

INSTITUTO DE PESQUISAS TECNOLÓGICAS DO ESTADO DE SÃO PAULO

MAURO KENDI NODA

Ferramenta para Monitoração Remota de Variáveis Ambientais de  
Laboratório

**São Paulo**

**2006**

---

**MAURO KENDI NODA**

**FERRAMENTA PARA MONITORAÇÃO REMOTA DE VARIÁVEIS  
AMBIENTAIS DE LABORATÓRIO**

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, para obtenção do título de Mestre em Engenharia de Computação.

Área de concentração: Engenharia de *Software*.

Orientador: Dr. Marco Dimas Gubitoso

São Paulo

2006

---

Ficha Catalográfica  
Elaborada pelo Departamento de Acervo e Informação Tecnológica – DAIT  
do Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT

N761f    Noda, Mauro Kendi

Ferramenta para monitoração remota de variáveis ambientais de laboratório. / Mauro  
Kendi Noda. São Paulo, 2006.  
49p.

Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas  
Tecnológicas do Estado de São Paulo. Área de concentração: Engenharia de  
Software.

Orientador: Prof. Dr. Marco Dimas Gubitoso

1. Automação de laboratório 2. Processo de ensaios 3. Calibração 4. Software  
5. Hardware 6. Tese I. Instituto de Pesquisas Tecnológicas do Estado de São Paulo.  
Coordenadoria de Ensino Tecnológico II. Título

07-68

CDU 004.41:061.62(043)

Dedico esta dissertação à minha esposa Asako e  
as minhas filhas Nadia, Nicole e Nathalie pelo  
constante incentivo e apoio.

---

## Agradecimentos

Agradeço a todo o corpo docente do Programa de Mestrado em Engenharia da Computação do Instituto de Pesquisas Tecnológicas de São Paulo, que possibilitou a mim um grande crescimento pessoal.

A comunidade de desenvolvedores de software de código aberto, sem o qual o desenvolvimento deste trabalho não seria possível.

Aos professores Agenor T. Fleury, Eduardo A. Tannuri e Marco D. Gubitoso, componentes da banca, pela contribuição na avaliação do trabalho e valiosos conselhos e opiniões.

A todos que de alguma forma contribuíram ou incentivaram meu trabalho, seja pela orientação e compartilhamento de conhecimentos, pelo apoio nas diversas atividades do trabalho, ou mesmo pela simples amizade e motivação.

---

## Resumo

A aplicação de sistemas computadorizados para aquisição de dados e controle de laboratório tornou-se uma necessidade para os laboratórios de ensaio e calibração, devido a diversos fatores econômicos e conjunturais.

Pelas exigências dos órgãos certificadores para que a temperatura e umidade do ambiente sejam controladas e/ou monitorados durante a realização de ensaios e calibrações, verificou-se a necessidade de um sistema automatizado para controlar e monitorar o sistema.

Devido à aquisição e controle estarem sendo feitas em um computador específico é necessário que os dados e controle do tempo de aquisição possam ser alterados e monitorados através de outro computador por meio de uma rede (Intranet, Internet), para possibilitar o controle e a monitoração em tempo real das medidas realizadas e possíveis correções no intervalo de aquisição dessas medidas, podendo estas ser acessadas de qualquer ponto da rede.

Verifica-se, então, a necessidade adicional do acesso aos dados de ambiente através de outros computadores do laboratório e também o histórico dessas medidas num determinado espaço de tempo requerido para efetuar a elaboração do relatório e também se certificar que o procedimento está sendo realizado dentro das exigências da norma do ensaio ou calibração.

Desta maneira, o objetivo do trabalho foi o projeto e desenvolvimento de uma ferramenta remota e descentralizada de monitoração e controle de variáveis ambientais de laboratório, onde todo o sistema pode ser projetado e desenvolvido com ferramentas *Open Source*.

Palavras-chave: Automação de laboratório, Processo de ensaios, Calibração, Software.

---

## ABSTRACT

The application of computerized systems for acquisition of data and control of laboratory became a necessity for the assay and calibration laboratories, because of several economic and conjuncture factors.

Because of the requirements of the certifying agencies, the temperature and humidity of the environment must be controlled and/or monitored during the accomplishment of assays and calibrations, it was verified the necessity of an automatized system for the accomplishment of this control.

Because the acquisition and control are done in a specific computer it is necessary that the data and control of the acquisition time can be modified and be monitored through another computer by means of a net (Intranet, Internet), in order to be possible to control and monitor the system in real time of the carried out measurements and possible corrections in the interval of acquisition of these measurements, being able to access these from any point of the net.

It is verified, then, the additional necessity of accessing the environment data through other computers of the laboratory and also to certify themselves that the procedure is being carried out following the requirements of the norm of the assay or calibration.

In this way, the objective of the work was the project and development of a remote and decentralized tool to monitor and control the laboratory's variable environment, being that all the system can be projected and developed with Open Source tools.

Keywords: Automation of laboratory, Process of assays, Calibration, Software.

---

## Lista de Ilustrações

FIGURA 1	ARQUITETURA DO SERVIDOR RMI. ....	20
FIGURA 2	MONTAGEM EXPERIMENTAL .....	22
FIGURA 3	MODELAGEM DO DOMÍNIO DO SISTEMA.....	26
FIGURA 4	ÍNDICE DA REPRESENTAÇÃO DAS FIGURAS DO DIAGRAMA UML.....	27
FIGURA 5	ANÁLISE DE ROBUSTEZ DO INÍCIO DE AQUISIÇÃO. ....	28
FIGURA 6	ANÁLISE DE ROBUSTEZ DA ALTERAÇÃO DE TAXA DE AQUISIÇÃO DO CRIADOR DE PROCESSOS. ....	29
FIGURA 7	ANÁLISE DE ROBUSTEZ DA ALTERAÇÃO DE TAXA DE AQUISIÇÃO DO PROCESSO LOCAL. ....	29
FIGURA 8	ANÁLISE DE ROBUSTEZ DA ALTERAÇÃO DE TAXA DE AQUISIÇÃO DO PROCESSO VIA WEB. ....	30
FIGURA 9	ANÁLISE DE ROBUSTEZ DA OBTENÇÃO DE TEMPERATURA MÉDIA, MÁXIMA E MÍNIMA E UM GRÁFICO TEMPORAL DE UM DETERMINADO PERÍODO. ....	31
FIGURA 10	ANÁLISE DE ROBUSTEZ DO CICLO DE AQUISIÇÃO.....	32
FIGURA 11	MODELAGEM DE INTERAÇÃO DO INÍCIO DE AQUISIÇÃO.....	33
FIGURA 12	MODELAGEM DE INTERAÇÃO DA ALTERAÇÃO DE TAXA DE AQUISIÇÃO DO CRIADOR DE PROCESSOS.....	34
FIGURA 13	MODELAGEM DE INTERAÇÃO DA ALTERAÇÃO DE TAXA DE AQUISIÇÃO DO PROCESSO LOCAL.....	35
FIGURA 14	MODELAGEM DE INTERAÇÃO DA ALTERAÇÃO DE TAXA DE AQUISIÇÃO DO PROCESSO VIA WEB.....	35
FIGURA 15	MODELAGEM DE INTERAÇÃO DA OBTENÇÃO DE TEMPERATURA MÉDIA, MÁXIMA E MÍNIMA E GRÁFICO TEMPORAL DE UM DETERMINADO PERÍODO.....	36
FIGURA 16	MODELAGEM DE INTERAÇÃO DE CICLO DE AQUISIÇÃO.....	37
FIGURA 17	ESPECIFICAÇÃO DO SISTEMA. ....	38
FIGURA 18	MONTAGEM EXPERIMENTAL EM BANCADA.....	41
FIGURA 19	TELA DE CONFIGURAÇÃO DO PROCESSO. ....	42
FIGURA 20	PROCESSOS LAB1 E LAB2. ....	42
FIGURA 21	TELA PRINCIPAL DO APLICATIVO WEB.....	43
FIGURA 22	TELA DE MONITORAÇÃO REMOTA. ....	43
FIGURA 23	FABRICAPROCESSO, LAB1, LAB2 DO LADO ESQUERDO DA TELA E MONITORAÇÃO REMOTA DE LAB1 DO LADO DIREITO DA TELA.....	44
FIGURA 24	MODIFICAÇÃO DA TAXA REMOTA DE MONITORAÇÃO LAB2 DO LADO DIREITO DA TELA.....	45
FIGURA 25	DADOS ARMAZENADOS NO BANCO DE DADOS.....	45
FIGURA 26	TELA DE PESQUISA DE VARIÁVEIS AMBIENTAIS EM UM PERÍODO DE TEMPO PARA LAB1. ....	46
FIGURA 27	RESULTADO DA PESQUISA POR PERÍODO.....	47

---



## Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
ASP	<i>Application Service Provider</i>
CORBA	<i>Common Object Request Broker Architecture</i>
DLL	<i>Dynamic Link Library</i>
FTP	<i>File Transfer Protocol</i>
Gbps	<i>Gigabit per second</i>
GPIB	<i>General Purpose Interface Bus</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
IDL	<i>Interface Definition Language</i>
INMETRO	Instituto Nacional de Metrologia, Normalização e Qualidade Industrial
IPT	Instituto de Pesquisas Tecnológicas do Estado de São Paulo
J2EE	<i>Java 2 Enterprise Edition</i>
JDBC	<i>Java Database Connectivity</i>
JDK	<i>Java Development Kit</i>
JNI	<i>Java Native Interface</i>
JRE	<i>Java Runtime Environment</i>
JVM	<i>Java Virtual Machine</i>
Kbps	<i>Kilobit per second</i>
Mbps	<i>Megabit per second</i>
ORB	<i>Object Request Broker</i>
PHP	<i>HyperText Preprocessor</i>
RMI	<i>Remote Method Invocation</i>
SGDB	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
UML	<i>Unified Modeling Language</i>
USB	<i>Universal Serial Bus</i>
Web	<i>World Wide Web</i>
XML	<i>Extensive Markup Language</i>

---

# Sumário

<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 MOTIVAÇÃO .....	1
1.1.1 Manutenção de equipamentos antigos.....	1
1.1.2 Necessidade de aumento de produtividade .....	2
1.1.3 Qualidade e redução de erros de medição.....	3
1.1.4 Outras razões.....	5
1.2 PROBLEMA.....	6
1.3 OBJETIVO .....	7
1.4 REVISÃO BIBLIOGRÁFICA.....	8
<b>2 SOLUÇÕES EXISTENTES .....</b>	<b>10</b>
2.1 <i>SOFTWARE</i> .....	10
2.1.1 Soluções desenvolvidas internamente .....	10
2.1.1.1 Baseados em planilhas eletrônicas ou gerenciador de banco de dados.....	10
2.1.1.2 Desenvolvimento utilizando linguagem de programação .....	11
2.1.2 Soluções comerciais.....	12
2.1.2.1 <i>Software</i> fornecido com o equipamento de aquisição e ou controle.....	12
2.1.2.2 <i>Software</i> integrado a planilhas eletrônicas .....	12
2.1.2.3 <i>Software</i> genérico para automatização de calibrações e ensaios.....	13
2.2 <i>HARDWARE</i> .....	13
<b>3 SOLUÇÃO PROPOSTA .....</b>	<b>15</b>
3.1 FERRAMENTAS DE <i>SOFTWARE</i> UTILIZADAS .....	16
3.1.1 Metodologia de projeto por casos de uso.....	16
3.1.2 Ferramentas de <i>software</i> .....	17
3.1.2.1 Linguagem de programação Java .....	17
3.1.2.2 IDE Eclipse.....	18
3.1.2.3 Tecnologia RMI .....	19
3.1.2.4 Bibliotecas Jcomm, Jamod, Jfreechart .....	20
3.1.2.5 Sistema gerenciador de banco de dados MySql .....	21
<b>4 IMPLEMENTAÇÃO .....</b>	<b>22</b>
4.1 MODELAGEM DO DOMÍNIO .....	26
4.2 ANÁLISE DE ROBUSTEZ .....	27
4.2.1 Início de aquisição.....	28
4.2.2 Alteração de taxa de aquisição do criador de processos .....	29
4.2.3 Alteração de taxa de aquisição do processo local.....	29
4.2.4 Alteração de taxa de aquisição do processo via Web .....	30
4.2.5 Obtenção de temperatura e umidade média, máxima e mínima e um gráfico temporal de um determinado período.....	30
4.2.6 Ciclo de aquisição .....	31
4.3 MODELAGEM DE INTERAÇÃO.....	32
4.3.1 Início da aquisição.....	33
4.3.2 Alteração de taxa de aquisição do criador de processos .....	34
4.3.3 Alteração de taxa de aquisição do processo local.....	34
4.3.4 Alteração de taxa de aquisição do processo via Web .....	35
4.3.5 Obtenção de temperatura e umidade média, máxima e mínima e gráfico temporal de um determinado período .....	36

---

4.3.4 Ciclo de aquisição .....	37
4.4 ESPECIFICAÇÃO DO SISTEMA .....	38
<b>5 RESULTADOS.....</b>	<b>41</b>
<b>6 CONCLUSÕES.....</b>	<b>48</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>49</b>
<b>REFERÊNCIAS CONSULTADAS .....</b>	<b>51</b>

---

# 1 INTRODUÇÃO

## 1.1 Motivação

A automação laboratorial tornou-se bastante difundida nos laboratórios de ensaio e calibração, envolvendo desde a aquisição dos dados, realização de correções, filtragens e tratamentos estatísticos até a elaboração dos relatórios de ensaio. Em alguns casos, a automação engloba também a atuação sobre posicionadores e/ou equipamentos de controle, prescindindo, ainda mais, da interferência de operadores.

Diversas razões levaram ao uso crescente da automação laboratorial em processos de ensaios e calibrações, sendo algumas detalhadas a seguir.

### 1.1.1 Manutenção de equipamentos antigos

Durante as décadas de 70 e 80, um grande número de equipamentos para ensaio foram adquiridos e colocados em operação, para atender a demanda da crescente indústria nacional para a realização de ensaios e calibrações. Muitos destes equipamentos não possuíam sistemas de aquisição de dados, o que era feito pelo operador por inspeção visual em ponteiros e/ou painéis e anotados em planilhas de ensaios. Alguns equipamentos, entretanto, eram dotados de registradores gráficos, que permitiam a geração de curvas temporais da grandeza em medição. Com máquinas de tração, por exemplo, geravam-se as curvas de carga para se avaliar a resistência do corpo de provas.

Tais sistemas de registro e coleta de dados apresentam, em geral, vida útil inferior ao próprio equipamento ao qual estão conectados. O número de registradores/coletores danificados ou em manutenção passou a ser grande a partir do final da década de 80. Entretanto, a manutenibilidade dos mesmos era prejudicada devido à falta de peças e ao fato de utilizarem componentes eletrônicos obsoletos. Em alguns casos, por exemplo, ainda empregavam-se válvulas no lugar de circuitos transistorizados.

---

Os aparelhos que realizam estes ensaios são dos mais diferentes tipos e de diferentes fabricantes, dificultando a integração destes instrumentos para a centralização dos dados para serem analisados pelos técnicos responsáveis pelos resultados do ensaio ou calibração. Alguns destes equipamentos dispõem de algum tipo de saída digital ou analógica, incluindo algum *software* proprietário de aquisição.

Assim, os laboratórios viram-se obrigados a buscar soluções para recolocar os antigos equipamentos em funcionamento e também para uma maior integração entres todos os equipamentos envolvidos no ensaio ou calibração.

Portanto, a utilização de sistemas computadorizados para aquisição de dados permite a modernização de antigos equipamentos, aumentando suas vidas úteis a um baixo custo. Isto foi um fator importante na implantação da automação no IPT (Instituto de Pesquisas Tecnológicas do Estado de São Paulo), dadas as restrições orçamentárias e a necessidade crescente de aumento de receitas próprias.

### **1.1.2 Necessidade de aumento de produtividade**

Laboratórios credenciados, em geral, proporcionam resultados mais confiáveis, pois operam sob sistemáticas padronizadas e idealizadas para este fim. Entretanto, há uma tendência a ter a competitividade comprometida em função dos investimentos em equipamentos, calibrações, padrões, pessoal mais qualificado, etc., além de processos mais documentados. Assim, a questão principal é como manter ou aumentar a competitividade com uma lucratividade adequada para estes ambientes.

Novamente, a automação surge como uma das alternativas para o aperfeiçoamento dos processos a fim de manter ou incrementar a qualidade, alcançando os níveis de produtividade e redução de custos desejados.

Dessa forma existem três maneiras possíveis para aumentar competitividade e lucratividade dos laboratórios:

1. Aumentar a margem de lucro dos serviços diminuindo os custos operacionais;
  2. Aumentar a demanda de serviços para que, mesmo com menores margens de lucro, possa ser auferido um lucro operacional desejado;
-

### 3. Aumentar a produtividade na realização dos serviços.

A automação laboratorial atingirá diretamente o primeiro e terceiro fator acima mencionados, pois acelera a execução dos procedimentos e reduz a mão de obra diretamente envolvida no ensaio.

#### 1.1.3 Qualidade e redução de erros de medição

Os procedimentos laboratoriais de calibração e ensaios requerem precisão, exatidão, repetibilidade, reprodutibilidade e alta confiabilidade, principalmente no contexto atual da Metrologia e da Tecnologia, que tem exigido uma forma de adequação às inovações tecnológicas e necessidades de um mercado globalizado extremamente competitivo.

Em um ensaio laboratorial ocorrem inúmeros erros, que devem ser levados em conta e de alguma forma tratados física ou matematicamente para uma redução dos seus impactos nos resultados gerados pelo laboratório.

As principais fontes de erro são:

- Variações do ambiente (temperatura, umidade, pressão etc.);
- Inabilidade técnica;
- Erros de leitura;
- Erros de digitação;
- Erros de transcrição dos resultados;
- Erros de cálculos em sistemas computacionais (truncamento, arredondamento, etc.).

A aquisição automática de medições por computador pode melhorar sensivelmente todos estes erros, pois torna os procedimentos laboratoriais menos ou totalmente imunes aos erros humanos, também padronizando a execução das tarefas, interagindo com todos os elementos envolvidos na calibração e ensaio de uma forma bastante confiável, precisa e pré-determinada. Assim, pode-se se dizer que:

---

- Os erros oriundos de variações do ambiente podem ser compensados ou minimizados, uma vez que o processo de coleta da leitura tende a ser mais rápido, reduzindo assim o tempo para a ocorrência destas variações;
- Quanto maior for o nível da automação, menor o envolvimento do operador técnico, logo, é menor a influência de sua inabilidade ou estado emocional;
- Na maioria dos casos, como o processo de coleta de dados é automático elimina-se a possibilidade de erros de transcrição, visto que nestes casos esta etapa não existe;
- A coleta de dados pode ser feita via alguma interface de comunicação entre o computador e equipamento, eliminando-se assim a possibilidade de leituras errôneas e erros de digitação.

Adicionalmente, as certificações e implantação de sistemas de qualidade passaram a ser necessárias para que os laboratórios pudessem prestar serviços a determinados setores da indústria. Normas técnicas mais rígidas começaram a ser publicadas e os laboratórios deveriam se adequar às mesmas. Dentre diversas exigências, destacava-se a calibração periódica de todos os equipamentos de testes e aquisição de dados. Entretanto, os leitores baseados em ponteiros ou mesmo em registradores gráficos não atendiam a tais normas. Algumas fontes de erros como paralaxe, espessura do traçado e desalinhamento não permitiam a calibração dos mesmos, nem mesmo a repetibilidade necessária nos processos de medição. Novamente, a utilização de sistemas digitais computadorizados para aquisição dos dados pôde dar uma solução para tal impasse a um custo baixo, sem a necessidade da substituição de todo o equipamento.

Os laboratórios de ensaios e calibrações devem requerer sua certificação como laboratório credenciado perante o INMETRO (Instituto Nacional de Metrologia, Normalização e Qualidade Industrial). Este órgão exige que as condições ambientais devem sejam consideradas, monitoradas e controladas dentro de certos limites, devido a sua influência na incerteza de medição.

Desta maneira, as medidas de ambiente (temperatura e umidade), devem ser monitoradas durante todo ensaio para que possam ser feitas correções das medidas

---

ou verificar se o ensaio ocorreu com a temperatura e umidade dentro dos padrões exigidos. Como alguns ensaios demandam de um período de tempo muito longo, a monitoração constante das variáveis de ambiente tornou-se obrigatória.

#### **1.1.4 Outras razões**

Além dos motivos mencionados anteriormente, a automação laboratorial possui as seguintes vantagens adicionais:

1. Possibilidade de se ter um ensaio contínuo fora do expediente normal do laboratório sem a necessidade de acompanhamento humano;
  2. Redução da incerteza de medição;
  3. Melhora na reprodutibilidade das medições;
  4. Preservação dos equipamentos com uma menor interação mecânica devido a mudanças através de operação manual;
  5. Trabalhos executados em um menor tempo, com conseqüente economia de energia;
  6. Com o uso da automação em uma rede local, intranet, ou através da Internet, permite-se que as calibrações e ensaios sejam monitorados e até controlados a distância, eliminando-se a necessidade da presença de um operador local.
-



## 1.2 Problema

Em razão do alto custo das redes de computadores na época da aquisição dos instrumentos dos laboratórios, os programas aplicativos fornecidos junto dos equipamentos ou mesmo aqueles desenvolvidos para estes instrumentos foram concebidos para funcionar em um único computador, desconectado da rede, e normalmente utilizam apenas o serviço de disponibilizar os dados em forma de um arquivo. Cada aplicação era concebida de forma independente uma da outra, permitindo apenas a troca de dados por arquivo.

Os laboratórios que trabalham com realização de ensaios de amostras fornecidas pelos clientes têm a finalidade básica de colecionar dados de medições realizadas ao longo do ensaio, fazendo comparações com padrões de referência ou comparando com normas existentes.

A necessidade de conexão das interfaces por meio de cabos de dados ligados ao micro torna a instrumentação presa a um microcomputador que possua interfaces compatíveis com os instrumentos.

Devido à aquisição e controle estarem sendo feitas em um computador específico é necessário que os dados e controle do tempo de aquisição possam ser alterados e monitorados através de outro computador por meio de uma rede (Intranet, Internet), para possibilitar o controle e a monitoração em tempo real das medidas realizadas e possíveis correções no intervalo de aquisição dessas medidas, podendo estas serem acessadas de qualquer ponto da rede.

É interessante para o laboratório que todas as medidas realizadas possam ser acessadas de alguma maneira para que o usuário possa ter uma idéia da medida instantânea e também o histórico dessas medidas num determinado espaço de tempo requerido pelo usuário.

---

### 1.3 Objetivo

A aplicação de sistemas computadorizados para aquisição de dados e controle de laboratório tornou-se uma necessidade para os laboratórios de ensaio e calibração, devido a diversos fatores econômicos e conjunturais mencionados anteriormente.

Devido a exigências dos órgãos certificadores para que a temperatura e umidade do ambiente sejam controladas e/ou monitorados durante a realização de ensaios e calibrações, verificou-se a necessidade de um sistema automatizado para a realização desta monitoração.

Os instrumentos e sensores que fazem as medidas e controle ambiental devem estar conectados a um microcomputador, o que obriga o sistema de conversão dos dados adquiridos a estar próximo aos sensores. Verifica-se, então, a necessidade adicional do acesso aos dados de ambiente através de outros computadores do laboratório para efetuar a elaboração do relatório e também se certificar que o procedimento está sendo realizado dentro das exigências da norma do ensaio ou calibração.

Desta maneira, o objetivo será o projeto e desenvolvimento de uma ferramenta remota e descentralizada de monitoração e controle de variáveis ambientais de laboratório.

---

## 1.4 Revisão Bibliográfica

O trabalho de Czapski, J. (2000) descreve detalhadamente a automação laboratorial, apresentado todas as classes de componentes envolvidos (*software* e *hardware*) e pode ser utilizado como fonte inicial de pesquisas e estudos nesta área. Adicionalmente, menciona-se o trabalho de Ramos, H.G.; Trindade, M.; Cruz Serra, A. (1997), que apresenta um caso estudo de automação laboratorial baseada em PC, que reforça todos os conceitos aqui apresentados.

Vários trabalhos relacionados à automação e controle remoto de experimentos através da Internet têm sido desenvolvidos, como por exemplo o trabalho de Paschoal, Jr. (2006).

O objetivo deste trabalho foi desenvolver a aplicação WebLab para medida remota do coeficiente de atenuação em uma fibra óptica. O controle dos instrumentos de medida foi feito localmente através de interfaces GPIB (*General Purpose Interface Bus* – Interface de Barramento de Propósito Geral), placa condicionadora de sinais e do programa de instrumentação LabView. O controle remoto do experimento é implementado utilizando-se uma arquitetura Cliente/Servidor dupla. No Servidor Local foi utilizado o Windows Server 2003 para a instalação e execução do LabView, no qual desenvolveu-se a interface homem-máquina local e os protocolos de aquisição dos dados dos instrumentos interconectados com o microcomputador local.

No Servidor Web foi instalado o sistema operacional Linux com Apache, PHP e MySql. O serviço remoto foi disponibilizado através de um navegador Web utilizando-se de aplicativos baseados na tecnologia Web, e é executado através do navegador. Os comandos do instrumento remoto são transmitidos para o Servidor Web e transmitidos para o Servidor Local.

Outro trabalho nesta área foi desenvolvido por Amorim, A.C.O. (2001). O objetivo do trabalho foi desenvolver uma aplicação remota para o ensaio de calibração de resistores padrão. Implementou-se uma estrutura Servidor-Cliente utilizando CORBA (*Common Object Request Broker Architecture*), na qual o instrumento é fisicamente interligado a um microcomputador Cliente através de GPIB

---

e a comunicação com o instrumento foi implementada em Delphi através de DLL (*Dynamic Link Library*).

O controle e monitoramento remoto através dos computadores da rede Intranet foi implementada utilizando-se o Java, dessa maneira, as DLLs em Delphi foram encapsuladas com JNI (*Java Native Interface*) para que o programa em Java conseguisse utilizá-las. O servidor de nomes utilizados foi o ORB (*Object Request Broken*) da SUN, o tnameserv, executado em um dos servidores da rede.

No trabalho apresentado em Tannuri, E.A.; Noda, M.K.; Romano, R. (2004), apresentam-se diversos estudos de caso de automação laboratorial. Verificou que com a automação laboratorial baseada em microcomputadores houve um aumento de produtividade, uma adequação as normas de qualidade e uma adaptação mais rápida dos técnicos dos laboratórios.

Os sistemas de aquisição e controle descritos foram implementados em Delphi, com interfaces gráficas com o usuário. O *hardware* utilizado para a comunicação com os instrumentos foram placas de aquisição de dados, devido aos instrumentos comportarem somente sinais analógicos.

---

## 2 SOLUÇÕES EXISTENTES

Diversas soluções de *hardware* e *software* podem ser utilizadas, dependendo das características do ensaio tratado. Utilizando-se automação baseada em PC, é possível garantir menores custos, maior facilidade de operação, maior aceitação e menor tempo de treinamento para os técnicos de laboratório.

A seguir, serão apresentadas algumas soluções utilizadas em automação laboratorial, no que se refere a *software* e *hardware*.

### 2.1 Software

Com a disponibilidade atual de uma infra-estrutura de rede básica (Internet, Intranet) cresce a possibilidade de monitorar e controlar remotamente os processos. A necessidade de integração de equipamentos antigos, mas de alta precisão, com equipamentos mais novos e sofisticados para realizar os ensaios, com uma crescente demanda de números de ensaios pelos clientes, mas que mantenham a qualidade do serviço, tem forçado a busca da automação do ensaio e uma disponibilidade mais rápida dos dados adquiridos para serem analisados pelos técnicos.

Hoje existem diferentes ferramentas que podem ser utilizadas para solucionar um mesmo problema de aquisição de dados. Desta forma é muito difícil perceber as diferenças relevantes entre estas soluções de automação.

Desta maneira é importante classificar os diferentes tipos de soluções existentes no mercado, verificando-se a existência de dois tipos de soluções para automação, soluções desenvolvidas internamente e soluções comerciais.

#### 2.1.1 Soluções desenvolvidas internamente

##### 2.1.1.1 Baseados em planilhas eletrônicas ou gerenciador de banco de dados

Este tipo de solução apresenta muitas limitações, devido a utilizarem normalmente o banco de dados Microsoft Access ou a planilha eletrônica Microsoft

---

Excel da Microsoft. Seu desenvolvimento fica dependente dos requisitos disponibilizados por estas ferramentas comerciais mais direcionadas à parte administrativa do que técnica.

O Microsoft Access permite a criação de um banco de dados para o armazenamento de um histórico de medidas, mas necessita de um programa auxiliar para a aquisição dos dados.

A planilha eletrônica Microsoft Excel é muito prática para o registro das medições e permite uma rápida visualização gráfica dos resultados, mas devido à programação ser feita internamente em cada uma de suas células dificulta-se o desenvolvimento e manutenção do programa e também é necessário um programa auxiliar para a aquisição dos dados.

Como estas duas ferramentas pertencem à família Microsoft, o funcionamento delas depende exclusivamente da plataforma Windows.

#### **2.1.1.2 Desenvolvimento utilizando linguagem de programação**

São Aplicativos desenvolvidos especificamente para um dado problema, utilizando linguagens de programação como Delphi, Visual Basic, Visual C++, .Net e Java. Estas linguagens permitem uma melhor interação do usuário com o *software*, e por permitirem o rápido desenvolvimento da interface gráfica, garantem também uma total flexibilidade na elaboração desta interface com os equipamentos e placas de aquisição.

Nesse tipo de desenvolvimento deve existir uma interação entre o programador e os técnicos de laboratório para que a análise de requisitos contemple todas as funcionalidades necessárias para o desenvolvimento do aplicativo de automação e controle.

Utilizando-se este tipo de desenvolvimento consegue-se armazenar os dados em arquivos ou mesmo trabalhar conjuntamente com banco de dados. Existe uma melhor visualização dos dados adquiridos em algum tipo de tabela ou mesmo através de gráficos temporais. Pode-se também implementar o relatório específico de resultados referentes ao ensaio ou calibração realizado.

---

## **2.1.2 Soluções comerciais**

Identificam-se três linhas de *softwares* disponíveis no mercado para a automação de calibração e ensaios, contemplando cálculo de incertezas, que são descritas a seguir.

### **2.1.2.1 Software fornecido com o equipamento de aquisição e ou controle**

São aplicativos conectados a banco de dados que possuem internamente algumas facilidades para registro de medições simplificadas. Eles realizam cálculos de incertezas de medição em um nível intermediário de automatização, onde as correções, coeficientes de sensibilidade, contribuições de incertezas e outros parâmetros podem ser obtidos do banco de dados ou podem ser digitados pelo usuário diretamente no registro das medições, com base nas características do equipamento e em seus certificados de calibração.

São utilizadas em aplicações com pouca sofisticação nos métodos de medição e requisito de qualidade, mas que exigem ferramentas para consulta, relatórios e gráficos específicos.

Como não permitem modificações significativas sobre seus requisitos, são pouco recomendadas para laboratórios de calibração ou ensaios, sendo destinados a aplicações mais simples.

### **2.1.2.2 Software integrado a planilhas eletrônicas**

São aplicativos que trabalham com banco de dados e são integrados a planilhas eletrônicas para permitir ao usuário a automatização do registro de cálculo das incertezas de medição. Dependem da implantação de planilhas eletrônicas para poderem operar na sua totalidade. Possibilitam o gerenciamento das informações e alguma facilidade para consultar os resultados obtidos nas planilhas.

Os métodos de medição devem estar implementados nas células da planilha eletrônica, exigindo então também um especialista nessa ferramenta.

---

Há a limitação da plataforma de trabalho, já que a maioria destes aplicativos opera com os produtos da Microsoft e do próprio Ms-Windows, não funcionando em ambientes Linux.

### **2.1.2.3 Software genérico para automatização de calibrações e ensaios**

São ferramentas desenvolvidas especialmente para laboratórios de calibração e ensaios, que envolvem os aspectos de gestão e automatização a eles pertinentes. São ferramentas flexíveis que permitem a integração com planilhas para facilitar a implantação e contemplam ferramentas para elaboração de qualquer método de medição no próprio *software*, dispensando o uso de planilhas e atingindo todos os níveis de automatização em calibrações e ensaios. Como exemplos, citam-se o Autolab e o Labview.

## **2.2 Hardware**

Para ensaios que requerem a monitoração de diversas variáveis ou que necessitam alta taxa de amostragem, utilizam-se condicionadores de sinais ligados a placas de aquisição de dados. Uma solução alternativa, com custo vantajoso em alguns casos, envolve a utilização de módulos de aquisição com comunicação serial/USB (*Universal Serial Bus*). Para ensaios com lenta taxa de amostragem (processos térmicos por exemplo) podem-se utilizar controladores ou indicadores digitais, que se comunicam com o microcomputador através da porta serial. Alguns laboratórios utilizam equipamentos que já possuem comunicação com computador, requerendo, em alguns casos, placas de interface apropriadas, como por exemplo IEE488 (GPIB).

Inicialmente os instrumentos disponibilizavam somente a interface de comunicação RS232-C, que é uma interface de controle serial em que a comunicação com equipamentos ocorre a uma taxa de transferência máxima de 115 Kbps (*Kilobits per second*), mas com um protocolo frágil.

Com o desenvolvimento de outras interfaces como a interface USB, GPIB e ETHERNET, possibilitando uma maior taxa de transferência, obteve-se um

---



desenvolvimento de protocolos de comunicação mais robustos entre o microcomputador e o equipamento.

A interface de comunicação USB 1.1 possibilita uma comunicação com taxa de transferência de 12 Mbps (*Megabits per second*). A versão 2.0 possui taxa de transferência de 480 Mbps, mas ainda com uma comunicação frágil.

A interface GPIB possui taxa de transferência de 1 Mbps, menor que a USB, mas com um protocolo robusto de comunicação.

A interface ETHERNET possui uma taxa de transferência de 10-100 Mbps chegando nos tempos atuais até a Gbps (*Gigabits per second*), também com um protocolo robusto.

Existem também instrumentos somente com saídas analógicas em tensão ou corrente, necessitando de uma placa condicionadora de sinais no microcomputador.

---

### 3 SOLUÇÃO PROPOSTA

A solução adotada no presente trabalho enquadra-se no grupo de aplicativos desenvolvidos internamente utilizando linguagem de programação.

Para a realização da monitoração e controle remoto da temperatura e umidade, deve ser considerada a existência de um amplo parque no IPT de plataforma Windows já instalado e a existência da plataforma Linux que deve crescer principalmente devido ao baixo custo e à alta confiabilidade do sistema.

As aplicações precisam interagir entre si, permitindo a troca de dados e a notificação de eventos entre aplicações, de forma a garantir a notificação de falhas e alterações importantes no estado dos processos.

Como o computador está fisicamente conectado ao equipamento, foi definida uma implementação onde:

- O cliente está restrito às funcionalidades de interface com o usuário via Navegador Web.
- A lógica da aplicação será alocada no servidor.

Optou-se pela utilização de ferramentas *Open Source* para o desenvolvimento de todos os módulos da ferramenta de monitoração remota de ambiente, desde a interface homem-máquina no Servidor Local com os módulos de aquisição de dados que se comunicam com o instrumento, o Servidor de Serviços baseado em tecnologia Java e a interface gráfica remota executada sob o navegador Web.

A seguir serão apresentadas as ferramentas computacionais envolvidas na solução proposta, descrevendo-se suas vantagens e as razões da escolha dos mesmos. Apresenta-se também a metodologia para o projeto do sistema de monitoração e controle de variáveis de ambiente, baseada em casos de uso.

---

## 3.1 Ferramentas de *software* utilizadas

### 3.1.1 Metodologia de projeto por casos de uso

O desenvolvimento de uma ferramenta de *software* pode ser baseada nos requisitos que o usuário deseja dela e de que maneira a ferramenta responde a esses requisitos. As principais etapas dessa metodologia são:

**Obtenção dos casos de uso.** Os casos de uso capturam os requisitos da aplicação através da representação dos atores que usam o (ou são usados pelo) sistema e possíveis ações. Eles são descritos sob o ponto de vista do usuário: o que ele pode fazer, como ele interage com o sistema. Os casos de uso são apresentados na forma de diagramas UML (*Unified Modeling Language* – Linguagem de Modelagem Padrão) de casos de uso para o sistema e descrição textual de cada caso de uso. A descrição textual contempla pelo menos o fluxo de eventos para a execução da ação associada ao caso de uso e os fluxos alternativos, caso alguma situação interrompa o fluxo principal de eventos.

**Modelagem do domínio.** A partir dos textos associados aos casos de uso, extrai-se (por uma análise dos substantivos presentes nos textos) os conceitos que serão potencialmente associados a objetos da aplicação. Representam-se então esses conceitos através de um diagrama de classes UML (neste momento, sem atributos ou métodos).

**Análise de robustez.** Para cada caso de uso, identificam-se os objetos e os eventos que os relacionam através de um diagrama de robustez. Objetos podem ser de três tipos: de fronteira, de controle ou de entidade. Atores só interagem com objetos de fronteira, que não podem interagir diretamente com objetos de entidade.

**Modelagem de interação.** Para cada caso de uso, um diagrama de seqüência UML é desenvolvido. Nesta etapa do processo, comportamento (um conjunto de métodos) é atribuído a cada objeto da aplicação.

**Especificação do sistema.** Com os métodos para cada tipo de objeto já definidos, o diagrama de classe é revisto para incorporar a definição dessa interface operacional a cada classe.

---

### 3.1.2 Ferramentas de *software*

#### 3.1.2.1 Linguagem de programação Java

A Sun estabeleceu a primeira linguagem de programação que não estava atada a nenhum sistema operacional ou microprocessador em particular, a linguagem de programação Java. Os aplicativos escritos em Java são executados em qualquer plataforma eliminando problemas de incompatibilidade entre sistemas operacionais e versões dos sistemas operacionais.

Características da Linguagem Java:

- Orientada a Objetos;
- Interpretada;
- Neutra em relação à arquitetura e portátil;
- Dinâmica;
- Distribuída;
- Simples;
- Concisa;
- Robusta;
- Segura;
- Multitarefa.

A linguagem Java é uma linguagem de alto nível, com sintaxe similar à do C++, e com diversas características herdadas de outras linguagens, como a Smalltalk e Modula-3.

Java é uma linguagem simples e independente de arquitetura e a memória alocada dinamicamente é gerenciada pela própria linguagem, que usa algoritmos de *garbage collection* para desalocar regiões de memória que não estão mais em uso.

Java é uma linguagem orientada a objetos que segue a linha purista iniciada por Smalltalk. Com a exceção dos tipos básicos da linguagem, a maior parte dos elementos de um programa Java são objetos onde o código é organizado em classes, que podem estabelecer relacionamentos de herança simples entre si.

As chamadas a funções de acesso remoto (*sockets*) e os protocolos Internet mais comuns (HTTP, FTP, Telnet, etc.) são suportadas em Java. Desta forma, torna-se mais simples a elaboração de aplicativos baseados em arquiteturas cliente-servidor.

---

Java tem o suporte a multitarefa embutido na linguagem, sendo que um programa Java pode possuir mais de uma linha de execução (*thread*). A programação concorrente em geral é uma tarefa difícil, entretanto a linguagem Java fornece diversos recursos de sincronização de processos que tornam a programação mais simples.

Programas desenvolvidos em Java costumam ser executados sem problemas no computador, já que a máquina virtual Java faz uma verificação em tempo de execução quanto aos acesso de memória, abertura de arquivos e uma série de eventos que podem gerar um “congelamento” em outras linguagens, mas que geram exceções em programas Java. Em geral, ao escrever programas Java utilizando-se de herança de classes predefinidas, o programador é obrigado a escrever algumas rotinas de tratamento de exceção.

Uma das características de Java que a tornou ideal para seu uso na elaboração de aplicativos distribuídos foi a sua independência de plataforma. Java consegue essa independência devido ao fato de que o compilador Java não gera instruções específicas a uma plataforma, mas sim um programa em um código intermediário, denominado *bytecode*, que pode ser descrito como uma linguagem de máquina destinada a um processador virtual que não existe fisicamente. O código Java compilado pode então ser executado por um interpretador de *bytecodes*, a JVM (*Java Virtual Machine* – Máquina Virtual Java), que é um emulador do processador.

### 3.1.2.2 IDE Eclipse

Escolheu-se a IDE Eclipse que é um arcabouço para integrar diferentes tipos de aplicações.

Essas aplicações são oferecidas em forma de bibliotecas e automaticamente reconhecidas e integradas pela plataforma. Possui recursos próprios para gerenciamento de mecanismo, que são geralmente arquivos no seu disco rígido. Eles residem no seu espaço de trabalho, em uma pasta especial localizada no seu sistema de arquivos. As aplicações instaladas comunicam-se entre si e, com isso, caso uma aplicação altere um recurso qualquer, todas as outras aplicações

---

instaladas serão notificadas sobre essa mudança, garantindo uma consistência e integração em todo o seu ambiente de desenvolvimento.

Um usuário sempre trabalha na bancada de trabalho, na parte "visível" da plataforma. A perspectiva escolhida determina a aparência atual da área de trabalho. A perspectiva é uma coleção conhecida como "janelas e editores" que contribuem com ações especiais para o menu e a barra de ferramentas.

A maioria das janelas mostra informações especiais sobre os recursos. Dependendo da janela, somente partes ou relacionamentos internos dos recursos poderão ser mostrados, como no caso de arquivos do tipo XML (*Extensive Markup Language*).

Um editor trabalha diretamente sobre um recurso (classes Java como exemplo). O Eclipse segue um ciclo de carrega-altera-salva que somente se um editor salvar suas alterações, a plataforma será capaz de notificar as outras aplicações.

Janelas especiais podem ser conectadas diretamente a um editor, adicionando recursos complementares ao manuseamento dos recursos.

O que torna o Eclipse uma IDE (*Integrated Development Environment* – Ambiente de Desenvolvimento Integrado) especial, é a extrema flexibilidade na qual podem ser combinadas janelas e editores. Dessa forma, o espaço de trabalho pode ser organizado de uma forma livre pelo desenvolvedor. As janelas e editores podem ser adicionados em uma perspectiva aberta (mesmo se eles foram definidos em um *plugin* totalmente diferente). Portanto é possível ter a total liberdade para criar o ambiente de desenvolvimento para o desenvolvedor, de uma forma customizada.

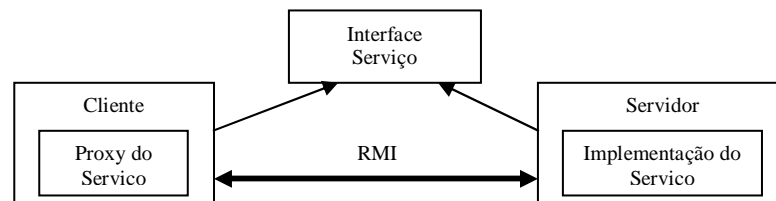
### **3.1.2.3 Tecnologia RMI**

A arquitetura RMI (*Remote Method Invocation*) estende a segurança e robustez da arquitetura Java para o mundo da computação distribuída.

É baseada em interfaces e classes. As interfaces implementam o comportamento e as classes definem a implementação.

---

A classe que implementa o serviço é executada do lado do servidor, e a classe que é executada no cliente atua como um *Proxy* para o serviço remoto conforme Figura 1.



**Figura 1 Arquitetura do Servidor RMI.**

O programa cliente faz chamadas de métodos pelo objeto Proxy, o RMI envia a requisição para a JVM remota e redireciona para a implementação e qualquer valor retornado pela implementação é devolvido ao Proxy e então ao programa cliente.

Para uma arquitetura orientada a objetos devem-se estabelecer regras, diretrizes e definir como as aplicações podem se comunicar e inter-operar. Assim, a maneira pela qual é realizada a implementação tem pouco interesse, mas sim como foi montadas a infra-estrutura e a interface entre os componentes da implementação.

Na plataforma Java, existem dois mecanismos para o desenvolvimento de aplicações usando o conceito de objetos distribuídos: Java RMI e Java IDL. Java RMI é um mecanismo para desenvolver aplicações com objetos distribuídos que opera somente com objetos Java e Java IDL utiliza a arquitetura padrão CORBA para integração de aplicações Java a aplicações desenvolvidas em outras linguagens.

#### **3.1.2.4 Bibliotecas Jcomm, Jamod, Jfreechart**

A biblioteca Jcomm foi desenvolvida para capturar dados da porta serial do microcomputador PC. A biblioteca Jamod foi desenvolvida para implementar, em conjunto com a biblioteca Jcomm, as características do protocolo Modbus, protocolo de comunicação de chão de fábrica.

A biblioteca Jfreechart foi desenvolvida para implementar gráficos em aplicativos Java pode ser utilizada em aplicações locais, *applets*, *servlets* e JSP. Pode ser usado para gerar gráficos de pizza, gráfico de barra, gráficos de linha, gráficos combinados, dentre diversos outros tipos de gráficos.

### **3.1.2.5 Sistema gerenciador de banco de dados MySql**

O MySql é um SGDB (Sistema Gerenciador de Banco de Dados), que utiliza a linguagem SQL (*Structured Query Language* – Linguagem de Consulta Estruturada) como interface. Ela possui a característica de ter o código aberto e ser portátil e compatível com diversas linguagens de programação.

É reconhecido pelo seu excelente desempenho, sendo bastante estável e também por ser multitarefa e multiusuário.

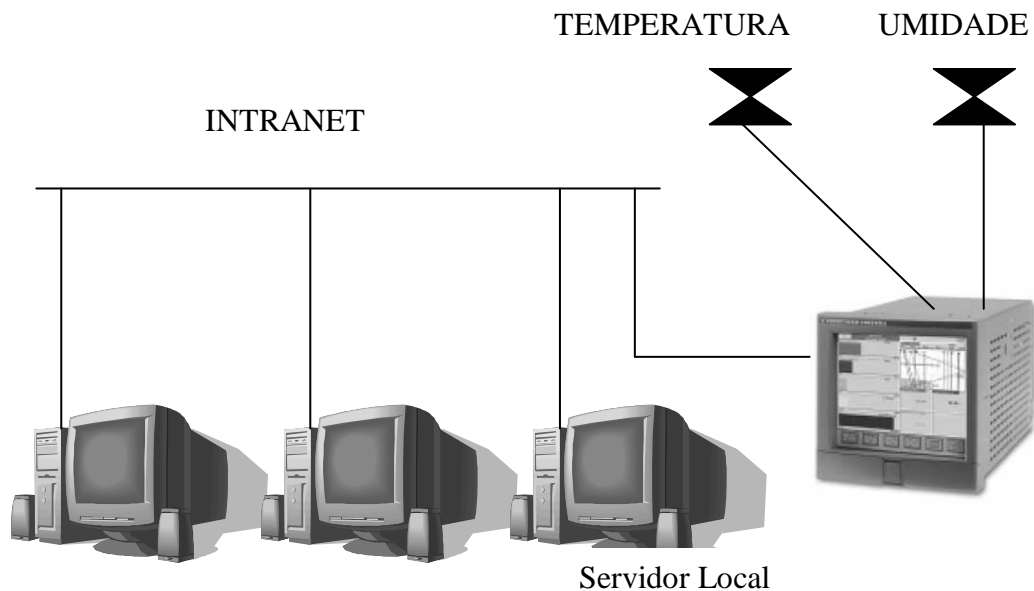
Através do mysqlconnector a linguagem Java se comunica com o banco de dados MySql.

---



## 4 IMPLEMENTAÇÃO

Na Figura 2 abaixo é mostrada a montagem experimental utilizada para a ferramenta de monitoração remota de ambiente.



**Figura 2 Montagem Experimental**

Utilizaram-se os seguintes equipamentos e aplicativos para a montagem experimental:

- Rede Intranet do laboratório 10/100 Mbps.
- Notebook como Servidor Local com placa Ethernet 100 Mbps..
- Sensor de temperatura padrão PT 100 e sensor de umidade, em bancada.
- Fonte de alimentação.
- Um Registrador Gráfico Eurotherm modelo 5100V com porta Ethernet 100 Mbps com protocolo de comunicação Modbus.
- IDE Eclipse com as bibliotecas instaladas (RMI, Tomcat Apache, Jcomm, Jmod, Jfreechart, Mysqlconnector).
- MySql Server instalado com o banco de dados criado e definido.
- JDK 1.5.0 e JRE 1.5.0 instalados.
- Servidor WEB Tomcat Apache instalado e configurado para a aplicação.

Utilizando-se o desenvolvimento através dos casos de uso obtiveram-se as seguintes funcionalidades da aplicação:

- Início de Aquisição

- Ator: usuário

Pré-condição: FabricaProcesso e o RMI iniciados.

Texto: O usuário seleciona porta de comunicação na lista de portas de comunicação, o valor da taxa de aquisição e o nome do laboratório onde será feita a monitoração, depois o usuário seleciona o botão de “Executar”, a classe FabricaProcesso cria o Processo, o cadastra e registra os seus serviços.

Pós-condição: O processo de aquisição é iniciado e os serviços do processo são registrados.

- Alteração da Taxa de Aquisição do criador de Processos

- Ator: usuário

Pré-condição: FabricaProcessoGui, RMI e MySql iniciados.

Texto: O usuário seleciona botão de “<” (decrementar) ou botão de “>” (incrementar) o valor alterado da taxa de aquisição é mostrado na interface gráfica.

Pós-condição: O valor da taxa de aquisição é decrementado ou incrementado.

- Alteração da Taxa de Aquisição do Processo Local

- Ator: usuário

Pré-condição: Processo de aquisição, RMI e MySql iniciados.

Texto: O usuário seleciona botão de “<” (decrementar) ou botão de “>” (incrementar) o valor alterado da taxa

---

de aquisição é mostrado na interface gráfica para os clientes cadastrados.

Pós-condição: O valor da taxa de aquisição é decrementado ou incrementado.

- Alteração da Taxa de Aquisição via Web

- Ator: usuário remoto

Pré-condição: Processo de aquisição, RMI, MySql e J2EE com servidor Tomcat iniciados.

Texto: O usuário seleciona o nome do laboratório que estão sendo adquiridos da lista, usuário seleciona botão de “<” (decrementar) ou botão de “>” (incrementar) o valor alterado da taxa de aquisição é mostrado na interface gráfica para os clientes cadastrados.

Pós-condição: O valor da taxa de aquisição é decrementado ou incrementado.

- Obtenção de temperatura e umidade média, máxima, mínima e um gráfico temporal de um determinado período.

- Ator: usuário

Pré-condição: Banco de dados MySql e servidor web J2EE com Tomcat devem estar inicializados.

Texto: O usuário digita o nome do laboratório e seleciona o dia inicial, mês inicial, ano inicial, hora inicial e minutos iniciais, e o dia final, mês final, ano final, e hora final e minutos finais depois ele seleciona o botão “Fazer Pesquisa”.

Pós-condição: O nome do laboratório com o período pesquisado com os valores de temperatura e umidade média,

---

máxima e mínima e um gráfico temporal serão mostrados para o usuário.

- Ciclo de Aquisição

- Ator: usuário

Pré-condição: Processo de aquisição, banco de dados e RMI (Remote Method Invocation) devem estar inicializados.

Texto: A cada ciclo da taxa de aquisição o sistema dispara um pedido de aquisição de temperatura e umidade, e ambas são adquiridas e armazenadas no banco de dados e mostrada na interface gráfica para os clientes cadastrados.

Pós-condição: Os atuais valores de temperatura e umidade são atualizadas para os clientes cadastrados.

---

## 4.1 Modelagem do domínio

Utilizando a Modelagem do domínio sobre os casos de uso obtidos, obteve-se o seguinte diagrama UML(*Unified Modeling Language*).

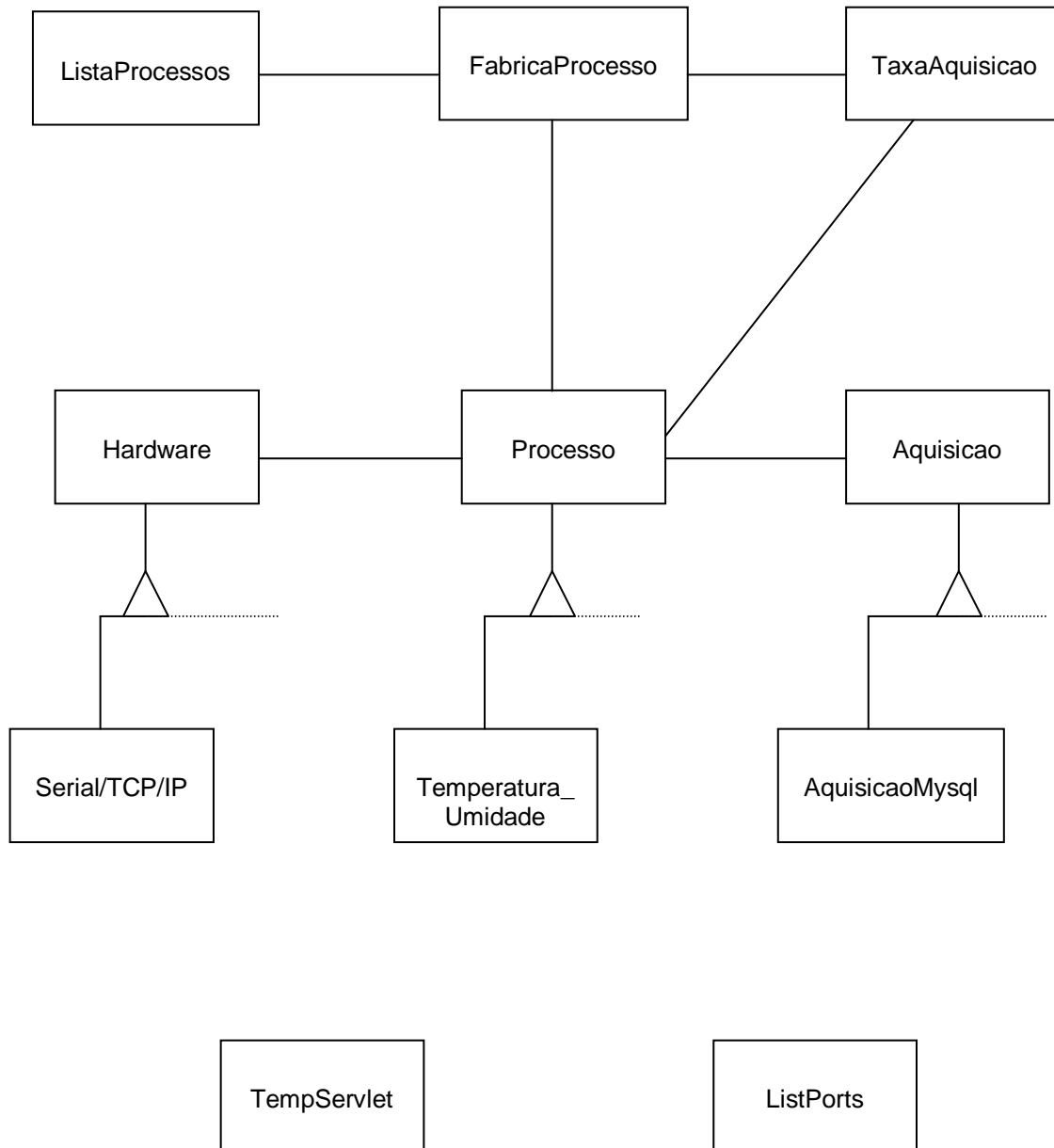
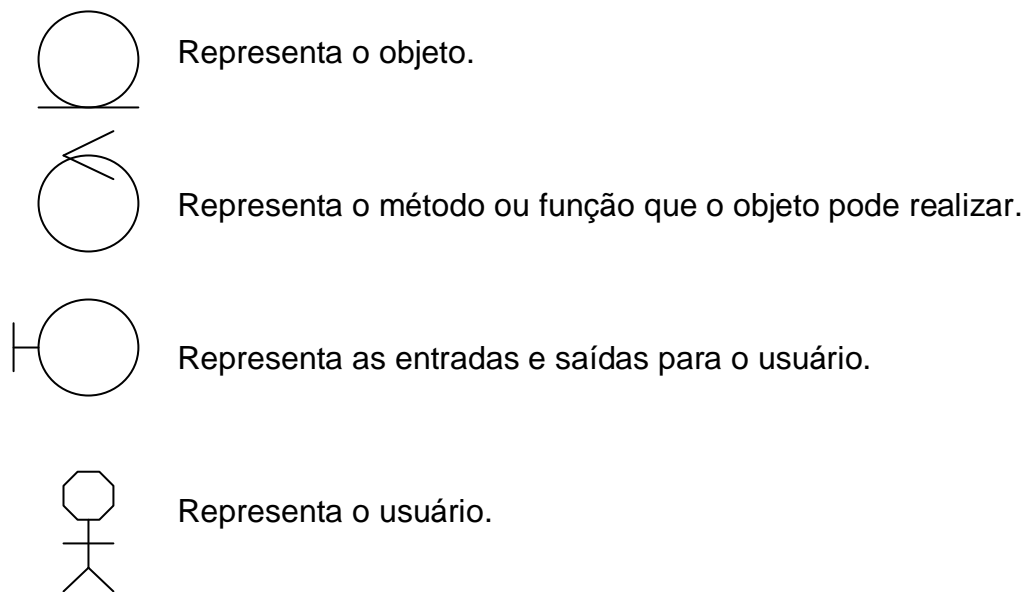


Figura 3 Modelagem do domínio do Sistema

## 4.2 Análise de robustez

Aplicou-se a análise de robustez sobre os casos de uso obtidos, a fim de verificar como os objetos relacionam-se entre si. O significado das figuras utilizadas no diagrama é descrito a seguir:

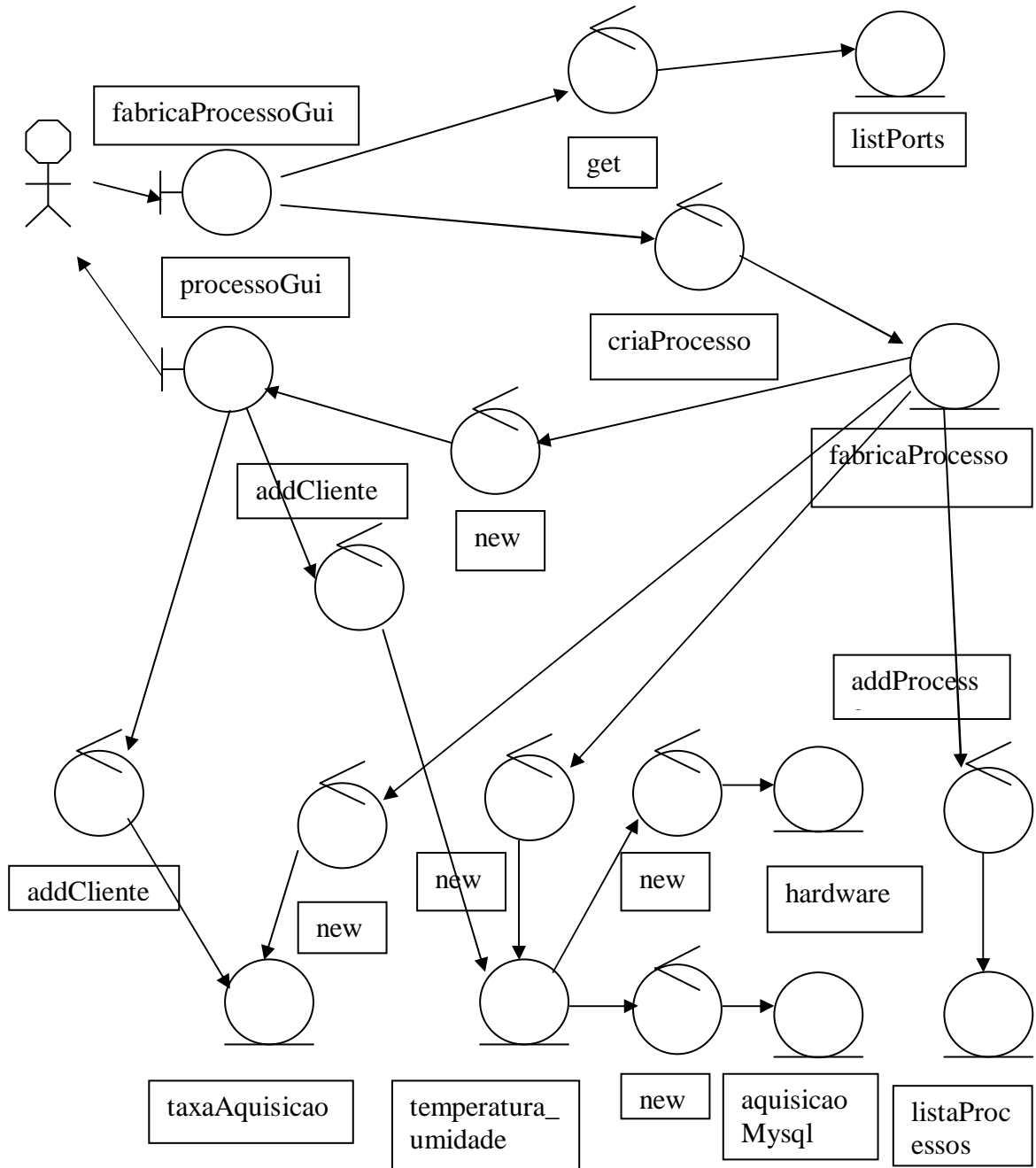


**Figura 4 Índice da representação das figuras do diagrama UML.**

---

#### 4.2.1 Início de aquisição

A análise de robustez do caso de uso início de aquisição é mostrada na Figura 5.



**Figura 5 Análise de Robustez do Início de Aquisição.**

#### 4.2.2 Alteração de taxa de aquisição do criador de processos

A análise de robustez do caso de uso de alteração de taxa de aquisição do criador de processos é mostrada na Figura 6.

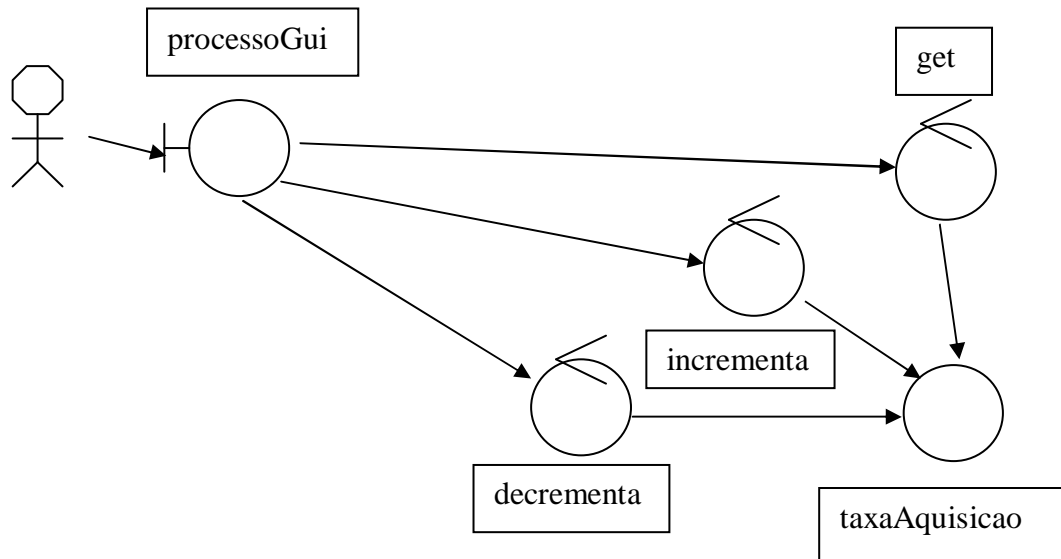


Figura 6 Análise de Robustez da Alteração de Taxa de Aquisição do criador de Processos.

#### 4.2.3 Alteração de taxa de aquisição do processo local

A análise de robustez do caso de uso de alteração de taxa de aquisição do criador de processos é mostrada na Figura 7.

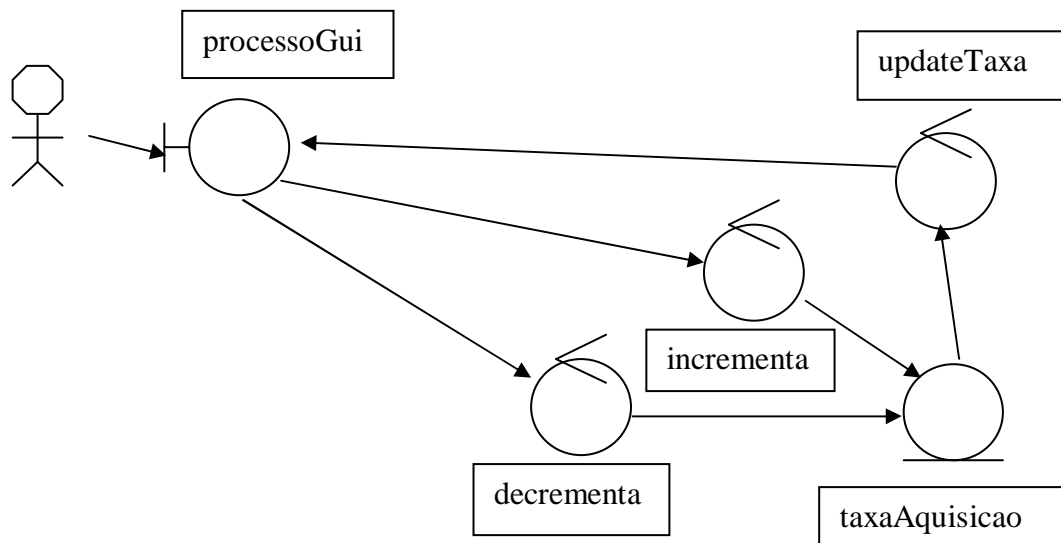


Figura 7 Análise de Robustez da Alteração de Taxa de Aquisição do Processo Local.



#### 4.2.4 Alteração de taxa de aquisição do processo via Web

A análise de robustez do caso de uso de alteração de taxa de aquisição do processo via Web é mostrada na Figura 8.

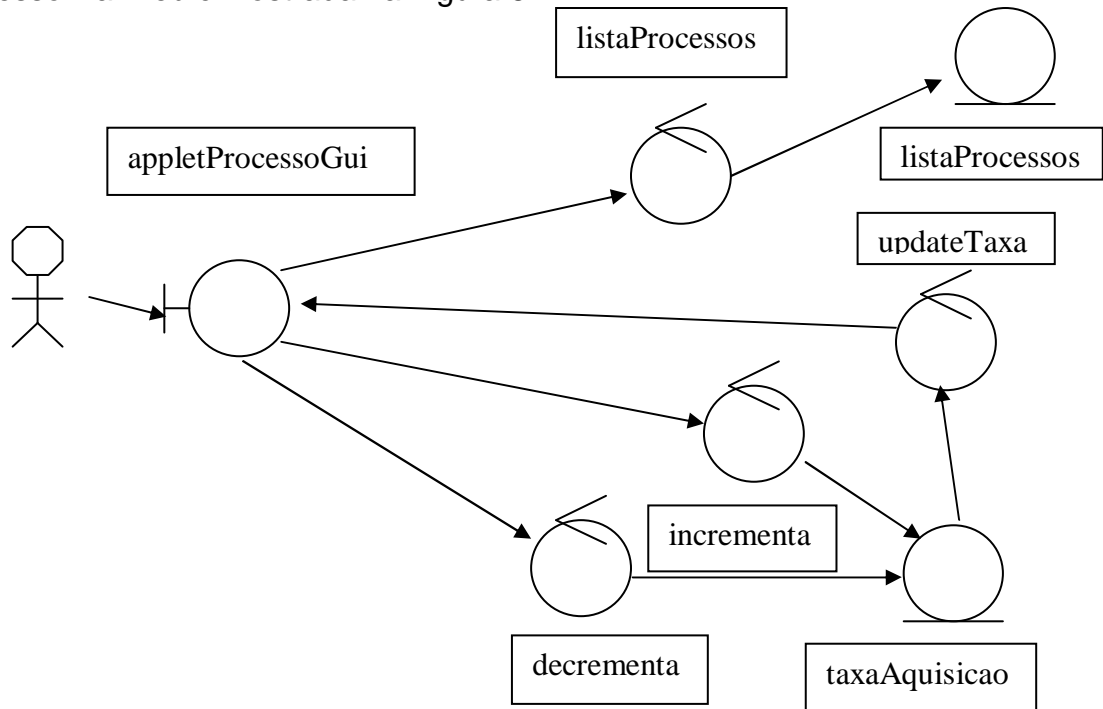
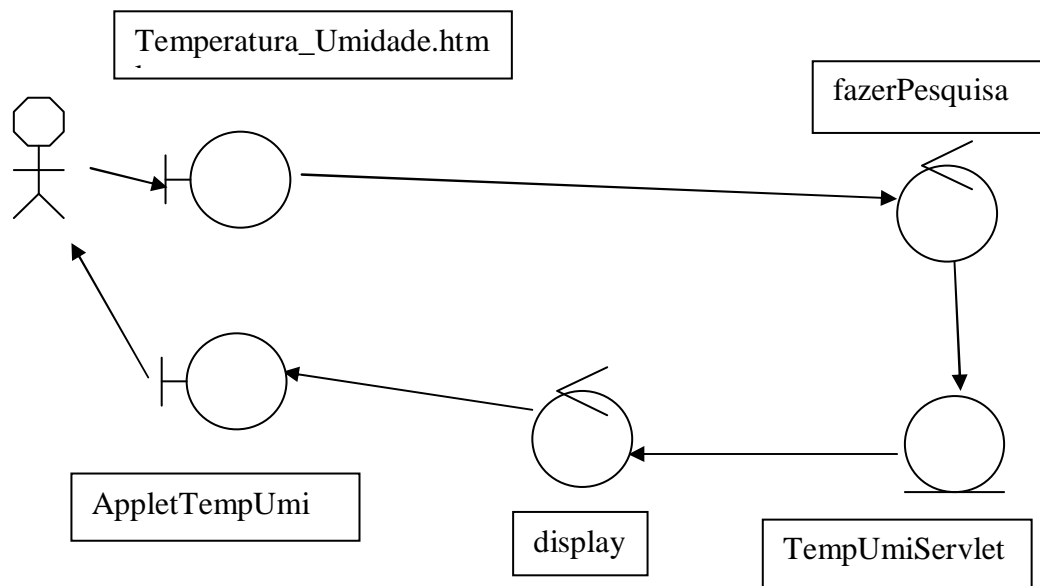


Figura 8 Análise de Robustez da Alteração de Taxa de Aquisição do Processo via Web.

#### 4.2.5 Obtenção de temperatura e umidade média, máxima e mínima e um gráfico temporal de um determinado período.

A análise de robustez do caso de uso de obtenção de temperatura e umidade média, máxima e mínima e um gráfico temporal de um determinado período é mostrada na Figura 9.



**Figura 9** Análise de Robustez da Obtenção de Temperatura e Umidade média, máxima, mínima e um gráfico temporal de um determinado período.

#### 4.2.6 Ciclo de aquisição

A análise de robustez do caso de uso do ciclo de aquisição é mostrada na Figura 10.

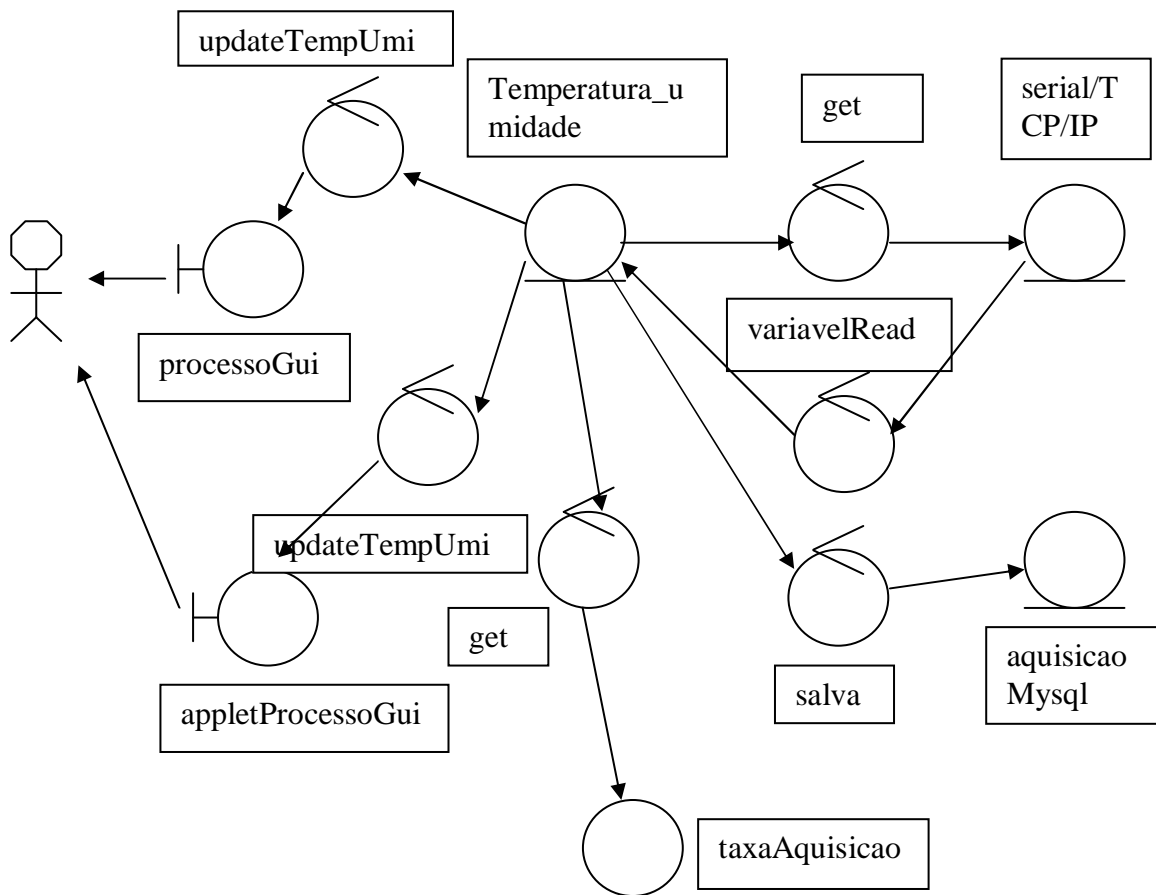


Figura 10 Análise de Robustez do Ciclo de Aquisição.

### 4.3 Modelagem de interação

Extraíu-se dos casos de uso obtidos uma Modelagem de Interação e representou-se através dos diagramas de classe UML. Obteve-se a representação do funcionamento do *software* sobre a seqüência de acontecimentos em relação ao tempo.

### 4.3.1 Início da aquisição

A modelagem de interação do caso de uso do início da aquisição é mostrada na Figura 11.

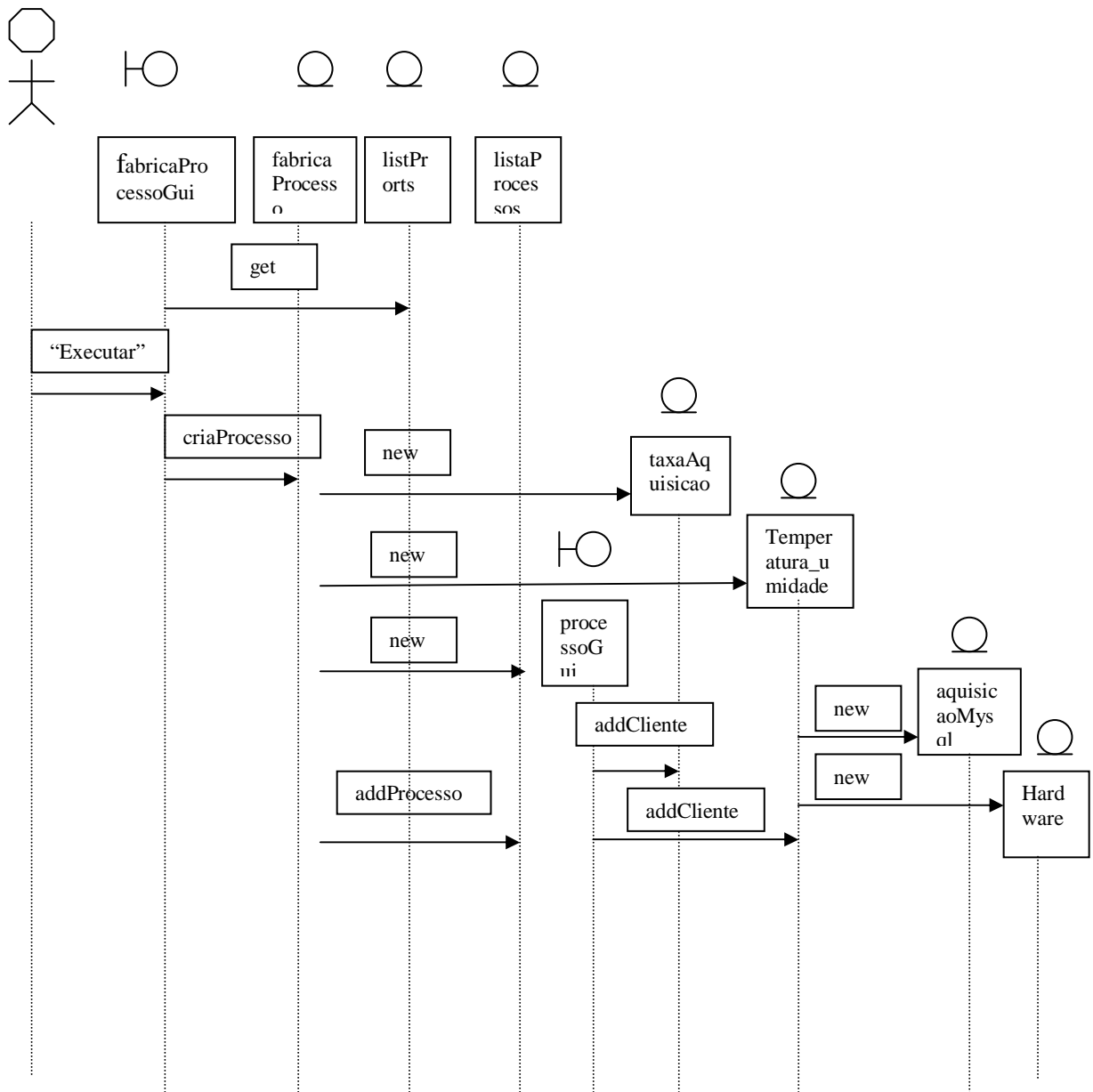


Figura 11 Modelagem de Interação do Início de Aquisição.

### 4.3.2 Alteração de taxa de aquisição do criador de processos

A modelagem de interação do caso de uso da alteração de taxa de aquisição do criador de processos é mostrada na Figura 12.

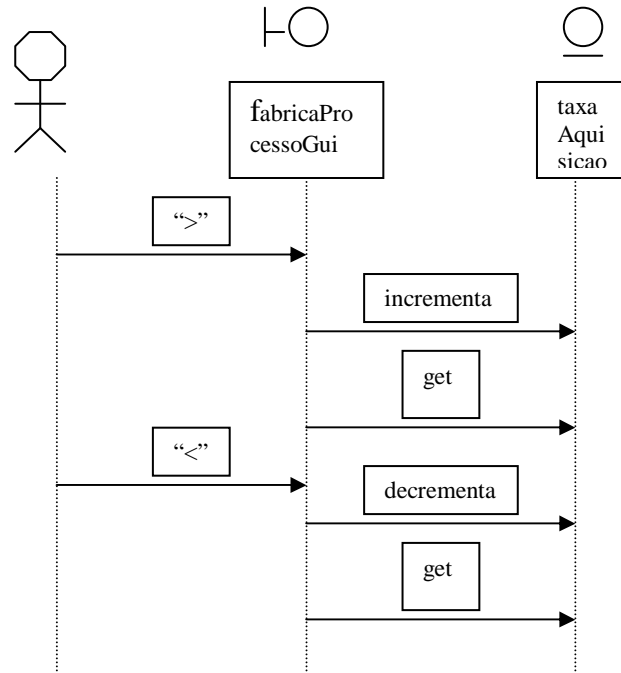


Figura 12 Modelagem de Interação da Alteração de Taxa de Aquisição do criador de Processos.

### 4.3.3 Alteração de taxa de aquisição do processo local

A modelagem de interação do caso de uso da alteração de taxa de aquisição do processo local é mostrada na Figura 13.

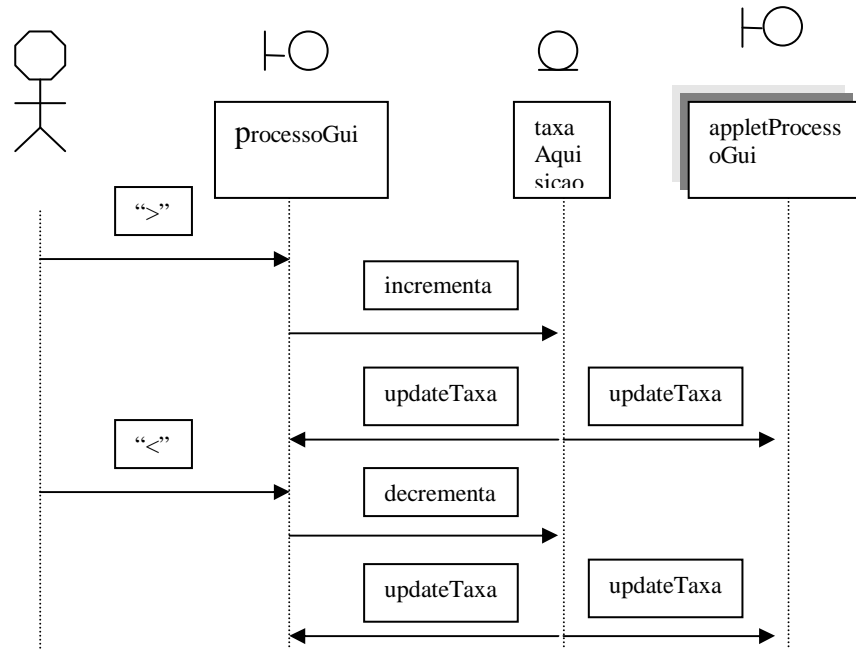


Figura 13 Modelagem de Interação da Alteração de Taxa de Aquisição do Processo Local.

#### 4.3.4 Alteração de taxa de aquisição do processo via Web

A modelagem de interação do caso de uso da alteração de taxa de aquisição do processo via Web é mostrada na Figura 14 .

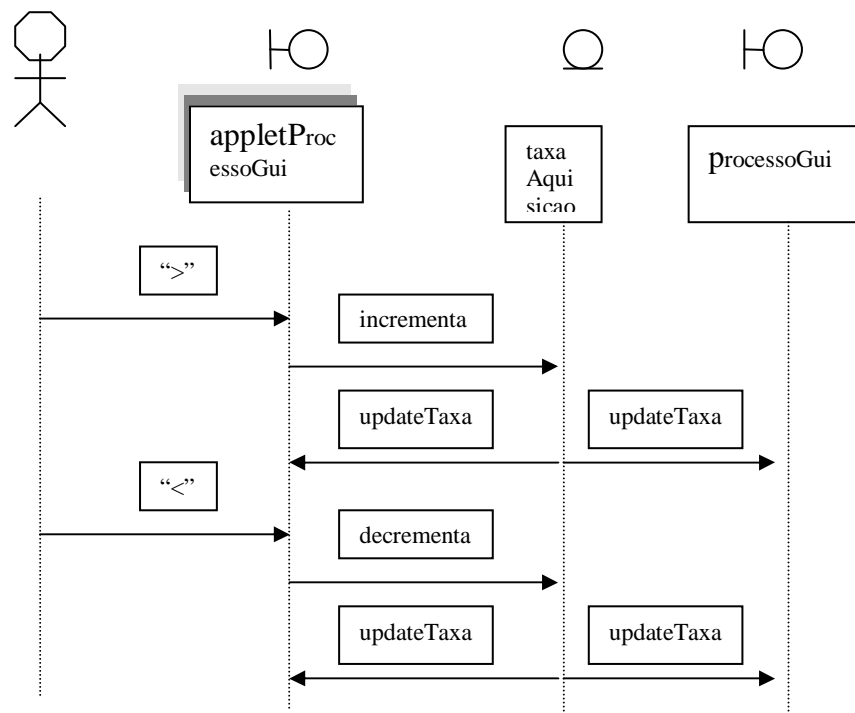


Figura 14 Modelagem de Interação da Alteração de Taxa de Aquisição do Processo via Web.

#### 4.3.5 Obtenção de temperatura e umidade média, máxima e mínima e gráfico temporal de um determinado período

A modelagem de interação do caso de uso da obtenção de temperatura e umidade média, máxima e mínima e o gráfico temporal de um determinado período é mostrada na Figura 15.

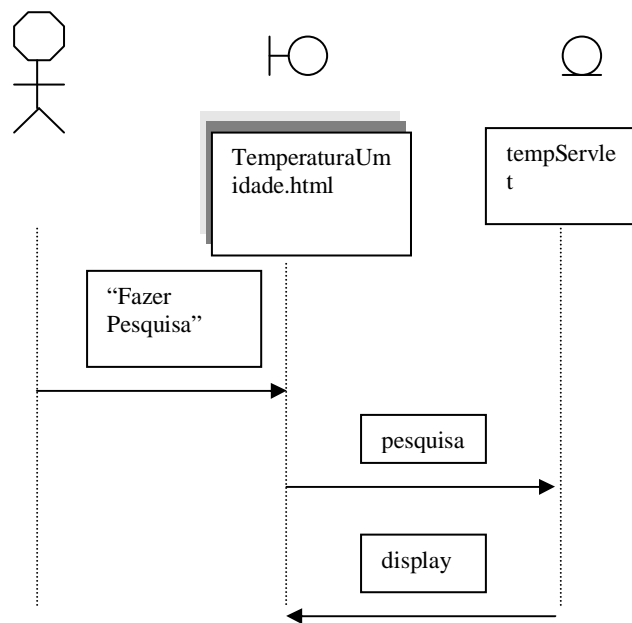


Figura 15 Modelagem de Interação da Obtenção de Temperatura e Umidade média, máxima e mínima e gráfico temporal de um determinado período.

#### 4.3.4 Ciclo de aquisição

A modelagem de interação do caso de uso do ciclo de aquisição é mostrada na Figura 16.

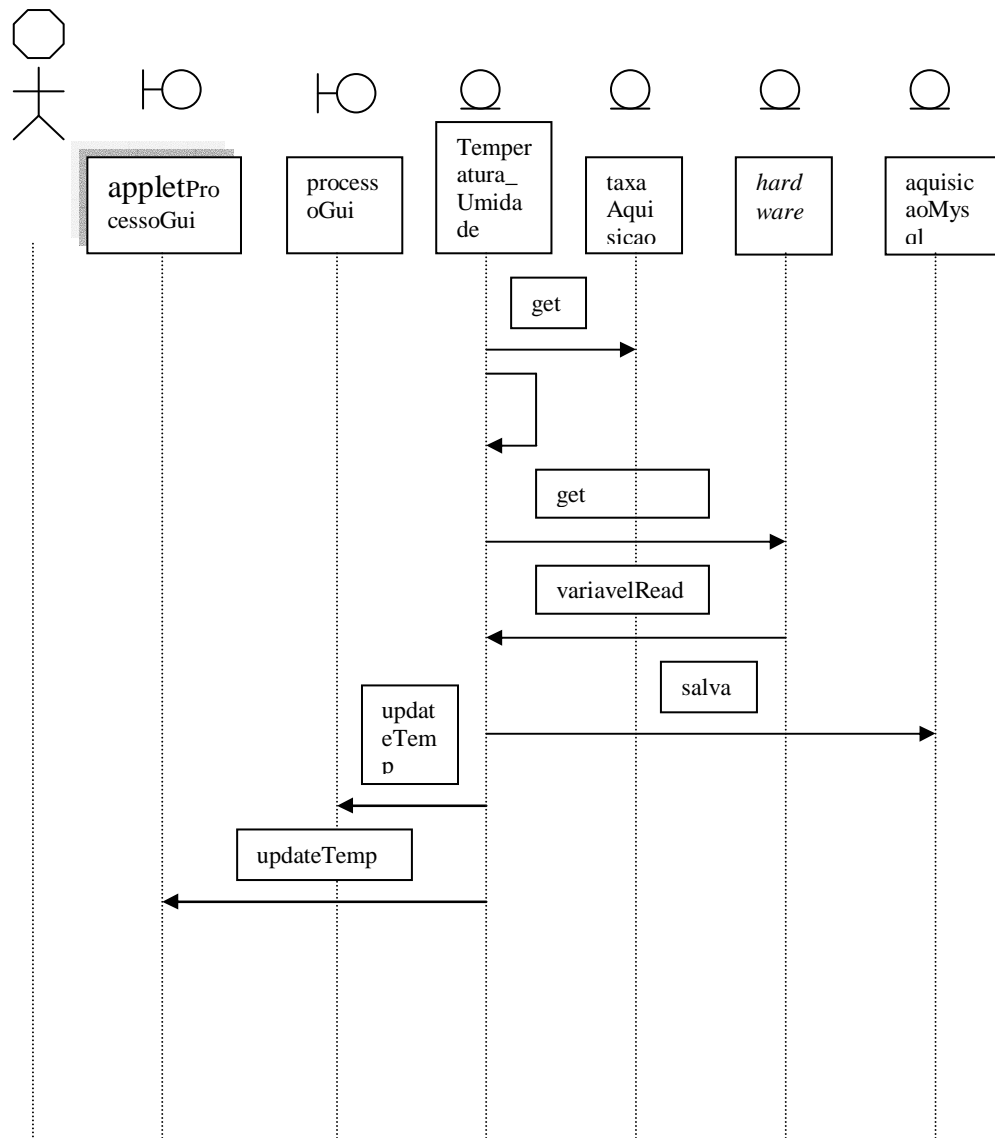


Figura 16 Modelagem de Interação de Ciclo de Aquisição.



## 4.4 Especificação do sistema

Dessa maneira obteve-se a seguinte especificação do sistema:

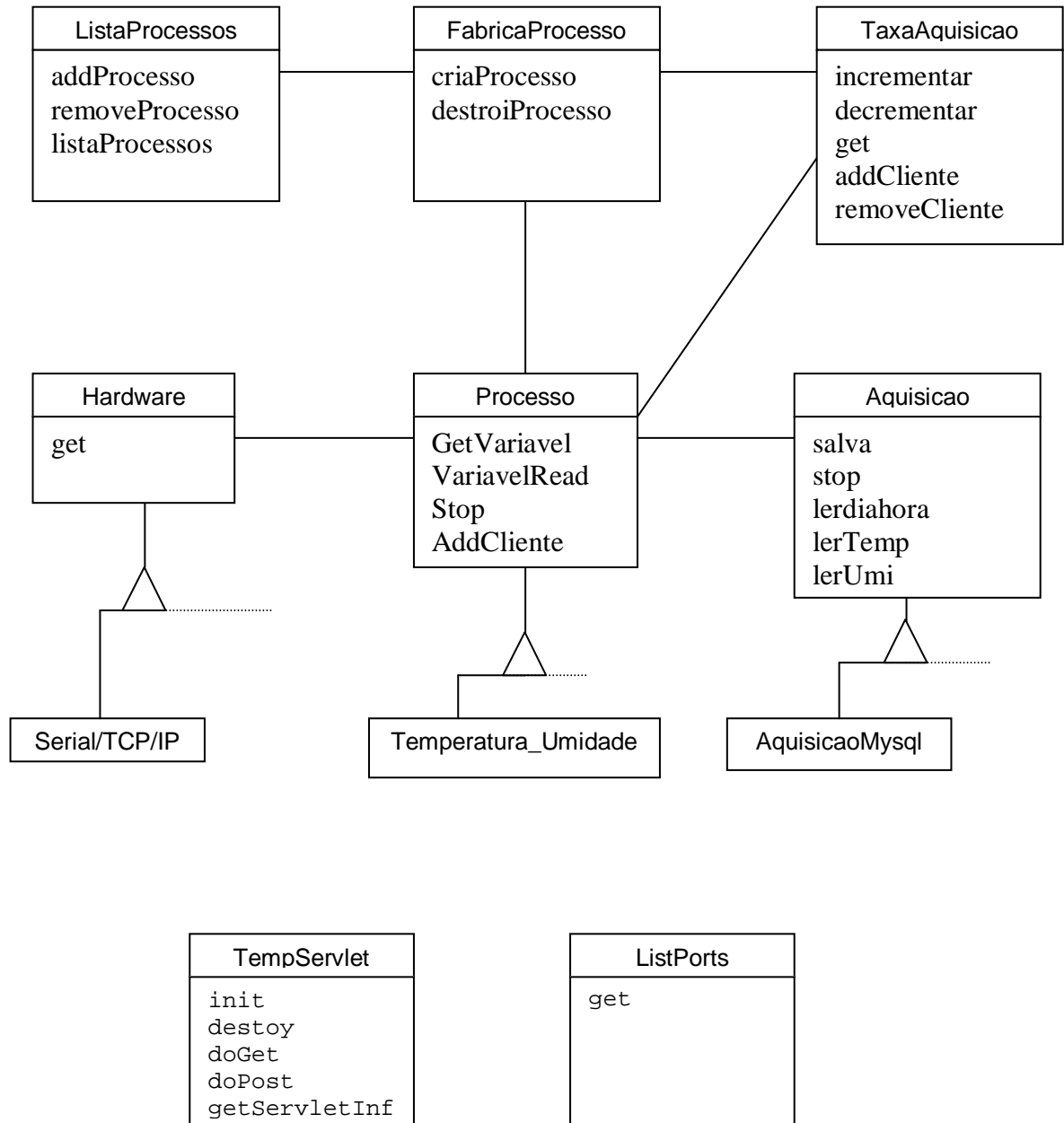


Figura 17 Especificação do Sistema.

Foi adotada uma modelagem mais flexível, com a identificação das principais categorias de elementos do sistema que correspondem a seus níveis de *hardware*:

- o nível mais baixo, das placas de interface (TCP/IP, Serial);
- o nível de processo, variável de temperatura, umidade;
- o nível da aquisição, banco de dados, arquivo;
- o nível mais alto, dos processos de monitoração e controle.

Utilizou-se uma arquitetura cliente-servidor para implementar uma arquitetura distribuída, clientes e servidor com um protocolo pré-estabelecido de troca de mensagens para a solicitação e resposta ao serviço. Utilizou-se um sistema de objetos distribuídos que permite a operação com objetos remotos. Desta maneira é possível, a partir de uma aplicação cliente orientada a objetos, obter uma referência para um objeto que oferece o serviço desejado e, através de uma referência, invocar métodos desse objeto, mesmo sem saber onde está a instância desse objeto, que pode estar em uma outra máquina que não a do objeto cliente.

Implementou-se uma aplicação cliente-servidor TCP/IP utilizando o padrão “Fabrica” para a criação do Processo. Este padrão é útil para se construir objetos individuais, para um mesmo propósito, sem que a construção requeira conhecimento das classes específicas que estão sendo instanciadas. A classe *FabricaProcesso* cria o Processo com padrão *Call Back*. O servidor que está registrado no *rmiregistry* deve implementar uma interface remota que permita que o cliente obtenha uma referência remota para o serviço desejado no serviço de registro RMI.

Desta maneira, quando cada módulo de aquisição e controle estiver sendo executado em qualquer máquina, o serviço será cadastrado no servidor de serviços para que o usuário remoto consiga utilizá-lo, mesmo não sabendo onde se encontra o computador que faz a aquisição e controle do equipamento no qual ele requer o serviço.

O instrumento de medição de temperatura e umidade deve estar conectado fisicamente ao microcomputador PC para que, através deste, ele possa ser adquirido e controlado. Desta maneira será criado um módulo de aquisição de temperatura e umidade que conterà o *driver* de comunicação com o equipamento e os possíveis protocolos de comunicação com o mesmo.

---

Como a leitura de temperatura e umidade com o equipamento é feita via interface ethernet, será utilizada a API Jamod que fornece a conexão TCP/IP com o instrumento e disponibiliza o protocolo Modbus para a comunicação assíncrona mestre-escravo com a busca da variável de temperatura e umidade em determinados registros do equipamento.

Dessa maneira será possível criar cada módulo de aquisição independentemente do outro módulo, sendo que cada módulo faz o controle e a aquisição dos dados do equipamento correspondente.

Utilizou-se um gerenciador de banco de dados para armazenar as medidas. Desta maneira será possível também ter acesso a um histórico de variáveis de medida no tempo. O banco de dados utilizado é o MySQL, por ser multiusuário e poder ser instalado em máquinas com plataforma Linux e Windows. O *driver* utilizado para a comunicação do SGDB do MySQL com o JDBC é o *mysql-connector*.

Implementou-se uma arquitetura de distribuição de três camadas para a pesquisa ao banco de dados usando *servlet* e JDBC, sendo o acesso a pesquisa ao banco de dados através de interface gráfica via Navegador Web. Utilizou-se o servidor web J2EE (*Java 2 Enterprise Edition*) para disponibilizar o *servlet*. A execução e manipulação de resultados a consulta SQL foi feita através da API JDBC (*Java Database Connectivity*).

Com uma arquitetura de objetos distribuídos, através de um Navegador Web carregando um *applet*, o cliente pode verificar qual é a última medida realizada e modificar sua taxa de aquisição. Como o cliente local ou outro cliente via web pode modificar a taxa de aquisição, será implementado o padrão *Call Back* para a taxa de aquisição e valor da temperatura e umidade.

O servidor Web que se utilizou para disponibilizar o *applet* para o Navegador Web é o servidor de código aberto Tomcat Apache.

---

## 5 RESULTADOS

Na Figura 18 é mostrada a montagem experimental em bancada para o teste da ferramenta de monitoração remota de variáveis de ambiente.

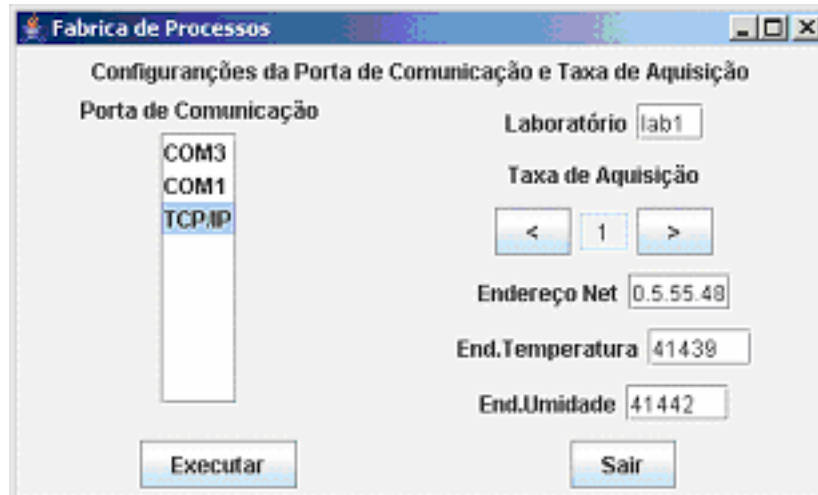


**Figura 18** Montagem experimental em bancada.

Utilizou-se o notebook como servidor local conectado ao instrumento, e também como cliente remoto através do navegador Web.

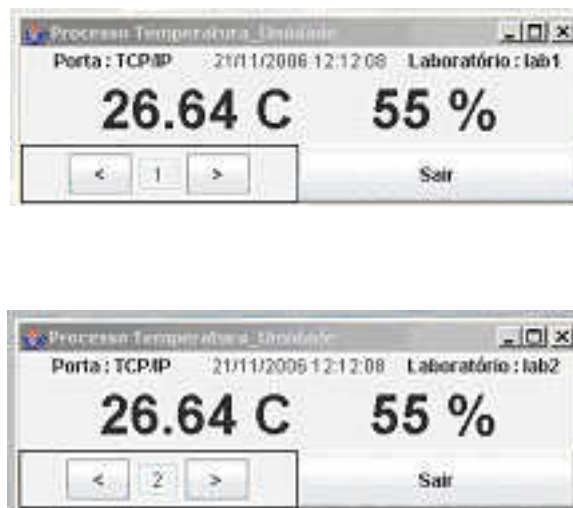
Primeiramente inicializou-se o servidor RMI, o Tomcat Apache e o MySQL no notebook.

Na parte servidora, iniciando-se o FabricaProcesso, seleciona-se o tipo de comunicação com o instrumento, o nome do laboratório e o valor da taxa de aquisição, o endereço do registro de temperatura e o endereço do registro de humidade e sendo a porta de comunicação (TCP/IP) selecionada necessita-se do endereço IP do instrumento. Após a configuração, seleciona-se o botão “Executar”, como mostrado na Figura 19.



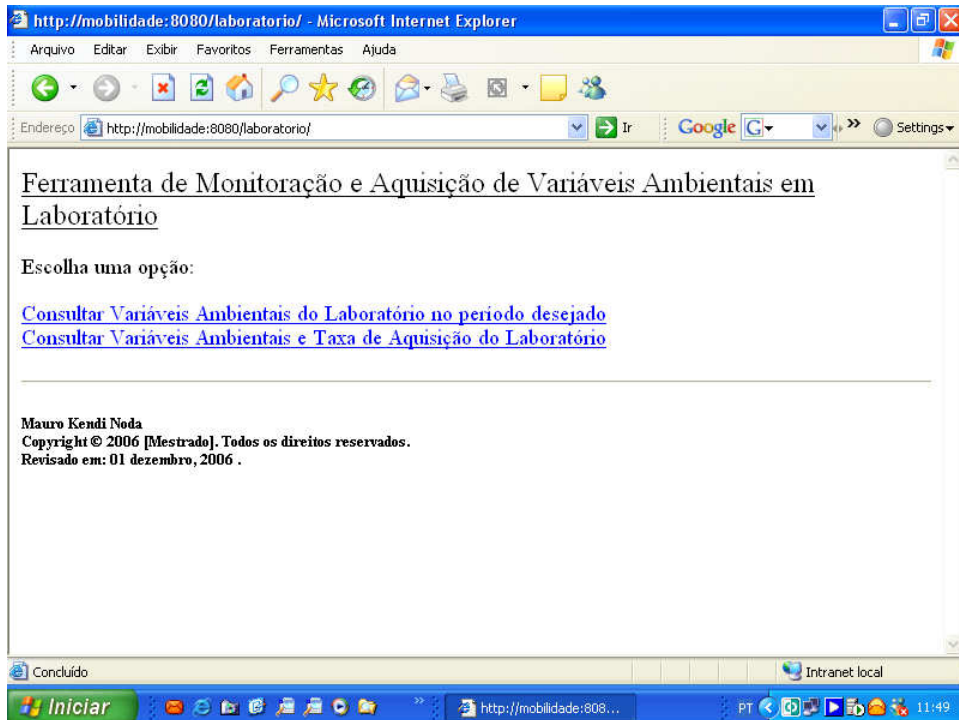
**Figura 19** Tela de configuração do processo.

No presente teste, inicializaram-se dois processos. O processo lab1 possui taxa de aquisição de 1 amostra/minuto. O processo lab2 possui taxa de 2 amostras/minuto. Os processos são mostrados na Figura 20.



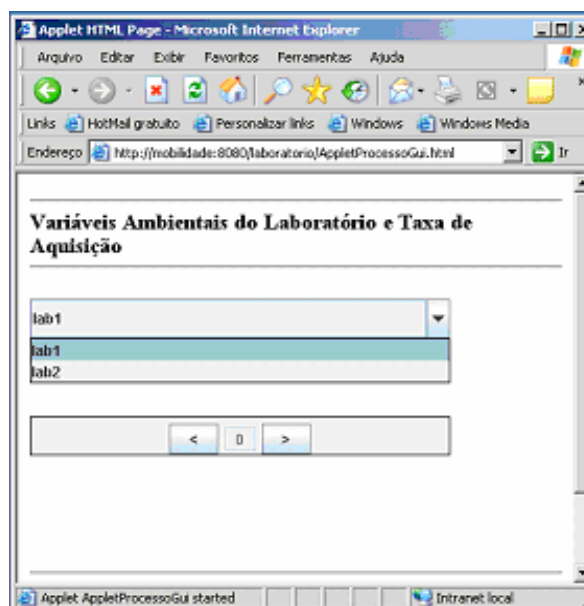
**Figura 20** Processos lab1 e lab2.

Na parte remota, inicia-se o navegador Web acessando o aplicativo laboratório cadastrado no servidor Web Tomcat Apache, através do endereço <http://mobilidade:8080/laboratorio>, como mostrado na Figura 21.



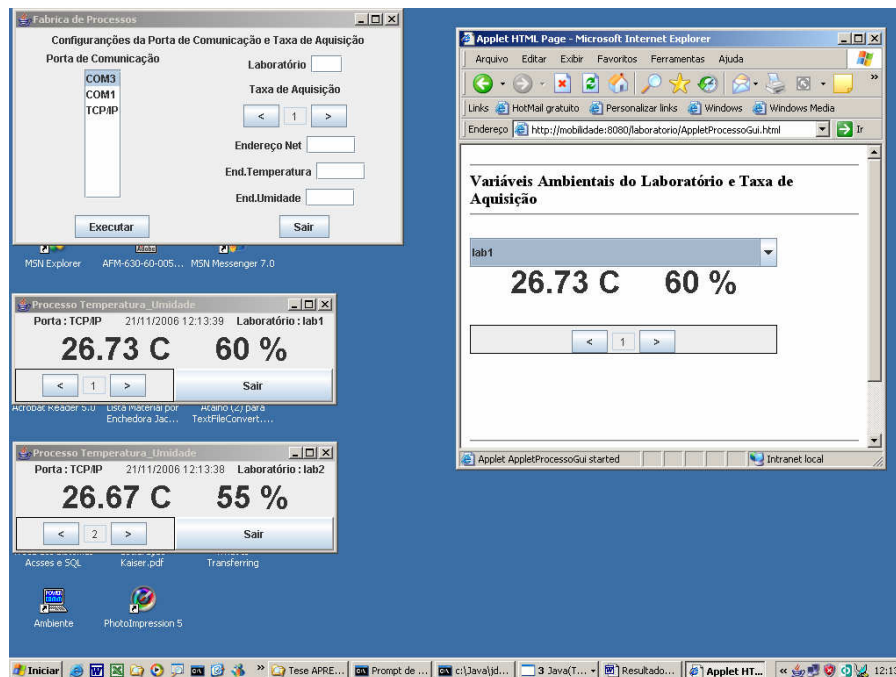
**Figura 21** Tela principal do aplicativo Web.

Selecionando-se a opção “Consultar variáveis ambientais e taxa de aquisição do laboratório”, mostrado na Figura 21. Desta maneira, é apresentada a tela de monitoração remota de variáveis ambientais. Nesta tela, seleciona-se o laboratório no qual se deseja monitorar ou modificar a taxa de aquisição, como é mostrado na Figura 22.



**Figura 22** Tela de monitoração remota.

Desta maneira, os aplicativos a esquerda da tela da Figura 23 são o FabricaProcesso (na parte superior) e os processos lab1 e lab2 (abaixo). A janela do lado direito é o navegador Web acessando remotamente o processo lab1. Através desta seleção, consegue-se monitorar remotamente as variáveis de ambiente referentes àquele processo, mesmo existindo mais processos sendo executados.



**Figura 23** FabricaProcesso, lab1, lab2 do lado esquerdo da tela e monitoração remota de lab1 do lado direito da tela.

Alterando-se a taxa de aquisição de forma remota ou local, verificou-se que, em ambos os casos, todas as aplicações recebem a mudança de valor, como mostrado em relação ao processo lab2 na Figura 24.

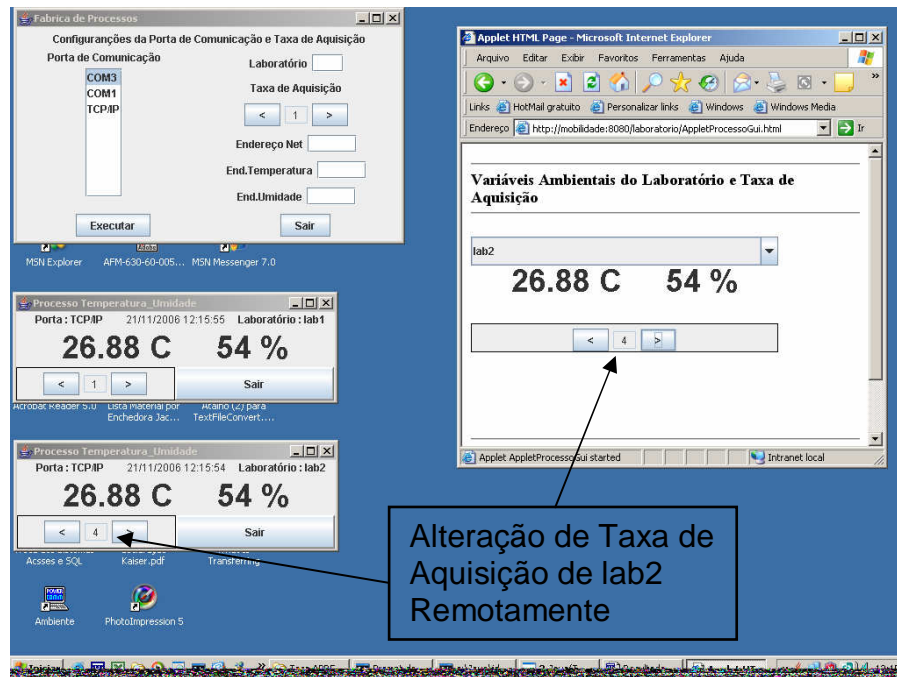


Figura 24 Modificação da taxa remota de monitoração lab2 do lado direito da tela.

Utilizando-se um programa auxiliar para manipulação de banco de dados, verificaram-se os dados armazenados pelo programa de monitoração. Confirma-se, conforme é mostrado na Figura 25 que os dados estão sendo armazenados conforme a taxa de aquisição selecionada para cada um dos processos lab1 e lab2.

laboratorio	dataensaio	temperatura	umidade
lab1	2006-11-21 14:31:57	26.74	54
lab2	2006-11-21 14:32:47	26.75	54
lab1	2006-11-21 14:32:58	26.76	54
lab1	2006-11-21 14:33:59	26.80	54
lab2	2006-11-21 14:34:49	26.84	54
lab1	2006-11-21 14:34:59	26.84	54
lab1	2006-11-21 14:35:59	26.87	54
lab2	2006-11-21 14:36:51	26.88	54
lab1	2006-11-21 14:36:59	26.89	53
lab1	2006-11-21 14:37:59	26.90	53
lab2	2006-11-21 14:38:51	27.33	85
lab1	2006-11-21 14:39:00	27.23	66
lab1	2006-11-21 14:40:00	27.86	79
lab2	2006-11-21 14:40:51	28.35	75
lab1	2006-11-21 14:41:00	28.30	65
lab1	2006-11-21 14:42:00	28.05	54
lab2	2006-11-21 14:42:52	27.83	53
lab1	2006-11-21 14:43:02	27.80	53
lab1	2006-11-21 14:44:02	27.65	53
lab2	2006-11-21 14:44:53	27.57	53
lab1	2006-11-21 14:45:04	27.55	53

Figura 25 Dados armazenados no banco de dados.



Pode-se também selecionar “Consultar variáveis ambientais do laboratório no período desejado” na tela principal do aplicativo Web, como mostrado na Figura 21. Neste caso, realiza-se uma pesquisa para o laboratório lab1 em um período de tempo em que se deseja verificar as médias, valores máximo e mínimo das variáveis de ambiente e um gráfico temporal, mostrada na Figura 26.

Selecione o laboratório - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://mobilidade:8080/laboratorio/Temperatura.html

## Ferramenta de Monitoração de Variáveis Ambientais

Nome do laboratório ?

Selecione o período, dia e hora inicial e dia e hora final, para a pesquisa ?

dia inicial : 21 / 11 / 2006 hora inicial : 14 :00

dia final : 21 / 11 / 2006 hora final : 15 :00

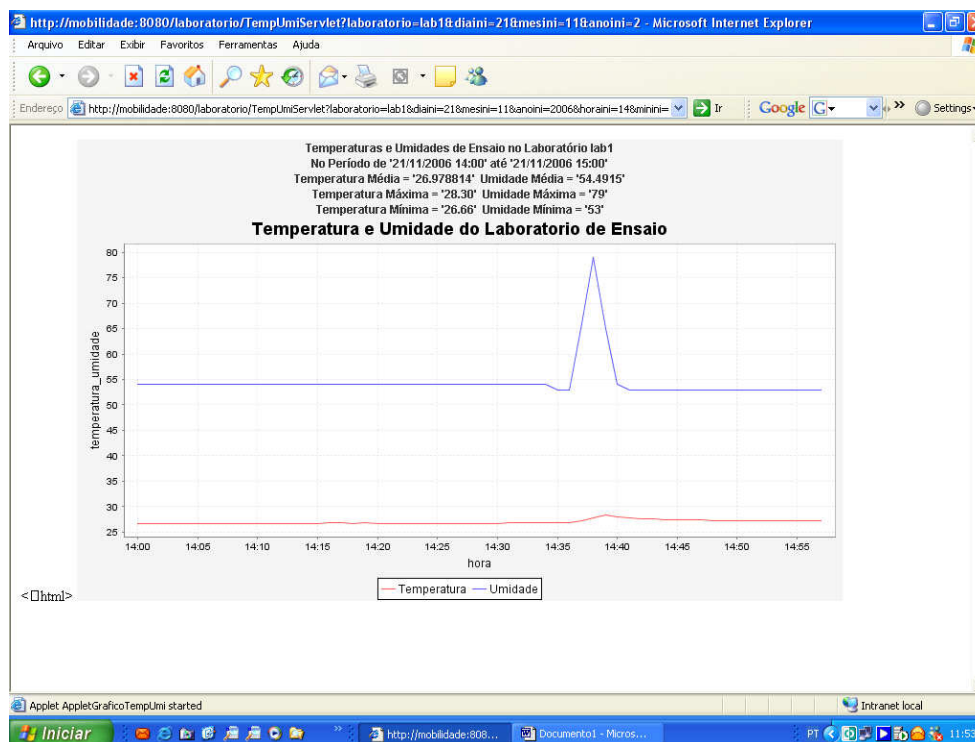
Fazer pesquisa Limpar dados

Mauro Kendi Noda  
Copyright © 2006 [Mestrado]. Todos os direitos reservados.  
Revisado em: 01 dezembro, 2006 .

Concluído Intranet local

**Figura 26** Tela de pesquisa de variáveis ambientais em um período de tempo para lab1.

O resultado desta pesquisa é apresentado no navegador Web, como mostrado na Figura 27.



**Figura 27 Resultado da pesquisa por período.**

Comparando-se os resultados aqui apresentados ao trabalho de Paschoal, Jr. (2006), verificou-se uma compatibilidade em relação aos aplicativos baseados na tecnologia Web. Em relação ao servidor local ocorre uma diferença pela opção do uso de um *software* de instrumentação comercial, o LabView, e optou-se no presente trabalho pela utilização de uma ferramenta *Open Source* que atendeu as expectativas.

O mesmo pode ser dito em relação ao trabalho de Amorim, A.C.O. (2001), através da utilização de DLLs em linguagem de programação Delphi, que estão encapsuladas para serem interpretadas pelos aplicativos de tecnologia Web. Um aperfeiçoamento do trabalho proposto por aquele autor foi a utilização da biblioteca Jcomm para o desenvolvimento do servidor local o que ocorreu com sucesso no presente trabalho.

Sobre o trabalho de Tannuri, E.A.; Noda, M.K., Romano, R. (2004), verificou-se a utilização da linguagem de programação Delphi para a automação e controle das medidas adquiridas em laboratório, mas não ocorrendo nenhuma interação com as tecnologias Web para uma monitoração ou controle remoto através da Internet ou Intranet.

## 6 CONCLUSÕES

Neste trabalho desenvolveu-se um aplicativo para monitoramento e controle remoto de variáveis ambientais que pode ser aplicado em laboratórios de calibração e ensaio.

O desenvolvimento foi feito através de programação orientada a objetos, como o qual as definições das funcionalidades e requisitos de cada objeto ficam mais claras e objetivas, e a estrutura do aplicativo torna-se mais simples de ser compreendida.

Utilizou-se desenvolvimento dirigido por casos de uso, e verificou-se que o desenvolvimento do aplicativo fica dirigido para os requisitos da aplicação, o que o usuário espera do aplicativo e o que o aplicativo oferece ao usuário.

O aplicativo foi concebido para que todo o sistema seja executado no “*localhost*” (computador local). Com algumas modificações, pode-se alocar o RMI, o servidor web e o banco de dados em outra máquina, e a parte servidora do aplicativo ligada localmente com o equipamento. Por fim o cliente poderá utilizar a ferramenta remota de qualquer outra máquina da rede. Desta maneira haverá um melhor balanceamento da rede.

Utilizando-se as máquinas ociosas e a rede intranet, pode-se automatizar e monitorar as medidas de ambiente dos ensaios mais rapidamente e com menos incertezas de manuseio.

Verificou-se que todo o sistema pode ser projetado e desenvolvido com ferramentas *Open Source*, verificando-se uma funcionalidade adequada para as variáveis de ambiente. Para processos com variáveis rápidas é necessário verificar uma melhor funcionalidade da rede Ethernet devido a largura de banda e o tráfego da rede.

Uma extensão a esse trabalho seria desenvolver mais subclasses de *hardware*, processo e aquisição para outros instrumentos de monitoração de variáveis ambientais.

O aplicativo foi testado com um indicador de temperatura e umidade comprovando seu funcionamento.

---

## REFERÊNCIAS BIBLIOGRÁFICAS

AMORIM, A.C.O. **Automação de medidas elétricas monitoradas e controladas remotamente através da Internet**. 2001. 67f. Dissertação (Mestrado em Engenharia de Computação), Instituto de Pesquisas Tecnológicas do Estado de São Paulo, 2001. Disponível em:

< <http://bdtd.ibict.br/busca/resultSimples.jsp> > ou < [http://santafe.ipt.br/tede/tde\\_busca/arquivo.php?codArquivo=8](http://santafe.ipt.br/tede/tde_busca/arquivo.php?codArquivo=8) >. Acesso em: 15 set., 2006.

CZAPSKI, J. Useful features of automated test systems in the R&D laboratory. In: Autotestcon, 36., 2000, Anaheim, California. **Proceedings...** USA: IEEE , 2000. p.601-613. ( IEEE Systems Readiness Technology Conference : Future sustainment for military and aerospace). Disponível em : < <http://www.automeasure.com/features.htm> >. Acesso em :16 set. 2006.

DRNOVSEK, J.; BOJKOVSKI, J.; PUSNIK, I. et al. Automation of a precision temperature calibration laboratory. In: IEEE Instrumentation and Measurement Technology Conference - IMCT 99, 16., 1999. Venice, Italy. **Proceedings...** USA: IEEE, 1999. v.3, p.1883-1887.

GALLARDO, D. **Getting started with the Eclipse Platform**: using Eclipse plug-ins to edit, compile, and debug your app. New York : IBM, 2002. Disponível em: < <http://www-128.ibm.com/developerworks/java/library/os-ecov/?t=egrL262&p=statedclipseplat> >. Acesso em: 22 set. 2006

NASCIMENTO, G. R.; CÂMARA, A.P. Retorno de investimento em automação de laboratórios. In: Encontro para a Qualidade de Laboratórios Rede Metrológica do Estado de São Paulo, 2005 . São Paulo. **Anais eletrônico....** São Paulo : REMESP, 2005. Disponível em: < <http://www.cali.com.br/artigos/Retorno%20de%20investimento%20em%20automacao%20de%20laboratorios.pdf#search=%22%22Retorno%20de%20investimento%20em%20automa%C3%A7%C3%A3o%20de%20laborat%C3%B3rios%22%22> > . Acesso em: 22 set. 2006

OLIVEIRA, R.O.O.; ALVES, H.A.; SOARES, D.C. et al. Estudo de viabilidade para implantação de sistemas de automação e informatização dos laboratórios da metrologia legal. In: Congresso Brasileiro de Metrologia,3., 2003. Pernambuco, Brasil. **Anais eletrônicos...** S.l.:SBA, 2003. Disponível em: <[http://www.metrologia2003.org.br/anais\\_congresso/MA0020.pdf#search=%22%22Estudo%20de%20viabilidade%20para%20implanta%C3%A7%C3%A3o%20de%20sistemas%20de%20automa%C3%A7%C3%A3o%20e%20informatiza%C3%A7%C3%A3o%20dos%20%22%22](http://www.metrologia2003.org.br/anais_congresso/MA0020.pdf#search=%22%22Estudo%20de%20viabilidade%20para%20implanta%C3%A7%C3%A3o%20de%20sistemas%20de%20automa%C3%A7%C3%A3o%20e%20informatiza%C3%A7%C3%A3o%20dos%20%22%22)>. Acesso em: 15 set. 2006-09-16

OBJECT Refinery Limited. **The Java Free Chart API JFreeChart and Jcommon**, Version 0.8.7 – 10. Disponível em < <http://www.jfree.org/index.php> >. Acesso em 22 set. 2006

PASCHOAL JUNIOR, F. **Desenvolvimento de um WEBLAB para experimentos remotos em fotônica integrado a um ambiente virtual de ensino-aprendizagem.**

2006. 107f. Dissertação (Mestrado em Engenharia Elétrica), Universidade Presbiteriana Mackenzie, 2006. Disponível em:

< [www.kyatera.fapesp.br/portal/V.public/teses/WebLab\\_FabioPaschoal.PDF](http://www.kyatera.fapesp.br/portal/V.public/teses/WebLab_FabioPaschoal.PDF) >

. Acesso em: 15 set., 2006.

PRESSMAN, R. S., **Engenharia de Software**, São Paulo : Makron Books, 1995.

RAMOS, H.G.; TRINDADE, M.; CRUZ SERRA, A. A measurement system to automate laboratory testing in soils. In: IEEE Instrumentation and Measurement Technology Conference -IMCT 97, 1997. Ottawa , Canada . **Proceedings...** USA: IEEE, 1997. v.2, p.1048-1051.

RICARTE, I.L.M. Um processo de desenvolvimento orientado a objetos.

DCA/FEEC/UNICAMP, 2003. Disponível em:

<<http://www.dca.fee.unicamp.br/cursos/PooJava/desenvolvimento/processo.html>>.

Acesso em 22 set. 2006

SALVATTI, M.; GALIB, F. Sistemas de controle e supervisão via Internet. **Revista Controle & Instrumentação**, São Paulo, n.109, out. 2005, p.95-101.

SourceForge, **The Java Modbus API Jamod**, Version 1.2 rc1 SourceForge.net, 2000 – 2004. Disponível em: < <http://jamod.sourceforge.net/> >. Acesso em 22 set.2006

STEMPNIAK, C.R.; LANDGRAF, W.R. Automatizando laboratórios de calibração e ensaios. **Revista Banas Metrologia & Instrumentação**, São Paulo, nov. 2003.

Disponível em: <

<http://www.banasmetrologia.com.br/textos.asp?codigo=967&secao=revista> >.

Acesso em: 22 set. 2006

Sun Developer Network (SDN) **The Java Communications API Javacomm**,

Version 2.0 USA : Sun Microsystem, Inc., 2003. Disponível em: <

<http://java.sun.com/products/javacomm/index.jsp> >. Acesso em 22 set. 2006

SUN Microsystem, Inc. **The Java Tutorial**: A practical guide for programmers. USA :

Sun Microsystem, Inc., 2003. Disponível em:

<<http://java.sun.com/docs/books/tutorial/information/download.html>> . Acesso em: 22 set. 2006

SUN Microsystem, Inc. **Java™ 2 SDK**, Standard Edition Documentation, Version 1.5.0. USA : Sun Microsystem, Inc., 2003. Disponível em: <

<http://java.sun.com/j2se/1.5.0/docs/index.html> >. Acesso em 22 set. 2006

TANNURI, E.A.; ROMANO, R.; NODA, M.K.. Estudos de casos em automação Laboratorial. In: Congresso internacional de automação e sistemas de

---

instrumentação,4. 2004. São Paulo. **Anais...** São Paulo : Intech, 2004. v. 67. p. 76-76.

## **REFERÊNCIAS CONSULTADAS**

SANTOS, E.B.; SECKLER, M.M. **Guia para elaboração de dissertação de mestrado**. 2.ed. São Paulo: IPT, 2005. 32p.