

KELLER SULLIVAN OLIVEIRA ROCHA

**SISTEMA COMPUTACIONAL MULTIPLATAFORMA PARA
ARMAZENAMENTO E TRANSMISSÃO REMOTA DE DADOS**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do programa de
Pós-Graduação em Engenharia Agrícola
para obtenção do título de *Magister
Scientiae*.

VIÇOSA
MINAS GERAIS – BRASIL
2008

**Ficha catalográfica preparada pela Seção de Catalogação e
Classificação da Biblioteca Central da UFV**

T

R672s
2008

Rocha, Keller Sullivan Oliveira, 1979-

Sistema computacional multiplataforma para armazenamento e transmissão remota de dados / Keller Sullivan Oliveira Rocha. – Viçosa, MG, 2008.
xv, 47f.: il. (algumas col.) ; 29cm.

Inclui apêndice.

Orientador: José Helvecio Martins.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f. 42-44.

1. Engenharia Agrícola. 2. Engenharia auxiliada por computador. 3. Armazenamento de dados. 4. Sistemas de controle inteligente. 5. Banco de dados - Gerência. 6. Sistemas de informação gerencial.
I. Universidade Federal de Viçosa. II. Título.

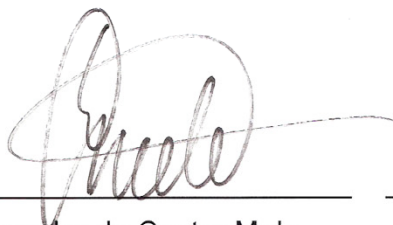
CDD 22.ed. 630

KELLER SULLIVAN OLIVEIRA ROCHA

**SISTEMA COMPUTACIONAL MULTIPLATAFORMA PARA
ARMAZENAMENTO E TRANSMISSÃO REMOTA DE DADOS**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do programa de
Pós-Graduação em Engenharia Agrícola
para obtenção do título de *Magister
Scientiae*.

Aprovada em 18 de fevereiro de 2008



Prof. Evandro de Castro Melo
(Co-Orientador)



Prof. Paulo Marcos de Barros Monteiro



Profª. Ilda de Fátima Ferreira Tinoco



Dr. Wathney Hermsdorff



Prof. José Helvecio Martins
(Orientador)

“... a mente que se abre a uma nova idéia...
... jamais voltará ao seu tamanho original...”

Albert Einstein

Aos meus pais Geraldo Rocha e Maria Odezi de Oliveira Rocha,
aos meus irmãos Kelly Cristina e Kelvin
e a minha namorada Damiana,
com Amor e Gratidão.

AGRADECIMENTOS

A Deus, pelo dom da vida, saúde, sabedoria e presença constante em minha caminhada.

Aos meus Pais, Geraldo Rocha e Maria Odezi de Oliveira Rocha, e aos meus irmãos, Kelly Cristina Oliveira Rocha e Kelvin Oliveira Rocha, pelas lições de vida.

A Márcio e Sibelle por toda ajuda e confiança.

À Damiana Diniz Rosa pelo amor, carinho e paciência em todos os momentos difíceis que passei em Viçosa.

Ao Pedro Bola pelo companheirismo e amizade.

À Fabiana Rodrigues Matheus pela nossa grande amizade e carinho.

Ao Prof. José Helvecio Martins, pela orientação, amizade, confiança e apoio constante nesta importante etapa da minha formação.

Ao Prof. Evandro de Castro Melo e à Dra. Daniela de Carvalho Lopes, pela co-orientação e apoio indispensável para a realização deste trabalho.

Ao Dr. Wathney Hermsdorff, pela amizade e apoio na elaboração deste trabalho.

Aos Professores Mauri Fortes, Marcos Teixeira Godinho, Ilda de Fátima Ferreira Tinôco e Paulo Marcos de Barros Monteiro pelo grande apoio na minha caminhada acadêmica, confiança, amizade e pelos valiosos ensinamentos e apoio constantes.

À Universidade Federal de Viçosa e, em especial, ao Departamento de Engenharia Agrícola, pela oportunidade de realização do curso.

À Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), pela concessão da bolsa de estudos.

À equipe da Energia na Agricultura, ao Núcleo de Pesquisa em Ambiência e Arquitetura de Sistemas Agroindustriais (AMBIAGRO) e toda a sua equipe pelo apoio.

Aos meus amigos e colegas pelo companheirismo, amizade, idéias e incentivo.

E Agradeço a todos que direta ou indiretamente auxiliaram na execução deste trabalho.

BIOGRAFIA

Keller Sullivan Oliveira Rocha, filho de Geraldo Rocha e Maria Odezi de Oliveira Rocha, nasceu em Belo Horizonte, Estado de Minas Gerais, em 10 de Janeiro de 1979.

Em Março de 2002, iniciou o curso de Sistemas de Informação no Centro Universitário UNA, graduando-se em 2005.

Em Março de 2006, ingressou no programa de Pós-Graduação em nível de Mestrado em Engenharia Agrícola da Universidade Federal de Viçosa (UFV), Área de Concentração em Energia na Agricultura, submetendo-se à defesa de tese, requisito indispensável para obtenção do título de *Magister Scientiae*, em fevereiro de 2008.

SUMÁRIO

LISTA DE FIGURAS	viii
RESUMO	xii
ABSTRACT	xiv
1. INTRODUÇÃO.....	1
2. OBJETIVOS.....	4
3. REVISÃO DE LITERATURA.....	5
3.1. AS REDES DE COMPUTADORES E A INTERNET.....	5
3.1.1. Topologia em barramento (bus)	6
3.1.2. Topologia em estrela.....	6
3.1.3. Topologia em anel.....	6
3.1.4. A INTERNET	7
3.1.5. Tipos de Rede	7
3.1.6. Protocolos	8
3.1.7. Arquitetura <i>Cliente-Servidor</i>	9
3.1.8. HTTP e Serviços Web.....	10
3.1.9. Sites Dinâmicos.....	12
3.2. LINGUAGENS DE PROGRAMAÇÃO.....	13
3.2.1. Hypertext Markup Language - HTML	13
3.2.2. Hypertext Preprocessor - PHP	14
3.2.3. Linguagem Java.....	14
3.2.4. Structured Query Language - SQL.....	15
3.2.5. eXtensive Markup Language - XML	16
3.3. Gerenciamento Remoto de Experimentos	17
3.4. Trabalhos Relacionados	18
4. METODOLOGIA.....	20
4.1. DESENVOLVIMENTO DE UMA PROPOSTA DE COMUNICAÇÃO ...	20
4.1.1. Terminal do Projeto - Cliente.....	22
4.1.2. Servidor do Projeto.....	23
4.1.3. Lado do Usuário	24
4.1.4. Servidor Web	25
4.2. DESENVOLVIMENTO DO MODELO DE BANCO DE DADOS.....	25
4.2.1. Dados em tempo real.....	27
4.2.2. Arquivos Anexados	27

4.3. ENSAIO DE AQUISIÇÃO DE DADOS.....	28
5. RESULTADOS E DISCUSSÃO	29
5.1. MODELO DE COMUNICAÇÃO	29
5.1.1. Ambiente de teste	29
5.1.2. Ambiente <i>Cliente-Servidor</i> Web	29
5.1.3. Ambiente <i>Cliente-Servidor</i> do Projeto	35
5.2. SISTEMA DE BANCO DE DADOS.....	37
5.3. TESTES DE ADEQUAÇÃO DO SISTEMA COMPUTACIONAL.....	37
6. CONCLUSÕES.....	40
6.1. SUGESTÕES PARA TRABALHOS FUTUROS.....	41
7. REFERÊNCIAS BIBLIOGRÁFICAS	42
APÊNDICE A.....	45
A.1. Rotina exemplificando a adaptação das funções do sistema 1-Wire™ no código fonte.	45

LISTA DE FIGURAS

Figura 1 - Esquema de estrutura de rede que pode ser empregada para redes de comunicação.	5
Figura 2 – Topologia em Barramento.	6
Figura 3 - Topologia Estrela.	6
Figura 4 – Topologia em anel.	7
Figura 5 – Exemplo esquemático de uma rede baseada no protocolo TCP/IP.	9
Figura 6 – Comportamento de requisição-resposta do HTTP.	11
Figura 7 – Servidores com participação no mercado. Fonte: NETCRAFT- http://news.netcraft.com , em 09/11/2007.	12
Figura 8 – Esquema de forma generalizada da arquitetura <i>Cliente-Servidor</i> para aquisição dos dados do projeto e consulta do usuário.	21
Figura 9 – Diagrama de Blocos do modelo geral <i>Cliente-Servidor</i>	21
Figura 10 – Fluxograma do programa - Lado do Cliente.	22
Figura 11 – Fluxograma do programa – Lado do Servidor.	23
Figura 12 – Fluxograma da atividade básica <i>Cliente-Servidor</i>	24
Figura 13 – Aplicação <i>Cliente-Servidor</i> representada em três camadas.	25
Figura 14 – Diagrama do Modelo de Banco de Dados Relacional.	27
Figura 15 – Tela de Cadastro do Projeto.	30
Figura 16 – Tela de Acesso ao Usuário do Projeto.	30
Figura 17 – Tela de Visualização Principal - Anexos e Dados no SGBD.	31
Figura 18 – Exemplo da Relação Geral dos Dados no SGBD.	32
Figura 19 – Listagem de um arquivo em formato texto.	32
Figura 20 – Exemplo da listagem do arquivo em formato xls importado pela planilha eletrônica do OpenOffice.	33
Figura 21 – Exemplo da estrutura de arquivo XML.	33
Figura 22 – Exemplo da estrutura de arquivo XML aberto no MS-Excel.	34
Figura 23 – Visualização da tela de arquivos anexados referente ao projeto.	34
Figura 24 – Visualização da tela Anexar Arquivo.	35
Figura 25 – Execução do programa do Cliente do Projeto na linha de comando (<i>prompt</i>) do Windows XP.	35

Figura 26 – Execução do programa Servidor do Projeto no terminal Linux Ubuntu.	36
Figura 27 – Exemplo da tabela “Dados Armazenados” em SQL.	37
Figura 28 – Valores de temperatura coletados durante os testes.....	38
Figura 29 – Transmissão dos dados do Cliente do Projeto para o Servidor do Projeto no ambiente Windows XP.....	39
Figura 30 – Transmissão dos dados do Servidor do Projeto para o SGBD em ambiente Linux Ubuntu.	39

LISTA DE ABREVIATURAS E SÍMBOLOS

API	Interface para Programação de Aplicativos (Application Program Interface)
Applet	Programa escrito em linguagem Java para ser inserido numa página da <i>Web</i>
ASP	Linguagem de programação (Active Server Pages)
ER	Entidade Relacional
FTP	Protocolo de Transferência de Arquivo (File Transfer Protocol)
GPL	Licença para software livre (General Public License)
Hipermídia	Documento que contém imagens, vídeos, áudios e textos com ligações de hipertextos para permitir saltos de um trecho para outro do documento Sinônimo de multimídia. Termo criado por Ted Nelson (1960).
HTML	Acrônimo para "HyperText Markup Language" (Linguagem de Marcação de Hipertexto)
HTTP	Protocolo de Transferência de Hipertexto (hypertext transfer protocol)
IMAP	Protocolo de Acesso a Mensagens na INTERNET (Internet Message Access Protocol)
ISO	Organização Internacional de Padrões (International Standards Organization)
Java	Linguagem de programação orientada a objeto (Sun Microsystems)
JBC	Java Byte-Code
LAN	Redes Locais (Local Area Network)
MAN	Rede Metropolitana (Metropolitan Area Network)
OSI	Interconexão de Sistemas Abertos (Open Systems Interconnection).
PHP	Acrônimo recursivo para "PHP: Hypertext Preprocessor"
POP	Protocolo de Agência de Correio (Post Office Protocol)
Script	Linguagem de computador interpretada
SGBD	Sistema de Gerenciamento de Banco de Dados

SMTP	Protocolo Simples para Transferência de Correspondência (Simple Mail Transport Protocol)
Socket	Elo bidirecional de comunicação entre dois programas em uma rede de computadores.
SQL	Linguagem de Consulta Estruturada (Structured Query Language)
Tags	Estruturas de linguagem de marcação que consistem em breves instruções, tendo uma marca de início e outra de fim.
TCP/IP	Protocolo de controle da conexão na INTERNET (Transmission Control Protocol/INTERNET Protocol)
Telnet	Protocolo de simulação terminal com conexão a um computador através da INTERNET ou linha telefônica (Terminal Emulation Protocol)
Threads	Processo de execução da parte de um aplicativo
Unicode	Padrão de codificação de caracteres desenvolvido pelo Unicode Consortium
URL	Localizador Uniforme de Recursos (Uniform Resource Locator)
WAN	Redes Geograficamente Distribuídas (Wide Area Network)
Web	Rede de documentos em formato HTML que estão interligados (WWW)
XML	Linguagem de Marcação Extensiva (eXtensible Markup Language)

RESUMO

ROCHA, Keller Sullivan Oliveira, M.Sc., Universidade Federal de Viçosa, fevereiro de 2008. **Sistema Computacional Multiplataforma Para Armazenamento e Transmissão Remota de Dados**. Orientador: José Helvecio Martins. Co-Orientadores: Evandro de Castro Melo, Daniela de Carvalho Lopes.

Com o grande avanço tecnológico em relação aos sistemas de instrumentação e medidas, são exigidos, cada vez mais, sistemas que proporcionem confiabilidade aos resultados e respostas rápidas nas diferentes formas de transferência de dados. Neste contexto, desenvolveu-se um sistema que permite transferir, armazenar e recuperar dados de experimentos por acesso remoto por meio da INTERNET.

A utilização da arquitetura *Cliente-Servidor*, com o uso da INTERNET para armazenamento e transmissão de dados em Sistemas de Gerenciamento de Banco de Dados (SGBD), é considerada promissora devido à popularidade da Rede mundial de Computadores (*Web*). Sua construção dentro da idéia de código livre possibilita o seu crescimento de forma colaborativa entre os próprios usuários.

Atualmente, equipamentos micro-processados como *desktops*, *notebooks*, *palmtops*, telefones celulares e outros, provêm algum tipo de navegador *Web*, não necessitando da instalação de outros programas aplicativos de acesso à INTERNET, possibilitando o acesso remoto em diversos tipos de sistemas operacionais.

A utilização do Lado do Usuário da arquitetura do sistema na *Web*, conforme desenvolvida neste trabalho, é auto-explicativa. O acesso pelo usuário aos dados transmitidos e aos arquivos anexos torna-se simples e em ambiente amigável. A formatação desses dados nos diversos tipos de arquivos padrão como XML, xls e txt facilita a leitura e possibilita a portabilidade deles para análise. É possível anexar arquivos resultantes de outros sistemas que não possuem uma padronização comum.

É exigida também pouca capacidade de processamento nos equipamentos do usuário e do Cliente do Projeto, uma vez que todos os dados estão no servidor *Web* e somente são apresentados pelo navegador.

Um fator a ser observado é a velocidade de conexão do usuário que é variável, dependendo do seu tipo, interferindo no tempo de resposta da solicitação. Assim, o que pode prejudicar o tempo de resposta ao usuário é a qualidade de sua conexão à INTERNET.

A configuração do servidor *Web* e do Cliente do Projeto, por sua vez, exige, de certa forma, um conhecimento especializado em tecnologias da informação, para garantir o seu funcionamento. O servidor *Web* necessita de uma hospedagem com suporte à linguagem PHP e MySQL. O código fonte em Java, do lado do Cliente do Projeto, é disponibilizado para que o usuário possa adaptar o programa referente ao seu experimento às variáveis padrões de inclusão ao SGBD. Para isso, torna-se necessário um conhecimento básico de programação.

Muitas são as possibilidades de melhorias e propostas de acréscimos futuros ao sistema proposto. Dentre elas, citam-se: (i) comunicação entre os usuários do mesmo projeto pelo sistema; (ii) segurança e controle por comunicação móvel; (iii) estudar a viabilidade de implementação, no sistema de gerenciamento e armazenamento, de um sistema de controle de dispositivos por meio da INTERNET, criando um laboratório virtual, com o objetivo de aproveitar melhor os recursos já existentes. Assim, seria possível o desenvolvimento de ferramentas relacionadas à automação por meio da INTERNET, como ferramenta fundamental para implementação de controle de ambientes remotos.

Encontram-se na literatura alguns trabalhos sobre laboratórios virtuais, mas de caráter muito específico, tornando-se necessário o desenvolvimento de um sistema multiplataforma para a transmissão de dados que viabilize a comunicação remota, facilitando o processo de análise desses dados.

ABSTRACT

ROCHA, Keller Sullivan Oliveira, M.Sc., Federal University of Viçosa, February, 2008. **Multiplatform Computational System for Storage and Remote Data Transmission.** Adviser: José Helvecio Martins. Co-Advisers: Evandro de Castro Melo, Daniela de Carvalho Lopes.

As instrumentation and measurement system technology develops the demand for faster and more reliable data transfer increases. In this context, it was developed a system for remote management and data transmission over diverse communication interfaces. This goal was achieved by developing a protocol system that allows for remote storage, access, and data transmission by the INTERNET.

The utilization of the Client/Server architecture, including the use of the INTERNET for storage and transmission of data to the database systems are promising due to popularity of the Web. The software was developed using open source code, allowing for its collaborative growth among the users.

At present, micro-processed equipment such as desktops, notebooks, palmtops and cellular telephones has been used to access the INTERNET. That is, there is no need for installing other software to remote accessing of data. The user access to the collected data is simple and done in a friendly interface. The stored data can also be saved as XML, xls and txt files, making their management and analysis easier by the user. Project and server terminals can have low processing capacity since all data are stored at the Web server and are showed only by the browser.

An important factor that should be observed is that the speed of user connection varies according to its type and may interfere in the response time of the request

The Web server and project client settings should be done by technicians with basic knowledge about information technologies. This will guarantee that the system will work properly. The web server needs a host with PHP and MySql support. The Java font code is freely available and the user can adapt it to other software used in specific experiments by including new variables to the database. Also, for this task it is necessary logical programming knowledge. Future works should be developed to improve the system. Some suggestions are: (i) the inclusion of communication between users of the same project by the

system; (ii) security and process control over mobile communications; (iii) study of the viability of implementing a management and storage system to control the devices connected to the projects by INTERNET. This will create a virtual laboratory which can improve the use of the available resources.

Several works about virtual laboratory can be found in the literature, although they are for specific applications. Therefore, it is necessary the development of a multiplatform computational system for remote data transmission that turns it possible the remote communication facilitating the data analysis.

1. INTRODUÇÃO

O desenvolvimento tecnológico, com relação aos sistemas de instrumentação e medidas, exige, cada vez mais, sistemas que proporcionem maior confiabilidade aos resultados e respostas rápidas para as diferentes formas de transferência destes dados. A tecnologia, em geral, é uma catalisadora de mudanças que tem conduzido a novos sistemas de comunicação científica entre os pesquisadores (HURD, 2000).

Nesse contexto, o desenvolvimento de pesquisas em campo requer um elevado custo devido à necessidade presencial, o que demanda tempo do pesquisador no projeto de pesquisa durante a aquisição de dados, principalmente quando afetado pelo fator geográfico em pontos de coleta distintos.

A utilização de redes de comunicação torna possível o armazenamento e a análise de dados em tempo real, possibilitando ao pesquisador explorar mais extensamente as suas habilidades cognitivas. Dessa forma, os pesquisadores podem atuar remotamente na monitoração desses dados.

De acordo com Marino et al. (2000), a utilização de redes em práticas laboratoriais, baseadas em instrumentação programável, permite a interação entre pesquisadores e seus projetos. Esses sistemas podem ser utilizados por múltiplos usuários, dispensando suas presenças físicas em campo, independentemente da sua localização geográfica, formando sistemas de gerenciamento remoto de coleta e transmissão de dados.

Uma rede de computadores é uma infra-estrutura que interliga o acesso a recursos remotos entre dois ou mais dispositivos compartilhando dados e, ou, processamentos entre si. Técnicas de comunicação de dados são desenvolvidas para uso deste meio. O exemplo mais comum desta forma de comunicação é a conexão de terminais de computadores por meio de modems (DUNLOP e SMITH, 1994), porém existem meios de comunicação com base em cabos coaxiais, pares trançados, fibra ótica, rádio frequência e por meio de satélite.

Para que a comunicação entre dois ou mais computadores seja possível, é necessário que exista um mesmo canal lógico de comunicação. Esse canal de comunicação é chamado de protocolo. As transferências de dados devem

ser feitas por meio de redes que utilizam protocolos padrões (BENTHAM, 2002).

Dessa forma, em dois ou mais dispositivos que possuem processadores, barramentos ou sistemas operacionais diferentes, mas que tenham um protocolo em comum, é possível realizar transferência de dados. Trocas de mensagens e ações podem ser realizadas por meio de hardware ou programa computacional que pode ser um roteador, um computador ou qualquer outro dispositivo habilitado para a rede.

Um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão e, ou, no recebimento de uma mensagem ou outro evento (KUROSE, 2005). Um arranjo topológico interliga vários módulos de processamento por meio de enlaces físicos dos meios de transmissão que são organizados, pelos protocolos, em um conjunto de regras nos diversos tipos de rede, sejam elas redes locais (LAN's), metropolitanas (MAN's) ou geograficamente distribuídas (WAN's), cabeadas ou com a tecnologia sem fio.

Dentre os protocolos mais comumente utilizados em redes de comunicação de dados, cita-se o TCP/IP (Transmission Control Protocol/ INTERNET Protocol). Esse protocolo provê serviços de comunicação que ocorrem entre a interface de programação de uma rede física e as aplicações do usuário. Ele habilita uma interface comum para estas aplicações, independentemente de sua rede física (RODRIGUES et al., 2001). Com o protocolo TCP/IP, é possível transferir ou receber dados por meio da INTERNET, INTRANET e EXTRANET.

A INTERNET é uma organização complexa de redes administradas por companhias que permite acesso a recursos internacionais, enquanto a INTRANET permite o acesso a um ambiente local e a EXTRANET o acesso entre diversas redes do tipo INTRANET (NAUGLE, 1998).

A escolha da topologia da rede caracterizará o seu tipo, a sua eficiência, a sua velocidade, o seu custo, a sua confiabilidade e a sua estabilidade indicando a forma como os enlaces físicos e os nós serão organizados nos diferentes tipos de rede. Em geral, topologias híbridas são empregadas para facilitar a transmissão de dados.

Nesse contexto, pode-se utilizar a arquitetura *Cliente-Servidor*, onde o cliente executa aplicações utilizando dos servidores para transferir os dados ao banco de dados por meio do mesmo tipo de protocolo de acesso.

Em um ambiente *Cliente-Servidor*, aplicações de clientes usam uma rede privada ou pública para transportar mensagens para um servidor central de aplicação para a obtenção dos dados exigidos (CHINNAPPEN-RIMER e HANCKE, 2003). Dessa forma, nas aplicações de tecnologia *Web* (rede de alcance mundial, *www*), que consiste em um sistema de documentos em hipermídia interligados e executados na INTERNET, tem-se uma solução viável para a integração de diversas mídias educacionais.

Devido à flexibilidade da rede *Web*, ela permite uma grande variedade de informações, possibilitando a disponibilidade do material resultante em plataformas e sistemas operacionais diferentes, independentemente do tipo de operações a serem realizadas na INTERNET (DOULAI, 1996).

O conceito de sistemas de gerenciamento remoto de transmissão de dados envolve ambientes interativos com respostas rápidas, agilizando a pesquisa e estimulando o pesquisador no seu desenvolvimento, independentemente do espaço ou do tempo. Assim, a construção de uma ferramenta, com essas características, que possa ser utilizada por pesquisadores nas diversas áreas do conhecimento, irá contribuir significativamente para a pesquisa científica.

A aplicação de um sistema de gerenciamento e monitoramento remoto na engenharia agrícola busca atender as necessidades de otimização dos processos, como ferramenta de apoio ao desenvolvimento de atividades que exijam segurança, confiabilidade e velocidade nos sistemas de informação em diversos projetos. Com isso, o sistema proposto é de baixo custo e auxiliará o pesquisador em seus experimentos, especificamente na aquisição de registro de dados obtidos a partir de sensores de medição de variáveis.

Encontram-se na literatura alguns trabalhos sobre laboratórios virtuais, mas de caráter muito específico, tornando-se necessário o desenvolvimento de um sistema multiplataforma para a transmissão de dados que viabilize a comunicação remota, facilitando o processo de análise desses dados.

2. OBJETIVOS

O presente trabalho foi realizado com o objetivo geral de desenvolver um sistema de gerenciamento remoto de transmissão de dados por meio de diferentes interfaces de comunicação.

Os objetivos específicos foram:

- 2.1.** Desenvolver um sistema que, por meio de protocolos, permita o acesso e a transmissão remota de dados.
- 2.2.** Desenvolver um modelo de banco de dados para o sistema computacional proposto.
- 2.3.** Validar o sistema computacional proposto, utilizando-se de dados obtidos em testes de simulação e projetos implementados.

3. REVISÃO DE LITERATURA

3.1. AS REDES DE COMPUTADORES E A INTERNET

Uma rede de computadores é formada por componentes capazes de compartilhar recursos e informações por um meio comunicante. A estrutura básica de uma rede de computadores está representada na Figura 1. Esta rede interliga seus vários componentes por meio de ligações físicas dos meios de transmissão e conjunto de regras necessárias para estabelecer a comunicação, denominados protocolos.

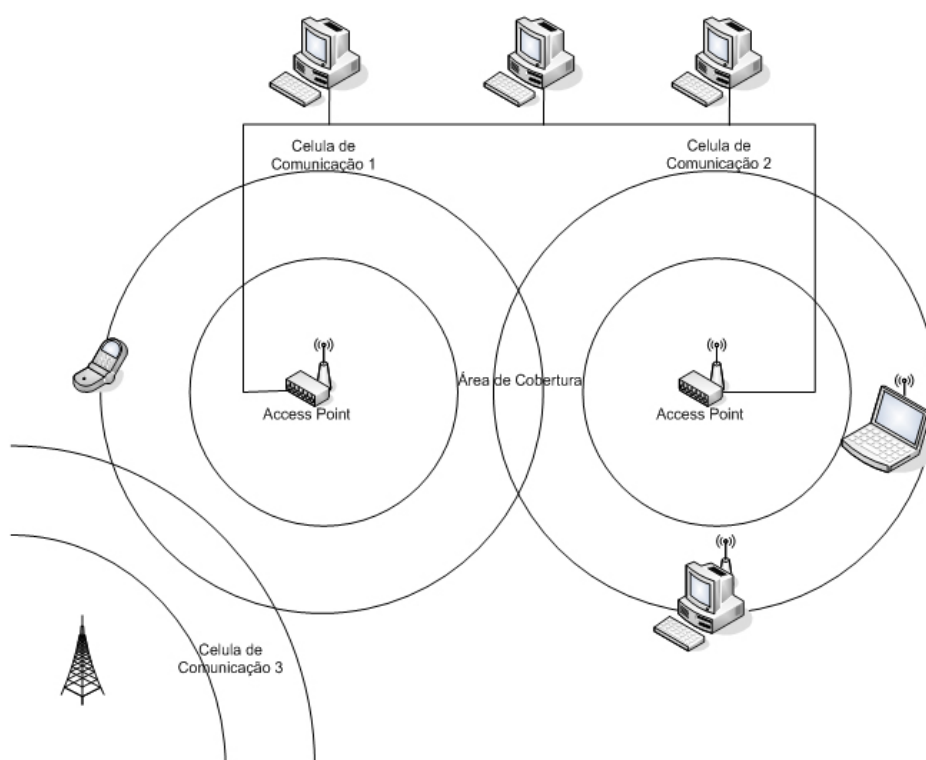


Figura 1 - Esquema de estrutura de rede que pode ser empregada para redes de comunicação.

A estrutura de rede determina muitos fatores importantes tais como custo operacional, custo estrutural e confiabilidade da comunicação da rede (KAMIYAMA, 2006).

Uma rede pode ser constituída por vários tipos de topologia. Essas topologias consistem de conexões físicas entre sistemas. A comunicação dessa estrutura com a rede é feita por meio de uma linguagem comum entre elas como, por exemplo, o protocolo TCP/IP. Esses diferentes tipos de topologias são descritos a seguir.

3.1.1. Topologia em barramento (bus)

Nesse tipo de topologia, como mostrado na Figura 2, os dispositivos são conectados a um barramento por um cabo coaxial central, apresentando a sua configuração em multiponto, com limitações em número de nós e distância entre os dispositivos. Cada nó pode “perceber” as informações transmitidas na rede.

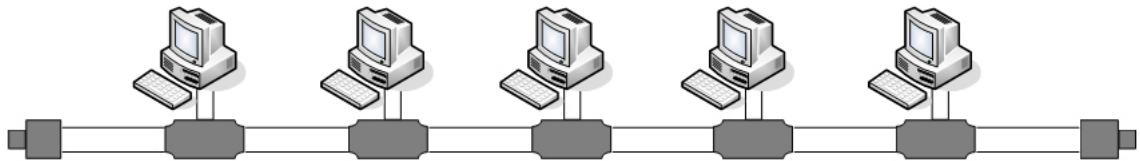


Figura 2 – Topologia em barramento.

3.1.2. Topologia em estrela

Nesse tipo de topologia, como mostrado na Figura 3, os dispositivos são ligados em um único nó central.

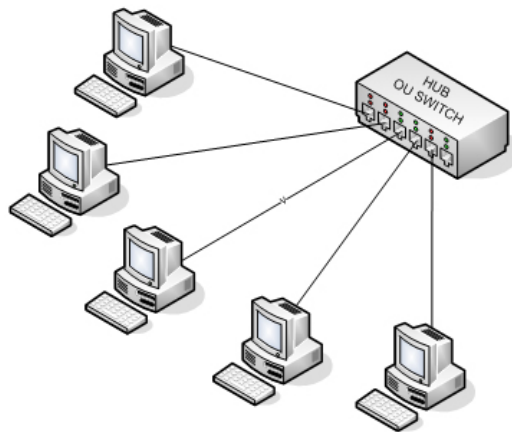


Figura 3 - Topologia estrela.

3.1.3. Topologia em anel

Nesse tipo de topologia, a rede é composta por estações conectadas a um cabo que percorre um caminho fechado, como mostrado na Figura 4.

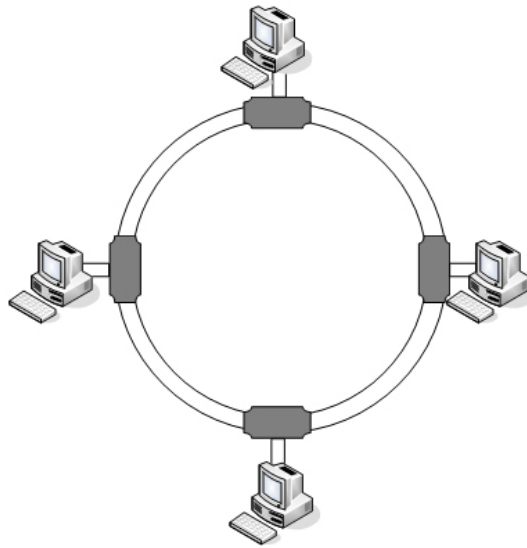


Figura 4 – Topologia em anel.

3.1.4. A INTERNET

A INTERNET foi fundada em 1969 com o projeto nomeado ARPANET, desenvolvida pela empresa ARPA (Advanced Research and Projects Agency) nos Estados Unidos. O protocolo TCP/IP foi estabelecido como protocolo da INTERNET, em meados de 1970, e foi adotado como padrão pelo Departamento de Defesa Norte Americano em 1980. Com isso, o objetivo principal foi administrar pesquisas e tecnologias nas interconexões de informações compartilhadas entre as agências do governo e as universidades e centros de pesquisas. Por volta de 1980, a INTERNET começou a ser utilizada para fins comerciais nos Estados Unidos.

Segundo INTERNET USAGE STATISTICS (2007), a INTERNET é utilizada hoje por mais de 1,1 bilhões de pessoas, o que representa 17% da população mundial.

3.1.5. Tipos de Rede

As redes são divididas em três tipos: (i) LANs; (ii) MANs e (iii) WANs. A seguir apresenta-se uma breve descrição de cada uma delas.

3.1.5.1. Redes LANs

As redes LANs permitem a interconexão de dados em um raio máximo de 10 km.

3.1.5.2. Redes MANs

As Redes MANs possuem características semelhantes às redes LANs, mas cobrem distâncias maiores e com velocidades superiores de transmissão de dados.

3.1.5.3. Redes WANs

As Redes WANs têm a característica de cobrir uma grande área geográfica.

3.1.6. Protocolos

Os diversos dispositivos de uma rede podem se intercomunicar por meio de uma linguagem comum, a qual é denominada Protocolo de comunicação. Para que esta comunicação seja possível, todos os dispositivos devem utilizar o mesmo protocolo.

O conjunto de protocolos TCP/IP é um dos mais utilizados para a transmissão de dados em redes de comunicação em alta velocidade (TANAMACHI et al., 2004).

O TCP/IP é um conjunto de protocolos que foi projetado para ser robusto, roteável, e funcionalmente eficiente, representando praticamente toda a comunicação da comunidade da INTERNET. As suas principais características são:

- É um conjunto de protocolos de padrão-industrial desenvolvido pelo Departamento de Defesa Americano. As definições das novas características que são publicadas para a INTERNET são detalhadas e disponibilizadas para uso.

- Permite vantagens de conexão por sistemas operacionais diferentes, como a utilização de FTP e TELNET. A conectividade não depende do sistema operacional que é usado em qualquer máquina.
- Utiliza interface por *sockets*, onde o Windows API prove desenvolvimento de uma interface padrão para o desenvolvimento de aplicações *Cliente-Servidor*. Todas as aplicações TCP/IP usam a interface *socket* entre aplicações e o protocolo de rede.
- Possibilita acesso a uma gama de informações, por meio da INTERNET, de qualquer lugar do mundo.

Todo dispositivo ou computador conectado a uma rede recebe um endereço numérico, chamado endereço IP. Estes números podem ser atribuídos de forma estática pelo usuário ou dinamicamente pelo servidor, quando o usuário se conecta à rede.

Na Figura 5, tem-se um exemplo esquemático de uma rede baseada no protocolo TCP/IP.

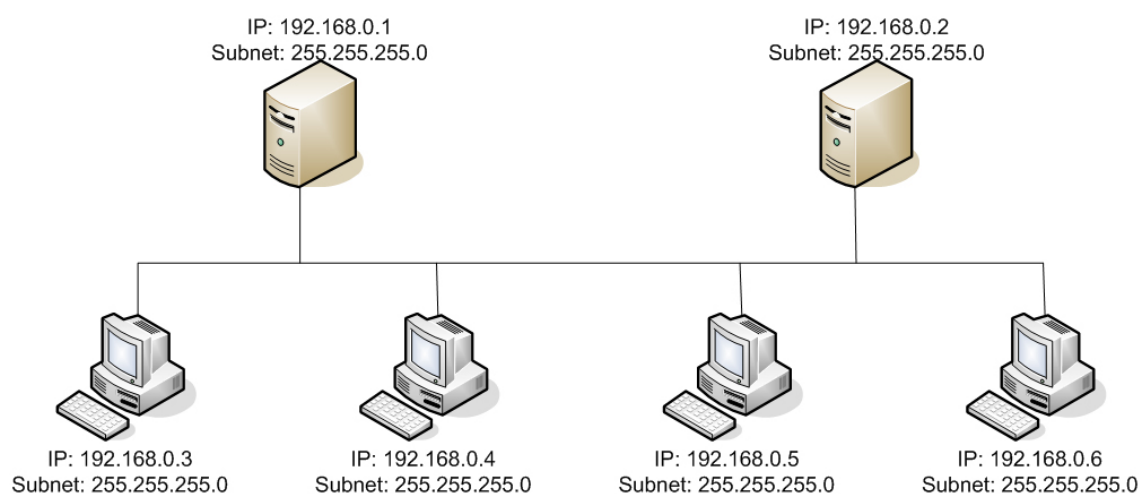


Figura 5 – Exemplo esquemático de uma rede baseada no protocolo TCP/IP.

3.1.7. Arquitetura *Cliente-Servidor*

Segundo Furmston-Evans (1995), a arquitetura *Cliente-Servidor* é definida em termos de várias estações de trabalho ou dispositivos microprocessados, que acessam servidores por meio de uma rede de computadores.

As aplicações *Cliente-Servidor* compreendem dois ou mais componentes, onde a aplicação cliente solicita ou envia informações a uma aplicação central, chamada servidor; e este, por sua vez, processa e devolve os resultados ao cliente (DUO et al., 2002). Algumas das principais aplicações *Cliente-Servidor* consistem em: (i) aplicações servidor e (ii) aplicações cliente.

3.1.7.1. Aplicações Servidor

Consistem em sistemas gerenciadores de banco de dados (SGBD), servidores de arquivos e impressão, servidores de comunicação e servidores de *e-mail*.

3.1.7.2. Aplicações Cliente

Consistem em interface entre o usuário e o SGBD, consulta ao banco de dados, ferramentas de apresentação e análise de informações.

O processamento é estruturado e distribuído com base no conceito de serviços. O Servidor atende diversos clientes permitindo o compartilhamento de recursos. A comunicação é por trocas de mensagens entre cliente (solicitando) e servidor (atendendo). Os serviços implementados pelo servidor não são mostrados ao cliente, havendo assim encapsulamento do serviço, apresentando somente a interface para solicitação da resposta e o resultado final (BAI e WEI, 2001). Dessa forma, o termo “*Cliente-Servidor*” é usado em termos de microcomputadores em uma rede de computador.

3.1.8. HTTP e Serviços Web

Por meio do protocolo TCP/IP, houve crescimento significativo de usuários na INTERNET que interagem por diferentes softwares e plataformas. Assim surgiu o modelo de aplicação *Cliente-Servidor*.

A *Web* tem a característica de funcionar por demanda, dispositivos gráficos, formulários, *applets Java* e outros aplicativos que facilitam a interação

com páginas e sites. A *Web* fornece interfaces para materiais em hipermídia, por ser uma tecnologia constituída por objetos que são interligados e executados na INTERNET.

O protocolo de aplicação da *Web*, o HTTP, é implementado na arquitetura *Cliente-Servidor*, sendo um protocolo da camada de aplicação do modelo OSI, que é um conjunto de padrões ISO relativo à comunicação de dados, utilizado para transferência de dados na INTERNET. Um navegador (browser) é um programa de usuário para *Web* que implementa o lado cliente do HTTP, habilitando o acesso de arquivos HTML que estão hospedados em servidores *Web*.

O protocolo HTTP utiliza o protocolo TCP para transferência de dados. O lado cliente do HTTP inicializa a conexão TCP com o servidor *Web* por meio de interfaces denominadas de *sockets*. O lado do cliente HTTP envia a requisição e recebe uma resposta na interface *socket*, e da mesma forma o lado servidor HTTP recebe a requisição e envia a mensagem de resposta para a interface *socket*, como mostrado na Figura 6.

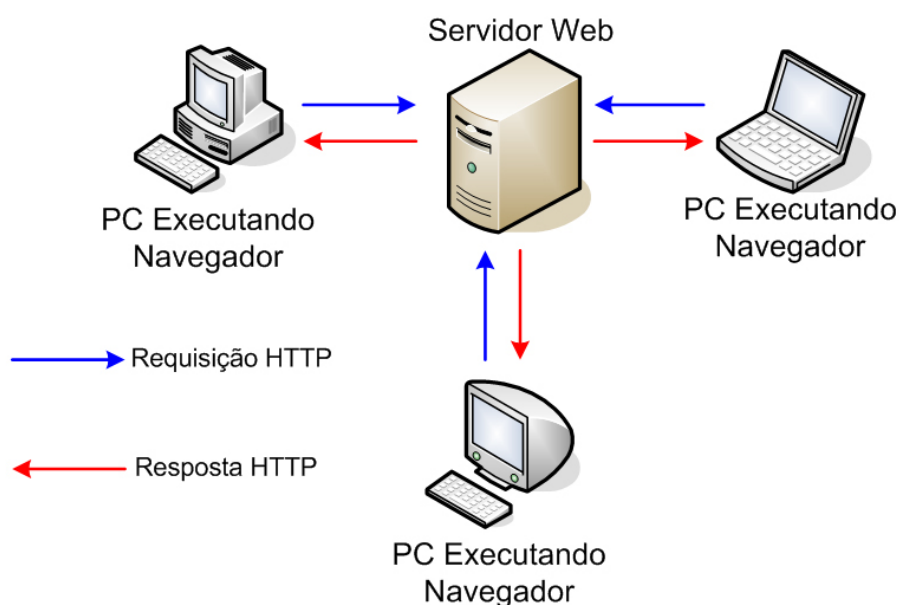


Figura 6 – Comportamento de requisição-resposta do HTTP.

O lado servidor do HTTP é implementado pelos servidores *Web*, onde abrigam os objetos que estão endereçados por um URL, e cada URL é formada por dois componentes: (i) o nome do hospedeiro do servidor que

abriga o objeto e (ii) o nome do caminho do objeto. Assim, o HTTP define como as partes *Cliente-Servidor* se comunicam.

O Servidor Apache (Apache Server) é um dos mais utilizados servidores *Web* de domínio público, criado por Rob McCool. Como mostrado na Figura 7, a sua utilização supera mais de 60% dos servidores no mundo (WIKIPÉDIA, 2007; NETCRAFT, 2007).

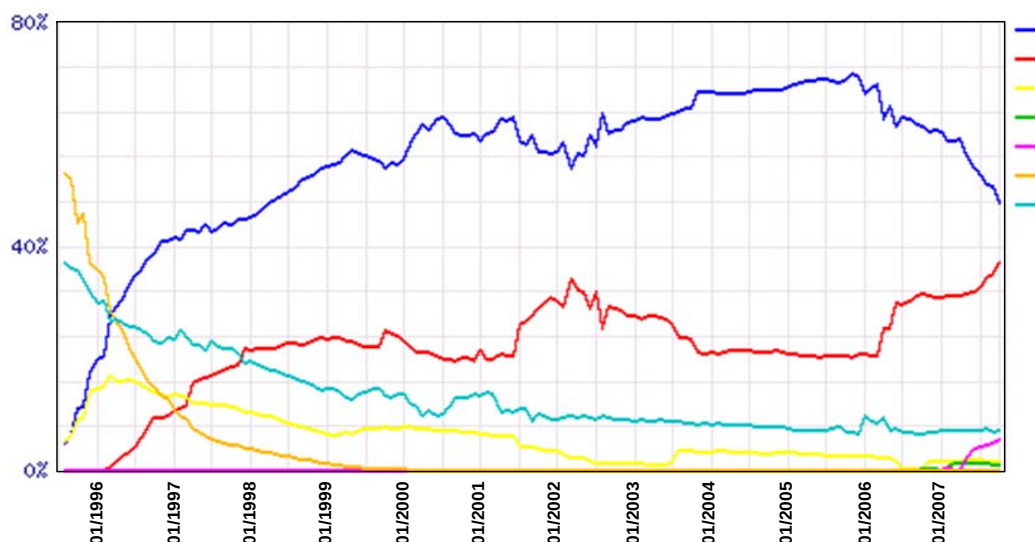


Figura 7 – Servidores com participação no mercado.

O Apache é um servidor com arquitetura modular que aceita inclusão, praticamente, de qualquer característica desejada de um servidor *Web*. Isto o torna flexível, atendendo às especificações para cada tipo de site.

3.1.9. Sites Dinâmicos

Páginas escritas em HTML são estáticas, limitando o Servidor *Web* a enviar apenas aquelas previamente construídas para o navegador. Estas páginas não variam, à exceção das atualizadas manualmente. Os usuários não podem interagir significativamente com as páginas, portanto utilizam-se de recursos que consistem na inserção de pequenos programas denominados *scripts* nos documentos HTML, para a sua transformação em páginas dinâmicas.

Esses *scripts* são escritos em linguagens de programação como PHP, JavaScript, ASP e outras, adequadas para esse fim, que suportam o uso de

Banco de Dados ou formas de armazenamento de dados específicos para o desenvolvimento de páginas ou aplicações para INTRANET e EXTRANET. Esses *scripts* permitem criar páginas dinâmicas susceptíveis a personalização e dotadas de informação em tempo real (WELLING e THOMSON, 2005).

As formas de geração de páginas dinâmicas podem ser do lado-cliente ou do lado-servidor.

Na geração do lado-cliente, os *scripts* são enviados pelo servidor junto aos códigos HTML e são interpretados e executados pelo navegador. Podem ser citados alguns *scripts* do lado-cliente tais como o *Applets Java*, *Java Script*, dentre outros.

Na geração do lado-servidor, o próprio servidor *Web* interpreta e executa os *Scripts*. O resultado é inserido na página e enviado ao navegador em formato HTML. Dessa forma, qualquer navegador é capaz de exibir a página, uma vez que os *scripts* são interpretados pelo servidor *Web*. As vantagens da geração de páginas no lado-servidor são:

- Minimizar o tráfego da rede.
- Reduzir o tempo de carregamento.
- Permitir que o conteúdo da página seja carregado pelo navegador.
- Evitar problemas de incompatibilidade por navegadores.
- Possibilitar medidas de segurança mais eficientes.

3.2. LINGUAGENS DE PROGRAMAÇÃO

3.2.1. Hypertext Markup Language - HTML

Muitas páginas *Web* possuem conteúdo estático. Essas páginas consistem em documentos simples de texto e hipermídia que, para alguns tipos de páginas simples geradas do lado-cliente, torna-se suficiente serem geradas utilizando-se somente o HTML. A página é transferida de um computador remoto para o navegador, no qual é feita a interpretação dos códigos no documento.

Quando uma requisição a uma página é feita pelo navegador, o servidor *Web* lê a requisição, procura a página e envia a resposta de volta para o navegador por meio da INTERNET. Porém, com a crescente utilização de

aplicações na INTERNET e, ou, INTRANET que incorporam bancos de dados e outras operações, é necessário o uso de *scripts* para que possam ser gerados conteúdos dinâmicos.

3.2.2. Hypertext Preprocessor - PHP

A linguagem de programação PHP é uma linguagem interpretada de ampla utilização que possibilita ser mesclada aos códigos HTML. Sua execução é exclusiva no servidor, permitindo aos desenvolvedores escrever páginas dinâmicas (ACHOUR et al., 2005).

PHP é uma linguagem de código fonte aberto, muito utilizada na INTERNET, criada para o desenvolvimento de aplicações *Web*, e portátil para uma grande quantidade de sistemas operacionais diferentes. O seu código funciona sem necessidade de modificação nos sistemas que o executam.

Em PHP é possível utilizar variáveis com formulários HTML, conferindo a eles a capacidade de gerar conteúdo por demanda. Uma de suas características é a de não ser necessária a declaração de variáveis antes da sua utilização, pois estas são automaticamente declaradas ao atribuí-las um valor. PHP permite suporte a uma grande variedade de banco de dados.

Segundo Schlossnagle (2004), os padrões de acesso ao banco de dados definem como será a interação do SGBD com o código PHP. Assim, o desenvolvimento de aplicações em PHP requer o uso de outras tecnologias como protocolos HTTP e linguagens como HTML e SQL.

PHP permite suporte a XML, protocolos de e-mail como IMAP, SMTP e POP, criação de documentos em formatos Pdf e PostScript, quando usada uma API apropriada.

3.2.3. Linguagem Java

A linguagem Java foi desenvolvida objetivando aplicações voltadas para produtos eletrônicos de grande consumo. Com o crescente uso da INTERNET e das páginas *Web*, cresceu também o interesse na aplicação de Java com

Applets, proporcionando animação e interatividade aos documentos estáticos em HTML.

Java é uma linguagem eficiente orientada a objetos, podendo servir a vários propósitos, pouco exigente em memória e podendo ser executada em equipamentos de pouca capacidade de processamento, sendo ela compilada para instruções em Java Byte Code (JBC). O JBC é o meio de interligação entre o código fonte e a aplicação final, interpretado e executado por uma *Máquina-Virtual* (BERNAT et al., 2000). Assim, Java foi criada para ser portátil, já que o JBC gerado pelo compilador para uma aplicação pode ser transportado entre plataformas distintas, sem a necessidade de recompilar o programa.

Segundo Waldron (1999), Java vem crescendo desde 1996, sendo particularmente interessante devido ao fato de usar uma linguagem intermediária, a JBC, para representar os programas, de forma que o mesmo programa possa ser executado em diversas máquinas com configurações diferentes.

Java oferece facilidades em programação com *sockets*, TCP/IP e outros protocolos, além de permitir a criação de múltiplos processos, podendo ser mapeado em processamento paralelo.

3.2.4. Structured Query Language - SQL

No início dos anos 70, a SQL foi desenvolvida pela IBM como forma de interface para SGBD relacional. Depois de 1986, a ANSI (American National Standard Institute) e a ISO (International Standard Organization) publicaram um padrão SQL. Assim, a SQL é uma linguagem de pesquisa declarativa para bancos de dados relacionais, em que muitas das suas principais características foram formadas a partir da Álgebra Relacional. Declarações em SQL executam tarefas como inclusão, atualização ou recuperação dos dados em um SGBD.

Segundo Brass e Goldberg (2005), uma grande porcentagem de programas é desenvolvida usando-se SGBD relacionais, em que, normalmente, os dados são acessados em linguagem SQL.

A SQL é composta por comandos, cláusulas, operadores e funções agregadas. Esses elementos, juntamente com as instruções da linguagem,

permitem criar, atualizar e manipular o banco de dados. Com a SQL pode-se ter:

- Criação de Bases de Dados.
- Segurança de acesso aos Dados.
- Recuperação das informações sem perda da integridade.
- Manipulação e controle das bases de dados.

A SQL é uma linguagem padrão para comunicação com SGBD, que permite trabalhar em diversas linguagens de programação.

3.2.5. eXtensive Markup Language - XML

A XML facilita a troca de informações entre aplicações, provendo funcionalidades de administração e integridade dos dados entre SGBD distintos e aplicação na INTERNET. É uma linguagem formada por *marcadores de início e fim de estruturas (tags)*, semelhante ao HTML, com suporte a um número maior de *marcadores de início e fim de estruturas* definidos pelo usuário.

Por ser definida como um padrão universal para dados estruturados na *Web*, a XML é uma linguagem que possibilita regras de escrita com alto nível de interpretabilidade. A XML não é uma tecnologia, mas um padrão que serve como um meio para descrever comunicações entre sistemas de informação. É entendida em um contexto de paradigma na *Web*. Assim, como as práticas de tecnologias da informação induzem aos grandes avanços da *Web*, este paradigma rompe barreiras existentes entre as diferentes plataformas e as aplicações de manipulação de dados, com alto nível de interpretabilidade por sistemas heterogêneos.

A expressividade de sistemas baseados nesta linguagem tem a importância de associar o significado real de um texto simples e, assim, facilitar a administração do conhecimento.

Segundo Harold (2001), a XML é constituída por um conjunto de regras que define a semântica dos *marcadores de início e fim de estruturas (tags)* que dividem o documento, identificando suas diferentes partes. Dessa forma, podem-se criar os *marcadores de início e fim de estruturas (tags)*, quando

necessários, definindo para o navegador como mostrar estes *marcadores de início e fim de estruturas (tags)*.

É uma linguagem que possui portabilidade, proporcionando exportação entre bases de dados distintas. Assim, a sua filosofia pode ser definida por:

- Separação do conteúdo da formatação.
- Simplicidade e legibilidade.
- Criação ilimitada de *marcadores de início e fim de estruturas (tags)*.
- Criação de arquivos para validação de estrutura.
- Interligação de bancos de dados distintos.

A XML pode conter e organizar qualquer tipo de informação de acordo com a necessidade do usuário. Como é um padrão aberto, não é vinculado a qualquer empresa.

A definição padrão de caractere da XML é o UNICODE, suportando um grande número de sistemas de escrita e símbolos. Essa linguagem oferece muitos modos de conferência da qualidade, como regras de sintaxe, verificação e comparação dos modelos de documentos. Com sua sintaxe clara, simples e estrutura inequívoca é fácil de ser lida e analisada gramaticalmente por seres humanos e por programas computacionais semelhantes.

A idéia central da XML, segundo Zhang et al. (2001), é permitir ao usuário criar qualquer documento com as estruturas de sua preferência e, assim, poder compartilhá-los com outras pessoas, facilitando a portabilidade das diferentes estruturas de documentos e a comunicação entre os profissionais.

3.3. Gerenciamento Remoto de Experimentos

O gerenciamento remoto permite o acesso a experimentos em tempo real, que seja possível a qualquer momento, e que possa ser compartilhado por várias pessoas, mesmo que estas estejam em pontos geograficamente distintos.

Um sistema de gerenciamento remoto pode ser definido como um ambiente interativo que pode ser utilizado para criar e administrar experiências (ALBU et al., 2004).

Segundo Roberts et al. (2004), os sistemas de gerenciamento remoto são caracterizados por sua versatilidade e flexibilidade por ter a sua base em programas computacionais. As suas vantagens são baixo custo, segurança e disponibilidade, pelo fato de poderem ser executados a qualquer hora e em qualquer lugar e por várias pessoas ao mesmo tempo. A padronização dos dados torna possível a sua apresentação em uma variedade de modos, potencializando a aprendizagem e gerando gravações dos dados coletados e de resultados.

3.4. Trabalhos Relacionados

Pizarro et al. (2001), realizaram um trabalho sobre telemedicina para o monitoramento de sinais bioelétricos de pacientes internados ou em casa. Os resultados mostraram que a análise dos dados pode ser feita por um médico em tempo real por meio de um navegador *Web* ou de uma aplicação de visualização que se comunica com o servidor.

Segundo Melo et al. (2006), o surgimento das técnicas de sensoriamento remoto e o desenvolvimento da cartografia assistida por computador possibilitam ao geógrafo trabalhar um conjunto de informações cada vez maior. Discutiram a possibilidade de um atlas digital interativo baseado em Sistemas de Informações Geográficas (SIG) voltado para o ensino e aprendizagem da geografia.

Santana et al. (2003) desenvolveram um sistema de regionalização da demanda de irrigação suplementar das principais culturas irrigadas no estado de Minas Gerais, disponibilizando os dados na rede mundial de computadores.

Bonho et al. (2006) desenvolveram um programa computacional que permite a recepção e a visualização de eletrocardiograma de pacientes em tempo real transmitidos por meio da INTERNET. Esse programa computacional permite também o cálculo da frequência cardíaca e de parâmetros relativos à análise da qualidade da transmissão dos sinais do eletrocardiograma.

Vargas (2004) desenvolveu um trabalho sobre a transmissão de dados através da rede elétrica, visando a sua utilização na área de automação residencial e predial. Foi mostrada, nesse trabalho, a viabilidade da tecnologia, embora seja difícil sua utilização em aplicações que demandem tempos de respostas determinísticos, devido à variabilidade nas taxas de transmissão.

4. METODOLOGIA

O presente trabalho foi desenvolvido no Setor de Energia na Agricultura, do Departamento de Engenharia Agrícola, da Universidade Federal de Viçosa, Minas Gerais.

O trabalho foi elaborado em três etapas: (i) Desenvolvimento do modelo de comunicação, (ii) Desenvolvimento do banco de dados e (iii) Testes de validação do modelo computacional.

O sistema foi implementado utilizando-se as linguagens de programação Java, versão 1.5.0.11 para o acesso *Cliente-Servidor*, PHP versão 5.2.1 e o banco de dados MySql versão 5.0.38 para o acesso e consulta pela *Web*, sendo denominado STRADA (Sistema de Transmissão de Dados).

O modelo de comunicação e o banco de dados foram desenvolvidos de acordo com diagramas de blocos, fluxogramas, esquemas e estruturas para apresentação nesse trabalho e foram construídos e modelados utilizando-se o programa computacional MS-Visio da Microsoft, versão 2003. A construção dos Scripts em PHP, Java e HTML foi realizada utilizando-se o programa de domínio público NotePad++, versão 4.1.2.

Para testes do sistema proposto, foi utilizado o Servidor *Web* Apache 2.2.3, no Sistema Operacional LINUX UBUNTU versão 7.04.

4.1. DESENVOLVIMENTO DE UMA PROPOSTA DE COMUNICAÇÃO

Na implementação do modelo de comunicação para a INTERNET utilizou-se a arquitetura *Cliente-Servidor*, mostrada de uma forma geral na Figura 8 e na forma de diagrama de blocos, apresentada na Figura 9, onde é representado o sistema computacional e como cada parte integrante interage com os demais componentes.

Pelo esquema da estrutura do sistema é possível identificar, de uma forma geral, as partes envolvidas como experimentos, sensores com os terminais, clientes, servidor e banco de dados interligados à INTERNET.

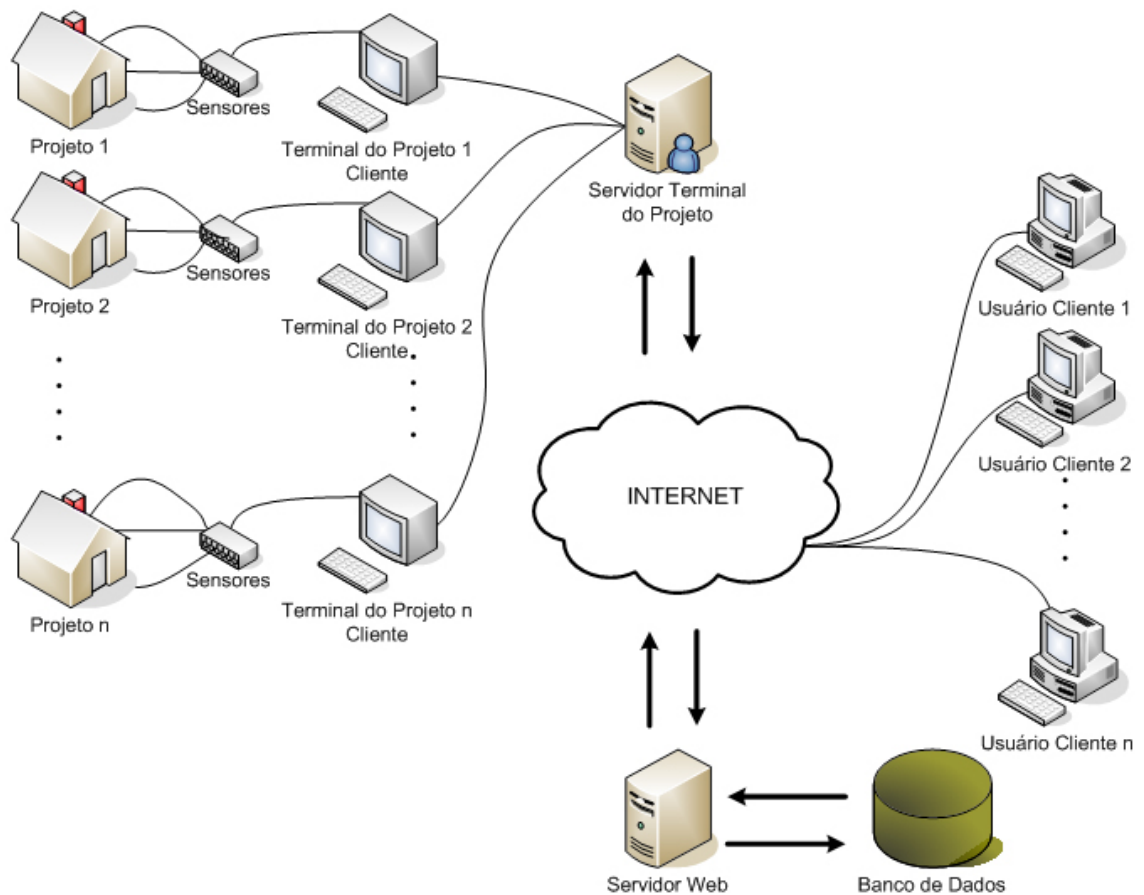


Figura 8 – Esquema de forma generalizada da arquitetura *Cliente-Servidor* para aquisição dos dados do projeto e consulta do usuário.

Assim, de uma forma hierárquica mostrada no diagrama de blocos da Figura 9, podem ser visualizados os caminhos de conexão entre a arquitetura *Cliente-Servidor* por meio do protocolo TCP/IP.

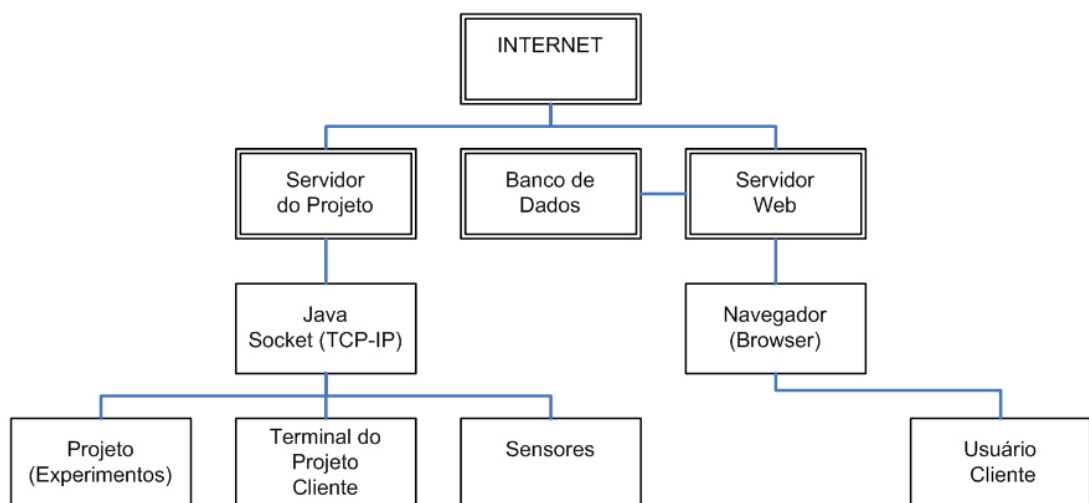


Figura 9 – Diagrama de blocos do modelo geral *Cliente-Servidor*.

Os usuários conectados ao servidor *Web* pelo navegador acessam a INTERNET, podendo cadastrar novo projeto ou visualizar cada informação contida no SGBD referente ao projeto previamente cadastrado.

Os sensores localizados no projeto enviam os sinais aos Terminais do Projeto que, por sua vez, comunicam por *sockets* com o Servidor do Projeto, incluindo essas informações no SGBD por meio da INTERNET.

O protocolo TCP/IP foi utilizado devido à sua segurança na transmissão dos dados pela rede de comunicação e por ser o protocolo padrão de comunicação na INTERNET.

Dessa forma, há interconexão por meio da INTERNET, entre as redes do lado do projeto e todos os dados adquiridos no experimento, o SGBD com as informações do usuário previamente cadastrado, e o acesso do lado dos usuários que são gerenciados por um modelo relacional.

4.1.1. Terminal do Projeto - Cliente

O Terminal do Projeto, como detalhado na Figura 10, comunica-se com o Servidor do Projeto por meio de *sockets* enviando-lhe os dados. O Terminal do Projeto estabelece a conexão com o novo processo criado pelo servidor, iniciando o processo de transferência dos dados.

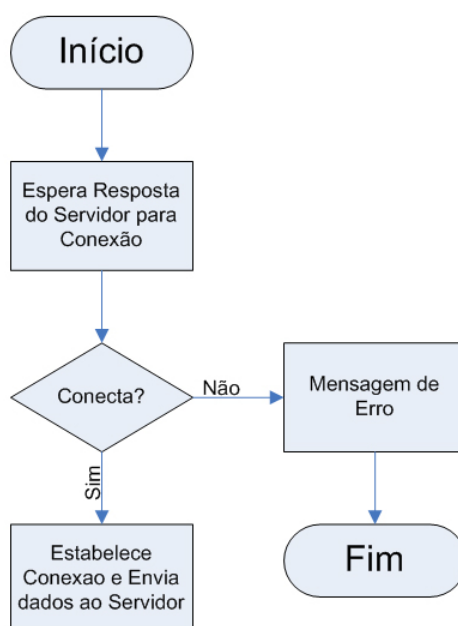


Figura 10 – Fluxograma do programa - *Lado do Cliente*.

Para a comunicação do cliente com o servidor, foi utilizada a linguagem Java e *socket* em modo de Orientação a Conexão, por funcionar com o protocolo TCP/IP, garantindo a ordem dos dados enviados e a confiabilidade na transmissão dos dados.

4.1.2. Servidor do Projeto

O Servidor do Projeto, como detalhado na Figura 11, está sempre à espera de uma comunicação do Terminal do Projeto. O Servidor do Projeto comunica-se com este por meio de *sockets*, recebendo os dados enviados pelo Terminal do Projeto, onde é feita a aquisição dos dados, transferindo-os para o Banco de Dados.

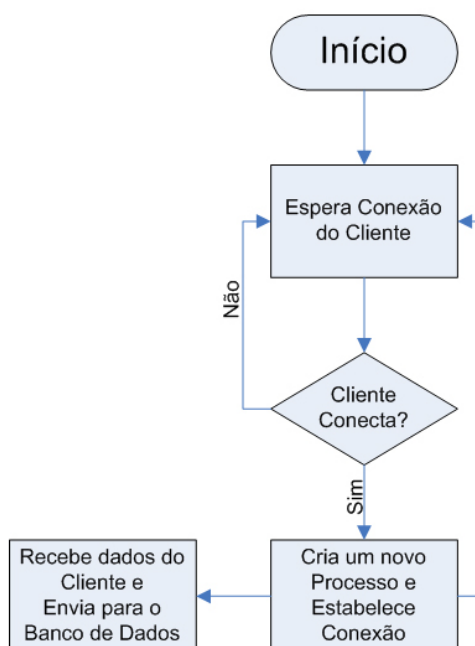


Figura 11 – Fluxograma do programa – *Lado do Servidor*.

A conexão entre o Terminal do Projeto e o Servidor do Projeto é mostrada na Figura 12. Como pode ser observado, tanto o cliente como o servidor tem a atividade básica de transmissão de dados do Terminal do Projeto para o Servidor do Projeto.

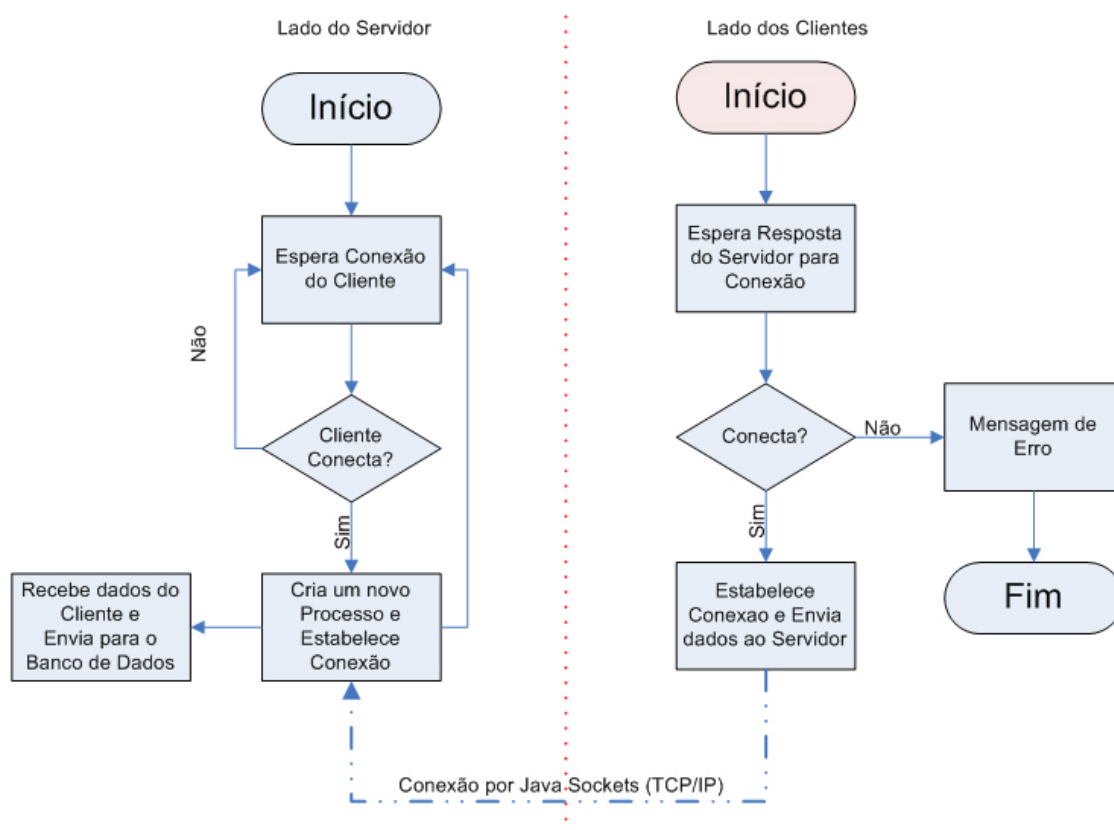


Figura 12 – Fluxograma da atividade básica *Cliente-Servidor*.

O cliente e o servidor se comunicam, basicamente, por meio de duas entidades, na linguagem Java na qual o programa foi desenvolvido, seguindo as premissas pré-definidas para o projeto, tais como:

- O programa como um todo deve ser capaz de abrir múltiplos processos com conexão ao servidor por meio de *sockets*.
- O programa deve ser capaz de, em cada processo, de tempos em tempos, realizar o fluxo de trabalho mantendo o SGBD atualizado de acordo com a programação para cada projeto.

4.1.3. Lado do Usuário

Os parâmetros são passados pelo Lado do Usuário que utiliza HTML com o navegador, enviando os pedidos para o servidor *Web*. Estas requisições atuam como cadastro de novo usuário ou consulta ao SGBD. Os procedimentos em PHP, situados no servidor *Web*, são responsáveis pelos processos de conexão e consulta ao SGBD, como mostrado na Figura 13, e

retornam os resultados para o navegador. Dessa forma, as respostas recebidas pelo servidor são escritas na linguagem HTML.

O HTTP é o nível de aplicação básico de protocolo de rede que coordena as trocas de dados entre o servidor *Web* e o navegador.

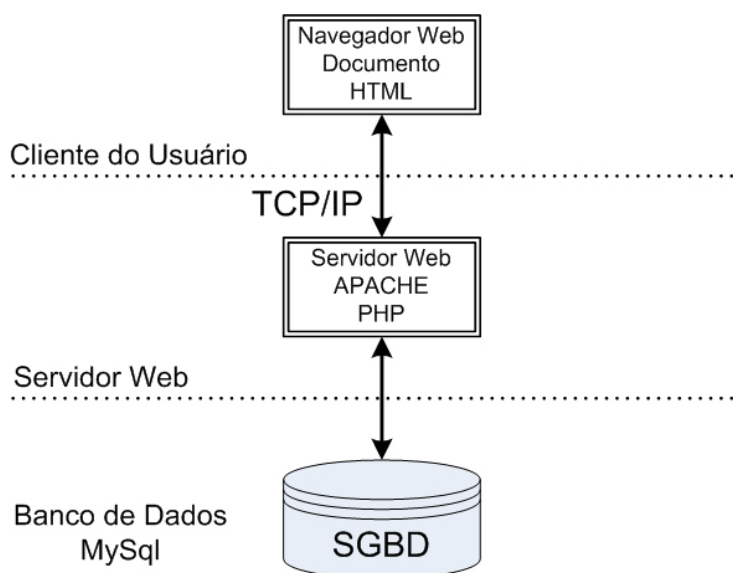


Figura 13 – Aplicação *Cliente-Servidor* representada em três camadas.

4.1.4. Servidor Web

Servidores *Web* são equipamentos que hospedam conteúdos da *Web* que, a partir de uma requisição por meio da INTERNET, podem ser acessados por equipamentos por meio de navegadores, os quais são denominados clientes.

Neste trabalho, o servidor *Web* foi de fundamental importância, por possibilitar a disponibilização do programa implementado para consulta pela INTERNET. O programa, desenvolvido em HTML e PHP junto ao SGBD, é visualizado do lado do Usuário pelo navegador. Um servidor Apache foi instalado como servidor *Web* por ser um dos mais utilizados mundialmente.

4.2. DESENVOLVIMENTO DO MODELO DE BANCO DE DADOS

Para a implementação da solução proposta neste trabalho, que utiliza a *Web* como interface com o usuário, fez-se necessária a escolha de um SGBD

que permitisse explorar as características básicas de implementação em aplicações *Cliente-Servidor*.

O MySQL é um SGBD relacional de código aberto, distribuído de forma gratuita e muito utilizado na *Web*. Dentre suas vantagens citam-se (CONVERSE et al., 2004):

- Desempenho em velocidade.
- Facilidade de uso, por utilizar estrutura SQL como forma de acesso e manipulação dos dados armazenados.
- Portabilidade, podendo ser executado em diversos sistemas operacionais.
- Segurança com senhas criptografadas, criando um sistema simples de autorizações e privilégios.
- Suporte a uma grande variedade de tecnologias como PHP, Java, Perl, dentre outras.
- Possibilidade de criação de tabelas com espaço superior a 4 *Gigabytes*.

A Figura 14 contém a forma organizacional do sistema proposto de acordo com a estrutura de banco de dados, apresentando o seu modelo em forma de um Diagrama de Entidades Relacionais.

A chave primária PK (*primary key*) é um campo ou um conjunto de campos na tabela que fornece ao SGDB um identificador exclusivo para cada linha.

A chave estrangeira FK (*foreign key*) faz o relacionamento entre as tabelas que se desejam relacionar.

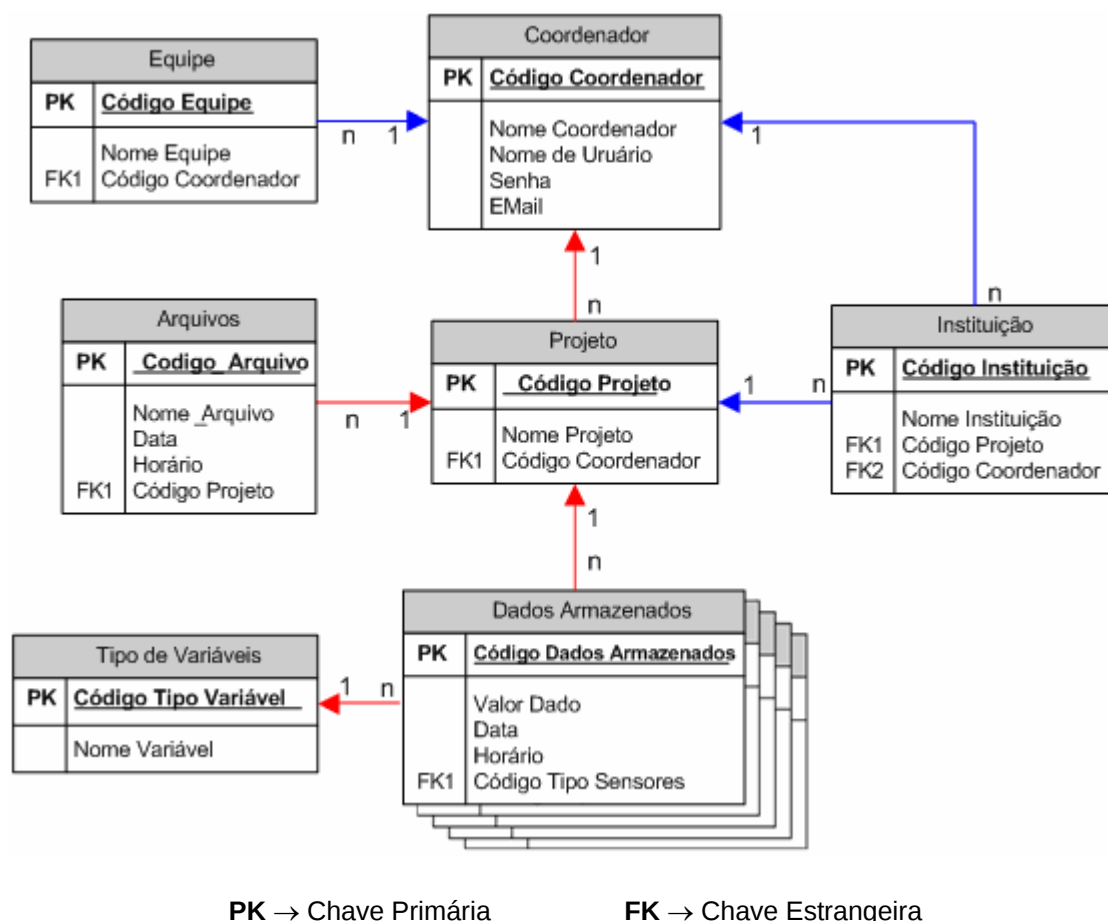


Figura 14 – Diagrama do *Modelo de Banco de Dados Relacional*.

O Banco de Dados foi estruturado para armazenar uma coleção de dados que foram classificados em 2 categorias: (i) dados em tempo real e (ii) arquivos anexados.

4.2.1. Dados em tempo real

São dados que se referem à coleta em tempo real e inclusão no banco de dados.

4.2.2. Arquivos Anexados

Inclusão de indexação para arquivos anexos referentes a outros sistemas de coleta em seus diversos formatos de arquivos.

As tabelas do modelo relacional são relevantes para o cadastramento, a identificação, a inclusão e a recuperação dos dados coletados.

4.3. ENSAIO DE AQUISIÇÃO DE DADOS

Foram realizados testes para validação do modelo proposto com base em experimentos simulados em laboratório e testes em tempo real.

Foram utilizados três sensores de temperatura (DS1820) endereçáveis do sistema 1-Wire™, devido a sua disponibilidade e ampla utilização em diversos trabalhos na Engenharia Agrícola (STEIDLE NETO et al., 2005; LOPES et al., 2006; LOPES et al., 2007; MARTINS et al., 2007), e um adaptador USB (*Universal Serial Bus*) DS9490R conectado a um computador para transferir os dados através de uma rede 1-Wire™ (DALLAS SEMICONDUCTOR, 2001; DALLAS SEMICONDUCTOR, 2002).

A comunicação entre os sensores 1-Wire™ e o computador foi estabelecida com a alteração do código fonte na parte Cliente do Projeto e as funções contidas no Apêndice A. Dessa forma, os dados foram coletados e transferidos para outro terminal por meio da INTERNET.

Os testes foram realizados no Laboratório de Instrumentação do Departamento de Engenharia Agrícola.

Dois sensores DS1820 foram expostos ao ambiente externo ao laboratório, sendo um deles encapsulado por um globo negro medindo 40 mm de diâmetro. O terceiro sensor foi utilizado para medir a temperatura interna do laboratório.

5. RESULTADOS E DISCUSSÃO

5.1. MODELO DE COMUNICAÇÃO

O sistema de comunicação foi codificado nas linguagens PHP, Java, SQL e HTML.

5.1.1. Ambiente de teste

Para ilustrar o uso do sistema, foi simulada a operação nos dois tipos de armazenamento de dados. As janelas apresentadas inicialmente são utilizadas pelo usuário para o cadastramento de seu projeto por meio do uso de algum navegador. Com isso, um projeto com nome “*Sistema de Transmissão de Dados*” teve seu cadastro inicial.

5.1.2. Ambiente *Cliente-Servidor Web*

Inicialmente, o usuário com conexão à INTERNET acessa a página de cadastro de projetos utilizando o endereço do servidor *Web*. Os dados fornecidos como Nome do Usuário, E-mail, Nome do Projeto, Login e Senha são gravados em uma tabela no Sistema Gerenciador de Banco de Dados (SGBD), para que a identidade do usuário possa ser certificada.

A Figura 15 mostra a janela principal do sistema com as opções de cadastro.

Pode-se observar que os campos a serem preenchidos na tela de cadastro são auto-explicativos e de fácil compreensão.

Coordenador:
Login:
Senha:
Repetir Senha:
Email:
Nome do projeto:

[VOLTAR PARA PAGINA DE LOGIN](#)

Figura 15 – Tela de Cadastro do Projeto.

A Figura 16 ilustra a tela de Login e Senha do usuário já previamente cadastrado. Depois da autenticação do usuário, o sistema identifica o projeto no qual o usuário está cadastrado, recupera os dados e os arquivos a ele referentes. Essa organização por projeto facilita a organização dos dados coletados e armazenados.

Login:
Senha:

[CADASTRAR UM NOVO PROJETO](#)

Figura 16 – Tela de Acesso ao Usuário do Projeto.

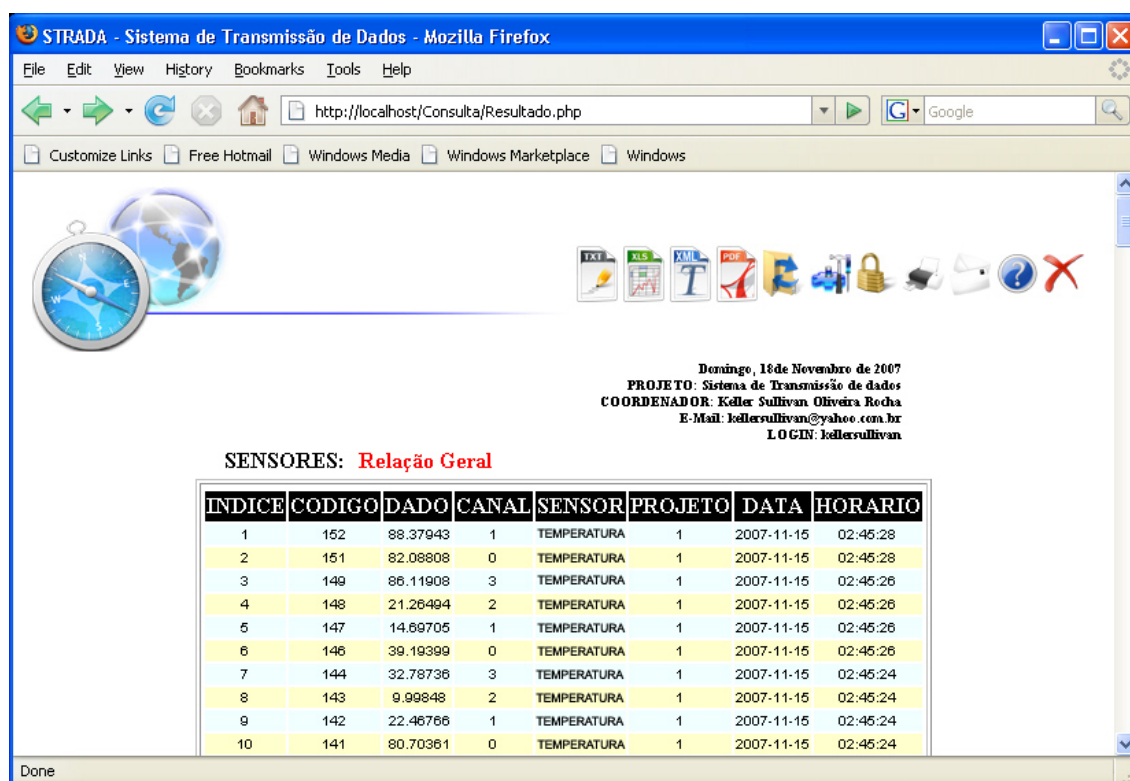
Na Figura 17 observam-se as opções do projeto, para que o usuário escolha o tipo de visualização tanto para o SGBD quanto para os arquivos anexados por ele.



Figura 17 – Tela de *Visualização Principal* - Anexos e Dados no SGBD.

5.1.2.1. Dados no Sistema Gerenciador de Banco de Dados - SGDB

Independentemente do tipo de variáveis que estão sendo medidas ou do tipo de dados que estão sendo coletados em um determinado experimento ou projeto, eles podem ser convertidos e visualizados nos formatos de arquivo texto (*.txt), arquivo para planilha eletrônica como OpenOffice, Excel e similares (*.xls) ou em formato de arquivo XML (*.xml) com as opções mostradas na Figura 18.



-  Exportar para Arquivo TXT
-  Exportar para Arquivo XLS
-  Exportar para Arquivo XML
-  Exportar para Arquivo PDF
-  Configurações do Projeto
-  Trocar Senha
-  Imprimir
-  Enviar para E-Mail

Figura 18 – Exemplo da relação geral dos dados no SGBD.

Para ilustrar a relação de dados contida no SGBD, apresenta-se, na Figura 19, os cinco primeiros dados de um arquivo exportado em formato texto, no qual o separador dos valores é determinado pelo espaço entre eles, definindo, assim, as colunas.

```

INDICE CODIGO DADO CANAL SENSOR PROJETO DATA HORARIO
1 152 88.37943 1 TEMPERATURA 1 2007-11-15 02:45:28
2 151 82.08808 0 TEMPERATURA 1 2007-11-15 02:45:28
3 149 86.11908 3 TEMPERATURA 1 2007-11-15 02:45:26
4 148 21.26494 2 TEMPERATURA 1 2007-11-15 02:45:26
5 147 14.69705 1 TEMPERATURA 1 2007-11-15 02:45:26
  
```

Figura 19 – Listagem de um arquivo em formato texto.

Da mesma forma, apresenta-se na Figura 20, um arquivo com os cinco primeiros dados exportados em formato xls, aberto pela planilha eletrônica do programa OpenOffice 2.2.

	A	B	C	D	E	F	G	H
1	INDICE	CODIGO	DADO	CANAL	SENSOR	PROJETO	DATA	HORARIO
2	1	152	88,38	1	TEMPERATURA	1	15/11/07	02:45:28
3	2	151	82,09	0	TEMPERATURA	1	15/11/07	02:45:28
4	3	149	86,12	3	TEMPERATURA	1	15/11/07	02:45:26
5	4	148	21,26	2	TEMPERATURA	1	15/11/07	02:45:26
6	5	147	14,7	1	TEMPERATURA	1	15/11/07	02:45:26

Figura 20 – Exemplo da listagem do arquivo em formato xls importado pela planilha eletrônica do *OpenOffice*.

A Figura 21 contém a primeira linha dos dados exportados em formato XML. Observa-se a sua estrutura, que pode ser aberta pelo navegador ou por uma planilha eletrônica, como mostrada na Figura 22, contendo as 5 primeiras linhas.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!--Gerado por Keller Sullivan - Data: /*DataAtual*/ -->
<ans:Strada xmlns:ans="http://www.strada.ufv.br">
<ans:Strada>
  <STRADA>
    <INDICE>1</INDICE>
    <CODIGO>152 </CODIGO>
    <DADO> 88.37943 </DADO>
    <CANAL>1</CANAL>
    <SENSOR>TEMPERATURA</SENSOR>
    <PROJETO>1 </PROJETO>
    <DATA>2007-11-15 </DATA>
    <HORARIO>02:45:28</HORARIO>
  </STRADA>
</ans:Strada>
```

Figura 21 – Exemplo da estrutura de arquivo XML.

	A	B	C	D	E	F	G	H
1	INDICE	CODIGO	DADO	CANAL	SENSOR	PROJETO	DATA	HORARIO
2	1	152	88,37943	1	TEMPERA	1	15/11/2007	02:45:28
3	2	151	82,08808	0	TEMPERA	1	15/11/2007	02:45:28
4	3	149	86,11908	3	TEMPERA	1	15/11/2007	02:45:26
5	4	148	21,26494	2	TEMPERA	1	15/11/2007	02:45:26

Figura 22 – Exemplo da estrutura de arquivo XML aberto no *MS-Excel*.

5.1.2.2. Arquivos Anexos

Os arquivos anexados têm a sua conexão (link) com comportamento de descarga (download), ou seja, clicando-se em qualquer arquivo da lista, é aberta uma janela onde o arquivo pode ser salvo no local desejado pelo usuário, conforme ilustrado na Figura 23

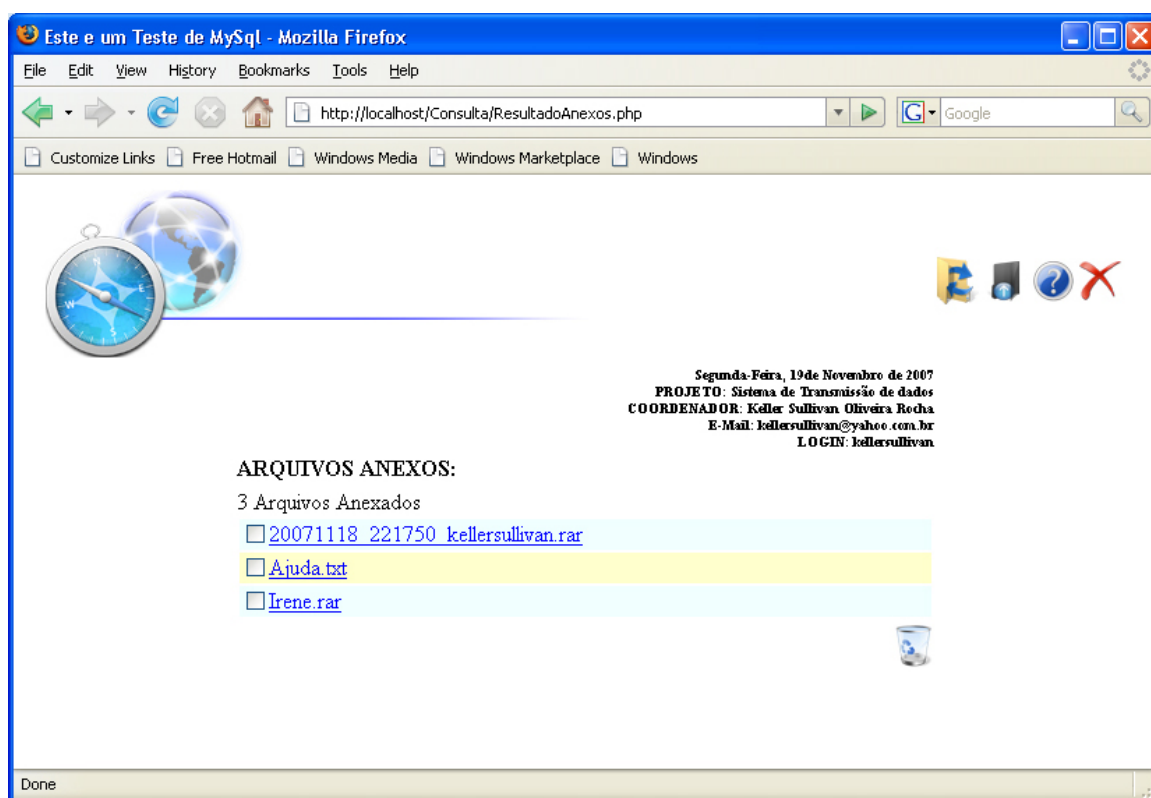


Figura 23 – Visualização da tela de arquivos anexados referente ao projeto.

Como mostrado na Figura 24, pode-se anexar arquivos referente ao experimento, enviando-os para o servidor *Web*, sendo este indexado na tabela do SGBD, de forma a ser visualizado e utilizado posteriormente.

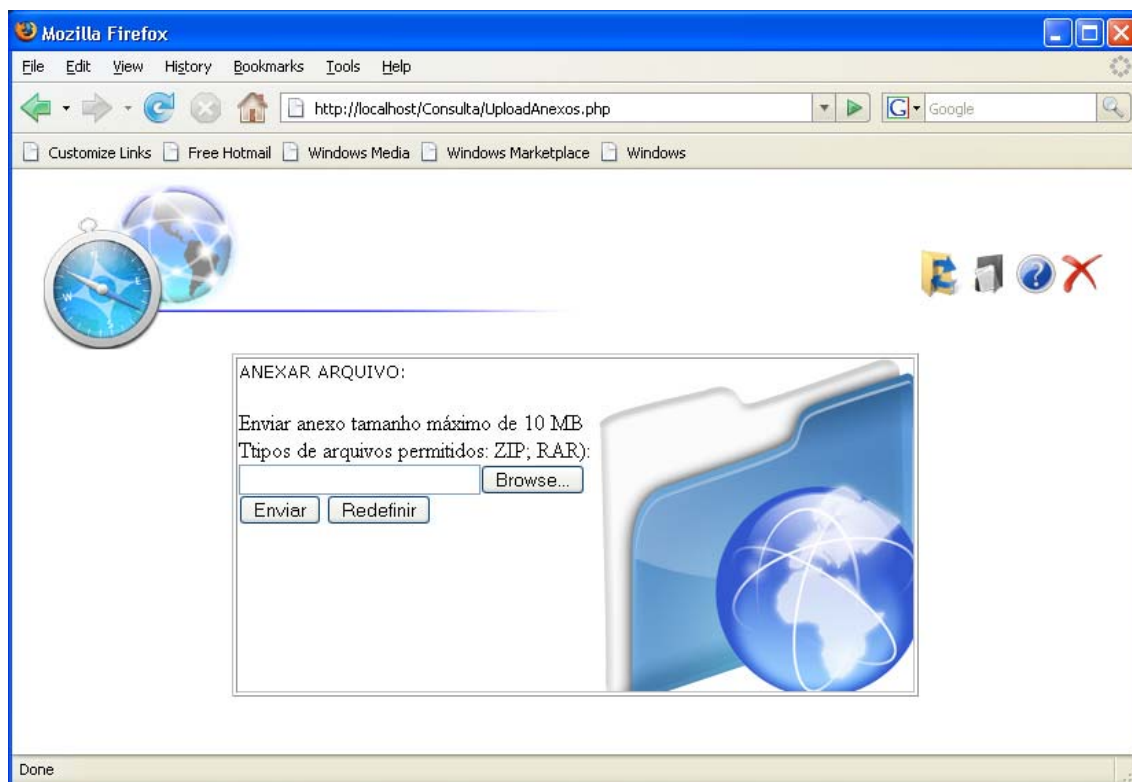


Figura 24 – Visualização da tela *Anexar Arquivo* do programa computacional STRADA.

5.1.3. Ambiente *Cliente-Servidor* do Projeto

Os dados do projeto são transmitidos para o servidor que está à espera, executando-se o programa desenvolvido em Java, como mostrado na Figura 25 e, logo em seguida, informando o *login* previamente cadastrado no SGBD.

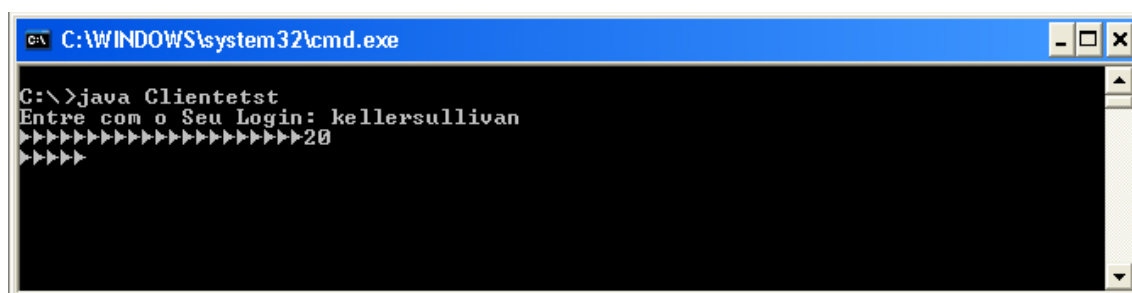


Figura 25 – Execução da sub-rotina *Cliente do Projeto* na linha de comando (*prompt*) do *Windows XP* do programa computacional STRADA.

As linhas a serem alteradas para adaptação do programa, por parte do *Cliente do Projeto*, com vistas à definição de variáveis ou tipo de dados a serem coletados, exportados ou importados, encontram-se exemplificados a seguir, com destaque (vermelho).

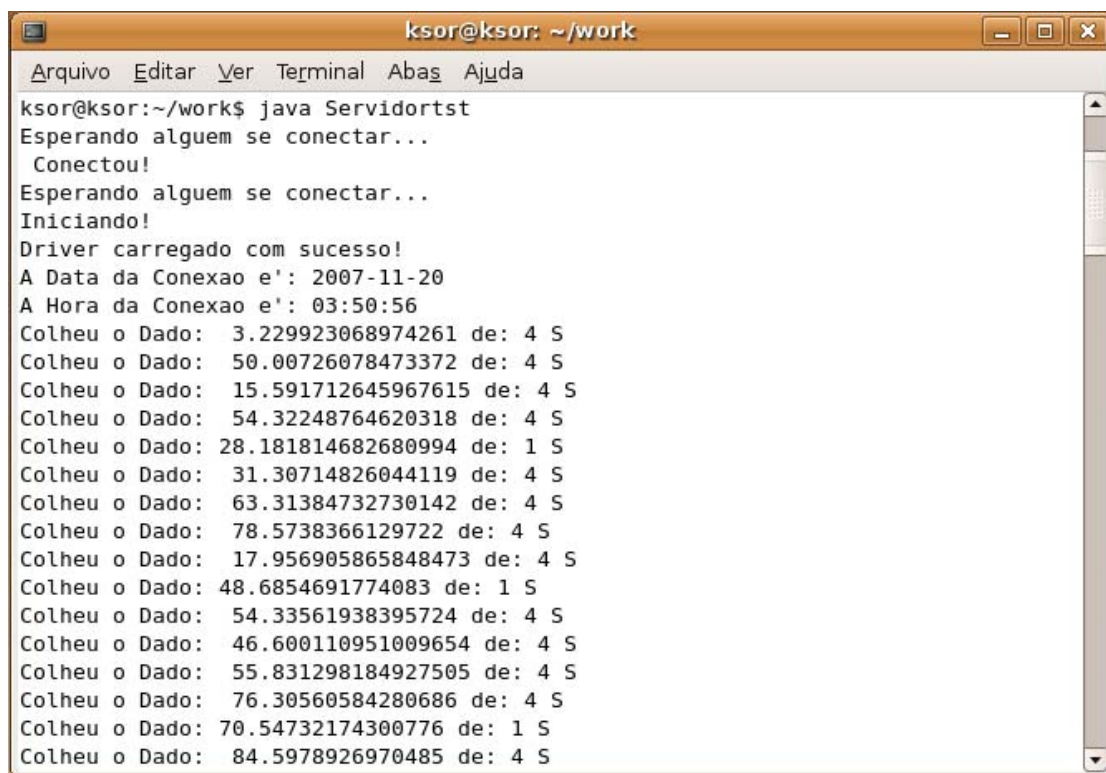
```

If(nowTotal>=2)//Tempo de coleta dos dados de cada sensor (Segundos)
{
    nowInicial=nowFinal;
    N="1";//Número de sensores do mesmo tipo, sempre informado em formato String
    S_CODIGO_TIPO_SENSORES="12"; //Código do sensor referente a lista da tabela
    IntN = Integer.parseInt(N);
    Sensores = new double[IntN];
    for(int i=0;i<IntN;i++)
    {
        saida.println(N);
        Sensores[i]=(Double) /* " Variável" */; //Função ou variável do sensor
        linha =saida.toString(Sensores[i]);
        saida.println(linha);
        saida.println(S_CODIGO_TIPO_SENSORES);
    }
}

```

Os comandos relacionados à comunicação entre os sensores e o computador, assim como pela codificação dos valores registrados, podem ser inseridos diretamente no código fornecido ou implementados em funções específicas.

O programa servidor, como mostrado na Figura 26, funcionando no Servidor do Projeto, aceita a conexão do cliente, identificando o usuário e adicionando os dados a uma tabela contida no SGBD e, ao mesmo tempo, é aberto um novo processo para um novo usuário. Com isso, pode-se ter múltiplas conexões com os processos criados depois da conexão anterior, referente ao mesmo experimento ou experimentos distintos.



```

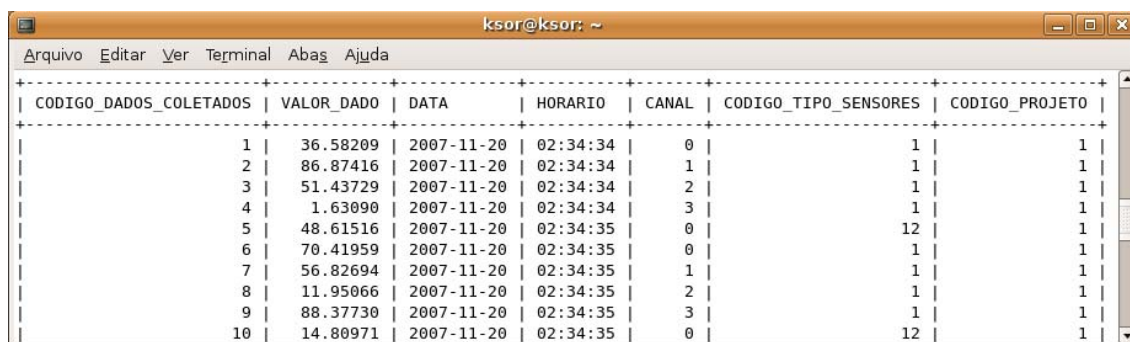
ksor@ksor: ~/work
Arquivo Editar Ver Terminal Abas Ajuda
ksor@ksor:~/work$ java Servidortst
Esperando alguem se conectar...
Conectou!
Esperando alguem se conectar...
Iniciando!
Driver carregado com sucesso!
A Data da Conexao e': 2007-11-20
A Hora da Conexao e': 03:50:56
Colheu o Dado: 3.229923068974261 de: 4 S
Colheu o Dado: 50.00726078473372 de: 4 S
Colheu o Dado: 15.591712645967615 de: 4 S
Colheu o Dado: 54.32248764620318 de: 4 S
Colheu o Dado: 28.181814682680994 de: 1 S
Colheu o Dado: 31.30714826044119 de: 4 S
Colheu o Dado: 63.31384732730142 de: 4 S
Colheu o Dado: 78.5738366129722 de: 4 S
Colheu o Dado: 17.956905865848473 de: 4 S
Colheu o Dado: 48.6854691774083 de: 1 S
Colheu o Dado: 54.33561938395724 de: 4 S
Colheu o Dado: 46.600110951009654 de: 4 S
Colheu o Dado: 55.831298184927505 de: 4 S
Colheu o Dado: 76.30560584280686 de: 4 S
Colheu o Dado: 70.54732174300776 de: 1 S
Colheu o Dado: 84.5978926970485 de: 4 S

```

Figura 26 – Execução do programa Servidor do Projeto no terminal Linux Ubuntu.

5.2. SISTEMA DE BANCO DE DADOS

Para cada projeto cadastrado, é criada, dinamicamente, uma tabela identificada pelo *login* do projeto e estruturado, como mostrado na Figura 27, para a transferência de dados. Dessa forma, tem-se uma tabela de transferência de dados para cada projeto, com as outras tabelas comunicando com esta de forma relacional.



CODIGO_DADOS_COLETADOS	VALOR_DADO	DATA	HORARIO	CANAL	CODIGO_TIPO_SENSORES	CODIGO_PROJETO
1	36.58209	2007-11-20	02:34:34	0	1	1
2	86.87416	2007-11-20	02:34:34	1	1	1
3	51.43729	2007-11-20	02:34:34	2	1	1
4	1.63090	2007-11-20	02:34:34	3	1	1
5	48.61516	2007-11-20	02:34:35	0	12	1
6	70.41959	2007-11-20	02:34:35	0	1	1
7	56.82694	2007-11-20	02:34:35	1	1	1
8	11.95066	2007-11-20	02:34:35	2	1	1
9	88.37730	2007-11-20	02:34:35	3	1	1
10	14.80971	2007-11-20	02:34:35	0	12	1

Figura 27 – Exemplo da tabela “Dados Armazenados” em SQL.

5.3. TESTES DE ADEQUAÇÃO DO SISTEMA COMPUTACIONAL

Os dados coletados, conforme descrito no item 4.3., foram armazenados no SGBD a cada segundo, durante um período de 1000 segundos. A Figura 28 mostra os resultados obtidos durante os testes realizados com os três sensores de temperatura.

Na Figura 28, a curva 01 representa a variação de temperatura do ar no interior do globo negro; a curva 02 corresponde às temperaturas do ar no ambiente externo e a curva 03 a variação da temperatura do ar interno do laboratório. Como esperado, os valores das curvas 02 e 03 foram inferiores aos valores de temperatura do globo negro.

Na curva 03, os valores são bem menores e menos dispersos do que os demais, o que está também de acordo com o esperado.

Desta forma, comprova-se a adequação do sistema proposto. Vale salientar que é irrelevante o tempo de coleta de dados e a natureza dos dados coletados para o contexto do sistema desenvolvido, por tratar-se de um sistema de caráter geral.

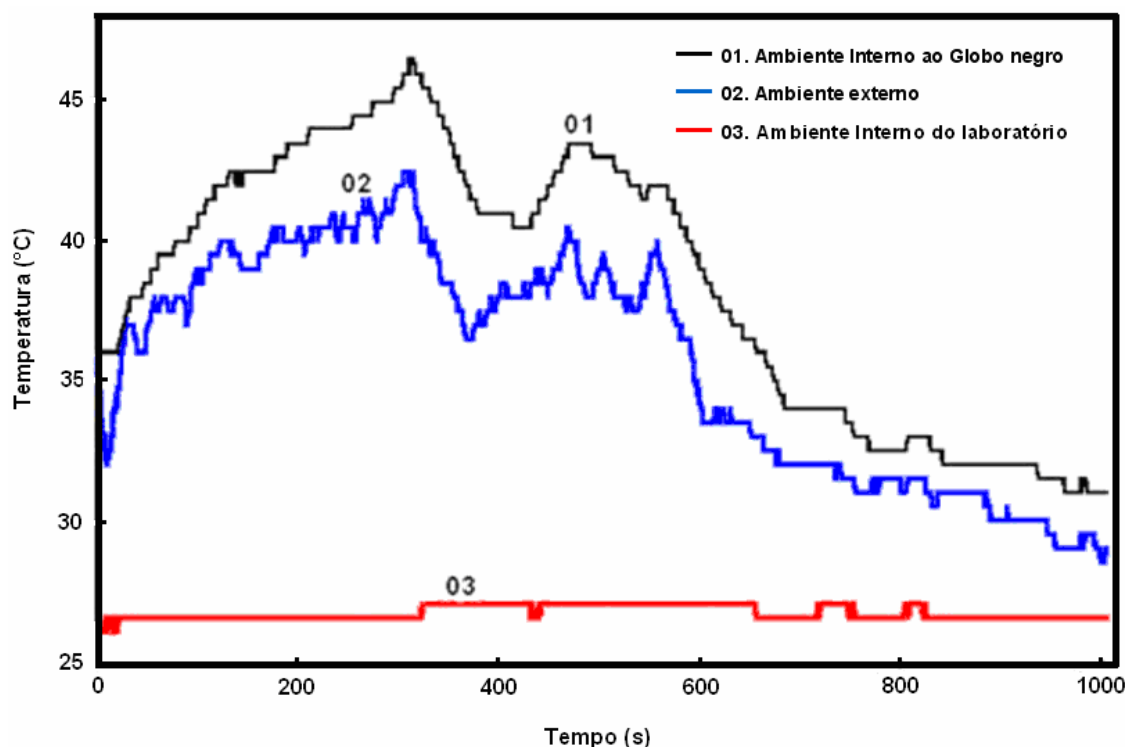


Figura 28 – Valores de temperatura coletados durante os testes.

As linhas alteradas, com os comandos relacionados à comunicação entre os sensores do sistema 1-Wire™ no Terminal do Projeto com o Servidor do Projeto, foram inseridas diretamente no código fonte, como mostradas em destaque (vermelho) a seguir.

```

If(nowTotal>=2)//Tempo de coleta dos dados de cada sensor (Segundos)
{
    nowInicial=nowFinal;
    N="3";//Número de sensores do mesmo tipo, sempre informado em formato String
    S_CODIGO_TIPO_SENSORES="1"; //Código do sensor referente a lista da tabela
    IntN = Integer.parseInt(N);
    Sensores = new double[IntN];
    for(int i=0;i<IntN;i++)
    {
        saida.println(N);
        Sensores[i]=(Double)le_temperatura(owd_vect, i, tentativas); //Função de
                                e reconhecimento dos valores do sensor
        linha =Saida.toString(Sensores[i]);
        saida.println(linha);
        saida.println(S_CODIGO_TIPO_SENSORES);
    }
}

```

As Figuras 29 e 30 mostram, respectivamente, como foi realizada a transmissão dos dados no programa computacional em tempo real no ambiente dos sistemas operacionais Windows e Linux.

6. CONCLUSÕES

Com base nos resultados e nas observações realizadas ao longo desse trabalho, concluiu-se que:

1. O sistema desenvolvido se aplica ao processo científico de monitoramento e armazenamento de dados de experimentos por meio da INTERNET, de forma a armazenar, organizar e distribuir esses dados, possibilitando aos usuários o acesso aos dados pela *Web*. Foi verificado que, independentemente do tipo de dados que se querem transmitir ou registrar, a criação de um Sistema de Gerenciamento e Transmissão Remota de Dados, tanto para divulgação de dados quanto para a aquisição desses pelo usuário, por meio da INTERNET, constitui-se uma ferramenta indispensável para melhorar a eficiência na execução de projetos de pesquisa.
2. Tanto a consulta de dados históricos quanto a transferência de dados recentes para o SGBD apresentaram um funcionamento adequado ao seu propósito, comprovando a possibilidade da implementação definitiva como uma ferramenta de auxílio aos pesquisadores em seus experimentos.
3. O sistema desenvolvido permite ao usuário visualizar os dados dos experimentos utilizando-se apenas do navegador *Web*, sendo desnecessária a instalação de programas adicionais.

O programa computacional desenvolvido possibilita a utilização para as seguintes atividades por parte do usuário:

- i. Cadastramento do projeto pelo usuário.
- ii. Inclusão e exclusão de arquivos anexos referente ao experimento.
- iii. Visualização dos dados pelo navegador.
- iv. Exportação dos dados do SGBD para as formas de arquivo texto (*.txt), planilha eletrônica (*.xls) e Linguagem de Marcação Extensiva (*.xml), facilitando a importação por programas de análise de dados.

4. O programa computacional desenvolvido neste trabalho é de aplicação geral, embora a ênfase tenha sido dada às aplicações de engenharia, em especial à Engenharia Agrícola, centralizando a visualização de dados colhidos em experimentos, em múltiplos projetos ou em projetos com múltiplos pontos de coletas distintos, independentemente de sua localização geográfica. Com isso, tem-se a redução com gastos operacionais e maximização do tempo útil do pesquisador.

6.1. SUGESTÕES PARA TRABALHOS FUTUROS

Muitas são as possibilidades de melhorias e propostas de acréscimos futuros ao sistema proposto. Dentre elas citam-se: (i) comunicação entre os usuários do mesmo projeto pelo sistema; (ii) segurança e controle por comunicação móvel; (iii) estudar a viabilidade de implementação no sistema de gerenciamento e armazenamento com sistema de controle de dispositivos por meio da INTERNET, criando um laboratório virtual, com o objetivo de aproveitar melhor os recursos já existentes. Assim, seria possível o desenvolvimento de ferramentas relacionadas à automação por meio da INTERNET, como ferramenta fundamental para implementação de controle de ambientes remotos.

7. REFERÊNCIAS BIBLIOGRÁFICAS

ACHOUR, M.; BETZ, F.; DOVGAL, A.; LOPES, N.; OLSON, P.; RICHTER, G.; SEGUY, D.; VRANA, J. **PHP Manual**. PHP Documentation Group, 2005.

ALBU, M. M.; HOLBERT, K. E.; HEYDT, G. T.; GRIGORESCU, S. D.; TRUSCA, V. **Embedding remote experimentation in power engineering education**. Power Systems, IEEE Transactions, Politehnica Univ. of Bucharest, Romania, v. 19, p. 139-143, 2004.

BAI, Y.; WEI, H. **Design and implementation of a client-server remote Windows-based signal generator**. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, v.1, p.78–83, 2001.

BENTHAM, J. **TCP/IP Lean-Web Servers for Embedded Systems**. CMP Books, Kansas, 2th ed., 2002.

BERNAT, G.; BURNS, A.; WELLINGS, A. **Portable worst-case execution time analysis using Java Byte Code**. 12th Euromicro Conference on Real-Time Systems, p.81-88, 2000.

BONHO, S.; KOLM, D.; BAGGIO, J. F. R.; SCHWARZ, L.; MORAES, R. **Monitoração de Sinais de ECG pela Internet em Tempo Real**. Instituto de Engenharia Biomédica, Universidade Federal de Santa Catarina, Brasil, 2006.

BRASS, S.; GOLDBERG, C. **Proving the safety of SQL queries**. Fifth International Conference on Quality Software, p.197-204, 2005.

CHINNAPPEN-RIMER, S.; HANCKE, G. P. **An XML model for use across heterogeneous client-server applications**. IEEE International Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, p.207–211, 2003.

CONVERSE, T.; PARK J.; MORGAN C. **PHP5 and MySQL Bible**. Wiley Publishing, Inc., Indianapolis, IN, 1080p, 2004.

DALLAS SEMICONDUCTOR. **DS18S20: High precision digital thermometer 2002**. <http://www.maxim-ic.com>. Acessado em 15/08/2007.

DALLAS SEMICONDUCTOR. **Guidelines for Reliable 1-Wire™ Networks, 2001**. http://www.maxim-ic.com/appnotes.cfm/appnote_number/148. Acessado em 15/08/2007.

DOULAI, P. **Engineering and science laboratory courseware delivery using World Wide Web technology**. IEEE International Conference on Multi Media Engineering Education, p.339–344, 1996.

DUNLOP, J.; SMITH, D. G. **Telecomunications Engineering**. Chapman & Hall, 3th ed., London, 593 p, 1994.

DUO L. I.; SERIZAWA, Y.; KIUCHI, M. **Extension of client-server applications to the Internet**. IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, v.1, p.355-358, 2002.

FURMSTON-EVANS, I. R. **Successful system architecture development for enterprise wide client server implementation.** IEEE International Seminar on Client/Server Computing. Seminar Proceedings (Digest No. 1995/184), v.1, p. 1- 5, 1995.

HAROLD, E. R. **XML Bible, New York, Second Edition.** Hungry Minds, Inc., 1015p, 2001.

HURD, J. M. **The transformation of scientific communication: A model for 2020.** Journal of the American Society for Information Science, Science Library, University of Illinois, Chicago, 1279p., 2000.

INTERNET USAGE STATISTICS. **The Big Picture.** Internet World Stats (19 de março de 2007). <http://www.internetworldstats.com/stats.htm>. Acessado em 05/11/2007.

KAMIYAMA, N. **Network Topology Design with Multiple Criteria.** 12th International Telecommunications Network Strategy and Planning Symposium, NETWORKS 2006, p.1-6, 2006.

KUROSE, F. J. **Computer networking: a top-down approach featuring the internet.** Pearson Education Inc., 3th ed., 712p, 2005.

LOPES, D. C.; MARTINS J. H.; MONTEIRO, P. M. B.; QUEIROZ, D. M. **Redes 1-Wire™ Aplicadas à aeração de grãos armazenados.** Acta Sci. Agron. Maringá, v.29, n.2, p. 157-163, 2007.

LOPES, D. C.; MARTINS, J. H.; MELO, E. C.; MONTEIRO, P. M. B. **Aeration simulation of stored grain under variable air ambient conditions.** Postharvest Biology and Technology, Pullman, WA, USA, v. 42, p. 115-120, 2006.

MARINO, P.; NOGUEIRA, J.; HERNA, H. **Laboratory of Virtual Instrumentation for Industrial Electronics.** Proceedings of IEEE International Conference on Industrial Technology 2000, v. 2, p. 249-253, 2000.

MARTINS, J. H.; LOPES, D. C.; MONTE, J. E. C.; MONTEIRO, P. M. B.; PINTO, P. R. **Sistema de pesagem automática de amostras em um secador de grãos em camada fina.** 2º Seminário de Sistemas Industriais e Automação, CEFET-MG, Belo Horizonte, MG, 2007.

MELO, A. A.; MENEZES, P. M. L.; SAMPAIO, A. C. F. **O uso de SIG na pesquisa geográfica voltada para o ensino e a aprendizagem.** Caminhos de Geografia - revista on-line, 2006.

NAUGLE, M. G. **Illustrated TCP/IP.** Wiley Computer Publishing, John Wiley & Sons, Inc., 4th ed., 499p, 1998.

NETCRAFT. **Market Share for Top Servers Across All Domains - August 1995 - October 2007.** <http://news.netcraft.com/>. Acessado em 09/11/2007.

PIZARRO, P. J. C.; LOPES, C.; MORAES, R.; MARQUES, J. L. B. **Monitoramento remoto de sinais bioelétricos.** Santa Catarina, Trindade, Florianópolis, SC, 2001.

ROBERTS, T. J.; SCOTT, N.; HADGRAFT, R.; ILIC, V. **The virtual machines laboratory**. Australasian Journal of Engineering Education, Australia, 2004.

RODRIGUES, A.; GATRELL, J.; KARAS, J.; PESCHKE, R. **TCP/IP Tutorial and technical overview**. RedBooks, IBM Corporation, 7th ed., North Carolina, 2001.

SANTANA, M. O.; RIBEIRO, A.; SEDIYAMA, G. C. **Sistema de geoespacialização da demanda de irrigação suplementar para o Estado de Minas Gerais**. Revista Brasileira de Engenharia Agrícola e Ambiental, Campina Grande, PB, v.7, n.1, p.57-63, 2003.

SCHLOSSNAGLE, G. **Advanced PHP Programming**. Developer's Library, Sams Publishing, Indianapolis, IN, 2004.

STEIDLE NETO, A. J. S.; BAÊTA F. C.; MARTINS J. H.; ZOLNIER S.; MONTEIRO P. M. B. **Avaliação da transmissão de dados de temperatura no sistema 1-Wire™**. Eng. Agríc., Jaboticabal, v.25, n.1, p.29-36, jan./abr. 2005.

TANAMACHI, K.; INOUE, K.; MOSHNYAGA, V.G. **Designing a TCP/IP core for power consumption analysis**. Proceedings of 2004 IEEE Asia-Pacific Conference on Advanced System Integrated Circuits, p. 412–413, 2004.

VARGAS, A. A. **Estudo sobre Comunicação de Dados via Rede Elétrica para Aplicações de Automação Residencial/Predial**. Porto Alegre, janeiro de 2004.

WALDRON, J. **Dynamic bytecode usage by object oriented Java programs**. Proceedings of Technology of Object-Oriented Languages and Systems, p. 384–393, 1999.

WELLING, L.; THOMSON, L. **Desarrollo web con PHP y MYSQL**. Naya Multimedia, 1^a ed., 976p, 2005.

WIKÍPEDIA. **Servidor HTTP Apache**. http://pt.wikipedia.org/wiki/Lista_de_servidores_web. Acessado em 02/09/2007.

ZHANG, K.; ZHANG, D.; DENG, Y. **A visual approach to XML document design and transformation**. Proceedings of IEEE Symposia on Human-Centric Computing Languages and Environments, p. 312–319, 2001.

[illegible]

