....	52

LISTA DE ILUSTRAÇÕES

Ilustração 1: Elementos de uma arquitetura geral de soluções de gerência.....	17
Ilustração 2: Primitivas básicas de comunicação do protocolo SNMP.....	19
Ilustração 3: Parte da árvore conceitual de atribuição de nomes a variáveis da MIB.....	21
Ilustração 4: Processo de desenvolvimento.....	32
Ilustração 5: Estrutura padrão de pastas do framework.....	35
Ilustração 6: Acesso ao sistema.....	38
Ilustração 7: Listando os usuários do sistema.....	39
Ilustração 8: Manutenção de usuário do sistema.....	39
Ilustração 9: Página principal de acesso para usuários não root do sistema.....	39
Ilustração 10: Listando os agentes do sistema.....	40
Ilustração 11: Manutenção de agentes do sistema.....	40
Ilustração 12: Listando plataformas do sistema.....	41
Ilustração 13: Manutenção de plataforma do sistema.....	41
Ilustração 14: Listando MIBs do sistema.....	41
Ilustração 15: Manutenção de MIB do sistema.....	42
Ilustração 16: Listando objetos MIB do sistema.....	42
Ilustração 17: Manutenção de objeto MIB do sistema.....	43
Ilustração 18: Visualizando histórico de consultas do sistema.....	43
Ilustração 19: Consultando um agente cadastrado no sistema.....	44
Ilustração 20: Diagrama de casos de uso.....	45
Ilustração 21: Modelo físico do banco de dados.....	57
Tabela 22: Fluxo listar agentes.....	54
Tabela 23: Fluxo para cadastrar agente.....	55
Tabela 24: Fluxo para editar agente.....	55
Tabela 25: Fluxo para excluir agente.....	56
Tabela 26: Fluxo para consultar agente.....	56
Tabela 27: Dicionário de dados.....	59
Tabela 28: Dicionário de dados (continuação).....	60

LISTA DE TABELAS E SÍMBOLOS

Tabela 1: Cronograma de atividades.....	26
Tabela 2: Tipos de artefatos.....	29
Tabela 3: Locais de armazenamento do projeto.....	30
Tabela 4: Descrição da estrutura padrão de pastas do framework.....	36
Tabela 5: Fluxo para efetuar login.....	46
Tabela 6: Fluxo para listar usuários.....	46
Tabela 7: Fluxo para cadastrar usuário.....	47
Tabela 8: Fluxo para editar usuário.....	47
Tabela 9: Fluxo para excluir usuário.....	48
Tabela 10: Fluxo para listar plataformas.....	48
Tabela 11: Fluxo para cadastrar plataforma.....	49
Tabela 12: Fluxo para editar plataforma.....	49
Tabela 13: Fluxo para excluir plataforma.....	50
Tabela 14: Fluxo para listar mibs.....	50
Tabela 15: Fluxo para cadastrar mib.....	51
Tabela 16: Fluxo para editar mib.....	51
Tabela 17: Fluxo para excluir mib.....	52
Tabela 18: Fluxo para listar objetos mib.....	52
Tabela 19: Fluxo para cadastrar objeto mib.....	53
Tabela 20: Fluxo para editar objeto mib.....	53
Tabela 21: Fluxo para excluir objeto mib.....	54
Tabela 22: Fluxo listar agentes.....	54
Tabela 23: Fluxo para cadastrar agente.....	55
Tabela 24: Fluxo para editar agente.....	55
Tabela 25: Fluxo para excluir agente.....	56
Tabela 26: Fluxo para consultar agente.....	56
Tabela 27: Dicionário de dados.....	59
Tabela 28: Dicionário de dados (continuação).....	60

LISTA DE ABREVIATURAS, SIGLAS E SÍMBOLOS

ASN.1	<i>Abstract Syntax Notation One</i> – Notação de Sintaxe Abstrata Um
CGI	<i>Common Gateway Interface</i> – Interface de Gateway Comum
CVS	<i>Concurrent Version System</i> – Sistema de Versões Concorrentes
DHTML	<i>Dynamic HTML</i> – HTML Dinâmico
GUI	<i>Graphical User Interface</i> – Interface Gráfica de Usuário
HTML	<i>Hyper Text Markup Language</i> – Linguagem de marcação de Hiper Texto
ISO	<i>International Standards Organization</i> – Organização Internacional de Normas
MIB	<i>Management Information Base</i> – Base de Informações de Gerenciamento
OID	<i>Object Identifier</i> – Identificador de Objeto
OSI	<i>Open Systems Interconnection</i> – Interconexão de Sistemas Abertos
PHP	<i>Hypertext Preprocessor</i> – Pré-processador de Hiper Texto
SNMP	<i>Simple Network Management Protocol</i> – Protocolo de Gestão Simples de Rede
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> – Protocolo de Controle de Transmissão/Protocolo da Internet

SUMÁRIO

1 INTRODUÇÃO	14
1.1 ORGANIZAÇÃO DO TRABALHO	15
2 REFERENCIAL TEÓRICO	16
2.1 REDE DE COMPUTADORES	16
2.1.1 Gerenciamento de Redes	16
2.1.2 Gerente de Rede	17
2.1.3 SNMP	18
2.1.3.1 Agentes	20
2.1.3.2 MIB	20
2.2 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	21
2.2.1 Framework	22
2.2.2 PHP	22
2.2.3 Recuperação após Desastre (Backups)	23
3 METODOLOGIA	24
3.1 TECNOLÓGICA	24
3.2 EXPLICATIVA	24
3.3 ESTUDO DE CASO	24
4 DESENVOLVIMENTO	25
4.1 PLANO DE GERENCIAMENTO	25
4.1.1 Projeto	25
4.1.1.1 Estrutura analítica	25
4.1.1.2 Cronograma	26
4.1.2 Comunicação	27
4.1.2.1 Acompanhamento das tarefas	27
4.1.2.2 Reuniões	27
4.1.2.3 Discussões técnicas	27
4.1.2.4 Lista de e-mails	27
4.1.2.5 Artefatos	27
4.1.3 Requisitos	28
4.2 PLANO DE GERÊNCIA DE CONFIGURAÇÃO	28
4.2.1 Atividades	28

4.2.1.1 Identificação.....	28
4.2.1.1.1 <i>identificador de documento</i>	28
4.2.1.1.2 <i>número de versão</i>	29
4.2.1.1.3 <i>local de armazenamento</i>	29
4.2.1.2 Controle de configuração.....	30
4.2.1.3 Auditoria de configuração.....	30
4.2.2 Plano de Contingência	30
4.2.3 Ferramentas	30
4.2.3.1 Software de Controle de Problemas (Bugzilla).....	31
4.2.3.2 Mensagens Instantânea (Windows Live Messenger).....	31
4.2.3.3 Fóruns de Discussão (phpBB).....	31
4.2.3.4 Sistema de Controle de Versões (WinCVS, TortoiseCVS).....	31
4.3 PROCESSO E METODOLOGIA DE DESENVOLVIMENTO	32
4.3.1 Processo	32
4.3.1.1 Fases.....	32
4.3.1.1.1 <i>inicialização</i>	32
4.3.1.1.2 <i>execução</i>	33
4.3.1.1.3 <i>transição</i>	33
4.3.2 Metodologia	33
4.3.2.1 Fusebox.....	33
4.3.2.1.1 <i>core (núcleo)</i>	34
4.3.2.1.2 <i>bibliotecas</i>	34
4.3.2.1.3 <i>interfaces com o usuário</i>	34
4.3.2.1.4 <i>validações</i>	35
4.3.2.1.5 <i>ações</i>	35
4.3.2.1.6 <i>pastas padrão</i>	35
4.4 REQUISITOS	36
4.4.1 Visão Geral do Produto	36
4.4.1.1 Descrição dos usuários.....	37
4.4.1.2 Parâmetros do sistema.....	37
4.4.1.2.1 <i>parâmetros alteráveis</i>	37
4.4.1.2.2 <i>parâmetros não-alteráveis</i>	37

4.4.1.2.3 <i>processos evolutivos</i>	37
4.4.2 Premissas e Restrições	37
4.4.3 Requisitos Funcionais	37
4.4.4 Requisitos Não-funcionais	38
4.5 ARQUITETURA	38
4.5.1 Modelo Navegacional	38
4.5.2 Modelagem dos Casos de Uso	44
4.5.2.1 Diagrama.....	45
4.5.2.2 Atores.....	45
4.5.2.3 Casos de uso.....	46
4.5.2.3.1 <i>efetuar login</i>	46
4.5.2.3.2 <i>listar usuários</i>	46
4.5.2.3.3 <i>cadastrar usuário</i>	47
4.5.2.3.4 <i>editar usuário</i>	47
4.5.2.3.5 <i>excluir usuário</i>	48
4.5.2.3.6 <i>listar plataformas</i>	48
4.5.2.3.7 <i>cadastrar plataforma</i>	49
4.5.2.3.8 <i>editar plataforma</i>	49
4.5.2.3.9 <i>excluir plataforma</i>	50
4.5.2.3.10 <i>listar mibs</i>	50
4.5.2.3.11 <i>cadastrar mib</i>	51
4.5.2.3.12 <i>editar mib</i>	51
4.5.2.3.13 <i>excluir mib</i>	52
4.5.2.3.14 <i>listar objetos mib</i>	52
4.5.2.3.15 <i>cadastrar objeto mib</i>	53
4.5.2.3.16 <i>editar objeto mib</i>	53
4.5.2.3.17 <i>excluir objeto mib</i>	54
4.5.2.3.18 <i>listar agentes</i>	54
4.5.2.3.19 <i>cadastrar agente</i>	55
4.5.2.3.20 <i>editar agente</i>	55
4.5.2.3.21 <i>excluir agente</i>	56
4.5.2.3.22 <i>consultar agente</i>	56

4.5.3 Armazenamento.....	57
4.5.3.1 Diagrama físico.....	57
4.5.3.1.1 relacionamentos.....	58
4.5.3.1.2 estrutura interna	58
4.5.3.2 Dicionário de dados.....	58
4.6 PLANO DE TESTES.....	60
4.6.1 Alvos do Teste.....	61
4.6.2 Estratégia de Testes.....	61
4.6.2.1 Testes de unidade.....	61
4.6.2.2 Testes de integração.....	61
4.6.2.3 Testes de sistema.....	61
4.6.2.3.1 testes de aceitação do sistema.....	62
4.6.3 Diretrizes para Execução dos Testes.....	62
4.6.4 Critérios de Aceitação dos Testes.....	62
4.6.5 Documentação de Defeitos.....	62
5 CONCLUSÃO.....	63
REFERÊNCIAS	64

A necessidade de mecanismos de gerenciamento nas redes baseadas em TCP/IP é atendida pelo padrão Internet (Internet Standard Network Management Framework), que define o SNMP (Simple Network Management Protocol) em associação com o modelo de informação MIB (Management Information Base). Na Internet, o SNMP alcançou grande sucesso por ter sido o primeiro protocolo de gerenciamento não proprietário, público, fácil de ser implementado, possibilitando o gerenciamento efetivo de ambientes heterogêneos. Surgiram no mercado vários tipos de aplicações baseadas no padrão SNMP, quase sempre dependentes de alguma plataforma de gerência. Aplicações como o Nagios, as quais permitem o gerenciamento unificado de agências através da Internet, a medida em que crescem seus recursos e funcionalidades, vão perdendo sua simplicidade e robustez, tornando em alguns casos inviável sua utilização.

Hoje, a tecnologia empregada no Web está sendo usada na solução deste problema. Esta tecnologia não está sendo aplicada apenas à gerência de redes, mas também em interfaces como forma de acessar bancos de dados, bases de conhecimentos, aplicações de escritórios, etc. O navegador vem se firmando como cliente universal, através do qual usuários acessam

1 INTRODUÇÃO

No início, as redes de computadores foram concebidas simplesmente como meio de compartilhar dispositivos periféricos caros como impressoras, discos e modems de alta velocidade. Porém, as redes corporativas foram ficando cada vez maiores e integradas às organizações, com muitas interligações com outras redes, a exemplo da Internet. Passaram então a fazer parte do cotidiano dos usuários como uma ferramenta que oferece recursos e serviços, permitindo interação e aumento da produtividade, restringindo o compartilhamento dos dispositivos a um aspecto secundário.

Num momento em que as redes se tornam cada vez mais importantes para as empresas, deixando de ser infra-estrutura de comunicação dispensável para se tornarem ferramentas cruciais ao desenvolvimento empresarial, tornando necessário seu gerenciamento para que permaneça sempre disponível a seus usuários.

Neste ambiente, diferentes tipos de atividades estão envolvidas para que os serviços sejam providos com qualidade. O modelo de gerência OSI (*Open System Interconnection*) definido pela ISO (*International Standards Organization*) desmembra as tarefas de gerenciamento de rede em cinco pontos: gerência de falhas, gerência de desempenho, gerência de configuração, gerência de contabilidade e gerência de segurança.

A necessidade de mecanismos de gerenciamento nas redes baseadas em TCP/IP é atendida pelo padrão Internet (*Internet Standard Network Management Framework*), que utiliza o SNMP (*Simple Network Management Protocol*) em associação com o modelo de informação MIB (*Management Information Base*). Na Internet, o SNMP alcançou grande sucesso por ter sido o primeiro protocolo de gerenciamento não proprietário, público, fácil de ser implementado, possibilitando o gerenciamento efetivo de ambientes heterogêneos. Surgiram no mercado vários tipos de aplicações baseadas no padrão SNMP, quase sempre dependentes de alguma plataforma de gerência. Aplicações como o Nagios, as quais permitem o gerenciamento remoto de agentes através da internet, a medida em que crescem seus recursos e funcionalidades, vão perdendo sua simplicidade e robustez, tornando em alguns casos inviável sua utilização.

Hoje, a tecnologia empregada na Web está sendo usada na solução deste problema. Essa tendência não está sendo aplicada apenas à gerência de redes, mas também em *intranets* como forma de acessar bancos de dados, bases de conhecimentos, aplicações de escritórios, etc. O navegador vem se firmando como cliente universal, através do qual usuários acessam

os dados corporativos. No propósito da gerência de redes, estende-se esta idéia como interface padrão do sistema, que pode usar tecnologias empregadas na Web (DHTML, *Servlets*, JSP, PHP, etc.) para torná-la mais interativa e sofisticada.

O objetivo geral deste trabalho é a análise de um processo de software aplicada a um gerente SNMP pela web, utilizando funções específicas da extensão SNMP do PHP para efetuar consultas a agentes disponíveis. Esta será composta por toda a estrutura organizacional necessária para a codificação da aplicação, organização a qual deverá ser utilizada durante todo o período do projeto, iniciando no plano de gerenciamento do projeto e passando pelos planos de gerência de configuração, processo e metodologia de desenvolvimento, requisitos, arquitetura e finalizando no plano de testes do projeto. Estes planos contemplam toda a informação necessário para o desenvolvimento da aplicação aqui descrita, faltando apenas sua codificação.

A análise de um processo de software aplicada a um gerente SNMP pela web como objetivo geral é a base para especificar um processo de desenvolvimento de projetos baseados em internet, onde as informações estarão padronizadas e organizadas para todos os envolvidos, tornando assim o processo menos burocrático no que diz respeito a documentos necessários para um projeto organizado e mais rápido quanto a sua codificação.

Uma breve investigação tecnológica sobre a gerência de redes baseada em internet deve ser elaborada durante o processo de desenvolvimento do projeto, tendo como base o protocolo SNMP o qual é utilizado na aplicação.

Justifica-se este trabalho pela importância de uma integração no gerenciamento de agentes SNMP em um rede de computadores, utilizando este protocolo como base, onde é possível coletar e armazenar dados sobre estes agentes para que em torno destes dados seja feito análise de tendências, detecção de problemas e diagnóstico.

1.1 ORGANIZAÇÃO DO TRABALHO

O capítulo 2 apresenta o Referencial Teórico onde serão abordados os conceitos necessários para a compreensão dos aspectos envolvidos no desenvolvimento do projeto.

O capítulo 3 apresenta esclarecimentos sobre as Metodologias seguidas para a realização deste trabalho.

O capítulo 4 apresenta toda a estruturação necessária para o desenvolvimento da aplicação descrita neste documento.

2 REFERENCIAL TEÓRICO

2.1 REDE DE COMPUTADORES

Para Tanenbaum (2003), uma rede de computadores é um conjunto de computadores autônomos interconectados por diferentes tecnologias com objetivo único de compartilhar todos os recursos disponíveis ao alcance de todos os usuários.

“Dois computadores estão interconectados quando podem trocar informações. A conexão não precisa ser feita por um fio de cobre; também podem ser usadas fibras ópticas, microondas, ondas de infravermelho e satélites de comunicações.” (TANENBAUM, 2003, p. 2).

Atualmente as redes de computadores são constituídas por diversos dispositivos, tais como impressores, câmeras de vídeo, computadores de mão entre outros, todos estes dispositivos podem e constituem uma rede de computador a partir do momento em que todos eles estiverem interconectados por algum meio, seja ele físico (cabos), por frequências de rádio (redes sem fio) ou afins.

2.1.1 Gerenciamento de Redes

O objetivo da Gerência de Redes é monitorar e controlar os elementos da rede (sejam eles físicos ou lógicos), assegurando um certo nível de qualidade de serviço. Para realizar esta tarefa, os gerentes de redes são geralmente auxiliados por um sistema de gerência de redes. Um sistema de gerência de rede pode ser definido como uma coleção de ferramentas integradas para a monitoração e controle da rede. (LOPES; SAUVÉ; NICOLLETTI, 2003, p. 4)

A gerencia de redes envolve todas as atividades necessárias para o bom funcionamento de uma rede de computadores, em outras palavras, a gerencia de redes encarrega-se de monitorar e controlar os elementos de uma rede, assegurando, na medida do possível, um certo nível de qualidade dos serviços.

O papel das redes de dados nas empresas vem se tornando cada vez mais importante, redes estas, que antes estavam presentes apenas no setor de informática, estão cada vez maiores, abrangendo usuários de todos os setores da organização, fornecedores e clientes, sendo muitas das negociações efetuadas devido a sua existência e bom funcionamento.

(conveniência. Porém, é comum que nesta equipe existam profissionais que exercem quatro funções distintas: o pessoal de help desk, o operador de rede, a equipe de suporte técnico e o gerente da equipe de gerência. (LOPES; SAUVÉ; NICOLLETTI, 2003, p. 4)

Consideramos que computadores são equipamentos que realizam tarefas realizadas por pessoas e operam com alguns funcionalidades adicionais como, por exemplo, definição de VLANs (Virtual LANs).

Na ilustração 1 vemos roteadores, comutadores¹, repetidores, impressoras, servidores e estações clientes. Todos estes equipamentos podem ter agentes instalados. A estação de gerência deve obter informações destes agentes usando o protocolo SNMP (ver 2.1.3).

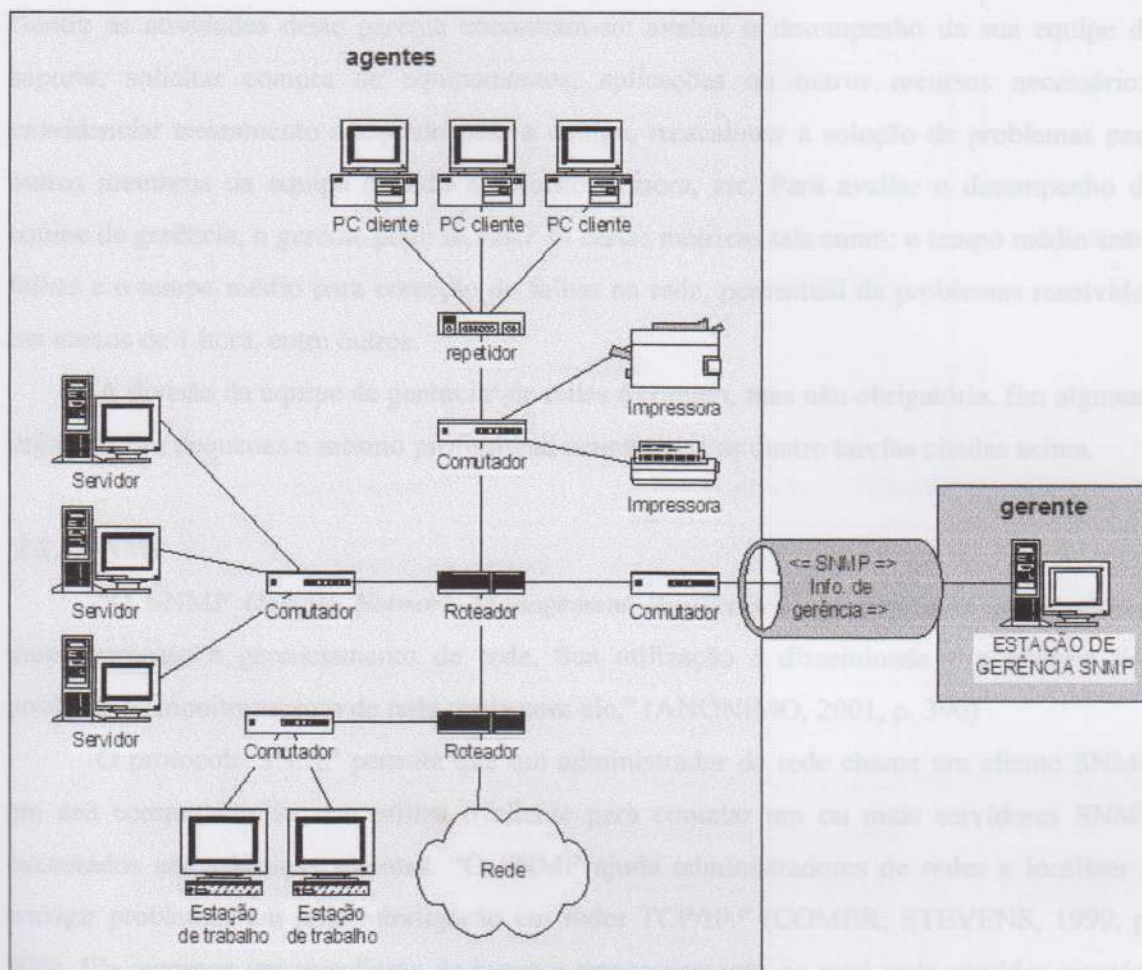


Ilustração 1: Elementos de uma arquitetura geral de soluções de gerência
Fonte: (LOPES; SAUVÉ; NICOLLETTI, 2003)

2.1.2 Gerente de Rede

Um dos objetivos da gerência de redes é prevenir e solucionar problemas na rede. Geralmente esta tarefa é realizada por uma equipe. Não existe uma regra rígida sobre os profissionais que fazem parte desta equipe. Cada organização tem autonomia para criar seu próprio time de gerência de redes de acordo com suas conveniências. Porém, é comum que nesta equipe existam profissionais que executem quatro tarefas distintas: o pessoal do *help desk*, o operador da rede, a equipe de suporte técnico e o gerente da equipe de gerência. (LOPES; SAUVÉ; NICOLLETTI, 2003, p. 6).

¹ Consideramos que comutadores são equipamentos que realizam tarefas realizadas por pontes e apresentam ainda algumas funcionalidades adicionais como, por exemplo definição de VLANs (Virtual LANs).

Na realidade não existe o gerente de rede, mas sim, uma equipe de gerência de redes, a qual é formada por diversos profissionais, os quais executam quatro tarefas distintas como: o pessoal do *help desk*, o operador da rede, a equipe de suporte técnico e o gerente da equipe.

O gerente tem um certo conhecimento em redes, mas não no nível do suporte técnico. Dentre as atividades deste gerente encontram-se: avaliar o desempenho da sua equipe de suporte, solicitar compra de equipamentos, aplicações ou outros recursos necessários, providenciar treinamento adequado para a equipe, reescalonar a solução de problemas para outros membros da equipe quando a solução demora, etc. Para avaliar o desempenho da equipe de gerência, o gerente pode se valer de certas métricas tais como: o tempo médio entre falhas e o tempo médio para correção de falhas na rede, percentual de problemas resolvidos em menos de 1 hora, entre outros.

A divisão da equipe de gerência² de redes é comum, mas não obrigatória. Em algumas organizações pequenas o mesmo profissional ocupa todas as quatro tarefas citadas acima.

2.1.3 SNMP

“O SNMP (*Simple Network Management Protocol*) é um protocolo para suportar monitoramento e gerenciamento de rede. Sua utilização é disseminada e a maioria dos produtos de monitoramento de rede conta com ele.” (ANÔNIMO, 2001, p. 390)

O protocolo SNMP permite que um administrador da rede chame um cliente SNMP em seu computador local e utiliza o cliente para contatar um ou mais servidores SNMP executados em máquinas remotas. “O SNMP ajuda administradores de redes a localizar e corrigir problemas em uma interligação em redes TCP/IP.” (COMER; STEVENS, 1999, p. 405). Ele emprega um paradigma de busca e armazenamento no qual cada servidor mantém um conjunto de variáveis conceituais que incluem estatísticas simples, tais como contagem de pacotes recebidos, e variáveis complexas que correspondem a estruturas de dados do TCP/IP.

Como o protocolo não abrange todas as operações, todo o controle precisa ser realizado por intermédio do paradigma de busca e armazenamento em um padrão a parte chamado de MIB (*Management Information Base – base de informações de gerenciamento*).

² Na realidade, o *help desk* não faz parte apenas da equipe da gerência da rede. Os usuários ligam para o *help desk* quando enfrentam problemas relacionados à tecnologia de informação, o que inclui problemas com a rede.

O protocolo SNMP é o mecanismo utilizado para permitir a troca das informações de gerência entre agentes e gerentes na rede. Sendo baseado em um simples mecanismo de busca/alteração, possui cinco primitivas básicas de comunicação, como mostra a ilustração 2, definidas como:

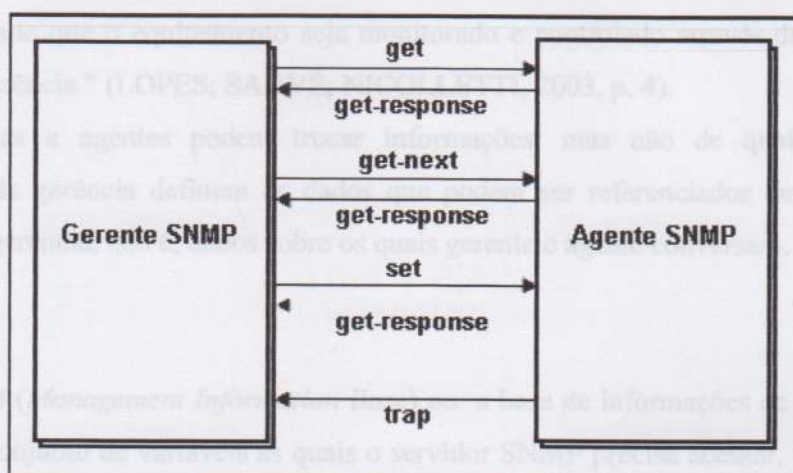


Ilustração 2: Primitivas básicas de comunicação do protocolo SNMP

Fonte: (COMER; STEVENS, 1999)

- **set**: permite ao gerente alterar o valor de uma variável presente em um agente.
- **get**: permite ao gerente obter o valor de uma variável presente em um agente.
- **get-next**: permite recuperar iterativa e sequencialmente a informação mantida pelo agente a partir das respostas obtidas por sucessivas requisições.
- **get-response**: contém a resposta a uma requisição SNMP ou o resultado da operação no caso de uma requisição do tipo **set**.
- **trap**: permite que o agente indique ao gerente alguma anormalidade ou erro ocorrido, sem nenhuma requisição prévia do gerente.

Atualmente, existem três versões deste protocolo. A primeira versão, cujas operações básicas foram comentadas acima, ficou conhecida como SNMPv1 e está sendo ainda bastante utilizada. A segunda versão, conhecida como SNMPv2, acrescentou funcionalidades para facilitar a recuperação da informação de gerência nos agentes e a comunicação entre gerentes distintos na rede. A última versão, conhecida como SNMPv3, preocupou-se com aspectos de segurança, buscando garantir a autenticidade e a integridade das operações de gerência.

2.1.3.1 Agentes

Os agentes existentes nos elementos gerenciados permitem o acesso à coleções de variáveis escalares armazenadas em uma base de gerência virtual, sendo as operações de gerência restritas à obtenção e controle dos valores destas variáveis.

“Elementos gerenciados possuem um software especial chamado agente. Este *software* permite que o equipamento seja monitorado e controlado através de uma ou mais estações de gerência.” (LOPES; SAUVÉ; NICOLLETTI, 2003, p. 4).

Gerentes e agentes podem trocar informações, mas não de qualquer tipo. As informações de gerência definem os dados que podem ser referenciados em operações de protocolo de gerência, isto é, dados sobre os quais gerente e agente conversam.

2.1.3.2 MIB

A MIB (*Management Information Base*) ou a base de informações de gerenciamento mantém um conjunto de variáveis as quais o servidor SNMP precisa acessar, elas podem ser divididas em duas classes: as variáveis simples e as tabelas. As variáveis simples abrangem tipos como inteiros com ou sem sinal e strings de caracteres. As tabelas correspondem a arrays unidimensionais, onde uma só tabela pode conter diversas instâncias de uma variável.

“A MIB define variáveis que um servidor SNMP precisa manter. Mais precisamente, a MIB define um conjunto de variáveis conceituais que um servidor SNMP precisa ter condições de acessar.” (COMER; STEVENS, 1999, p. 407).

As variáveis da MIB são representadas utilizando seus nomes, mas em alguns casos isso se torna muito extenso, assim, ficando mais difícil de ser utilizado, devido a isto, foi criada uma forma numérica (ASN.1³ – *Abstract Syntax Notation One*) para representar as variáveis da MIB, esta forma atribui um só inteiro a cada rótulo contido em um nome e representa o nome como uma sequência de inteiros onde são conhecidos como OID (*Object Identification – identificadores de objetos*).

A ASN.1 define uma ordenação lexicográfica de nomes que provê uma parte fundamental da funcionalidade do SNMP. A ordenação léxica permite que o cliente peça a um servidor o conjunto de variáveis correntemente disponíveis e consulte uma tabela sem conhecer seu tamanho. (COMER; STEVENS, 1999, p. 409).

³ Os nomes ASN.1 são conhecidos como *identificadores de objetos*.

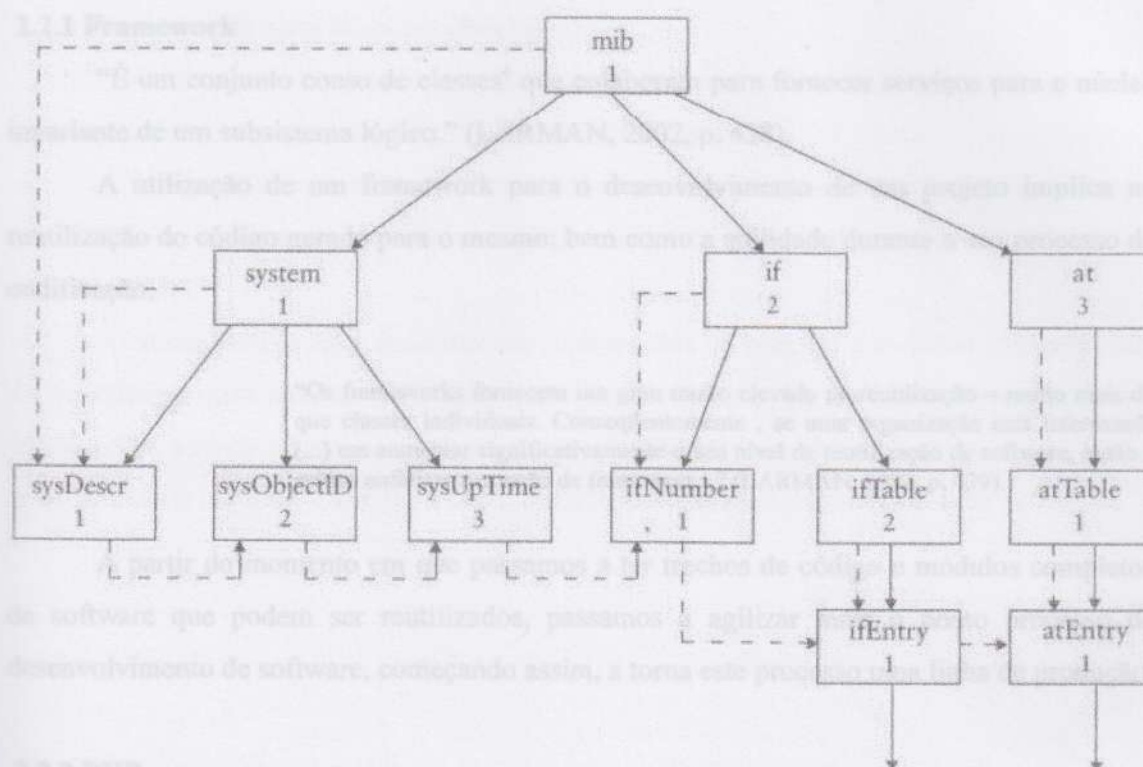


Ilustração 3: Parte da árvore conceitual de atribuição de nomes a variáveis da MIB

Fonte: (COMER; STEVENS, 1999)

Na ilustração 3 é apresentada como exemplo uma parte da árvore conceitual de atribuição de nomes a variável da MIB.

2.2 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

“Um processo de desenvolvimento de software é um método para organizar as atividades relacionadas com a criação, entrega e manutenção de sistemas de software.” (LARMAN, 2002, p. 40).

No processo de desenvolvimento de software devemos planejar todo o projeto do software a ser produzido, analisar quais são seus requisitos, quais ferramentas serão utilizadas durante o desenvolvimento do mesmo, quais as tecnologias aplicadas, o seu riscos, etc. Todo esse planejamento deve estar documentado e disponível para os participantes do projeto, de forma organizada e de fácil localização.

2.2.1 Framework

“É um conjunto coeso de classes⁴ que colaboram para fornecer serviços para o núcleo invariante de um subsistema lógico.” (LARMAN, 2002, p. 438).

A utilização de um framework para o desenvolvimento de um projeto implica na reutilização do código gerado para o mesmo, bem como a agilidade durante o seu processo de codificação.

“Os frameworks fornecem um grau muito elevado de reutilização – muito mais do que classes individuais. Conseqüentemente, se uma organização está interessada (...) em aumentar significativamente o seu nível de reutilização de software, então é crítico enfatizar a criação de frameworks.” (LARMAN, 2002, p. 439).

A partir do momento em que passamos a ter trechos de código e módulos completos de software que podem ser reutilizados, passamos a agilizar mais o nosso processo de desenvolvimento de software, começando assim, a torna este processo uma linha de produção.

2.2.2 PHP

Segundo Achour et al. (2005) o PHP (*Hypertext Preprocessor*) é uma linguagem de script Open Source de uso geral na web, muito utilizada para o desenvolvimento de aplicações baseadas em internet.

O PHP é focado para ser uma linguagem de script do lado do servidor, portanto, pode-se fazer qualquer coisa que outro programa CGI pode fazer, como: coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies.

Pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes Unix (incluindo HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS, e provavelmente outros. O PHP também é suportado pela maioria dos servidores web atuais, incluindo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet Servers, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd, e outros. O PHP pode ser configurado como módulo para a maioria dos servidores, e para os outros como um CGI comum.

⁴ Referindo a classes e/ou funções

2.2.3 Recuperação após Desastre (Backups)

“Em termos muito simples, recuperação após desastre é o ato de recompor-se depois que seus dados foram destruídos.” (ANÔNIMO, 2000, p. 554).

A recuperação após desastre focaliza atos nocivos humanos e outras ameaças (forças incontroláveis, equívocos inocentes, mecânica, bugs de software) que podem colocar em perigo os dados.

As recuperações após desastres são restaurações de backup⁵ dos dados efetuado antes da ocorrência, onde existem dois tipos principais de backup, os backups incrementais, que fazem cópia apenas daqueles arquivos que mudaram desde o último backup completo e o próprio backup completo, que faz a cópia de tudo que está contido na unidade de disco.

3.3 ESTUDO DE CASO

De acordo com Jung (2004), “Tudo nos sugere que a criação de uma nova tecnologia é um processo iterativo, onde se aprende com os erros e se ajusta o processo ou mesmo o produto, para que posteriormente seja reavaliado, desenvolvido e implantado em uma nova máquina, produto ou sistema de informação”.

⁵ Cópia extra de arquivos de dados

3 METODOLOGIA

3.1 TECNOLÓGICA

Segundo Jung (2004), a presente pesquisa científica é relatada como tecnológica, a qual utiliza conhecimentos básicos, tecnologias existentes e conhecimentos tecnológicos que tem como objetivo um novo produto ou processo.

3.2 EXPLICATIVA

Esta prática visa ampliar generalizações, definir leis mais amplas, estruturar e definir modelos teóricos, relacionar hipóteses em uma visão mais unitária do universo ou âmbito produtivo em geral e gerar hipóteses ou ideias por força de dedução lógica. A pesquisa explicativa exige maior investimento em síntese, teorização e reflexão a partir do objeto em estudo. (JUNG, 2004, p. 152).

3.3 ESTUDO DE CASO

De acordo com Jung (2004), "Tanto nas engenharias como na computação são necessários estudos anteriores para que se possa determinar como está funcionando o processo ou como é o processo, para que posteriormente seja recomendado, desenvolvido e implantado um novo sistema, produto ou técnica de otimização".

4 DESENVOLVIMENTO

4.1 PLANO DE GERENCIAMENTO

O objetivo do plano de gerenciamento é definir e planejar o desenvolvimento e acompanhamento do projeto, formas de comunicação entre os integrantes da equipe, estrutura do projeto, cronograma e requisitos.

4.1.1 Projeto

O desenvolvimento do projeto deve seguir o processo e metodologia definida a frente, na seção 4.3, bem como sua estrutura analítica e seu cronograma os quais são descritos nesta seção.

4.1.1.1 Estrutura analítica

A estrutura analítica do projeto trata-se da decomposição do trabalho em partes manejáveis, uma estrutura em árvore exaustiva, hierárquica de tarefas que precisam ser feitas para completar o projeto.

Atividade	Duração
1. Planejamento	5
1.1. Criar plano de projeto	1
1.2. Criar plano de gerência de configuração	5
2. Requisitos	3
2.1. Elicitar requisitos	2
2.2. Gerenciar requisitos	1
3. Arquitetura	2
3.1. Definir arquitetura	1
3.2. Reavaliar arquitetura	1
4. Interface com o usuário	2
4.1. Logotipos	1
4.2. Web	1
5. Implementação	2
5.1. Codificar projeto	2
6. Testes	1
6.1. Planejar execução de testes	1
6.2. Executar testes	1
6.3. Notificar erros	1
Total	61

7. Release

7.1. Gerar build

7.2. Disponibilizar release

8. Acompanhamento

8.1. Reunião de status

8.2. Resolução de conflitos

8.3. Ajustes no processo

8.4. Realização de backups

4.1.1.2 Cronograma

No cronograma do projeto conforme a tabela 1, será descrito as atividades e a quantidade de dias a qual cada uma deve durar, a partir do momento em que o projeto for aceito, incluindo seus riscos e as condições de níveis de serviço, deve ser adicionado a este cronograma as datas de início e fim de cada atividade descrita.

Tabela 1: Cronograma de atividades

Atividades	Duração
Definição dos requisitos	5
Definição da arquitetura	5
Validação e ajustes dos requisitos	5
Compatibilizar requisitos e arquitetura	2
Desenvolvimento do primeiro protótipo	5
Validação e ajustes do primeiro protótipo	3
Ajustes no processo de desenvolvimento	2
Desenvolvimento do segundo protótipo	8
Validação e ajustes do segundo protótipo	5
Ajustes no processo de desenvolvimento	2
Desenvolvimento do terceiro protótipo	6
Validação e ajustes do terceiro protótipo	4
Ajustes do processo de desenvolvimento	2
Documentação do projeto	6
Apresentação do projeto	1
Total	61

Fonte: Dados da pesquisa (2006)

4.1.2 Comunicação

Os meios de comunicação definidos para o projeto deve conter lista de e-mails, reuniões, discussões, artefatos e acompanhamento das tarefas.

4.1.2.1 Acompanhamento das tarefas

Todo o acompanhamento da tarefas deverá ser feito através de um software de controle de problemas (ver ferramentas em 4.2.3), onde as tarefas serão reportadas como problemas a serem resolvidos. O método básico de acompanhamento dessas tarefas será por meio dos relatórios gerados através software de controle.

4.1.2.2 Reuniões

Existem dois tipos de reuniões propostas para o projeto, reuniões presenciais, as quais devem ser marcadas com no mínimo de 24 horas de antecedência e sua pauta enviada por e-mail para todos os participantes, o responsável em registrar os tópicos tratados na reunião deve ser definido durante a mesma, reuniões virtuais, as quais serão executadas como uma conferência utilizando um software de mensagens instantânea (ver ferramentas em 4.2.3), todos os participantes devem ser cadastrados nos contatos de cada um, da mesma forma que a anterior a pauta da reunião deve ser enviada por e-mail para todos.

4.1.2.3 Discussões técnicas

As discussões técnicas do projeto serão realizadas utilizando uma ferramenta específica, baseada em internet, para fóruns de discussão (ver 4.2.3), onde cada um dos participantes do projeto deverá possuir um usuário e senha de acesso a esta ferramenta.

4.1.2.4 Lista de e-mails

Deverá existir um arquivo texto contendo todos os dados necessários (nome, e-mail, contato de mensagem instantânea e telefone) dos participantes do projeto em um local público (pasta compartilhada) definido no momento em que a equipe for definida.

4.1.2.5 Artefatos

São todos os documentos necessários para a manutenção do projeto, o responsável por eles deverá anunciar toda e qualquer alteração efetuada a todos os participantes.

4.1.3 Requisitos

O responsável pelo gerenciamento dos requisitos do projeto somente deverá aceitar novas requisições para que possam ser especificadas e analisadas até a fase/atividade de Documentação do projeto conforme mostra a tabela 1.

4.2 PLANO DE GERÊNCIA DE CONFIGURAÇÃO

O plano de gerencia de configuração tem por objetivo estabelecer toda a infraestrutura inicial necessária para a evolução do projeto.

4.2.1 Atividades

As atividades descritas abaixo devem ser seguidas pelos colaboradores do projeto, as quais são necessárias para manter a ordem e uma padronização sobre seus artefatos durante sua evolução.

4.2.1.1 Identificação

As atividades de identificação tratam de proporcionar ao colaborador do projeto uma visão de como os artefatos do mesmo serão identificados, seus pontos de localização, suas baselines e quais são os tipos de artefatos disponíveis.

4.2.1.1.1 identificador de documento

Todos os itens de configuração do projeto, com exceção do código fonte, devem possuir um identificador de documento (doc_id). Os doc_ids devem seguir o seguinte esquema de nomenclatura: **<nome_projeto>_<tipo_artefato>_<data_criacao>**

Onde:

- **<nome_projeto>** é o nome do projeto;
- **<tipo_artefato>** é o tipo de artefato, conforme apresentado na tabela 2;
- **<data_criacao>** é a data de criação do documento, no formato AAAAMMDD;
 - AAAA ano em quatro dígitos (ex: 2006);
 - MM mês em dois dígitos (ex: 11);
 - DD dia em dois dígitos (ex: 29).

Tabela 2: Tipos de artefatos

Artefato	tipo_artefato *
Plano de Gerência de Projeto de Software	SPMP
Plano de Gerência de Configuração de Software	SCMP
Plano de Testes de Software	STP
Pauta e Ata de Reunião	MEET
Relatórios	RPT
Documentos de Arquitetura	ARCH
Documentos de Requisitos	REQ
Documentos de Processos	PRC
Documentos de Testes	TST

(*) Abreviatura utilizada para definir o tipo de artefato que o documento representa

Fonte: Dados da pesquisa (2006)

4.2.1.1.2 número de versão

Todos os artefatos, excluindo o código, devem possuir um número de versão seguindo o padrão a seguir: **XX.YY**

Onde:

- **XX** é o número da versão do documento;
- **YY** é o número de rascunho do documento.

A evolução do número de versão deve seguir as seguintes regras:

- A primeira versão do documento de ver 00.01;
- A cada modificação do documento, o valor de YY deve ser incrementado;
- Após cada aprovação do documento, sua versão XX deve ser incrementada em um, e o valor YY volta para 00, formando assim uma versão oficial.

É considerada uma aprovação do documento:

- Após uma revisão;
- Após a aprovação de pelo menos dois membros do grupo responsável pelos artefatos.

4.2.1.1.3 local de armazenamento

Os locais de armazenamento do projeto são pastas compartilhadas através do repositório do CVS (*Concurrent Version System*) entre seus participantes. A estrutura de pastas é descrita na tabela 3.

Tabela 3: Locais de armazenamento do projeto

Pasta	Descrição
projeto/	Pasta base do projeto
projeto/codigos	Códigos do projeto, nesta pasta deve conter as plataformas de códigos do projeto caso o projeto possua mais de uma plataforma de trabalho
projeto/codigos/www	Códigos para plataforma web do projeto
projeto/docs	Documentação do projeto
projeto/docs/plan	Documentos de planejamento do projeto
projeto/docs/req	Documentos de requisitos do projeto
projeto/docs/design	Artefatos de design do projeto, modelos, diagramas e descrições da arquitetura do projeto
projeto/docs/conf	Documentos de gerência de configuração do projeto de software
projeto/docs/test	Documentos de teste para o projeto
projeto/relatorios	Relatórios do projeto

Fonte: Dados da pesquisa (2006)

4.2.1.2 Controle de configuração

Cada projeto deve possuir um core team (grupo principal) definido para cada papel. Estes grupos irão fazer o controle das configurações do projeto e decidirão a necessidade e urgência da implementação de solicitações de mudanças. Os core teams devem ser formados por no mínimo dois colaboradores.

4.2.1.3 Auditoria de configuração

As auditorias devem ser realizadas para cada ciclo do processo de desenvolvimento de forma a assegurar que os processos de gerência de configuração estão sendo seguidos e devem ser reportados.

4.2.2 Plano de Contingência

Todas as informações contidas no servidor CVS devem ser replicadas para um outro servidor a cada uma hora de trabalho, sendo esta feita de forma automática, em caso de indisponibilidade do servidor principal o secundário deve ser utilizado.

4.2.3 Ferramentas

Todas as ferramentas utilizadas durante o desenvolvimento do projeto estão descritas nos sub-tópicos abaixo, seus propósitos e objetivos.

4.2.3.1 Software de Controle de Problemas (Bugzilla)

O software de controle de problemas escolhido para o projeto é o Bugzilla, o qual esta disponível no site www.bugzilla.org, ele será utilizado para controlar os problemas encontrados durante o desenvolvimento do projeto e as tarefas de cada colaborador. O objetivo da utilização desta ferramenta é auxiliar e organizar o controle de tarefas e problemas os quais o projeto venha a ter.

4.2.3.2 Mensagens Instantânea (Windows Live Messenger)

O software para mensagens instantâneas é o Windows Live Messenger devido a sua grande utilização por parte da maioria das pessoas e seus recursos integrados, o mesmo pode ser encontrado em messenger.msn.com.br, ele será utilizado para as conferências/reuniões entre os colaboradores do projeto. O objetivo da utilização desta ferramenta é viabilizar durante as reuniões um canal de texto, voz e vídeo para uma melhor compreensão da mesma.

4.2.3.3 Fóruns de Discussão (phpBB)

A ferramenta que deverá ser utilizada para o gerenciamento das discussões técnicas será o phpBB, disponível em www.phpbb.com, onde os colaboradores do projeto deverão postar suas mensagens. Ela tem por objetivo organizar as informações que ocasionaram algum conflito durante o desenvolvimento do projeto afim de manter estes dados para uma futura consulta em caso de conflitos semelhantes.

4.2.3.4 Sistema de Controle de Versões (WinCVS, TortoiseCVS)

O sistema de controle de versões dos artefatos do projeto, inclusive os códigos fonte do projeto é o WinCVS, disponível em www.wincvs.org, opera completamente em ambiente Windows e deverá possuir dois servidores operando com o mesmo sendo um primário e outro secundário para réplica, o TortoiseCVS, disponível em www.tortoisecvs.org, será utilizado pelos colaboradores do projeto afim de compartilhar, sincronizar e gerenciar os arquivos nos servidores CVS. O objetivo do WinCVS é centralizar toda a informação do projeto e permitir o acesso simultâneo as mesmas. O objetivo do TortoiseCVS é facilitar o acesso a esta informação por meio dos colaboradores.

4.3 PROCESSO E METODOLOGIA DE DESENVOLVIMENTO

O plano para o processo e metodologia de desenvolvimento do projeto tem como base a agilidade para com o mesmo, assim tornando todo o andamento do projeto mais estável, rápido e seguro.

4.3.1 Processo

O processo de desenvolvimento adotado para este projeto tem como base o mesmo utilizado pela USINA (*United Software Infra-structure Alliance*), onde é dividido em três fases (ver ilustração 5) as quais se repetem até o lançamento do release final do projeto.

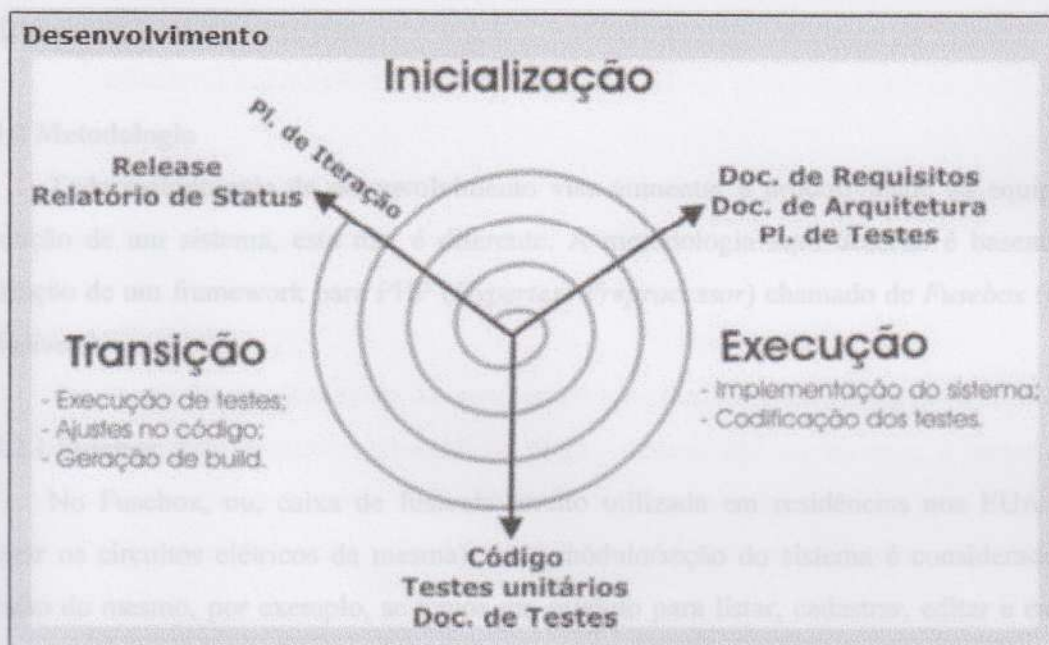


Ilustração 4: Processo de desenvolvimento
Fonte: USINA (2006)

4.3.1.1 Fases

As fases de desenvolvimento do projeto são em espiral, como mostra a ilustração 5, estas fases irão se repetir até que o projeto esteja concluído.

4.3.1.1.1 inicialização

Onde é definido o plano para cada uma das iterações realizadas. Nesta fase também é fechado, ou ajustado, acordos sobre os requisitos e arquitetura de software, permitindo a criação do plano de testes. Os marcos finais desta fase são os documentos de requisitos, arquitetura e plano de testes disponíveis.

4.3.1.1.2 execução

Nesta etapa é realizada a implementação do software de acordo com que foi definido nos documentos de requisitos e arquitetura. Ao término desta fase deverá ser apresentado o código do sistema relativo ao que foi planejado na iteração, bem como os testes unitários e o documento de testes que será usado na etapa seguinte.

4.3.1.1.3 transição

É realizado os testes e são tomadas medidas corretivas necessárias para a correção dos erros detectados nos testes de forma que seja possível a realização do release. Ao final desta fase deverá ser apresentado um release, bem como, um relatório indicando o andamento do projeto.

4.3.2 Metodologia

Toda metodologia de desenvolvimento visa aumentar a produtividade da equipe na produção de um sistema, esta não é diferente. A metodologia aqui descrita é baseada na utilização de um framework para PHP (*Hypertext Preprocessor*) chamado de *Fusebox* (caixa de fusíveis).

4.3.2.1 Fusebox

No Fusebox, ou, caixa de fusíveis (muito utilizada em residências nos EUA para separar os circuitos elétricos da mesma), cada módulo/seção do sistema é considerado um circuito do mesmo, por exemplo, se temos um módulo para listar, cadastrar, editar e excluir usuários podemos dizer que temos um circuito de usuários no sistema o qual permite estas ações. Esta divisão do sistema em circuitos promove a reutilização e manutenção do código, tornando assim o trabalho mais rápido e em linha de produção.

Os circuitos em Fusebox são pastas na raiz do site, toda pasta de circuito por padrão deve possuir o nome do mesmo, os sub-circuitos são pastas na raiz do circuito pai e também devem possuir o mesmo nome dos sub-circuitos. Os circuitos que são integrados ao framework devem estar localizados na pasta `_default/internal/`.

Em Fusebox é permitida a utilização de vários layouts para um mesmo sistema e para cada circuito, desvinculando assim, a interface do usuário (design) do código da lógica do sistema.

4.3.2.1.1 core (núcleo)

Todos os arquivos core do framework possuem o pré-fixado **fbx_**, assim são identificados mais facilmente, apenas com a exceção do **index.php** o qual é o arquivo de inicialização do mesmo. Os arquivos core do framework são:

- a) **fbx_fusebox.php**: Este arquivo é o núcleo principal do framework, nele são processadas todas as requisições ao sistema e inclusões base.
- b) **fbx_settings.php**: Arquivo que define todas as configurações do sistema.
- c) **fbx_circuits.php**: Define todos os circuitos existentes no sistema.
- d) **fbx_layouts.php**: Define quais os layouts disponíveis para utilização no sistema e suas respectivas configurações.
- e) **fbx_switch.php**: Faz as inclusões necessárias para satisfazer as requisições de ações solicitadas pelo circuito ao sistema.

Em comum, todos os circuitos devem possuir os arquivos **fbx_settings.php**, **fbx_layouts.php** e **fbx_switch.php** devidamente configurados para o circuito.

4.3.2.1.2 bibliotecas

Da mesma forma e seguindo as considerações do framework, os arquivos de biblioteca terão o pré-fixado **lib_** e estarão situados na pasta **_default/lib/** do sistema, a maioria das bibliotecas do sistema serão incluídas no arquivo **fbx_fusebox.php**, com exceção apenas das que são dependentes de uma outra biblioteca ou as que tem sua utilização específica em algum circuito.

4.3.2.1.3 interfaces com o usuário

Seguindo a padronização proposta, os arquivos de display, ou seja, as interfaces com o usuário, devem possuir sempre o pré-fixado **dsp_**. Para arquivos de layout do sistema ou circuito também é adotado este pré-fixado, sendo que, para arquivos de layout do sistema, os mesmos devem estar localizados na pasta **_default/layout/** e para circuitos com mais de um layout devem estar localizados na pasta **_layout/** dentro da raiz do circuito referenciado.

4.3.2.1.4 validações

Os arquivos de validações são antecedentes aos arquivos de ações, todas as validações que são necessariamente processadas do lado do servidor estarão programadas nestes arquivos, o pré-fixos para estes tipos de arquivos é **vld_**.

4.3.2.1.5 ações

Os arquivos de ações por sua vez serão os responsáveis em executar todas as ações do circuito o qual reside, por padrão todos devem possuir o pré-fixos **act_**.

4.3.2.1.6 pastas padrão

A estrutura padrão de pastas do framework é apresentada conforme ilustração 6 e descrita na tabela 4, esta estrutura, por regra, não pode ser alterada, a mesma será criada automaticamente quando o arquivo compactado de projeto base do framework for descompactado dentro da pasta do sistema.

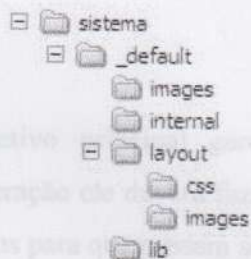


Ilustração 5: Estrutura padrão de pastas do framework
Fonte: Dados da pesquisa (2006)

Tabela 4: Descrição da estrutura padrão de pastas do framework

Pasta	Descrição
/	Raiz do sistema.
/_default/	Pasta padrão para armazenamento de recursos internos do sistema.
/_default/lib/	Pasta para o armazenamento de arquivos da biblioteca do sistema.
/_default/internal/	Pasta para o armazenamento de circuitos internos do sistema.
/_default/layout/	Pasta para o armazenamento de arquivos de layout do sistema.
/_default/layout/css/	Pasta para o armazenamento de arquivos CSS de definições visuais do sistema.
/_default/layout/images/	Pasta para o armazenamento de arquivos de imagens utilizadas nos layouts do sistema.
/_default/images/	Pasta para o armazenamento de arquivos de imagens do sistema. Utilizadas em todos os circuitos.

Fonte: Dados da pesquisa (2006)

4.4 REQUISITOS

Os requisitos do sistema são definidos para que o mesmo tenha objetivos, nesta etapa do projeto são apresentados uma visão do produto, que tipo de usuários o mesmo terá que satisfazer, quais as funcionalidades que ele deverá possuir e quais suas restrições.

4.4.1 Visão Geral do Produto

O sistema tem como objetivo principal gerenciar agentes SNMP disponíveis, utilizando um ambiente de fácil interação ele deverá fazer consultas em agentes e armazenar seus resultados em uma base de dados para que possam ser utilizados posteriormente.

O público alvo do sistema são os administradores de pequenas e grandes redes que buscam uma ferramenta que possa fazer consultas em agentes da sua rede de forma simples e ao mesmo tempo segura, através de qualquer ambiente, consultas que podem ter seus resultados armazenados em uma base de dados para que se possa gerar informação através delas.

O sistema terá uma grande importância quanto a agilidade na resolução e detecção de problemas na rede, obtendo assim um grande sucesso como ferramenta de auxílio ao gerenciamento de redes.

4.4.1.1 Descrição dos usuários

Os usuários finais do sistema são administradores de redes que possuem conhecimentos básicos sobre agentes SNMP e desejam utilizar dos recursos disponíveis por eles para resolução de problemas em sua rede.

4.4.1.2 Parâmetros do sistema

Os parâmetros do sistema devem informar quais configurações o projeto deve receber em relação a sua utilização e seriam os seus processos os quais lhe permitiram evoluir.

4.4.1.2.1 parâmetros alteráveis

- O sistema permitirá a utilização de outra base de dados (do mesmo sistema).

4.4.1.2.2 parâmetros não-alteráveis

- O sistema não promove a mudança em sua forma e metodologia de trabalho (do sistema).

4.4.1.2.3 processos evolutivos

- O sistema possibilitará a exportação dos seus dados.

4.4.2 Premissas e Restrições

- O sistema se restringe a disponibilizar apenas 1 (um) usuário com privilégios avançados;
- O sistema se restringe ao uso apenas por usuários devidamente credenciados.

4.4.3 Requisitos Funcionais

- O sistema deve funcionar em qualquer sistema operacional através de um navegador de internet;
- O sistema deve armazenar na base de dados o resultado de qualquer consulta a um agente originada por ele;
- O sistema deve possibilitar o backup de todas as informações contidas em sua base de dados;
- O sistema deve disponibilizar uma ajuda on-line para o auxílio aos usuários.

4.4.4 Requisitos Não-funcionais

- O banco de dados do sistema deve ser protegido com acesso apenas a usuários devidamente credenciados;
- Os usuários devem ter noções de informática e redes para a manipulação do sistema.

4.5 ARQUITETURA

A arquitetura do sistema apresenta todo o modelo funcional do projeto, como deve ser a navegação pelo mesmo, qual a estrutura do seu banco de dados e a descrição dos seus casos de uso.

4.5.1 Modelo Navegacional

O modelo navegacional permite ao usuário entender a como utilizar o sistema proposto, mostrando como é feita a transição entre as telas da aplicação e o propósito de cada uma delas.

Web-SNMP

Acesso ao Sistema

E-Mail:

Senha:

Cancelar

OK

Ilustração 6: Acesso ao sistema
Fonte: Dados da pesquisa (2006)

No momento em que o usuário efetuar o acesso ao sistema (ver Ilustração 6), o sistema irá verificar se o usuário é o root ou não, as ilustrações 7 e 8 são telas disponíveis apenas para o usuário root, as demais telas são específicas para o restante dos usuários e o root não tem acesso a elas.

Web-SNMP

Bem vindo ROOT! Listando usuários do sistema...

[Cadastrar novo usuário](#)

EDITAR	CÓDIGO	E-MAIL	SENHA	STATUS
E	1	usuario@dominio.com.br	123456	Ativo

Ilustração 7: Listando os usuários do sistema

Fonte: Dados da pesquisa (2006)

A ilustração acima é apresentada quando o usuário root acessa o sistema, nesta tela ele pode clicar no ícone de edição “E” ou em “Cadastrar novo usuário”, ambas as ações o levarão para a ilustração abaixo, sendo que, quando for uma edição os dados do usuário selecionado estarão já disponíveis no formulário, caso contrário, o formulário aparecerá em branco.

Web-SNMP

Bem vindo ROOT! Manutenção de usuário do sistema...

[Listar usuários](#)

E-MAIL	
SENHA	
STATUS	Ativo v
Cancelar Limpar Excluir Salvar	

Ilustração 8: Manutenção de usuário do sistema

Fonte: Dados da pesquisa (2006)

A ilustração acima possibilita o root alterar os dados de um usuário existente ou adicionar um novo, nesta tela ele também pode efetuar a exclusão do usuário que estará sendo editado no momento.

Web-SNMP

Bem vindo USUARIO! O que deseja fazer hoje?

[Listar Agentes](#)
[Listar Plataformas](#)
[Listar MIBs](#)
[Listar Objetos MIB](#)
[Visualizar Historico](#)

Ilustração 9: Página principal de acesso para usuários não root do sistema

Fonte: Dados da pesquisa (2006)

A ilustração 9 é apresentada quando um usuário comum do sistema (não root) efetua o acesso ao mesmo, esta tela possibilita o usuário navegar por todo o sistema, as telas apresentadas abaixo mostrará toda a navegação na mesma ordem do menu da ilustração 6.

Web-SNMP

Bem vindo USUARIO! Listando agentes do sistema...

[Cadastrar novo agente](#)

EDITAR	CÓDIGO	NOME	DESCRIÇÃO	HOST	PLATAFORMA	STATUS
E	1	ZEUS	Computador pessoal	127.0.0.1	Windows 2003 Enterprise	Ativo

Ilustração 10: Listando os agentes do sistema

Fonte: Dados da pesquisa (2006)

Acima como mostra a ilustração 10, possibilita o usuário editar ou inserir um novo agente ao sistema.

Web-SNMP

Bem vindo USUARIO! Manutenção de agente do sistema...

[Listar agentes](#)

PLATAFORMA	Windows 2003 Enterprise
NOME	
HOST	
COMUNIDADE	
DESCRIÇÃO	
NOME DE SEGURANÇA	
NÍVEL DE SEGURANÇA	
PROTOCOLO DE AUTENTICAÇÃO	
FRASE PASSE DE AUTENTICAÇÃO	
PROTOCOLO PRIVADO	
FRASE PASSE PRIVADA	
STATUS	Ativo
Cancelar Limpar Excluir Salvar	

Ilustração 11: Manutenção de agentes do sistema

Fonte: Dados da pesquisa (2006)

O usuário poderá editar ou excluir um registro existente ou inserir um novo registro de agente ao sistema (ver ilustração 11).

Web-SNMP

Bem vindo USUARIO! Listando plataformas do sistema...

[Cadastrar nova plataforma](#)

EDITAR	CÓDIGO	NOME	DESCRIÇÃO	STATUS
E	1	Windows 2003 Enterprise	Ambiente Windows 2003 com suporte a Net-SNMP	Ativa

Ilustração 12: Listando plataformas do sistema
Fonte: Dados da pesquisa (2006)

A ilustração 12 apresenta ao usuário uma listagem de plataformas cadastradas no sistema e possibilita o mesmo acessar a manutenção de cada registro.

Web-SNMP

Bem vindo USUARIO! Manutenção de plataforma do sistema...

[Listar plataformas](#)

NOME	
DESCRIÇÃO	
STATUS	Ativo ▾
MIBs	☐ SNMPv2-TM ☐ SNMPv2-MIB ☐ SNMP-FRAMEWORK-MIB ☐ SNMP-TARGET-MIB
Cancelar Limpar Excluir Salvar	

Ilustração 13: Manutenção de plataforma do sistema
Fonte: Dados da pesquisa (2006)

A ilustração 13 mostra a tela de manutenção (cadastro, alteração e exclusão) de plataformas do sistema.

Web-SNMP

Bem vindo USUARIO! Listando MIBs do sistema...

[Cadastrar nova mib](#)

EDITAR	CÓDIGO	NOME	DESCRIÇÃO	RFC	STATUS
E	1	SNMPv2-MIB	The MIB module for SNMP entities.	rfc3418	Ativa

Ilustração 14: Listando MIBs do sistema
Fonte: Dados da pesquisa (2006)

Na ilustração 14 é apresentado ao usuário uma listagem de MIBs disponíveis no sistema.

Ilustração 15: Manutenção de MIB do sistema
Fonte: Dados da pesquisa (2006)

A ilustração 15 disponibiliza a tela de manutenção de MIBs do sistema.

EDITAR	CÓDIGO	NOME	OID	MIB	STATUS
E	1	coldStart	.1.3.6.1.6.3.1.1.5.1	SNMPv2-MIB	Ativo

Ilustração 16: Listando objetos MIB do sistema
Fonte: Dados da pesquisa (2006)

A ilustração 16 apresenta a listagem dos objetos de cada MIB disponível no sistema.

Web-SNMP

Bem vindo USUARIO! Manutenção de objeto mib do sistema...

[Listar objetos mib](#)

MIB	SNMPv2-MIB
NOME	
OID	
TIPO	
ACESSO	
DESCRIÇÃO	
STATUS	Ativo
Cancelar Limpar Excluir Salvar	

Ilustração 17: Manutenção de objeto MIB do sistema

Fonte: Dados da pesquisa (2006)

A ilustração 17 mostra o formulário de manutenção dos objetos das MIBs do sistema.

Web-SNMP

Bem vindo USUARIO! Visualizando histórico do sistema...

[Consultar agente](#)

DATA	USUÁRIO	AGENTE	HOST	MIB	OBJETO MIB	RESULTADO
01/12/2006 00:00:00	usuario@dominio.com.br	ZEUS	127.0.0.1	SNMPv2-MIB	coldStart (1.3.6.1.6.3.1.1.5.1)	?????????

Ilustração 18: Visualizando histórico de consultas do sistema

Fonte: Dados da pesquisa (2006)

Na ilustração 18 a única possibilidade oferecida ao usuário é para efetuar uma consulta a um agente cadastrado no sistema, esta ilustração esboça os dados do histórico de consultas do sistema gravados de consultas anteriores.

Web-SNMP

Bem vindo USUARIO! Consultar agente do sistema...

AGENTE	ZEUS
MIB	SNMPv2-MIB
OBJETO DA MIB	coldStart(1.3.6.1.6.3.1.1.5.1)
RESULTADO	

Ilustração 19: Consultando um agente cadastrado no sistema
Fonte: Dados da pesquisa (2006)

A ilustração 19 possibilita o usuário efetuar a consulta a um agente, o resultado da consulta é gravado no banco de dados do sistema e após exibido nesta mesma tela apenas o resultado da mesma.

4.5.2 Modelagem dos Casos de Uso

A modelagem dos casos de uso esboça o comportamento do sistema perante requisições do usuário, quais são os tipos de usuários que interagem com o mesmo e qual o fluxo de cada caso de uso.

Ilustração 20: Diagrama de casos de uso
Fonte: Dados da pesquisa (2006)

4.5.2.2 Atores

Usuário: Identifica uma pessoa que é responsável por fazer os cadastros usuais do sistema, que são: cadastro de plataforma, cadastro de agente, cadastro de mib e cadastro de objetos de mib; bem como responsável por fazer consultas a agentes cadastrados.

4.5.2.1 Diagrama

O diagrama de casos de uso pode ser visualizado na ilustração 20 e seus fluxos serão descritos na seção 4.5.2.3.

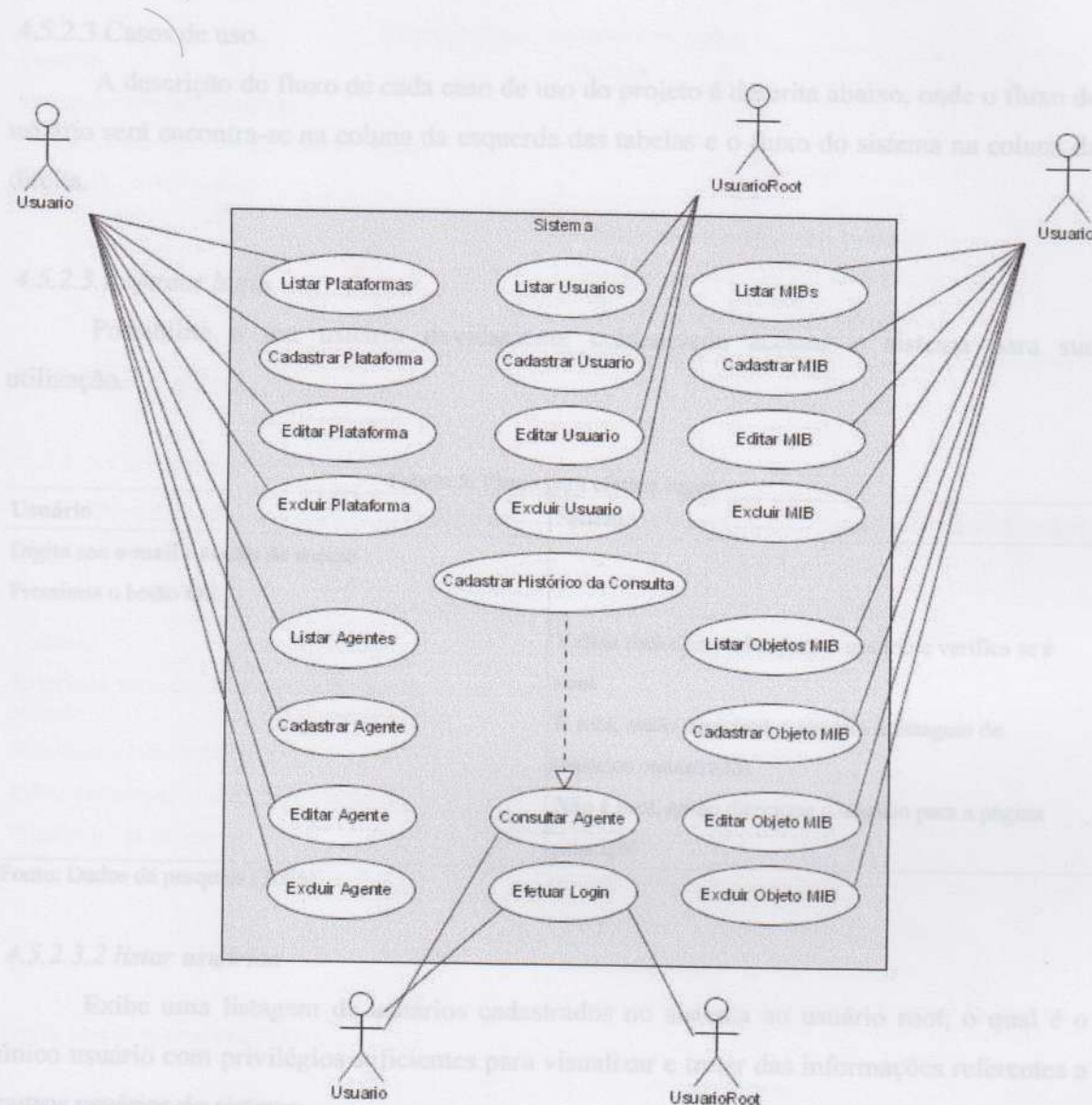


Ilustração 20: Diagrama de casos de uso
Fonte: Dados da pesquisa (2006)

4.5.2.2 Atores

- **Usuario:** Identifica uma pessoa que é responsável por fazer os cadastros usuais do sistema, que são: cadastro de plataforma, cadastro de agente, cadastro de mib e cadastro de objetos da mib; bem como responsável por fazer consultas a agentes cadastrados.

- **UsuárioRoot:** Identifica uma pessoa que é responsável por fazer o cadastro de outros usuário no sistema, usuários estes que irão o utilizar.

4.5.2.3 Casos de uso

A descrição do fluxo de cada caso de uso do projeto é descrita abaixo, onde o fluxo de usuário será encontra-se na coluna da esquerda das tabelas e o fluxo do sistema na coluna da direita.

4.5.2.3.1 efetuar login

Possibilita a um usuário devidamente credenciado acessar o sistema para sua utilização.

Tabela 5: Fluxo para efetuar login

Usuário	Sistema
Digita seu e-mail e senha de acesso	
Pressiona o botão OK	
	Valida dados enviados para o usuário e verifica se é root
	É root, então direciona o usuário a listagem de usuários cadastrados
	Não é root , então direciona o usuário para a página principal

Fonte: Dados da pesquisa (2006)

4.5.2.3.2 listar usuários

Exibe uma listagem de usuários cadastrados no sistema ao usuário root, o qual é o único usuário com privilégios suficientes para visualizar e tratar das informações referentes a outros usuários do sistema.

Tabela 6: Fluxo para listar usuários

Usuário	Sistema
Pressiona link listar usuários	
	Consulta lista de usuários no banco de dados
	Existe usuários, então mostra os usuários cadastrado
	Não existe usuários, então exibe mensagem de registros não encontrados

Fonte: Dados da pesquisa (2006)

4.5.2.3.3 cadastrar usuário

Permite efetuar o cadastro de um novo usuário do sistema.

Tabela 7: Fluxo para cadastrar usuário

Usuário	Sistema
Pressiona link novo usuário	
Preenche campos (e-mail, senha e status)	
Pressiona o botão salvar	
	Valida campos preenchidos pelo usuário Insere informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de usuários

Fonte: Dados da pesquisa (2006)

4.5.2.3.4 editar usuário

Permite que um usuário já cadastrado no sistema tenha suas informações modificadas.

Tabela 8: Fluxo para editar usuário

Usuário	Sistema
Seleciona na listagem de usuários o usuário que será editado	
Pressiona o ícone editar	
Edita os campos (e-mail, senha e status)	
Pressiona o botão salvar	
	Valida campos preenchidos Altera informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de usuários

Fonte: Dados da pesquisa (2006)

4.5.2.3.5 excluir usuário

Permite a exclusão de um usuário já cadastrado no sistema.

Tabela 9: Fluxo para excluir usuário

Usuário	Sistema
Seleciona na listagem de usuários o usuário que será excluído Pressiona o ícone editar Pressiona o botão excluir	Exclui informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de usuários

Fonte: Dados da pesquisa (2006)

4.5.2.3.6 listar plataformas

Permite ao usuário visualizar uma lista das plataformas cadastradas no banco de dados do sistema.

Tabela 10: Fluxo para listar plataformas

Usuário	Sistema
Pressiona link listar plataformas	Consulta lista de plataformas no banco de dados Existe plataformas, então mostra as plataformas cadastradas Não existe plataformas, então exibe mensagem de registros não encontrados

Fonte: Dados da pesquisa (2006)

4.5.2.3.7 cadastrar plataforma

Efetuar o cadastro de uma nova plataforma no banco de dados do sistema.

Tabela 11: Fluxo para cadastrar plataforma

Usuário	Sistema
Pressiona link nova plataforma Preenche campos (nome, descrição, status e seleciona as mibs suportadas) Pressiona o botão salvar	Valida campos preenchidos pelo usuário Insere informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de plataformas

Fonte: Dados da pesquisa (2006)

4.5.2.3.8 editar plataforma

Efetuar a edição de uma plataforma já cadastrada no banco de dados do sistema.

Tabela 12: Fluxo para editar plataforma

Usuário	Sistema
Seleciona na listagem de plataformas a plataforma que será editada Pressiona o ícone editar Edita os campos (nome, descrição, status e seleciona as mibs suportadas) Pressiona o botão salvar	Valida campos preenchidos Altera informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de plataformas

Fonte: Dados da pesquisa (2006)

4.5.2.3.9 excluir plataforma

Efetua a exclusão de uma plataforma já cadastrada no banco de dados do sistema.

Tabela 13: Fluxo para excluir plataforma

Usuário	Sistema
Seleciona na listagem de plataformas a plataforma que será excluída Pressiona o ícone editar Pressiona o botão excluir	Exclui informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de plataformas

Fonte: Dados da pesquisa (2006)

4.5.2.3.10 listar mibs

Permite que usuários do sistema visualizem as mibs cadastradas no banco de dados.

Tabela 14: Fluxo para listar mibs

Usuário	Sistema
Pressiona link listar mibs Pressiona o ícone editar Editar os campos (nome, rdt, descrição, arquivo, status)	Consulta lista de mibs no banco de dados Existe mibs, então mostra as mibs cadastradas Não existe mibs, então exibe mensagem de registros não encontrados

Fonte: Dados da pesquisa (2006)

4.5.2.3.11 cadastrar mib

Permite cadastrar nova mib no banco de dados do sistema.

Tabela 15: Fluxo para cadastrar mib

Usuário	Sistema
Pressiona link nova mib Preenche campos (nome, rfc, descrição, arquivo, status) Pressiona o botão salvar	Valida campos preenchidos pelo usuário Insere informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de mibs

Fonte: Dados da pesquisa (2006)

4.5.2.3.12 editar mib

Permite a edição de uma mib já cadastrada no banco de dados do sistema.

Tabela 16: Fluxo para editar mib

Usuário	Sistema
Seleciona na listagem de mibs a mib que será editada Pressiona o ícone editar Edita os campos (nome, rfc, descrição, arquivo, status) Pressiona o botão salvar	Valida campos preenchidos Altera informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de mibs

Fonte: Dados da pesquisa (2006)

4.5.2.3.13 excluir mib

Permite que uma mib já cadastrada no banco de dados do sistema seja excluída.

Tabela 17: Fluxo para excluir mib

Usuário	Sistema
Seleciona na listagem de mibs a mib que será excluída Pressiona o ícone editar Pressiona o botão excluir	Exclui informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de mibs

Fonte: Dados da pesquisa (2006)

4.5.2.3.14 listar objetos mib

Exibe uma lista dos objetos de mibs cadastradas no banco de dados do sistema.

Tabela 18: Fluxo para listar objetos mib

Usuário	Sistema
Pressiona link listar objetos mib	Consulta lista de objetos mib no banco de dados Existe objetos mib, então mostra os objetos mib cadastradas Não existe objetos mib, então exibe mensagem de registros não encontrados

Fonte: Dados da pesquisa (2006)

4.5.2.3.15 cadastrar objeto mib

Permite o cadastro de um objeto mib no banco de dados do sistema.

Tabela 19: Fluxo para cadastrar objeto mib

Usuário	Sistema
Pressiona link novo objeto mib Preenche campos (seleciona a mib do objeto, nome, oid, tipo, acesso, descrição e status) Pressiona o botão salvar	Valida campos preenchidos pelo usuário Insere informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de objetos mib

Fonte: Dados da pesquisa (2006)

4.5.2.3.16 editar objeto mib

Permite que um objeto mib já cadastrado no banco de dados do sistema seja alterado.

Tabela 20: Fluxo para editar objeto mib

Usuário	Sistema
Seleciona na listagem de objetos mib o objeto mib que será editado Pressiona o ícone editar Edita os campos (seleciona a mib do objeto, nome, oid, tipo, acesso, descrição e status) Pressiona o botão salvar	Valida campos preenchidos Altera informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de objetos mib

Fonte: Dados da pesquisa (2006)

4.5.2.3.17 excluir objeto mib

Permite que um objeto mib já cadastrado no banco de dados do sistema seja excluído.

Tabela 21: Fluxo para excluir objeto mib

Usuário	Sistema
Seleciona na listagem de objetos mib o objeto mib que será excluído Pressiona o ícone editar Pressiona o botão excluir	Exclui informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de objetos mib

Fonte: Dados da pesquisa (2006)

4.5.2.3.18 listar agentes

Permite visualizar uma lista de agentes cadastrados no banco de dados do sistema.

Tabela 22: Fluxo listar agentes

Usuário	Sistema
Pressiona link listar agentes	Consulta lista de agentes no banco de dados Existe agentes, então mostra os agentes cadastrados Não existe agentes, então exibe mensagem de registros não encontrados

Fonte: Dados da pesquisa (2006)

4.5.2.3.19 cadastrar agente

Permite que seja inserido um novo agente ao banco de dados do sistema.

Tabela 23: Fluxo para cadastrar agente

Usuário	Sistema
Pressiona link novo agente	
Preenche campos (seleciona a plataforma, nome, descrição, host, comunidade, nome de segurança, nível de segurança, protocolo de autenticação, frase passe de autenticação, protocolo privado, frase passe privada e status)	
Pressiona o botão salvar	Valida campos preenchidos pelo usuário Insere informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de agentes

Fonte: Dados da pesquisa (2006)

4.5.2.3.20 editar agente

Permite que agentes já cadastrado no banco de dados do sistema tenham seus dados alterados.

Tabela 24: Fluxo para editar agente

Usuário	Sistema
Seleciona na listagem de agentes o agente que será editado	
Pressiona o ícone editar	
Edita os campos (seleciona a plataforma, nome, descrição, host, comunidade, nome de segurança, nível de segurança, protocolo de autenticação, frase passe de autenticação, protocolo privado, frase passe privada e status)	
Pressiona o botão salvar	Valida campos preenchidos Altera informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de agentes

Fonte: Dados da pesquisa (2006)

4.5.2.3.21 *excluir agente*

Permite que um agente já cadastrado no banco de dados do sistema tenha seus dados removidos.

Tabela 25: Fluxo para excluir agente

Usuário	Sistema
Seleciona na listagem de agentes o agente que será excluído Pressiona o ícone editar Pressiona o botão excluir	Exclui informações no banco de dados Exibe mensagem de êxito ao usuário Direciona o usuário a página de listagem de agentes

Fonte: Dados da pesquisa (2006)

4.5.2.3.22 *consultar agente*

Permite que o usuário do sistema faça uma consulta a um objeto mib o qual o agente selecionado tem suporte.

Tabela 26: Fluxo para consultar agente

Usuário	Sistema
Seleciona na listagem de agentes o agente que será consultado Pressiona o botão OK	Exibe uma lista de objetos mib o qual o agente tem suporte
Seleciona o objeto mib que ele deseja consultar Pressiona o botão Consultar	Efetua consulta ao objeto mib do agente selecionado Grava dados da consulta realizada no banco de dados Exibe a informação consultada na tela para o usuário

Fonte: Dados da pesquisa (2006)

4.5.3 Armazenamento

Os dados deste projeto serão armazenados em um banco de dados o qual será descrito o seu modelo físico abaixo, a utilização desse banco de dados permitira com que as informações do projeto possam ser acessadas por outros aplicativos, desenvolvidos sob medida ou não para ler os dados no banco de dados.

4.5.3.1 Diagrama físico

O diagrama lógico (operacional) do banco de dados pode ser visto na ilustração 22 e seu dicionário de dados descrito na seção 4.5.3.2.

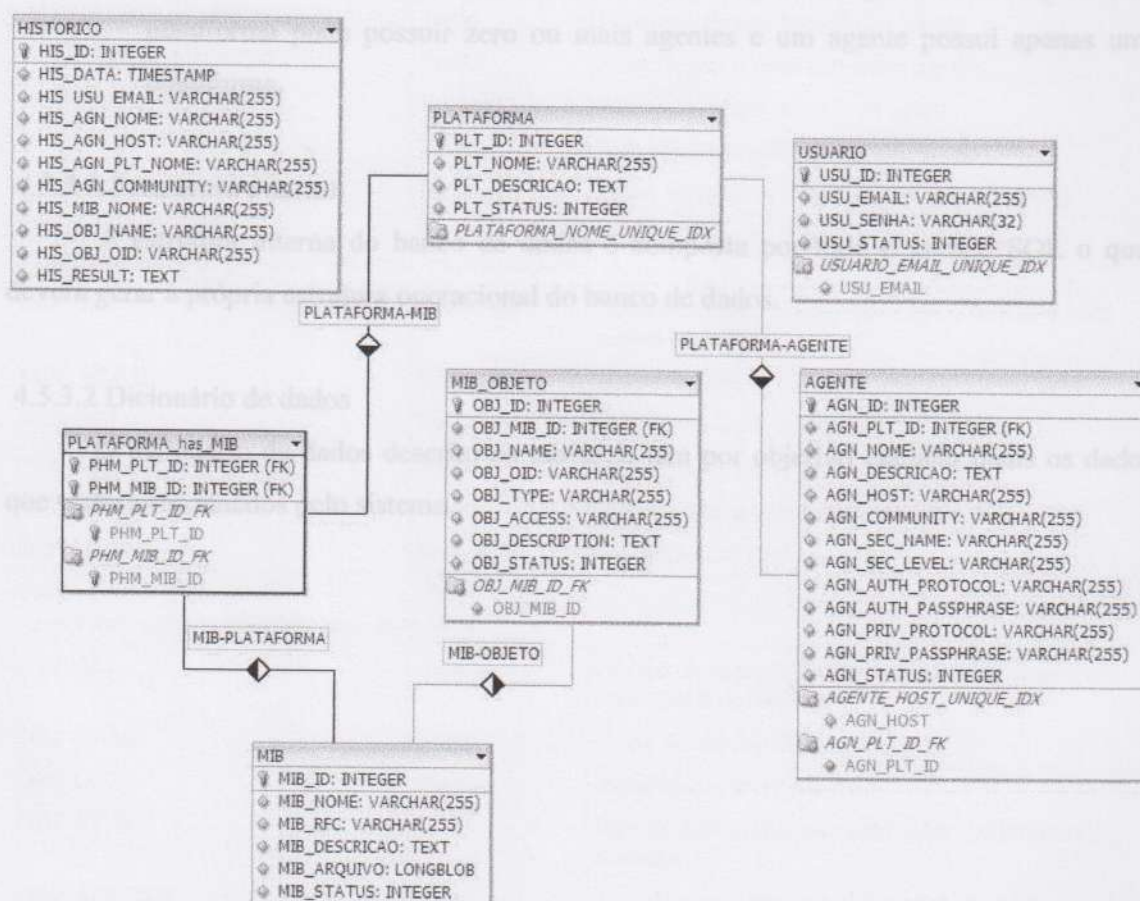


Ilustração 21: Modelo físico do banco de dados
Fonte: Dados da pesquisa (2006)

4.5.3.1.1 relacionamentos

- a) **PLATAFORMA-MIB, MIB-PLATAFORMA:** Estes dois relacionamentos (ver Ilustração 4) esboçam uma cardinalidade de muitas plataformas para muitas mibs, significa dizer que: Uma mib pode possuir zero ou muitas plataformas e que uma plataforma pode possuir zero ou muitas mibs.
- b) **MIB-OBJETO:** Este relacionamento (ver Ilustração 4) esboça uma cardinalidade de uma mib para muitos objetos da mib, significa dizer que: Uma mib pode possuir zero ou mais objetos e um objeto possui apenas uma mib.
- c) **PLATAFORMA-AGENTE:** Este relacionamento (ver Ilustração 4) esboça uma cardinalidade de uma plataforma para muitos agentes, significa dizer que: Uma plataforma pode possuir zero ou mais agentes e um agente possui apenas uma plataforma.

4.5.3.1.2 estrutura interna

A estrutura interna do banco de dados é composta por todo o código SQL o qual deverá gerar a própria estrutura operacional do banco de dados.

4.5.3.2 Dicionário de dados

O dicionário de dados descrito na tabela 5, tem por objetivo explicar quais os dados que serão armazenados pelo sistema.

Tabela 5: Dicionário de dados		
Tabela	Nome	Descrição
MIB STATUS	Númerico	Status da MIB (0 para inativa e 1 para ativa)
Tabela: MIB OBJETO		
Armazena os dados dos objetos de cada MIB.		
OBJ_ID	Númerico	Código do objeto MIB, identificador único para um objeto MIB cadastrado
OBJ_NAME	Texto	Nome do objeto MIB
OBJ_OID	Texto	Identificador de objeto MIB
OBJ_TYPE	Texto	Tipo de dados gerado pelo objeto MIB após a consulta
OBJ_ACCESS	Texto	Tipo de acesso disponível a este objeto MIB
OBJ_DESCRIPTION	Texto	Descrição do objeto MIB
OBJ_STATUS	Númerico	Status do objeto MIB (0 para inativa e 1 para ativa)

Fonte: Dados da pesquisa (2016)

Tabela 27: Dicionário de dados

Tabela: Agente

Armazena informações de agentes da rede, as informações solicitadas para esta tabela são relativa as versões 1, 2 e 3 do protocolo SNMP.

Coluna	Tipo de dados	Descrição
AGN_ID	Númerico	Código do agente, identificador único para um agente cadastrado no banco de dados
AGN_NOME	Texto	Nome do agente
AGN_DESCRICAO	Texto	Descrição do agente
AGN_HOST	Texto	Endereço ou nome do host do agente
AGN_COMMUNITY	Texto	Comunidade do agente
AGN_SEC_NAME	Texto	Nome de segurança do agente
AGN_SEC_LEVEL	Texto	Nível de segurança do agente
AGN_AUTH_PROTOCOL	Texto	Protocolo de autenticação do agente
AGN_AUTH_PASSPHRASE	Texto	Frase passe de autenticação do agente
AGN_PRIV_PROTOCOL	Texto	Protocolo de privacidade do agente
AGN_PRIV_PASSPHRASE	Texto	Frase passe de privacidade do agente
AGN_STATUS	Númerico	Status do agente (0 para inativo e 1 para ativo)

Tabela: MIB

Armazena os dados de cada MIB e seu referido arquivo.

MIB_ID	Númerico	Código da MIB, identificador único para uma MIB cadastrada
MIB_NOME	Texto	Nome da MIB
MIB_RFC	Texto	Documento RFC da MIB
MIB_DESCRICAO	Texto	Descrição da MIB
MIB_ARQUIVO	Texto	Conteúdo do arquivo de estrutura da MIB
MIB_STATUS	Númerico	Status da MIB (0 para inativo e 1 para ativo)

Tabela: MIB OBJETO

Armazena os dados dos objetos de cada MIB.

OBJ_ID	Númerico	Código do objeto MIB, identificador único para um objeto MIB cadastrado
OBJ_NAME	Texto	Nome do objeto MIB
OBJ_OID	Texto	Identificador do objeto MIB
OBJ_TYPE	Texto	Tipo de dados retornado pelo objeto MIB após a consulta
OBJ_ACCESS	Texto	Tipo de acesso disponível a este objeto MIB
OBJ_DESCRIPTION	Texto	Descrição do objeto MIB
OBJ_STATUS	Númerico	Status do objeto MIB (0 para inativo e 1 para ativo)

Fonte: Dados da pesquisa (2006)

Tabela 28: Dicionário de dados (continuação)

Tabela: PLATAFORMA

Armazena os dados básicos sobre as plataformas suportadas pelo sistema e suas respectivas MIBs, entraria no cadastro todo o tipo de plataforma, desde sistemas operacionais a plataformas proprietárias.

PLT_ID	Numérico	Código da plataforma, identificador único para uma plataforma cadastrada
PLT_NOME	Texto	Nome da plataforma
PLT_DESCRICAO	Texto	Descrição da plataforma
PLT_STATUS	Numérico	Status da plataforma (0 para inativa e 1 para ativa)

Tabela: PLATAFORMA_has_MIB

Armazena o relacionamento de muitas plataformas para muitas MIBs.

PHM_PLT_ID	Numérico	Código da plataforma para o relacionamento
PHM_MIB_ID	Numérico	Código da MIB para o relacionamento

Tabela: USUARIO

Armazena apenas os dados necessários para um usuário acessar o sistema.

USU_ID	Numérico	Código do usuário, identificador único para um usuário cadastrado
USU_EMAIL	Texto	E-Mail do usuário
USU_SENHA	Texto	Senha do usuário
USU_STATUS	Numérico	Status do usuário (0 para inativo e 1 para ativo)

Tabela: HISTORICO

Armazena os dados de todas as consultas feitas a um agente utilizando o sistema.

HIS_ID	Numérico	Código do histórico, identificador único para um histórico cadastrado
HIS_DATA	Data/Hora	Data e Hora de cadastro do histórico
HIS_USU_EMAIL	Texto	E-Mail do usuário do histórico
HIS_AGN_NOME	Texto	Nome do agente do histórico
HIS_AGN_HOST	Texto	Endereço ou nome do host do agente do histórico
HIS_AGN_PLT_NOME	Texto	Nome da plataforma do agente do histórico
HIS_AGN_COMMUNITY	Texto	Comunidade do agente do histórico
HIS_MIB_NOME	Texto	Nome da MIB do histórico
HIS_OBJ_NAME	Texto	Nome do objeto da MIB do histórico
HIS_OBJ_OID	Texto	Identificador do objeto da MIB do histórico
HIS_RESULT	Text	Resultado da consulta

Fonte: Dados da pesquisa (2006)

4.6 PLANO DE TESTES

O plano de testes tem por finalidade descrever o planejamento geral das atividades de teste bem como os envolvidos e critérios de aceitação dos artefatos a serem testados. O plano se destina a uma ampla audiência dentro do ciclo de vida do software, dentre os principais envolvidos estão papéis ligados ao desenvolvimento e requisitos do projeto.

4.6.1 Alvos do Teste

Os principais alvos a serem testados contemplados neste plano são:

- **Unidades de código**, onde o código utilizado para implementar é o principal alvo do processo de teste, denominado teste de sistema. Os códigos desenvolvidos em uma linguagem de programação devem estar de acordo com requisitos e projeto do sistema. De maneira geral, qualquer implementação que interaja com o sistema.
- **Sistema completo**, onde nos testes de integração e aceitação o software permitirá o teste da interconexão entre os módulos e a funcionalidade final do software de acordo com os requisitos.

4.6.2 Estratégia de Testes

As estratégias de testes adotadas para este projeto são descritas nas seções subsequentes.

4.6.2.1 Testes de unidade

Os testes de unidades serão realizados pelos desenvolvedores e serão rodados a cada término de uma tarefa de desenvolvimento. Os testes de unidades serão aplicados ao término de cada função implementada para o sistema, afim de estabilizar a utilização da mesma para que em um próximo momento de uso a mesma esteja funcionando adequadamente.

4.6.2.2 Testes de integração

Os testes de integração são constituídos por um conjunto de testes unitários cumulativos e outros construídos para testar especificamente o funcionamento de um componente/circuito individual do projeto após sua integração.

4.6.2.3 Testes de sistema

Os testes de sistema devem focar na interação entre os componentes/circuitos do sistema.

4.6.2.3.1 testes de aceitação do sistema

Os testes de aceitação do sistema devem ser realizados pelo usuário final do sistema após a liberação de cada release, onde o usuário final será convidado a interagir livremente com o sistema a procura de erros ao mesmo tempo em que verifica os requisitos do sistema.

4.6.3 Diretrizes para Execução dos Testes

Para cada iteração deve haver um plano de testes a ser executado, dentre as informações do planejamento devem constar:

- Os testes a serem realizados
- De que forma cada teste será realizado
- A quantidade total de testes a ser realizada

4.6.4 Critérios de Aceitação dos Testes

Os critérios de aceitação de testes base para o projeto, visualizando todas as etapas de testes e visando o produto final é de 100% (cem por cento) de aceitação. É extremamente importante que o projeto satisfaça todos os requisitos e o seu usuário final.

4.6.5 Documentação de Defeitos

Quando forem identificados erros em quaisquer artefatos do software, inclusive códigos, deve ser aberto um chamado na ferramenta de controle de problemas para que o erro fique documentado e o responsável pela parte onde o mesmo foi encontrado seja notificado sobre.

5 CONCLUSÃO

Neste trabalho, foi proposto e especificado o desenvolvimento de uma aplicação baseada em internet, para o monitoramento de agentes SNMP em uma rede, cujo o objetivo principal é facilitar as atividades de gerentes e administradores de redes de computadores. Concluímos neste capítulo que, problemas originados, devido a grande complexidade de uso das ferramentas atuais, serão satisfeitos com a codificação da solução descrita, vimos que as soluções de gerência tradicionais apresentam problemas que impõem sérias limitações ao desenvolvimento das atividades de gerência. Assim, ao levantar os requisitos que caracterizam uma solução mais adequada, fomos capazes de especificar o projeto de uma aplicação de gerência que oferece um conjunto de mecanismos sofisticados para atender os objetivos propostos.

Diante dos requisitos levantados, propomos uma solução que fornece além de reutilização de código, independência de plataforma e integração com a Web, uma grande capacidade de modularização e flexibilidade para adição de novas funcionalidades, bem como um processo para o desenvolvimento de softwares baseados em internet mais robusto, onde os documentos são menos burocráticos e mais diretos, tornando assim toda a atividade mais ágil.

REFERÊNCIAS

ACHOUR, Mehdi et al. **Manual do PHP**. PHP Documentation Group, 2005. Disponível em: <http://www.php.net/manual/pt_BR/>. Acesso em: 04 dez. 2006.

ANÔNIMO. **Segurança Máxima para Linux**. Rio de Janeiro: Campus, 2000.

_____. **Segurança Máxima**. Rio de Janeiro: Campus, 2001.

COMER, Douglas E.; STEVENS, David L. **Interligação em Redes com TCP/IP**. Rio de Janeiro: Campus, 1999.

JUNG, Carlos Fernando. **Metodologia para pesquisa e desenvolvimento**. Rio de Janeiro: Axcel Books, 2004.

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

LOPES, Raquel Vigolvino; SAUVÉ, Jacques Philippe; NICOLLETTI, Pedro Sérgio. **Melhores Práticas para Gerência de Redes de Computadores**. Rio de Janeiro: Campos, 2003.

TANENBAUM, Andrew S. **Redes de Computadores**. Rio de Janeiro: Campus, 2003.

USINA. **United Software Infra-structure Alliance**. Disponível em: <<http://usina.tigris.org/>>. Acesso em: 04 dez. 2006.