

DESENVOLVIMENTO E APLICABILIDADE EM EAD DE UM SISTEMA
DE INSTRUMENTAÇÃO VIRTUAL DE ACESSO A DISTÂNCIA

MÁRLON CAETANO RAMOS PESSANHA

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE - UENF
CAMPOS DOS GOYTACAZES -RJ
Fevereiro - 2009

DESENVOLVIMENTO E APLICABILIDADE EM EAD DE UM SISTEMA
DE INSTRUMENTAÇÃO VIRTUAL DE ACESSO A DISTÂNCIA

MÁRLON CAETANO RAMOS PESSANHA

“Dissertação apresentada ao Centro de
Ciência e Tecnologia da Universidade
Estadual do Norte Fluminense, como
parte das exigências para obtenção do
título de Mestre em Ciências Naturais.”

Orientador: Prof. Marcelo de Oliveira Souza

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE - UENF
CAMPOS DOS GOYTACAZES -RJ
Fevereiro – 2009

FICHA CATALOGRÁFICA

Preparada pela Biblioteca do **CCT / UENF**

05/2009

Pessanha, Márton Caetano Ramos

Desenvolvimento e aplicabilidade em EAD de um sistema de instrumentação virtual de acesso a distância / Márton Caetano Ramos Pessanha. – Campos dos Goytacazes, 2009.

vii, 108 f. : il.

Dissertação (Mestrado em Ciências Naturais) --Universidade Estadual do Norte Fluminense Darcy Ribeiro. Centro de Ciência e Tecnologia. Laboratório de Ciências Físicas. Campos dos Goytacazes, 2009.

Orientador: Marcelo de Oliveira Souza.

Área de concentração: Ensino de Ciências.

Bibliografia: f. 104-108

1. Educação a distância 2. Instrumentação virtual 3. Ensino de ciências I. Universidade Estadual do Norte Fluminense Darcy Ribeiro. Centro de Ciência e Tecnologia. Laboratório de Ciências Físicas II.

Título

CDD 371.334

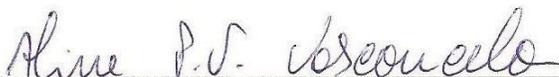
DESENVOLVIMENTO E APLICABILIDADE EM EAD DE UM SISTEMA
DE INSTRUMENTAÇÃO VIRTUAL DE ACESSO A DISTÂNCIA

MÁRLON CAETANO RAMOS PESSANHA

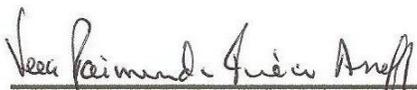
“Dissertação apresentada ao Centro de
Ciência e Tecnologia da Universidade
Estadual do Norte Fluminense, como
parte das exigências para obtenção do
título de Mestre em Ciências Naturais.”

Aprovada em 5 de Fevereiro de 2009:

Comissão Examinadora:



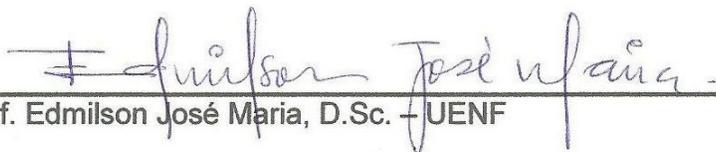
Prof.^a Aline Pires Vieira Vasconcelos, D.Sc. – CEFET-Campos



Prof.^a Vera Raimunda Amério Asseff, D.Sc. – CEFET-Campos



Prof. Delson Ubiratan da Silva Schramm, D.Sc. – UENF



Prof. Edmilson José Maria, D.Sc. – UENF



Prof. Marcelo de Oliveira Souza, D.Sc. – UENF
Orientador

AGRADECIMENTOS

À Deus, pela vida e pela saúde. Por ter me guiado e me dado forças para finalizar mais esta etapa de minha vida.

Aos meus pais, Aloísio e Romirte, que investiram na minha educação, me ensinaram valores, e colaboraram em muito para que me tornasse a pessoa que hoje eu sou.

Ao meu irmão Maikon, pela amizade e companheirismo, e por servir de exemplo de dedicação ao estudo.

À Sabrina, por estar do meu lado nos bons e maus momentos.

Ao Prof. Marcelo Souza por todo apoio, orientação e amizade.

Ao amigo Rafael da Costa, pela ajuda disponibilizando um DNS para os testes iniciais do software servidor.

À Gerência de Recursos Computacionais da UENF, em especial ao funcionário Carlos, pela colaboração, liberando o acesso externo aos softwares servidores.

Aos colegas de disciplinas na graduação e mestrado, Sr. Honoris Tião (Joziel), Sr. Honoris Little Ronald (Ronaldo), Sr. Honoris Léo (Leonardo) e Sr^a. Honoris (Aline); pela amizade, pelas conversas de corredor, e por terem compartilhado comigo tantos momentos.

Ao técnico Luís, do laboratório de eletrônica, pelo auxílio nos testes dos circuitos e pelas dicas sobre componentes e montagens.

Aos colegas de laboratório: Adriana, Davi, Keitty, Juliana, Samara, Rudson, Verônica e Victor Hugo.

À Universidade Estadual do Norte Fluminense e aos professores, pela formação propiciada.

A todos que de forma direta ou indireta colaboraram para o êxito deste trabalho.

“Educar é crescer.
E crescer é viver.
Educação é, assim,
vida no sentido mais
autêntico da palavra”
Anísio Teixeira

SUMÁRIO

	Pág.
LISTA DE FIGURAS	i
LISTA DE TABELAS	iii
LISTA DE SIGLAS	iv
RESUMO	vi
ABSTRACT	vii
1- INTRODUÇÃO	1
1.1- Objetivos	4
1.2- Estrutura da Dissertação	4
2- REVISÃO BIBLIOGRÁFICA	6
3- BASES TEÓRICAS	9
3.1- Redes de Computadores e Internet	9
3.1.1- Histórico da Internet	9
3.1.2- Histórico da Internet no Brasil	10
3.1.3- Protocolos TCP/IP	11
3.2- Educação a Distância	12
3.2.1- Histórico da Educação a Distância	12
3.2.2- Internet e Educação a Distância	14
3.3- Teorias Pedagógicas e Tecnologia da Informação	14
3.3.1- Jean Piaget	15
3.3.2- Lev Vygotsky	16
3.3.3- David Ausubel	18
3.3.4- Seymour Papert	19
3.4- Ensino Experimental	20
3.5- Desenvolvimento de Softwares – Linguagens de Programação	21
3.5.1- Histórico das Linguagens de Programação	23
3.5.2- Linguagens de Programação Orientadas a Objetos	25
3.5.3- Delphi e C++Builder	26
3.5.4- Qualidade de Software	28
3.6- Portas de Comunicação com Hardwares Externos	29
3.6.1- Porta paralela	29
3.6.2- Porta USB	32
4- METODOLOGIA	35
4.1- Primeira Etapa: Definição da Estrutura de Funcionamento do Sistema de Instrumentação Virtual	35
4.2- Segunda Etapa: Desenvolvimento dos Softwares	37
4.2.1- Principais componentes no desenvolvimento dos softwares	37
4.2.2- Software Cliente	39
4.2.3- Software Servidor de Comunicação	43
4.2.4- Software Servidor de Controle – USB	45
4.2.5- Software Servidor de Controle – Paralela	48
4.2.6- Funções comuns entre os softwares desenvolvidos	50
4.3- Terceira Etapa: Configuração do Software Encoder	52
4.4- Quarta Etapa: Configuração e desenvolvimento de Circuitos de Interface	55
4.4.1- Circuito de Interface: Porta USB	55
4.4.2- Circuito de Interface: Porta paralela	58

4.5- Quinta Etapa: Desenvolvimento de elementos para testes	60
4.5.1- Dispositivo de Controle 1: Controle de Cor	60
4.5.2- Dispositivo de Controle 2: Protótipo robótico	62
4.5.3- Experimento 1: Absorção e reflexão da luz por materiais claros e escuros	64
4.5.4- Experimento 2: Aplicação da Lei de Ampère	71
4.5.5- Experimento 3: Indução Eletromagnética de Faraday	72
4.6- Sexta Etapa: Testes de funcionamento do sistema desenvolvido	73
4.6.1- Testes de transmissão de dados	73
4.6.2- Testes com dispositivos de controle e experimentos	75
5 – RESULTADOS	77
5.1- Softwares desenvolvidos	77
5.1.1- Software Cliente	77
5.1.2- Software Servidor de Comunicação	81
5.1.3- Software Servidor de Controle – USB	82
5.1.4- Software Servidor de Controle – Paralela	85
5.2- Testes de funcionamento do sistema	88
5.2.1- Testes de transmissão de dados	88
5.2.2- Testes com dispositivos de controle e experimentos	90
5.3- Análise Técnica	94
5.3.1- Qualidade de software	94
5.3.2- Análise técnica e comparativa entre portas paralela e USB	96
6 – APLICABILIDADE DA FERRAMENTA NO ENSINO A DISTÂNCIA	98
7 – CONCLUSÃO	101
7.1- Considerações Finais	101
7.2- Trabalhos Futuros	102
8 – REFERÊNCIAS	104

LISTA DE FIGURAS

	Pág.
Figura 1 – Evolução histórica das linguagens de programação de alto nível.	25
Figura 2 – Montagem visual de um software em um ambiente que utiliza GUI.	26
Figura 3 – Tela capturada do Delphi.	27
Figura 4 – Pinos do conector DB25.	32
Figura 5 – Ilustração do funcionamento da codificação NRZI.	33
Figura 6 – Pinos do conector USB Tipo A.	34
Figura 7 – Esquema Geral do Sistema.	36
Figura 8 – Esquema da interação entre softwares cliente e servidor.	38
Figura 9 – Esquema de funcionamento do componente IdTCPClient para o software Cliente.	40
Figura 10 – Esquema representativo da função de envio de comandos.	41
Figura 11 – Esquema de funcionamento do componente WebBrowser e arquivos “imagem.toe” e “noconnect.toe”.	42
Figura 12 – Esquema de funcionamento do componente IdTCPServer.	44
Figura 13 – Esquema de funcionamento do componente IdTCPClient e filtro de verificação de comandos.	47
Figura 14 – Representação da lógica da função Carregamento de Esquemas de Cores.	51
Figura 15 – Servidor Encoder: Criando uma nova sessão de transmissão ao vivo.	52
Figura 16 – Servidor Encoder: Configurando origem de imagem.	53
Figura 17 – Servidor Encoder: Configurando do método de difusão de imagem.	53
Figura 18 – Servidor Encoder: Configurando a porta de comunicação em rede.	53
Figura 19 – Servidor Encoder: Configurando taxas de bits e quadros e tamanho da imagem.	54
Figura 20 – Software Encoder em Funcionamento.	54
Figura 21 – Foto da placa R-Control30.	56
Figura 22 – Esquema de montagem do Circuito Auxiliar de Controle.	57
Figura 23 – Foto do Circuito Auxiliar de Controle.	58
Figura 24 – Esquema de montagem do Circuito de Interface (Porta paralela). .	59
Figura 25 – Foto do Circuito de Interface (Porta paralela).	60
Figura 26 – Foto do dispositivo 1.	61
Figura 27 – Esquema do circuito do dispositivo 1.	62
Figura 28 – Emissor de RF e Movimentação possível do protótipo robótico.	63
Figura 29 – Foto do protótipo robótico.	64
Figura 30 – Esquema de montagem do Experimento 1.	65
Figura 31 – Foto do Experimento 1.	65
Figura 32 – Circuito auxiliar para sensores do Experimento 1.	66
Figura 33 – Gráfico $R_T \times T$ para os termistores utilizados.	67
Figura 34 – Gráfico $U_{R10K\Omega} \times T$ para os termistores utilizados.	68
Figura 35 – Gráfico $U_{R10K\Omega} \times T$ (entre 0°C e 40°C), para os termistores utilizados.	69
Figura 36 – Resposta relativa do LDR.	70
Figura 37 – Esquema de montagem do Experimento 2.	71
Figura 38 – Foto do Experimento 2.	71
Figura 39 – Esquema de montagem do Experimento 3.	72

Figura 40 – Foto do Experimento 3.	73
Figura 41 – Esquema representando o teste de transmissão de imagens.	75
Figura 42 – Tela capturada do software para o teste de transmissão de imagem.	75
Figura 43 – Tela capturada do Software Cliente.	77
Figura 44 – Regiões do Software Cliente.	78
Figura 45 – Tela capturada da Seção de Conteúdos Teóricos, acessível através do Software Cliente.	79
Figura 46 – Tela capturada da Seção de ajuda do Software Cliente.	79
Figura 47 – Tela capturada do Software Cliente: Carregamento de esquema de cores.	80
Figura 48 – Tela capturada do Software Cliente: Configuração de Comandos.	80
Figura 49 – Tela capturada do Software Servidor de Comunicação.	81
Figura 50 – Regiões do Software Servidor de Comunicação.	81
Figura 51 – Tela capturada do Software Servidor de Controle (USB).	83
Figura 52 – Regiões do Software Servidor de Controle (USB).	83
Figura 53 – Destaque da região de configuração de dados de estado, do Software Servidor de Controle (USB).	85
Figura 54 – Tela capturada do Software Servidor de Controle (Paralela).	86
Figura 55 – Regiões do Software Servidor de Controle (Paralela).	86
Figura 56 – Destaque da região de configuração de dados de estado, do Software Servidor de Controle (Paralela).	88
Figura 57 – Gráfico representando o tempo de atraso na transmissão de imagens.	89
Figura 58 – Experimento 1: Resposta Relativa dos sensores de luminosidade x tempo - Verificação da luminosidade refletida por cada placa.	91
Figura 59 – Experimento 1: Temperatura x Tempo - Temperatura relativa a absorção de cada placa.	91
Figura 60 – Experimento 2: Detecção de campo magnético em função do tempo.	92
Figura 61 – Experimento 3: Tensão elétrica verificada (1ª Sessão-USB).	93
Figura 62 – Experimento 3: Tensão elétrica verificada (2ª Sessão-Paralela). ...	93
Figura 63 – Mapa dos cursos EaD de Física, Química, Matemática e Ciências.	99

LISTA DE TABELAS

	Pág.
Tabela 1 – Períodos do desenvolvimento mental da Teoria de Piaget	15
Tabela 2 – Exemplos de partes de códigos fonte de linguagens em diferentes níveis	22
Tabela 3 – Características e Sub-características de um software de qualidade	28
Tabela 4 – Distribuição e função dos bits para cada endereço da Porta paralela	30
Tabela 5 – Total de bits para saída e entrada de dados	31
Tabela 6 – Pinos e endereços dos quais pertencem	32
Tabela 7 – Pinos e suas respectivas funções do conector USB tipo A	34
Tabela 8 – Relação Diferença de potencial X Ativação de Saídas do circuito auxiliar de controle	57
Tabela 9 – Componentes do Circuito Auxiliar de Controle	58
Tabela 10 – Componentes do Circuito de Interface (Porta paralela)	59
Tabela 11 – Cores emitidas para cada combinação dos leds	61
Tabela 12 – Componentes do circuito C-D1	62
Tabela 13 – Dados gerais da realização dos testes	76
Tabela 14 – Tipo de dados coletados e de controle	76
Tabela 15 – Intervalo de tempo (ms) entre envio e recepção de mensagens	89
Tabela 16 – Velocidade máxima de tráfego de dados das portas paralela e USB	96
Tabela 17 – Quantidade de cursos EaD em cada área, por estado	98

LISTA DE SIGLAS

ACM	Association for Computing Machinery
AIRDATA	Serviço Internacional de Comunicação de Dados Aeroviários da Embratel
ALGOL	Algorithmic Language
ARPANET	Advanced Research Projects Agency Network
BCPL	Basic Combined Programming Language
CPL	Combined Programming Language
DARPA	Defense Advanced Research Projects Agency
DMA	Direct Memory Access
DNS	Domain Name System
EaD	Ensino/Educação a Distância
EMBRATEL	Empresa Brasileira de Telecomunicações
ENIAC	Electronic Numerical Integrator and Computer
EPP	Enhanced Parallel Port
FINDATA	Serviço Internacional de Acesso a Informações Financeiras da Embratel
FORTRAN	Formula Translation
GAMM	Gesellschaft für Angewandte Mathematik und Mechanik - Sigla alemã de Sociedade para Matemática Aplicada e Mecânica
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
IAL	International Algorithmic Language
IDE	Integrated Development Environment
INTERDATA	Serviço Internacional de Comunicação de Dados da Embratel
IP	Internet Protocol
ISO/IEC	International Organization for Standardization/Information Centre
LDR	Light Dependent Resistor

Led	Light-emitting diode
LNCC	Laboratório Nacional de Computação Científica do CNPq
MEC	Ministério da Educação
MILNET	Military Networking
MIT	Massachusetts Institute of Technology
NCP	Network Control Protocol
NRZI	No Return to Zero Inverted
NTC	Negative Temperature Coefficient
OOP	Object-Oriented Programming
PWM	Pulse With Modulation
RF	Rádio-Frequência
SiedSup	Sistema Integrado de Informações da Educação Superior
SPP	Standard Parallel Port
SRI	Stanford Research Institute
UNIVAC	Universal Automatic Computer
USB	Universal Serial Bus
UTI	Unidade de Tratamento Intensivo
TCP	Transmission Control Protocol
Telebrás	Telecomunicações Brasileiras
TRANSDATA	Serviço Digital de Transmissão de Dados Via Terrestre da Embratel
TTL	Transistor-Transistor Logic
ZDP	Zona de Desenvolvimento Proximal

RESUMO

O uso de experimentos no processo ensino-aprendizagem já teve sua eficiência estudada e comprovada por diversos pesquisadores. A experimentação permite uma maior motivação para aprendizagem, permite que o estudante se torne um agente ativo no processo, e possibilita ao estudante um confronto entre o que ele constrói mentalmente e o que ele visualiza em uma situação real, o que pode levar que ocorra a aprendizagem. Para o ensino de determinadas áreas da Ciência, como Física e Química, a experimentação possui um papel muito importante.

Com a regulamentação da Educação a Distância no país, a quantidade de cursos que possuem em suas grades curriculares aulas experimentais, tem aumentado cada vez mais. Nestes cursos, as aulas experimentais vêm ocorrendo em aulas presenciais, onde o estudante deve se deslocar até um pólo do curso. Para que estas aulas possam também ser realizadas a distância, é necessário o desenvolvimento de novas ferramentas tecnológicas, que permitam o controle e a coleta de dados de experimentos, a distância.

Baseado nisto, foi desenvolvido um Sistema de Instrumentação Virtual, adaptável a diferentes procedimentos experimentais ou de controle de dispositivos, com o acesso a distância, através da Internet. O sistema consiste num conjunto interativo de softwares e hardwares. Foram desenvolvidos três tipos de softwares que se comunicam entre si por meio de protocolo TCP/IP. Um primeiro Software, denominado "Cliente", é direcionado para os usuários (estudantes), permitindo o acesso ao sistema. O segundo software, denominado "Servidor de Controle", se comunica diretamente com o experimento ou dispositivo de controle o qual se pretende acessar, controlando e coletando dados para envio aos softwares do tipo "Cliente". O último software desenvolvido é o "Servidor de Comunicação" que permite o gerenciamento dos usuários conectados e a comunicação entre estes. É utilizado ainda um software servidor encoder, que coleta imagens dos experimentos e dispositivos de controle, disponibilizando o acesso destes pelos usuários.

Foram desenvolvidos circuitos eletrônicos para permitir o interfaceamento entre os experimentos e dispositivos de controle e um computador. Foram criados circuitos para a comunicação através da porta paralela e através da porta USB.

Para que fosse possível a realização de testes de funcionamento, foram criados dois dispositivos de controle, que consistem em elementos que tem suas ações controladas a distância. Para testes não somente de controle, mas de coleta de dados, foram criados três experimentos.

Foram realizadas sessões de testes onde foi possível analisar não somente o funcionamento do sistema desenvolvido, mas também a aplicabilidade da ferramenta na Educação a Distância. Os resultados obtidos permitem concluir que o sistema pode ser uma eficiente ferramenta para este propósito, podendo ainda ser utilizada para o controle de equipamentos ou quaisquer outros elementos onde seja necessário o controle e a coleta de dados a distância.

Palavras-chave: Educação a Distância, Instrumentação Virtual, Ensino de Ciências

ABSTRACT

The use of experiments in the teaching-learning process already had studied efficiency and proven for several researchers. The experimentation allows a larger motivation for learning, it allows the student to become an active agent in the process, and make possible the student a confrontation among which he builds mentally and the one that he visualizes in a real situation, what can cause the learning. For the teaching certain areas of the Science, as Physics and Chemical, the experimentation possesses a very important paper.

With the regulation of the Distance Education in Brazil, the amount of courses that have experimental classes, has been increasing more and more. In these courses, the experimental classes are happening in presencial classes, where the student must move to a nucleus of the course. For these classes be accomplished by distance, it is necessary the development of new technological tools, that allow the control and the collection of experimental data, by distance.

Based on this, a System of Virtual Instrumentation was developed, adaptable to different experimental procedures or of control of devices, with the access by distance, through the Internet. The system consists of an interactive group of softwares and hardwares. Were developed three types of softwares that communicate amongst themselves through TCP/IP protocol. A first Software, denominated "Cliente", it is addressed for the users (students), allowing the access to the system. The second software, denominated "Servidor de Controle", communicate directly with the experiment or control device that it intends to access, controlling and collecting data for sending to the type "Cliente" softwares. A last developed software is the "Servidor de Comunicação" that allows the connected users' administration and the communication among these. It is still used a software server encoder, that collects images of the experiments and control devices, making available the access of these by the users.

Electronic circuits were developed to allow the interface between the experiments and control devices and a computer. Circuits were created for the communication through the Parallel Port and through the USB Port.

So that it was possible the accomplishment of operation tests, two control devices were created, that consist of elements that have their controlled actions by distance. For tests not only of control, but of collection of data, were developed three experiments.

Sessions of tests were accomplished where was possible to analyze not only the operation of the developed system, but also the applicability of the tool in the Distance Education. The obtained results allow to conclude that the system can be an efficient tool for this purpose, could still be used for the control of equipments or any other elements where it is necessary the control and the collection of data by distance.

keywords: Distance Education, Virtual Instrumentation, Teaching of Sciences

1 - INTRODUÇÃO

A capacidade de armazenamento, transmissão e processamento de informações vem se aperfeiçoando desde os tempos antigos, onde o armazenamento e a transmissão ainda eram feitos por meio de hieróglifos e pinturas em paredes de cavernas. (Bakos, 1996)

Algum tempo depois a informação passou a ser armazenada e transmitida em objetos móveis, tais como pedaços de rochas, tábuas de madeira ou pedra, etc. (Bakos, 1996, Pignatari, 2002)

Com o surgimento de alfabetos, a informação tornou-se mais compacta, podendo ser descrita por meio de letras, formando palavras, que simbolizam ações e objetos.

O processamento da informação sofreu um grande passo no seu desenvolvimento com o estabelecimento dos algarismos, tornando possível quantificar objetos, e processá-los por cálculos. As primeiras formas de quantificação se davam com o uso de partes do corpo, como dedos, e com objetos, como pedras. (Ponczek, 2002)

O ser humano tem como habilidade, o desenvolvimento de meios que facilitem algum trabalho. Desta forma, várias ferramentas foram desenvolvidas, como é o caso das ferramentas para facilitar a quantificação. É o caso do Ábaco, que já era utilizado pelos babilônicos em 3000 a.C. Com este era possível a realização de cálculos envolvendo unidades, dezenas, centenas, etc. (Boyer, 1996)

Ao longo dos anos novas ferramentas foram criadas para realização de cálculos e para a produção de um trabalho físico, facilitando a agricultura, a manufatura, entre outros.

Segundo Kuhn, uma revolução na ciência ocorre quando, após a descoberta de incompatibilidade do real com conceitos pré-estabelecidos, novos modelos conceituais surgem, isto é, estabelecem-se novos paradigmas. Como consequência de uma revolução na ciência, ocorre um período de intensa construção de novos equipamentos e ferramentas, baseadas nos novos alicerces científicos. (Kuhn, 2003)

Isto foi exatamente o que ocorreu com a revolução científica do século XVI, onde um conjunto de novos conceitos científicos estabeleceu-se, e baseada nestes,

a revolução industrial do século XVII foi possível, onde várias máquinas, equipamentos e ferramentas surgiram.

Processos automatizados foram comuns neste período, e no século XIX as indústrias e laboratórios de pesquisa, já contavam com instrumentos de acompanhamento e controle de processos, estabelecendo uma área da tecnologia conhecida atualmente como instrumentação.

Assim como se fazia nos séculos anteriores, o ser humano continuava a desenvolver ferramentas cada vez mais eficientes para a facilitação do trabalho e agora também para facilitar o acompanhamento de processos.

A instrumentação do final do século XIX se baseava em aparatos eletromagnéticos e mecânicos. O advento da computação no século XX, aliado ao surgimento da eletrônica, possibilitou um grande avanço na instrumentação. (Fialho, 2002)

A computação e outros dispositivos baseados na eletrônica permitem que, atualmente, a informação possa ser armazenada e processada em meios digitais, que possibilitam uma grande capacidade de armazenamento num pequeno espaço físico e uma grande velocidade de processamento.

O uso de computadores na instrumentação deu origem a um novo campo, a instrumentação virtual, onde a informação, referente a algum procedimento físico, pode ser processada rapidamente e através do uso de softwares é possível efetuar o controle de dispositivos e coletar dados sensorizados.

O monitoramento de processos industriais e laboratoriais dinamizou-se com esta tecnologia de instrumentação. Um monitoramento que antes exigia diversos maquinários, que demandava maior espaço, energia e tempo, pôde ser realizado com o uso de placas de aquisição de dados padronizadas e softwares específicos.

Com o surgimento da Internet, iniciou-se um processo de grande evolução na transmissão da informação. A Internet, que hoje se define como uma grande rede mundial de comunicação entre computadores, abre uma nova possibilidade para a instrumentação virtual, que é a possível comunicação com processos automatizados a longas distâncias.

Este formato de instrumentação virtual tem um vasto campo de aplicação, podendo se desenvolver para diversos setores como, além da experimentação laboratorial e do controle e sensoriamento em processos industriais, o acompanhamento de sinais vitais na medicina, no ensino a distância, e outros.

Na experimentação e nos processos laboratoriais, uma instrumentação virtual a distância já pode ser notada principalmente nas pesquisas e nas coletas de dados na meteorologia. São comuns estações meteorológicas em locais distantes, onde através de redes e/ou satélites, os dados coletados nas estações podem ser acessados a distância.

Na Física, em casos específicos, é possível também se utilizar uma instrumentação virtual a distância, como em processos experimentais longos que se dão naturalmente sem intervenção humana contínua.

Além disto, grupos de pesquisas diferentes podem compartilhar seus procedimentos experimentais por meio de um acesso a distância, com o uso da internet.

O emprego da instrumentação virtual a distância em indústrias possui fortes motivações, como exemplo, em ambientes onde não é possível uma presença contínua do homem, mas que necessita de acompanhamento contínuo ou periódico de processos. Neste caso, quando se tem consideráveis distâncias entre o ponto de acompanhamento e o ponto onde se coletam os dados, o uso da instrumentação virtual mediada pela internet, ou rede local, é uma interessante alternativa.

Outra interessante aplicação é no Ensino a Distância. Com a regulamentação do Ensino a Distância para diferentes níveis, incluindo o ensino superior, instituições de ensino superior no país criaram cursos em diversas áreas, inclusive cursos que possuem tradicionalmente em suas grades curriculares, disciplinas com práticas experimentais.

Para suprir esta necessidade de prática experimental, pode-se desenvolver um sistema que permita o acesso a distância aos procedimentos experimentais. A instrumentação virtual com acesso pela internet apresenta-se como uma interessante solução que permite este acesso a distância.

Mesmo em cursos presenciais, um sistema deste tipo pode ser utilizado, permitindo que estudantes possam acompanhar procedimentos experimentais que sua instituição não dispõe, mas que outra instituição desenvolva.

Os sistemas de acesso a distância através da Internet que atualmente existem são pacotes comerciais de preços elevados e específicos para algumas aplicações, e com o funcionamento vinculado a um hardware incluso no pacote.

O desenvolvimento de sistemas, onde softwares possam se adaptar a hardwares comerciais ou mesmo aqueles que venham a ser desenvolvidos, e se adaptar a diversas aplicações, é algo novo.

Na elaboração de um sistema deste tipo, de instrumentação virtual pela internet de múltiplas aplicações, deve se buscar uma distribuição facilitada, na qual os softwares sejam aplicativos executáveis, de fácil manipulação e de simples configuração, com o intuito de possibilitar uma adaptação facilitada, para a aplicação desejada.

A presente dissertação apresenta o desenvolvimento de um sistema de instrumentação virtual adaptável a múltiplas aplicações, de acesso a distância através da Internet, com ênfase na aplicação no ensino de Física.

1.1- Objetivos

Os objetivos pretendidos com esta pesquisa foram:

- Desenvolver um sistema de instrumentação virtual de acesso a distância, através da internet, que possa ser adaptável a várias aplicações, e que dê suporte a multiusuários (acesso simultâneo de vários usuários).
- Desenvolver elementos (experimentos didáticos e dispositivos de controle) para a realização de testes do funcionamento do sistema;
- Executar testes do sistema desenvolvido;
- Com base nos resultados, verificar a aplicabilidade do sistema desenvolvido.

1.2- Estrutura da Dissertação

A dissertação inclui além da introdução, outros sete capítulos.

No segundo capítulo é apresentada uma breve revisão bibliográfica na área de instrumentação virtual aplicada no ensino; e instrumentação virtual com acesso a distância, através da internet.

O capítulo terceiro apresenta a base teórica para o desenvolvimento da pesquisa. Inicialmente são abordados os conceitos de Redes de Computadores e Internet com um detalhamento histórico do desenvolvimento destes. É apresentado os aspectos relativos ao Ensino a Distância (EaD), com o desenvolvimento histórico, e sobre o uso da Internet como ferramenta em EaD.

Ainda no terceiro capítulo, são apresentadas as teorias pedagógicas de Piaget, Vygotsky, Ausubel e Papert, assim como uma análise das relações entre estas teorias e o uso de novas tecnologias na educação, com destaque no ensino experimental.

Em seguida são apresentados os aspectos envolvidos na programação de softwares, com a descrição histórica, e detalhes sobre os ambientes de desenvolvimento Delphi e C++ Builder, que foram utilizados nesta pesquisa.

Ao final deste terceiro capítulo são apresentadas as informações técnicas sobre as portas de comunicação Paralela e USB.

O capítulo quarto apresenta a descrição dos materiais e métodos utilizados na pesquisa, detalhando o processo de desenvolvimento do sistema de instrumentação virtual e dos elementos para testes, além do detalhamento da execução dos testes do sistema desenvolvido.

Os softwares criados, e os resultados e análise dos testes, são apresentados no capítulo quinto.

No capítulo sexto é apresentado um levantamento dos dados quantitativos relacionados a Educação a Distância no Brasil, e a aplicabilidade do sistema desenvolvido.

E por fim, no capítulo sétimo são apresentados os comentários finais sobre a pesquisa desenvolvida, e no capítulo oitavo as referências bibliográficas do trabalho.

2- REVISÃO BIBLIOGRÁFICA

São poucos os trabalhos envolvendo Instrumentação virtual de acesso a distância, principalmente no que se refere a aplicação desta tecnologia em EaD.

No entanto, são comuns projetos envolvendo o controle e coleta de dados de experimentos a partir de computadores, mas apenas com um acesso local. Nestes são utilizadas as mais variadas tecnologias de software e hardware.

Como exemplo, foi montado em 1988 na UFF, um sistema para medição de oscilações elétricas, para utilização em cursos de Física experimental em nível superior. Para este sistema foi desenvolvido um circuito de interface, baseado em um conversor A/D de 8 bits, com comunicação com um computador Apple II, através de um “slot” de conexão de placas. (Eichler, 1988)

Em 1990, no artigo “Verification of fundamental principles of mechanics in the computerized student laboratory” é apresentado um sistema de aquisição de dados em tempo real para experimentos envolvendo os conceitos de leis de Newton, conservação de momento e energia e força magnética. (Eckstein, 1990)

Em um trabalho intitulado “*Aquisição de dados via porta de jogos de um PC utilizando programação LOGO*” desenvolvido na UFRJ, no ano de 2000, apresenta uma alternativa para a coleta de dados, utilizando-se a porta de jogos como canal de comunicação com o computador. O software de interação com os dispositivos externos foi desenvolvido no ambiente LOGO, numa versão não executável (não convertido em um aplicativo executável), que interpretava os dados coletados através da porta de jogos. (Aguiar e Laudares, 2001)

Ao longo dos anos 90, diversas outras pesquisas semelhantes foram desenvolvidas, envolvendo a coleta e análise de dados experimentais em tempo real, com acesso somente local. (Gonçalves et al, 1991; Maps, 1993; Preston e Good, 1996; Sokoloff e Thornton, 1997; Ribas et al, 1998)

Na UENF foram desenvolvidos, durante os anos de 2002 a 2006, diversos experimentos de Física, com o controle e coleta de dados mediados por computador, na pesquisa intitulada “*Sistemas Virtuais Automatizados para o Ensino de Física*”; com o uso da porta paralela como canal de comunicação, e softwares desenvolvidos no ambiente de programação Borland Delphi. Foram criados experimentos com o uso de materiais de baixo custo, os circuitos de interface que permitiam o controle e coleta de dados, e softwares específicos para cada experimento ou conjuntos de

experimentos. Nesta pesquisa os experimentos também só poderiam ser acessados localmente. (Pessanha, 2006)

Estes trabalhos envolvem o desenvolvimento de tecnologias para o acesso a experimentos locais, mediado por computadores, que podem ser utilizados unicamente em aulas presenciais. No que se refere aos sistemas de acesso a experimentos remotos, isto é, acesso a distância aos experimentos, podem ser destacados alguns trabalhos.

Em 1998 foi criado o Laboratório de Experimentação Remota da UFSC, RExLab. Desde sua criação, em conjunto com outras instituições, várias pesquisas têm sido realizadas envolvendo o controle e monitoramento de dados a distância. Cabe citar duas pesquisas desenvolvidas em parceria com a Unisul: “RoboCAR” e “Monitoramento de pacientes em UTI utilizando Microservidores WEB”. A primeira pesquisa trata do desenvolvimento de um Robô autônomo equipado com um microservidor e ponto de acesso wireless. A segunda pesquisa consiste num aparato tecnológico que armazena informações sobre alguns sinais vitais de paciente de UTI, disponibilizando estes dados para acesso remoto. (RExLab, 2003)

Foi desenvolvido entre os anos de 1999 e 2002 na USP, um projeto envolvendo instrumentação virtual a distância, para uma aplicação específica, intitulada “*Instrumentação Virtual aplicada a um laboratório com acesso pela Internet*”. O sistema desenvolvido permitia o acesso de um usuário a um experimento de caracterização elétrica de resistores e diodos, em função da temperatura. O software era uma versão que funcionava a partir do ambiente de desenvolvimento, ou seja, não era gerado um aplicativo executável. O sistema foi desenvolvido no ambiente de instrumentação virtual LabView, e a aquisição de dados era realizada através de um hardware específico. (Borges, 2002)

Na UFRGS, já era desenvolvida em 2002, com apoio do Centro de Referência para o Ensino de Física (CREF), uma pesquisa intitulada “*Experimentação Remota para o Ensino de Física*”. Nesta pesquisa, experimentos de Física Moderna eram conectados a uma interface, que por sua vez, era conectada a um computador servidor, e um software interpretava os dados obtidos no experimento. Através de um site era possível o acesso a estes dados. (CREF, 2002)

Cabe ressaltar, que em todos estes trabalhos, a instrumentação virtual é específica para o que foi desenvolvido, não sendo adaptável a outras aplicações.

Estes sistemas, diferentemente do sistema desenvolvido e que é apresentado nesta dissertação, possuem o seu uso específico para um experimento ou conjunto experimental, impossibilitando assim, que sejam utilizados no acesso a distância a outros experimentos. A adaptabilidade a diversas aplicações, possibilitando o acesso a vários experimentos e a variados dispositivos de controle, é a principal diferença entre estes sistemas e o que foi desenvolvido na pesquisa apresentada nesta dissertação.

Além disto, muitos destes projetos de instrumentação virtual não consideram o acesso simultâneo de vários usuários a um mesmo procedimento de controle e coleta de dados e o acompanhamento do dispositivo que está sendo acessado através de imagens de vídeo em tempo real. Tais características foram consideradas no desenvolvimento da pesquisa apresentada nesta dissertação, sendo, então, características importantes que diferenciam o sistema desenvolvido, dos que são apresentados neste capítulo de revisão.

3-BASES TEÓRICAS

3.1-Redes de Computadores e Internet

Redes de Computadores são conjuntos de computadores e outros dispositivos (impressoras, Hubs, Switchs, Roteadores, etc.) interligados por um sistema de comunicação constituído de enlaces físicos (cabos e conexões entre estes) e regras que disciplinam esta comunicação, que permitem a troca de informações e compartilhamento de recursos.

Uma grande rede que se tem em evidência é a chamada Internet. Esta é composta pela conexão de várias redes de menor porte.

3.1.1- Histórico da Internet

Foi em meados dos anos 60, que surgiu um interessante meio de transmissão de informação: rede de computadores. No ano de 1965, os cientistas Lawrence Roberts e Thomas Merrill realizaram uma experiência de comunicação entre computadores, por meio de linha telefônica, permitindo uma conexão entre dois centros de pesquisa em Massachusetts e na Califórnia. (Internet Society, 2003)

Em decorrência da Guerra Fria, existia nos EUA um estímulo a pesquisas tecnológicas para a criação de um sistema descentralizado de armazenamento e transmissão de informações. A Agência de Projetos de Pesquisas Avançadas do Departamento de Defesa dos EUA (DARPA) iniciou em 1969 o projeto ARPANET (Advanced Research Projects Agency Network), que consistiu em uma rede de computadores que permitiu a comunicação entre quatro servidores, em quatro instituições de pesquisa: Universidade da Califórnia em Los Angeles (UCLA), Universidade da Califórnia em Santa Bárbara (UCSB), Universidade de Utah, e Instituto de Pesquisa de Stanford (SRI-Stanford Research Institute). Nos anos seguintes vários outros computadores foram integrados a Rede ARPANET. (Internet Society, 2003)

Em 1983 a ARPANET foi desmilitarizada com o surgimento de uma rede específica de aplicação militar, a MILNET. Outras redes já existiam nos anos 80 além da ARPANET e a nova MILNET. Ao final dos anos 80 várias das redes existentes se interligaram formando o que hoje conhecemos como Internet. (Internet Society, 2003)

3.1.2- Histórico da Internet no Brasil

Segundo Carvalho (2006), em 1976 a Empresa Brasileira de Telecomunicações (Embratel) instalou as primeiras linhas específicas para transmissão digital de dados, em caráter experimental, entre Rio de Janeiro e São Paulo. A partir destes testes a empresa estatal criou uma série de serviços de transmissão de dados destacando-se a rede TRANSDATA (Serviço Digital de Transmissão de Dados Via Terrestre) que inicialmente comunicava 30 cidades brasileiras, e era formada por circuitos privados e alugados a preços fixos.

A Embratel lançou em 1981 seu primeiro serviço de transferência de dados internacional, o AIRDATA (Serviço Internacional de Comunicação de Dados Aeroviários). Em 1982 foi disponibilizado o INTERDATA (Serviço Internacional de Comunicação de Dados) e no ano seguinte um serviço específico para transmissão de informações financeiras, o FINDATA (Serviço Internacional de Acesso a Informações Financeiras). (Carvalho, 2006)

Em 1982 foi criada pela Embratel a primeira comunidade interligada por rede de computadores no Brasil, o “Ciranda”, direcionado a funcionários da empresa. O projeto consistia em um servidor de informações capaz de atender 300 funcionários simultâneos, para os quais eram disponibilizados serviços de correio eletrônico, acesso a informações corporativas, agendas de aniversários, informações sobre benefícios aos funcionários, e outros. Foram colocados computadores nos escritórios da empresa e disponibilizados financiamentos aos funcionários para a compra de computadores próprios, para o uso doméstico. (Carvalho, 2006)

A transferência de informações em redes despertou também o interesse acadêmico. Um dos primeiros projetos envolvendo redes de computadores no meio acadêmico foi realizado em 1981 na UFRJ. Já a transferência de dados entre diferentes instituições acadêmicas ocorreu em 1982, com a REDPUC, em um trabalho conjunto entre PUC/RJ, Telebrás, Embratel e USP. Em 1988 ocorre a primeira comunicação em rede, entre o Brasil e uma instituição internacional, com caráter acadêmico, foi a comunicação entre o Laboratório Nacional de Computação Científica do CNPq (LNCC) e a Universidade de Maryland nos EUA. (Carvalho, 2006)

A primeira conexão com a Internet foi realizada pela Fapesp em Fevereiro de 1991, e em 1995 com o surgimento de serviços comerciais de acesso a Internet,

iniciou-se no Brasil um intenso processo de popularização do uso da Internet. (Carvalho, 2006)

3.1.3 – Protocolos TCP/IP

Para que computadores em uma rede possam trocar informações é necessário que todos adotem um mesmo conjunto de regras de envio e recebimento de informações. Este conjunto de regras é o que se denomina como protocolos.

Até o final dos anos 70 a ARPANET fazia uso do protocolo NCP (Network Control Protocol), que foi criado em dezembro de 1971. O protocolo NCP não possuía controle de erros; qualquer erro que ocorresse, como a perda de pacotes de dados durante a transferência, ocasionaria a paralisação da transferência dos dados. Ao final dos anos 70 surge um conjunto de protocolos, o TCP/IP (Transmission Control Protocol/Internet Protocol), que apresentava vantagens em relação ao protocolo NCP, como o tratamento de erros e múltiplos identificadores disponíveis para serem atribuídos aos computadores que se conectassem a uma rede. Em Janeiro de 1983 ocorreu então a transferência do uso do protocolo NCP para protocolos TCP/IP. (Internet Society, 2003)

O IP (Internet Protocol) é um protocolo que define uma identificação para cada computador em uma rede. Esta identificação é feita através de um código formado por um conjunto de 4 bytes, que se denomina endereço IP.

Já o TCP (Transmission Control Protocol), é um protocolo de transporte que fornece confiabilidade aos dados transmitidos, algo que é necessário em qualquer transmissão de dados entre computadores. Este protocolo estabelece um conjunto de regras que permitem que os computadores em uma rede possam interpretar os dados compartilhados.

Em resumo, na comunicação entre computadores de uma rede, cada um possui uma identificação IP, e esta comunicação ocorre obedecendo a um conjunto de regras estabelecidas pelo protocolo TCP. Quando um software se comunica com um servidor, é necessário então, que seja fornecido ao software o IP do servidor.

Atualmente, grande parte dos servidores na internet possui uma identificação denominada DNS (Domain Name System), com a qual, não é necessário que se saiba o IP, e sim apenas um nome de domínio, de melhor memorização.

Um exemplo do uso do DNS é no acesso a Websites através de softwares navegadores de Internet. Para que um Website seja acessado, é fornecido um endereço nominal (ex.: www.nomedosite.com) que geralmente possui alguma relação com o tema do Website. Este endereço nominal é o DNS que corresponde a uma identificação IP.

Outra característica da comunicação em redes é a utilização de portas virtuais de comunicação. Cada software utiliza portas virtuais de comunicação específicas, identificadas por um número inteiro, pela qual o software cliente se comunica com um software servidor.

Softwares do tipo cliente ou servidor possuem estas definições com base em um paradigma, onde um aplicativo servidor é aquele que espera passivamente por uma conexão ou solicitação de outros softwares (clientes), enquanto um aplicativo cliente inicia a comunicação ativamente. (Comer, 2001)

3.2- Educação a Distância

A Educação a Distância (EaD) consiste em um processo de ensino-aprendizagem, mediado por tecnologias, onde professores e estudantes estão separados espacial e/ou temporalmente. As tecnologias utilizadas neste processo são diversas, como o correio, o rádio e televisão, vídeos, o CD-ROM, telefone e fax, e destacando-se, a Internet. (Moran, 2002)

3.2.1- Histórico da Educação a Distância

Segundo Landim (1997), a Educação a Distância teria surgido com o aparecimento da escrita. Já no século primeiro, líderes do cristianismo teriam distribuído mensagens escritas com seus ensinamentos a discípulos fisicamente ausentes. Estes exemplares teriam sido reescritos e redistribuídos pela Europa, norte do continente Africano e Oriente Próximo. (Landim, 1997, Andrade, 2001, p.8-22)

Estes livros, apesar de alcançarem grande distribuição, eram inacessíveis à plebe, por possuírem um custo elevado. Tais livros eram acessíveis principalmente aos mestres, que os utilizavam no ensino de seus conteúdos, efetuando a leitura destes.

Com a invenção da imprensa por Johannes Guttenberg, livros impressos surgiram, tornando desnecessário que até mesmo alunos presenciais fossem às

escolas para assistir as leituras do mestre, de um raro livro copiado. Há relatos de resistências ao uso destes livros escolares impressos mecanicamente, por parte de escolas contemporâneas a Guttenberg, por mestres acreditarem que estes novos livros poderiam fazer com que se tornasse desnecessária a figura do mestre. (Alves, 1994; Andrade, 2001, p.8-22)

No século XVIII aparecem os primeiros cursos de Educação a Distância em um escopo moderno. Em 1728, foi publicado um anúncio no Jornal Gazeta de Boston de um material de ensino e tutoria por correspondência. (Andrade, 2001, p.8-22)

Na década de 50 do século XIX, tem-se como destaque duas daquelas que são consideradas as primeiras iniciativas institucionais de EaD no mundo. Uma primeira iniciativa ocorreu na Rússia em 1850, com o Instituto para o Ensino de Correspondência. Outra iniciativa ocorreu em 1856 na Alemanha, pela Sociedade de Línguas Modernas, na qual os professores Gustave Langeschbeidt e Charles Toussaint ofereciam o ensino de inglês por correspondência. (Andrade, 2001, p.8-22)

Nas décadas seguintes surgiu uma série de outras iniciativas envolvendo o ensino a distância por meio de materiais impressos. O uso de outras tecnologias no ensino a distância ocorre somente no século XX, em 1947, quando a Faculdade de Letras e Ciências Humanas de Paris começou a transmitir, através de ondas de rádio, várias das matérias literárias. (Andrade, 2001, p.8-22)

No Brasil não há registros precisos sobre as origens da EaD. Segundo Alves (1998) o Jornal do Brasil, que iniciou suas atividades em 1891, já teria publicado, em sua primeira edição da seção de classificados, um anúncio de curso profissionalizante em datilografia por correspondência. Alves (1998) aponta ainda a implantação das “Escolas Internacionais” em 1904 como um marco inicial da EaD no Brasil.

Outros autores, consideram a criação em 1939 do Instituto Rádio Monitor como a origem da EaD no Brasil. Tem-se ainda como destaque o surgimento em 1941 do Instituto Universal Brasileiro, com a oferta de cursos profissionalizantes a distância, por correspondência. (Volpato *et al*)

Na década de 50 do século XX chega ao Brasil a TV aberta. Desta forma, além do uso de materiais impressos e do rádio, a televisão também passaria a ser utilizada na Educação a Distância. Neste âmbito destaca-se o surgimento, em 1978,

do “Telecurso 2º Grau”, e depois o “Telecurso 1º Grau”, com o objetivo de levar a educação a pessoas que por algum motivo abandonaram os estudos. Atualmente esta iniciativa se mantém, com o “Telecurso 2000”. (Rocha, 2000)

No Brasil, foi autorizado com a lei nº. 9.394, de 20 de dezembro de 1996 e regulamentado com o decreto nº. 5.622 (publicado no Diário Oficial da União de 20 de dezembro de 2005), o Ensino a distância para diferentes níveis. (Brasil, 1996; Brasil, 2005)

Desta forma, além da Educação a Distância em nível técnico que já vinha ocorrendo, instituições de ensino puderam oferecer também cursos em nível básico, de educação especial, de educação de jovens e adultos, e de nível superior, observando-se a regulamentação vigente.

3.2.2- Internet e Educação a Distância

Com o advento e popularização da computação, esta nova tecnologia passou a ser considerada na Educação a Distância. Vários são os meios computacionais que podem ser utilizados para transmissão/construção do conhecimento em EaD, como as mídias removíveis (disquete, cd-roms, dvd-rom, etc.) contendo apostilas e exercícios digitalizados, imagens, vídeos-aula e softwares diversos, ou mais recentemente a Internet.

A Internet passou a um lugar de destaque como importante ferramenta para a EaD, onde de forma facilitada é possível a transmissão de texto, vídeo, áudio, imagens, além de transmissões ao vivo de acontecimentos.

Este recente uso da Internet em EaD intensificou-se na segunda metade dos anos 90, do século XX. Porém pode-se destacar o uso da Internet por parte da Universidade de Brasília (UnB) em cursos de extensão, desde os anos 80. Cabe citar ainda a criação do Laboratório de Educação a Distância (LED) na Universidade Federal de Santa Catarina (UFSC) em 1995, vinculado ao Programa de Pós-Graduação em Engenharia de Produção, que tem direcionado várias pesquisas sobre EaD, em especial com a mediação da Internet. (Andrade, 2001)

3.3- Teorias Pedagógicas e Tecnologia da Informação

O uso de novas tecnologias na prática educacional é alvo de muita discussão, e atualmente encontra grande apoio de pesquisadores da área pedagógica.

Existem teorias pedagógicas que apóiam o desenvolvimento de novas formas de despertar o interesse, desenvolver a mente e o raciocínio, e estimular a criatividade. As novas tecnologias se apresentam como interessantes ferramentas para este propósito. (Boechat, 2006)

Entre estas teorias destacam-se as dos pesquisadores Jean Piaget, Lev Vygotsky, David Ausubel e Seymour Papert. Tais teorias foram consideradas no desenvolvimento da pesquisa aqui apresentada, e um resumo de cada uma destas e os pontos que se relacionam com o uso de novas tecnologias, em especial ao uso da tecnologia da informação (computação), são apresentados a seguir.

3.3.1- Jean Piaget

Piaget nasceu em 9 de agosto de 1896 na cidade Suíça de Newchatel, e faleceu em Genebra no dia 16 de setembro de 1980. Estudou inicialmente biologia e depois se dedicou ao estudo de Psicologia, Epistemologia e Educação, sendo professor de psicologia na Universidade de Genebra de 1929 a 1954.

A teoria de Piaget se baseia na defesa do conceito de que o indivíduo ao longo da sua vida percorre etapas de desenvolvimento. Estas etapas são apresentadas na Tabela 1, a seguir:

Tabela 1 – Períodos do desenvolvimento mental da Teoria de Piaget

Etapa	Idade	Características
Sensório-motor	0 à 2 anos	Desenvolvimento da consciência do próprio corpo, diferenciando do restante do mundo físico. Desenvolvimento da inteligência em três estágios: reflexos de fundo hereditário, organização das percepções e hábitos e inteligência prática.
Pré-operacional	2 à 7 anos	Desenvolvimento da linguagem, com três conseqüências para a vida mental: a) socialização da ação, com trocas entre os indivíduos; b) desenvolvimento do pensamento, a partir do pensamento verbal: finalismo (porquês), animismo e artificialismo; c) desenvolvimento da intuição.
Operatório-Concreta	7 à 11 ou 12 anos	Desenvolvimento do pensamento lógico sobre coisas concretas; compreensão das relações entre coisas e capacidade para classificar objetos; superação do egocentrismo da linguagem; aparecimento das noções de conservação de substância, peso e volume.
Operatório-Formal	A partir dos 12 anos	Desenvolvimento da capacidade para construir sistemas e teorias abstratos, para formar e entender conceitos abstratos, como os conceitos de amor, justiça, democracia, etc.; do pensamento concreto, sobre coisas, passa para o pensamento abstrato, "hipotético-dedutivo", isto é, o indivíduo se torna capaz de chegar a conclusões a partir de hipóteses: se A é maior que B e B é maior que C, A é maior que C.

Adaptado de Piletti (1999) e Boechat (2006)

Segundo a teoria construtivista de Piaget, é a partir de experiências de acomodação e assimilação que o indivíduo constrói significados, onde relacionando experiências novas com anteriores ocorre o desequilíbrio. Para entender o evento causador do desequilíbrio, o indivíduo deverá reajustar seu esquema mental ou criar um novo esquema.

O reajuste de um esquema mental ou o novo esquema passará então por processos de experimentação, onde a realidade elaborada mentalmente pelo indivíduo irá se confrontar conceitualmente com os fatos concretos. A partir deste conflito conceitual ocorre então a aquisição/construção do conhecimento. (Boechat, 2006)

O uso da computação no processo ensino-aprendizagem pode ocorrer através de vários recursos, como softwares educacionais, textos, vídeos e simulações, experimentos computadorizados, entre outros.

Os softwares educacionais permitem que os estudantes não necessariamente percorram seqüências pré-estabelecidas, onde se assume uma postura ativa na construção do seu próprio conhecimento, podendo desenvolver livremente suas potencialidades. (Netto, 2005; Boechat,2006).

O uso de vídeos e simulações pode proporcionar aos estudantes o conflito conceitual com fatos concretos (ou simulações destes). O mesmo ocorre com experimentos computadorizados, que de forma bem específica podem apresentar situações reais no momento em que ocorrem, permitindo uma aproximação maior entre o indivíduo e os fatos concretos.

No desenvolvimento de tais recursos, no entanto, com base na teoria de Piaget, deve-se levar em conta a idade dos estudantes, sendo projetados os recursos de acordo com sua etapa cognitiva, e levando-se em conta ainda os possíveis desvios, isto é, indivíduos que podem estar adiantados ou atrasados nestas etapas.

3.3.2- Lev Vygotsky

Vygotsky nasceu na cidade de Orsha na Bielo-Rússia em 17 de novembro de 1896 e faleceu prematuramente aos 37 anos em 11 de junho de 1934 em Moscou. Formou-se em Direito pela Universidade de Moscovo em 1918, e a partir de 1924 dedicou-se ao estudo no campo da Psicologia e Educação.

De acordo com Vygotsky, existem dois níveis de desenvolvimento. Um nível é o desenvolvimento real, que é a capacidade intelectual já consolidada pelo indivíduo, possibilitando a resolução de problemas sem o auxílio de outra pessoa, e outro nível é o desenvolvimento potencial, que é a capacidade de resolver um problema somente com o auxílio de um outro indivíduo. Segundo Vygotsky a aprendizagem é uma consequência do desenvolvimento real e potencial, os quais não ocorreriam sem a aprendizagem. (Vygotsky, 1998)

A distância entre o nível de desenvolvimento real e o nível de desenvolvimento potencial é denominada “Zona de Desenvolvimento Proximal” (ZDP). Esta distância refere-se ao deslocamento percorrido pelo indivíduo em que desenvolve funções que irão amadurecer até serem estabelecidas no seu nível de desenvolvimento real. (Oliveira, 1995, p.60)

Vygotsky valoriza ainda o meio como forte influência no desenvolvimento humano, ressaltando os aspectos sócio-culturais como influenciadores no processo de aprendizagem. (Vygotsky, 1998)

Em resumo, é uma teoria que ressalta a capacidade individual, isto é o desenvolvimento real, ao mesmo tempo em que valoriza o meio como algo que influencia no processo, inclusive as interações sociais. Logo, a aprendizagem segundo a teoria, não apenas ocorre de forma ativa por parte do indivíduo, mas também interativa.

Na utilização dos recursos computacionais, deve ser levado em conta o desenvolvimento real do indivíduo, de tal forma que os softwares, vídeos, experimentos computadorizados, e outros, ofereçam os dispositivos que facilitem tal desenvolvimento, ampliando as possibilidades de conhecimento do indivíduo.

O conceito de desenvolvimento potencial aponta para a participação do professor durante a utilização dos recursos computacionais, além da interação entre os próprios estudantes como forte aspecto interpessoal.

Os recursos devem ainda ser contextualizados, isto é, não devem estar isolados do meio e sim envolver os aspectos do cotidiano e do ambiente do indivíduo.

3.3.3- David Ausubel

Ausubel nasceu em 1918 na cidade de Nova York nos EUA. Formou-se em psicologia em 1939 pela universidade da Pennsylvania. Pesquisou sobre o processo de aprendizagem desenvolvendo sua própria teoria da aprendizagem.

A teoria de Ausubel define dois tipos de aprendizagem, aquela que ocorre de forma mecânica, e outra, que foi denominada como “aprendizagem significativa”.

A Aprendizagem Significativa é o conceito central de sua teoria, que se define como aquela aprendizagem em que o indivíduo relaciona a nova informação com algum aspecto de relevância na sua estrutura de conhecimento, isto é, a nova informação se assemelha ou completa conceitos, ou idéias pré-existentes (subsunçor) na estrutura cognitiva (estrutura de armazenamento organizado de informação) do indivíduo. (Ausubel *et al.*, 1980)

Para que este tipo de aprendizagem ocorra é necessário que o novo conhecimento seja estruturado de maneira lógica, que existam subsunçores na estrutura cognitiva do aprendiz, e que este esteja motivado e aberto ao novo conhecimento.

A Aprendizagem Mecânica é um processo em que há pouca ou nenhuma associação entre os novos conceitos e os conceitos já existentes na estrutura cognitiva; isto é, a nova informação é armazenada de forma arbitrária na estrutura cognitiva, sem ligar-se a conceitos subsunçores específicos. Por este motivo, é possível o esquecimento, após pouco tempo, daquilo que foi aprendido mecanicamente. (Ausubel *et al.*, 1980)

Ausubel define como ideal a aprendizagem significativa, a menos que não existam na estrutura cognitiva dos alunos algum subsunçor em que se possa ligar o novo conceito. No entanto, Ausubel sugere como mais adequado para esta situação a utilização do que ele definiu como “Organizadores Prévios” que é uma abordagem inicial do novo conceito de forma mais geral e inclusiva, encaminhando-se para as noções mais específicas, para que assim o indivíduo possa desenvolver os subsunçores necessários. (Ausubel *et al.*, 1980)

Na utilização dos recursos computacionais, estes devem ser apresentados de forma lógica e com conceitos bem estruturados. É interessante uma análise prévia à utilização de tais recursos, a fim de se identificar aquilo que os estudantes já tem conhecimento, e só então, baseado nisto, utilizar tais recursos. Caso não seja

possível tal identificação, é necessário que os recursos sejam utilizados em uma abordagem mais geral, e com o tempo, progredir para abordagens mais específicas.

O uso da computação destaca-se um ponto muito importante, baseado na teoria de Ausubel, que é a motivação para a aprendizagem. É notório o estímulo para a aprendizagem, ocasionado pelo uso das novas tecnologias, em especial o uso de computadores.

3.3.4- Seymour Papert

Papert nasceu em 1º de março de 1928 em Pretória, na África do Sul. Formou-se em Matemática e participou de pesquisas neste campo, na Universidade de Cambridge entre os anos de 1954 e 1958. Depois, entre os anos de 1958 e 1963, desenvolveu pesquisas em conjunto com Jean Piaget na Universidade de Geneva. Ainda no início dos anos 60 ingressou como pesquisador no Instituto de Tecnologia de Massachusetts (MIT - Massachusetts Institute of Technology), participando da fundação do o Laboratório de Inteligência Artificial do MIT. (Takahashi e Hattori, 2000).

Cunhou o termo “Construcionismo”, como uma abordagem construtivista em que é permitido aos estudantes construir seu próprio conhecimento, com intermédio de alguma ferramenta, como o computador. Nesta concepção o indivíduo tem a possibilidade de construir o conhecimento a partir da interação com o computador, a partir da resolução de problemas.

Criou entre os anos de 1967 e 1968 a linguagem de programação “Logo”, específica para utilização no ensino, que objetivava permitir ao indivíduo a familiarização com conceitos lógicos e matemáticos a partir da exploração de atividades espaciais que auxiliam na formalização dos raciocínios cognitivos do indivíduo. (Valente e Almeida, 1997).

Apoiando-se na teoria construcionista de Papert, os recursos computacionais devem dar suporte as necessidades, fornecendo as bases para que o estudante possa construir o seu conhecimento.

Os softwares devem assim, ser interativos permitindo ao estudante interpretar, tomar decisões, e verificar os resultados destas decisões.

A interatividade deve também estar presente na experimentação computadorizada, onde o estudante possa controlar o procedimento experimental, receber os dados experimentais, interpretar, etc.

3.4- Ensino Experimental

O ensino experimental consiste na utilização de experimentos didáticos durante um processo de ensino-aprendizagem.

A experimentação é importante, e de certa forma fundamental no ensino das ciências naturais. É uma oportunidade de visualização da ação dos conceitos abordados em aula, além de permitir ao estudante que conheça algumas das ferramentas utilizadas na prática científica.

Baseado na Teoria de Piaget, a abordagem experimental se mostra como ótima ferramenta auxiliar no processo de construção do conhecimento. A apresentação de novos conceitos aos estudantes gera o que em sua teoria denomina-se desequilíbrio, e após isto, com a realização de práticas experimentais, é oferecido ao estudante uma oportunidade de confronto entre o que foi construído mentalmente e o que é real, ocorrendo então um reajuste mental ou construção de um novo esquema mental.

Outro aspecto relevante é a apropriação do experimento para a etapa cognitiva do indivíduo. O experimento ou a forma de proceder a experimentação, deve ser apropriada para a etapa cognitiva do indivíduo.

A presente pesquisa desenvolveu um sistema para ser utilizado por indivíduos que já possuam a capacidade de abstração e dedução, características estas da última etapa de desenvolvimento cognitivo segundo Piaget, a etapa operatório-formal.

A prática experimental deve ser administrada por um professor ou tutor. Baseado na teoria de Vygotsky, o experimento deve ser preparado com o intuito de permitir o desenvolvimento real do indivíduo, no entanto, é importante a presença do professor durante a prática experimental, de forma a possibilitar o desenvolvimento potencial. Ainda com base nas idéias de Vygotsky, a participação de grupos de estudantes no procedimento experimental apresenta-se como um ponto positivo que auxilia no desenvolvimento cognitivo destes estudantes.

Experimentos que possuam relações com situações cotidianas, ou propriedades de elementos já conhecidas pelos estudantes, são ótimas ferramentas de contextualização. Em outras palavras, a experimentação pode ser utilizada como ponte entre os conceitos ensinados e o que é real e presente no cotidiano dos estudantes.

A motivação explicitada por Ausubel como colaboradora na construção do conhecimento, é mais uma vantagem presente na prática experimental. A interação entre os estudantes e o experimento é algo que traz motivação para os estudantes, que têm a oportunidade de não serem apenas ouvintes de explicações, como ocorre na pedagogia tradicional, mas de serem indivíduos participativos no processo ensino-aprendizagem.

Em resumo, de acordo com a abordagem construtivista, experimentação deve, entre outras formas, ser realizada de forma dedutiva-racionalista ou hipotética-dedutivas, em que o ponto de partida consiste em uma construção teórica que precede as observações. Assim, cabe ao “professor desafiar os alunos a planejar e buscar soluções, participativamente, consciente da diversidade das concepções envolvidas e da sua impregnação pelo senso comum”. (Borges, 2007)

3.5- Desenvolvimento de Softwares – Linguagens de Programação

Programação é a definição de ações futuras (instruções) que ocorrerão em função do tempo ou serão condicionais a outras ações ou a propriedades destas outras ações. Uma seqüência de forma não ambígua de instruções, obedecendo a regras bem definidas, é chamada de Algoritmo.

No desenvolvimento de softwares se utiliza uma programação, que definirá as ações do software em função de outras ações (por exemplo, pressionar uma tecla), ou a partir de dados fornecidos pelo usuário do software, ou ainda, em função do tempo (por exemplo, uma ação que ocorrerá sempre em um determinado horário).

O ato de programar exige uma linguagem. Esta linguagem é composta por um conjunto de termos, caracteres, etc., que obedecem a regras sintáticas e semânticas. O programador, com o uso de uma linguagem, especifica de forma precisa o funcionamento do software.

O conjunto de termos, obedecendo as regras da linguagem de programação é denominado código fonte. Ao criar um software, este código fonte é convertido para linguagem de máquina, para que assim, o hardware possa interpretar o software.

As linguagens de programação podem ser classificadas pelo nível de compreensão dos termos utilizados na programação. Linguagens que se aproximam mais da linguagem binária utilizada em computadores são denominadas linguagens

de baixo nível, enquanto as linguagens que possuem termos que se aproximam da linguagem humana são consideradas como de alto nível. Não é uma classificação que define linguagens melhores ou piores, mas apenas o seu grau de aproximação a compreensão humana.

Na programação de Hardwares é comum o uso de linguagem binária (ou linguagem de máquina), compostas apenas por números formados pelos algarismos “0” e “1”, são as linguagens de mais baixo nível na computação. Uma outra linguagem de baixo nível é denominada Linguagem Assembly, que se aproxima da linguagem de máquina, mas utiliza não somente termos numéricos, mas também letras, ou seja, aceita termos alfanuméricos.

Linguagens mais recentes permitem uma maior rapidez e praticidade no desenvolvimento de um software. Estas linguagens são geralmente de alto nível, e utilizam termos que remetem diretamente a ação, como exemplo “begin” e “end” para, respectivamente, iniciar e finalizar um procedimento em um software. A Tabela 2, a seguir, demonstra partes de códigos fontes de softwares, de diferentes linguagens, de diferentes níveis.

Tabela 2 – Exemplos de partes de códigos fonte de linguagens em diferentes níveis

Linguagem	Parte de Código Fonte
Linguagem de Máquina	0110 1011 1111 0011
Assembly	add bx, 4 cmp [adr], 3 jmp address
C	printf("Digite a temperatura em Celsius "); scanf("%f",&t1); t2 = converte(t1); printf("A temperatura em Fahrenheit vale: %f\n", t2);
Delphi (Object Pascal)	a:=strtoint(edit1.Text); if (a mod 2) = 1 then showmessage('Número Ímpar') else showmessage('Número Par');
SQL	SELECT NOME,IDADE FROM GRUPO1 WHERE INICIAL = "J" AND IDADE IN ("21","22","23","24","25")

No desenvolvimento de softwares, é necessária além da linguagem de programação, uma série de ferramentas (softwares), destacando-se como as principais, o Editor, o Depurador, o Compilador e o Linker.

O Editor, como se pode deduzir do próprio nome, é onde se edita o código fonte do software, semelhante a um editor de textos, onde o programa é escrito com o uso de uma linguagem de programação. O Depurador é uma ferramenta que busca por falhas no código-fonte (erros de sintaxe e semântica), e auxilia na

correção destas falhas. O Compilador é um conversor do código-fonte com uma linguagem de programação, na linguagem de máquina. E por fim, o Linker é a ferramenta responsável pela criação de um aplicativo executável, a partir da linguagem de máquina já convertida.

Existem ambientes que integram todas estas principais ferramentas, tais ambientes são denominados IDE (Integrated Development Environment).

3.5.1- Histórico das Linguagens de Programação

Segundo Sebesta (2000), a primeira linguagem de programação, foi criada entre os anos de 1943 e 1945 pelo cientista alemão Konrad Zuse, que, no entanto, não chegou a ser implementada. Sua linguagem, denominada Plankalkül (Cálculo de Programa), foi registrada em um manuscrito datado de 1945, mas com publicação somente em 1972. O manuscrito possuía algoritmos na linguagem criada para a resolução de vários problemas, inclusive 49 páginas de algoritmos para jogar xadrez.

A primeira linguagem de programação de alto nível em comparação a linguagem de máquina, e que chegou a ser implementada, foi a Linguagem Short Code, desenvolvida por John Mauchly em 1949. Mauchly era um físico norte-americano, que participou da construção do primeiro computador eletrônico, o ENIAC, que foi concluído três anos antes. (Sebesta, 2000)

No ano de 1951, a equipe criadora do computador UNIVAC, primeiro computador comercial nos EUA, desenvolveu, chefiada por Grace Hopper, o primeiro sistema compilador denominado A-0. Nos dois anos seguintes foram desenvolvidos outros dois compiladores, o A-1 e A-2. (Sebesta, 2000)

Em 1957 surge uma linguagem de programação, que até os dias atuais é utilizada, em versões mais recentes, o FORTRAN (Formula Translation), que influenciou várias outras linguagens subseqüentes. Foi criada por uma equipe da IBM chefiada por John W. Backus. Esta linguagem foi muito utilizada no ensino de programação dos anos seguintes a sua criação até meados dos anos 70. (Sebesta, 2000)

Entre 27 de maio e 1º de junho de 1958, ocorreu em Zurique um encontro entre membros de duas entidades, a GAMM(Sigla alemã de Sociedade para Matemática Aplicada e Mecânica) e ACM (Association for Computing Machinery). Durante este encontro foi projetada a linguagem de programação IAL (International

Algorithmic Language) que depois teve seu nome modificado para ALGOL (Algorithmic Language). Esta primeira versão ficou conhecida como ALGOL 58, e dois anos após surgiu uma segunda versão, a ALGOL 60, que serviu como base para mais duas outras versões, o ALGOL W em 1966 e o ALGOL 68. (Sebesta, 2000)

O ALGOL W foi a referência para outra linguagem de grande destaque, o Pascal. Com o nome em homenagem ao matemático Blaise Pascal, a linguagem Pascal foi criada pelo suíço Niklaus Wirth, em 1970, como uma linguagem para ser utilizada no ensino de programação. Tal iniciativa obteve sucesso, visto que em 1975 já era comum o uso da linguagem no ensino de programação, ultrapassando a até então absoluta, neste propósito, linguagem FORTRAN. (Sebesta, 2000)

Em paralelo a evolução do ALGOL, surge em 1962, outra linguagem de programação de destaque, o CPL (Combined Programming Language), desenvolvida na universidade de Cambridge. A partir desta surge em 1967 o BCPL (Basic Combined Programming Language) criada por Martin Richards. (Sebesta, 2000)

Dando seqüência a uma evolução de linguagens com origem no CPL, foi criada em 1970 por Ken Thompson no Bell Laboratories, baseada no BCPL, a linguagem de programação B. Depois, com base na linguagem B e algumas influências do ALGOL 68, surge outra linguagem, o C, desenvolvida também no Bell Laboratories, em 1972 por Dennis Titchie. (Sebesta, 2000)

Juntamente com a linguagem Pascal, a linguagem C obteve grande aceitação, sendo estas duas muito utilizadas até os dias atuais. Ambas possuem um nível de compreensão alto, por serem linguagens que utilizam termos próximos à linguagem humana.

Com a incorporação de alguns conceitos envolvidos no modo de programação denominado “Programação Orientada a Objeto” (que será discutido a frente), surgiram duas novas linguagens: “Object Pascal” e “C++”.

A Linguagem C++ surgiu em 1983 no Bell Laboratories, criada por Bjarne Stroustrup. Já o Object Pascal surgiu somente em 1995, em conjunto com a criação do ambiente integrado de desenvolvimento (IDE) Delphi, da empresa Borland. A Borland desenvolveu ainda um ambiente para programação em C++, o C++ Builder. (Sebesta, 2000)

A Figura 1 apresenta um gráfico com a evolução histórica das linguagens de programação de alto nível aqui abordadas.

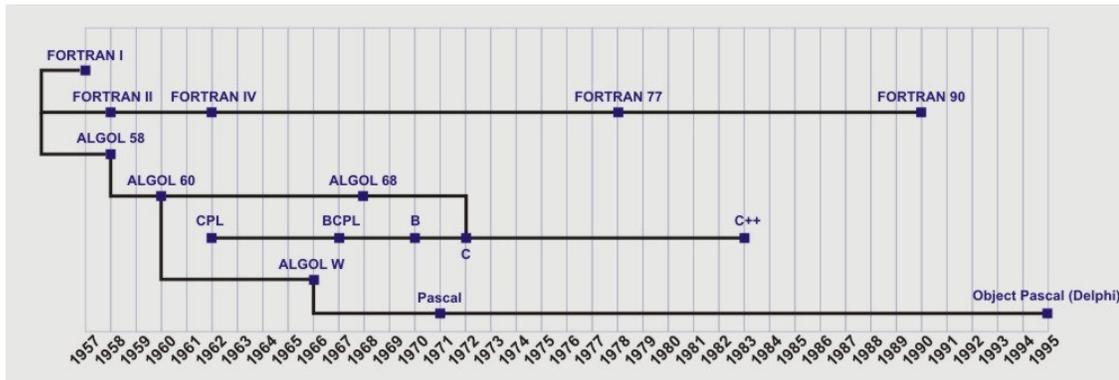


Figura 1 – Evolução histórica das linguagens de programação de alto nível.

3.5.2- Linguagens de Programação Orientadas a Objetos

Entre as linguagens de programação existentes, as que oferecem grande facilidade e agilidade no desenvolvimento e modificação de softwares são as chamadas Linguagens de Programação Orientadas a Objetos (OOP - Object-Oriented Programming).

A principal característica que difere as linguagens OOP das demais linguagens é a forma de se programar se referenciando aos chamados objetos, que são entidades que possuem propriedades e eventos associados.

Estes objetos podem ser variados, cada um com suas propriedades e eventos específicos. Como exemplo de objetos tem-se botões, caixas de texto, textos comuns, figuras, menus, entre outros, que são componentes visuais em um software. Existe ainda uma série de componentes (objetos) não visuais nestas linguagens, como registradores de tempo, comunicadores de rede, controladores de sons, controladores de janelas de diálogo, etc.

As propriedades dos objetos se referem às informações como dimensões, textos inclusos nos componentes, cor, posicionamento na tela, etc; em componentes visuais. Quanto aos componentes não visuais tem-se como exemplo de propriedade, o intervalo de tempo entre as ações de um componente registrador de tempo.

Os eventos dos componentes estão associados às ações. Um evento é uma ação primária fixa de um componente que provocará uma outra ação, ou conjunto de ações, secundária, que será definida pelo programador.

Como exemplo de eventos em componentes visuais, no caso de um botão que tem como finalidade, encerrar o software, tem-se o evento primário que é o “Clicar” do botão, que está associado a um evento secundário, que é definido pelo programador, que neste exemplo, trata-se de uma rotina que finaliza o software.

Os ambientes de desenvolvimento (IDE) mais utilizados, que usam linguagens OPP, oferecem grande facilidade na montagem visual dos softwares. Muitos destes utilizam o chamado GUI (Graphical User Interface), que é uma interface gráfica com o usuário do ambiente. Nestas para a montagem visual de um software, basta que o programador selecione os objetos visuais que deseja e os adicione posicionando em um plano, que será o fundo do software. Este plano, geralmente chamado de “formulário”, também é um componente. A Figura 2 apresenta uma tela capturada de uma destes IDE, no momento da montagem visual de um software.

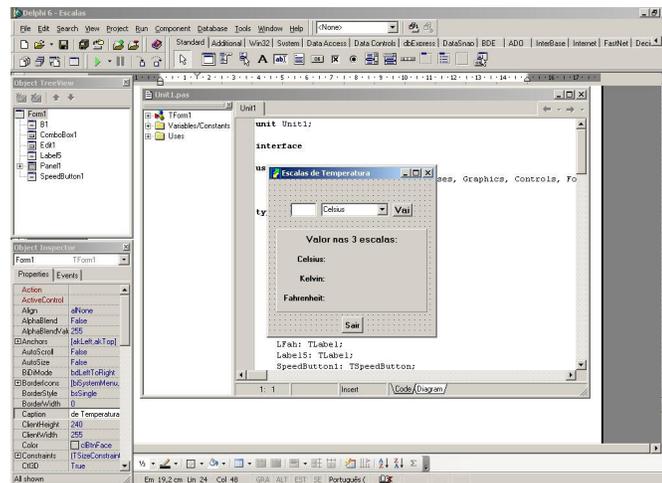


Figura 2 – Montagem visual de um software em um ambiente que utiliza GUI.

Destacam-se como linguagens OPP mais utilizadas o Object Pascal e o C++, tendo como exemplos de ambientes de desenvolvimento, o Delphi (Object Pascal), e o C++ Builder (C++).

3.5.3- Delphi e C++Builder

No ano de 1995, a empresa Borland lançou o ambiente de desenvolvimento (IDE) Delphi. Este ambiente integrado de desenvolvimento baseia-se na linguagem de programação orientada a objetos (OOP) Object Pascal. (Boratti, 2004)

Semelhante ao Delphi, mas possuindo uma linguagem de programação OPP diferente, o C++, foi lançado também pela empresa Borland, o IDE C++Builder. (Bueno, 2003)

Estes ambientes possuem interfaces gráficas (GUI) que facilitam a construção visual e programação de softwares. É disponibilizada, ao programador, uma grande quantidade de componentes, para as mais variadas funções, com a possibilidade de edição de eventos e propriedades destes componentes.

A Figura 3 apresenta uma tela capturada de uma das versões do ambiente Delphi:

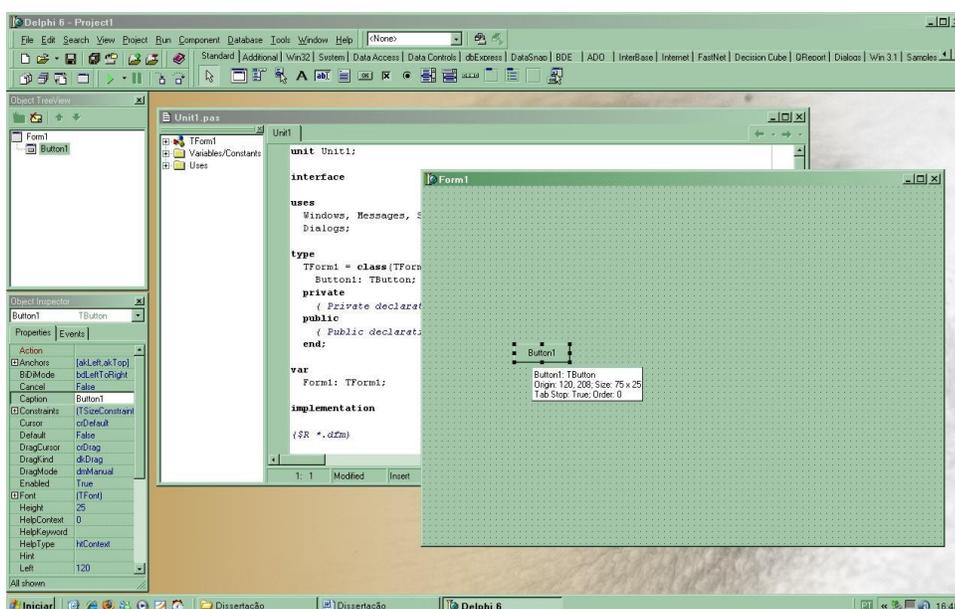


Figura 3 – Tela capturada do Delphi.

Nestes ambientes, conforme se observa na Figura 3, os componentes são organizados em uma paleta acima, e à esquerda encontra-se uma paleta com duas abas, a primeira por onde se acessa a edição das propriedades dos componentes, e uma segunda aba em que se encontra o acesso a programação de cada evento de um componente. Grande parte do desenvolvimento de um software é realizado através destas paletas.

Em versões mais recentes destas IDEs, os componente ficam em uma paleta à direita da tela.

No desenvolvimento do sistema apresentado nesta dissertação foram utilizados os IDEs Delphi e C++Builder.

3.5.4- Qualidade de Software

Um software deve possuir ao final de seu desenvolvimento um funcionamento facilitado, livre de falhas e correto conforme sua função. Estas são características que se busca e são pesquisadas em Engenharia de Software, em uma área de conhecimento já consolidada denominada “Qualidade de Software”.

Durante o desenvolvimento de um software, o programador deve estar ciente das várias características que tornam o software de qualidade, tentando ao máximo se aproximar de tais características.

Segundo a norma ISO/IEC 9126-1 (International Standard, 1997), um software de qualidade deve ter como características: Funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. A Tabela 3 apresenta cada uma destas características com suas sub-características, e respectivas descrições.

Tabela 3 – Características e Sub-características de um software de qualidade

Características	Sub-características	Descrição
<u>Funcionalidade:</u> <i>Funções atendendo as necessidades explícitas e implícitas de acordo com a finalidade do software</i>	Adequação	Funções apropriadas para as tarefas
	Acuraria	Gera resultados corretos e conforme previstos
	Interoperabilidade	Consegue interagir com sistemas propostos
	Conformidade	Está em conformidade com as normas ou convenções específicas
	Segurança de Acesso	Evita o acesso não autorizado a programas e dados
<u>Confiabilidade:</u> <i>Continuidade do desempenho ao longo do tempo nas condições estabelecidas para o software</i>	Maturidade	Freqüência com que ocorrem falhas.
	Tolerância a Falhas	Comportamento em caso de falhas
	Recuperabilidade	Capacidade de se restabelecer e recuperar dados em caso de falhas
<u>Usabilidade:</u> <i>Facilidade de utilização do software</i>	Inteligibilidade	Facilidade de entendimento dos conceitos utilizados no sistema
	Apreensibilidade	Facilidade de aprendizado na manipulação e dos conceitos envolvidos no software
	Operacionalidade	Facilidade na operação do software
<u>Eficiência:</u> <i>Recursos e tempos envolvidos no software conforme os requerimentos previstos de sua utilização</i>	Comportamento em relação ao Tempo	Tempo de resposta e de processamento
	Comportamento em relação a recursos	Quantidade de recursos de hardware que utiliza
<u>Manutenibilidade:</u> <i>Facilidade em correções de falhas, alterações e atualizações do software</i>	Analisabilidade	Facilidade na identificação de falhas e suas causas
	Modificabilidade	Facilidade de modificação e na remoção de falhas
	Estabilidade	Ausência de riscos de efeitos inesperados
	Testabilidade	Facilidade de ser testado
<u>Portabilidade:</u> <i>Capacidade de utilização do software em múltiplas plataformas com o mínimo esforço de adaptação</i>	Adaptabilidade	Capacidade de adaptar a outros ambientes
	Capacidade para ser Instalado	Facilidade na instalação
	Conformidade	Estar de acordo com normas e convenções de portabilidade
	Capacidade para substituir	Facilidade pela substituição por outro software

Adaptado de (Tsukumo *et al*, 1997)

Durante a pesquisa de desenvolvimento do sistema de instrumentação virtual, procurou-se adequar os softwares a estas características. No entanto, conforme é apresentado e analisado a frente, algumas sub-características ainda podem ser otimizadas.

3.6- Portas de Comunicação com Hardwares Externos

Qualquer comunicação entre um computador e dispositivos externos; como impressoras, scanner, mouses, teclados; é feita através de portas de comunicação de hardware¹.

Tais portas podem possuir funcionamento paralelo ou serial, onde no funcionamento paralelo a transferência de um conjunto de dados ocorre de forma simultânea, enquanto em portas seriais, esta transferência do conjunto de dados não é simultânea, ocorrendo de forma seqüencial (1 bit por vez) e em função do tempo.

Uma porta de comunicação muito utilizada é a porta paralela do computador, que, como já evidenciado no próprio nome, tem funcionamento paralelo. Outra porta também muito utilizada, e baseada numa tecnologia mais nova, é a porta USB (Universal Serial Bus), que possui funcionamento serial.

Tendo em visto a utilização destas duas portas de comunicação, na pesquisa de desenvolvimento do sistema de instrumentação virtual, segue-se um detalhamento sobre cada uma destas portas.

3.6.1- Porta paralela

A Porta paralela é um canal normalmente utilizado para comunicação com impressoras, podendo ser utilizada também para conexão de outros diversos dispositivos.

Consiste em um conjunto de pinos, os quais podem ter um estado de nível lógico alto, isto é, com um potencial elétrico de 5 volts, ou podem ter um estado de nível lógico baixo, onde se terá um potencial elétrico nulo, igual a 0 volt. Estes pinos são representados por bits, os quais para um nível lógico alto possuem um valor binário igual a 1, e para nível lógico baixo possuem um valor binário igual a 0.

¹ Cabe enfatizar que portas de comunicação de hardware são diferentes daquelas utilizadas para a comunicação em rede. As portas de comunicação em rede, descritas no tópico 3.1.3, são virtuais. As portas de comunicação de hardware, abordadas neste tópico, são portas físicas, nas quais equipamentos como impressoras, mouses, teclados, etc., podem ser conectados.

Neste canal de comunicação, os bits são agrupados em três tipos de endereços (378h/278h, 379h/279h e 37Ah/27Ah), onde cada um possui funções específicas para seus bits.

Estas funções são basicamente de saída ou entrada de dados. Bits de saída de dados são utilizados para efetuar o controle de dispositivos conectados a porta paralela, enquanto os bits de entrada de dados são utilizados para efetuar a leitura de informações enviadas por um dispositivo conectado a porta paralela.

A porta paralela possui modos de funcionamento unidirecional e bidirecional. O modo de funcionamento unidirecional SPP (Standard Parallel Port), é mais antigo, e alcança velocidades de transmissão de 150Kb/s. Neste modo, os endereços da Porta paralela são fixos, isto é, ou são somente utilizados para leitura ou para envio de dados. (Pessanha, 2006)

Neste modo, os bits de cada endereço possuem funções definidas, que são apresentadas na Tabela 4:

Tabela 4 – Distribuição e função dos bits para cada endereço da Porta paralela

Endereço		Nome	Função	Descrição
LPT1	LPT2			
378h	278h	Registrador de Dados	Envia dados de controle para um dispositivo	Byte (8 bits) de saída de dados
379h	279h	Registrador de Status	Ler o Status de um dispositivo	5 bits de entrada de dados
37Ah	27Ah	Registrador de Controle	Envia dados de controle para um dispositivo	Nibble (4 bits) de saída de dados

O modo bidirecional EPP (Enhanced Parallel Port), permite uma velocidade de comunicação de até 2Mb/s. Já o modo bidirecional ECP (Enhanced Capabilities Port) possui as mesmas características do modo EPP, mas utiliza o DMA (Direct Memory Access), isto é, tem acesso direto a memória, sem a necessidade do uso do processador. (Pessanha, 2006)

A diferença do modo unidirecional para o modo bidirecional está na utilização do endereço 378h/278h. No modo bidirecional este endereço pode ser utilizado tanto para a saída de dados, como canal de entrada de dados. Quando utilizado para leitura, os bits deste endereço possuem lógica invertida, isto é, quando é fornecido um sinal lógico alto a um dos pinos, o software que fizer a leitura deste pino identificará um nível lógico baixo, e logo, ao chegar um nível lógico baixo, o software identificará um nível lógico alto. Todos os bits deste endereço apresentarão este comportamento quando a função de leitura for ativada.

A opção pelo modo de funcionamento unidirecional ou bidirecional é feita através do “setup” do computador, e após habilitado o modo bidirecional, a escolha pelo uso do endereço 378h/278h como entrada ou saída de dados, se dará pelo estado de um “bit virtual” do endereço 37Ah, que seria o 5º bit deste endereço.

O software responsável pela manipulação da Porta paralela deverá, para habilitar a entrada de dados no endereço 378h/278h, atribuir ao bit virtual um nível lógico alto (1), e logo, para ativar a função de saída de dados no endereço 378h/278h, atribuir ao bit virtual um nível lógico baixo (0).

O total de bits disponíveis para uso como entrada e saída de dados, de acordo com o estado de funcionamento do endereço 378h/278h, é apresentado na Tabela 5:

Tabela 5 – Total de bits para saída e entrada de dados

Modo de Funcionamento	Total de bits para saída de dados (Controle)	Total de bits para entrada de dados (Leitura)
Unidirecional ou Bidirecional com 378h/278h em modo de saída de dados	12	5
Bidirecional com 378h/278h em modo de entrada de dados	4	13

Quanto maior o número de bits disponíveis para saída de dados, maior número de funções de controle de dispositivos podem ser implementadas. Já quanto maior o número de bits disponíveis para entrada de dados, maior o número de funções de leitura de dados podem ser implementadas, ou maior poderá ser a resolução de uma medida de algum valor analógico.

O conector da Porta paralela possui um total de 25 pinos, e cada pino ou representa um bit de um dos três endereços ou é um pino de aterramento. Os pinos de aterramento servem como referencial de potencial elétrico nulo.

O conector da porta paralela é denominado “DB25”, e pode ser do tipo Macho ou Fêmea, onde o conector DB25 Fêmea é o que se encontra no computador, e o conector DB25 Macho é o conector presente nos cabos que ligam um dispositivo à Porta paralela.

É demonstrado através da Figura 4, o posicionamento de cada pino para os conectores DB25 Fêmea e Macho.

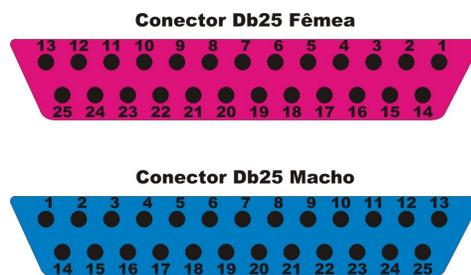


Figura 4 – Pinos do conector DB25.

Na Tabela 6 é apresentada a relação dos pinos e endereço ao qual pertencem.

Tabela 6 – Pinos e endereços dos quais pertencem

378h/278h		379h/279h		37Ah/27Ah		Aterramento
Pino	Nome	Pino	Nome	Pino	Nome	Pino
<u>2</u>	D0	--	S0	1	C0	18
<u>3</u>	D1	--	S1	14	C1	19
<u>4</u>	D2	--	S2	16	C2	20
<u>5</u>	D3	15	S3	17	C3	21
<u>6</u>	D4	13	S4	--	C4	22
<u>7</u>	D5	12	S5	V	C5	23
<u>8</u>	D6	10	S6	--	C6	24
<u>9</u>	D7	11	S7	--	C7	25

Os números em **negrito** identificam bits que possuem lógica invertida. Os números sublinhados identificam bits que terão lógica invertida quando utilizados para leitura, e lógica normal quando utilizados para envio de dados. O bit representado pela letra "**V**" é um bit virtual, que não possui um pino correspondente na porta paralela. Este bit é utilizado na habilitação do modo de saída ou entrada de dados do endereço 378h/278h

A porta paralela pode ser utilizada então, tanto para o controle de dispositivos externos, como para a coleta de dados destes dispositivos.

3.6.2- Porta USB

A porta USB (Universal Serial Bus) é uma porta de comunicação que vem sendo utilizada para a comunicação com diversos dispositivos, como impressoras, scanners, mouses, aparelhos celular, memórias flash (pendrivers), etc.

Esta porta de comunicação surgiu em 1995 devido a uma parceria entre as companhias Microsoft, Compaq, Hewlett-Packard, Intel, Lucent, Nec e Philips. Atualmente, a versão 2.0 desta tecnologia, lançada ao final de 2000, permite uma

comunicação de até 480Mbps. Além desta altíssima velocidade de tráfego de dados, outra grande vantagem da porta de comunicação USB é a possibilidade de se utilizar diversos dispositivos em um mesmo momento, em uma única porta USB. É possível, com o uso de Hubs USB, a conexão de até 127 dispositivos em uma mesma porta. (Messias, 2007)

Diferentemente da Porta paralela, que possui cada bit relacionado a um pino, a porta USB possui a informação de seus bits em função do tempo, em apenas dois dos quatro pinos da porta USB. A porta USB tem um funcionamento bidirecional, onde, a seleção de uso como controle ou leitura de dados é, também, realizada em função de um tempo.

Na comunicação USB, um “clock” interno do hardware controlador da porta USB gera pulsos periódicos, de forma que um intervalo entre o dois destes pulsos é caracterizado como um bit. Neste “clock” os pulsos são uma variação periódica de nível lógico no interior de um chip, onde se eleva a um nível lógico alto (1), e após este retorna a um nível lógico baixo (0).

Os pinos responsáveis pela transferência de dados terão uma diferença de potencial entre eles, e de acordo com as mudanças nesta diferença de potencial em sincronismo com os pulsos gerados pelo “clock”, será definido o nível lógico dos bits.

Este tipo de sincronismo é denominado codificação NRZI (No Return to Zero Inverted). Nesta codificação, quando ocorre uma variação da diferença de potencial nos pinos de transferência de dados, no momento em que o pulso do clock retorna ao nível lógico baixo (0), o bit relacionado àquele momento é definido com um nível lógico alto (1). Já quando não ocorre uma variação da diferença de potencial, o bit relacionado àquele momento é definido com um nível lógico baixo (0).

A Figura 5 apresenta uma ilustração do funcionamento da Codificação NRZI:

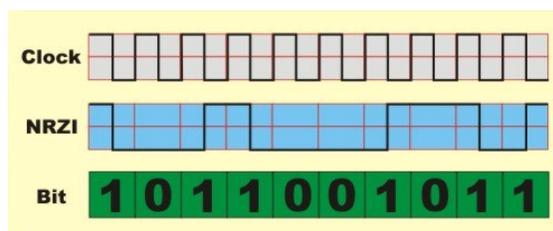


Figura 5 – Ilustração do funcionamento da codificação NRZI.

A porta USB presente em um computador possui um conector do tipo “A Fêmea”, no qual se conecta dispositivos através de cabos com um conector do tipo

“A Macho”. A Figura 6 demonstra os conectores USB tipo A, e a Tabela 7 apresenta a função de cada pino:

Conector Tipo A - Fêmea



Conector Tipo A - Macho



Figura 6 – Pinos do conector USB Tipo A.

Tabela 7 – Pinos e suas respectivas funções do conector USB tipo A

Pino	Nome	Cor padrão do fio	Função
1	VBus(5V)	Vermelho	Alimentação positiva da Porta USB
2	D+	Verde	Tráfego de dados
3	D-	Branco	Tráfego de dados
4	GND(0V)	Preto	Alimentação negativa da Porta USB

4 - METODOLOGIA

Neste quarto capítulo, será apresentado todo o processo de desenvolvimento e realização de testes de um sistema de instrumentação virtual de acesso a distância, adaptável a diversas aplicações.

O sistema possibilita o acompanhamento de procedimentos experimentais e de controle através de imagens em tempo real e do acesso a dados coletados nestes dispositivos, por meio de sensores. É possível ainda, com este sistema, o acesso multiusuário, isto é, o acesso simultâneo de vários usuários a um mesmo procedimento experimental ou de controle, permitindo inclusive a interação entre estes usuários.

O desenvolvimento da pesquisa ocorreu em seis etapas. Em uma primeira etapa foi definida a estrutura de funcionamento do sistema de instrumentação virtual. Nas etapas seguintes foram desenvolvidos e configurados todos os elementos de software e hardware que compõem o sistema. Foram criados ainda, cinco elementos (dois dispositivos de controle e três experimentos) para a realização de testes de funcionamento sistema. Em seguida, ao final da pesquisa, foram realizados os testes de funcionamento do sistema, além de testes de transmissão de dados.

Cada uma das seis etapas de desenvolvimento da pesquisa é apresentada a seguir, de forma detalhada:

4.1- Primeira Etapa: Definição da Estrutura de Funcionamento do Sistema de Instrumentação Virtual a ser desenvolvido

Esta etapa consistiu em um momento de planejamento, onde se buscou elaborar uma estrutura geral de funcionamento do sistema de instrumentação virtual.

O sistema é composto por um conjunto de elementos de software e hardware, e tais elementos, assim como a forma de interação entre estes diferentes elementos, foi definida nesta etapa.

Como produto desta etapa, foi elaborado o esquema geral, apresentado na Figura 7, o qual é explicado em seguida.

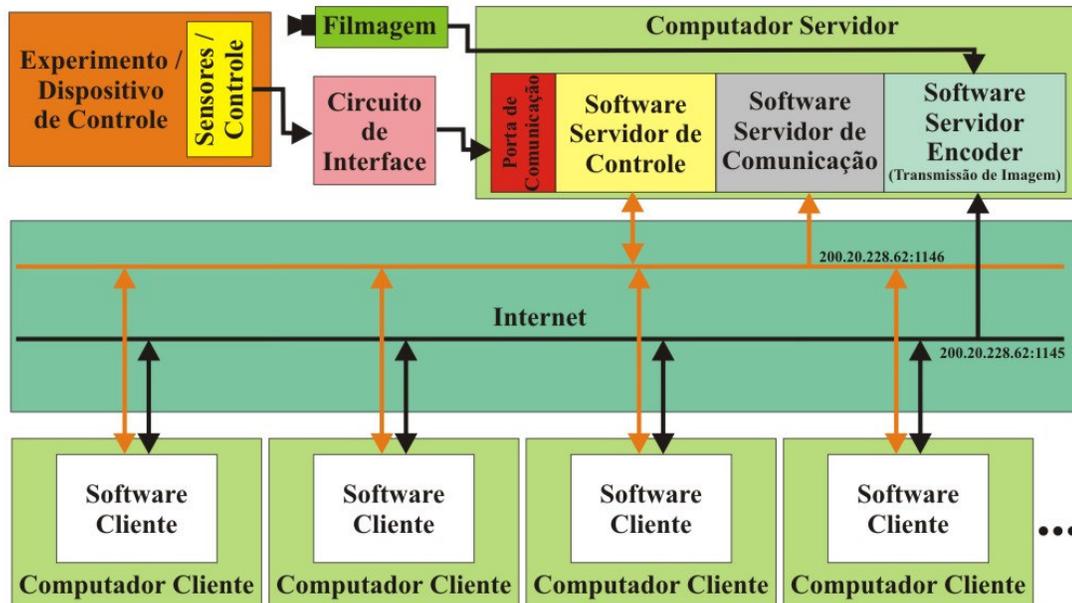


Figura 7 – Esquema Geral do Sistema.

Ao experimento ou dispositivo de controle são conectados sensores e/ou componentes de controle. Os sensores e componentes eletrônicos utilizados são variados, e de acordo com a necessidade de cada dispositivo. Alguns exemplos de componentes utilizados são termistores, sensores LDR, relés, transistores, reed switches, entre outros, que serão alvo de maiores explicações à frente.

Estes componentes se conectam a um circuito de interface (placa de aquisição e controle de dados), que permite a comunicação entre experimento ou dispositivo, e o computador. Este circuito, por sua vez, se conecta com o computador através de uma porta de comunicação (porta paralela ou porta USB). Para a comunicação via porta paralela, foi desenvolvido um circuito baseado no componente eletrônico ADC0804, e para a comunicação através da porta USB, foi utilizada a placa comercial R-Control30.

Em paralelo, uma câmera coleta imagens do experimento ou dispositivo, enviando estes dados ao computador através, também, de uma porta de comunicação (porta USB). A câmera utilizada foi uma Webcam CLONE 11086, de resolução máxima de imagem de 350K Pixel.

Neste computador operam três softwares servidores, um responsável pelo gerenciamento das imagens (Servidor Encoder), outro pelo gerenciamento de controle e coleta de dados (Servidor de Controle) e um último pelo gerenciamento de conexões de softwares clientes (Servidor de Comunicação).

O Servidor de Controle não possui comunicação direta com os Softwares Clientes. Esta comunicação é mediada pelo Servidor de Comunicação, de tal forma que para este, o Servidor de Controle é apenas mais um Cliente conectado. Foram criados dois Servidores de Controle, um para comunicação através da porta USB e um segundo para a comunicação através da porta paralela.

O Software Cliente se conecta aos Servidores de Comunicação e Encoder, recebendo e enviando dados. Os usuários do sistema utilizam o Software Cliente, sendo possível a conexão simultânea de vários destes usuários.

Os softwares Servidor de Comunicação, Servidor de Controle, e Cliente foram desenvolvidos na pesquisa.

O Software Encoder utilizado foi o Microsoft Windows Media Encoder 9.0 (Microsoft, 2007), de licença gratuita para versões originais do Windows XP e Vista.

4.2- Segunda Etapa: Desenvolvimento dos Softwares

Entre os softwares que compõem o sistema, apenas o Servidor Encoder não foi desenvolvido durante a pesquisa. Todos os demais softwares foram elaborados com a utilização dos ambientes de desenvolvimento (IDE) Delphi e C++Builder.

O IDE Delphi foi utilizado no desenvolvimento dos softwares Cliente, Servidor de Comunicação e Servidor de Controle via porta paralela; enquanto o IDE C++Builder foi utilizado para o desenvolvimento do software Servidor de Controle via porta USB.

No desenvolvimento dos softwares, foram implementadas diversas funções, como a comunicação entre os softwares por protocolo TCP/IP; funções que permitissem a adaptabilidade e o gerenciamento de procedimentos experimentais e de controle, entre outras funções.

Tais funções puderam ser implementadas com o uso de diversos componentes disponíveis nos IDEs Delphi e C++Builder. A seguir, é apresentado o processo de implementação de cada uma das funções disponíveis nos softwares.

4.2.1- Principais componentes no desenvolvimento dos softwares

Conforme já abordado nos tópicos 3.5.2 e 3.5.3, os IDEs Delphi e C++Builder possuem componentes pelos quais podem ser programadas diversas funções. No desenvolvimento dos softwares, alguns componentes se destacam

pelas funções que estes executam, ou por estarem presentes no desenvolvimento de todos os softwares.

Os componentes `IdTCPClient` e `IdTCPServer` foram utilizados na implementação da função de conexão entre softwares numa rede local ou internet. Estes componentes se comunicam entre si através de protocolo TCP/IP, permitindo a comunicação entre os softwares.

O componente `IdTCPServer` é utilizado no desenvolvimento de softwares servidores, os quais, quando em funcionamento, devem esperar passivamente por uma conexão ou solicitação de softwares cliente. (Hower, 2004)

Já o componente `IdTCPClient` é utilizado no desenvolvimento de softwares clientes, que são os softwares que se conectarão a um servidor. O componente `IdTCPClient` é o responsável pelo envio e recepção de dados na interação de um software cliente com um software servidor. (Hower, 2004)

A Figura 8 apresenta um esquema representativo da interação entre softwares Cliente e o software Servidor, desenvolvidos com o uso destes componentes.

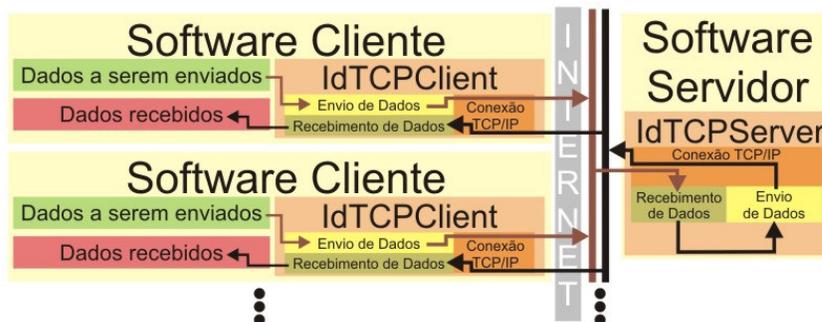


Figura 8 – Esquema da interação entre softwares cliente e servidor.

Outros componentes de destaque são o `OpenDialog` e o `SaveDialog`. Estes componentes permitem, respectivamente, que sejam abertos e importados dados de um arquivo, e que possa ser salvo um arquivo com informações do software.

O componente `OpenDialog`, quando em ação, possibilita a abertura de uma janela de diálogo pela qual pode ser selecionado um arquivo a ser aberto. Após escolhido o arquivo, os dados contidos neste podem ser importados pelo software.

Quando o componente `SaveDialog` é utilizado, é possível também a abertura de uma janela de diálogo, pela qual pode ser escolhido o nome do arquivo e o diretório do computador onde se pretende salvar um arquivo com um conteúdo definido.

Os componentes OpenFileDialog e SaveDialog podem ser configurados para permitir o acesso somente a arquivos com determinadas extensões. Com isto, é possível definir a extensão dos arquivos que armazenam informações dos softwares desenvolvidos.

Estes componentes, OpenFileDialog e SaveDialog, foram utilizados para variados propósitos, durante o desenvolvimento de cada software, e esta utilização será detalhada em tópicos à frente.

Outro componente, o Timer, foi utilizado para diversas funções no desenvolvimento dos softwares. É um componente pelo qual podem ser programadas ações dependentes do tempo. A função do componente em cada software também será apresentada adiante; no detalhamento do desenvolvimento de cada um dos softwares.

4.2.2- Software Cliente

É através do Software Cliente que estudantes ou usuários podem acessar o sistema, sendo possível o acompanhamento de procedimentos experimentais e de controle através de imagens dos dispositivos e de dados coletados nestes dispositivos. É possível ainda o envio de comandos de controle, e a interação entre os usuários conectados ao sistema.

No desenvolvimento do Software Cliente foi considerada a utilização do componente IdTCPClient para permitir a interação com o Software Servidor de Comunicação; além do componente WebBrowser para possibilitar o acesso as imagens enviadas através do Servidor Encoder.

O componente IdTCPClient deve ser configurado, no desenvolvimento do software ou em tempo de execução (configuração dinâmica na execução do software), fornecendo a este um nome para o Cliente, o endereço IP do servidor e a porta virtual de comunicação utilizada pelo servidor.

No software desenvolvido, esta configuração ocorre em tempo de execução. O usuário do software deve informar o endereço e porta de comunicação, para somente assim, efetuar a conexão. O software já possui uma configuração padrão, com o endereço IP do servidor como sendo “200.20.228.62” e Porta virtual de Comunicação como sendo “1146”, dados estes relativos ao software Servidor de Comunicação que opera na UENF.

No envio de dados ao Servidor, as mensagens ou comandos do usuário do Software Cliente são direcionados ao componente IdTCPClient, que por sua vez envia tais dados ao servidor.

Os dados enviados pelo servidor aos clientes, são recebidos pelo componente IdTCPClient, e armazenadas numa memória temporária. Um componente de ação temporal (Timer) é o responsável pela solicitação periódica destes dados ao componente IdTCPClient. Tais dados podem ser mensagens enviadas por outros usuários do sistema ou pelos Softwares Servidor de Controle, ou mesmo informações do Servidor de Comunicação, como a listagem dos usuários conectados ao sistema. O Timer direciona as mensagens de texto para um componente visual de apresentação de textos (Memo), e a listagem dos usuários conectados para um componente visual de apresentação de listas (ListBox). O componente Timer foi configurado para efetuar estas ações a cada décimo de segundo.

A Figura 9 demonstra um esquema de representação do funcionamento do componente IdTCPClient e dos demais componentes auxiliares.

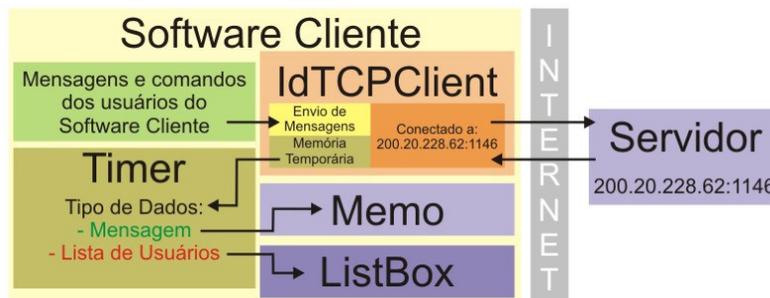


Figura 9 – Esquema de funcionamento do componente IdTCPClient para o software Cliente.

Para que o usuário possa efetuar o controle dos dispositivos que são acessados a distância, foi implementada a função de envio de comandos, de duas formas: Envio por linha de comando ou por botões de comando.

Estas duas formas se diferenciam basicamente pelo componente que o usuário utiliza para o envio do comando ao componente IdTCPClient. Para o modo de linha de comando, através do campo de envio de mensagens o usuário pode escrever e enviar os comandos. Já para o modo de envio por botões, foi implementado no software uma seção, onde é possível atribuir um comando para

cada um dos vinte botões disponíveis, e uma vez acessados estes botões, os comandos a eles relacionados são enviados ao componente IdTCPClient.

Nesta Seção de Botões, foram elaboradas ainda as opções para carregar e salvar uma configuração para os botões. Para este fim foi utilizado os componentes OpenFileDialog e SaveDialog.

A configuração dos botões de comandos, onde se atribui para cada botão um determinado comando, pode ser salva em um arquivo, devido a utilização do componente SaveDialog.

Os dados de configuração dos botões, salvos em um arquivo criado, podem ser acessados, devido a utilização do componente OpenFileDialog.

Durante a programação do Software Cliente, os componentes OpenFileDialog e SaveDialog foram configurados para utilizar (acessar e criar) somente arquivos com extensão “.est”. Desta forma, foram definidos os arquivos com esta extensão como sendo arquivos de configuração de botões de comandos do Software Cliente.

A Figura 10 apresenta um esquema representativo da função de envio de comandos, além da opção de configuração dos botões:

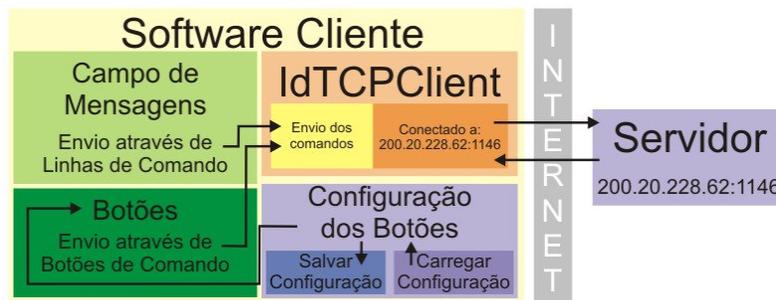


Figura 10 – Esquema representativo da função de envio de comandos.

Uma outra importante função implementada no Software Cliente é o acesso às imagens coletadas dos dispositivos. Isto é possível com a utilização do componente WebBrowser, e de um arquivo externo, o “imagem.toe”.

O componente WebBrowser consiste em um componente visual, que executa e apresenta arquivos HTML, ou arquivos que tenham sido escritos com esta linguagem. É um componente comum no desenvolvimento de navegadores de internet feitos nos IDEs Delphi e C++Builder.

O arquivo “imagem.toe” foi escrito em HTML, e em sua programação foi incluído o acesso as imagens do experimento. Foi configurado o acesso ao Servidor

Encoder, incluindo na programação o endereço IP e a porta virtual de comunicação, como sendo, respectivamente, “200.20.228.62” e “1145”, dados estes relativos ao Servidor Encoder que opera na UENF.

Logo, o componente WebBrowser apresenta as imagens em vídeo dos dispositivos, através da execução do arquivo “imagem.toe”.

Na programação do software, o acesso às imagens do experimento está vinculado ao status de conexão ao Servidor de Comunicação, ou seja, somente quando o componente IdTCPCliente está conectado ao Servidor de Comunicação, é que o componente WebBrowser disponibiliza as imagens dos dispositivos, executando o arquivo “imagem.toe”.

Quando a conexão não está ativa, o componente WebBrowser carrega o arquivo “noconnect.toe”, que também foi escrito com a linguagem HTML, e quando executado apresenta uma mensagem informando que o software não está conectado ao Sistema. Os arquivos “imagem.toe” e “noconnect.toe” ficam localizados no mesmo diretório do Software Cliente.

A Figura 11 apresenta um esquema que representa o funcionamento do componente WebBrowser, e dos arquivos “imagem.toe” e “noconnect.toe”.



Figura 11 – Esquema de funcionamento do componente WebBrowser e arquivos “imagem.toe” e “noconnect.toe”.

Outra função, o acesso à conteúdos teóricos, foi elaborada através de um componente comum, um botão. Na programação do evento “OnClick” do botão foi inserido um código que abre no navegador padrão do Windows, um site. Este site oferece informações sobre os dispositivos de controle e experimentos, além de textos explicativos sobre os conceitos de Física envolvidos nos experimentos.

As principais funções implementadas no desenvolvimento do Software Cliente são apresentadas a seguir:

- Conexão ao Servidor de Comunicação;

- Apresentação de listagem de usuários conectados;
- Envio e recebimento de mensagens;
- Envio de comandos (Linhas de comandos e Botões de Comandos);
- Salvar e carregar configuração dos botões de comandos;
- Conexão ao Servidor Encoder;
- Opção de visualização ou não das imagens dos dispositivos;
- Acesso aos “Conteúdos Teóricos”;
- Carregamento de esquemas de cores do software;
- Carregamento automático de configurações iniciais;
- Acesso à seção de ajuda do Software.

As funções “Carregamento de esquemas de cores do software”, “Carregamento automático de configurações iniciais” e “Acesso à seção de ajuda do Software” são comuns à todos os software desenvolvidos, e o desenvolvimento destas funções será apresentado no tópico 4.2.6 deste capítulo.

4.2.3- Software Servidor de Comunicação

O Software Servidor de Comunicação é responsável por concentrar a conexão dos clientes, redirecionando todos dados que chegam ao servidor, aos clientes conectados a ele.

Vários componentes foram utilizados no desenvolvimento do Software Servidor de Comunicação, como botões, caixas de texto, painéis e um gerenciador de comunicação, o IdTCPSTerver.

Com a utilização do componente IdTCPSTerver é possível que os softwares clientes se conectem ao Software Servidor de Comunicação, além de permitir o gerenciamento das conexões.

O componente deve ser configurado durante a programação, ou mesmo durante o uso do software, fornecendo a ele o nome do Servidor e a porta de comunicação que ele deverá utilizar.

No software Servidor de Comunicação, foram definidos como padrões “1146” e “Servidor” como, respectivamente, porta virtual de comunicação e o nome do Servidor.

Quando um Cliente solicita a conexão, o componente IdTCPSTerver permite esta conexão, envia mensagens informando que a conexão foi iniciada, e adiciona o nome do cliente na listagem dos usuários conectados ao Sistema (ListBox).

Qualquer mensagem enviada pelos usuários, ou pelo Servidor de Controle (que também se conecta ao Servidor de Comunicação) é redirecionada para todos os usuários conectados ao Sistema.

O esquema apresentado na Figura 12, representa o funcionamento do componente IdTCPServer.

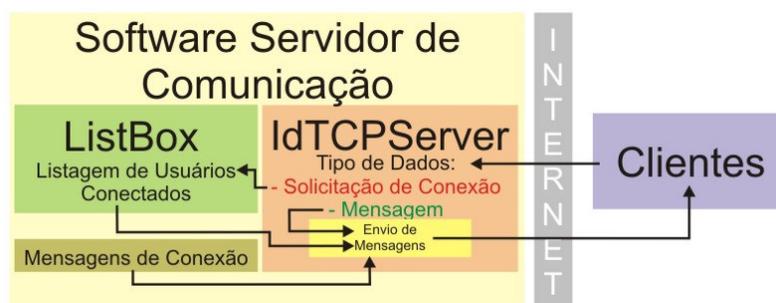


Figura 12 – Esquema de funcionamento do componente IdTCPServer.

A função “Mensagens de Conexão” (Configuração de Mensagens de Entrada) foi elaborada para permitir que sejam enviadas mensagens a um cliente no momento em que ele se conecta ao servidor. A mensagem a ser enviada pode ser editada e salva, ou mesmo carregada de um arquivo.

Para isto foram utilizados os componentes OpenFileDialog e Save Dialog, os quais foram configurados durante a programação, para acessarem somente arquivos de extensão “.txt”.

Além disto, uma outra função permite que sejam editadas “palavras-chave”, e informações relacionadas a estas (Configuração de Informações). No momento que um cliente envia ao servidor uma palavra-chave configurada, a informação a ela relacionada é enviada ao cliente pelo servidor.

As palavras-chaves assim como as informações a estas relacionadas podem ser salvas, devido a utilização dos componentes SaveDialog e OpenFileDialog, os quais permitem salvar e carregar tais dados. Estes componentes foram configurados também para permitir o acesso, somente, aos arquivos de extensão “.txt”.

No Software Servidor de Comunicação foi implementada ainda a função de gerenciamento das conexões, onde é possível ações para os clientes como o envio de mensagens privadas ou a desconexão de um cliente, e a função de acesso ao endereço IP ou nome do computador numa rede local, de cada cliente. As funções

foram implementadas com a utilização de recursos oferecidos pelo componente IdTCPServer.

Todas as funções implementadas durante o desenvolvimento do Software Servidor de Comunicação são apresentadas a seguir:

- Configuração da porta virtual para a comunicação do sistema;
- Gerenciamento e apresentação da lista de usuários (nome dos clientes) conectados;
- Redirecionamento de mensagens para os clientes;
- Configuração de mensagem de conexão;
- Configuração de “palavras-chave” e informações associadas a estas;
- Gerenciamento de conexão – Envio de mensagens privadas e desconexão de clientes;
- Identificação do endereço IP ou nome do computador na rede local, do computador Servidor;
- Carregamento de esquemas de cores do software;
- Carregamento automático de configurações iniciais;
- Acesso à seção de ajuda do Software.

Conforme já comentado, por serem comuns a todos os softwares desenvolvidos, as funções “Carregamento de esquemas de cores do software”, “Carregamento automático de configurações iniciais” e “Acesso à seção de ajuda do Software”, será apresentado adiante, no tópico 4.2.6 deste capítulo, o desenvolvimento destas funções.

4.2.4- Software Servidor de Controle – USB

O Software Servidor de Controle (USB) foi desenvolvido para permitir o acesso aos dados coletados e o controle de experimentos e dispositivos. Este software é responsável pela interação, através da porta USB, com um circuito de interface, o qual será apresentado no tópico 4.4.1.

Diferentemente dos demais softwares, que foram desenvolvidos com o IDE Delphi, o Software Servidor de Controle (USB) foi desenvolvido com o IDE C++Builder. A opção por esta IDE ocorreu devido a disponibilidade de dois arquivos que foram fornecidos em conjunto com a Placa R-Control30, que é o circuito de interface utilizado para a comunicação através da porta USB.

Os arquivos “FTD2XX.H” e “ftd2xx.lib” possuem rotinas de programação na linguagem C e C++, e assim, o desenvolvimento do software foi facilitado com a utilização da IDE C++Builder.

Este software não possui uma conexão direta com os softwares clientes. A comunicação entre Software Servidor de Controle e um Software Cliente ocorre através do Software Servidor de Comunicação. O Servidor de Controle se conecta ao Servidor de Comunicação da mesma forma que um Software Cliente.

Logo, a implementação da função de conexão, foi semelhante à do Software Cliente. Apesar de ser um IDE diferente, o componente IdTCPClient também é disponibilizado para o IDE C++Builder.

Assim como no Software Cliente, para se conectar, é informado ao Software Servidor de Controle o endereço e porta de comunicação, além do nome de conexão (nome de cliente) do software. O software também possui uma configuração padrão, com o endereço IP do servidor como sendo “200.20.228.62” e Porta virtual de Comunicação como sendo “1146”, e nome de conexão como “Experimento”, sendo o endereço IP e a porta virtual de comunicação relativa ao software Servidor de Comunicação que opera na UENF.

Quando solicitada a conexão ao sistema, estes dados são fornecidos ao componente IdTCPClient, que efetua a conexão com o Software Servidor de Comunicação.

O envio de dados ocorre da mesma forma que no Software Cliente, através do componente IdTCPClient, com o funcionamento representado na Figura 10. A única diferença está na adição de mais um componente que encaminha dados ao IdTCPClient para que sejam enviados. Foi adicionado um componente Timer, que verifica periodicamente de forma automática, se existem dados de experimentos e dispositivos de controle a serem enviados, e existindo, encaminha para ao IdTCPClient para o envio.

Já a recepção de dados possui além da análise de tipo de dados que chega (mensagens ou lista de usuários), um filtro que verifica se entre as mensagens que chegam existe algum comando de controle direcionado ao experimento ou dispositivo de controle.

Este filtro efetua uma comparação entre os dados recebidos e os comandos cadastrados no software. Além disto, o filtro verifica se o usuário que está enviando o comando está habilitado para efetuar o controle. Caso esteja habilitado será

enviado para o circuito de interface R-Control-30, através da porta USB, os dados de controle do dispositivo.

A Figura 13 demonstra um esquema de representação do funcionamento do componente IdTCPClient e dos demais componentes auxiliares, além do filtro implementado.



Figura 13 – Esquema de funcionamento do componente IdTCPClient e filtro de verificação de comandos.

Foi implementada a função de cadastro de comandos, onde podem ser definidos até vinte comandos, relacionados a dez saídas de controle de uma placa (circuito auxiliar, que se conecta a placa R-Control30). É possível a configuração de um comando para ligar e outro para desligar cada uma das dez saídas, permitindo desta forma o controle dos dispositivos conectados a estas saídas.

Outra função envolve a coleta de dados dos dispositivos. Existem dois modos de coleta de dados, um para leitura de grandezas físicas, e outro de leitura do estado dos dispositivos.

No modo de leitura de grandezas físicas, podem ser configurados os quatro canais de coleta de dados da placa R-Control-30. A configuração consiste na definição da grandeza física envolvida e sua unidade física, além de informações para o ajuste linear, através de um cálculo de interpolação, do valor coletado.

No modo de leitura de estado de dispositivos, os canais de coleta de dados são utilizados para informar se o dispositivo, ou partes dele, está ativo ou não. É possível configurar até dezesseis informações de estado.

Com outra função elaborada, a de ações automáticas condicionadas a um estado, é possível configurar uma ação automática (ligar ou desligar umas das dez saídas) para cada um dos dezesseis estados de dispositivos que podem ser configurados. Desta forma, quando o software identifica uma das dezesseis situações configuradas, uma das saídas poderá ser ligada ou desligada.

O software é configurado de acordo com o experimento ou dispositivo de controle que será utilizado. Pela quantidade de detalhes de configuração que devem ser observadas, e estas específicas para cada procedimento experimental e de controle, foi elaborada uma função que permite que se salve ou carregue estas configurações.

Para isto foram utilizados os componentes OpenFileDialog e SaveDialog. Foi definido durante a programação destes componentes, que estes acessariam a arquivos de extensão “.exp”.

Todas as funções implementadas para este software se seguem:

- Conexão ao Servidor de Comunicação;
- Apresentação de listagem de usuários conectados;
- Envio e recebimento de mensagens;
- Filtro de mensagens para identificação de comandos;
- Configuração de usuário que pode controlar o dispositivo conectado;
- Configuração de comandos;
- Controle em função dos comandos;
- Coleta de dados dos dispositivos;
- Configuração do tipo de dados coletados nos dispositivos;
- Salvar e carregar configurações;
- Carregamento de esquemas de cores do software;
- Carregamento automático de configurações iniciais;
- Acesso à seção de ajuda do Software.

4.2.5- Software Servidor de Controle – Paralela

O Software Servidor de Controle (Paralela) foi desenvolvido para permitir o acesso aos dados coletados e o controle de experimentos e dispositivos. Este software é responsável pela interação, através da porta paralela, com um circuito de interface (CI-PP), o qual será apresentado no Tópico 4.4.2.

O software foi desenvolvido no IDE Delphi, e com a utilização de um arquivo dll, de uso livre, o qual possui funções de acesso a porta paralela. Na programação do software foi elaborada uma lógica que possibilita a utilização das funções “inportb” e “outportb” do arquivo “inpout32.dll” (Logix4u.net, 2003).

Através da função “inportb” é efetuada a leitura dos endereços da porta paralela, efetuando assim a leitura dos dados que são enviados para o computador

através desta porta de comunicação. O comando “outportb” permite que dados sejam enviados para endereços da porta paralela, possibilitando o controle de experimentos e dispositivos.

Este arquivo dll fica armazenado no mesmo diretório do executável do software desenvolvido.

As funções do Software Servidor de Controle (Paralela) são semelhantes ao desenvolvido para comunicação através da porta USB.

O Software Servidor de Controle (Paralela) não se comunica diretamente com os softwares clientes. Esta comunicação é também intermediada pelo Servidor de Comunicação.

Foi utilizado também o componente IdTCPClient, e os dados de conexão que devem ser informados ao Software são o endereço e porta de comunicação, além do nome de conexão do software. A configuração padrão para estas informações é também “200.20.228.62” para o endereço IP do servidor e “1146” para a porta virtual de comunicação, sendo “Experimento” o nome de conexão.

O envio e coleta de dados é semelhante ao Servidor de Controle (USB), com a utilização de um componente Timer que verifica periodicamente, de forma automática, se existem dados de experimentos e dispositivos de controle a serem enviados, encaminhando-os ao IdTCPClient para o envio; e um filtro de verificação das mensagens que chegam, em busca de comandos.

A Figura 13, apresentada no Tópico 4.2.4, também representa o funcionamento do componente IdTCPClient e do filtro de comandos, para o Servidor de Controle (Paralela).

O circuito de Interface com o qual o software interage foi desenvolvido durante a pesquisa, e será apresentado no Tópico 4.4.2.

Na função de cadastro de comandos, podem ser definidos até oito comandos, relacionados a saída de dados do endereço 37AH da porta paralela. Esta saída é composta por quatro bits, dos quais, três foram utilizados, que combinados podem definir oito números binários, possibilitando o controle de até oito saídas de controle. O número binário representado por estes três bits é informado a um componente eletrônico do circuito de interface, que com base no número, controla as oito saídas, sendo possível ligar e desligar cada uma delas.

Existem também dois modos de coleta de dados, um para leitura de grandezas físicas, e outro de leitura de Status dos dispositivos. No entanto, para o

modo de leitura de grandezas físicas, somente é configurado um canal de coleta de dados, definindo-se a grandeza física envolvida e sua unidade física, e informações para o ajuste linear, através de um cálculo de interpolação, do valor coletado.

Pelo circuito de interface para a porta paralela possuir apenas um canal de coleta de dados, o software permite a configuração e acesso de apenas este único canal.

Já na função elaborada para o modo de leitura de estado de dispositivos, é possível, também a configuração de até dezesseis informações de estado, além de dezesseis ações automáticas condicionadas a cada estado configurado.

Foi implementada também a função que permite salvar e carregar configurações, com a utilização dos componentes OpenFileDialog e SaveDialog, configurados para o acesso a arquivos de extensão “.exp”.

Todas as funções implementadas para este software se seguem:

- Conexão ao Servidor de Comunicação;
- Apresentação de listagem de usuários conectados;
- Envio e recebimento de mensagens;
- Filtro de mensagens para identificação de comandos;
- Configuração de usuário que pode controlar o dispositivo conectado;
- Configuração de comandos;
- Controle em função dos comandos;
- Coleta de dados dos dispositivos;
- Configuração do tipo de dados coletados nos dispositivos;
- Salvar e carregar configurações;
- Carregamento de esquemas de cores do software;
- Carregamento automático de configurações iniciais;
- Acesso à seção de ajuda do Software.

4.2.6- Funções comuns entre os softwares desenvolvidos

Os softwares desenvolvidos possuem algumas funções em comum. Um exemplo é a função de “Carregamento de Esquema de Cores” que foi adicionada em todos os softwares.

Esta função permite que o usuário do software possa modificar as cores do software, através do carregamento de um esquema de cores. Para isto foi utilizado o componente “OpenDialog”.

Durante o desenvolvimento dos softwares, um botão foi programado para interagir com este componente OpenFileDialog.

Esta interação ocorre em dois momentos: Em um primeiro momento o botão solicita ao componente OpenFileDialog que seja aberta a janela de diálogo para a escolha de um arquivo; No segundo momento, após a seleção do arquivo, o conteúdo deste é importado, e com base neste conteúdo, as cores dos componentes visuais do software são definidas.

Tanto no Delphi, como no C++Builder, as cores dos componentes visuais são definidas pela propriedade “Color”, através de um número hexadecimal. Sendo assim, os arquivos de esquema de cores possuem um conjunto de números hexadecimais, que ao serem acessados, são direcionados para a propriedade “Color” dos componentes, definindo assim as cores do software.

Foi feita ainda, durante a programação dos softwares, a configuração do componente OpenFileDialog, para permitir o acesso somente a arquivos com a extensão “.cor”. Foram criados seis arquivos com esta extensão (seis esquemas de cores), os quais podem ser utilizados por todos os softwares desenvolvidos.

A lógica da função Carregamento de Esquemas de Cores é representada pela Figura 14:

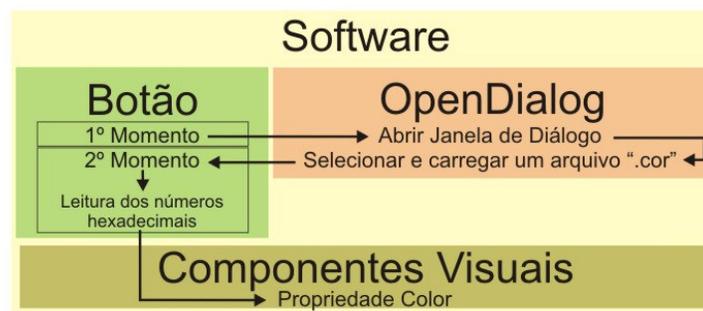


Figura 14 – Representação da lógica da função Carregamento de Esquemas de Cores.

Uma outra função comum entre os softwares desenvolvidos, é a utilização de arquivos de inicialização “.ini” (carregamento de configurações iniciais). Cada software, quando finalizado, armazena em dois arquivos, “configcolor.ini” e “configconex.ini”, respectivamente, os códigos hexadecimais relativos as cores do software e os dados de conexão utilizados.

No momento que os softwares são inicializados, ocorre o carregamento automático das informações presentes nestes arquivos, e a aplicação no software.

Para os softwares Cliente e Servidor de Controle, os dados de conexão armazenados no arquivo “configconex.ini” são o endereço IP do Servidor de Comunicação e a Porta Virtual de Comunicação. Já para o Servidor de Comunicação, são armazenados no “configconex.ini” o nome do Servidor e a Porta Virtual de Comunicação.

Uma última função comum aos softwares desenvolvidos, é a Seção de Ajuda. Esta Seção permite conteúdos explicativos sobre o funcionamento do software.

Foi adicionado um botão nos softwares, no qual foi programado para o evento “OnClick” (“Clicar” do botão), o acesso a um arquivo HTML, que fica armazenado no mesmo diretório do software. Quando solicitado, este arquivo HTML de ajuda é aberto no navegador padrão do sistema operacional.

Foi criado um arquivo HTML específico para cada software desenvolvido, onde se buscou explicar cada uma das funções e configurações possíveis do software.

4.3- Terceira Etapa: Configuração do Software Encoder

Foi necessário que se configurasse o software encoder para que fosse possível a transmissão das imagens dos experimentos e dispositivos de controle.

Inicialmente foi criada uma sessão de transmissão de eventos ao vivo (Figura 15), e em seguida foi selecionada a origem das imagens, ou seja, a câmera utilizada (VIMICRO USB PC Camera), conforme apresentado na Figura 16.

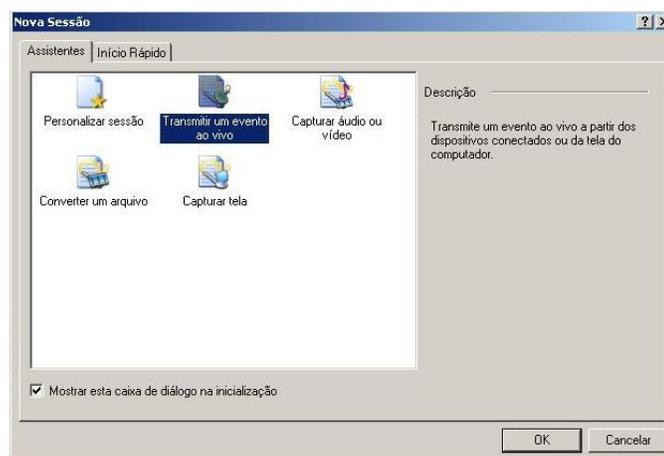


Figura 15 – Servidor Encoder: Criando uma nova sessão de transmissão ao vivo.



Figura 16 – Servidor Encoder: Configurando origem de imagem.

Foi habilitada ainda a permissão para que os clientes acessem as imagens (Figura 17), e foi configurada a porta virtual de comunicação em rede (Figura 18) como sendo a “1145”.



Figura 17 – Servidor Encoder: Configurando do método de difusão de imagem.

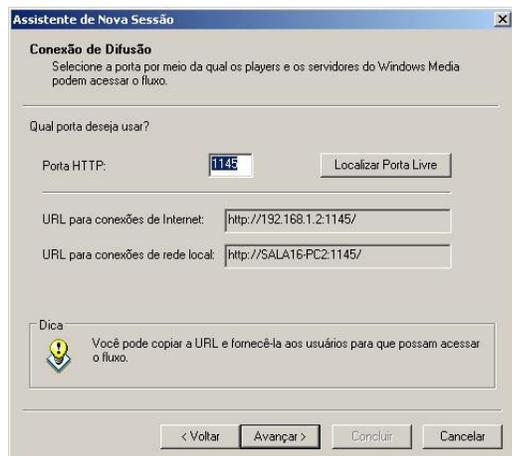


Figura 18 – Servidor Encoder: Configurando a porta de comunicação em rede.

Por fim, foi configurada a taxa de transferência (bits) em 300Kbps, a taxa de atualização de imagens em 29,97 quadros/s, e o tamanho das imagens em 320x240 Pixels, conforme pode ser observado na Figura 19.

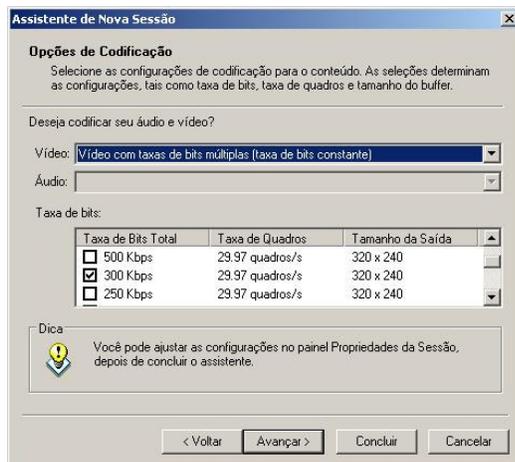


Figura 19 – Servidor Encoder: Configurando taxas de bits e quadros e tamanho da imagem.

A Figura 20, a seguir, apresenta o software encoder durante o seu funcionamento, na transmissão de imagens de um dos experimentos:

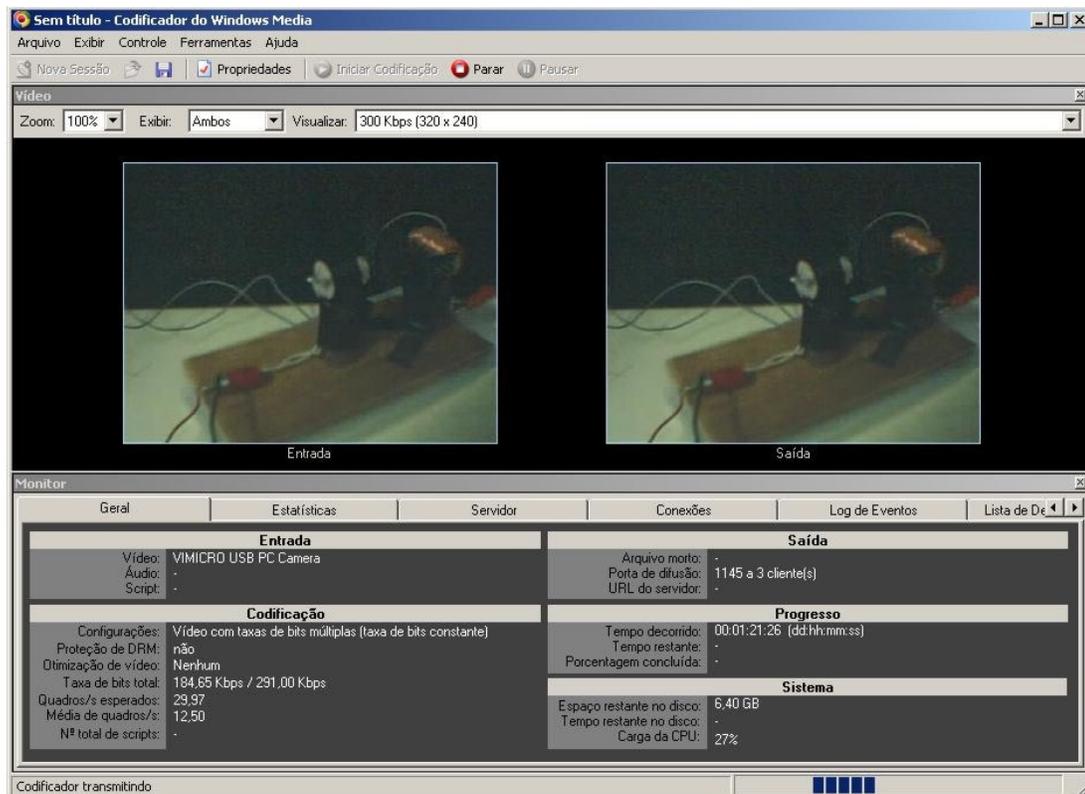


Figura 20 – Software Encoder em Funcionamento.

4.4- Quarta Etapa: Configuração e desenvolvimento de Circuitos de Interface

Conforme já comentado, é necessária a utilização de circuitos de interface, para permitir a comunicação entre o computador e os dispositivos de controle e experimentos.

Estes circuitos são elaborados de acordo com a porta de comunicação do computador a qual serão conectados. Como foram consideradas as portas de comunicação Paralela e USB, foi necessário um circuito para cada caso.

4.4.1- Circuito de Interface: Porta USB

Para a comunicação através da porta USB foi considerada a utilização de um circuito de Interface comercial, a Placa R-Control30 da Rogercom. (RogerCom.com, 2007).

Esta placa possui diversas funções de controle e coleta de dados, que são apresentadas a seguir:

- Oito pinos de saída de dados TTL (25mA/5V);
- Quatro relés para o controle de dispositivos que funcionem com tensão elétrica de até 240V e corrente elétrica de até 7 A;
- Conversor digital-analógico de 1 canal e resolução de 8 bits, para o controle de uma saída de dados analógicos de tensão elétrica variando entre 0V e 5V;
- Uma saída PWM (Pulse With Modulation) com frequência fixa em torno de 19,52 KHz, tensão elétrica variando entre 0V e 12V com resolução de 10 bits, e corrente elétrica de até 150mA;
- Oito pinos de entrada de dados TTL (25mA/5V);
- Quatro entradas fotoacopladas, para a leitura de dados de estado (Ligado ou desligado), com detecção por meio de sinais analógicos com valor de tensão elétrica que varie entre 5V e 12V;
- Conversor analógico-digital de 4 canais e resolução de 8 bits, para leitura de valores de tensão elétrica entre 0V e 5V.

Uma foto desta placa de interface é apresentada na Figura 21, a seguir:



Figura 21 – Foto da placa R-Control30.

Entre as funções de controle e coleta de dados, foi utilizado para o Sistema de Instrumentação Virtual, o Conversor analógico-digital (A/D) de 4 canais e o Conversor digital-analógico (D/A) de 1 canal.

O conversor A/D foi utilizado para a leitura de até quatro sensores ou para a leitura de estado (ligado ou desligado) de até dezesseis dispositivos. Os sensores são utilizados em procedimentos onde se faz necessário a medição de grandezas físicas, enquanto a leitura de estado foi utilizada para a identificação de ações dos experimentos ou dispositivos de controle.

O conversor D/A foi utilizado para ligar e desligar experimentos e dispositivos de controle, ou seja, para efetuar o controle destes elementos.

Foi elaborado durante a pesquisa, um circuito auxiliar de controle, que de acordo com a diferença de potencial fornecida pelo conversor D/A, é ligada uma de suas dez saídas.

O circuito auxiliar de controle (CAC-USB) possui como componente eletrônico principal o Driver LM3914. Este componente possui dez saídas analógicas que podem ser ligadas ou desligadas, de acordo com a diferença de potencial que é aplicada em dois pinos do componente. (National Semiconductor, 2003)

Desta forma, a saída do conversor D/A da placa R-Control30 foi conectada ao circuito auxiliar de controle, e assim, de acordo com a diferença de potencial fornecida pelo conversor D/A, foi possível ligar uma das saídas do Driver LM3914.

A Tabela 8 apresenta os valores de diferença de potencial fornecida pelo conversor D/A para ligar cada uma das saídas do circuito auxiliar de controle.

Tabela 8 – Relação Diferença de potencial X Ativação de Saídas do circuito auxiliar de controle

Saída	Diferença de Potencial
Saídas Desligadas	0,00 V
1ª Saída Ligada	0,39 V
2ª Saída Ligada	0,59 V
3ª Saída Ligada	0,80 V
4ª Saída Ligada	1,10 V
5ª Saída Ligada	1,29 V
6ª Saída Ligada	1,49 V
7ª Saída Ligada	1,80 V
8ª Saída Ligada	2,00 V
9ª Saída Ligada	2,20 V
10ª Saída Ligada	2,39 V

As saídas acionam Relés, que são dispositivos que podem ligar e desligar equipamentos, sensores, componentes elétricos e eletrônicos, etc.

A Figura 22 apresenta o esquema, e a Figura 23, uma foto do circuito auxiliar de controle:

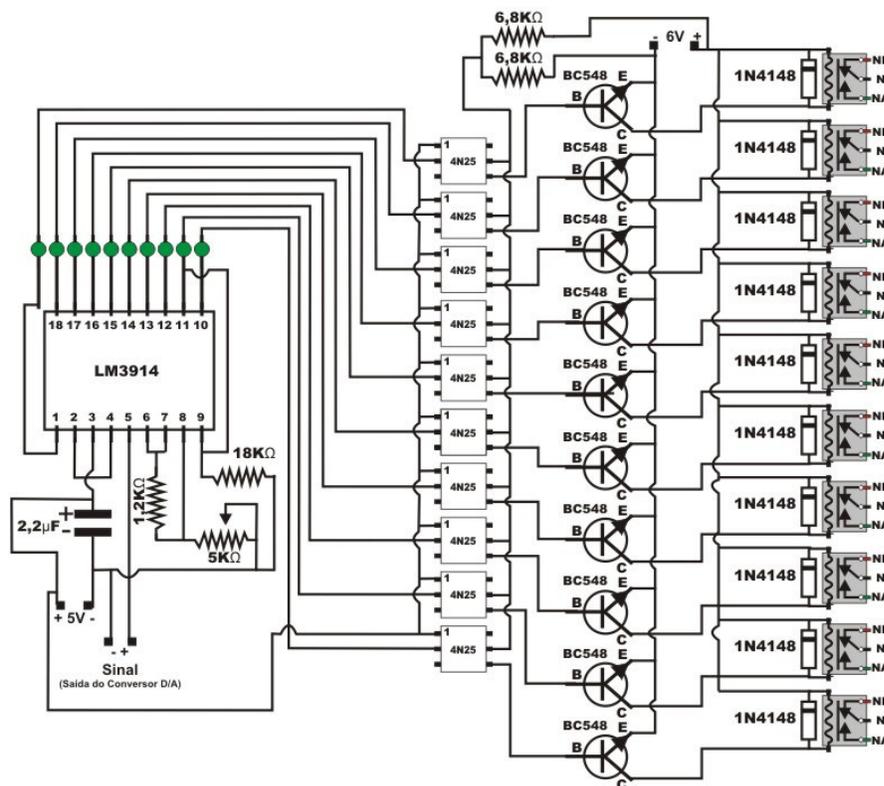


Figura 22 – Esquema de montagem do Circuito Auxiliar de Controle.

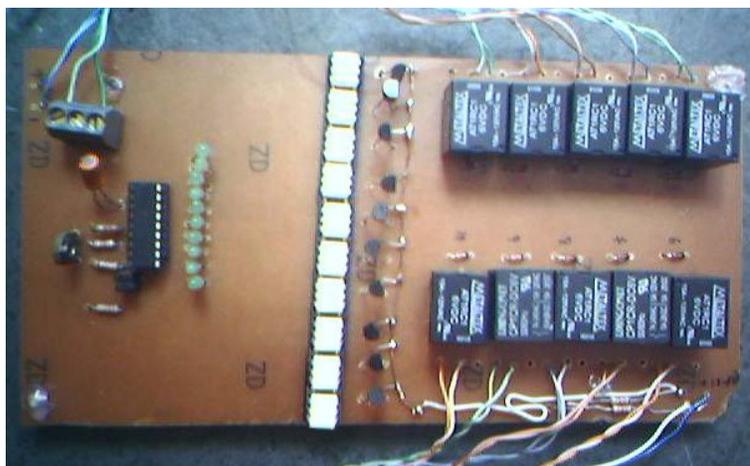


Figura 23 – Foto do Circuito Auxiliar de Controle.

A Tabela 9 apresenta os componentes utilizados no desenvolvimento do Circuito Auxiliar de Controle:

Tabela 9 – Componentes do Circuito Auxiliar de Controle

Componente	Quantidade	Valor/Tipo
Driver	1	LM3914
Resistor	1	1,2K Ω
Resistor	1	5K Ω
Resistor	1	18K Ω
Capacitor Eletrolítico	1	2,2 μ F
Led	10	Verde
Foto-acoplador	10	4N25
Relé	10	6 V / 220V-7A
Diodo	10	1N4148

4.4.2- Circuito de Interface: Porta paralela

Para permitir a comunicação entre dispositivos e o computador, através da porta paralela, foi desenvolvido um circuito de interface (CI-PP).

O circuito possui como principal componente eletrônico o conversor analógico-digital ADC0804. Este conversor A/D possui apenas um canal para coleta de dados, com conversão do valor analógico entre 0 e 5 volts para um valor digital de 8 bits. (National Semiconductor, 1999)

Outro componente importante é o multiplexador CD4051, que possui 8 saídas pelas quais pode ser efetuado o controle de experimentos e dispositivos. Através de três pinos (três bits) deste componente é possível controlar (ligar ou desligar) uma das 8 saídas do componente.

O circuito de interface é então, composto por um módulo de coleta de dados analógicos (tensão elétrica de 0 à 5 volts) e um módulo de controle, onde é possível ligar ou desligar 8 saídas.

A Tabela 10 apresenta os componentes utilizados no desenvolvimento do Circuito de Interface (Porta paralela):

Tabela 10 – Componentes do Circuito de Interface (Porta paralela)

Componente	Quantidade	Valor/Tipo
Conversor A/D	1	ADC0804
Multiplexador	1	CD4051
Buffer	1	SN74LS07N
Capacitor	1	104pF
Capacitor	1	150pF
Resistor	1	10 K Ω
Resistor	10	2,2 K Ω
Transistor NPN	10	BC548
Relé	10	6 V / 220V-7A
Diodo	10	1N4148

É apresentado na Figura 24 o esquema eletrônico, e na Figura 25 uma foto, do circuito de interface desenvolvido.

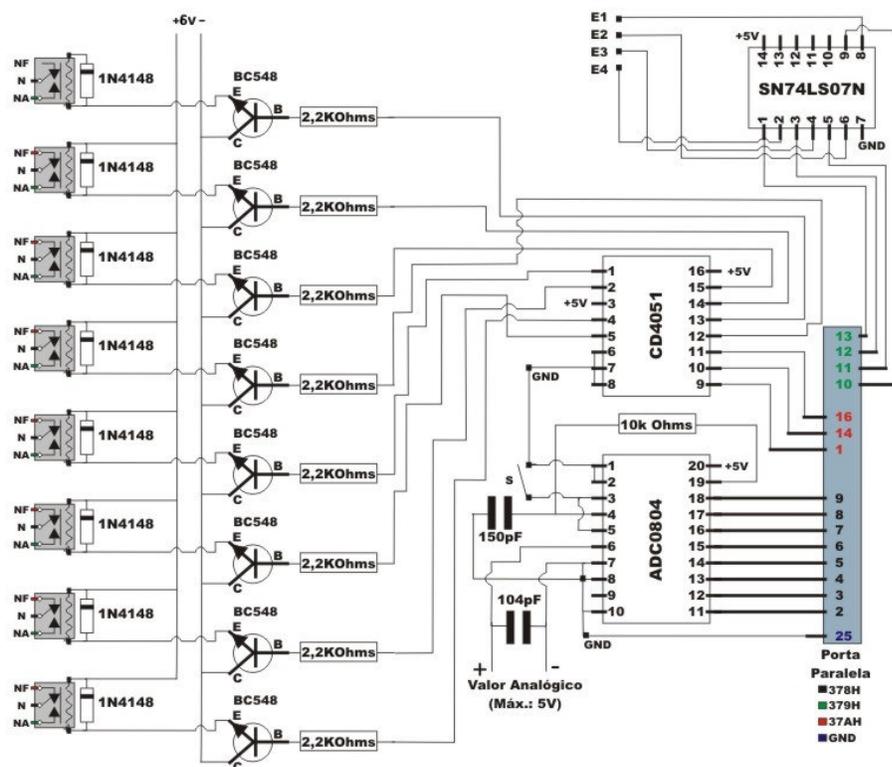


Figura 24 – Esquema de montagem do Circuito de Interface (Porta paralela).

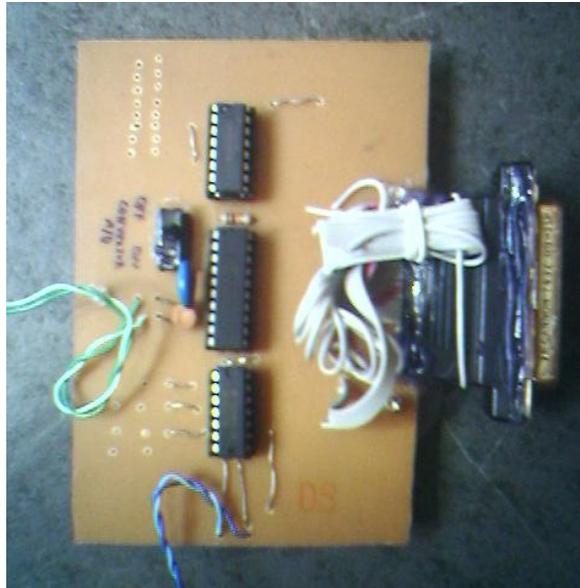


Figura 25 – Foto do Circuito de Interface (Porta paralela).

4.5- Quinta Etapa: Desenvolvimento de elementos para testes

Para que fosse testado o funcionamento do sistema de instrumentação virtual desenvolvido foi necessária a utilização de alguns elementos.

Tais elementos foram desenvolvidos em uma quinta etapa da pesquisa, e são classificados em dois tipos: Dispositivos de Controle e Experimentos.

Os dispositivos de controle foram construídos para que fossem realizados testes da função de controle a distância do sistema. Já os experimentos foram desenvolvidos para que fosse testada, além da função de controle a distância, a coleta e envio automático de dados.

Foram construídos dois dispositivos de controle e três experimentos didáticos de Física.

4.5.1- Dispositivo de Controle 1: Controle de Cor

Um primeiro dispositivo de controle criado foi um emissor de luz visível em oito cores diferentes.

As cores são obtidas através da combinação de três leds de alto brilho. Os três leds possuem cores diferentes, sendo um de cor vermelha, um segundo de cor amarela e o último de cor azul.

Para permitir que a luz fosse emitida de forma uniforme, com uma mistura ideal das cores, os leds tiveram a parte superior lixada, e foram colocados dentro de uma esfera branca e opaca.

Esta esfera foi colocada no interior de uma caixa, com apenas uma parte da esfera visível.

A Figura 26 apresenta uma foto deste dispositivo:



Figura 26 – Foto do dispositivo 1.

A combinação do acionamento dos três leds para a obtenção de cada cor emitida pelo dispositivo é apresentada na Tabela 11:

Tabela 11 – Cores emitidas para cada combinação dos leds

Cor	Led Vermelho	Led Amarelo	Led Azul
Branco (Opaco)	Desligado	Desligado	Desligado
Azul	Desligado	Desligado	Ligado
Amarela	Desligado	Ligado	Desligado
Azul Claro	Desligado	Ligado	Ligado
Vermelha	Ligado	Desligado	Desligado
Violeta	Ligado	Desligado	Ligado
Laranja	Ligado	Ligado	Desligado
Violeta Claro	Ligado	Ligado	Ligado

Um circuito eletrônico (C-D1) foi desenvolvido para auxiliar no acionamento dos leds. Ao circuito C-D1 são conectadas sete saídas do circuito de interface (Placa R-Control e Circuito auxiliar de controle ou circuito de interface para porta paralela), sendo cada saída responsável por definir uma cor emitida.

O circuito eletrônico C-D1 possui uma conexão comum (CM), que é conectada aos pinos “N” de sete relés presentes em sete saídas de controle do circuito de interface. Outros sete pontos de conexão (A1, A2, A2, A4, A5, A6 e A7) devem ser conectadas aos pinos “NA” das sete saídas utilizadas, do circuito de interface.

Um esquema deste circuito é apresentado na Figura 27, a seguir:

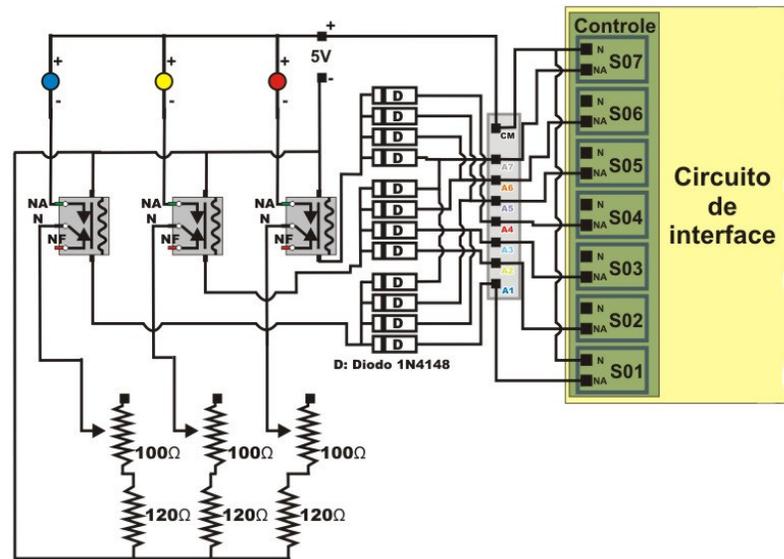


Figura 27 – Esquema do circuito do dispositivo 1.

A Tabela 12 apresenta os componentes utilizados no desenvolvimento do circuito C-D1:

Tabela 12 – Componentes do circuito C-D1

Componente	Quantidade	Valor/Tipo
Led de alto brilho	1	Vermelho
Led de alto brilho	1	Amarelo
Led de alto brilho	1	Azul
Trimpot	3	100 Ω
Resistor	3	120 Ω
Relé	3	6 V / 220V-7A
Diodo	12	1N4148

4.5.2- Dispositivo de Controle 2: Protótipo robótico

Um segundo dispositivo desenvolvido para que fossem realizados testes de controle, foi um protótipo robótico. Este protótipo robótico tem o seu movimento controlado.

Foram utilizados dois motores para permitir a movimentação do robô, um para o movimento frontal e traseiro, e o segundo para o movimento lateral. Estes motores são conectados a sistemas mecânicos, compostos por engrenagens, as quais permitem que o robô faça movimentos lentos, evitando uma velocidade excessiva.

Os motores são conectados a um circuito rádio controlado (C-D2C). Trata-se de um circuito receptor de radiofrequência (RF). Um segundo circuito (C-D2B) é o responsável pelo envio de sinais de controle, é um circuito emissor RF. Ambos

foram adquiridos especificamente para o uso neste dispositivo, e operam na faixa de frequência de 27MHz.

Ao circuito emissor RF (C-D2B) é conectado um circuito auxiliar (C-D2A), que por sua vez se conecta a seis das saídas do circuito de interface utilizado. O acionamento de cada uma destas saídas definirá seis movimentos possíveis do robô. O circuito emissor RF, e os movimentos possíveis do protótipo robótico são representados pela Figura 28:

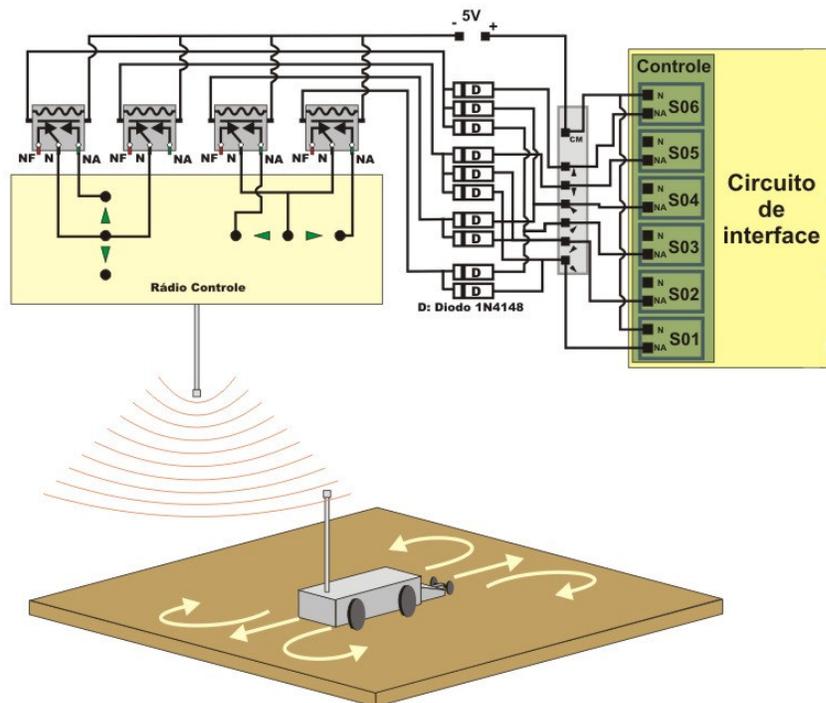


Figura 28 – Emissor de RF e Movimentação possível do protótipo robótico.

Além dos movimentos de deslocamento para frente ou para trás, é possível efetuar o movimento de mudança de direção, ou seja, efetuar curvas, a partir da combinação do acionamento dos dois motores do protótipo robótico.

Uma foto deste dispositivo é apresentada a seguir:



Figura 29 – Foto do protótipo robótico.

4.5.3- Experimento 1: Absorção e reflexão da luz por materiais claros e escuros

Materiais claros e escuros possuem comportamentos diferentes quando submetidos a radiações luminosas. Materiais escuros absorvem uma quantidade maior desta radiação, re-emitindo na forma de radiação infravermelha (energia térmica). Já materiais claros possuem uma absorção menor, e uma taxa maior de reflexão da luz incidente, se comparados aos materiais escuros.

Foi construído um experimento com o objetivo de demonstrar a diferença da absorção e reflexão da luz, entre corpos claros e escuros.

Para isto, em uma caixa foram colocadas duas placas metálicas idênticas, mas pintadas por tintas de cores diferentes, uma pintada de preto e outra de branco.

Uma lâmpada fluorescente de 9 Watts, de baixa radiação térmica, emite luz branca sobre estas placas.

Atrás das placas foram colocados sensores de verificação de temperatura (termistores), possibilitando verificar a absorção da energia luminosa, que provoca o aumento da temperatura da placa.

À frente de cada placa foram colocadas ainda sensores de luminosidade LDR (Light Dependent Resistor), com o objetivo de verificar a intensidade de radiação luminosa refletida por cada placa.

Estes sensores são resistências variáveis em função das grandezas físicas que verificam (luminosidade e temperatura).

As Figuras 30 e 31 apresentam, respectivamente, o esquema de montagem e uma foto deste experimento:

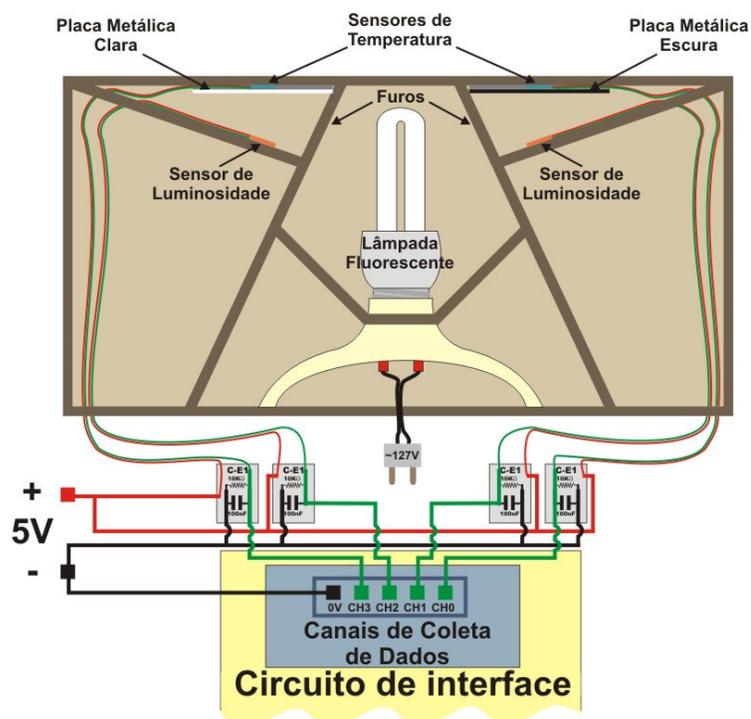


Figura 30 – Esquema de montagem do Experimento 1.



Figura 31 – Foto do Experimento 1.

Para que seja efetuada a medida das grandezas físicas, os sensores utilizados neste experimento são conectados ao circuito de interface por meio de circuitos auxiliares (C-E1).

Estes circuitos auxiliares são simples, compostos apenas por um resistor e um capacitor, além de uma fonte de alimentação. São necessários para fornecer corrente elétrica aos sensores e facilitar a linearização dos dados coletados. Um esquema deste circuito é apresentado na Figura 32:

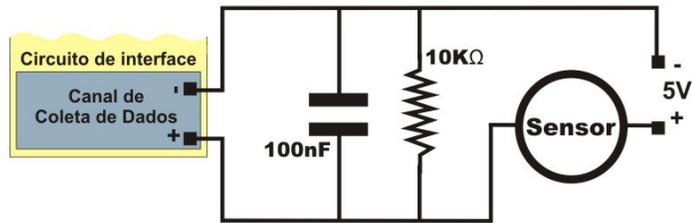


Figura 32 – Circuito auxiliar para sensores do Experimento 1.

Por se tratar de uma medida de quatro grandezas (canais), e por somente a placa R-Control30 possuir quatro canais de coleta de dados, a comunicação com este experimento é realizado somente através da porta USB.

O circuito de interface para porta paralela, conforme já comentado, foi projetado para efetuar a leitura de apenas um canal de coleta, porém, é possível o desenvolvimento de outros circuitos de interface através da porta paralela, que dêem suporte a uma quantidade maior de canais de coleta de dados analógicos.

Após a montagem do experimento, os sensores de temperatura foram calibrados. Para isto, foi tomado como referência os dados técnicos fornecidos pelo fabricante (Maxim, 2001).

Foram utilizados dois termistores NTC (Negative Temperature Coefficient). Estes sensores possuem seu valor de resistência inversamente proporcional à temperatura.

Esta relação obedece a uma expressão, que é apresentada a seguir:

$$R_T = R_{25^{\circ}C} \cdot \exp \left[B \cdot \left(\frac{1}{T + 273} - \frac{1}{298} \right) \right] \quad \text{eq.1}$$

R_T é a resistência elétrica do Termistor; T é a temperatura em $^{\circ}C$ que está sendo verificada pelo termistor. Já $R_{25^{\circ}C}$ (resistência do termistor à $25^{\circ}C$) e B são valores informados pelo fabricante. Os termistores utilizados possuem os valores 10000 para $R_{25^{\circ}C}$ e 3965 para B .

A Figura 33 apresenta um gráfico da relação $R_T \times T$:

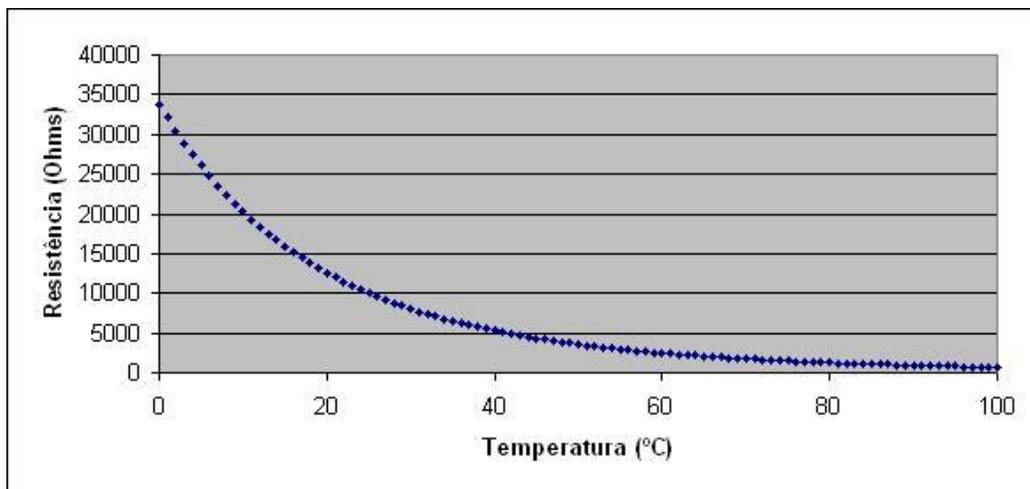


Figura 33 – Gráfico $R_T \times T$ para os termistores utilizados.

Conforme pode ser observado, o comportamento deste sensor não é linear, tendo para o intervalo entre 0°C e 100°C um coeficiente de correlação R^2 igual a 0,7387. Desta forma, é necessária que seja utilizada alguma alternativa que permita a linearização dos termistores.

O circuito auxiliar C-E1 possui também este objetivo. O circuito, conforme pôde ser observado na Figura 32, trata-se de uma associação em série entre o sensor, neste caso o termistor, e um resistor de 10KΩ. É efetuada a leitura da tensão elétrica atuante no resistor.

Partindo da Lei de Ohm, é possível deduzir que para este circuito, temos as seguintes equações:

$$U = R.I \quad \text{eq.2}$$

$$U = (R_T + R_{10k\Omega}).I \quad \text{eq.3}$$

Sabendo-se que o circuito é alimentado por uma tensão elétrica de 5 Volts, tem-se então:

$$5 = (R_T + R_{10k\Omega}).I \quad \text{eq.4}$$

$$I = \frac{5}{10000 + R_T} \quad \text{eq.5}$$

Em posse da equação que define a corrente elétrica do circuito, que por ser uma associação em série é a mesma corrente que passa pelo resistor, é possível

chegar até a equação que descreverá a tensão elétrica atuando neste resistor, em função da resistência elétrica do termistor:

$$U_{R_{10K\Omega}} = 10000 \cdot I \quad \text{eq.6}$$

$$U_{R_{10K\Omega}} = 10000 \cdot \frac{5}{10000 + R_T} \quad \text{eq.7}$$

Substituindo o valor de R_T , descrito por eq.1, em eq.7, é possível traçar o seguinte gráfico para a relação entre $U_{R_{10K\Omega}}$ e T :

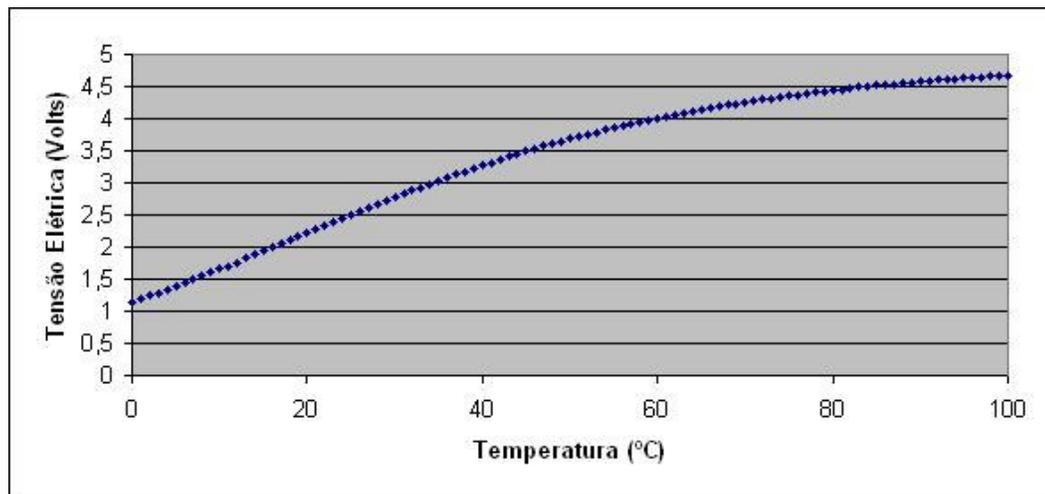


Figura 34 – Gráfico $U_{R_{10K\Omega}}$ x T para os termistores utilizados.

Durante a coleta de dados, a grandeza física medida é a tensão elétrica atuante no resistor; e conforme é possível observar na Figura 34, o gráfico mostra-se mais linear se comparado ao apresentado na Figura 33. O coeficiente de correlação para este gráfico (Figura 34) é R^2 igual a 0,9479.

É possível observar ainda, através da Figura 34, que para o intervalo entre 0°C e 40°C o gráfico possui um comportamento bem próximo de uma linha reta. A Figura 35, que se segue, apresenta um gráfico somente desta região, efetuando uma comparação com uma linha reta.

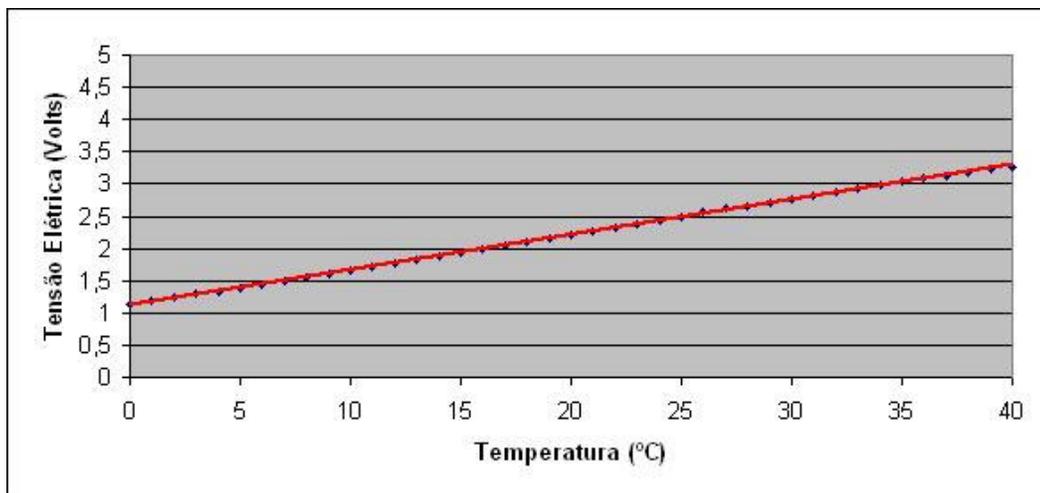


Figura 35 – Gráfico $U_{R10K\Omega} \times T$ (entre 0°C e 40°C), para os termistores utilizados.

Para este intervalo, o coeficiente de correlação se aproxima muito da linha reta, possuindo R^2 igual a 0,9996.

Para este caso, devido a aproximação a um comportamento linear, a tensão elétrica obtida no resistor, em função da temperatura verificada no termistor, pode ser descrita pela seguinte equação:

$$U_{R10K\Omega} = 0,0546.T + 1,1207 \quad \text{eq.8}$$

Para o experimento proposto, o intervalo entre 0°C e 40°C é satisfatório, podendo-se tomar como referência a equação eq.8.

Baseado nesta equação, os valores de T para tensões elétricas iguais a 0V e 5V, os quais devem ser informados ao software servidor de controle, são, respectivamente, -20,5256°C e 71,0495°C. Conforme será descrito à frente, o software relaciona a tensão elétrica medida à grandeza física sendo analisada, através de um cálculo de interpolação, onde devem ser fornecidos os valores da grandeza física que correspondem as tensões elétricas limites, 0V e 5V.

Já em relação aos sensores LDR, devido a algumas características da fonte de luz utilizada e dos próprios sensores, é feita a medição de um valor relativo, em relação ao máximo de intensidade de luz que o LDR consegue verificar.

De acordo com as informações do fabricante (Silonex, 2007a; 2007b), para o tipo de sensor LDR utilizado (CdS - sulfeto de cádmio – 550nm), o fenômeno fotoelétrico ocorre com maior facilidade quando se incide sobre o sensor uma

radiação eletromagnética na faixa de 550nm. A Figura 36, que se segue, apresenta um gráfico da resposta relativa do tipo de sensor LDR utilizado, em relação ao comprimento de onda da radiação ao qual o sensor é exposto.

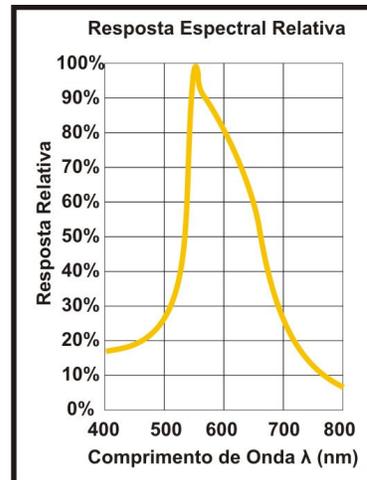


Figura 36 – Resposta relativa do LDR.

O gráfico se refere a um mínimo de intensidade da radiação de comprimento de onda de 550nm, necessária para que a resposta relativa do sensor seja o máximo, ou seja, 100%. No gráfico, é apresentada ainda, a resposta relativa que o sensor apresenta, para radiações em outras faixas de comprimento de onda, que possuem a mesma intensidade da radiação de 550nm.

Quando o sensor apresenta uma resposta relativa igual a 100%, sua resistência elétrica está bem próxima de zero, e logo, uma resposta relativa igual a 0% indica que o sensor apresenta uma resistência elétrica alta.

Cabe ressaltar que, é possível, com radiações de comprimento de onda diferentes de 550nm, alcançar uma resposta próxima ou igual a 100%, mas para isto, será necessária uma intensidade maior da radiação.

Esta intensidade diz respeito a densidade de energia luminosa que incide sobre o sensor, ou seja, energia/área do sensor.

Como a fonte luminosa utilizada no experimento emite radiação luminosa em diferentes comprimentos de onda, torna-se difícil a coleta de informações com estes sensores, a não ser a resposta relativa do sensor.

Cada sensor LDR é conectado a um circuito auxiliar C-E1, e a tensão elétrica medida é proporcional à resposta relativa do sensor. Sendo assim, para que o software servidor de controle apresente os valores medidos em “resposta relativa”,

devem ser fornecidos os valores que se referem aos limites de medição, que serão assim, 0% para o mínimo 0V, e 100% para o máximo 5V.

4.5.4- Experimento 2: Aplicação da Lei de Ampère

Foi desenvolvido um experimento simples que busca demonstrar o fenômeno de obtenção de um campo magnético em torno de um condutor, a partir de uma corrente elétrica que percorre este condutor. Tal fenômeno é descrito pela lei de Ampère.

Um condutor foi enrolado, com uma grande quantidade de voltas, em uma barra na forma de “U”, formando uma bobina.

À frente desta barra metálica foi colocado um interruptor magnético, denominado “Reed Switch” que é utilizado para a identificação da presença de campo magnético.

Tanto a bobina como o interruptor magnético foram fixados em uma mesma base de madeira.

Um esquema de montagem deste experimento, assim como uma foto, são apresentados respectivamente, nas Figuras 37 e 38 a seguir:

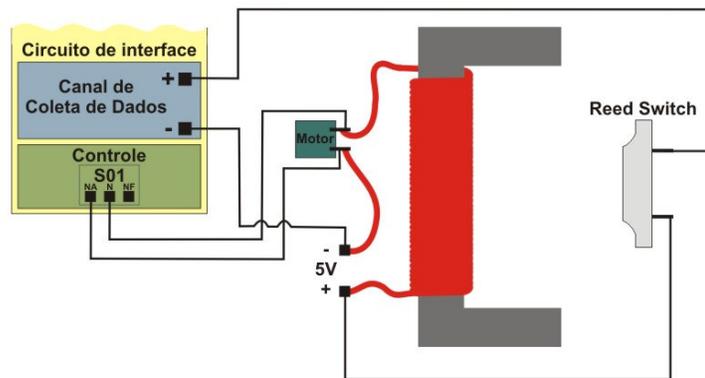


Figura 37 – Esquema de montagem do Experimento 2.



Figura 38 – Foto do Experimento 2.

O interruptor magnético, quando acionado, fornece a placa de aquisição de dados uma tensão elétrica igual a 5 V, e quando desligado, fornece uma tensão elétrica igual a 0V, permitindo assim a identificação da geração ou não de um campo magnético.

4.5.5- Experimento 3: Indução Eletromagnética de Faraday

Um terceiro experimento desenvolvido apresenta o fenômeno em que se obtém uma corrente elétrica em um condutor, quando este é exposto a um campo magnético variável. É o fenômeno conhecido como indução de Faraday.

Foi montada uma hélice com um conjunto de ímãs, e esta foi presa ao eixo de um motor. Quando o motor é ligado, o movimento do conjunto de ímãs irá ocasionar um campo magnético variável. Nas proximidades desta “hélice magnética” é colocada uma bobina, composta por um fio enrolado com um grande número de voltas em uma barra em forma de “U”.

Desta forma, quando o motor é ligado, é gerada uma corrente elétrica no condutor, e assim uma diferença de potencial elétrico será verificado nas extremidades deste condutor. Um diodo é utilizado para retificar a corrente elétrica gerada.

Um esquema de montagem e uma foto deste experimento são apresentados, respectivamente, nas Figuras 39 e 40, a seguir:

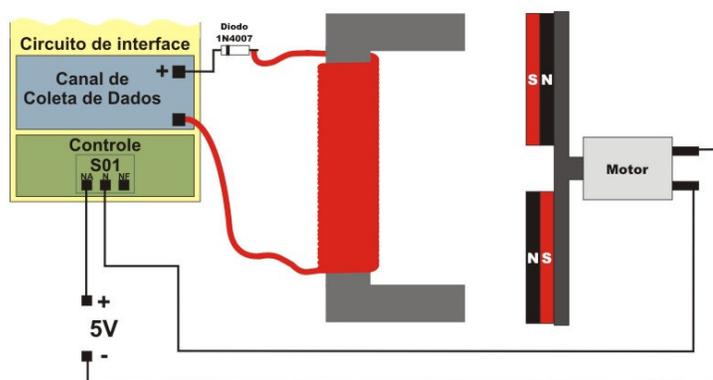


Figura 39 – Esquema de montagem do Experimento 3.

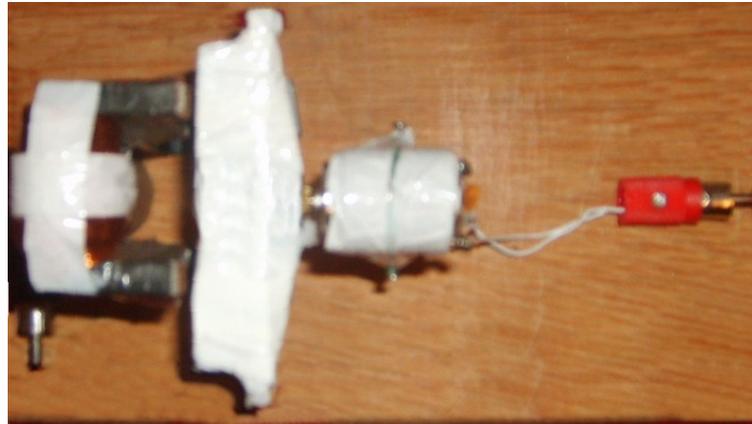


Figura 40 – Foto do Experimento 3.

Este experimento fornece à placa de aquisição de dados uma tensão elétrica. Como o circuito de interface coleta dados através de valores analógicos de tensão elétrica, não é necessário cálculos de calibração para este experimento. Logo, deve ser informado ao software servidor de controle os valores limites de 0V e 5V.

4.6- Sexta Etapa: Testes de funcionamento do sistema desenvolvido

Em uma sexta etapa da pesquisa foram realizados testes com o intuito de efetuar uma avaliação do funcionamento do Sistema de Instrumentação Virtual desenvolvido.

Foram dois os tipos de testes: um em que se buscou verificar alguns aspectos relacionados a transmissão de dados durante a comunicação dos softwares; e outros testes onde foram verificados o uso do sistema desenvolvido para a realização de procedimentos experimentais e de controle.

4.6.1- Testes de transmissão de dados

Nestes testes foram verificados os tempos de transferência de mensagens entre o software cliente e software servidor, e o tempo de atraso no recebimento de imagens de vídeo.

Para que fosse possível a medição do tempo entre o envio e recebimento de mensagens, um software do tipo cliente foi modificado, gerando assim um novo tipo de software que foi denominado como “Software Cliente-Timer”.

Visualmente, este software é idêntico ao software Cliente. A modificação consistiu na adição de mais uma função no software. Para isto, foi adicionado um

componente Timer, no qual foi implementado um contador de tempo, que registrava o intervalo entre o envio de uma mensagem ao software servidor, e o recebimento desta mensagem de volta.

Conforme já descrito, qualquer mensagem enviada por um software cliente é direcionada ao servidor, que por sua vez, redireciona para todos os clientes conectados, inclusive para aquele que enviou a mensagem.

Sendo assim, o componente timer registra o momento em que o usuário envia a mensagem e o momento em que o usuário recebe de volta esta mesma mensagem. O intervalo de tempo entre estes momentos é armazenado em um arquivo, no mesmo diretório em que se encontra o executável do software.

Este software foi utilizado no lugar do software cliente, por apenas um dos usuários do sistema, durante os testes com experimentos, os quais serão descritos no tópico 4.6.2.

Já o registro do tempo de atraso no recebimento de imagens de vídeo; apresentou-se como um desafio durante a pesquisa.

É difícil precisar um tempo de atraso das imagens, devido quantidade em bytes de dados envolvidos em uma transferência de vídeo. Isto causa uma grande variação no tempo de transmissão dos dados, em função da capacidade do link com a Internet do computador servidor e do computador cliente, da quantidade de clientes conectados, etc.

A transmissão de eventos ao vivo por vídeo, através da internet, é ainda um campo tecnológico que precisa de considerável avanço.

Apesar disto, foi realizado um teste de tempo de transferência de vídeo, com o intuito de se ter um referencial a mais para efetuar a análise da aplicabilidade da ferramenta desenvolvida.

Para este teste foi desenvolvido um software, também através do IDE Delphi, o qual possui um cronômetro virtual, e o acesso às imagens transmitidas pelo servidor. Quando o cronômetro virtual do software está em funcionamento, é feita uma captura da tela do software a cada trinta segundos, e salva em um arquivo de imagem.

Durante os testes, o cronômetro virtual do software foi disparado no mesmo momento que um cronômetro real, que tem seu visor filmado pela câmera de captura de imagens.

Desta forma, nos arquivos que são criados periodicamente, haverá informações do tempo dos dois cronômetros. A diferença de tempo entre estes dois valores de tempo é o atraso no recebimento das imagens.

A Figura 41 apresenta um esquema de como isto ocorre:

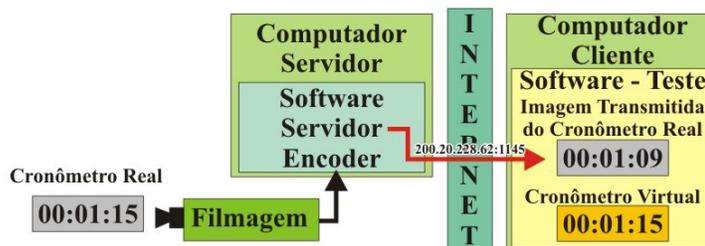


Figura 41 – Esquema representando o teste de transmissão de imagens.

A seguir, através da Figura 42, é apresentada uma tela capturada deste software, durante sua execução:



Figura 42 – Tela capturada do software para o teste de transmissão de imagem.

Este teste foi realizado durante 1 hora, com a coleta do tempo de atraso ocorrendo a cada 30 segundos, e os resultados são apresentados no tópico 5.2.1.

4.6.2- Testes com experimentos e dispositivos de controle

Estes testes consistiram em cinco sessões com o uso da placa de aquisição e controle de dados através da porta USB e quatro sessões com o uso da placa de aquisição e controle de dados através da porta paralela. Durante estas sessões, o sistema esteve em operação durante uma hora ininterrupta para procedimentos com

dispositivos de controle; e cinco horas ininterruptas para procedimentos com experimentos.

Exceto para o Experimento 1, foram duas sessões de testes para cada um dos experimentos ou dispositivos de controle que foram criados, sendo uma sessão com o uso de cada placa de aquisição.

Durantes a realização destes testes, buscou-se observar possíveis falhas de funcionamento do sistema, verificar características dos softwares no que diz respeito a qualidade destes, e a viabilidade de uso como ferramenta pedagógica.

São apresentados, através da Tabela 13, dados gerais dos testes realizados:

Tabela 13 – Dados gerais da realização dos testes

Elementos	Tempo	Intervalo no envio de dados coletados	Intervalo no envio de comandos
Dispositivos de Controle	1 hora	5 segundos	4 minutos
Experimentos	5 horas	20 segundos	20 minutos

A Tabela 14, a seguir, apresenta os tipos de dados coletados e de controle, para cada um dos elementos:

Tabela 14 – Tipo de dados coletados e de controle

Experimento/Dispositivo de Controle	Tipo de dados coletados	Tipo de dados de controle
Dispositivo de Controle 1	Informação sobre qual cor está sendo emitida	Controle da cor que deve ser emitida.
Dispositivo de Controle 2	Informação sobre qual o tipo de movimento está sendo realizado	Controle de qual movimento deve ser realizado
Experimento 1	Medidas de luminosidade refletida e temperatura de duas placas metálicas	Ligar ou desligar fonte de luz (lâmpada fluorescente)
Experimento 2	Identificação de presença de campo magnético com intensidade mínima para ativação do sensor	Ligar ou desligar a passagem de corrente elétrica em condutor
Experimento 3	Medida de diferença de potencial elétrico entre as extremidades de um fio condutor	Ligar ou desligar motor que permite um campo magnético variável

5 - RESULTADOS

Os resultados obtidos com o desenvolvimento da pesquisa são apresentados neste capítulo. São apresentados resultados específicos quanto ao desenvolvimento de cada software, quanto ao desenvolvimento dos experimentos e dispositivos de controle, além dos resultados dos testes de funcionamento do sistema.

5.1- Softwares desenvolvidos

Neste tópico são apresentados os resultados do desenvolvimento gráfico e o acesso às funções dos softwares.

5.1.1- Software Cliente

Uma tela capturada do Software Cliente é apresentada na Figura 43, que se segue:



Figura 43 – Tela capturada do Software Cliente.

O software possui diversas funções, distribuídas conforme pode ser observado na Figura 44:

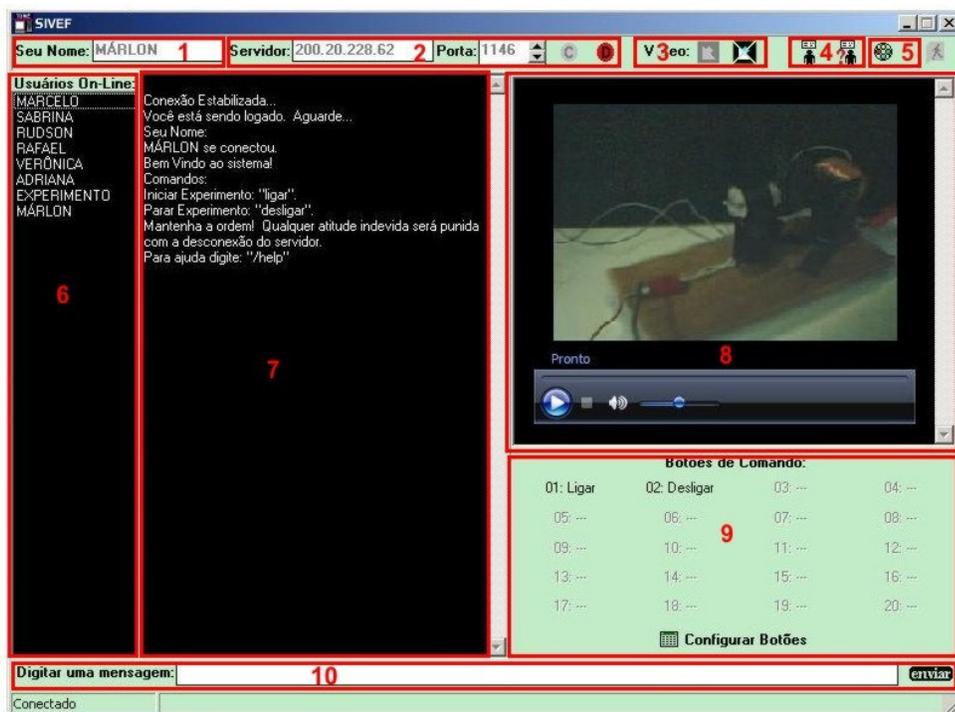


Figura 44 – Regiões do Software Cliente.

Na barra superior o usuário pode configurar o seu nome de conexão (1), o endereço e porta de comunicação do servidor (2). Existem ainda botões para a conexão e desconexão do sistema (2), para a Habilitação ou não da visualização da imagem dos experimentos ou dispositivos de controle (3), além de botões para o acesso a conteúdos teóricos e de acesso a seção de ajuda do software (4), e de carregamento de esquema de cores do software (5).

Abaixo da barra superior, à esquerda, é apresentada a lista de usuários conectados ao sistema (6), e dados enviados pelo servidor e por outros usuários do sistema (7). Estes dados podem ser, por exemplo, os valores medidos nos experimentos, ou mesmo mensagens enviadas por outros usuários (chat).

À direita são apresentadas imagens dos experimentos e dispositivos de controle que estejam sendo utilizados (8). Abaixo destas imagens, encontram-se os botões de comando (9), que podem ser configurados para facilitar o controle de dispositivos e experimentos.

O Envio de Mensagens e de Comandos no formato de texto é feito através de uma caixa de texto e um botão (10), que se encontra na parte inferior do software.

As Figuras 45, 46, 47 e 48 apresentam outras telas do Software Cliente: a seção de conteúdos teóricos, a seção de ajuda do software, carregamento de esquema de cores do software, e a configuração de botões de comando.

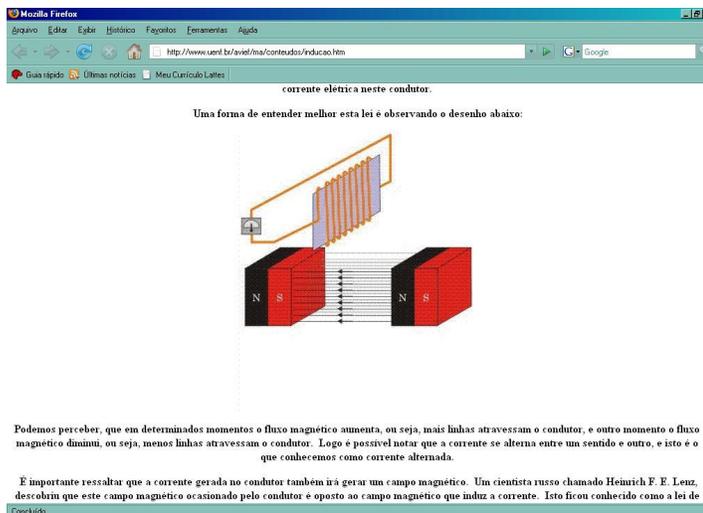


Figura 45 – Tela capturada da Seção de Conteúdos Teóricos, acessível através do Software Cliente.

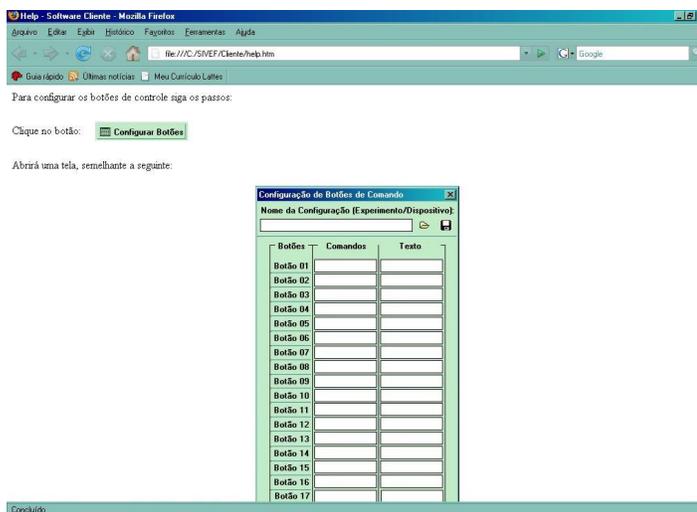


Figura 46 – Tela capturada da Seção de ajuda do Software Cliente.

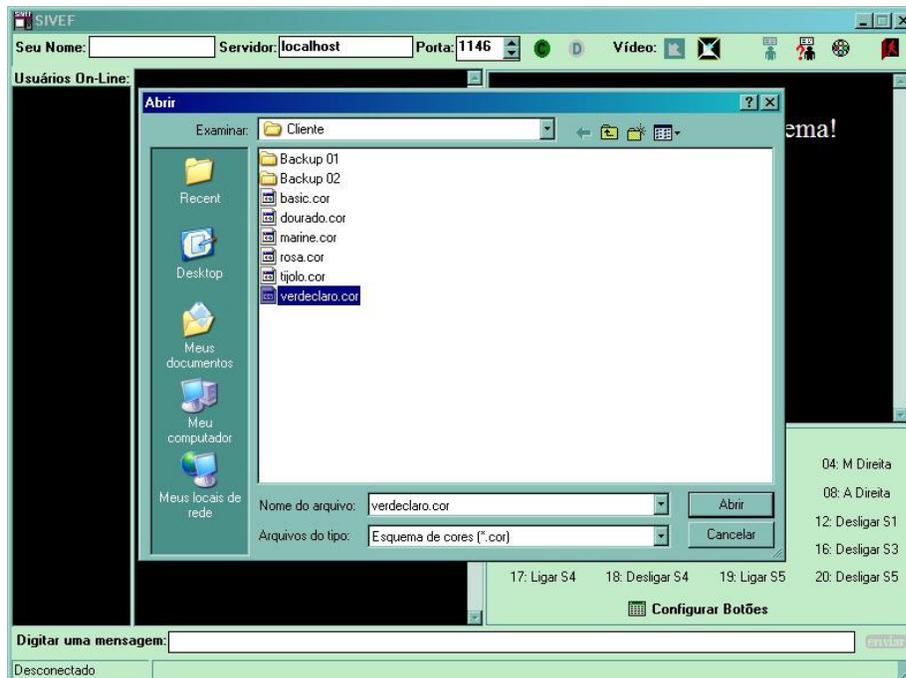


Figura 47 – Tela capturada do Software Cliente: Carregamento de esquema de cores.

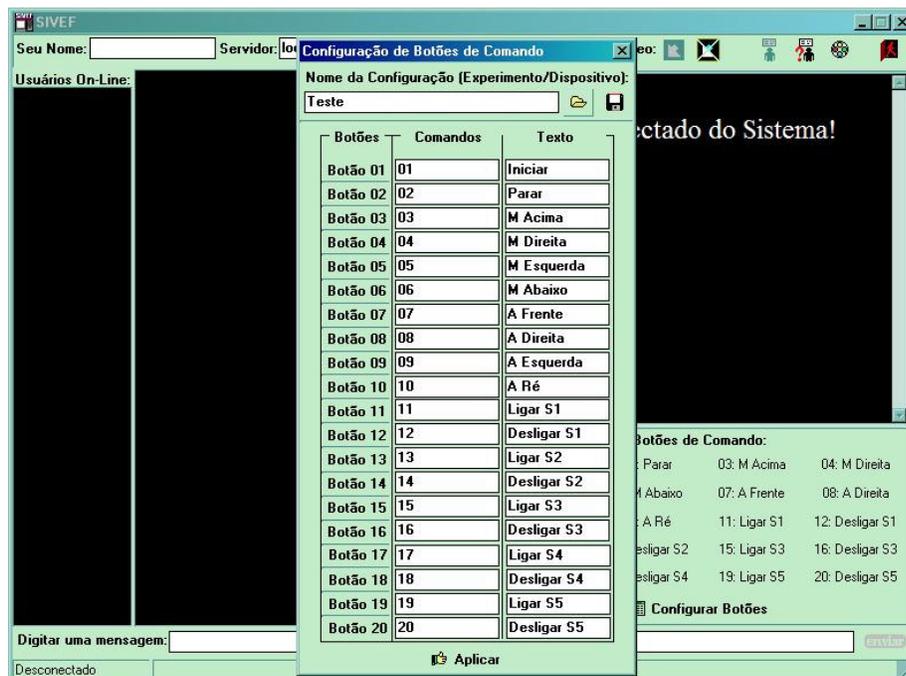


Figura 48 – Tela capturada do Software Cliente: Configuração de Comandos.

5.1.2- Software Servidor de Comunicação

A Figura 49 apresenta a tela capturada do Software Servidor de Comunicação:

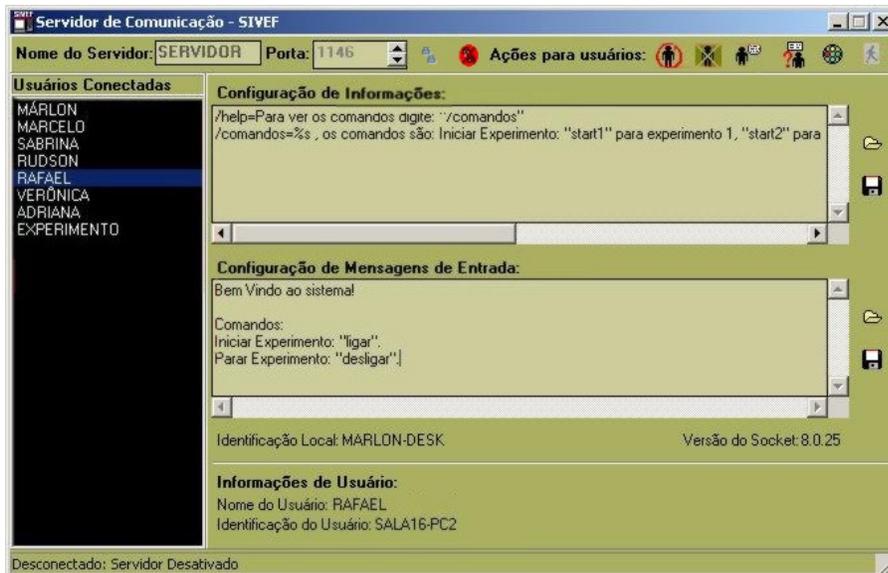


Figura 49 – Tela capturada do Software Servidor de Comunicação.

Assim como no software cliente, o acesso às funções ocorre através de componentes distribuídos em algumas regiões da tela do software, conforme é possível se observar na Figura 50:

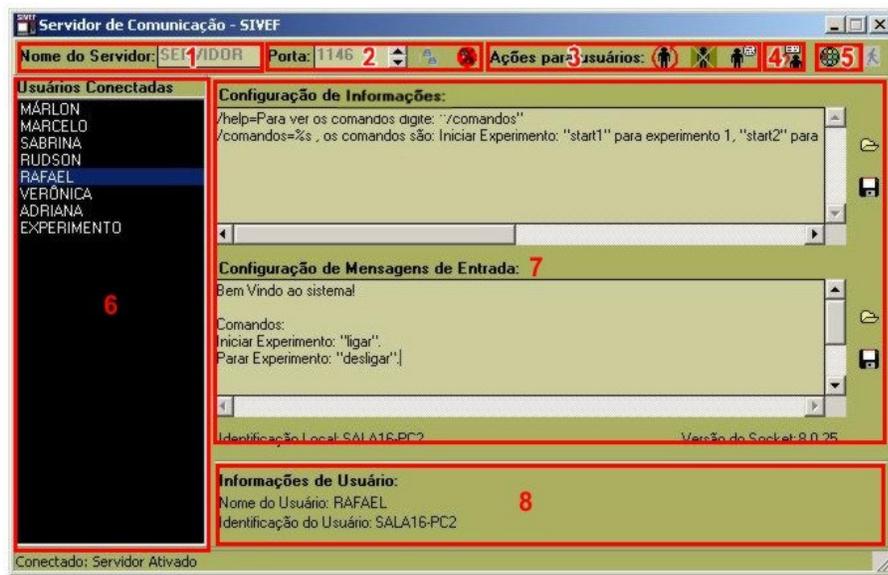


Figura 50 – Regiões do Software Servidor de Comunicação.

O software possui em uma barra superior, o acesso à configuração do nome do servidor (1) e da porta de conexão (2) que deverá ser utilizada pelos clientes, além de botões para iniciar e finalizar o serviço de comunicação (2). Nesta barra existem ainda, botões para o gerenciamento da conexão de usuários do sistema (3), como o botão de desconexão de um cliente e o botão para envio de mensagem privativa a um usuário do sistema.

É possível também, através de um botão na barra superior, o acesso a uma seção de ajuda do software (4). Outro botão permite ainda a escolha de esquemas de cores para o software (5). As telas da seção de ajuda do software e de carregamento de esquemas de cores são semelhantes as do software cliente, as quais são apresentadas, respectivamente, nas Figuras 46 e 47.

À esquerda da tela principal do software há uma lista dos usuários conectados ao sistema (6). Quando um dos nomes é selecionado, são apresentadas em uma região inferior da tela do software (8), informações de identificação do usuário como o IP de um usuário conectado ao servidor através da internet, ou o nome do computador, para um usuário conectado na rede local onde se encontra o servidor.

Existem ainda as opções de configuração de informações e de mensagens de entrada (7). Os comandos de informações são comandos que se enviados para o servidor, irão retornar alguma informação, como a listagem de comandos possíveis para o controle de experimentos, ou outras informações que sejam necessárias no procedimento experimental ou de controle realizado. Já as mensagens de entrada são aquelas que são enviadas ao usuário no momento em que este se conecta ao servidor.

5.1.3- Software Servidor de Controle – USB

É apresentado na Figura 51 a tela capturada do Software Servidor de Controle (USB):

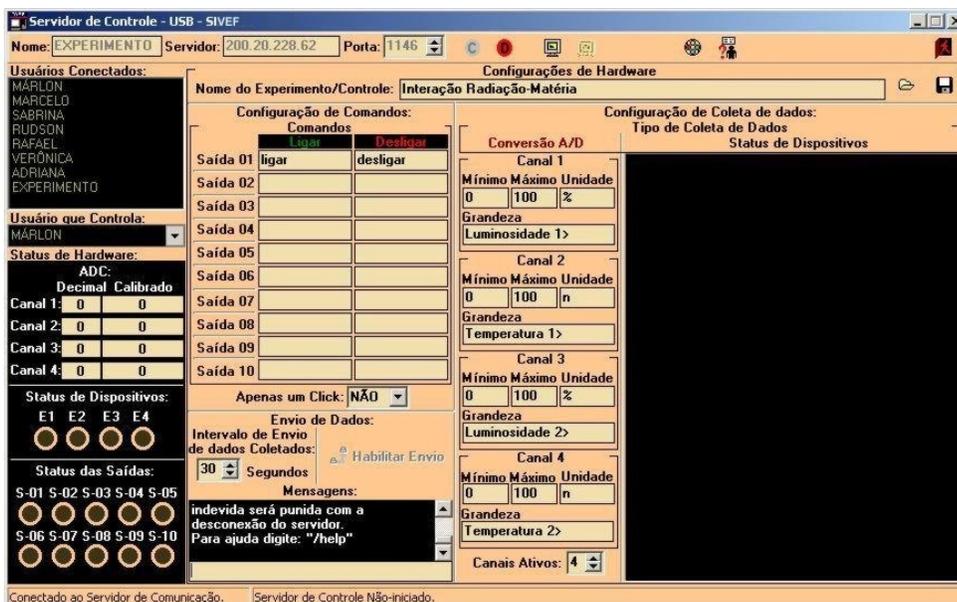


Figura 51 – Tela capturada do Software Servidor de Controle (USB).

A seguir, na Figura 52, são apresentadas as regiões do software, as quais possuem o acesso a algumas das funções implementadas:

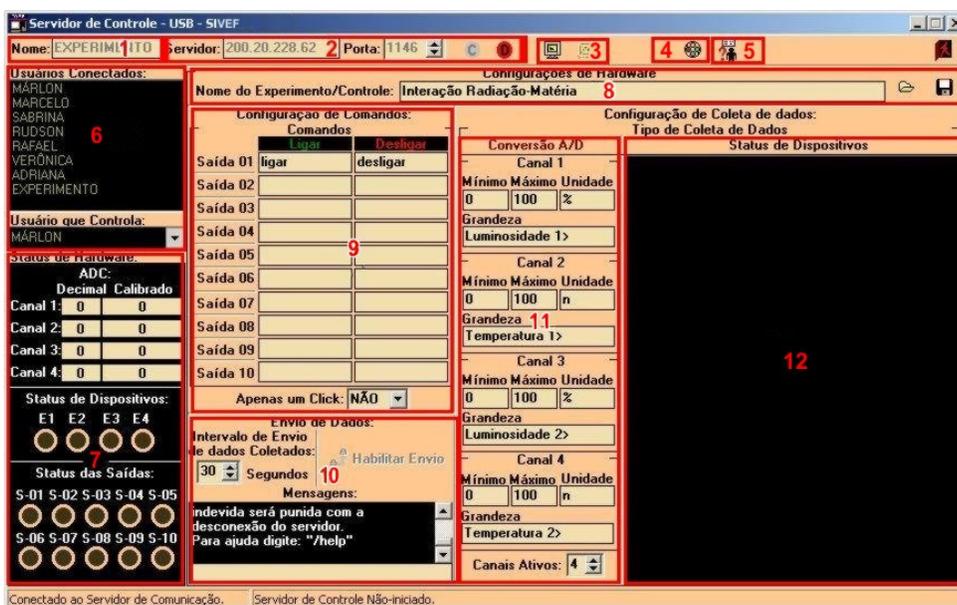


Figura 52 – Regiões do Software Servidor de Controle (USB).

Na barra superior o usuário informa através de caixas de texto (1) o nome de conexão, o endereço e porta de comunicação do servidor. Através do acionamento de um botão (2) o software se conecta ao sistema. Um segundo botão nesta mesma região (2) permite a desconexão do sistema.

Para iniciar ou finalizar o serviço de Servidor, existem dois botões (3) na barra superior do software. Também nesta barra superior se encontram os botões de acesso ao carregamento de esquemas de cores (4) e da seção de ajuda do software (5). Assim como nos demais softwares, as telas da seção de ajuda do software e de carregamento de esquemas de cores são semelhantes as do software cliente, que são apresentadas, respectivamente, nas Figuras 46 e 47.

À esquerda se encontra uma região (6) na qual é apresentada a lista de usuários conectados ao sistema, e uma caixa de seleção em que se pode escolher, entre os usuários conectados, aquele que estará habilitado ao controle do experimento ou dispositivo de controle.

Abaixo desta região, existe um painel (7) em que é apresentado o status do canal de coleta de dados e das saídas de controle.

Abaixo da barra superior do software existe uma região (8) onde é possível configurar o nome do procedimento experimental ou de controle, além de botões para salvar ou carregar as configurações de algum procedimento experimental ou de controle.

A configuração de comandos é realizada através de caixas de texto presentes numa região (9) central do software. Podem ser definidos comandos para ligar ou desligar uma das dez saídas do circuito auxiliar de controle. É possível ainda, com a opção “Apenas um Click” ativa, que as saídas ao serem ligadas se desliguem após frações de segundo.

Abaixo desta região central se encontram as opções (10) de configuração do tempo de intervalo de envio de dados coletados nos dispositivos, assim como um botão para a habilitação ou não deste envio.

A configuração de coleta de dados é feita de acordo com o tipo de dados. Na Figura 52 é possível observar duas regiões para a configuração de coleta de dados, uma primeira para dados analógicos (11) e outra para dados de estado dos dispositivos (12). Somente uma das duas regiões é ativada para cada procedimento experimental ou de controle.

Na região de dados analógicos (11) é possível efetuar a configuração de cada um dos quatro canais disponíveis, definindo valores relativos a grandeza física que será medida.

Na região de configuração de dados de estado (12), em destaque na Figura 53, é possível configurar dezesseis tipos de estado dos dispositivos, e mais

dezesseis ações automáticas vinculadas a estes estados. Conforme é possível se observar na Figura 53, existe uma tabela informando valores binários para quatro entradas de estado (E1, E2, E3 e E4). O algarismo “1” indica que a entrada de estado está ativa (potencial elétrico de 5 volts é fornecido a entrada), enquanto o algarismo “0” indica que a entrada não está ativa (potencial elétrico de 0 volts fornecido a entrada).



Figura 53 – Destaque da região de configuração de dados de estado, do Software Servidor de Controle (USB).

O software identifica o estado de cada uma das entradas, atribuindo a estas um valor digital (binário) de acordo com seu estado (ligada ou desligada). Desta forma, nesta região do software (12) é configurada uma mensagem de estado para cada tipo de combinação possível das entradas. Esta mensagem de estado é enviada aos usuários do sistema.

É possível ainda configurar uma ação automática, ligando ou desligando uma das saídas da placa auxiliar de controle, de acordo com a combinação de entradas ligadas e desligadas.

5.1.4- Software Servidor de Controle – Paralela

Uma tela capturada do Software Servidor de Controle (Paralela) é apresentada através da Figura 54:

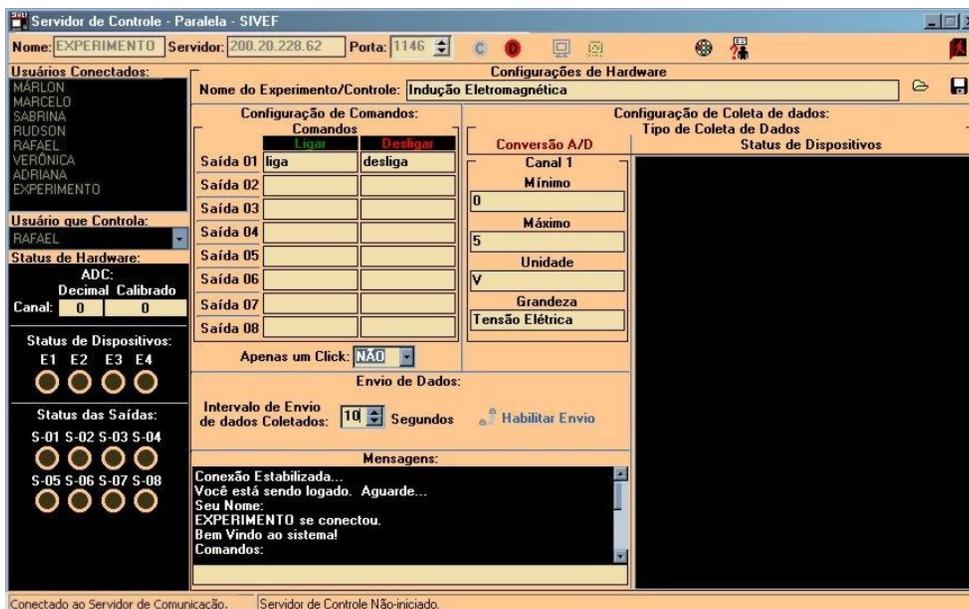


Figura 54 – Tela capturada do Software Servidor de Controle (Paralela).

As funções são distribuídas, assim como nos demais softwares desenvolvidos, em regiões da tela do software. A Figura 55 apresenta estas regiões:

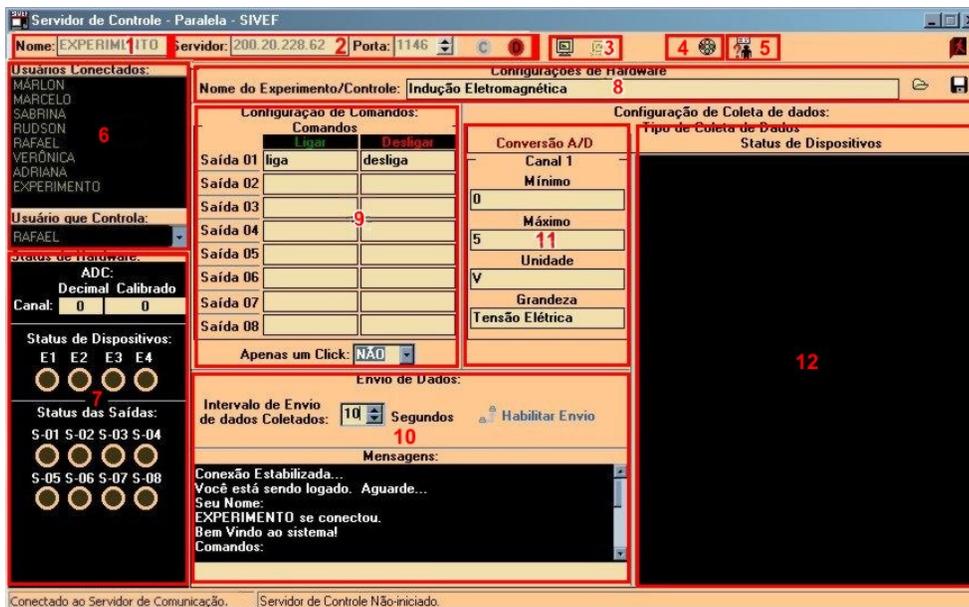


Figura 55 – Regiões do Software Servidor de Controle (Paralela).

Conforme pode ser observado nas Figuras 52 e 55, o Servidor de Controle para comunicação através da porta paralela é semelhante ao desenvolvido para comunicação através da porta USB. Em uma barra superior se encontram as opções para configuração de conexão (1,2), botões (3) para iniciar e finalizar o

serviço de Servidor, além de botões de acesso ao carregamento de esquemas de cores (4) e da seção de ajuda do software (5). As telas da seção de ajuda do software e de carregamento de esquemas de cores são parecidas com as do software cliente, apresentadas, respectivamente, nas Figuras 46 e 47.

À esquerda se encontram também uma região (6) onde é apresentada a lista de usuários conectados ao sistema, e uma caixa de seleção para a escolha do usuário que estará habilitado ao controle do experimento ou dispositivo de controle, além da região (7) que possui um painel em que é apresentado o status do canal de coleta de dados e das saídas de controle.

Algumas diferenças podem ser observadas neste painel, que dizem respeito a quantidade de canais analógicos de leitura de dados, e a quantidade de saídas de controle. Para o software desenvolvido para comunicação através da porta paralela estas quantidades são menores.

Isto se deve principalmente a diferenças entre os circuitos de interface que são utilizados para comunicação através de cada porta.

Uma caixa de texto, e dois botões localizados em uma região (8) abaixo da barra superior, permitem que seja definido o nome do procedimento experimental ou de controle, e que se salve ou carregue configurações para um determinado procedimento.

Em uma região (9) central, podem ser definidos os comandos para ligar e desligar as oito saídas disponíveis no circuito de interface, e ser ativada ou não a opção “Apenas um Click”, que também está presente no Servidor de Controle para porta USB.

Assim como no software para porta USB, abaixo desta região central se encontram as opções (10) de configuração do tempo de intervalo de envio de dados coletados nos dispositivos, e opção, através de um botão, de habilitação ou não do envio de dados.

Existem também dois tipos de configurações de coleta de dados, uma para coleta de dados analógicos (11), e outro (12) para coleta de informações de estado dos dispositivos. Nestas configurações a única diferença para o Servidor de Controle USB está na quantidade de canais de coleta de dados analógicos, que é de apenas um canal para o Servidor de Controle com comunicação através da porta paralela.

Conforme se observa nas Figuras 53 e 56, as informações que são fornecidas na configuração são semelhantes entre os dois softwares.



Figura 56 – Destaque da região de configuração de dados de estado, do Software Servidor de Controle (Paralela).

5.2- Testes de funcionamento do sistema

São apresentados neste tópico os resultados obtidos durante a realização dos testes de funcionamento do sistema de Instrumentação Virtual.

5.2.1- Testes de transmissão de dados

Os testes de transmissão de dados no formato de texto (mensagens), foram realizados durante os testes de funcionamento com os experimentos.

Foram no total, cinco sessões de testes, sendo uma com o Experimento 1, duas com o Experimento 2 e mais duas com o Experimento 3.

Cada sessão durou cinco horas, com a medição do intervalo de tempo no momento que era enviado um comando de controle, o que ocorria a cada vinte minutos. Logo, foram feitas 15 medições do intervalo entre envio e recebimento de mensagens, para cada sessão de teste.

A Tabela 15, que se segue, apresenta os intervalos, em milissegundos, verificados em cada uma das 5 sessões:

Tabela 15 – Intervalo de tempo (ms) entre envio e recepção de mensagens

	E1- USB	E2- USB	E3- USB	E2- PAR	E3- PAR
1	129	104	59	108	101
2	103	68	112	82	139
3	90	117	135	111	96
4	131	93	116	115	67
5	81	54	105	71	103
6	86	131	68	135	117
7	144	83	144	103	93
8	97	104	134	98	121
9	110	91	129	125	60
10	119	131	114	54	133
11	120	84	144	85	74
12	57	55	59	100	148
13	91	85	125	122	138
14	65	122	76	51	114
15	111	101	63	74	127
Média:	102,3	94,9	105,5	95,6	108,7

A partir da Tabela 15, pode-se observar que a média geral obtida para o intervalo entre envio e recebimento de resposta, foi de aproximadamente 0,1 segundos.

Este intervalo de tempo é muito satisfatório. Um intervalo com este baixo valor permite que mensagens sejam trocadas quase que instantaneamente, e levando se em conta ainda a alta velocidade na coleta de dados de experimentos e dispositivos de controle, pode-se afirmar que tais procedimentos terão um acompanhamento em tempo real.

Já durante o teste de transferência de imagens, foi verificado, conforme pode ser observado na Figura 57, um tempo de atraso constante.

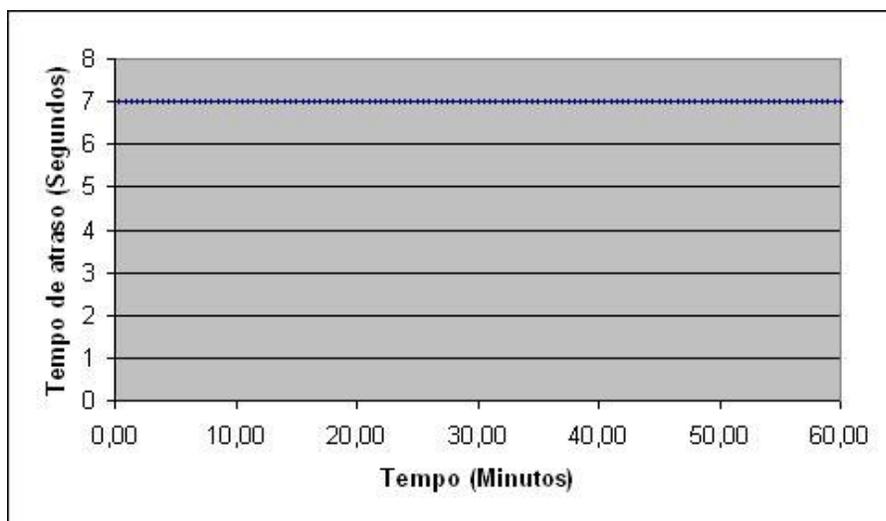


Figura 57 – Gráfico representando o tempo de atraso na transmissão de imagens.

O tempo de atraso no recebimento das imagens foi constante em 7 segundos. É um tempo alto, mas que, no entanto, dependendo do experimento ou dispositivo de controle utilizado, não compromete o acompanhamento destes.

Para experimentos e dispositivos de controle mais dinâmicos, como alguns experimentos de mecânica ou robôs de movimentação rápida, o sistema não se mostrou apropriado. Ressalta-se, no entanto, que esta é uma limitação da tecnologia de transmissão de imagens através da internet que se tem atualmente, que tende a evoluir conforme a também evolução das bandas de internet.

5.2.2- Testes com dispositivos de controle e experimentos

Neste tópico são apresentados alguns dos resultados obtidos durante o teste com cada dispositivo de controle e experimento.

Durante os testes com os dispositivos de controle, foi possível, com o uso do Sistema de Instrumentação Virtual desenvolvido, o acesso ao controle e coleta de informações de estado de cada dispositivo.

O sistema mostrou-se eficiente no controle e no acompanhamento dos procedimentos, a partir das informações de estado que eram enviadas pelo software Servidor de Controle.

Foi possível, para o Dispositivo de Controle 1, selecionar qual cor que deveria ser emitida pelo dispositivo. Durante os testes com o Dispositivo de Controle 2, foi possível efetuar o controle de movimentação do protótipo robótico construído.

No entanto, conforme já foi abordado, as imagens transmitidas apresentaram um atraso considerável. Apesar disto, o Sistema desenvolvido mostrou-se como uma interessante alternativa para o controle de dispositivos a distância, através da Internet ou de uma rede local.

Durante os testes com os experimentos, o Sistema também se mostrou eficiente, sendo possível o acompanhamento e controle dos procedimentos experimentais.

No Experimento 1, foram obtidas informações relativas a temperatura e luminosidade refletida por duas placas metálicas.

As Figuras 58 e 59 apresentam os gráficos obtidos a partir dos valores coletados neste experimento, ao longo das 5 horas de execução do procedimento experimental.

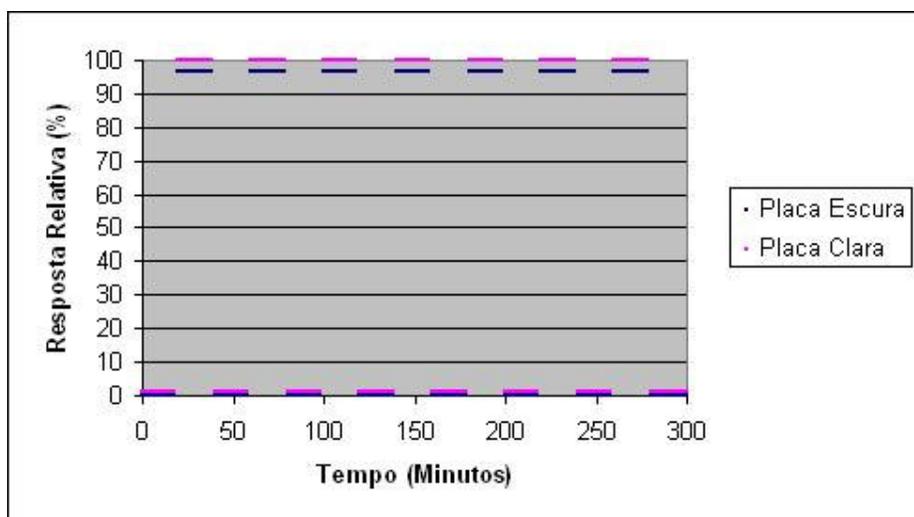


Figura 58 – Experimento 1: Resposta Relativa dos sensores de luminosidade x tempo - Verificação da luminosidade refletida por cada placa.

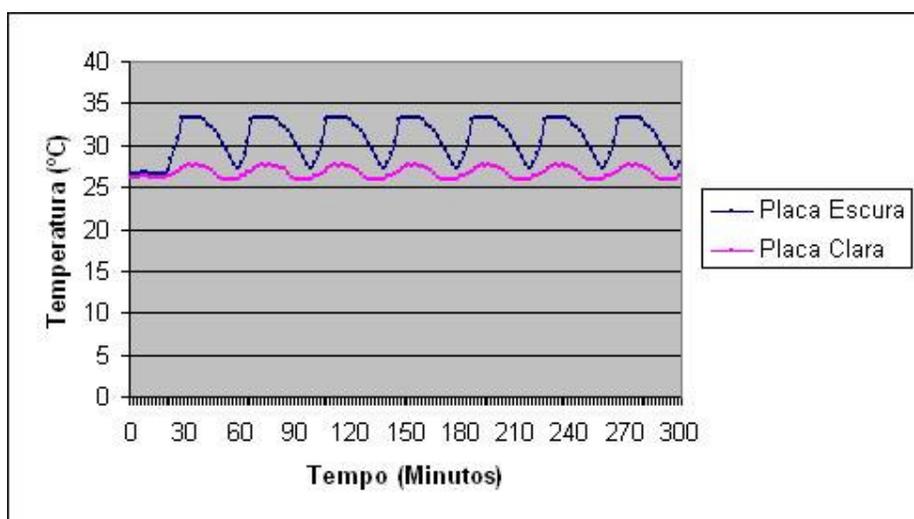


Figura 59 – Experimento 1: Temperatura x Tempo - Temperatura relativa a absorção de cada placa.

Conforme pode ser observado, este experimento permite diferenciar nitidamente, a capacidade de reflexão e absorção da luz, por placas claras e escuras. Desta forma, é possível concluir a adequação do experimento proposto, para a discussão conceitual sobre o fenômeno envolvido.

O Sistema de Instrumentação Virtual desenvolvido permitiu o acesso em tempo real a estes dados, além do controle do experimento, onde foi possível ligar ou desligar a fonte de luz do experimento.

O Experimento 2 buscava verificar a existência de um campo magnético nas proximidades de um condutor. Os resultados obtidos com este experimento são simples, e conforme podem ser interpretados a partir da Figura 60, se resumem a confirmação de que um campo magnético era gerado no momento em que o fio condutor era energizado.

Este experimento passou por duas sessões de testes, uma com a utilização da placa de aquisição de dados através da porta USB e outro através da porta paralela. Os resultados obtidos, para os dois casos, foram idênticos, e são representados através da Figura 60.

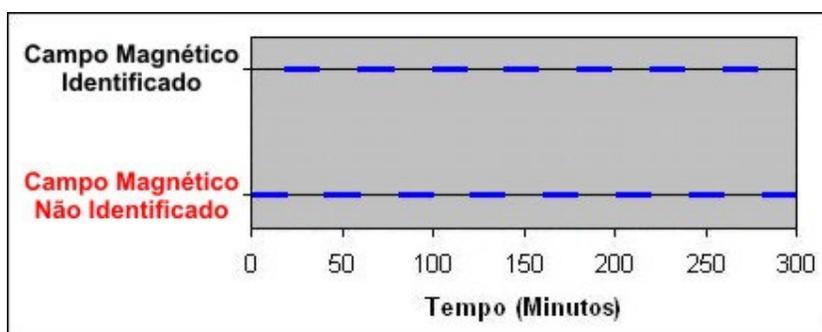


Figura 60 – Experimento 2: Detecção de campo magnético em função do tempo.

Conforme era esperado, no momento em que o condutor era energizado, era identificada a geração de um campo magnético.

Através do Sistema desenvolvido foi possível ligar ou desligar o fornecimento de energia do condutor. Foi possível, ainda, o acesso ao estado do experimento, sendo possível o acompanhamento da geração ou na ode um campo magnético.

O Experimento 3 passou também por duas sessões de testes, uma com o uso de cada placa de aquisição e controle de dados. Foi verificada a tensão elétrica gerada nas extremidades de um condutor.

Os dados coletados durante este procedimento experimental, foram semelhantes para as duas sessões de testes, conforme é possível de se observar, através das Figuras 61 e 62.

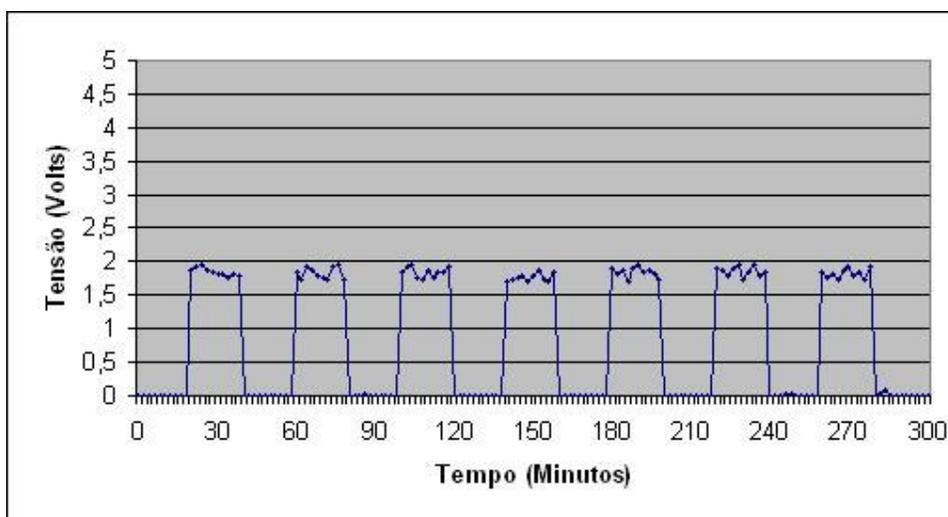


Figura 61 – Experimento 3: Tensão elétrica verificada (1ª Sessão-USB).

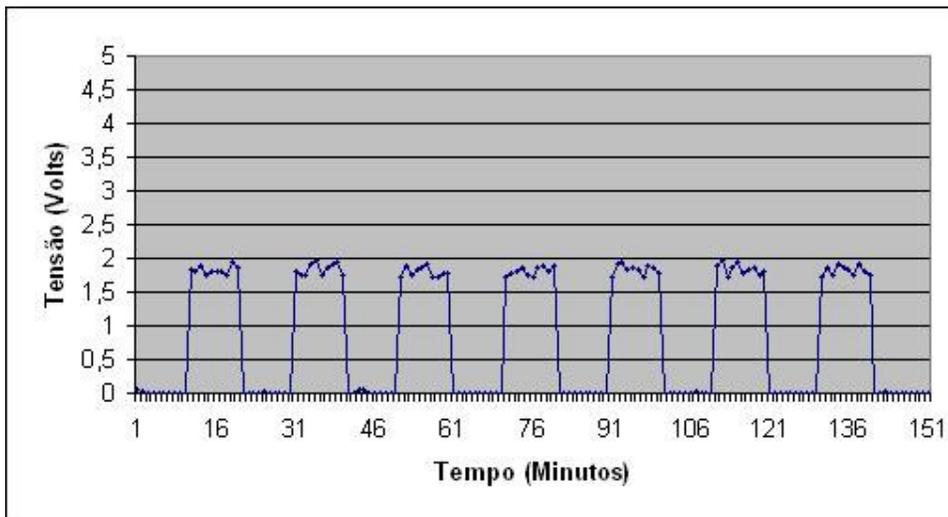


Figura 62 – Experimento 3: Tensão elétrica verificada (2ª Sessão-Paralela).

No momento em que a “hélice magnética” encontrava-se em movimento, foi verificada uma tensão elétrica nas extremidades de um condutor que se encontrava próximo a esta hélice.

O Sistema de Instrumentação Virtual desenvolvido permitiu o controle sobre o motor que movia as “hélices magnéticas” do experimento.

Os dados coletados no experimento, que puderam ser acessados a distância, mostram que no momento em que o motor estava desligado, não foi identificada uma tensão elétrica nas extremidades do condutor, e no momento em que o motor encontrava-se ligado, foi observada uma tensão elétrica que oscilou entre 1,71 Volts e 1,97 Volts.

Através das imagens coletadas e transmitidas aos softwares clientes, foi possível acompanhar a movimentação das hélices, possibilitando relacionar o que era visto com os dados coletados no experimento.

5.3- Análise Técnica

Neste tópico é apresentada uma análise técnica do Sistema de Instrumentação Virtual desenvolvido.

Inicialmente é efetuada uma análise da qualidade dos softwares, segundo as normas técnicas estabelecidas. A seguir, é apresentada uma análise comparativa entre as portas paralela e USB, assim como os hardwares utilizados.

5.3.1- Qualidade de software

No momento do desenvolvimento dos softwares, foram consideradas as características e sub-características previstas para um software de qualidade. No entanto, algumas destas podem ser aperfeiçoadas, conforme se evidencia na análise que se segue.

Funcionalidade

Os softwares mostraram-se adequados para as funções para quais foram criados, os resultados das medições apresentaram-se confiáveis para o uso como ferramenta pedagógica, com precisão das medidas em 8 bits. Os softwares conseguem interagir entre si.

A única sub-característica que deve ser aperfeiçoada diz respeito a segurança de acesso. Qualquer pessoa utilizando um software cliente pode se conectar ao sistema, se possuir o endereço do servidor e o número da porta de comunicação. É interessante a elaboração de um controle de acesso, mediado por “login” e “senha”, o qual permitirá o acesso, somente, de usuários cadastrados.

Confiabilidade

As versões finais dos softwares mostraram-se confiáveis. Foram utilizadas lógicas durante a programação dos softwares, com o objetivo de se prevenir erros. Como exemplo, na função de carregamento automático de configurações iniciais dos softwares, o código utilizado verifica se existem os arquivos com tais configurações, e somente se existirem, efetua-se a função.

As poucas falhas identificadas foram corrigidas e não eram ocasionadas pelos softwares. Tais falhas identificadas eram de hardware, e ocorriam devido a

interferências nas placas de aquisição de dados, principalmente na placa USB. Verificou-se que o hardware deve ser mantido distante de campos magnéticos mais intensos, por exemplo, dos ímãs dos experimentos, e deve-se evitar ligar a placa de aquisição e controle de dados em uma fonte que oscile muito. Logo, faz-se necessário a utilização de um estabilizador de tensão, de uso comum em computadores, para se ligar a placa.

Usabilidade

Dos softwares desenvolvidos, o Servidor de Comunicação e o Cliente atendem as sub-características de usabilidade, que dizem respeito a facilidade de uso do software.

No entanto, os softwares do tipo Servidor de Controle, podem ainda ser aperfeiçoados. Os softwares oferecem uma certa dificuldade de configuração, é necessário um conhecimento prévio e básico de eletrônica, ou um estudo do tutorial disponível no software.

No entanto ressalta-se, que tal dificuldade é proporcionada ainda pelo caráter de adaptabilidade do software. Como se trata de softwares que são adaptáveis a várias aplicações, faz-se necessário a configuração para cada experimento ou dispositivo que é utilizado.

Eficiência

No que diz respeito ao tempo de processamento e a quantidade de recursos necessários, os softwares apresentaram um alto grau de eficiência. Os softwares não requerem um uso considerável do computador em que estejam sendo utilizados, utilizando pouco os recursos de hardware.

Manutenibilidade

A facilidade em correções de falhas, alterações e atualizações dos softwares, foi também considerada durante a programação dos mesmos. Os códigos fontes dos softwares foram escritos com diversos comentários sobre cada parte do código, permitindo que a correção de falhas e alterações ocorram com grande facilidade.

Já sobre a facilidade de atualizações, é algo que pode ser otimizado. Os softwares utilizam poucos arquivos externos, os quais permitiriam uma atualização facilitada. Podem ser desenvolvidas bibliotecas de funções, através do uso de arquivos "dll", facilitando assim, as atualizações do software.

No momento, para que sejam feitas atualizações dos softwares, novas versões dos arquivos executáveis devem ser compiladas, e estas versões devem substituir as versões mais antigas. O uso de arquivos externos, como arquivos do tipo “dll”, permitiria que fossem necessárias poucas atualizações dos arquivos executáveis, podendo ocorrer atualizações pela substituição dos arquivos externos.

Porém, levando-se em conta que o tamanho em bytes dos executáveis dos softwares não são grandes, não ultrapassando 1 megabytes, as atualizações com substituição destes executáveis não se apresenta como um problema.

Portabilidade

Os softwares foram desenvolvidos para a utilização em sistemas operacionais Windows, especialmente em versões mais recentes, como o Windows XP e Vista.

No entanto, testes de uso destes softwares em Sistemas Operacionais Linux, através do emulador de aplicativos windows, o Wine, foram realizados. O funcionamento apresentado foi perfeito, não sendo perceptíveis falhas.

Apesar disto, é interessante o desenvolvimento de aplicativos específicos para Linux, permitindo assim, a utilização direta, sem o uso de emuladores de aplicativos. Isto facilitaria o uso, principalmente dos softwares clientes.

5.3.2- Análise técnica e comparativa entre portas paralela e USB

Com base na experiência de uso e implementação de comunicação através das portas paralela e USB, foi possível concluir diversos parâmetros.

Sobre a velocidade de tráfego de dados, conforme pode ser observado na Tabela 16, em uma comparação entre as versões mais recentes das portas paralela e USB, verifica-se que a porta USB é bem superior.

Tabela 16 – Velocidade máxima de tráfego de dados das portas paralela e USB

Porta de Comunicação	Velocidade Máxima de Tráfego de Dados
Paralela (Modelo SPP)	150Kb/s
Paralela (Modelo EPP)	2Mb/s
USB (Modelo 1.0)	1,5Mb/s
USB (Modelo 1.1)	Entre 1,5Mb/s e 12Mb/s
USB (Modelo 2.0)	480Mb/s

Pela altíssima velocidade de tráfego de dados da porta USB, é possível o desenvolvimento de hardwares com vários canais de coleta de dados.

Ressalta-se ainda a possibilidade de conexão de vários dispositivos através de uma mesma porta USB, com o uso de Hubs USB, o que não é possível através da porta paralela. A porta paralela, além de não ser mais disponibilizada em alguns computadores mais recentes, não permite uma conexão facilitada de mais de um dispositivo.

Quanto ao desenvolvimento de softwares para a comunicação através de cada porta, pode-se concluir que a porta paralela apresenta um nível bem maior de facilidade de desenvolvimento.

Os códigos utilizados para se estabelecer a comunicação, efetuar o controle de saída de dados, e para efetuar a leitura de dados, através da porta paralela, são compactos, ocupando poucas linhas, ao contrário daqueles utilizados no uso da porta USB, que além de ocuparem várias linhas, foram determinados em função da placa de aquisição de dados utilizada.

Desta forma, pode-se concluir que o desenvolvimento de softwares para a comunicação através da porta paralela, é não somente mais fácil se comparado ao desenvolvimento de softwares de comunicação através da porta USB, mas também mais rápido.

Quanto ao desenvolvimento de hardwares para a comunicação através de cada porta, a porta USB leva mais uma vez desvantagem. Hardwares para a porta paralela são normalmente mais simples e com um custo mais baixo.

Durante a etapa de planejamento da pesquisa, optou-se pelo uso de uma placa comercial de comunicação USB, devido principalmente a este fato. O desenvolvimento de um hardware iria demandar um tempo grande, e era previsto para este desenvolvimento um custo superior ao da placa de aquisição de dados utilizada.

Apesar desta maior complexidade na utilização da porta USB, por uma série de fatores, o uso das portas USB mostra-se mais interessante se comparadas ao uso das portas paralela. Entre esses fatores têm-se que as portas USB possuem uma velocidade de comunicação bem superior; são disponibilizadas várias em um mesmo computador e com possibilidade de conexão facilitada de mais de um dispositivo em uma mesma porta; e ainda há uma tendência de extinção da porta paralela (que já é percebida em alguns computadores mais recentes, principalmente em laptops).

6 – APLICABILIDADE DA FERRAMENTA NO ENSINO A DISTÂNCIA

Neste capítulo, é apresentado um estudo sobre a aplicabilidade do sistema de instrumentação virtual desenvolvido.

O sistema de instrumentação virtual desenvolvido pode ser utilizado em cursos superiores de ensino a distância.

O Brasil já conta com diversos cursos a distância. Dentre estes, os que se destacam como aqueles que necessitam de aulas experimentais, são os de Física, Química, Matemática e Ciências. Tais cursos possuem geralmente em suas grades curriculares, disciplinas experimentais de Física e Química.

A partir de um levantamento realizado em 2008, através do Portal SiedSup do MEC (SiedSup/MEC, 2008), foi possível verificar a quantidade de cursos de Física, Química, Matemática e Ciências, na modalidade de EaD, que se encontram autorizados a funcionar. Os cursos autorizados são todos de Licenciatura Plena na área.

A Tabela 17 apresenta a quantidade de cursos de cada uma destas áreas, oferecidos em cada estado.

Tabela 17 – Quantidade de cursos EaD em cada área, por estado.

	Física	Química	Matemática	Ciências	Total
Acre	0	0	0	0	0
Alagoas	1	0	0	0	1
Amapá	0	0	0	0	0
Amazonas	0	0	0	0	0
Bahia	1	0	3	0	4
Ceará	2	2	2	0	6
Distrito Federal	0	0	0	0	0
Espírito Santo	0	0	0	0	0
Goiás	0	0	0	0	0
Maranhão	0	1	0	0	1
Mato Grosso	0	1	0	1	2
Mato Grosso do Sul	0	0	0	0	0
Minas Gerais	2	5	4	0	11
Pará	1	9	2	0	12
Paraíba	0	0	1	1	2
Paraná	0	0	0	0	0
Pernambuco	1	0	1	0	2
Piauí	1	1	1	0	3
Rio de Janeiro	1	1	3	0	5
Rio Grande do Norte	1	1	1	0	3
Rio Grande do Sul	0	1	3	0	4
Rondônia	0	0	0	1	1
Roraima	0	0	0	0	0

Tabela 17, Continuação

	Física	Química	Matemática	Ciências	Total
Santa Catarina	1	0	4	0	5
São Paulo	1	1	6	0	8
Sergipe	1	1	2	1	5
Tocantins	0	0	1	0	1
Total	14	24	34	4	76

Fonte: SiedSup/MEC (2008)

A Figura 63 apresenta estes mesmos dados graficamente:



Figura 63 – Mapa dos cursos EaD de Física, Química, Matemática e Ciências.

Segundo os dados do Portal SiedSup do MEC, apenas 9 unidades federativas ainda não possuem cursos EaD nas áreas destacadas. São, atualmente, 76 cursos superiores EaD, e no mínimo, 26808 vagas nestes cursos.

Muitas das disciplinas experimentais presentes nas grades curriculares destes cursos, ocorrem em aulas presenciais.

Tecnologias que permitam que estas aulas ocorram de forma não presencial, a distância, são interessantes, pois permitem uma maior facilidade para os estudantes e professores destes cursos.

A ferramenta desenvolvida permitiria que estas aulas experimentais pudessem ocorrer a distância, através da internet, possibilitando o controle e acompanhamento simultâneo de um procedimento experimental por até 50 estudantes. Esta quantidade de 50 estudantes está relacionada a uma limitação do software encoder, que permite a conexão de no máximo 50 usuários.

O sistema desenvolvido mostrou-se, durante os testes, como uma ferramenta eficiente para permitir o controle e coleta de dados de experimentos.

Com base na teoria de Vygotsky, a elaboração de conteúdos teóricos, acessíveis através do software Cliente e específicos para cada experimento, permitiria um desenvolvimento real do estudante, e a partir da interação com outros usuários ou mesmo com tutores, seria possível também um desenvolvimento potencial.

A visualização dos experimentos através de imagens reais destes, permite que os estudantes possam relacionar mais facilmente aquilo que está sendo visualizado, com o conteúdo teórico que lhe foi apresentado.

Logo, levando-se em conta uma quantidade cada vez maior de cursos de educação a distância em que a experimentação se faz necessária, com base nos resultados obtidos, e considerando ainda os variados aspectos positivos do uso de experimentos e novas tecnologias no ensino, é possível concluir que a ferramenta desenvolvida possui um alto grau de aplicabilidade.

7 - CONCLUSÃO

7.1- Considerações Finais

No capítulo inicial foi apresentada uma introdução sobre a Instrumentação Virtual, com um breve retrospecto do desenvolvimento do tratamento da informação, até o surgimento desta área tecnológica, a instrumentação mediada por computadores.

Ainda neste primeiro capítulo, foram apresentadas algumas possíveis aplicações deste tipo de instrumentação. A pesquisa foi justificada e os objetivos apresentados, também neste capítulo introdutório.

O segundo capítulo consistiu em uma revisão de trabalhos já desenvolvidos, principalmente no Brasil, que abordavam temas relacionados a presente pesquisa.

Em um terceiro capítulo foram apresentadas as bases teóricas pedagógicas e técnicas, utilizadas neste trabalho. Este capítulo foi adicionado como forma de facilitar o entendimento durante a leitura da dissertação. Isto foi necessário, pois a pesquisa possui um caráter interdisciplinar, possuindo referenciais de diferentes áreas do conhecimento, incluindo desde parâmetros pedagógicos até parâmetros técnicos, como os envolvidos no desenvolvimento de softwares e hardwares.

A descrição do desenvolvimento da pesquisa, com os materiais e métodos adotados, foi apresentada através do capítulo quarto. Conforme descrito neste quarto capítulo, a pesquisa teve seis etapas de desenvolvimento, incluindo uma etapa de planejamento, etapas de desenvolvimento de softwares, hardwares, de elementos para que fosse possível a realização de testes, e em uma última etapa, que consistiu na realização de testes do sistema desenvolvido.

No quinto capítulo são disponibilizados alguns dos resultados obtidos, em relação ao desenvolvimento dos softwares e aos testes de funcionamento do sistema. Neste capítulo foram apresentadas ainda, além de uma análise sobre os resultados dos testes, uma análise técnica sobre os parâmetros de qualidade dos softwares, e uma análise comparativa entre as porta de comunicação paralela e USB.

Em um sexto capítulo foi apresentado um levantamento de dados relativos a Educação a Distância no Brasil, com o quantitativo de cursos em que comumente possuem aulas experimentais em sua grade curricular. Com bases nestes dados e

nos resultados obtidos, foi realizada uma análise da aplicabilidade do sistema de instrumentação virtual desenvolvido, no EaD.

Baseado nas informações apresentadas neste trabalho, e na própria experiência durante o desenvolvimento da pesquisa, é possível concluir que o sistema desenvolvido pode ser uma eficiente ferramenta para o uso no EaD, podendo se adaptar a diversos procedimentos experimentais.

Esta característica de adaptabilidade do sistema permite ainda que o mesmo possa ser utilizado para outros fins, onde seja necessário o acompanhamento e controle de processos a distância, através da Internet ou mesmo através de uma rede local.

Outra importante característica do sistema é a possibilidade de acesso simultâneo de vários usuários. Além de permitir que diversos usuários possam em um mesmo momento acompanhar o controle e a coleta de dados do dispositivo que esteja sendo acessado, possibilita a interação entre os usuários conectados, característica esta de grande importância para um sistema aplicável em um processo de aprendizagem.

7.2- Trabalhos Futuros

Como possíveis trabalhos futuros, podem ser realizadas pesquisas envolvendo testes do sistema desenvolvido, junto a um grupo grande de estudantes. Tais testes poderiam ocorrer em conjunto com instituições que oferecem cursos superiores, os quais possuam disciplinas que exijam aulas experimentais.

Outro possível trabalho está relacionado ao aperfeiçoamento dos softwares, que conforme já relatado, podem ter ainda algumas novas funções implementadas.

É interessante que seja elaborado um controle de acesso, de forma que no momento da conexão de um software Cliente, seja solicitado um “login” e uma senha, e assim, somente usuários cadastrados em um banco de dados poderiam acessar o sistema. Para isto, faz-se necessário que esta função seja adicionada no software Servidor de Comunicação.

Nos softwares Servidores de Controle, poderá ser implementada ainda uma função que permita a medição de intervalos de tempo. Alguns experimentos, como os de mecânica, possuem como principal grandeza a ser medida, o intervalo de tempo entre dois ou mais momentos.

Uma última função que pode ser elaborada é a de medição de valores não lineares. Os softwares do tipo servidor de controle possuem a interpretação somente de medições com comportamentos lineares, o que faz com que seja necessário que os valores obtidos em sensores não-lineares sejam linearizados, com a utilização de circuitos auxiliares e aproximações matemáticas. É possível a implementação de uma função que reconheça, além de grandezas lineares, outras com comportamentos não-lineares, como comportamentos polinomiais, exponenciais, etc.

8 - REFERÊNCIAS

Aguiar, C. E.; Laudares (2001), F. Aquisição de Dados com Logo e Porta de Jogos do PC. Revista Brasileira de Ensino de Física. v.23; n.4; p. 371-380, 2001

Alves, J. (1994), A Educação a Distância no Brasil: Síntese Histórica e Perspectivas. Rio de Janeiro, Instituto de Pesquisas Avançadas em Educação.

Andrade, M. E. C. de, (2001), Processo de Ensino-Aprendizagem via Internet. Tese (Mestrado em Engenharia de Produção) – Florianópolis-SC, Universidade Federal de Santa Catarina – UFSC, 179p.

Ausubel, D., Novak., J., Hanesian, H. (1980) Psicologia Educacional, Rio de Janeiro: Interamericana.

Bakos, M. M. (1996), O que são hieróglifos, 1ª Ed., Brasília, Brasiliense, 1996.

Boechat, V. A. P. (2006), Ambientes Virtuais para o Ensino de Física. Tese (Mestrado em Ciências Naturais) – Campos dos Goytacazes-RJ, Universidade Estadual do Norte Fluminense – UENF, 152p.

Boratti, I. C. (2004), Programação orientada a objetos usando Delphi, Florianópolis, Visual Books, 2004.

Borges, A. P. (2002), Instrumentação Virtual aplicada a um Laboratório com acesso pela Internet. Tese (Mestrado em Engenharia Elétrica) – São Paulo-SP, Universidade de São Paulo – USP, 128p.

Borges, R. M. R.,(2007), Em debate: cientificidade e educação em ciência, 2ª ed. rev. ampl. - Porto Alegre: EDIPUCRS, 2007. Pág. 29.

Brasil (1996), Lei nº 9.394, de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional, 1996.

Brasil (2005), Decreto nº 5.622, de 19 de dezembro de 2005, Presidência da República, Casa Civil, Subchefia de Assuntos Jurídicos, 2005.

Bueno, A. D. (2003), Programação Orientada a Objeto com C++, 1ªEd., São Paulo, Novatec, 2003.

Carvalho, M. S. R. M de, (2006) A trajetória da Internet no Brasil: do surgimento das redes de computadores à instituição dos mecanismos de governança. Tese (Mestrado em Engenharia de Sistemas e Computação) – Rio de Janeiro-RJ, COPPE/Universidade Federal do Rio de Janeiro – COPPE/UFRJ, 239p.

Comer, D. E. (2001), *Redes de Computadores e Internet*, 2ª Ed., Porto Alegre, Bookman, 2001.

CREF (2002), *Experimentação Remota para o Ensino de Física*. Disponível em: <http://www.if.ufrgs.br/cref/hp/LER.html>. Acesso em 1 de Out. 2008.

Eckstein, S. G. (1990), Verification of fundamental principles of mechanics in the computerized student laboratory. *American Journal of Physics*, Woodbury, v. 58, n.10, p. 909-915, Oct. 1990.

Eichler, J. (1988), Um sistema de medição usando microcomputador acoplado a um experimento em Física (Oscilações Elétricas). *Revista Brasileira de Ensino de Física*, v. 10, n. 2, p. 12-23, dez. 1988.

Fialho, A. B. (2002), *Instrumentação Industrial: Conceitos, Aplicações e Análises*, 1ª Ed., Tatuapé, Érica, 2002.

Gonçalves, W. M.; Heinrich, A. F.; Sartorelli, J. C. (1991), Aquisição de dados com a porta de jogos de microcomputadores Apple. *Revista Brasileira de Ensino de Física*, v. 13, n. 5, p. 63-76, dez. 1991.

Hower, C. Z. (2004), *Indy 9.0 Help*, Disponível em: <http://www.projectindy.org/downloads/Indy-9-0-Help-Pdf.zip>. Acesso em 1 de out. 2008.

International Standard, (1997), *ISO9126-1: ISO/IEC 9126-1. Information Technology - Software quality characteristics and metrics - Part 1: Quality characteristics and sub-characteristics*; Jan/ 1997 (CD).

Internet Society (2003), *Histories of the Internet: A Brief History of the Internet*, 2003. Disponível em: <http://www.isoc.org/internet/history/brief.shtml>. Acesso em 9 de Set. 2007.

Kuhn, T. (2003), *A Estrutura das Revoluções Científicas*, 9ª Ed., São Paulo, Perspectiva, 2003.

Landim, C. M. F. (1997), *Educação a distância: algumas considerações*. Rio de Janeiro, UFRJ.

Logix4u.net (2003), *Input32.dll for WIN NT/2000/XP*. Disponível em: <http://www.logix4u.net/input32.htm>. Acesso em 14 de set. 2008.

Maps, J. A. (1993), A computer-based data acquisition laboratory for undergraduates. *American Journal of Physics*, Woodbury, v. 61, n.7, p. 651-655, July 1993.

Maxim, (2001), APPLICATION NOTE 817 - Using Thermistors in Temperature Tracking Power Supplies. Disponível em:

<http://pdfserv.maxim-ic.com/en/an/AN817.pdf>. Acesso em 10 de jul. 2008.

Messias, A. R., (2007), Curso USB/Serial - Controle de Dispositivos, RogerCom.com, 2007.

Microsoft, (2007), Windows Media Encoder 9 Series, 2007. Disponível em: <http://www.microsoft.com/windows/windowsmedia/pt/9series/encoder/default.aspx>.

Acesso em 9 de set. 2007.

Moran, J. M. (2002), O que é educação a distância. , 2002. Disponível em: <http://www.eca.usp.br/prof/moran/dist.htm>. Acesso em 1 de out. 2008.

National Semiconductor (1999), Data Sheet ADC0801/ADC0802/ADC0803/ADC0804/ADC0805, Disponível em: <http://www.national.com/ds.cgi/DC/ADC0801.pdf>. Acesso em 10 de set. 2008.

National Semiconductor (2003), Data Sheet LM3914: Dot/Bar Display Driver. Disponível em: <http://www.national.com/green/pdf/LM3914VNOPB.pdf>. Acesso em 15 de jul. 2008.

Netto, A. A. O. (2005) Novas Tecnologias & Universidade: da didática tradicionalista à inteligência artificial: desafios e armadilhas. Petrópolis, RJ: Vozes.

Oliveira, M. K. de (1995) Aprendizado e desenvolvimento um processo sócio-histórico. São Paulo: Scipione.

Pessanha, M. C. R. (2006), Sistema Didático Automatizado para o Ensino de Conceitos de Eletromagnetismo, Monografia Final de Curso (Licenciatura em Física) - Campos dos Goytacazes-RJ, Universidade Estadual do Norte Fluminense - UENF.

Pignatari, D. (2002), Informação Linguagem Comunicação, 25ª Ed., São Paulo, Ateliê Editorial.

Piletti, N. (1999), Psicologia educacional. São Paulo: Ática.

Ponczek, R. L. (2002), Da Bíblia a Newton: uma visão humanística da Mecânica. In(itálico): Rocha, J. F. M., Origens e Evolução das Idéias da Física(itálico). 1ª Ed. Salvador, p. 42-44, EDUFBA, 2002.

Preston, D. W.; Good, R. H. (1996), Computers in the general physics laboratory. American Journal of Physics, Woodbury, v. 64, n.6, p. 766-772, June 1996.

Rexlab (2003), Portal RExLab - Multidisciplinaridade e Conhecimento, Disponível em: <http://www.rexlab.ufsc.br/>. Acesso em 1 de out. 2008.

Ribas, R. V.; de Souza, A. F.; Santos, N. (1998), Um sistema de aquisição de dados de baixo custo para o laboratório didático. *Revista Brasileira de Ensino de Física*, v. 20, n. 3, p. 293-295, set. 1998.

Rocha, H. C. da, (2000), *Educação a Distância Concepções, Metodologias e Recursos*. Tese (Mestrado em Engenharia de Produção) – Florianópolis-SC, Universidade Federal de Santa Catarina – UFSC, 101p.

Rogercom.com (2007), *Manual da Placa R-CONTROL 30*, 2007. Disponível em: <http://www.rogercom.com/Produtos/PlacaRcontrol30/Manual-RCONTROL30.pdf>. Acesso em 9 de set. 2007.

Sebesta, R. (2000), *Conceitos de Linguagens de Programação; Tradução de José Carlos Barbosa dos Santos*. 4ª ed. - Porto Alegre: Bookman

SiedSup/MEC, (2008), *Cadastro das Instituições de Ensino Superior: Busca de Curso*, Disponível em: http://www.educacaosuperior.inep.gov.br/funcional/busca_curso.stm. Acesso em 8 de out. 2008

Silonex (2007a), *Standard Optoelectronic Components*. Disponível em: <http://www1.silonex.com/datasheets/specs/images/pdf/103259.pdf>. Acesso em 10 de jul. 2008.

Silonex (2007b), *TO-5 Photocells*. Disponível em: <http://www1.silonex.com/optoelectronics/photocells.html>. Acesso em 10 de jul. 2008.

Sokoloff, D. R.; Thornton, R. K. (1997), Using interactive lecture demonstrations to create an active learning environment. *The Physics Teacher*, College Park, v. 35, n.6, p. 340-347, Sep.1997.

Takahashi, H. L., Hattori, K. H. (2000) *Inteligência Artificial voltada à Educação*. Disponível em: http://www.din.uem.br/ia/a_correl/iaedu/biografia.htm#papert. Acesso em 1 de out. 2008.

Valente, J. A., Almeida, F. J. (1997) *Visão Analítica da Informática na Educação no Brasil: A Questão da Formação do Professor*. Disponível em: http://www.edutec.net/Textos/Alia/PROINFO/prf_txtie13.htm Acesso em 9 de set. 2007.

Vygotsky, L. (1998) *Pensamento e linguagem*. SP: Martins Fontes.

Volpato, A. N.; Soprano, A.; Bottan, E. R.; Diehl, F.; Provesi, J. R.; da Rosa, L. M.; Pinheiro, M. M.; Marqueze, M.. *Mídia e Conhecimento: Educação a Distância*.

Disponível em: <http://www.paulofreire.ufpb.br/paulofreire/Files/arceloni.htm>. Acesso em 1 de out. 2008