

NETDESIGN – FERRAMENTA DE AUXÍLIO PARA AS TAREFAS DE PROJETO, DOCUMENTAÇÃO E GERÊNCIA DE REDES DE COMPUTADORES

Renato Bobsin Machado <mailto:renato@dcc.unioeste-foz.br>

Antonio Marcos Massao Hachisuca <mailto:shiro@dcc.unioeste-foz.br>

UNIOESTE – Universidade Estadual do Oeste do Paraná
Departamento de Ciência da Computação – Centro de Ciências Exatas e Tecnológicas
Campus Foz do Iguaçu
Av. Tancredo Neves, Estrada de acesso à Furnas, Km 1.3
Caixa Postal 961 CEP 85855-000

RESUMO

Motivado pela necessidade provocada pelo rápido avanço tecnológico, o *NetDesign* é um sistema que foi desenvolvido para a automatização de processos relacionados com a documentação, administração, gerência e elaboração de projetos físicos e lógicos de redes de computadores.

Para a definição do escopo do projeto estudou-se modelos de gerência de redes e ferramentas similares.

No intuito de debelar carências identificadas e necessidades impostas, o *NetDesign* documenta redes geograficamente distribuídas e provê mecanismos de gerência nas áreas funcionais de falha e configuração de redes.

Os resultados alcançados demonstram que as características implementadas atendem a demanda da maioria das redes hoje implantadas e que deve ser dada continuidade ao projeto para se agregar funcionalidades, adequando a ferramenta aos requisitos atuais e emergentes.

ABSTRACT

Motivated by the needs caused by the technology's rapid advance, NetDesign is a system that was developed to automate the documentation, administration and management processes and to build physical and logical projects.

Network management models and similar tools were studied to define the project's goals.

The NetDesign documents geographically distributed networks and provides management over failure control and network setup, filling in the gap from the lack of such tools.

The achieved results show that the features that were implemented in the project meet the requirement of most networks that are in use nowadays and that the project must be continued to add new features, upgrading the system to satisfy the requirements of the present and future technological context.

INTRODUÇÃO

A rápida evolução tecnológica tem permitido um aumento no número de computadores na sociedade. A partir dos anos 70 surgiram as redes distribuídas que evoluíram de uma apresentação simples com mesma tecnologia e pouca complexidade para um ambiente de redes LAN (Local Area Network), MAN (Metropolitan Area Network) e WAN (Wide Area Network) interligadas [MAC 98].

Esse desenvolvimento tem feito emergir diversas carências por ferramentas aptas a funções de documentação, administração e gerência de redes de computadores; fator que estimula universidades e empresas a trabalhar utilitários para estes propósitos através da utilização de um padrão aberto.

Dentro deste contexto desenvolveu-se o *NetDesign*, que foi projetado para ser uma ferramenta de apoio para a construção de projetos físicos e lógicos de redes de computadores, com ênfase aos processos de supervisão, monitoração e controle.

O presente artigo tem por objetivo apresentar a ferramenta *NetDesign*, salientando sua especificação de projeto, funcionalidades, operacionalização, estratégia de testes, resultados obtidos e conclusões.

1. DEFINIÇÃO DO PROBLEMA

A necessidade de softwares que auxiliem na construção de projetos físicos e lógicos de redes de computadores, assim como a atividade de monitorar, proteger e administrar redes com grande número de estações constitui um caso prático com soluções não triviais.

O NetDesign é um projeto de pesquisa, cuja primeira versão ataca dois problemas característicos: primeiro apresenta de forma visual e em bases de dados toda a documentação de uma rede distribuída, envolvendo tecnologia, periféricos, localização física, serviços, etc de uma rede distribuída. A outra função do sistema é facilitar o processo de monitoração de redes Internet através do protocolo ICMP, disparando alarmes administrativos, registrando a ocorrência dos principais eventos ocorridos e apresentando visualmente o status de toda a rede.

O problema descrito pode crescer muito face ao tamanho e características de sistemas distribuídos, nesse sentido o NetDesign objetiva criar diversas linhas de pesquisa para o atendimento à questões como gerenciamento distribuído e segurança de redes.

A seção seguinte fará uma explanação do projeto de sistema e de arquitetura do NetDesign.

2. ESPECIFICAÇÃO DO PROJETO NETDESIGN

Para a definição de requisitos funcionais, arquitetura e características técnicas do projeto, estudou-se conceitos concernentes a tecnologia de redes de computadores [BRI94] [COM95-1] [TAN94].

Em adicional levantou-se os principais equipamentos e formas de interconexão [DER93] [COM93-2] [COM5-3]. Como complemento substancial pesquisou-se modelos, formas e implementações a respeito da gerência de redes de computadores [MAC 98] [TAR 95] [TAR 96] [TAR 97] [STA 97].

Além do estudo teórico, avaliou-se duas ferramentas com propósitos similares ao NetDesign, o What's Up Gold e o TK-Inet, no intuito de levantar suas carências, deficiências, vantagens e desvantagens; consistindo fator determinístico à definição do escopo do projeto.

Esta seção tem por premissa apresentar decisões de projeto relativas a requisitos funcionais, tecnologia utilizada, arquitetura de classes, relacionamentos e características elementares do sistema.

2.1 Definição de requisitos funcionais

Os requisitos funcionais abrangeram duas áreas centrais, sendo uma voltada para documentação e projeto e outra para gerência de configuração e de falha.

2.1.1 Documentação e projeto

Para o atendimento às necessidades de projeto e documentação definiu-se os seguintes requisitos de software:

- Registro da rede de forma a permitir uma visualização de sua distribuição geográfica;
- Permissão para se ter registro completo dos componentes que integram cada equipamento, incluindo sua tecnologia e periféricos;
- Implementação de mecanismos para cadastro de novas tecnologias que surjam no mercado;
- Utilização de ícones para representar os diferentes elementos de rede, podendo o usuário adequá-los a visualizações que melhor representem a realidade do mercado.

2.1.2 Gerência de Configuração e de Falhas

Na área de gerência de redes, foram levantadas necessidades a partir de pesquisas sobre ferramentas similares, entre as quais destaca-se:

- Verificação de estados dos equipamentos através do protocolo ICMP;
- Representação gráfica de cada elemento, indicando seu *status* (*Up* ou *Down*) através de ícones diferenciados e configuráveis pelos usuários;
- Registro em base de dados das mudanças de estado ocorridas em cada objeto;
- Desenvolvimento de interface para a definição de quais equipamentos efetivamente se pretende gerenciar;
- Implementação de funções sobre objetos, tais como: *Ping*, *Telnet* e *Traceroute*;
- Envio de notificação sempre que algum elemento gerenciável deixar de responder;

2.2 Tecnologia Utilizada

Para atender aos objetivos propostos, optou-se pela utilização de recursos aptos a facilitar o processo de desenvolvimento, os quais vale destacar:

- Linguagem Delphi: A escolha da linguagem se deu face a flexibilidade da mesma para o trabalho com interface e com arquivo de objetos;
- Banco de Dados: A utilização do banco de dados objetivou o cadastro da tecnologia utilizada pelos equipamentos, facilitando o registro de novos modelos, fabricantes e a evolução de suas características técnicas;
- Classe *TStream*: Trata-se de um componente da linguagem Delphi voltada para o trabalho com arquivos de objetos, possibilitando que se salve e recupere objetos inteiros do disco.

2.3 Arquitetura de Classes

As classes do NetDesign foram projetadas com o propósito de implementar os requisitos definidos nos itens 1.1.1 e 1.1.2.

Dado a abrangência do projeto, estipulou-se categorias de classes, as quais serão explanadas a seguir.

2.3.1 Categoria de Classes Geográficas

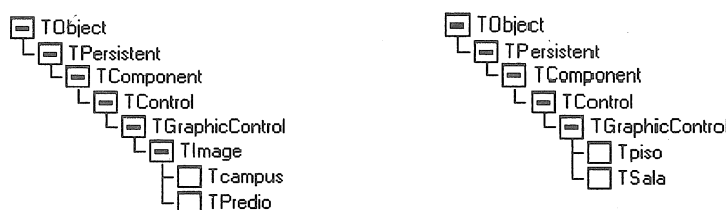
Esta categoria foi projetada no intuito de prover componentes visuais para representar a distribuição geográfica de redes distribuídas (campus, prédios, pisos e salas de equipamentos).

Campus e Prédios são implementados pelas classes *TCampus* e *TPredio* para uma visão de alto nível dos segmentos de rede existentes. As propriedades visuais destas classes podem ser alteradas a qualquer tempo para melhor representar as particularidades de cada instituição, como por exemplo colocar fotos de prédios da organização em seus ícones.

Ambas as classes permitem a configuração de *links* para outras estruturas geográficas. Para atender a esses objetivos, *TCampus* e *TPredio* herdam *TImage* na estrutura da linguagem de programação utilizada, conforme mostra a figura 1a.

O componente Piso é implementado pela classe *TPiso* e é composto por salas de equipamentos. Salas são implementadas pela classe *TSala*, representando o último nível hierárquico.

As características visuais de *TPiso* e *TSala* se restringem a espessura e cores de paredes, fundos, etc. As duas classes herdam *TGraphicControl* na estrutura da linguagem Delphi, conforme mostra a figura 1b.



a) Estrutura de Classes *TCampus* e *TPredio*

b) Estrutura de Classes *TPredio* e *TPiso*

Figura 1 – Árvore de Objetos para categoria Geográficos

2.3.2 Categoria de Classes Periféricos

Nesta categoria encontram-se as classes que documentam periféricos. Para tanto implementa-se uma classe genérica, *TPeriféricos_Comum*, que contém propriedades inerentes a todos os elementos desta categoria, tais como: Fabricante, Modelo, Descrição.

Atualmente estão implementados os periféricos Disco, Memória, Monitor, CD-Rom, FitaDat e Processador através das classes *TDisco*, *TMemória*, *TMonitor*, *TCD-Rom*, *TFitaDat*, *TProcessador*. Cada uma destas classes encapsula dados técnicos de cada tipo de equipamento.

Ainda nesta categoria tem-se a classe *TPorta*, que é responsável pelo registro de endereços IP, possuindo as seguintes propriedades centrais: endereço IP, máscara de endereçamento, nome (para equipamentos DHCP), tecnologia (características da conexão de rede)

A estrutura de herança da categoria periféricos é apresentada na figura a seguir:

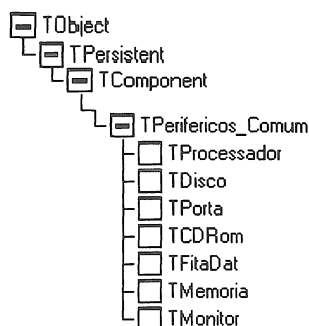


Figura 2 – Classes do grupo periféricos

2.3.3 Categoria de Classes de Equipamentos

Esta categoria implementa os objetos gerenciáveis. Em primeira instância definiu-se uma classe genérica, *TEquipamentos*, que contém propriedades e métodos comuns a todos os objetos gerenciáveis. Entre suas características essenciais cita-se:

- Propriedades Visuais: Possui ícone para representar os estados *Down* e *Up* dos objetos, assim como métodos para alterar esses ícones.
- Propriedades de Documentação: Definiu-se dados relativos a fabricantes e modelos.
- Propriedades de Gerência: Provê métodos para a atribuição de endereços IP, sem limite numérico, sendo compatível com os mais diversos meios de interconectividade existentes. Esses endereços IP são objetos da classe *TPorta*.

A partir desta classe principal, implementou-se classes para a representação dos seguintes equipamentos: Computador, Hub, Switch, Roteador, Impressora e Plotter. Cada uma dessas classes possui propriedades e métodos para o atendimento de suas particularidades. A figura 3 mostra a estrutura de herança dessas classes.

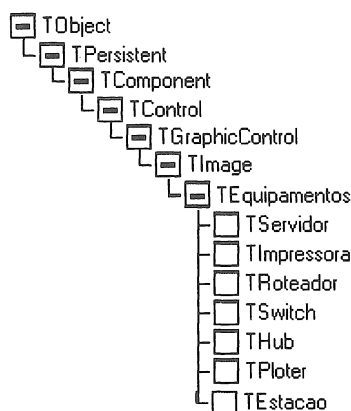


Figura 3 – Classes de Objetos para representar equipamentos

2.3.4 Categoria de objetos para Gerência

A arquitetura do sistema de gerência baseia-se na utilização do protocolo ICMP. Os objetos com propósito de gerenciamento são definidos através da adição de portas IP. Esse gerenciamento é feito através do envio de *Pings* a todos os elementos, sendo os ícones atualizados em função da resposta ICMP, assim como o registro de eventos em *log* e o envio de notificações.

A seguir faz-se uma breve descrição das classes que compõe o sistema de gerência:

- TICMP:** A classe *TICMP* implementa o protocolo ICMP (camada de rede do modelo TCP-IP), definindo códigos e mensagens de erros, tamanho do pacote a ser enviado a um IP remoto, tempo de vida (TTL) do pacote, Nome do Host, número IP e máscara da sub-rede.

Além da estrutura de dados, define-se métodos para tratar esses dados, envio e recepção de pacotes, resolução de nomes e tratamento de erros.

Como eventos criou-se:

- *OnEchoRequest*: Envia pacotes via *socket*, requisitando que o mesmo pacote seja retornado (*echo*).
- *OnEchoReply*: Consiste no retorno do pacote que havia sido enviado a um destino. A partir dessa resposta que são tratados os códigos de retorno da operação.

- b) *TPing*: A presente classe é responsável pela interface entre uma porta de gerenciamento e o protocolo *ICMP*. Para isso define-se alguns métodos importantes, como o “*execute*”, que é responsável por captar dados de gerenciamento de uma porta IP e iniciar a operação do protocolo.

Para a atualização de dados referentes a resposta do protocolo, associa-se eventos locais com o *TICMP* para traduzir os códigos recebidos a uma estrutura compatível com a classe *TPorta*.

- c) *TKernel*: A classe *TKernel* consiste em uma *Thread* que é iniciada para cada tela de gerenciamento configurada. Ela recebe uma lista com todos os equipamentos gerenciáveis desta tela e faz uma lista circular para a gerência de cada porta. É também responsável por tratar a atualização gráfica de cada equipamento, a fim de apresentar as alterações de estado sofridas pelos objetos.

2.3.5 Classe TTela

A classe *TTela* não se enquadra em nenhuma das categorias acima e é responsável pela administração do sistema. Cada tela é uma instância de *TTela* e representa uma rede, sendo visualizados graficamente todas as instância de equipamentos.

Para o cumprimento desta funcionalidade, a classe em questão mantém registro dos objetos por categoria e possui uma instância de *TKernel* para ativar e desativar o gerenciamento desta rede.

2.4 Relacionamentos entre as Classes

A figura 5, na próxima página, demonstra a disposição das categorias de componentes e como ocorre a interação entre as classes.

Em termos gerais pode-se dizer que *TTela* é o elemento que faz o vínculo entre as características visuais e funcionais do projeto.

As instâncias de *TTela* representam redes individuais ou uma visualização geográfica de uma rede mais ampla. Por conseguinte, todas as instâncias das categorias geográfica e equipamentos são parte integrante de uma instância *TTela*.

Para administrar todos esses objetos e prover mecanismos de gerência, *TTela* possui uma instância de *TKernel* para que possa iniciar e finalizar a atividade de gerência.

Os objetos da categoria periféricos por sua vez, são sempre atribuídas a uma instância da categoria Equipamentos. O desenvolvimento desta categoria foi com vista a possibilitar uma documentação completa dos diferentes elementos gerenciáveis do sistema e possibilitar que se mantenha essa estrutura para futuras implementações de outras classes e para a representação de novas tecnologias.

A presente seção demonstrou como foi especificado o projeto de sistema do NetDesign, enfatizando seu escopo, requisitos funcionais, arquitetura de classes e seus relacionamentos.

A seção seguinte apresentará as principais características operacionais do NetDesign.

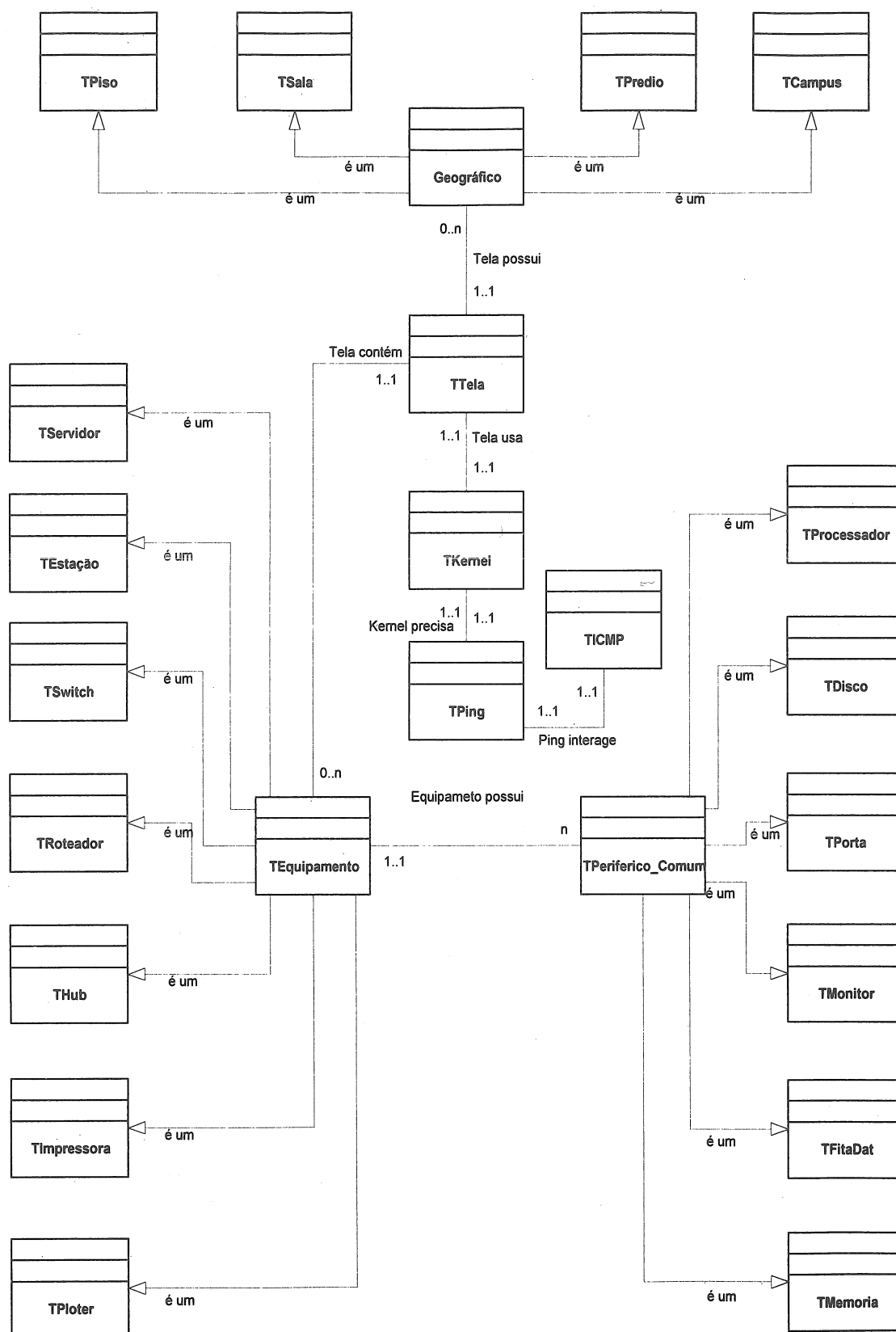
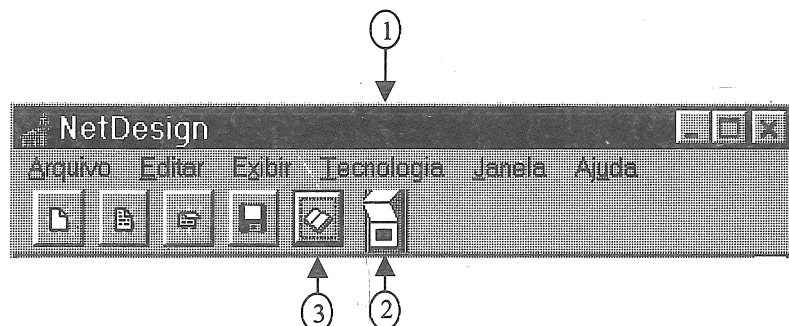


Figura 5 – Diagrama de Classes do NetDesign

3. Características Operacionais do NetDesign

Esta seção tem por meta mostrar a forma de operacionalização do NetDesign, no que tange a suas características fundamentais.

Ao carregar o sistema, é disponibilizado uma barra de ferramentas e um menu com as funcionalidades do sistema, conforme a figura abaixo.



1. *Menu Tecnologia* – Responsável pela atualização de dados sobre tecnologia de periféricos.
2. *Botão Switch* – Ativa e desativa o sistema de gerência.
3. *Botão Ferramentas* – Ativa e desativa a barra de ferramentas.

Figura 4 – Tela inicial do NetDesign

Ao ativar o botão de ferramentas tornam-se disponíveis as opções para inserção de objetos geográficos, equipamentos gerenciáveis e mídias de comunicação. Esta barra é apresentada na figura 6.

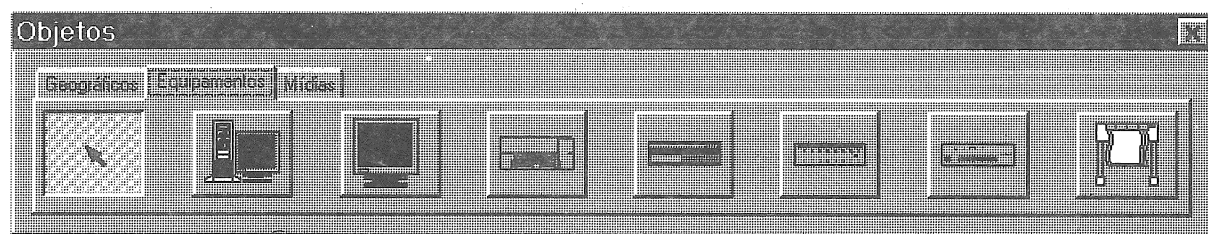
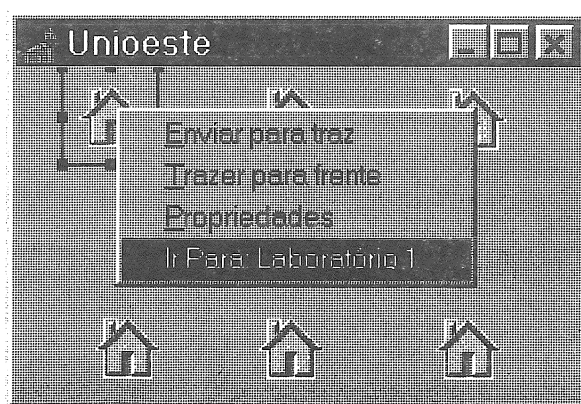


Figura 6 – Barra de Ferramentas do NetDesign

O sistema propicia um ambiente no qual se pode definir níveis geográficos através da representação de *campus*, *prédio*, *piso* e *sala*. A esses componentes associam-se *links* para outras telas, sendo possível mapear visões sucessivas até se chegar a unidade mínima onde estão dispostos os equipamentos propriamente ditos. A figura 7 mostra respectivamente uma visão desde um mapeamento de campus até a representação de um laboratório.



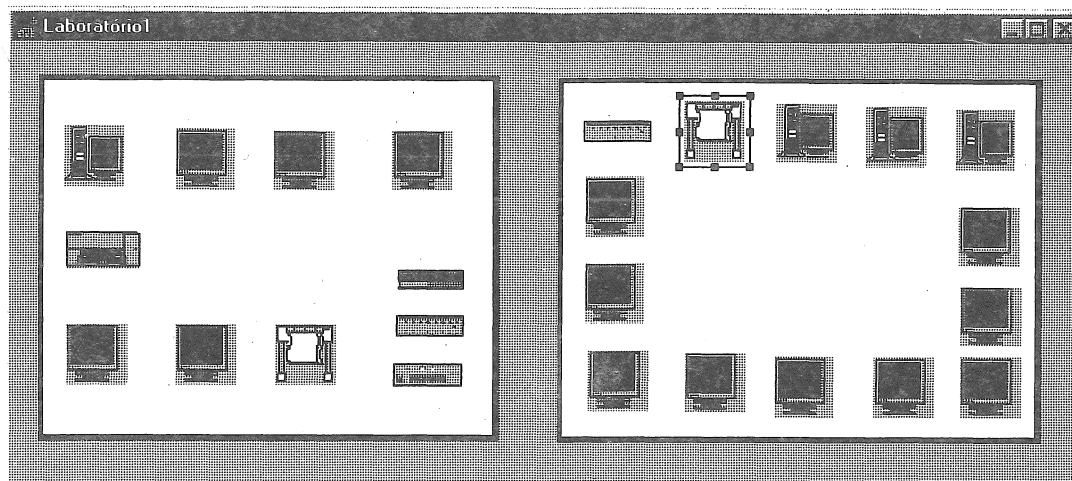


Figura 7 – Visualização Geográfica de uma rede distribuída

A cada elemento inserido (geográfico ou gerenciável), tem-se um *popup menu* que apresenta as atividades que se pode realizar com o objeto clicado. Todos os componentes possuem a opção *Propriedades* que implementa as funções de configuração, tanto de características visuais, quanto técnicas e de documentação.

Todas as opções de características de objetos são definidas através da disponibilização de alternativas. No entanto permite-se a atualização destas informações para possibilitar a adaptação a novas tecnologias através do menu *tecnologia*.

A cada inserção de objeto abre-se uma tela de configuração, cujas propriedades variam em função da classe a qual o objeto pertence.

A figura 8 mostra a configuração de um computador, onde se pode definir propriedades visuais (ícones para representar os estados *down* e *up*, dimensões do ícone), técnicas (nome, descrição, barramento) e se pode ativar ou desativar o gerenciamento deste equipamento. As opções especiais permitem documentar periféricos, processadores e portas de endereçamento.



Figura 8 – Visualização da Tela de Edição de Propriedades de um Computador

Todos os objetos gerenciáveis possuem uma ou mais portas de endereçamento IP associado, em que se pode definir a tecnologia implementada, o nome DHCP e o número IP quando houver.

A figura 9 apresenta a tela de configuração de uma porta gerenciável.

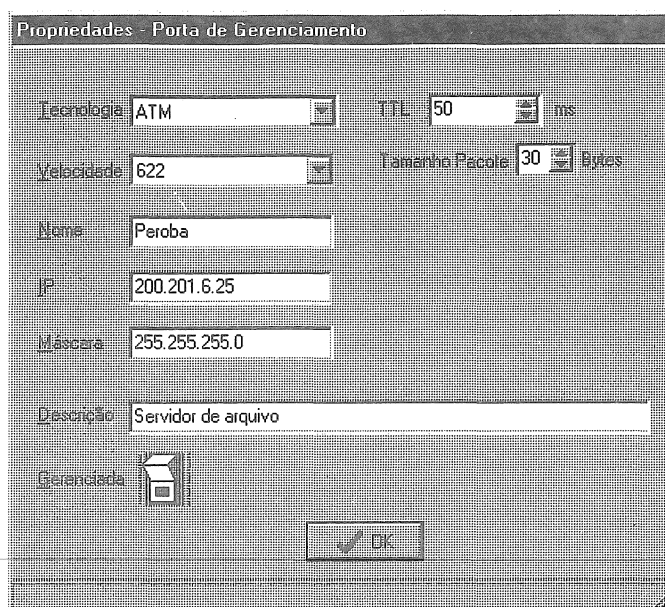


Figura 9 - Configuração de uma porta gerenciável

Os objetos instanciados de Computador tem a possibilidade de inserir outros componentes para documentar os periféricos que possui, conforme a figura 10.

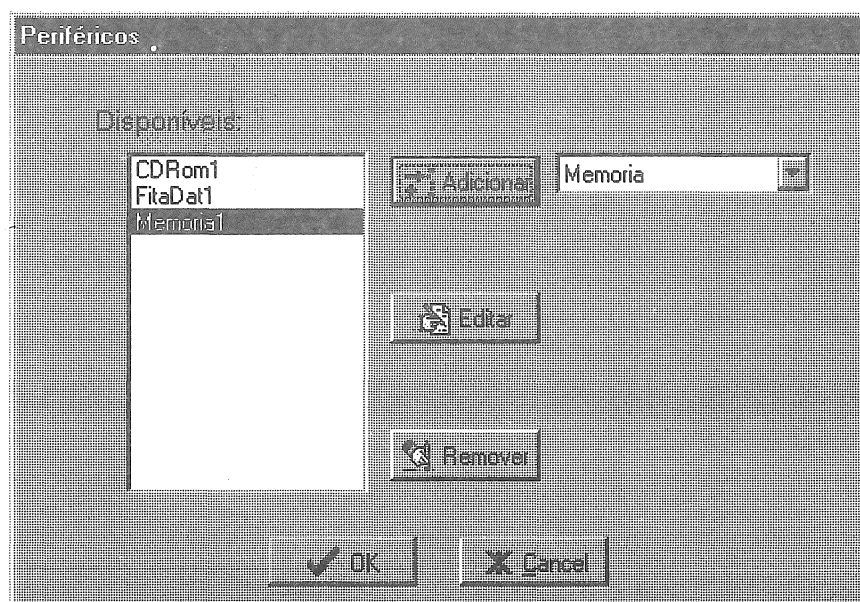


Figura 10 - Registro de Periféricos

Uma vez estruturada a rede, pode-se ativar a gerência, onde o sistema passa a verificar que equipamentos estão ou não ativos no momento. Esses resultados são demonstrados graficamente através de ícones diferenciados e os principais eventos são guardados em bases de dados.

Associado a cada equipamento são disponibilizados alguns serviços, como *Ping*, *Telnet* e *Tracerout*. A tela abaixo demonstra a utilização do serviço *Tracerout*.

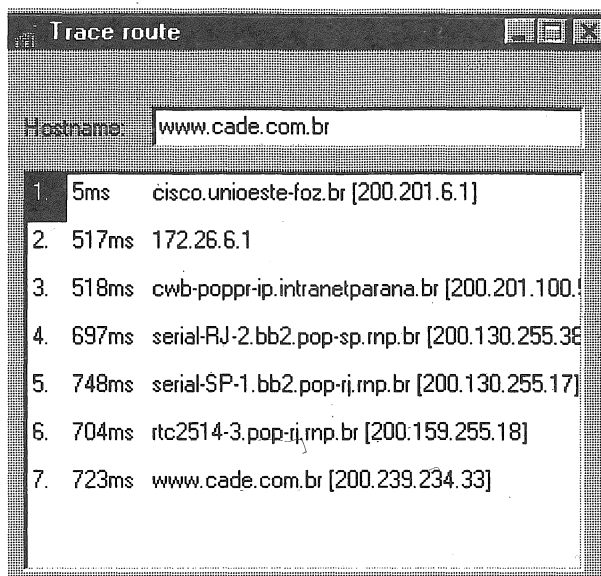


Figura 12 – Execução de uma busca *Tracerout*

4. METODOLOGIA DE TESTES

A metodologia de testes foi definida conforme o modelo espiral proposto por [PRE 97], onde valida-se progressivamente desde a classe individual até o sistema de forma integrada.

4.1 Testes de Unidade

Os testes de unidade foram realizados ao término da implementação de cada classe no intuito de avaliar o grau de eficácia com que as propriedades e métodos atendiam a seus requisitos funcionais. Os resultados geraram necessidades de revisão estrutural e de codificação em algumas classes.

O principal proveito destes testes foi a possibilidade de se reavaliar características de projeto, em função do aumento do conhecimento sobre o problema.

4.2 Testes de Integração

Os testes de integração foram utilizados no intuito de verificar o inter-relacionamento entre as classes já validadas.

Estes testes foram realizados através da simulação de execução das funcionalidades propostas, descritas abaixo:

- a) Cadastro de tecnologia;
- b) Funcionamento do editor gráfico de objetos.
- c) Agregação de periféricos, processadores e portas gerenciáveis aos equipamentos.
- d) Gravação e abertura de arquivos de objetos.
- e) Utilização das classes de gerência interrelacionada com propriedades dos equipamentos gerenciáveis.

Toda esta gama de testes gerou uma série de implicações de projeto, como a necessidade da utilização de novas classes, particionamento de classes e alteração na estrutura de herança.

Mais uma vez, além da descoberta de erros de projeto e codificação, os testes possibilitaram um ganho de conhecimento sobre o escopo do problema, que resultou em remodelagens de algumas classes.

4.3 Teste de Validação

Por ser um sistema com fins acadêmicos e científicos, os testes de validação foram adaptados a aprovação dos requisitos propostos por intermédio de reuniões da equipe de desenvolvimento com o orientador.

4.5 Teste de Sistema

Uma vez que se tinha aprovado a interface entre as classes e as funcionalidades implementadas, o teste de sistema teve por função avaliar a atuação do software em seu ambiente operacional. Para tanto testou-se toda a questão de projeto e documentação da rede utilizando o editor gráfico.

Para o teste de gerência utilizou-se uma máquina conectada na rede do departamento de computação da Universidade com a configuração de elementos gerenciáveis válidos. Então desligou-se alguns destes elementos gerenciáveis para verificar o funcionamento do módulo de gerência e as mudanças de interface resultantes. Além desses testes utilizou-se testes de limites e stress[PRE 97].

Como resultado dos primeiros testes, descobriu-se algumas falhas de sincronização entre os *poolings* e a atualização da interface.

A figura 13, na página seguinte, representa uma rede gerenciada depois que algumas máquinas foram desligadas. Nesse contexto, ícones com ponto de interrogação significam máquinas que não estão respondendo.

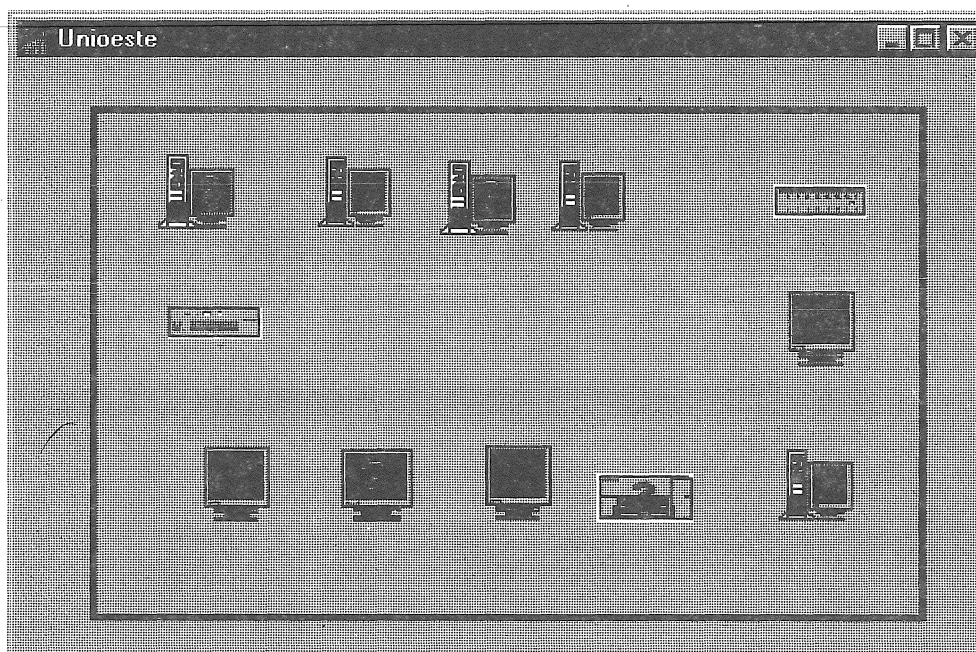


Figura 13 - Teste de sistema

5. RESULTADOS OBTIDOS

Os testes de campo realizados nas redes do Departamento de Computação da Unioeste (Universidade Estadual do Oeste do Paraná) demonstraram que os requisitos funcionais implementados atenderam plenamente às funcionalidades definidas no escopo do projeto, tornando-se hábil ao suporte as macro atividades de elaboração de projetos físicos e lógicos, documentação, administração e gerência de redes de computadores.

No que concerne a parte de projeto, conseguiu-se documentar toda a estrutura de uma rede distribuída, registrando a configuração de rede, de periféricos e de tecnologia de todos os equipamentos da rede.

Na área de gerência, conseguiu-se: monitorar a rede distribuída de forma on-line através do ambiente gráfico, identificar problemas ocorridos através dos alarmes administrativos e teve-se completo suporte à informações sobre o sistema através dos logs gerados.

A seguir apresenta-se as inovações implementadas pelo NetDesign em relação a ferramentas similares:

- a) Vinculação de mais de um endereço IP ao objeto gerenciado.
- b) Mecanismos para documentação geográfica.
- c) Facilidades para documentação de equipamentos.
- d) Noções sobre a tecnologia empregada.
- e) Registro de eventos em base de dados.
- f) Suporte a relatórios estatísticos.

6. CONCLUSÃO

O avanço tecnológico computacional fez aumentar a complexidade e dificultar a operabilidade das redes de computadores, tendo por consequência a criação de necessidades pelo desenvolvimento de ferramentas de apoio.

Dentro deste contexto, o NetDesign embasado em pesquisas sobre tecnologia de redes, modelos de gerência e pelo estudo de ferramentas similares, conseguiu implementar funcionalidades adicionais com vista a facilitar a construção de projetos físicos e lógicos de redes de computadores.

Os testes e aplicações desta primeira versão do sistema validam os objetivos iniciais, implementando com sucesso os requisitos funcionais definidos no escopo do projeto, tanto no que se refere a documentação, quanto em termos de funcionalidades de gerência e administração de redes distribuídas.

Os resultados obtidos motivam a continuação do projeto, visando agregar novas funcionalidades por intermédio da criação de linhas de pesquisa abrangendo as áreas de projetos, segurança, monitoração e supervisão de redes. Entre as pesquisas já iniciadas cita-se a incorporação de uma camada plataforma CORBA para a comunicação do sistema, a substituição da atual arquitetura por uma estrutura de comunicação *Master/Slave* e a incorporação de multiagentes integrando as duas tecnologias anteriormente citadas.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- [BRI 94] BRISA. Sociedade brasileira para interconexão de sistemas abertos. *Arquitetura de redes de computadores OSI e TCP/IP*. Organizada e coordenada por Tereza Cristina Melo de Brito Carvalho. São Paulo: Makron Books; Rio de Janeiro: EMBRATEL, 1994.
- [COM 95-1] COMER, Douglas E. *Internetworking with TCP-IP*. 3. ed. New Jersey: Prentice Hall, 1995. 1v. V.1: Principles, protocols, and architecture.
- [COM 93-2] COMER, Douglas E., STEVENS, David L. *Internetworking with TCP-IP*. 2. ed. New Jersey: Prentice Hall, 1993. 2v. V.2: Design, implementation, and internals.
- [COM 93-3] _____. New Jersey: Prentice Hall, 1993. 3v. V.3: Client-Server programming and applications BSD socket version.
- [DER 93] DERFLER Jr., Frank J. *Guia para a interligação de redes locais*. Traduzida por Fernando Barcellos Ximenes. Rio de Janeiro: Campus, 1993. Tradução de: Guide to linking LANs.
- [MAC 98] MACHADO, Nelson. *Gerência de redes de computadores*. Dissertação (Especialização em Engenharia de Software) - Convênio Universidade Federal do Paraná / Universidade Estadual do Oeste do Paraná / ITAIPU, 1998.
- [PRE 97] PRESSMAN, Roger S. *Software Engineering: a practitioner's approach*. 4. ed. Singapore: McGraw-Hill Book, 1997.
- [TAN 97] TANENBAUM, Andrew S. *Redes de Computadores*. 3. ed. Traduzida por Insight Serviços de Informática. Rio de Janeiro: Campus, 1997. Tradução de: Computer Networks.
- [TAR 95] TAROUCO, Liane. *Gerenciamento de redes: SNMP x CMIP*. In: EXPONET'95, 1995, São Paulo.
- [TAR 96] _____. *Como gerenciar sua rede usando RMON2*. In: EXPONET'96, 1996, São Paulo.
- [TAR 97] _____. *Análise e soluções avançadas na gerência de redes*. In: EXPONET'97, 1997, São Paulo.
- [STA 97] STALLINGS, Willian. *SNMP, SNMPv2, and RMON: practical network management*. 2. ed. Massachusetts: Addison Wesley, 1997.

