



**UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA
DE CONTROLE E AUTOMAÇÃO - CEAU**



RUI FERREIRA DA SILVA NETO

**LABORATÓRIOS REMOTOS:
ANÁLISE DE ARQUITETURAS E PROPOSTA TEÓRICA DE MODELO
GENÉRICO**

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Ouro Preto, 2016

RUI FERREIRA DA SILVA NETO

**LABORATÓRIOS REMOTOS:
ANÁLISE DE ARQUITETURAS E PROPOSTA TEÓRICA DE MODELO
GENÉRICO**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Alan Kardek Rêgo Segundo

Co-orientador: Prof. Msc. José Alberto Naves Cocota Junior

**Ouro Preto
Escola de Minas – UFOP
Agosto/2016**

S586l

Silva Neto, Rui Ferreira da.

Laboratórios Remotos [manuscrito]: Análise de arquiteturas e proposta teórica de modelo genérico / Rui Ferreira da Silva Neto. - 2016.

78f.: il.: color; tabs.

Orientador: Prof. Dr. Alan Kardek Rêgo Segundo.

Coorientador: Prof. Msc. José Alberto Naves Cocota Junior.

Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais.

I. Redes de Computadores. 2. Ensino à Distância. 3. Laboratórios. 4. Controle automatico. I. Rêgo Segundo, Alan Kardek . II. Cocota Junior, José Alberto Naves. III. Universidade Federal de Ouro Preto. IV. Título.

CDU: 681.5

Catálogo: ficha@sisbin.ufop.br

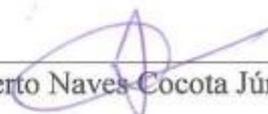
Monografia defendida e aprovada, em 11 de agosto de 2016, pela comissão avaliadora constituída pelos professores:



Prof. Dr. Alan Kardek Rêgo Segundo - Orientador



Prof. Dr. Paulo Marcos de Barros Monteiro – Professor Convidado



Prof. M. Sc. José Alberto Naves Cocota Júnior – Professor Convidado

A meus pais e avós por fazerem das conquistas de seus filhos suas prioridades de vida.

AGRADECIMENTOS

Agradeço a meus pais, irmãs e avós pelo apoio, amor e carinho. À minha namorada pela dedicação nos momentos difíceis. À república Favela pelos muitos anos de irmandade. Aos professores pela orientação nos momentos críticos e pelo aprendizado transmitido. E por fim a UFOP/EM pelas oportunidades e a qualidade do ensino oferecida.

“Per ardua surgo”
(Lema da Bahia)

RESUMO

Com o aprimoramento e a difusão dos sistemas de redes de computadores e da Internet, criou-se um meio para o desenvolvimento de uma gama de aplicações que tornaram tais ferramentas essenciais na transmissão de informações. Aliado a essa crescente, o surgimento de dispositivos baseados na utilização de sistemas embarcados vem possibilitando que soluções para problemas cotidianos, tanto industriais quanto residenciais, ofereçam maior conforto e praticidade na execução de tarefas ao fazê-lo de maneira remota. Ao mesmo passo, a Educação à Distância (EaD) tem evoluído muito nos últimos anos, tornando-se cada vez mais uma aliada aos métodos de ensino, posto que os Laboratórios Remotos tem integrado parte dessa evolução, ao aliar os conceitos de ensino prático à distância e acesso remoto. Dessa forma, uma análise da arquitetura desses sistemas comprova que, os pesquisadores da área têm buscado um melhor entendimento e aperfeiçoamento na aplicação de tais técnicas. Com isso a união dos conceitos de rede de computadores, acesso remoto via Internet e sistemas embutidos, permitiu a formalização do conceito desses sistemas além de elaborar uma arquitetura mais genérica e estável, com base em um servidor web computacional unido a plataforma microcontrolada Arduino. A meta principal é vincular o uso da Internet à utilização de sistemas embarcados por meio dos protocolos de comunicação de rede e de uma comunicação serial com um servidor, possibilitando realização de leitura e escrita de portas digitais e analógicas, acionar e controlar dispositivos elétricos, remotamente, viabilizando o conceito de Laboratório Remoto funcional no futuro.

Palavras-chaves: Redes de computadores, Sistemas Embarcados, Acesso Remoto, Internet, Arduino, Laboratórios Remotos, Educação a Distância e Controle e Automação.

ABSTRACT

With the improvement and diffusion of computer networks and the Internet systems, created a means for the development of a range of applications that have made such essential tools in the transmission of information, combined with this growing, the emergence of devices based on use of embedded systems is enabling solutions to everyday problems, both industrial as residential, offer greater comfort and convenience in performing tasks to do it remotely. At the same step Distance Education (EAD) has evolved in recent years, becoming an increasingly allied to teaching methods, since the Remote Labs has integrated part of this evolution, by combining the practical teaching concepts to distance and remote access. Thus, one of these systems architecture analysis shows that researchers in the field have sought a better understanding and improvement in the application of such techniques. Thus the union of the concept of computer network, remote access via Internet and embedded systems, has allowed the formalization of the concept of these systems as well as develop a more generic and stable architecture, based on a computer attached web server microcontrolled platform Arduino. The main goal is to link the use of the Internet the use of embedded systems through the network communication protocols and serial communication with a server, allowing performing reading and writing of digital and analog ports, trigger and control electrical devices remotely , enabling the concept of Remote Laboratory functional in the future.

Key-words: Computer networks, embedded systems, remote access, Internet, Arduino, Remote Laboratories, Distance Education and Control and Automation.

LISTA DE FIGURAS

FIGURA 1 - Descrição detalhada implementação geral do sistema	19
FIGURA 2 - Modelo Proposto	20
FIGURA 3 - Produto final	21
FIGURA 4 - Camadas do firmware utilizado no MSW	21
FIGURA 5 - Arquitetura DIESEL genérica	22
FIGURA 6 - Ambiente integrado de aprendizagem DIESEL com trabalho colaborativo	23
FIGURA 7 - Protocolos de comunicação na arquitetura DIESEL com trabalho cooperativo	24
FIGURA 8 - Modelo de comunicação com quatro camadas DIESEL	24
FIGURA 9 - Estrutura do Sistema.....	25
FIGURA 10 - Arquitetura definida	27
FIGURA 11 - Página do experimento	28
FIGURA 12 - Relações globais para a arquitetura do sistema de rede proposto	29
FIGURA 13 - Estrutura geral de interconexão para o componente da 3ª camada de rede.....	30
FIGURA 14 - Esquema de interconexão para a troca de variável entre o aplicativo de controle local e a interface gráfica	31
FIGURA 15 - Estrutura geral entre os componentes da 3ª camada e o MAS na 2ª camada....	31
FIGURA 16 - Interação entre o MAS e os LMS para a utilização de um laboratório remoto através do sistema	32
FIGURA 17 - Experimento	33
FIGURA 18 - Ambiente de Interação do experimento remoto	34
FIGURA 19 - Diagrama de Blocos de um Controlador PID.....	35
FIGURA 20 - Bancada Simulada da VISIR.....	37
FIGURA 21 - Componentes da VISIR e o ciclo de operação.....	38
FIGURA 22 - Visão geral da infraestrutura RELDES.	39
FIGURA 23 - Diagrama de processo do laboratório.....	40
FIGURA 24 - Interface web do RELDES.....	40
FIGURA 25 - Classificação dos laboratórios remotos de acordo com o tipo de servidor	41
FIGURA 26 - Arquitetura da Ethernet original.....	45
FIGURA 27 - Visão geral da Internet.	46
FIGURA 28 - Camada de aplicação	48
FIGURA 29 - Camada de transporte.	49
FIGURA 30 - Tridente.	52

FIGURA 31 - Elementos básicos de um sistema embutido	54
FIGURA 32 - Diagrama de pinos do ATmega168/328.....	60
FIGURA 33 - Laboratório remoto genérico.....	63
FIGURA 34 - Fluxo de dados da arquitetura cliente/servidor do sistema.....	64
FIGURA 35 - Componentes de hardware do sistema	65

LISTA DE TABELAS

Tabela 1 - Descrição dos Blocos no Ambiente de Desenvolvimento	34
Tabela 2 - Comparação dos modelos estudados	42
Tabela 3 - Classificação das redes de computadores.....	45
Tabela 4 - A pilha de protocolo da Internet com base no modelo de referencia TCP/IP	47
Tabela 5 - Tabela de padrão de cores	53
Tabela 6 - Especificações Técnicas do Arduino uno.....	57

LISTA DE SIGLAS

EaD	Educação à Distância
IoT	<i>Internet of Things</i>
MSW	Micro Servidor Web
TCP/IP	<i>Transmission Control Protocol/ Internet Protocol</i>
RS232	<i>Recommended Standard 232</i>
CI	Circuito Integrado
MCU	Microcontrolador
IC	<i>Inter Integrated Circuit</i>
EEPROM	<i>Electrically Erasable Programmable Read Only Memory</i>
HTTP	<i>HyperText Transfer Protocol</i>
DIESEL	<i>Distance Internet-Based Embedded System Experimental Laboratory</i>
GPIB	<i>General Purpose Interface Bus</i>
P2P	<i>Peer-to-Peer</i>
SOAP	<i>Simple Object Access Protocol</i>
TTL	<i>Transistor-Transistor Logic</i>
IDE	<i>Integrated Development Environment</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
JSP	<i>Java Server Pages</i>
JSF	<i>Java Server Faces</i>
PIC	<i>Programmable Interface Controllers</i>
LED	<i>Light Emitting Diode</i>
LCD	<i>Liquid Crystal Display</i>
E/S	Entrada/Saída
PWM	<i>Pulse Width Modulation</i>
AJAX	<i>Asynchronous Javascript and XML</i>
UCA	<i>User-Client Applications</i>
LMS	<i>Learning Administration System</i>
LPA	<i>Lab Process Applications</i>
MAS	<i>Management and Administration System</i>
GUI	<i>Graphical User Interface</i>
JIL	<i>Java Internet Labview</i>
EJS	<i>Easy Java Simulation</i>
MVC	<i>Model-View-Controller</i>
URL	<i>Uniform Resource Locator</i>
PID	<i>Proportional-Integral-Derivative</i>
VISIR	<i>Virtual Instrument Systems in Reality</i>
PCI	<i>Peripheral Component Interconnect</i>
PXI	<i>PCI eXtensions for Instrumentation</i>
NI	<i>National Instruments</i>
DPST	<i>Double-Pole Single-Throw</i>
SPST	<i>Single-Pole Single- Throw</i>
SGA	<i>Sistema de Gestión de Aprendizaje</i>

PHP	<i>Personal Home Page</i>
EC	<i>Experiment Client</i>
WS	<i>Web Server</i>
MS	<i>Measurement Server</i>
EP	<i>Experiment Protocol</i>
ES	<i>Equipment Server</i>
IVI	<i>Instrumentos Virtuales Intercambiables</i>
LXI	<i>LAN eXtensions for Instrumentation</i>
RELDES	<i>REmote Laboratory for Embedded Systems Design</i>
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
JS	<i>Java Script</i>
LAN	<i>Local Area Network</i>
MAN	<i>Metropolitan Area Network</i>
WAN	<i>Wide Area Network</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
RFC	<i>Request for Comments</i>
UDP	<i>User Datagram Protocol</i>
MAC	<i>Media Access Control</i>
ARP	<i>Address Resolution Protocol</i>
USB	<i>Universal Serial Bus</i>
NRZI	<i>No Return Zero Inverted</i>
CPU	<i>Central Processing Unit</i>
ALU	<i>Arithmetic Logic Unit</i>
ROM	<i>Read Only Memory</i>
RAM	<i>Random Access Memory</i>
SPI	<i>Serial Peripheral Interface</i>
SS	<i>Slave Select</i>
MOSI	<i>Master Out Slave In</i>
MISO	<i>Master In Slave Out</i>
SCK	<i>Serial Clock</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
SDA	<i>Serial Data Line</i>
SCL	<i>Serial Clock Line</i>
AREF	<i>Analog Reference</i>
IOREF	<i>Input-Output Reference</i>
PAN	<i>Personal Area Network</i>
WoT	<i>Web of Things</i>
RFID	<i>Radio-Frequency IDentification</i>
WWW	<i>World Wide Web HTTP</i>

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 Considerações Iniciais.....	14
1.2 Origem do Trabalho.....	15
1.3 Objetivos.....	15
1.3.1 Objetivo geral.....	15
1.3.2 Objetivos específicos.....	15
1.4 Justificativa.....	16
1.5 Estrutura do Trabalho.....	17
2 ESTADO DA ARTE.....	19
2.1 Micro Servidor Web.....	19
2.2 Projeto DIESEL.....	22
2.3 Sistema de Acesso Remoto à Experiências de Eletrônica.....	25
2.4 Laboratório de Experimentação Remota com Microcontroladores PIC.....	27
2.5 Sistema de Laboratório Acadêmico Remoto Multiusuário.....	29
2.6 Experimento Educacional Remoto.....	33
2.7 Experimentação <i>On-line</i> Utilizando a Plataforma VISIR.....	36
2.8 Projeto RELEDÉS.....	38
2.9 Comparação entre os Modelos Estudados.....	41
3 FUNDAMENTAÇÃO TEÓRICA.....	44
3.1 Redes de Computadores e a Internet.....	44
3.1.1 Camada de aplicação.....	47
3.1.2 Camada de transporte.....	49
3.1.3 Camada de rede.....	50
3.1.4 Camada de enlace.....	50
3.1.5 Camada física.....	51
3.2 Comunicação via Barramento USB.....	51
3.3 Sistemas Embarcados.....	53
3.3.1 Unidade central de processamento.....	54
3.3.2 Memória.....	55
3.3.3 Dispositivos de entrada/saída.....	55
3.3.4 Arduino.....	56
3.4 Computação Ubíqua e IoT.....	61

3.5 Ensino a Distância	62
3.6 Laboratórios Remotos.....	63
4 DESCRIÇÃO DO SISTEMA PROPOSTO	64
4.1 Firmware.....	64
4.2 Hardware	65
4.3 Software.....	65
4.3.1 Aplicação física.....	66
4.3.2 Central de controle	66
4.3.3 Interface gráfica de usuário	66
5 CONSIDERAÇÕES FINAIS	68
REFERÊNCIAS	65
ANEXO I	73
ANEXO II	75

1 INTRODUÇÃO

1.1 Considerações Iniciais

A difusão do computador pessoal, da Internet e, mais recentemente, dos dispositivos móveis e redes sem fio, nas últimas décadas, transformou substancialmente o modo como as pessoas interagem entre si e o ambiente a sua volta (BOLZANI, 2010). Aliado à isso, a constante modernização dos dispositivos eletrônicos e o advento dos equipamentos e ambientes “inteligentes” provocou a necessidade de se controlar dispositivos elétricos remotamente (LOUREIRO; BARRETO e BIBAS FILHO, 2009), tornando viável o conceito de uma aplicação pouco difundida nas instituições de ensino brasileiras, devido a pouca disponibilidade desse recurso voltado ao ensino prático ou em, laboratórios de experimentação remota (FELIPE; KELVIN e TEIXEIRA, 2010).

Esse corpo surgiu a partir da integração dos conceitos de acesso remoto e ensino à distância (GUIDA; NUNES e MOSCOSO, 2007) e ao observar a atual expansão dos sistemas embarcados no mercado, a possibilidade de integra-los à Internet, pelo uso do método adequado, tem obtido maior enfoque para as possíveis aplicações envolvidas, permitindo o vislumbre de uma nova perspectiva para o monitoramento e controle de processos (SANTANA, 2008).

Atualmente é de conhecimento comum à importância de experiências práticas no desenvolvimento de competências em engenharia, e os sistemas de informação têm sido de grande valia no ensino e aprendizado de processos acadêmicos, possibilitando o desenvolvimento de atividades que antes eram restritos a ambientes assistidos (BARRIOS *et al.*, 2013).

A Educação à Distância (EaD) tem evoluído muito nos últimos anos mostrando-se cada vez mais como uma aliada aos métodos de ensino. A ampla utilização de plataformas embarcadas associadas ao potencial das redes se apresenta como uma alternativa muito próxima da realidade como dito por Beetner *et al.* (2000 *apud.* CALLAGHAN *et al.*, 2007). Posto que os laboratórios de experimentação remota, descritos como uma extensão interativa dos seus equivalentes físicos por Marchezan *et al.* (2004 *apud.* GUIDA; NUNES e MOSCOSO, 2007), vêm integrando-se como parte dessa evolução.

Dito isso, o uso e desenvolvimento de interfaces de sistemas remotos no monitoramento e controle de processos industriais têm sido amplamente abordados (SANTOS; SÁ, 2012), principalmente em atividades onde é necessária a substituição da presença do homem, pois

dessa forma o usuário poderá realizar o acionamento de um equipamento qualquer sem a necessidade de ir ao local do dispositivo. Isso acarretará numa drástica redução dos riscos à vida em ambientes com índices de periculosidade elevados, dentre outros benefícios.

1.2 Origem do Trabalho

O crescente uso e desenvolvimento dos sistemas de controle e automação, em ambientes industriais, têm direcionado pesquisadores da área a buscar um melhor entendimento e aperfeiçoamento na aplicação de tais técnicas, pois os avanços prático-teóricos obtidos em controle automático permitem que seja alcançado o ótimo desempenho dos sistemas dinâmicos estudados (OGATA, 2010).

Tendo isso em mente, a disciplina de microcontroladores aplicados a engenharia de controle e automação, código CAT341, da grade curricular do curso de engenharia de controle e automação da Universidade Federal de Ouro Preto desenvolveu um trabalho prático sobre acesso remoto com base em sistemas microcontrolados. Assim, o presente trabalho teve origem a partir do interesse do autor em projetar, pesquisar e desenvolver sistemas inovadores com as tecnologias de acesso remoto associadas a sistemas embarcados.

1.3 Objetivos

1.3.1 Objetivo geral

O objetivo deste trabalho é propor um método genérico que permita o desenvolvimento de um sistema de acesso remoto, acessível pela Internet ou pela rede local, capaz de realizar acionamentos de potência e leitura de sensores acoplados à bancada de um sistema de controle e monitoramento, que viabilize a realização de ensaios e ajuste dos parâmetros que envolvem o controle à distância, para que seja possível a utilização remota dos recursos oferecidos pelo laboratório.

1.3.2 Objetivos específicos

Para a realização do objetivo proposto, as etapas descritas a seguir foram concebidas:

- Fazer uma revisão bibliográfica atualizada sobre sistemas de acesso remoto para acionamento e monitoramento de plataformas microcontroladas voltadas para o ensino;
- Expor os principais conceitos que envolvem as abordagens apresentadas na revisão, elaborando dessa forma, uma fundamentação teórica apropriada para o presente trabalho;
- Elaborar um método genérico para um sistema de acesso remoto conectado a uma plataforma embarcada, fazendo uso da arquitetura cliente-servidor, que possibilite o uso de recursos do laboratório remotamente;
- Propor um sistema de monitoramento e controle virtual via web;
- Sugerir um sistema de visualização em tempo real.

1.4 Justificativa

O uso da Internet até agora, tem sido orientado principalmente à conexão de dispositivos pessoais, mas em breve todos os tipos de aparelhos com “computadores integrados” vão trocar informações através da Internet (CAN FILIBELI; OZKASAP e REHA CIVANLAR, 2007).

Por muito tempo os esquemas de comunicação tradicionais têm feito uso de protocolos heterogêneos de softwares e interfaces de usuário, o que vinha dificultando a interação de dispositivos à Internet. Em contrapartida, a evolução dos dispositivos embarcados alterou a forma dos usuários de interagir e gerenciar tais dispositivos, ampliando sua necessidade de acessar facilmente os equipamentos públicos, seja quais forem as suas escolhas de implantação. Dessa forma, hoje mais e mais dispositivos estão funcionando de forma ubíqua ao nosso redor. A isso a literatura tem atribuído o nome de dispositivos globais de interconexão à Internet das Coisas (IoT) (DUQUENNOY; GRIMAUD e VANDEWALLE, 2009).

Neste sentido, os Laboratórios Remotos que usam este tipo de tecnologia, e são aplicados à operação remota de processos reais, ampliam as suas capacidades de ensino para além do âmbito dos laboratórios convencionais ao ensinar um determinado assunto (NETO *et al.*, 2013).

Os laboratórios convencionais apresentam limitações de recursos impostas pelo tempo, número de experimentos físicos disponíveis e número de usuários por experimento. Dessa forma, propostas alternativas sobre esse tema sugerem a operação remota de processos físicos, principalmente nas universidades que muitas vezes não possuem um quadro didático-

pedagógico bem estruturado, que é essencial para conduzir de forma adequada experiências remotas no controle automático (BARRIOS *et al.*, 2013).

Sendo assim, os detalhes dessa modalidade merecem cuidado extra em sua implantação, necessitando de um conjunto de ferramentas que façam com que o experimento realizado à distância se aproxime o máximo possível do experimento presencial como mencionado por Soares (2001, *apud*. FELIPE; KELVIN e TEIXEIRA, 2010).

Dito isso, pensou-se em propor um sistema que conecte a plataforma embarcada Arduino a um servidor Apache 2 por meio das portas seriais USB da plataforma e do computador no qual o servidor for instalado. Além disso, usando-se o sistema operacional Linux com distribuição Ubuntu versão 12.04 (Canonical Ltd.,www.ubuntu.com) e possibilitando o acesso remoto a essa plataforma e a bancada a ela acoplada, viabilizando o conceito de laboratório remoto.

Dessa forma, visa-se abordar as seguintes áreas do curso de Engenharia de Controle e Automação:

- Eletromecânica;
- Programação e redes de computadores;
- Sistemas embutidos;
- Aplicação da interface homem-máquina.

1.5 Estrutura do Trabalho

O presente texto é composto por cinco capítulos onde, no capítulo 1 é apresentada uma introdução contendo alguns dos conceitos utilizados, por parte da literatura abordada, no desenvolvimento do presente trabalho, expondo a origem, os objetivos e a relevância científica do projeto.

No capítulo 2 é feita uma abordagem histórica atualizada, considerando-se as escolhidas como as mais relevantes para o tema do presente texto, pontuando-se brevemente os principais métodos e recursos utilizados pelos autores dos trabalhos e projetos anteriores.

O capítulo 3 traz uma descrição dos conceitos utilizados pela literatura apresentada no capítulo anterior, e considerados fundamentais para o desenvolvimento da base teórica deste trabalho.

No capítulo 4 tem-se o conjunto de métodos e recursos propostos como sugestão que, teoricamente, ao ser colocado em execução possibilita que o projeto discutido por este texto seja viabilizado.

O capítulo 5 realiza a conclusão do trabalho, inserindo-se as considerações finais da abordagem feita conforme discutido pela proposta bem como as sugestões para trabalhos futuros.

2 ESTADO DA ARTE

Para a elaboração da presente seção foi necessário efetuar uma pesquisa no âmbito dos projetos relacionados ao tema deste texto. O conjunto de métodos descritos a seguir pertence à aqueles que mais se assemelham com a proposta deste trabalho, por apresentarem abordagens e finalidades muito próximas.

Nesta análise histórica, compreendida entre os anos 2000 e 2015, primeiramente tem-se em evidência a proposta apresentada na dissertação desenvolvida por Silva (2002), na qual foi enfatizada a viabilidade de se desenvolver e implantar um micro servidor web (MSW), utilizando em sua abordagem um sistema embutido, de dimensões reduzidas, versátil, expansível e microcontrolado, com a capacidade de acionar outros dispositivos elétricos.

2.1 Micro Servidor Web

Na linha de trabalho aqui apresentada, tem-se o desenvolvimento teórico-prático para conectar dispositivos embarcados a uma rede padrão Ethernet, fazendo uso dos protocolos TCP/IP (*Transmission Control Protocol/ Internet Protocol*) e um microcontrolador para a comunicação de dados, ao qual denominou-se micro servidor web (MSW). Possibilitou a obtenção do controle e monitoramento remoto destes dispositivos de uma forma segura, eficaz e econômica, por meio da utilização de um navegador padrão (*browser*) para Internet (SILVA, 2002).

Para modelar a aplicação objeto do referenciado estudo o autor realizou uma pesquisa sobre os tipos de tecnologias associadas, levando-o aos esquemas representados nas figuras 1 e 2.

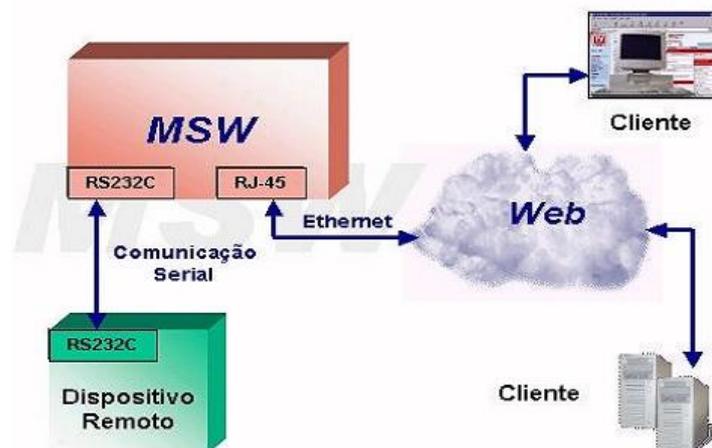


FIGURA 1 - Descrição detalhada implementação geral do sistema

Fonte: SILVA, 2002, p. 14.

Na Figura 1 é possível observar que para atingir o MSW, Silva (2002) define a necessidade de os aparelhos que compõe o sistema disporem de uma interface serial RS232 padrão (*Recommended Standard 232*), pois por esta interface pode ser feita a aquisição de dados, e também de uma interface responsável por conectar o MSW com a Internet, em que das duas opções pesquisadas no trabalho em questão optou-se pelo uso da Ethernet devido às condições técnicas e logísticas.

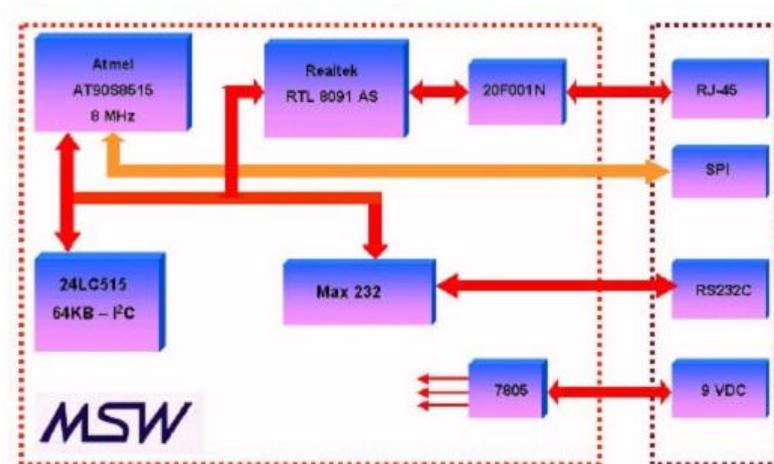


FIGURA 2 - Modelo Proposto
Fonte: SILVA, 2002, p. 43.

Já na Figura 2 tem-se representação do modelo proposto do MSW com a relação dos componentes essenciais utilizados na construção do hardware desse sistema embutido, que é composto por quatro circuitos integrados e mais alguns componentes discretos (SILVA, 2002).

Sendo os circuitos integrados (CI), um microcontrolador AT90S8515 da Atmel responsável pelo gerenciamento das funcionalidades do sistema, um controlador RTL 8091 AS voltado para o acesso à ethernet e conectado diretamente às portas do MCU citado. Uma memória I²C (*Inter Integrated Circuit*) EEPROM serial (*Electrically Erasable Programmable Read Only Memory*) modelo 24LC515, fabricada pela empresa Microchip, encarregada do armazenamento das aplicações do sistema, funcionando como um tipo de sistema de arquivos do MSW (SILVA, 2002).

Para a comunicação serial e aquisição de dados do sistema, foi adicionado um circuito integrado MAX 232 fabricado pela Maxim (SILVA, 2002). Os componentes discretos que integram a parte eletrônica do projeto estão apresentadas na Figura 3.



FIGURA 3 - Produto final
Fonte: SILVA, 2002, p. 46.

No desenvolvimento do firmware foi utilizado um kernel de organização simples como o da Figura 4, no qual o controle ethernet dispõe de suporte para as camadas mais baixas de controle, enquanto que o MCU se encarrega das superiores. Nas camadas inferiores estão contidos o driver do adaptador de rede e a pilha TCP/IP e na camada de aplicação está o servidor web HTTP (*HyperText Transfer Protocol*) (SILVA, 2002).



FIGURA 4 - Camadas do firmware utilizado no MSW
Fonte: SILVA, 2002, p. 50.

Assim, o MSW adota um padrão de firmware contendo um debug minúsculo, um interpretador para pseudocódigo (pcode, usado para a programação das aplicações), um driver adaptador de rede, uma pilha TCP/IP e um servidor HTTP. Logo, quando conectado à rede, o firmware do MSW é capaz de suportar uma série de protocolos de rede. Isso é detalhado como o software básico do sistema (SILVA, 2002).

2.2 Projeto DIESEL

Já Callaghan *et al.* (2007) após um levantamento, sobre as tecnologias utilizadas na implantação de recursos de acesso remoto em experimentos, realizado em meados de 2002, desenvolve uma arquitetura e uma metodologia genéricas para acesso/controlado remoto, que viabilizasse esse tipo de recurso, integrando eficientemente instrumentação e componentes de equipamentos experimentais, batizando esse sistema de DIESEL (*Distance Internet-Based Embedded System Experimental Laboratory*).

Esse modelo é composto por um servidor gateway conectado à Internet que executa a função de administrador do laboratório, além de, possuir um número de estações de trabalho experimentais, as quais estão conectadas ao servidor por meio de uma rede regulada através da utilização de um ponto central de distribuição (*hub*) como indicado na Figura 5 (CALLAGHAN *et al.*, 2007).

Assim, essa arquitetura está funcionalmente dividida em três subsistemas que interagem entre si, sendo um servidor com sistema de reserva acessível através da web, um aplicativo cliente que o usuário final instala em seu computador pessoal, e um aplicativo servidor instalado em cada estação de trabalho remota (CALLAGHAN *et al.*, 2007).

A aplicação do servidor é acessada por meio da aplicação do cliente, garantindo ao usuário acesso e controle de todas as funcionalidades do sistema necessárias à conclusão de uma sessão de laboratório em uma das estações de trabalho remotas (CALLAGHAN *et al.*, 2007).

Ainda segundo o autor, cada estação de trabalho individual é idêntica e acomoda uma variedade relacionada de ferramentas de hardware e software experimentais, necessários para realizar experimentos práticos.

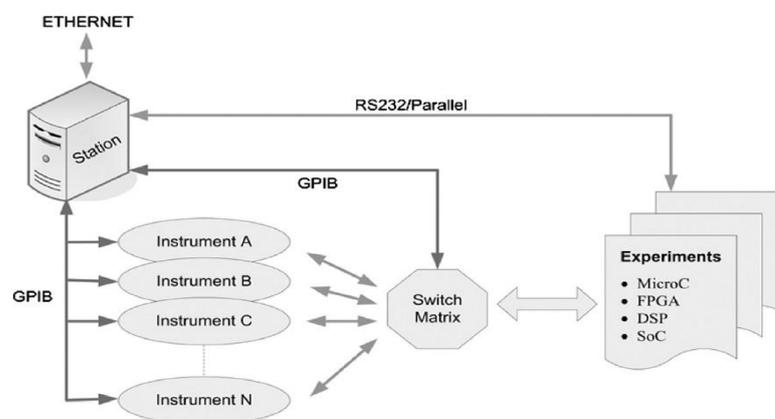


FIGURA 5 - Arquitetura DIESEL genérica
Fonte: CALLAGHAN *et al.*, 2007, p. 1298.

Os instrumentos de teste são configurados e controlados a partir da estação de trabalho usando-se o protocolo *General Purpose Interface Bus* (GPIB), enquanto as placas experimentais são acessadas tanto por meio da interface serial RS232 como através de conexões paralelas, conforme for requerido. Uma matriz de comutação controlada GPIB permite que os instrumentos de teste sejam ligados a um número de pontos de teste nas placas experimentais conforme a necessidade do experimento (CALLAGHAN *et al.*, 2007).

Assim, a abordagem cliente-servidor DIESEL utiliza uma arquitetura de software distribuída desenvolvida por meio da tecnologia Microsoft .NET, conforme pode ser verificado na Figura 6 (CALLAGHAN *et al.*, 2007).

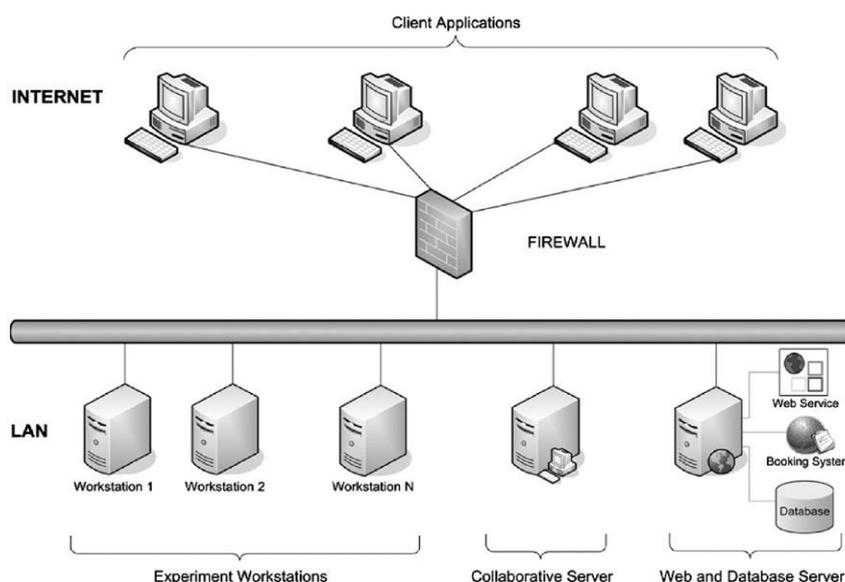


FIGURA 6 - Ambiente integrado de aprendizagem DIESEL com trabalho colaborativo
Fonte: CALLAGHAN *et al.*, 2007, p. 1302.

Com a finalidade de garantir a entrega segura de comandos e dados entre o aplicativo cliente e o aplicativo servidor, na estação de trabalho, o autor estabelece o uso da comunicação TCP *Peer-to-Peer* (P2P) direta operando mais de um canal remoto .NET protegido com criptografia 256 bits (CALLAGHAN *et al.*, 2007).

Para evitar problemas de acesso com firewalls e problemas de plataforma cruzada, as interações com o serviço web ocorrem através de HTTP e trocas de mensagens e dados abordando o *Simple Object Access Protocol* (SOAP) (CALLAGHAN *et al.*, 2007). Na Figura 7 tem-se o diagrama de funcionamento dos protocolos de comunicação e do fluxo de dados do sistema.

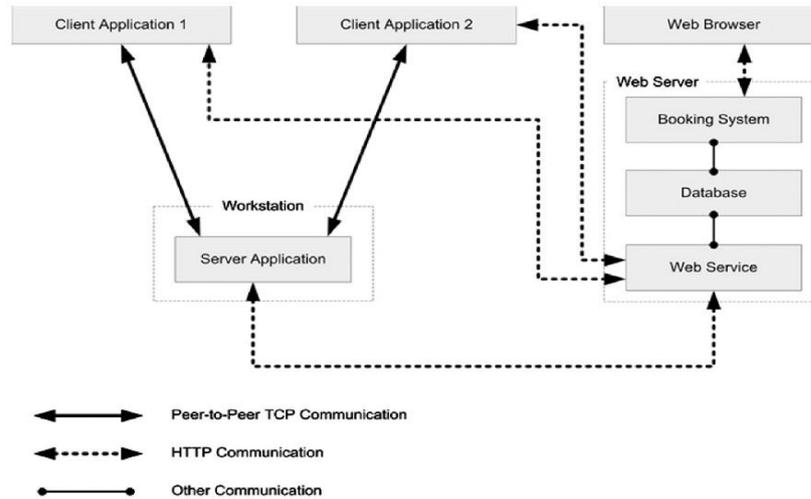


FIGURA 7 - Protocolos de comunicação na arquitetura DIESEL com trabalho cooperativo
 Fonte: CALLAGHAN *et al.*, 2007, p.1303

Este sistema foi modelado em quatro camadas de comunicação, sendo elas a camada de apresentação, a camada de dados, a camada de negócios e a camada física (CALLAGHAN *et al.*, 2007). Conforme mostrado na Figura 8.

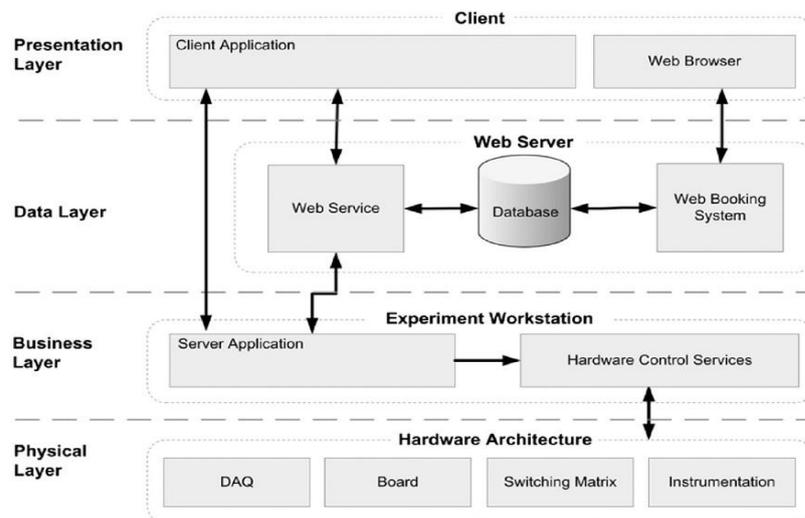


FIGURA 8 - Modelo de comunicação com quatro camadas DIESEL
 Fonte: CALLAGHAN *et al.*, 2007, p. 1300.

A camada de apresentação consiste na aplicação cliente DIESEL e no sistema de reserva, que é acessado através de um navegador web. O cliente DIESEL fornece a interface que permite ao usuário configurar e manipular sistemas e instrumentos embutidos remotamente. O sistema de reservas oferece uma interface que permite o usuário realizar e gerenciar reservas.

A camada de dados provê acesso ao banco de dados, seja por meio do serviço web ou pelo sistema de reserva web. A camada de negócios está na aplicação servidor DIESEL, e fornece

acesso e controle para a camada física. A camada física implica em todos os recursos de hardware (placas experimentais e instrumentos de teste). Um serviço web é usado como um gateway entre as camadas de apresentação, negócios e dados que permita às aplicações cliente e servidor acessarem o banco de dados, pois dessa forma é possível separar os aplicativos cliente e servidor do processo de armazenamento de dados (CALLAGHAN *et al.*, 2007).

A aplicação do servidor (na camada de negócios) responde a comandos da aplicação cliente através da execução dos programas de controle adequados na arquitetura de hardware (camada física) ao configurar os circuitos embutidos, os roteadores de sinal e instrumentos enquanto envia comandos para o circuito em teste (CALLAGHAN *et al.*, 2007).

2.3 Sistema de Acesso Remoto à Experiências de Eletrônica

No sistema de acesso remoto à experiências de eletrônica desenvolvido por Guida, Nunes e Moscoso (2007), a estação do laboratório e estação de serviço são as grandes partes que o compõe. Vislumbrando a expansão do sistema com a inclusão de novos experimentos sem alteração da estrutura base, o proposto sistema foi planejado para ser modular, estando sua organização apresentada na figura 9.

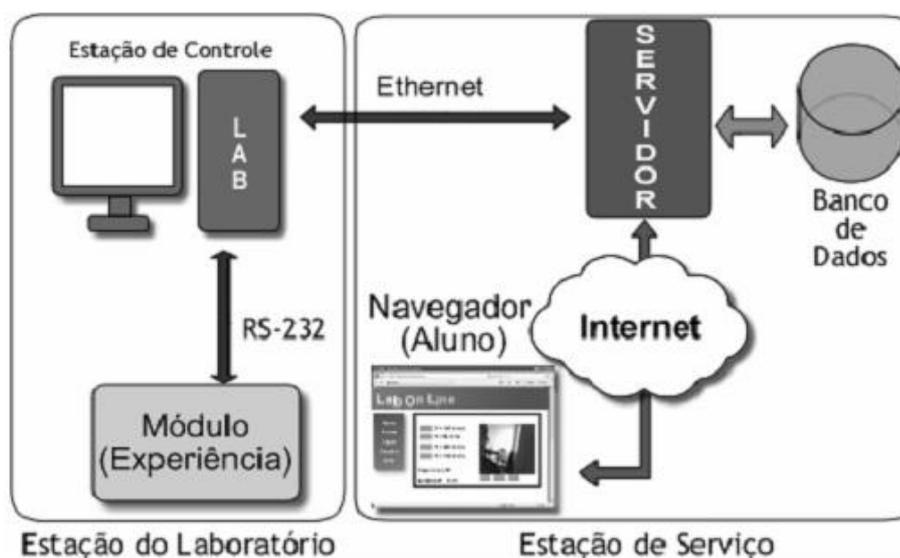


FIGURA 9 - Estrutura do Sistema
Fonte: GUIDA; NUNES e MOSCOSO, 2007, p. 2.

Dividiu-se a estação do laboratório em estação de controle e módulo de experiência. Assim, é possível a inserção de uma nova experiência apenas substituindo-se o módulo e informando as

configurações da experiência no banco de dados localizado no servidor (GUIDA; NUNES e MOSCOSO, 2007).

Na estação de controle ocorre o controle da experiência e da comunicação entre esta e o servidor. A comunicação é realizada por meio do protocolo RS232 serial, e é utilizada como forma de aquisição de dados e envio de comandos. Com o servidor a comunicação foi fundamentada no protocolo Ethernet, ocorrendo via soquete, e para a conversão do sinal padrão RS232 para sinal TTL (*Transistor-Transistor Logic*), um circuito integrado MAX232 foi utilizado entre a porta serial do computador e o módulo experimental a ele conectado (GUIDA; NUNES e MOSCOSO, 2007).

Toda operação executada pode ser armazenada em formato de arquivo de texto (extensão *.txt) em local previamente especificado, pois a estação de controle gera um registro na área de log que pode ser salvo. Utilizando-se a o ambiente integrado de desenvolvimento (IDE) Netbeans 5.5.1 e a linguagem JavaTM o software de controle foi desenvolvido (GUIDA; NUNES e MOSCOSO, 2007).

Para uma melhor compreensão a estação de serviço foi dividida em servidor e sistema web, apesar da última ser parte da primeira. O servidor contém o banco de dados e o container do sistema web. Todas as informações do sistema estão armazenadas no banco, desde os cadastros de usuários até os dados gerados por cada experiência realizada. Sendo administrado pelo sistema de gerenciamento de banco de dados (SGBD) do MySQL 5.0.45 community-nt.

O container escolhido foi o Tomcat 5.5.17. Container é uma aplicação Java responsável pela gestão de chamada dos *servlets* (BASHMAN; SIERRA e BATES, 2005, apud., GUIDA; NUNES e MOSCOSO, 2007). Na abordagem dos autores o contêiner armazenava as páginas e controlava as solicitações do sistema web.

O sistema web é o responsável pela interação dos usuários com os demais componentes do sistema e foi desenvolvido em linguagem JavaTM, por meio das tecnologias *Java Server Pages* (JSP) e *Java Server Faces* (JSF) para web, com o *plugin visual web pack* da IDE Netbeans 5.5.1. A interface web é dividida em uma seção dedicada aos professores e outra aos alunos, onde o professor, por meio da sua seção, disponibiliza uma experiência aos alunos, especificando o prazo para a execução da mesma e também possibilita ao professor acesso a todos os relatórios gerados pelos alunos (GUIDA; NUNES e MOSCOSO, 2007).

O modelo proposto requer um operador no laboratório para realizar a inicialização do servidor, o seu controle de requisições e fornecer suporte técnico a estação do laboratório (GUIDA; NUNES e MOSCOSO, 2007).

2.4 Laboratório de Experimentação Remota com Microcontroladores PIC

Outro estudo significativo para o presente trabalho foi desenvolvido por Felipe, Kelvin e Teixeira (2010), no qual se utiliza uma arquitetura com base em um microservidor construído com um microcontrolador da família PIC da empresa microchip. Essa estrutura está detalhada na Figura 10.

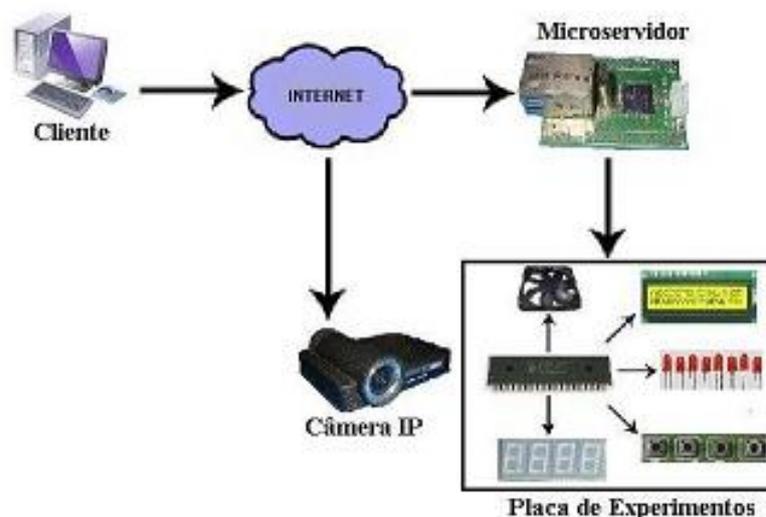


FIGURA 10 - Arquitetura definida
Fonte: FELIPE; KELVIN e TEIXEIRA, 2010, p. 6.

Por meio da Figura 10 é possível perceber que o citado estudo adota uma placa de experimentos que possui um microcontrolador, especificado pelos autores como sendo da linha PIC16F, como base. Além de poder ser utilizada localmente, a placa de experimento também foi projetada para ser controlada e observada à distância, sendo constituída de um conjunto de periféricos de oito LED's, quatro displays de sete segmentos, um display LCD, um motor de corrente contínua e quatro botões que podem ser acionados a partir da página do experimento (FELIPE; KELVIN e TEIXEIRA, 2010).

Assim, é possível a utilização da interface de E/S do microcontrolador para acionamento de dispositivos, leitura de botões, exibição de dados em display de sete segmentos e LCD, e aplicação do módulo PWM (*Pulse Width Modulation*) para acionamento de motor, como atividades a serem executadas (FELIPE; KELVIN e TEIXEIRA, 2010).

Para efetuar a realimentação do sistema do referenciado trabalho, foi adotado o modelo de câmera IP para monitorar o hardware previamente descrito. Esse modelo foi selecionado por ser uma câmera digital com interface de rede padrão ethernet, ou seja, por não depender de

um computador para disponibilizar imagens. Dessa forma, as alterações sofridas pelo mesmo são enviadas à interface cliente permitindo a obtenção dos resultados do experimento em tempo real (FELIPE; KELVIN e TEIXEIRA, 2010).

O sistema funciona com o envio de um arquivo de programação pelo cliente, quando este acessa a página do experimento, à placa de experimentos que está conectada ao microservidor. Concluído o processo de gravação, o experimento passa a ser controlado por meio da página e monitorado pela câmera IP (FELIPE; KELVIN e TEIXEIRA, 2010).

A página do experimento é composta por um pequeno texto informativo acerca do projeto, uma área destinada ao envio do arquivo de programação, outra com os botões que executam o experimento, um link direcionado às imagens da câmera e um formulário que permite acesso a uma página onde se encontram as configurações de rede do microservidor conforme segue a Figura 11 (FELIPE; KELVIN e TEIXEIRA, 2010).



FIGURA 11 - Página do experimento
Fonte: FELIPE; KELVIN e TEIXEIRA, 2010, p. 7.

A fim de desenvolver aplicações de alto grau de interatividade os botões foram concebidos utilizando-se a técnica *Assynchronous Javascript and XML* (AJAX). Ela possibilita que os serviços web sejam requisitados de maneira assíncrona, ou seja, ao disparar uma determinada ação não é necessário que o cliente aguarde por uma resposta do servidor para que o experimento continue em funcionamento, e quando a resposta é recebida, a página não seja totalmente recarregada. Apenas o que for afetado será modificado, aproximando o experimento remoto o máximo possível de um experimento real (FELIPE; KELVIN e TEIXEIRA, 2010).

2.5 Sistema de Laboratório Acadêmico Remoto Multiusuário

Em 2013 um estudo propôs uma arquitetura multiusuário. Esse modelo foi estruturado em três camadas de rede independentes (BARRIOS *et al.*, 2013), conforme demonstrado pela Figura 12.

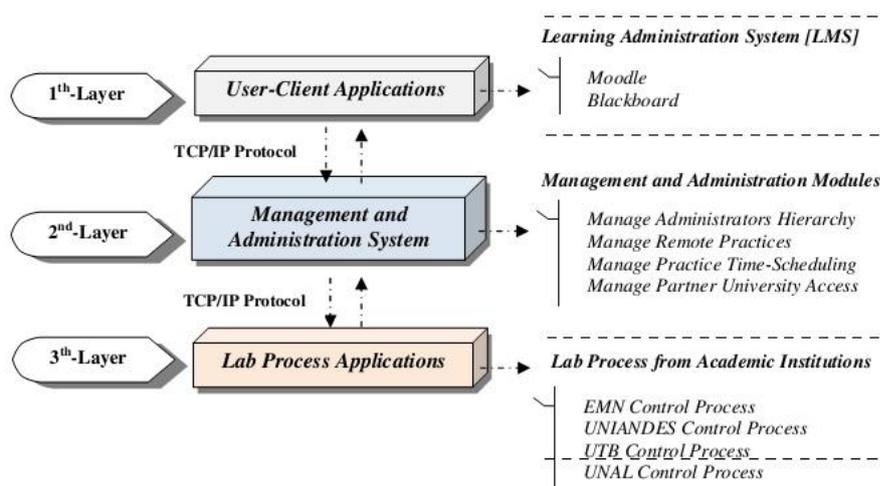


FIGURA 12 - Relações globais para a arquitetura do sistema de rede proposto
Fonte: BARRIOS *et al.*, 2013, p. 113.

Na Figura 12 tem-se as relações globais de cada uma das três camadas, onde a primeira camada de rede, denominada UCA (*User-Client Applications*), está vinculada ao sistema administrativo de aprendizagem, dito LMS, que cada universidade utiliza para gerir academicamente o controle de seus próprios cursos. Na terceira camada, está uma interface de operação remota responsável por manipular os LPA's que estão associados ao controle de processos reais de uma dada instituição (BARRIOS *et al.*, 2013).

A fim de possibilitar a fácil integração de novas instituições e processos de controle, um sistema de gestão e administração (MAS), foi inserido na segunda camada rede, tendo como principal função ser responsável pelos recursos do laboratório, ao mesmo tempo em que age como uma ponte de intercomunicação entre a primeira e a terceira camadas de rede (BARRIOS *et al.*, 2013).

Também é possível observar através da Figura 12, a descrição do processo de interação geral entre cada uma das camadas de rede para o sistema proposto. No qual, para a terceira camada de rede, a interface de operação remota GUI foi desenvolvida. Essa interface está conectada a um aplicativo de controle local que efetua a manipulação do processo real por meio de um protocolo TCP/IP com soquetes de dados (BARRIOS *et al.*, 2013).

Com a interface remota pronta, ela pôde ser carregada para a segunda camada e gerenciada pelo MAS no que tange a acessibilidade e o planejamento de tempo. Sendo aplicada a um curso pelo LMS na primeira camada de rede, dessa forma, a interface de operação remota deve ser previamente reservada, de modo que ao ser acessada, o usuário final possa realizar o trabalho laboratorial remoto (BARRIOS *et al.*, 2013).

Nesta abordagem fez-se necessário o estabelecimento de uma interconexão entre a aplicação desenvolvida em LabVIEW e a aplicação gráfica desenvolvida em Java, na terceira camada. Para o desenvolvimento dessa interligação foi utilizada a aplicação de servidor JIL (*Java Internet Labview*) (VARGAS, 2010 *apud.* BARRIOS *et al.*, 2013), que se baseia no uso de um módulo de comunicação genérico capaz de instalar a camada de comunicação necessária para permitir o controle de instrumentos virtuais locais através da Internet (BARRIOS *et al.*, 2013). Na Figura 13 detalha-se essa interconexão.

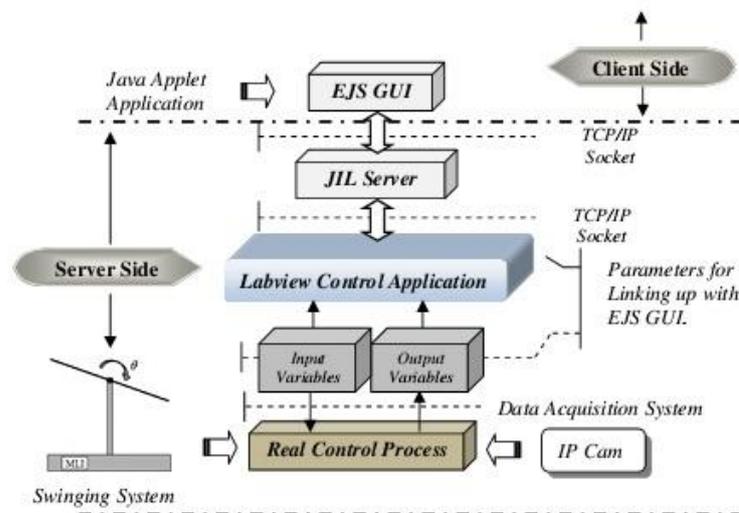


FIGURA 13 - Estrutura geral de interconexão para o componente da 3ª camada de rede
Fonte: BARRIOS *et al.*, 2013, p. 114.

Por meio da Figura 13 é possível visualizar que para realizar as interconexões necessárias entre as aplicações de interface gráfica (GUI) desenvolvidas em *Easy Java Simulation* (EJS) e em LabVIEW (plataforma em que está lotado o processo real). O estudo declara a ocorrência de trocas de variáveis de entrada e saída para permitir que os parâmetros que se deseja manipular na interface de operação remota sejam ligados. Isso foi projetado levando-se em consideração as especificações de cada variável e que tipo de função cada uma desempenha (BARRIOS *et al.*, 2013). Esse processo pode ser melhor visualizado na Figura 14.

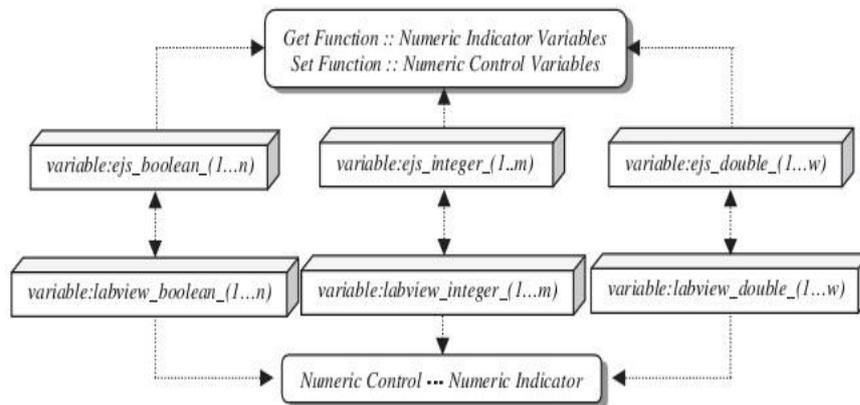


FIGURA 14 - Esquema de interconexão para a troca de variável entre o aplicativo de controle local e a interface gráfica

Fonte: BARRIOS *et al.*, 2013, p. 114.

A fim de possibilitar a interação remota da aplicação local com o EJS, tal aplicação (GUI) foi criada como um arquivo de distribuição fonte. Dessa forma, o servidor JIL pode chamá-lo, executá-lo e controlá-lo através do protocolo de comunicação TCP/IP. Uma vez feito isso, a aplicação desenvolvida foi carregada no Sistema de Gestão e Administração MAS. Na Figura 15 está detalhada a estrutura geral definida para a interação entre os componentes de rede da terceira camada e do MAS (BARRIOS *et al.*, 2013).

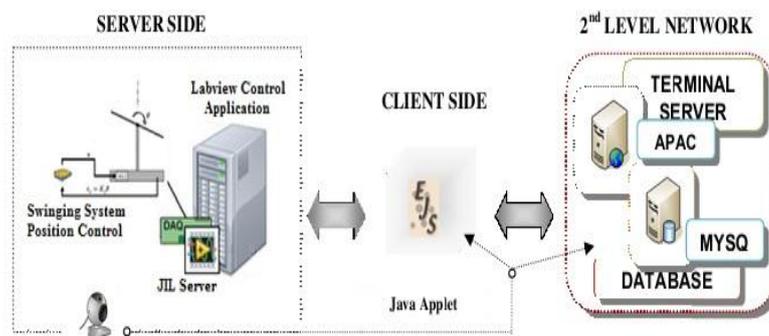


FIGURA 15 - Estrutura geral entre os componentes da 3ª camada e o MAS na 2ª camada

Fonte: BARRIOS *et al.*, 2013, p. 115.

Dá um dos principais requisitos na segunda camada para o sistema de gestão e administração é permitir a integração de novas Universidades e novas práticas remotas. Isso envolve a gestão adequada de recursos (para evitar problemas com fusos horários), prevenção de conflitos de usuário (quando duas Universidades tentam acessar a mesma prática) e restrição de acesso a potenciais intrusos.

O MAS foi desenvolvido com base na arquitetura de software MVC (*Model-View-Controller*), sendo o sistema centralizado através de um servidor Apache com banco de dados

MySQL localizado em uma instituição remota, e constituído de quatro módulos específicos definidos pelo autor e seus colaboradores como: módulo gerenciador de administradores, módulo gerenciador de práticas, módulo gerenciador de universidades e o módulo gerenciador de reservas. Cada módulo tem uma função específica sendo dividida em duas opções, criar/editar um recurso e acessar os recursos já criados (BARRIOS *et al.*, 2013).

Por fim, o objetivo da primeira camada é apresentar a prática remota para o utilizador final, independentemente do dispositivo a ser utilizado, através do GUI. Normalmente, segundo o estudo, a interface pode ser acessada como uma página web independente ou ser integrada a outras plataformas como o LMS (BARRIOS *et al.*, 2013).

Para utilizar as práticas dentro do Moodle ou Blackboard, o MAS cria dinamicamente um endereço URL (*Uniform Resource Locator*) com as informações definidas e a aplicação carregada pelo Administrador. Este URL contém o endereço da prática e o período de tempo programado pelo módulo gerenciador de reservas, para os alunos realizarem a prática. Dessa forma, a única preocupação do administrador é criar a atividade ou funções em Moodle ou Blackboard e introduzir a URL para permitir o acesso (BARRIOS *et al.*, 2013).

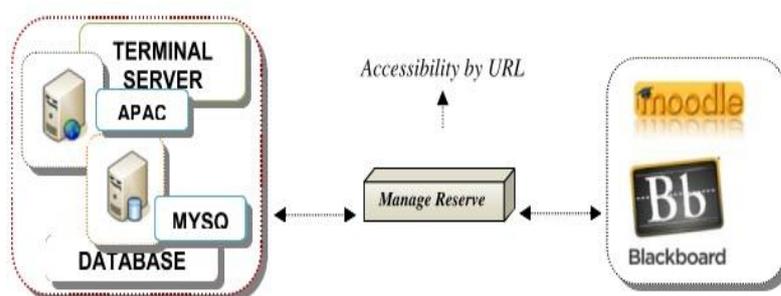


FIGURA 16 - Interação entre o MAS e os LMS para a utilização de um laboratório remoto através do sistema
Fonte: BARRIOS *et al.*, 2013, p. 116.

Na Figura 16 tem-se a descrição da interação da interface de práticas, que consiste na aplicação desenvolvida com o EJS, que permite operar remotamente a prática e um streaming de vídeo em tempo real para visualizar as reações do processo de controle ao ser manipulado (BARRIOS *et al.*, 2013).

Nessa abordagem o autor e seus colaboradores destacam o MAS como núcleo da arquitetura estudada, por garantir a correta utilização dos recursos e por este funcionar como uma ponte entre a camada de controle de processos e a camada aplicação do usuário (BARRIOS *et al.*, 2013).

2.6 Experimento Educacional Remoto

Ainda em 2013, o estudo desenvolvido por Neto e seus colaboradores tem na Figura 17 o diagrama básico da arquitetura do experimento remoto, no qual as principais características relativas à estrutura proposta estão detalhadas abaixo.

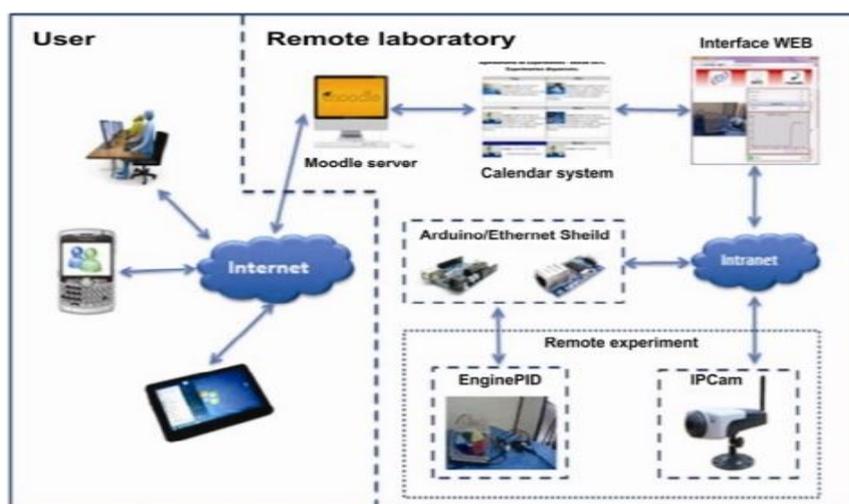


FIGURA 17 - Experimento
Fonte: NETO *et al.*, 2013, p. 48.

O hardware desta abordagem é constituído de um Arduino Uno, uma placa eletrônica com base nos circuitos integrados ATmega328 da Atmel. Ele controla o motor elétrico através da leitura do tacômetro. Para realizar as conexões do experimento proposto, a Internet é utilizada como meio de comunicação entre o usuário e o experimento educacional remoto, sendo necessário adicionar à placa Arduino Uno uma placa eletrônica auxiliar denominada Ethernet *shield*, que tem como principal componente o circuito integrado ENC28J60 fabricado pela Microchip (NETO *et al.*, 2013).

O ambiente virtual foi desenvolvido em linguagem Java, sendo o local utilizado para criar uma conexão entre o usuário e o experimento remoto. Permitindo-se ao usuário interagir com o experimento por meio de comandos disponíveis na aplicação, o que permitiu a realização da experiência em ambiente virtual, utilizando-se a Internet como meio de comunicação para receber informações do experimento e controlá-lo. Esta aplicação permitiu a hospedagem em uma página web semelhante ao da Figura 18. Dessa forma, esta aplicação foi instalada no computador do usuário, permitindo a utilização de todos os recursos do aplicativo por meio do próprio navegador (NETO *et al.*, 2013).

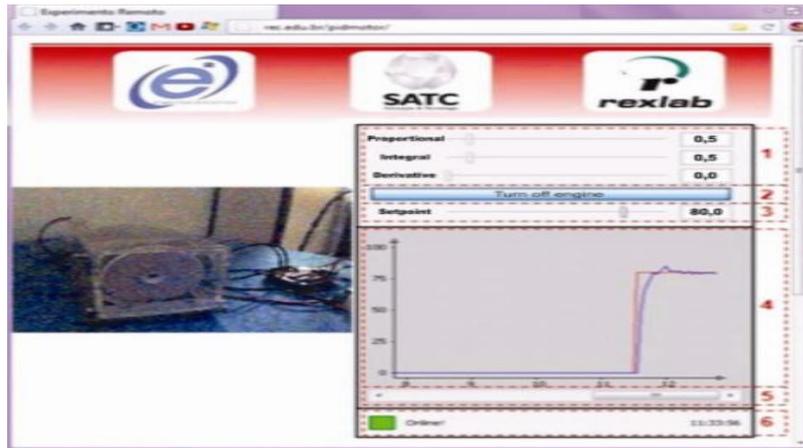


FIGURA 18 - Ambiente de Interação do experimento remoto
Fonte: NETO *et al.*, 2013, p. 48.

Como mostrado na Figura 18, na página web o motor de corrente contínua pôde ser visualizado por meio de uma webcam, o que permitiu ao usuário visualizar a rotação do motor. Além disso, a aplicação possibilitou que o motor fosse ligado e desligado, bem como habilitou as funções de modificar os parâmetros do controlador PID e a de visualizar a rotação do motor em um gráfico (NETO *et al.*, 2013).

Essa aplicação possui vários mecanismos de interação com o usuário. As funções de cada item são apresentadas e exibidas, detalhando-se o efeito de cada um dos blocos no ambiente desenvolvido para a aplicação, na Tabela 1 (NETO *et al.*, 2013).

Tabela 1 - Descrição dos Blocos no Ambiente de Desenvolvimento

Bloco	Nome	Propósito
1	Parâmetros do Controlador PID	Selecionar os valores dos parâmetros do controlador PID (k_p , k_i e k_d).
2	Chave Liga/Desliga	Ligar ou desligar o motor.
3	Seletor de <i>Setpoint</i>	Escolher o valor de referência desejado para a velocidade do motor.
4	Gráfico de Rotação	Visualizar os valores da velocidade do motor e do <i>Setpoint</i> ao longo do tempo.
5	Gráfico de Barras	Paralisar o gráfico para visualizar os valores anteriores de velocidade.
6	Estado da Aplicação	Mostrar o tempo e as mensagens da operação da aplicação.

Fonte: Adaptado de NETO *et al.*, 2013, p. 48.

Para o desenvolvimento do software a ser inserido no microcontrolador, da plataforma embarcada anteriormente citada, o autor e seus colaboradores destacam dois pontos das características presentes na estrutura do software. O primeiro ponto é a estrutura de controle que foi utilizada, o PID, como base para o desenvolvimento do software. Na figura 19 está ilustrado o diagrama de blocos de um sistema de controle de circuito fechado utilizando um controlador PID genérico (NETO *et al.*, 2013).

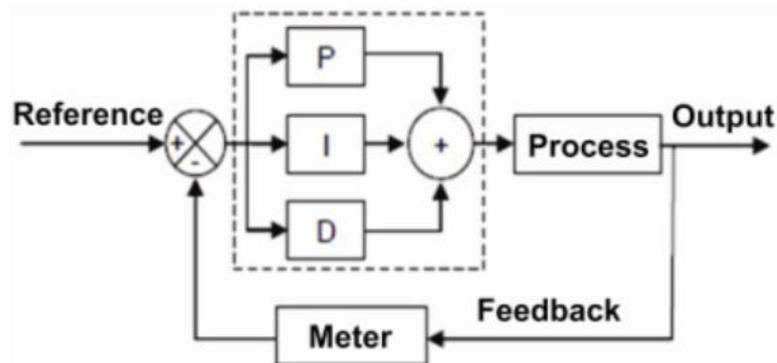


FIGURA 19 - Diagrama de Blocos de um Controlador PID
Fonte: NETO *et al.*, 2013, p. 49.

O diagrama de blocos do controlador PID exibido acima, tem um adicionador que integra os controladores das ações proporcional, integral e derivativa, que resultaram em um controlador PID em paralelo, o que originou a equação a seguir (NETO *et al.*, 2013).

$$PID = k_p e(t) + K_i T \sum_{i=0}^t e(i) + K_d \frac{e(t) - e(t-1)}{T} \quad (1)$$

A equação acima permitiu que a sub-rotina (ANEXO I) do PID digital inserida no Arduino fosse desenvolvida, sendo os parâmetros do controlador PID, os ganhos proporcional, integral e derivativo, sujeitos a modificações por parte do usuário através da interface gráfica (NETO *et al.*, 2013).

O segundo ponto destacado foi o método para realizar a comunicação com a Internet para a aplicação Java, por meio da qual o utilizador acessaria o experimento educacional, sendo necessário inserir o protocolo TCP/IP e o método de acesso por soquete na estrutura do software. Logo, quando o Arduino recebesse ou enviasse os pacotes de dados utilizaria as funções que fazem parte da biblioteca Ethernet *Shield* de acordo com o descrito em anexo (NETO *et al.*, 2013).

2.7 Experimentação *On-line* Utilizando a Plataforma VISIR

O trabalho desenvolvido por Tawfik *et al.* (2014), é um sistema de experimentação com base na plataforma de laboratório remoto VISIR (*Virtual Instrument Systems in Reality*), no qual a plataforma de instrumentação é baseada em extensões PCI (*Peripheral Component Interconnect*) para instrumentos (PXI) da NI aplicados ao desenvolvimento e controle de conversores de corrente contínua (TAWFIK *et al.*, 2014).

O hardware desse sistema é composto por uma plataforma NI PXI, que consiste de uma placa controladora (ou seja, um PC embarcado), placas de instrumentos (a fonte de alimentação de corrente contínua, multímetro digital, osciloscópio e gerador de funções) e um módulo no qual todas as placas estão conectadas. Os terminais dos módulos NI PXI estão, por sua vez, interligados a matriz de componentes. E a matriz de componentes se comunica com o condutor através de um cabo USB (TAWFIK *et al.*, 2014).

A matriz de comutação de relé é uma pilha de placas PCI/104. Os circuitos são gerados pela manipulação (abertura e fechamento) dos vários relés de acordo com o desenho do circuito recebido pelo controlador, e a interligação entre os terminais dos módulos NI PXI e os componentes utilizados são gerados pelos 10 nós comuns (A-I, 0) das placas PCI/104 da matriz (TAWFIK *et al.*, 2014).

A matriz conta com três placas de instrumentos e com até 16 placas de componentes. Cada placa de componente dispõe de 10 tomadas para componentes e cada tomada está ligada a um interruptor de polo duplo e disparo simples (DPST), e ainda quatro destas tomadas podem substituir os DPST por dois interruptores unipolares de disparo simples (SPST) (TAWFIK *et al.*, 2014).

Portanto, a matriz de componentes pode conter um máximo de 16 por 10 relés DPST. Sendo que, os terminais de dois componentes ocupam um soquete, enquanto os componentes com mais terminais ocupam mais de um soquete. De acordo com a folha de dados do fabricante, a carga máxima dos relés é de 2 A e a expectativa de vida mínima é de 3×10^8 operações (aproximadamente duas operações por segundo continuamente por cinco anos). A matriz tem um controlador de interface periférico (PIC18F4550), além de um controlador para cada uma das placas (PIC16F767), que é responsável pelo envio de comandos para os relés se abrirem ou fecharem (TAWFIK *et al.*, 2014).

O software do projeto VISIR é de código aberto e distribuído sob Licença Pública Geral (GPL) da GNU. O código pode ser obtido em <http://svn.openlabs.bth.se/trac>. Ele Possui um Sistema de Gestão de Aprendizagem (SGA), sendo o portal responsável por todos os

processos relacionados à administração, acesso, autenticação e reserva do sistema, desenvolvido em PHP e armazenado em um servidor web Apache HTTP com uma base de dados MySQL, permitindo aos usuários acessarem o Experimento Cliente, uma vez que estes sejam autenticados pelo Servidor Web. O Experimento Cliente (EC) é a interface gráfica do usuário (GUI), a bancada simulada de VISIR é um aplicativo produzido em *Action Script* para Adobe Flash, e também foi hospedado no Servidor Web (WS) conforme exibido na Figura 20 (TAWFIK *et al.*, 2014).

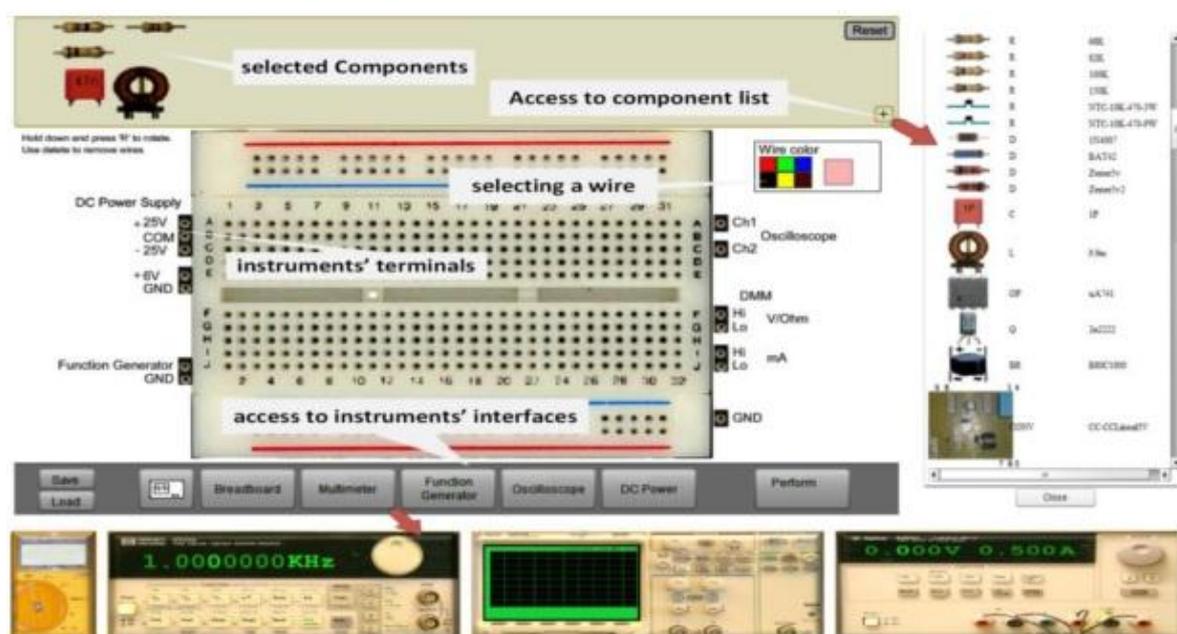


FIGURA 20 - Bancada Simulada da VISIR
Fonte: TAWFIK *et al.*, 2014, p. 192.

Por meio do mouse foi permitido ao usuário selecionar e arrastar os componentes para a *proto-board* simulada, conectar seus circuitos e configurar os instrumentos, em seguida pressionando-se o botão “*Perform*” para receber os resultados dos equipamentos físicos. O circuito modelado pelo usuário é transferido para o Servidor de Medida (MS) como um protocolo XML, denominado Protocolo de Experimentação (EP), que utiliza soquete XML ou o protocolo TCP sobre TCP/IP para transportar os dados necessários (TAWFIK *et al.*, 2014). O Servidor de Medida (MS) é uma aplicação de software desenvolvido no Microsoft Visual C++, sendo executado em um computador pessoal separadamente (ou seja, como um servidor). Nesta abordagem é responsável por verificar periodicamente a autenticação de usuários durante as sessões, gerenciar a fila de solicitações simultâneas, e validar os circuitos projetados pelos usuários e os valores máximos permitidos para os instrumentos de acordo com o definido pelo professor nos “*maxlist*” (listas com as especificações dos componentes e

instrumentos disponíveis e valores suportados pelos experimentos) para evitar circuitos perigosos. Após validar e gerenciar a fila de solicitações, o servidor a envia sequencialmente ao Servidor de Equipamentos (ES) (TAWFIK *et al.*, 2014).

O Servidor Equipamentos (ES) é uma aplicação de software para controle da instrumentação desenvolvida pelo LabVIEW e incorporado no controlador PXI NI. Esse software recebe pedidos de acesso aos equipamentos dos circuitos projetados pelo usuário que são verificados, pelo "Servidor de Medida", sob a forma de "Protocolo de Experimentação" e executados. O ciclo de operação se encerra com os resultados, sendo exibidos na tela do computador dos usuários. Todos os drivers dos instrumentos são *Instrumentos Virtuales Intercambiabiles* (IVI). Os IVI's são capazes de suportar outras plataformas com extensões LAN para Instrumentação (LXI) ou por barramento de dados de uso geral IEEE-488.2 (GPIB). A "lista de componentes" é inserida no "Servidor de Equipamentos" para identificar todos os dispositivos montados e instrumentos conectados.

Os principais componentes e o ciclo de operação da abordagem desenvolvida no projeto VISIR podem ser identificados na Figura 21.

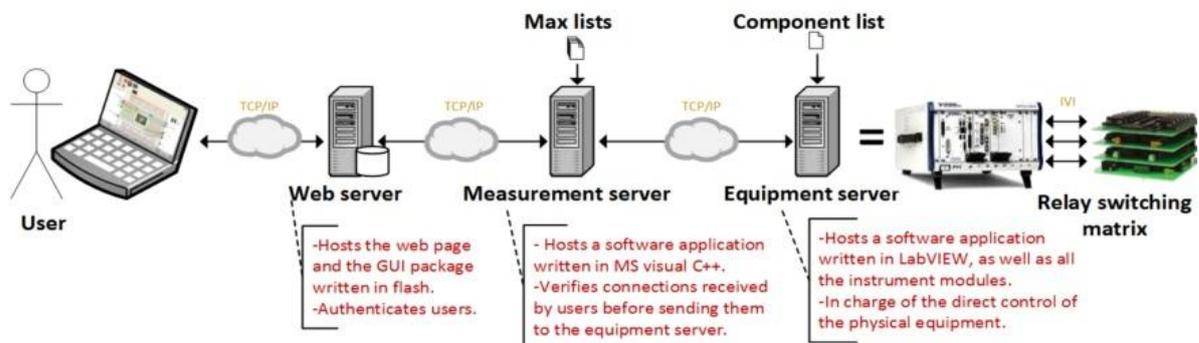


FIGURA 21 - Componentes da VISIR e o ciclo de operação

Fonte: TAWFIK *et al.*, 2014, p. 193.

2.8 Projeto REDES

Em 2015 conclui-se esta conjuntura histórica de trabalhos científicos, com o projeto REDES (*REmote Laboratory for Embedded Systems Design*) desenvolvido por, Anzhelika *et al.*(2015), no qual o laboratório remoto especializado REDES foi focado em fornecer aos desenvolvedores uma escolha rápida e eficaz nas decisões de design padrão com base em plataformas prontas de hardware e software (ANZHELIKA *et al.*, 2015).

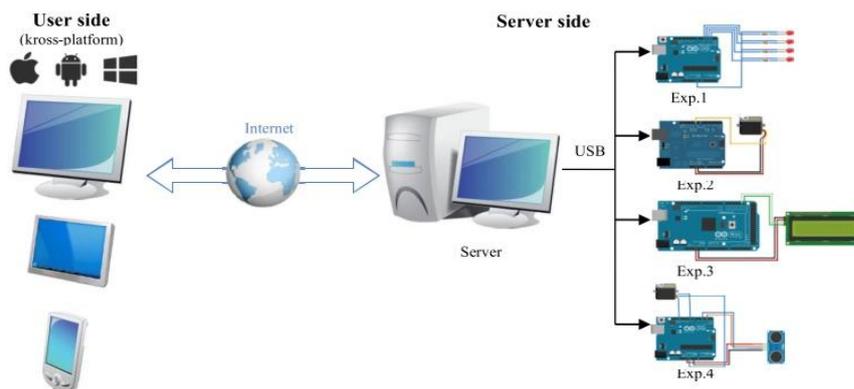


FIGURA 22 - Visão geral da infraestrutura REDES
 Fonte: ANZHELIKA *et al.*, 2015, p. 29.

Segundo Anzhelika *et al.* (2015), na Figura 22 está a ilustração da visão geral da infraestrutura REDES, no qual, o laboratório fornece acesso controlado à distância ao equipamento experimental baseado em Arduino. Pois, em comparação com outras plataformas semelhantes, Arduino possui vantagens:

- O projeto foi desenvolvido em software de código aberto, funcionando como um projeto rede/comunitário, permitindo aos participantes a troca de experiências e práticas aplicadas, e acelerando ainda mais o processo de desenvolvimento e depuração (ANZHELIKA *et al.*, 2015);
- O microcontrolador e suas expansões possuem um baixo custo (ANZHELIKA *et al.*, 2015);
- O ambiente de programação (Windows, OSX e Linux, 32/64 bit) é simples e multiplataforma (ANZHELIKA *et al.*, 2015).

Nesta abordagem Anzhelika *et al.* (2015) determina que o suporte com as experiências deve ser conectado ao servidor do laboratório através da interface serial. Além de ter um computador com o sistema operacional Linux Debian atuando como servidor para o laboratório remoto. O servidor garante acesso à programação da experiência e ao canal de visualização por vídeo do laboratório. O canal de vídeo foi desenvolvido através do uso de “ffmpeg”.

De acordo com os dados obtidos o servidor processa pedidos de clientes web e realiza as ações de compilar o código inicial recebido (Arduino, com o uso do utilitário de console Ino), enviar os pedidos e resultados de compilação (usando o protocolo HTTP), carregar dados para o microcontrolador (desde que controlador estivesse livre) e gerenciar a fila de clientes (caso a experiência estiver ocupada) (ANZHELIKA *et al.*, 2015).

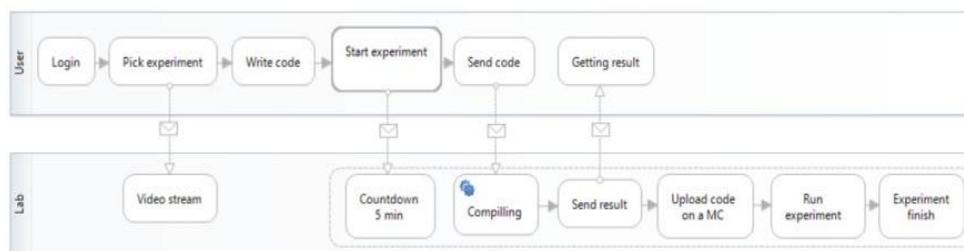


FIGURA 23 - Diagrama de processo do laboratório

Fonte: ANZHELIKA *et al.*, 2015, p. 29.

O diagrama de processo RELEDs é exibido na Figura 23. Por meio dele percebe-se que para que se tivesse acesso aos experimentos, os usuários precisariam ser registrados. Sendo o acesso à página de experimentos fechado para usuários não registrados. Os usuários poderiam carregar seu código de programação para um microcontrolador na placa Arduino via Web (ANZHELIKA *et al.*, 2015).

Com a interface Web, o usuário era capaz de criar código ou carregar arquivos com um código inicial. O servidor do laboratório compilava o código inicial recebido. Após a compilação, o código gerado era enviado para o microcontrolador na placa Arduino. Se o programa apresentasse erros, o servidor geraria e enviaria os resultados da compilação. Portanto, os usuários poderiam criar e editar o próprio código de programa diretamente pela interface do cliente Web, sem qualquer necessidade de possuir as ferramentas de desenvolvimento no computador pessoal do usuário (ANZHELIKA *et al.*, 2015).

A comunicação cliente/servidor é realizada através de uma interface web construída com HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) e JS (*Java Script*). O serviço é executado em um servidor Apache utilizando-se o banco de dados MySQL. Na Figura 24 tem-se ilustrada uma impressão da página de interface Web do experimento (ANZHELIKA *et al.*, 2015).

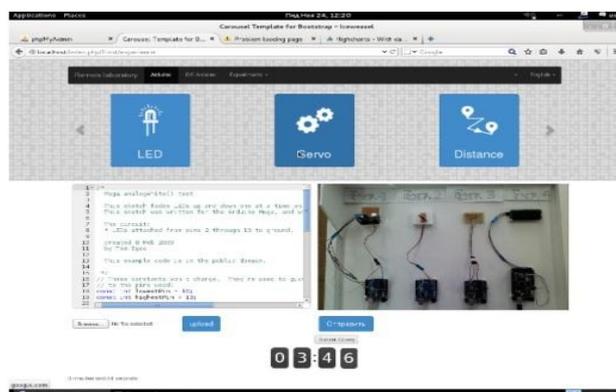


FIGURA 24 - Interface web do RELEDs

Fonte: ANZHELIKA *et al.*, 2015, p. 30.

É possível observar na ilustração que, por meio do “*slider*” localizado no topo da página, o usuário poderia mudar rapidamente de experimento. A área restante foi dividida para exibir vídeo (bloco à direita) e editar código (bloco à esquerda). Ao iniciar o trabalho com o experimento remoto, o usuário poderia observar o tempo que restava para o fim da experiência (relógio digital ao fim da página). E por fim, nenhuma ferramenta extra ou software adicional, além do navegador web, era necessário para acessar o laboratório remoto (ANZHELIKA *et al.*, 2015).

2.9 Comparação entre os Modelos Estudados

Conforme o que foi apresentado neste capítulo, é possível observar que os sistemas dos Laboratórios Remotos, além de possibilitar o acesso remoto a equipamentos físicos a eles conectados por meio de suas variadas interfaces gráficas, possuem uma característica em comum: os sistemas necessitam estar conectados a uma rede com um servidor dedicado ao sistema. Portanto, é possível classificar estes sistemas de acordo com o tipo de estrutura de servidor adotada. No diagrama a seguir ilustra-se essa classificação para os sistemas que foram apresentados neste estudo e na sequência uma tabela comparativa desses sistemas será exibida.

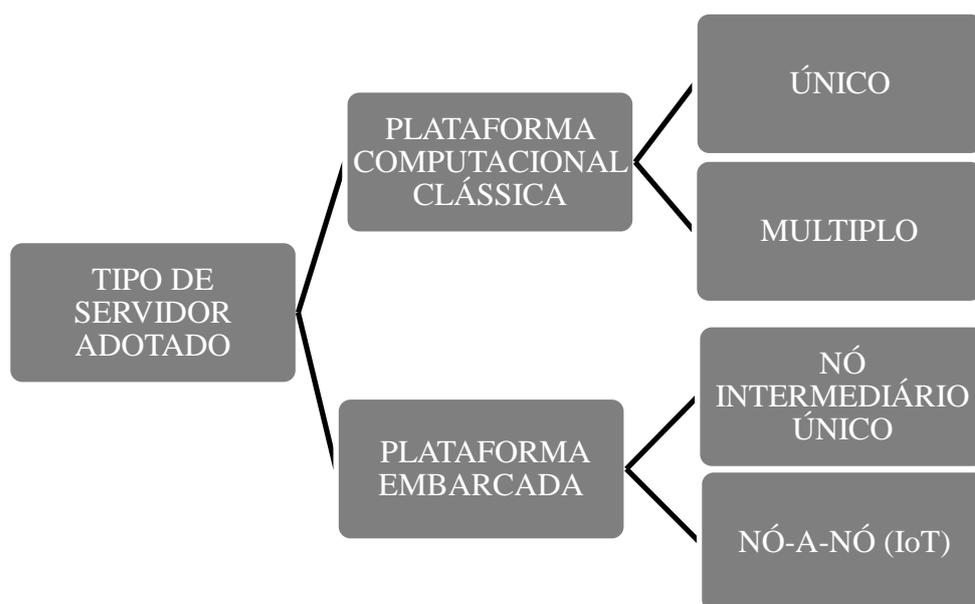


FIGURA 25 - Classificação dos laboratórios remotos de acordo com o tipo de servidor

Com isso é possível estabelecer uma tabela comparativa entre os diversos modelos estudados até o momento.

Tabela 2 - Comparação dos modelos estudados

MODELO	TIPO DE SERVIDOR	FLEXIBILIDADE
MSW	EMBARCADO DE NÓ ÚNICO	MUITO BAIXA
DIESEL	COMPUTACIONAL MULTIPLO	MUITO ALTA
ACESSO REMOTO À EXPERIÊNCIAS	COMPUTACIONAL MULTIPLO	ALTA
LABORATÓRIO DE EXPERIMENTAÇÃO REMOTA	EMBARCADO DE NÓ ÚNICO	MUITO BAIXA
LABORATÓRIO ACADÊMICO REMOTO MULTIUSUÁRIO	COMPUTACIONAL ÚNICO	MÉDIA
EXPERIMENTO EDUCACIONAL REMOTO	EMBARCADO NÓ-A- NÓ	BAIXA
PLATAFORMA VISIR	COMPUTACIONAL MULTIPLO	MÉDIA
RELDES	COMPUTACIONAL ÚNICO	ALTA

Na Tabela 2 atenta-se na classificação dos sistemas quanto a flexibilidade de acordo com a quantidade de aplicações distintas que eles podem executar, por isso que os modelos MSW, DIESEL e VISIR estão classificados em diferentes categorias, mesmo eles sendo plataformas voltadas para aplicações de eletrônica.

Dessa forma é possível perceber que delegar as funções corretas às estruturas adequadas (exemplo deixar o computador a cargo do gerenciamento dos serviços de comunicação, enquanto que as aplicações são realizadas por plataformas microcontroladas) se mostra como

o melhor caminho para a construção dos Laboratórios Remotos. Na seção a seguir serão apresentados os conceitos fundamentais para a construção destes.

3 FUNDAMENTAÇÃO TEÓRICA

Como mencionado pelas seções anteriores, esta seção aborda os fundamentos científicos presentes nos trabalhos apresentados e que possibilitaram a elaboração do conceito fundamental defendido pela proposta do presente estudo: os Laboratórios Remotos. Dessa forma, destacam-se os conceitos de:

- Redes de Computadores e a Internet;
- Comunicação serial via barramento USB;
- Sistemas embarcados, definindo-os, sua composição, incluindo-se a plataforma arduino;
- Computação ubíqua, em particular conceito de IoT;
- Ensino à distancia, foco em prática à distância;
- Definição formal de Laboratórios Remotos.

3.1 Redes de Computadores e a Internet

Aqui será apresentada uma visão geral sobre as redes de computadores, sua relação com a Internet, seus principais componentes e funcionamento.

Ao longo dos anos inúmeros especialistas tem definido uma rede de computadores como sendo a interconexão entre ao menos dois computadores, que possibilita o compartilhamento de recursos e a troca de informações entre essas máquinas. Como fez Andrew S. Tanenbaum, em sua obra *Redes de Computadores* de 2003, ao determinar que:

[...] utilizaremos a expressão "rede de computadores" quando quisermos mencionar um conjunto de computadores autônomos interconectados por uma única tecnologia. Dois computadores estão interconectados quando podem trocar informações. [...] (TANENBAUM, 2003, p. 18).

De uma forma simplificada é possível dividir as redes em LANs, MANs, WANs e inter-redes, cada qual com suas próprias especificidades, tecnologias, velocidades e mercados alvo. Enquanto as LANs operam em edifícios ou residências, as MANs englobam estados ou municípios e as WANs um país ou continente. E ao conjunto de diferentes redes interconectadas atribui-se o nome de inter-redes (TANENBAUM, 2003). Na tabela 3 apresenta-se de forma mais detalhada como as redes de computadores podem ser classificadas.

Tabela 3 - Classificação das redes de computadores

Tipo	Classificação
De Processamento	<i>Batch; On-line; Real-time.</i>
De Topologia	Malha; Barramento (linear); Estrela; Anel; Híbrida.
De Transmissão	Por difusão; Ponto-a-ponto.
De Escala	Residencial/predial; Municipal/estadual; Nacional/continental.

Dentre os diversos exemplos de redes conhecidas serão destaque neste trabalho a Internet e a Ethernet. Segundo Tanenbaum (2003), a Ethernet é o mais antigo e popular sistema de rede local (LAN) já desenvolvido, e sua arquitetura original pode ser visualizada na ilustração da Figura 26.

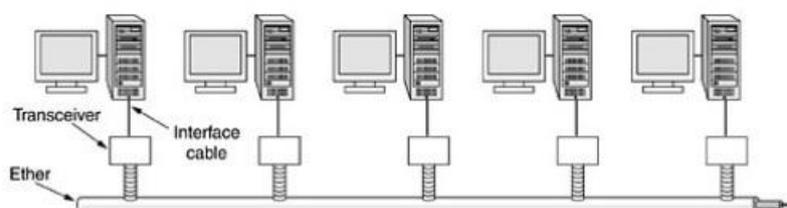


FIGURA 26 - Arquitetura da Ethernet original
Fonte: TANENBAUM, 2003, p. 66.

Na Figura 26, por meio de uma derivação realizada com um dispositivo de hardware conhecido como transceptor várias máquinas puderam ser conectadas em paralelo a um cabo coaxial chamado de cabo multiponto, também denominado *ether* (COMER, 1998). Operando a uma taxa de transmissão de 2,94 Mbps, a Ethernet apresentou um importante aperfeiçoamento referente ao momento de pré-transmissão, no qual, inicialmente fez-se com que um computador verificasse o cabo, a fim de saber se este já estava transmitindo algo e em caso afirmativo o computador era impedido até o término da transmissão corrente. Isso foi desenvolvido no intuito de evitar interferências nas transmissões, proporcionando uma eficiência muito maior do sistema (TANENBAUM, 2003).

Enquanto “A Internet não é de modo algum uma rede, mas sim um vasto conjunto de redes diferentes...” (TANENBAUM, 2003), ou ainda, “[...] uma rede de computadores mundial, isto é, uma rede que interconecta milhões de equipamentos de computação em todo mundo...” (KUROSE; ROSS, 2006). Na Figura 27 detalha-se a arquitetura (hardware) da Internet.

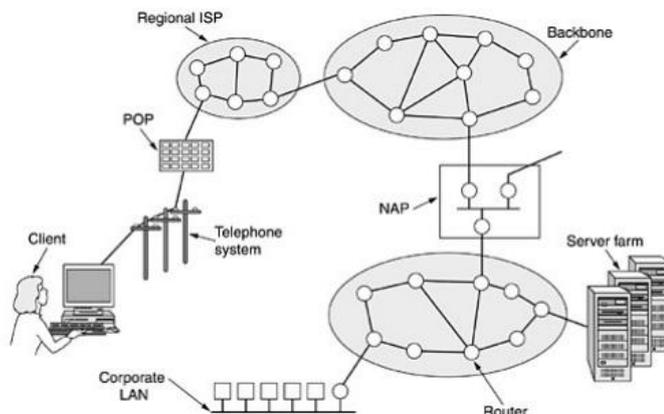


FIGURA 27 - Visão geral da Internet
Fonte: TANENBAUM, 2003, p. 60.

Para entender melhor a definição de Internet, ambos os autores, anteriormente citados, dedicam-se em detalhar os seus componentes e como é o seu funcionamento (em *software*). Kurose e Ross (2006) descrevem ainda, que a Internet pode ser uma estrutura de rede que fornece serviços para aplicações distribuídas. Na sequência deste tópico será feita uma descrição mais detalhada dos componentes de *software* de rede com enfoque naqueles que integram a Internet, para facilitar o seu entendimento.

O crescente desenvolvimento das redes gerou a necessidade de padronizá-las, tornando os *softwares* de rede altamente estruturados (TANENBAUM, 2003). Para isso, um passo importante foi dividi-las em protocolos e camadas.

A Internet e as redes de computadores utilizam os protocolos como forma de trocar informações entre duas ou mais entidades que se comunicam, como definido a seguir:

“Um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas nas transmissões e/ou no recebimento de uma mensagem ou outro evento.” (KUROSE; ROSS, 2006)

Segundo Tanenbaum (2003) cada camada funciona como uma máquina virtual que oferece serviços determinados à camada do nível acima, de acordo com os protocolos utilizados por ela. Dividir as redes em camadas possibilitou que os dados tivessem seu preparo facilitado,

simplificando as atribuições de cada camada para nada além do que preparar as informações para que o próximo nível as receba (SANTOS; SÁ, 2012).

Arquitetura de rede é como se denomina um conjunto de camadas e protocolos. Cada arquitetura deve armazenar informações suficientes em sua especificação para possibilitar o desenvolvimento do programa ou construção do hardware de cada camada, de forma que se obedeça corretamente ao protocolo adequado. A lista de protocolos utilizados por um sistema é definida como pilha de protocolos (TANENBAUM, 2003).

O modelo de referência TCP/IP começou a ser projetado em 1977 com a intenção de conectar uniformemente o crescente número de redes que aderiam a ARPANET, permitindo um crescimento ordenado da rede ao eliminar os problemas e restrições dos protocolos ocorridos em redes mais recentes (TANENBAUM, 2003).

A seguir será detalhada cada uma das camadas do modelo de arquitetura de rede TCP/IP, por ser utilizado pela maior parte dos sistemas de rede e por boa parte de sua pilha de protocolos ser utilizada pela Internet (KUROSE; ROSS, 2006). As camadas desse modelo estão organizadas como na Tabela 4:

Tabela 4 – A pilha de protocolo da Internet com base no modelo de referencia TCP/IP

Pilha de camadas
Camada de Aplicação
Camada de Transporte
Camada de Rede
Camada de Enlace
Camada Física

Fonte: Adaptado de KUROSE; ROSS, 2006, p. 36.

3.1.1 Camada de aplicação

De acordo com as definições estabelecidas por Tanenbaum (2003), Kurose e Ross (2006), é possível sintetizar que os serviços prestados na camada de aplicação são direcionados ao usuário, por estar localizada no topo da pilha tanto do cliente como do servidor e logo abaixo encontra-se a camada de transporte (SANTOS; SÁ, 2012).

Servidor é como se chama o dispositivo que oferece uma resposta a um serviço solicitado e denomina-se cliente o dispositivo que solicita um serviço na rede. Cada aplicação interage com um dos protocolos da camada de transporte para enviar ou receber dados e cabe a cada um definir o estilo de transporte a ser executado, podendo ser uma sequencia de mensagens

individuais ou um fluxo contínuo de bytes (COMER, 1998). Neste trabalho destacam-se as aplicações Web e o correio eletrônico. Seus protocolos são HTTP (*HyperText Transfer Protocol*) e SMTP (*Simple Mail Transfer Protocol*), respectivamente, e para garantir a entrega dos dados ambas utilizam o protocolo TCP (*Transmission Control Protocol*) na camada de transporte (SANTOS; SÁ, 2012).

O protocolo HTTP é definido nos RFC (*Request for Comments*) 1945 e 2616. Ele está presente na camada de aplicação da Web e é considerado por Kurose e Ross (2006) como “o coração da web”. Esse protocolo possibilita a troca de mensagens entre dois sistemas finais (sistemas computacionais), está alocado em dois programas cada qual funcionando em um sistema final diferente, um chamado de cliente e outro servidor. É responsável também, por definir a estrutura das mensagens e o modo como o cliente e o servidor as trocam (KUROSE; ROSS, 2006).

O protocolo responsável pelas aplicações de correio eletrônico é o SMTP, pois ele foca na forma como o sistema de transmissão de mensagens as envia por meio do enlace de um equipamento a outro, sem especificar como é realizado o armazenamento da mensagem e com qual frequência o sistema tenta transmiti-la. O formato das mensagens e cabeçalhos por ele carregado consiste em um texto ASCII de sete bits (COMER, 1998).

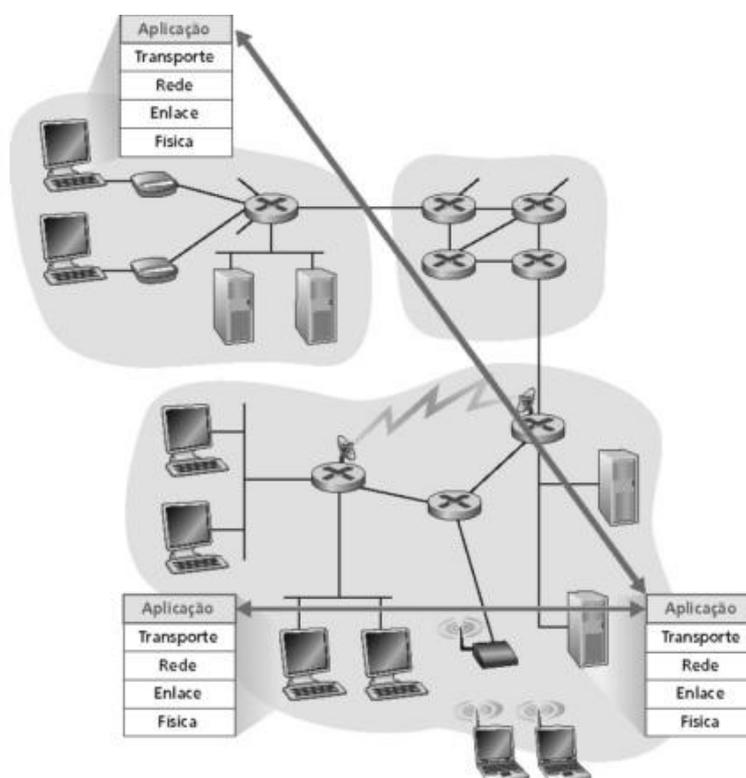


FIGURA 28 - Camada de aplicação
Fonte: KUROSE; ROSS, 2006, p. 59.

3.1.2 Camada de transporte

A camada de transporte é responsável pela comunicação “fim-a-fim” entre dois programas (aplicativos) regulando o fluxo de informações e fornecendo transporte confiável, sem erros e em sequência. Para assegurar o transporte (sem erros e em sequência) o protocolo da camada faz com que o receptor envie uma confirmação e o transmissor reenvie os pacotes (COMER, 1998). Logo, essa camada deve receber as mensagens da camada de rede e entregá-las à camada de aplicação, além de também realizar a operação contrária (sentido da aplicação para a de rede). Por isso está presente aos finais de processo (KUROSE; ROSS, 2006).

No modelo de referência TCP/IP a camada de transporte possui dois protocolos o UDP (*User Datagram Protocol*) e o TCP (SANTOS; SÁ, 2012). O protocolo UDP não é orientado à conexão, ou seja, a entrada de dados não é confiável. Sendo empregado em aplicações em que a velocidade da informação é mais importante do que a precisão das informações (KUROSE; ROSS, 2006).

Já o protocolo TCP, é o oposto do UDP, pois o TCP garante uma entrega confiável das informações por ser orientado a conexão, e ser utilizado onde é necessário prioritariamente a confiabilidade dos dados, mesmo sendo muito mais complexo (KUROSE; ROSS, 2006).

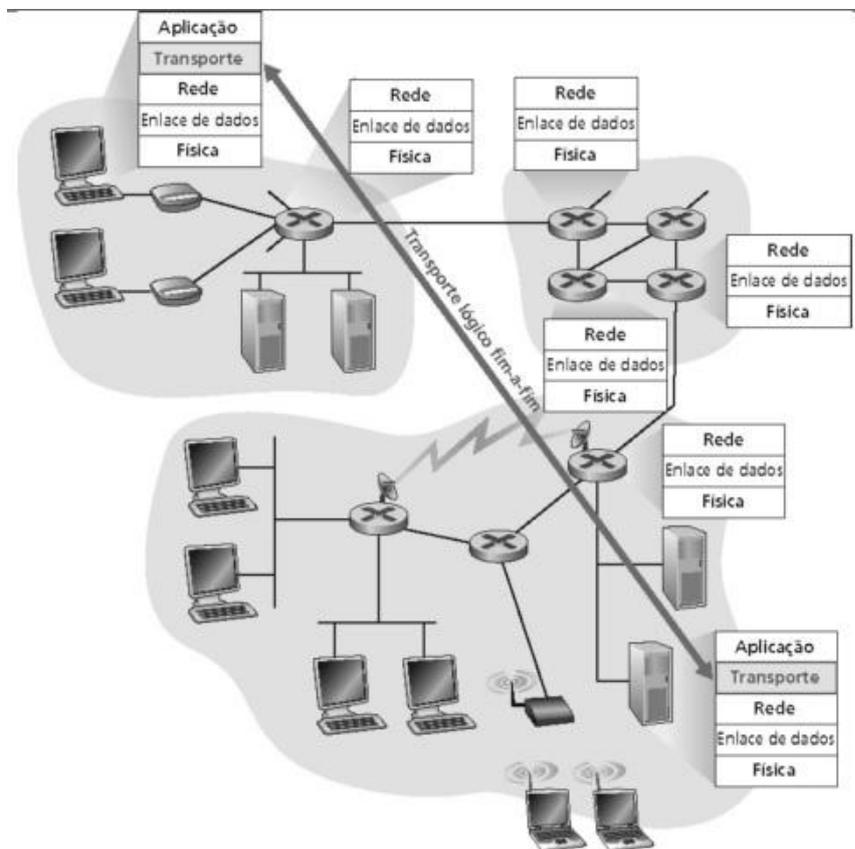


FIGURA 29 - Camada de transporte
Fonte: KUROSE; ROSS, 2006, p. 144.

3.1.3 Camada de rede

A camada de rede tem por função garantir a transferência de pacotes de um ponto da rede a qualquer outro ponto de destino utilizando vários roteadores para isso. Cabe a essa camada ter conhecimento dos roteadores que estão na rede e definir a melhor rota (serviços de repasse e roteamento), evitando-se o congestionamento das linhas (TANENBAUM, 2003).

Portanto a camada de rede é composta de dois componentes principais, um deles é o protocolo IP (*Internet Protocol*), que define os campos no datagrama e como os sistemas finais e roteadores atuam nestes campos; o outro é o protocolo de roteamento que estabelece a rota que a informação segue entre origens e destinos (KUROSE; ROSS, 2006).

Dessa forma a camada de rede deve oferecer serviços à camada de transporte tendo como princípio fundamental garantir a compatibilidade entre os diferentes tipos de hardware e de software que compõe um *host* destinatário e um *host* remetente durante todo o percurso do pacote. Esses serviços foram projetados com os objetivos de manter-se independentes da tecnologia de subrede, resguardando a camada de transporte da quantidade, do tipo e da topologia das subredes presentes e fazer com que o endereçamento disponível para a camada de transporte utilize um plano de numeração uniforme, mesmo nas LANs e WANs (TANENBAUM, 2003).

3.1.4 Camada de enlace

Segundo descreve Tanenbaum (2003), a camada enlace de dados ou link de dados é responsável pelo enquadramento, controle de erros, controle de fluxo e gerenciamento de enlace. Para alcançar esses objetivos, ela recebe os pacotes da camada de rede e os encapsula em quadros para efetuar a transmissão, e o gerenciamento destes quadros é considerado o cerne dessa camada.

A camada de enlace é composta de protocolos capazes de estabelecer uma comunicação segura e eficiente entre duas redes ligadas fisicamente e por isso é considerada a camada mais próxima das ligações físicas (SANTOS; SÁ, 2012).

O enquadramento é a restrição dos pacotes a serem enviados em quadros e sua estrutura deve ser especificada pelo protocolo da camada de enlace. São várias formas diferentes de controle de erros que podem exercer esse serviço, então é suficiente que se saiba, quando se envia um pacote de dados, um cabeçalho é anexado a ele contendo informações sobre os dados que estão sendo enviados e como a checagem deverá ser efetuada pela camada de enlace de

destino. Dessa forma, caso o conteúdo do pacote não esteja de acordo com o que informa o cabeçalho da camada enlace de envio, uma mensagem de erro é enviada e o pacote ou parte dele é retransmitido. Por fim, para que um receptor lento não seja sobrecarregado, o controle de fluxo assegura um sistema de rápida transmissão de dados (SANTOS; SÁ, 2012).

3.1.5 Camada física

Essa é a camada responsável pelo acesso ao meio físico. A tarefa desta camada é movimentar bits individuais contidos em um quadro de um nó para o próximo. Seus protocolos dependem do enlace e de seu meio de transmissão, pois em cada casa a transmissão ocorre de uma forma diferente (KUROSE; ROSS, 2006).

Dessa forma, sua principal funcionalidade é a entrega de dados entre duas máquinas conectadas entre si, podendo isso ser realizado conforme a necessidade e independente do tipo de hardware da conexão física de diferentes formas (SANTOS; SÁ, 2012).

Os endereços IP são ditos lógicos, pois entende-se que só existam em nível de software. Mas o mapeamento dos endereços, no entanto, é feito ao ser adicionado o endereço físico do *host* no endereço IP (SANTOS; SÁ, 2012).

Ao endereço físico através do qual qualquer destino é encontrado na rede local é atribuído o nome de *Media Access Control* (MAC). O MAC é um número em código hexadecimal gravado internamente nos adaptadores de rede sendo único, ou seja, não existem dois iguais no mundo (SANTOS; SÁ, 2012).

A interface existente entre o endereço lógico (IP) da camada de redes e o endereço físico (MAC) da camada de enlace é chamada de protocolo ARP (*Address Resolution Protocol*). O módulo ARP da máquina transmissora solicita da rede de destino o endereço MAC referente ao IP de destino para o qual os dados serão enviados e dessa forma um datagrama IP pode ser enviado (SANTOS; SÁ, 2012).

3.2 Comunicação via Barramento USB

Esta seção apresenta a porta USB (*Universal Serial Bus*), seus principais conceitos e funcionamento, pois através dela ocorre a comunicação entre o servidor e o sistema físico.

Para que os barramentos paralelo ou serial funcionassem, os dispositivos deveriam estar conectados à unidade central de processamento (CPU) e em seguida era necessária a

instalação do *driver* do dispositivo por meio de um programa de instalação específico, o que poderia acarretar em um conflito no sistema.

Com a finalidade de tornar mais prática a conexão de dispositivos é que surgiu o padrão USB, Tecnologia na qual, o dispositivo, após a conexão, é rapidamente localizado pelo sistema operacional e instalado, facilitando sua adição, remoção e movimentação (LOUREIRO; BARRETO e BIBAS FILHO, 2009).

O desenvolvimento da tecnologia USB tinha como principal objetivo tornar a instalação de dispositivo algo simples e intuitiva, até para usuários sem experiência. Desenvolvida com base no conceito *plug and play*, que implica em “conectar” o dispositivo periférico ao computador, e após a conexão o sistema o reconhece sem que seja necessário reiniciá-lo (*hot swap*), ou ao sistema operacional e sem qualquer tipo de intervenção por parte do usuário (LOUREIRO; BARRETO e BIBAS FILHO, 2009).

Com esta ótica, em 1995, a parceria realizada entre as empresas Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC e Philips criaram a conexão USB. Realizada através de um barramento que possui um conector comum a todos os dispositivos externos que o utilizarem, facilitando a instalação de dispositivos periféricos, de forma a possibilitar sua imediata utilização logo após a sua conexão. Permitindo-se ao sistema operacional e a placa mãe diferenciar a classe dos equipamentos, quais as necessidade de alimentação, largura de banda entre outros, ao carregar os drives necessários para sua imediata utilização (LOUREIRO; BARRETO e BIBAS FILHO, 2009).

As primeiras versões desenvolvidas foram as 1.0 com velocidade de 1,5 Mbps, em seguida foram as 1.1 com velocidade que vai de 1,5 Mbps a 12 Mbps foram concebidas. Ao fim do ano 2000, foram lançadas as versões 2.0 com velocidade de 480 Mbps, equivalente a 60 MBPs compatível com as versões anteriores, embora com algumas alterações na tecnologia e topologia (LOUREIRO; BARRETO e BIBAS FILHO, 2009).



FIGURA 30 - Tridente
Fonte: LOUREIRO; BARRETO e BIBAS FILHO, 2009, p. 21.

A Figura 30 é a representação simbólica da conexão USB. Quatro fios e uma malha de eliminação de ruídos compõe o cabo USB. Sendo que, dos quatro fios dois são para o

transporte de energia e alimentação do dispositivo e os dois restantes utilizados para a transferência de dados entre o dispositivo e o sistema. A codificação desses cabos está apresentada na Tabela 5 de acordo com o padrão de cores (LOUREIRO; BARRETO e BIBAS FILHO, 2009).

Tabela 5 - Tabela de padrão de cores

CORES	TIPO	TENSÃO	POLARIDADE
VERMELHO	VBUS	5 V	POSITIVO
PRETO	GND	0 V	NEGATIVO
VERDE	D+	BIT 1	
BRANCO	D-	BIT 0	

Fonte: Adaptado de LOUREIRO; BARRETO e BIBAS FILHO, 2009, p. 22.

O máximo fornecido pelo “*bus*” USB é uma tensão de 5 V e uma corrente de 500 mA, isso em cada porta do *Root Hub* do *host*. Via *software* é possível configurar a quantidade de corrente um dispositivo irá necessitar para funcionar (LOUREIRO; BARRETO e BIBAS FILHO, 2009).

Os fios restantes, D+ (dado positivo) e D- (dado negativo), são utilizados pelo sistema USB para realizar transferência de dados entre o *host*, *hub* e dispositivos. Por meio da codificação NRZI (*No Return Zero Inverted*), todos os sinais de dados trafegam por esses dois fios. Isso implica que a codificação do bit 1 ocorre por meio da transição entre a maior e a menor voltagem, podendo ocorrer da maior para a menor ou o inverso da menor para a maior. Já com o bit 0 ocorre o oposto, pois ele é codificado sem transição. No intervalo de um bit a tensão se mantém constante (LOUREIRO; BARRETO e BIBAS FILHO, 2009).

3.3 Sistemas Embarcados

Quando uma atividade demandar ser executada de forma automática por algum dispositivo eletromecânico de qualquer natureza, tem-se um sistema embarcado (SANTANA, 2008).

Em muitos segmentos da indústria faz-se uso de tecnologias embarcadas para as mais diversas atividades. Presentes na maioria dos dispositivos modernos como os aparelhos celulares, os sistemas embutidos utilizam tecnologias embarcadas para realizarem a comunicação com outros equipamentos eletrônicos, seja para troca de informações seja controlando suas funções (NOERGAARD, 2005 *apud*. SANTANA, 2008).

Os principais elementos presentes em um sistema embarcado também são elementos essenciais em qualquer sistema de computação, sendo eles uma CPU, memória e dispositivos de entrada/saída (SANTANA, 2008). Na ilustração da Figura 31 tem-se a representação de um esquema contendo esses elementos.

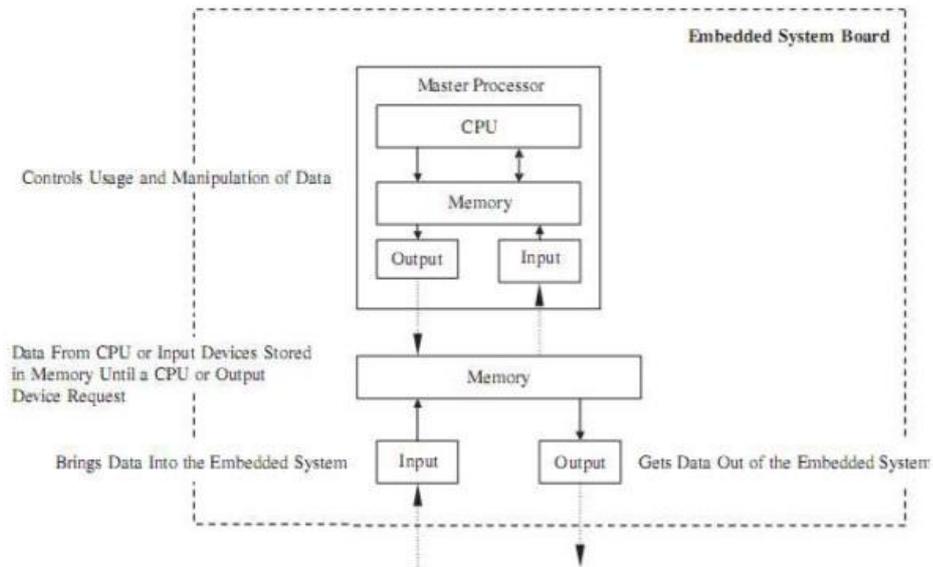


FIGURA 31 - Elementos básicos de um sistema embutido
Fonte: SANTANA, 2008, p. 7

3.3.1 Unidade central de processamento

O conjunto de instruções chamado programa ou caminho de dados é executado na CPU e tem seus resultados armazenados na memória do sistema (registradores). A ULA (Unidade Lógica Aritimética) (ALU) é a parte da CPU responsável pela execução de operações matemáticas, uma vez que esse tipo de operação compõe a maior parte do conjunto de instruções do sistema, e para resolvê-las a ULA é imbuída de circuitos lógicos capazes de realizar tais operações (BATES, 2006 *apud*. SANTANA, 2008).

Um sistema embarcado possui duas opções viáveis capazes de atuar como CPU, os microprocessadores e os microcontroladores. Porém, pelos microprocessadores não conterem os outros componentes básicos integrados, apenas os microcontroladores serão abordados a seguir (SANTANA, 2008).

Um microcontrolador é um circuito integrado (CI) que contém todos os principais componentes de um computador típico (BAYLE, 2013). Originou-se pela diversificação dos

processos de controle antes efetuados por microprocessadores (WILMSHURST, 2007 *apud*. SANTANA, 2008).

3.3.2 Memória

As memórias são o elemento computacional onde estão alojadas as rotinas e as variáveis que regulam o fluxo de informações dentro do sistema (SANTANA, 2008). Muitas vezes são um espaço comum de execução tanto para a rotina interna do núcleo quanto para os elementos do usuário (geralmente chamadas ROM e RAM) (BAYLE, 2013).

Tradicionalmente as memórias podem ser classificadas em duas categorias: voláteis e não voláteis. As memórias voláteis operam apenas no momento em que o sistema se encontra energizado, além de perderem seu conteúdo quando o sistema deixa de ser energeticamente alimentado. A memória conhecida por *Random Access Memory* (RAM) é constituída usualmente de material semicondutor de fácil escrita da perspectiva elétrica, também pode ser chamada de memória de dados (SANTANA, 2008).

Enquanto que as memórias não voláteis armazenam seu conteúdo quando o sistema é desativado, do aspecto elétrico dificilmente podem ser modificadas, intituladas memórias ROM (*Read Only Memory*), são conhecidas também pela denominação de memória de programa (SANTANA, 2008).

Os sistemas embutidos usualmente utilizam memórias do tipo *flash*. Elas funcionam em alta velocidade, possuem baixo gasto energético e permitem que sejam realizadas modificações através de sinais elétricos (WILMSHURST, 2007 *apud*. SANTANA, 2008).

3.3.3 Dispositivos de entrada/saída

Não é possível controlar ou automatizar nenhum tipo de processo com sistemas embarcados sem a presença desses componentes, por serem responsáveis pela troca de informações entre o sistema e o meio físico (SANTANA, 2008).

Dispositivos de entrada, é como são chamados os dispositivos que permitem ao usuário interagir com o sistema embutido. Eles permitem o controle sobre variáveis ou sobre algum dispositivo de saída específico. Entre os exemplos mais convencionais têm-se teclados, botões e sensores. Já os dispositivos de saída estão associados à interação dos usuários com o funcionamento do sistema embarcado, exibindo os resultados gerados pelo programa de acordo com os estímulos recebidos dos dispositivos de entrada. Dentre os mais comuns estão

os *displays* LCDs, os *displays* de sete segmentos, LEDs, monitores e impressoras (NOERGAARD, 2005 *apud.* SANTANA, 2008).

Existem ainda, dispositivos com a capacidade de realizar as duas tarefas simultaneamente, por exemplo, controladores de rede e módulos de rádio. Pois gerenciam a comunicação entre dois ou mais elementos, dessa forma deve-se integrá-los bidirecionalmente, transformando-os em dispositivos tanto de entrada, como de saída (SANTANA, 2008).

A comunicação para os dispositivos de E/S pode ser realizada de duas formas: serial e paralela. Na comunicação paralela, a transmissão dos dados ocorrem a 8 bits por vez, enquanto a serial transmite 1 bit por vez em fila. No geral a forma paralela é mais rápida do que a forma serial, mas requer mais pinos para ser executada e isso implica em uso maior dos recursos do microcontrolador. Por isso, tornaram-se populares os protocolos de comunicação serial (BATES, 2006 *apud.* SANTANA, 2008).

3.3.4 Arduino

Arduino é uma placa microcontrolada com base nos microcontroladores ATmega da Atmel, possui um conector USB para estabelecer uma comunicação com o computador e um número de pinos de ligação que podem ser utilizados para comunicar-se com dispositivos eletrônicos periféricos, como motores, relés, sensores de luz, diodos de laser, alto-falantes, microfones, e outros. A placa pode ser alimentada através da conexão USB com computador, ou a partir de uma bateria ou fonte de alimentação externa. Pode ser controlada a partir do computador ou programada através de um computador sendo desconectada em seguida para que trabalhe de forma autônoma. O projeto da placa é de código aberto e isto significa que qualquer pessoa está autorizada a desenvolver placas compatíveis com a Arduino (MONK, 2011).

A plataforma Arduino foi concebida, em primeiro lugar, para popularizar o uso da eletrônica em projetos de design de interação de universitários. O projeto foi iniciado na Itália em 2005 pelos fundadores Massimo Banzi e David Cuartielles. Hoje é um dos mais belos exemplos do conceito de código aberto, trazidos ao mundo do hardware e sendo frequentemente usados no mundo do software. O site oficial do projeto é <http://www.arduino.cc> (BAYLE, 2013).

Arduino é uma plataforma microcontrolada potente com um grande volume de memória que permite a criação de dispositivos eletrônicos complexos (ANZHELIKA *et al.*, 2015).

Na Tabela 6 está a descrição das características técnicas da placa Arduino uno contendo seus componentes.

Tabela 6 – Especificações Técnicas do Arduino uno

Microcontrolador	ATmega328P
Tensão de operação	5 V
Tensão de entrada (recomendada)	7 – 12 V
Tensão de entrada (limite)	6 – 20 V
Número de pinos E/S digitais (dos quais 6 oferecem saída PWM)	14
Número de pinos PWM de E/S digitais	6
Número de pinos de entrada analógica	6
Valor de corrente DC por pino de E/S	20 mA
Valor de corrente para pinos com tensão de 3,3 V	50 mA
Memória flash	32 KB (ATmega328P) dos quais 0,5 KB usados pelo bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Frequência	16 MHz
Comprimento da placa	68,6 mm
Largura da Placa	53,4 mm
Peso da Placa	25 g

Fonte: Adaptado de www.arduino.cc.

Cada um dos 14 pinos digitais do Arduino Uno pode ser utilizado como uma entrada ou saída, através das funções “pinMode()”, “digitalWrite()” e “digitalRead()”. Eles operam a 5 volts. Cada pino pode fornecer ou receber 20 mA como condição de funcionamento recomendada e tem um resistor *pull-up* interno (desconectado por padrão) de 20 a 50 k Ω . Um máximo de 40 mA é o valor que não deve ser excedido em nenhum pino de E/S para que danos permanentes ao microcontrolador sejam evitados (texto adaptado de www.arduino.cc). Além disso, alguns pinos possuem funções especiais, por exemplo, os pinos Serial 0 (RX) e 1 (TX), utilizados para receber (RX) e transmitir dados seriais (TX) TTL. Estes pinos são ligados aos pinos correspondentes do chip do ATmega8U2 USB para TTL Serial. Os pinos de interrupções externas 2 e 3, podem ser configurados para disparar uma interrupção por um valor baixo, uma borda de subida ou de descida, ou uma mudança de valor. Verifique a função “attachInterrupt()” para mais detalhes. Os pinos PWM 3, 5, 6, 9, 10, e 11. Fornecem saída PWM de 8 bits, por meio da função “analogWrite()”(texto adaptado de www.arduino.cc).

Os pinos SPI 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK), suportam comunicação SPI usando a biblioteca SPI. No pino LED 13, existe um *built-in* LED impulsionado pelo pino digital 13, dessa forma quando o pino recebe um valor alto de tensão, o LED é ligado, e quando o pino recebe um valor baixo é desligado. Os pinos TWI A4 ou pino SDA e A5 ou pino SCL, permitem comunicação TWI através da biblioteca “Wire” (texto adaptado de www.arduino.cc).

O Uno possui 6 entradas analógicas, rotuladas de A0 a A5, cada uma das quais com 10 bits de resolução. Por padrão elas medem de 0 a 5 volts, embora seja possível mudar o limite superior da sua faixa de operação usando o pino AREF e a função “analogReference()”. O que implica em outro par de pinos com função especial na placa. O pino AREF de tensão de referência para as entradas analógicas, sendo utilizado em conjunto com “analogReference()”. O pino “RESET”, o qual ao receber um valor de tensão baixo para reiniciar o microcontrolador da placa, normalmente utilizado para adicionar um botão de reiniciar para *shields* que bloqueiam o da placa (texto adptado de www.arduino.cc).

O Arduino/Genuino Uno possui uma série de facilidades para se comunicar com um computador, ou com outra placa Arduino/Genuino, ou com outros microcontroladores. O ATmega328 oferece comunicação serial por meio da UART TTL (5V), que está disponível nos pinos digitais 0 (RX) e 1 (TX). Um ATmega16U2 nos canais da placa está comunicação serial por USB que aparece como uma porta virtual COM para o software no computador. O firmware 16U2 usa os *drivers* USB COM padrão, sem a necessidade de *driver* externo.

Porém, no Windows, um arquivo de extensão .inf é necessário (texto adaptado de www.arduino.cc).

O Software Arduino (IDE) inclui um monitor serial que permite que dados simples de texto sejam enviados para a e, da placa. Os LEDs RX e TX na placa piscam quando os dados estiverem sendo transmitidos através do chip USB para serial e da conexão USB para o computador (mas não para comunicação serial nos pinos 0 e 1) (texto adaptado de www.arduino.cc).

A biblioteca “*SoftwareSerial*” permite comunicação serial em qualquer um dos pinos digitais do Arduino Uno. O ATmega328 também suporta comunicação I2C (TWI) e SPI. O Software Arduino (IDE) inclui uma biblioteca chamada “*Wire*” para simplificar o uso do barramento I2C e para a comunicação SPI, utiliza-se a biblioteca SPI (texto adaptado de www.arduino.cc).

Em vez da necessidade de se pressionar fisicamente o botão de reiniciar antes de um *upload*, a placa Arduino/Genuino Uno foi concebida de uma forma que é possível reiniciá-la por um software rodando em um computador a ela conectado. Uma das linhas de controle de fluxo de hardware (DTR) do ATmega8U2/16U2 está ligada à linha de reiniciar dos ATmega328 através de um capacitor de 100 nF. Quando esta linha recebe um valor de tensão baixo, a linha “*RESET*” desativa por tempo suficiente para o chip seja reiniciado. O Software Arduino (IDE) usa esse recurso para permitir que o usuário envie código, simplesmente pressionando o botão de *upload* na barra de ferramentas da interface. Isto significa que o *bootloader* pode ter um tempo de espera mais curtos, como a redução dos DTR pode ser bem coordenado com o início do carregamento do código (texto adaptado de www.arduino.cc).

Esta configuração tem outras implicações. Quando o Arduino Uno é conectado a um computador com Mac OS X ou Linux, ele reinicia a cada vez que uma conexão é feita com o software (via USB). Para o segundo seguinte, o *bootloader* estará em execução no Uno. Enquanto ele é programado para ignorar os dados malformados, ele irá interceptar os primeiros bytes de dados enviados para a placa depois que uma conexão for aberta. Se um programa estiver rodando na placa recebe configuração de uma só vez ou outros dados quando ele começar, certifique-se de que o software com o qual ele se comunica espera um segundo depois de abrir a conexão e antes de enviar esses dados (texto adaptado de www.arduino.cc).

A placa Arduino Uno contém uma caminho que pode ser cortado para desativar o auto-reset. Os conectores em ambos os lados do caminho podem ser soldadas em conjunto para reativá-lo. Está identificada como “*RESET-EN*”, também pode é possível desativar o auto-

reset, ligando-se um resistor 110 Ω de pino 5V para a o pino “RESET” (texto adptado de www.arduino.cc).

A Revisão 3 da placa recebeu novos recursos, dentre os quais, pinagem 1.0, os pinos SDA e SCL foram adicionados aos que estão perto do pino AREF e outros dois novos pinos colocados perto do pino de “RESET”. O IOREF permite que os *Shield* se adaptem à voltagem fornecida a partir da placa. No futuro, “*Shields*” serão compatíveis tanto com a placa que usa o AVR, que opera com 5V, quanto com o Arduino Due que opera com 3.3V. O segundo é um pino não conectado, que está reservada para futuros propósitos. Além disso a terceira revisão também ganhou um circuito de “RESET” mais potente e o microcontrolador ATmega 16U2 substituiu o 8U2. Na Figura 32 exibe-se o diagrama de pinos do microcontrolador ATmega168/328 (texto adptado de www.arduino.cc).

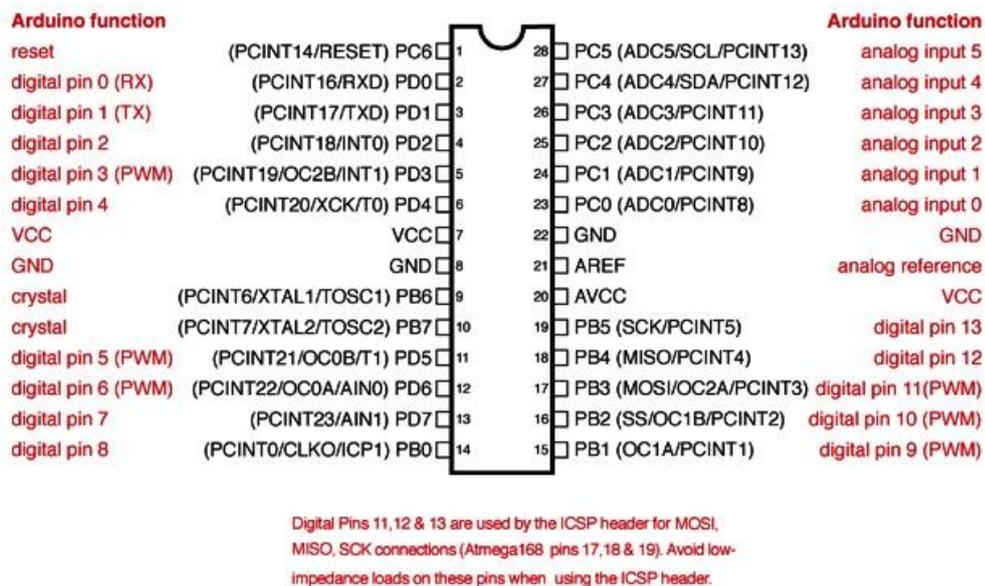


FIGURA 32 - Diagrama de pinos do ATmega168/328

Fonte: Adptado de www.arduino.cc

O hardware computacional da plataforma Arduino é um dos mais populares e simples. Arduinos são perfeitos para projetos eletrônicos de prototipagem, pois permitem uma rápida e barata construção de protótipos de sistemas embarcados. As Vantagens do Arduino reside nas bibliotecas e código comunitários, além da grande quantidade de placas de extensão auxiliares (chamados de *Shields*), que oferecem aos usuários ferramentas de hardware e software para atingir seus objetivos (ANZHELIKA et al., 2015).

3.4 Computação Ubíqua e IoT

A definição para computação ubíqua, como sendo o desenvolvimento de um ambiente dotado de poder computacional compartilhado com as interações dos usuários humanos (WEISER, 1991 *apud*. BOLZANI, 2010), vem sendo fomentada com o passar dos anos.

Com isso inúmeros trabalhos têm sido desenvolvidos para definir a Internet das Coisas (IoT) e para associar tecnologias e arquiteturas de rede para este conceito abstrato. A quantidade exponencialmente crescente de dispositivos interconectados, presentes no ambiente diário das pessoas, exige soluções de interação eficientes para permitir que qualquer pessoa acesse facilmente qualquer objeto (DUQUENNOY; GRIMAUD e VANDEWALLE, 2009).

O conceito de rede de área pessoal (PAN) tem sido frequentemente utilizado para referir-se às redes de Internet das Coisas (IoT). A PAN é uma rede volátil que inclui todos os objetos com os quais uma pessoa deveria interagir. As principais preocupações das PAN's para a Internet das Coisas residem na questão de a PAN ser uma rede local espacialmente volátil, que inclui os dispositivos que um usuário pode acessar fisicamente (DUQUENNOY; GRIMAUD e VANDEWALLE, 2009).

A descoberta e o endereçamento dos objetos devem ser governados pelos protocolos da PAN, além de, permitir aos dispositivos serem capazes de interagir com toda a PAN e, possivelmente, com redes externas e, ainda, fazer com que as operações e interações realizadas em uma PAN sejam asseguradas (DUQUENNOY; GRIMAUD e VANDEWALLE, 2009).

Deve ser notado que os nós da PAN são principalmente dispositivos de hardware limitado. Geralmente, não possui mais do que poucos KB de memória volátil e não volátil, uma frequência de CPU de alguns MHz, rígidas restrições de autonomia e conexões físicas com baixas taxas de transferência (por exemplo, Bluetooth, USB, ZigBee) (DUQUENNOY; GRIMAUD e VANDEWALLE, 2009).

O uso de aplicações simples fez o sucesso da Internet, principalmente as aplicações Web e e-mails. Conforme, o passar dos anos e os avanços tecnológicos, percebeu-se que as tecnologias da Web estão se tornando cada vez mais generalizadas, possibilitando que qualquer pessoa acesse os servidores Web a partir de um computador pessoal, ou telefone celular. Além disso, sabe-se que é tecnicamente possível incorporar servidores Web em dispositivos minúsculos. Aos quais Duquennoy, Grimaud e Vandewalle (2009) denominaram por *Web of Things* (WoT) o conceito de desenvolvimento de PANs por meio de servidores Web instalados em todos os nós.

O WoT consiste na utilização de protocolos e tecnologias da Web em uma arquitetura PAN focada no cliente. O computador, ou telefone celular do usuário executa um cliente (um navegador padrão da Web), enquanto cada nó no PAN executa um servidor Web. O cliente é capaz de acessar qualquer servidor incluído no seu PAN. Um objeto pode, eventualmente, ser acessado por vários usuários, em tal situação, estará incluído em várias PAN's (DUQUENNOY; GRIMAUD e VANDEWALLE, 2009).

Uma das vantagens óbvias da WoT é que ele usa as tecnologias existentes e generalizadas: TCP/IP, HTTP e aplicativos Web. Isso garante uma grande acessibilidade bem como facilidade de desenvolvimento, pelas numerosas e bem conhecidas ferramentas de design existentes (DUQUENNOY; GRIMAUD e VANDEWALLE, 2009).

A Internet das Coisas refere-se exclusivamente aos objetos identificáveis e suas representações virtuais em uma estrutura de Internet. O termo “Internet das Coisas” foi usado pela primeira vez em 1999, se tornando popular através da Auto-ID Center e publicações de análise do mercado. As etiquetas RFID, sensores, atuadores e telefones celulares são muitas vezes vistos como pré-requisitos para a Internet das Coisas. As principais tecnologias da Internet das Coisas incluem a tecnologia RFID, rede de sensores e tecnologia de detecção (FANG SHIFENG *et al.*, 2014).

3.5 Ensino a Distância

O termo Educação à Distância (EaD) tem sido utilizado desde o século XIX e tem sido definida como o método de ensino caracterizado pela separação entre alunos e professores no espaço e/ou tempo (PERRATON, H., 1986 *apud.* GUIDA; NUNES e MOSCOSO, 2007), além de proporcionar aos alunos um aprendizado muito mais intenso sob essas condições (JONASSEN *et al.*, 1992 *apud.* GUIDA; NUNES e MOSCOSO, 2007). A EaD foi construída ao longo do tempo e com a evolução das tecnologias de rede e da popularização da Internet. Encontra-se em sua terceira geração, caracterizada pelo meio dos Ambientes Interativos (GUIDA; NUNES e MOSCOSO, 2007).

Durante a última década houve um rápido desenvolvimento das redes de computadores e da Internet. A WWW (*World Wide Web*), um dos serviços mais populares da Internet, oferece meios de publicação e acesso de grandes volumes de informação em formatos textuais e multimídia. Essas características da WWW deu um grande potencial para o desenvolvimento de ambientes de educação a distância. (NETO *et al.*, 2013)

3.6 Laboratórios Remotos

Concluindo-se a apresentação dos fundamentos conceituais presentes até o momento, nesta seção será realizada a elaboração de uma definição formal para o conceito de Laboratórios Remotos e uma generalização de sua arquitetura.

Os Laboratórios Remotos são classificados por Marchezan *et al.* (2004, *apud.* GUIDA; NUNES e MOSCOSO, 2007) como a disponibilização dos recursos de um ou mais instrumentos de medição reais por meio de uma rede distribuída, como a Internet, à usuários remotamente localizados.

Dito isso, pode-se ainda definir um Laboratório Remoto como um conjunto de dispositivos de propósito geral, que devem estar conectados de forma que seja possível acessá-los fisicamente por meio de uma comunicação em rede, que garanta a configuração e o controle desses dispositivos e à análise dos dados gerados por eles (FELIPE; KELVIN e TEIXEIRA, 2010).

Na Figura 33 detalha-se o modelo genérico adotado neste trabalho para os Laboratórios Remotos. A proposta de metodologia para a concepção deste modelo será apresentada na seção a seguir.



FIGURA 33 - Laboratório remoto genérico
Fonte: Adaptado de FELIPE; KELVIN e TEIXEIRA, 2010, p. 3.

4 DESCRIÇÃO DO SISTEMA PROPOSTO

Nesta seção será apresentada uma metodologia que poderá corroborar o objeto tema do presente estudo, bem como unir as tecnologias descritas até o momento pelo mesmo, de forma a ser apenas uma sugestão (na qual apenas testes preliminares foram realizados) que possibilite a efetivação da aplicação desejada.

A elaboração do sistema, o qual denominou-se Laboratório Remoto, foi pensada em nível de firmware, hardware e software, pois desse forma tem-se uma visualização mais completa e didática de todo o sistema. Dividiu-se estruturalmente o laboratório remoto em três componentes funcionais essenciais sendo eles: a aplicação física, a central de controle e a interface gráfica de usuário.

4.1 Firmware

A arquitetura projetada demonstrou um sistema que pode integrar servidores web a aparelhos com microcontroladores embutidos para controlá-los e gerenciá-los através da Internet por meio de navegadores web, via páginas web (CAN FILIBELI; OZKASAP e REHA CIVANLAR, 2007). Atendendo-se ao modelo genérico proposto na seção anterior, o firmware dessa arquitetura foi pensado conforme a Figura 34.

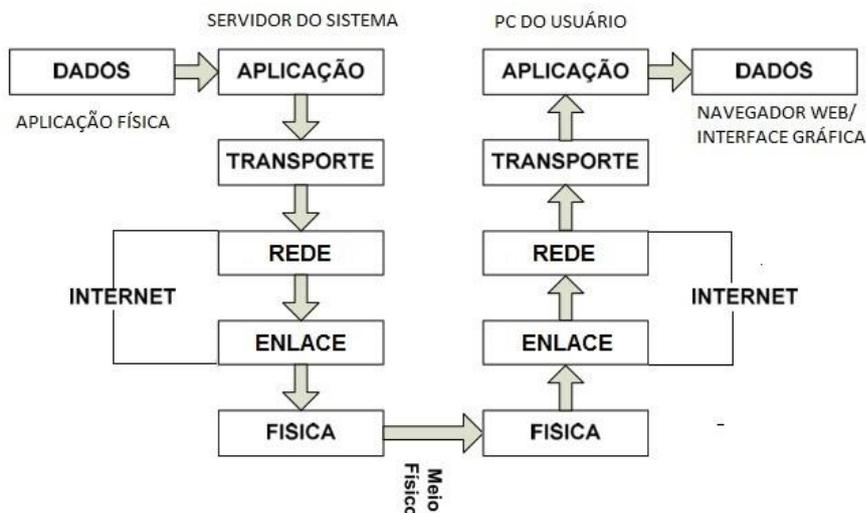


FIGURA 34 - Fluxo de dados da arquitetura cliente/servidor do sistema

As principais contribuições desta arquitetura são seu design leve, a configuração automática e utilização de protocolos de rede amplamente disponíveis de TCP/IP e HTTP. Pois utiliza a

pilha de protocolos apresentada na seção anterior. Na Figura 34 demonstra-se como as requisições realizadas pelo usuário por meio da interface gráfica (página web) são gerenciadas pela central de controle (servidor web) enviando ou não essas requisições para a aplicação física.

4.2 Hardware

As Especificações de hardware seguem de conforme ilustrado na Figura 35, demonstrando-se a visão geral do sistema.



FIGURA 35 - Componentes de hardware do sistema

Dessa forma o hardware do sistema deve ser composto de ao menos um computador com um processador Pentium T2390 da Intel, 2 GB de memória RAM, um disco rígido com 160 GB SATA com servidor web Apache2 instalado no sistema operacional Linux distribuição UBUNTU 12.04 operando como a central de controle, uma plataforma embarcada Arduino Uno Rev3 para a aplicação física com uma fonte de corrente contínua de 12 volts, um cabo USB para estabelecer a conexão serial entre o Servidor e a Arduino e uma conexão de rede com acesso a Internet para o servidor. Optou-se por este conjunto de hardware pois não foi necessário adquirir um novo PC uma vez que já havia um disponível gratuitamente, além de o Sistema Operacional Linux ser uma plataforma de código livre com grande suporte comunitário.

4.3 Software

O software do sistema deve ser desenvolvido de acordo com os componentes essenciais. Dessa forma, é necessário o desenvolvimento de ao menos, três rotinas de programação em diferentes linguagens, cada uma atendendo um dos componentes essenciais.

4.3.1 Aplicação física

Na aplicação física deve ser desenvolvido um programa para governar a placa Arduino Uno e sua extensão (*Shield*) auxiliar de acordo com a necessidade da aplicação. Esse programa pode ser mais facilmente desenvolvido na IDE da Arduino em linguagem C ou C++ conforme o exemplo apresentado no ANEXO II. É por meio dessa rotina que a aplicação física é controlada.

4.3.2 Central de controle

Antes das outras rotinas serem desenvolvidas, primeiro deve-se instalar o servidor no computador que será utilizado como central de controle. No Linux isso pode ser feito via terminal ou via disco de instalação. Isso deve ser realizado antecipadamente, pois é a central de controle a responsável por designar um IP para a interface gráfica de usuário, além de também ser responsável pela modificação em qualquer rotina de programação do sistema como um todo. Após isso, devem ser armazenadas as outras duas rotinas de programação no diretório exclusivo do servidor. Uma delas é voltada para o gerenciamento de dados e a comunicação entre a aplicação física (Arduino) e a interface gráfica do usuário (página web do laboratório). A outra se trata da rotina de desenvolvimento da própria interface gráfica. Existe uma gama de linguagens que podem ser utilizadas para o desenvolvimento dessa rotina, mas nenhuma será mais fácil, amigável e com grande suporte quanto a PHP. Um exemplo desse código se encontra no ANEXO II.

4.3.3 Interface gráfica de usuário

Para o desenvolvimento da rotina da interface gráfica, a linguagem mais facilmente empregada é o HTML. É nessa rotina que deve ser desenvolvida toda estrutura de interatividade do sistema com o usuário, podendo ser efetuado o acionamento, leitura e/ou controle de portas, digitais, PWM, analógicas e exibição de imagens de vídeo (por meio de

uma câmera IP) e gráficos gerados em tempo real pelo sistema em operação. Um exemplo de código pode ser visualizado no ANEXO II.

É bem sabido que em laboratórios é possível aplicar e testar o conhecimento teórico em situações práticas. O ensino de sistemas de controle e automação coloca os alunos em contato com situações reais que lhes permitam fazer comparações da teoria clássica e moderna com a prática (NETO *et al.*, 2013).

5 CONSIDERAÇÕES FINAIS

O conceito de Internet das Coisas (IoT) tem sido proposto por mais de 10 anos, e ele não reside apenas no nível de conceito, mas está se tornando uma realidade com o rápido desenvolvimento e ampla aplicação da rede de sensores sem fio e da computação em nuvem. Nesse contexto, os Laboratórios Remotos surgem como uma solução interessante para a limitada disponibilidade de recursos de laboratório das Universidades, pela possibilidade de serem acessados a qualquer momento e estarem conectados à rede mundial de computadores com disponibilidade de 24 horas por dia, sete dias por semana.

Diversas arquiteturas foram estudadas e avaliadas por este trabalho, e cada qual com seus pontos positivos e negativos, por isso não é definida uma solução definitiva para o desenvolvimento dos Laboratórios Remotos. Neste texto avaliaram-se as diferentes arquiteturas e optou-se por aquela considerada a mais adequada para atender às necessidades de maneira mais prática e flexível possível, bem como os critérios de gestão de recursos corretos, abrangendo a maior gama de aplicações possíveis e a um custo razoável. A adição principal é uma camada que permite administrar os recursos e integrar novas aplicações e experimentos e, por isso, essa arquitetura não é limitada por um número fixo de processos ou participantes. Há também uma noção de administradores locais e globais que controlam os recursos e o uso de práticas, evitando problemas com o aparecimento de muitos usuários. Essa abordagem permitirá que o laboratório remoto seja desenvolvido para operar a partir da Internet.

Em trabalhos futuros, aponta-se: para (i) a realização de testes de funcionamento do sistema proposto aplicado a uma bancada de experimentos dos laboratórios, realizando-se a constatação da efetividade funcional do sistema; (ii) Elaborar um sistema de interação virtual entre os usuários e o administrador do sistema; (iii) Propor um sistema de mudança de experimento conforme o aumento das aplicações físicas disponíveis para serem integradas ao sistema; (iv) Substituir o computador pessoal da central de controle por uma placa chamada Raspberry PI, para aplicar os conceitos de IoT e tornar o sistema mais modular e flexível, permitindo novos tipos de aplicações.

REFERÊNCIAS

ANZHELIKA, P.; OLGA, G.; IVANOV, E.; SOKOLYANSKII, A.; KURSON, S. Development and Application of Remote Laboratory for Embedded Systems Design. **12th International Conference on Remote Engineering and Virtual Instrumentation (REV)**, v. 11, n. 3, p. 27–31, 2015. Disponível em: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7087265&url=http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7087265> . Acesso em: 14 dez. 2015.

BARRIOS, A.; PANCHE, S.; DUQUE, M.; GRISALES, V. H.; PRIETO, F.; VILLA, J. L.; CHEVREL, P.; CANU, M. A multi-user remote academic laboratory system. **Computers and Education**, v. 62, p. 111–122, 2013. Disponível em: <<http://dx.doi.org/10.1016/j.compedu.2012.10.011>>. Acesso em: 14 dez. 2015.

BASHMAN, B.; SIERRA, K. e BATES, B. **Use a Cabeça! Servlets & JSP**. 1.ed. Rio de Janeiro: Alta Books, 2005.

BATES, M. **Interfacing PIC Microcontrollers: Embedded Design by Interactive Simulation**. 1st. ed. Burlington: Elsevier, 2006.

BAYLE, J. C **Programming for Arduino**. 1st. ed. Birmingham: Packt Publishing, 2013. Disponível em: <www.packtpub.com>. Acesso em: 8 dez. 2015.

BEETNER, D.; POTTINGER, H. e MITCHELL, K. Laboratories teaching concepts in microcontrollers and hardware-software co-design. **30th Annual frontiers in education conference**, v. 2, 2000. p. S1/1-5.

BOLZANI, C. A. M. **Análise de Arquiteturas e Desenvolvimento de uma Plataforma para Residências Inteligentes**. [s.l.] Universidade de São Paulo, 2010. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3142/tde-12082010-112005/publico/Tese_Caio_Augustus_Morais_Bolzani.pdf>. Acesso em: 29 set. 2015.

CALLAGHAN, M. J.; HARKIN, J.; MCCOLGAN, E.; MCGINNITY, T. M.; MAGUIRE, L. P. Client-server architecture for collaborative remote experimentation. **Journal of Network**

and Computer Applications, v. 30, n. 4, p. 1295–1308, 2007. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1084804506000749>>. Acesso em: 14 dez. 2015.

CAN FILIBELI, M.; OZKASAP, O. e REHA CIVANLAR, M. Embedded web server-based home appliance networks. **Journal of Network and Computer Applications**, v. 30, n. 2, p. 499–514, 2007. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1084804506000257>>. Acesso em: 14 dez. 2015.

COMER, D. E. **Interligação em rede com TCP/IP Volume 1: Princípios , protocolos e Arquiteturas**. 3ª. ed. Rio de Janeiro: Editora Campus, 1998.

DUQUENNOY, S.; GRIMAUD, G. e VANDEWALLE, J. J. Smews: Smart and mobile embedded web server. **Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems, CISIS 2009**, p. 571–576, 2009. Disponível em: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5066843&url=http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5066843>. Acesso em: 14 dez. 2015.

FANG SHIFENG, L. D. X.; ZHU, Y.; AHATI, J.; PEI, H.; YAN, J.; LIU, Z. An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things. **IEEE Transactions on Industrial Informatics**, v. 10, n. 2, p. 1596–1605, 2014. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6725615>>. Acesso em: 14 dez. 2015.

FELIPE, Á.; KELVIN, A. e TEIXEIRA, C. **Laboratório de experimentação remota com microcontroladores pic**. V Congresso Norte-Nordeste de Pesquisa e Inovação. **Anais...Maceió: CONNEPI - IFCE**, 2010. Disponível em: <<http://connepi.ifal.edu.br/ocs/anais/>>. Acesso em: 3 set. 2015.

GUIDA, L. C.; NUNES, T. M. e MOSCOSO, M. N. A. **Sistema de Acesso Remoto Via Web a Experiências de Eletrônica** Belém Instituto de Estudos Superiores da Amazônia - IESAM, 2007. Disponível em: <<http://www3.iesam-pa.edu.br/ojs/index.php/computacao/article/viewFile/136/125>>. Acesso em: 1 Out. 2015.

JONASSEN, D. H. **Evaluating Constructivistic Learning. In Constructivism and the Technology of Instruction: A Conversation**, eds. T Duffy and D. Jonassen. Hillsdale, NJ: Lawrence Erlbaum. 1992.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores: Uma abordagem top-Down**. 3^a. ed. São Paulo: Pearson Education do Brasil, 2006. v. 5

LOUREIRO, A. F.; BARRETO, I. F. DA C. e BIBAS FILHO, L. C. T. **Acionamento de dispositivos elétricos/eletrônicos por meio da porta usb, microcontrolador PIC e transceptor de RF**. Belém Universidade da Amazônia - UNAMA, , 2009. Disponível em: <<http://www.trabalhosfeitos.com/ensaios/Acionamento-De-Dispositivos-Eletricos/42853696.html>>. Acesso em: 3 set. 2015.

MARCHEZAN, A. R.; CHELLA, M. T. e FERREIRA, E. C. **Laboratório remoto aplicado ao ensino de engenharia eletrônica**. 2004. Disponível em <http://www.nied.unicamp.br/~siros/doc/2211.pdf>.

MONK, S. **Arduino + Android Projects for the Evil Genius: Control Arduino with Your Smartphone or Tablet**. p. 224, 2011.

NETO, J. M.; PALADINI, S.; PEREIRA, C. E.; MARCELINO, R.; GRUBER, V.; SILVA, J. B. Remote educational experiment applied to electrical engineering. **International Journal of Online Engineering**, v. 9, n. 1, p. 47–51, 2013. Disponível em: <<http://www.periodicos.capes.gov.br/>>. Acesso em: 17 dez. 2015.

NOERGAARD, T. **Embedded systems architecture - a comprehensive guide for engineers and programmers**. 1st. ed. Burlington: Elsevier, 2005.

OGATA, KATSUHIKO. **Teoria de controle moderno**. 4^a Ed., Editora Pearson, 2010.

PERRATON, H. Distance Education: An Economic and Educational Assessment of its Potential for Africa. **EDT Discussion Paper No.43**. Washington, D.C.: World Bank. 1986.

SANTANA, L. V. **O uso da Internet e da comunicação sem fio via zigbee em sistemas**

embutidos. Monografia de Graduação em Engenharia de Controle e Automação, UFOP/EM – Universidade Federal de Ouro Preto, Agosto de 2008. Disponível em: <[http://em.ufop.br/cecau/monografias/2008/LUCAS VAGO.pdf](http://em.ufop.br/cecau/monografias/2008/LUCAS_VAGO.pdf)>. Acesso em: 19 set. 2015.

SANTOS, F. C. DOS; SÁ, P. S. S. Embarcando uma Página Web em um Microcontrolador para Comunicação Via Internet Através do Protocolo TCP / IP. **Revista das Faculdades Integradas Claretianas**, v. 5, p. 70–89, 2012. Disponível em: <<http://claretianorc.com.br/download?caminho=upload/cms/revista/sumarios/160.pdf&arquivo=sumario6.pdf>>. Acesso em: 3 set. 2015.

SILVA, J. B. **Monitoramento , aquisição e controle de sinais elétricos , via Web , utilizando microcontroladores** .Florianópolis, 2002. Disponível em: <<https://repositorio.ufsc.br/bitstream/handle/123456789/82454/190481.pdf?sequence=1>>. Acesso em: 17 fev. 2016.

SOARES, J. M. **Um Sistema de Gestão para a Educação Tecnológica à Distância – Projeto e Implementação**, Dissertação de Mestrado, UFC, 2001.

TANENBAUM, A. S. **Redes de computadores**. [s.l.: s.n.]. v. 18

TAWFIK, M. et al. Experimentación Online en la Plataforma VISIR con Convertidores CC / CC. **VAEP-RITA**, v. 2, n. 4, p. 191–198, 2014. Disponível em: <http://rita.det.uvigo.es/VAEPRITA/index.php?content=Num_Pub&idiom=Es&visualiza=3&volumen=2&numero=4&articulo=6>. Acesso em: 19 jan. 2016.

WEISER, M. **The Computer for the 21st Century**Scientific American, 1991.

WILMSHURST, T. **Designing Embedded Systems with PIC Microcontrollers**. 1st. ed. Burlington: Elsevier, 2007.

ANEXO

ANEXO I – CÓDIGO PARA IMPLEMENTAÇÃO DE UM CONTROLADOR PID

///código de implementação do PID desenvolvido por NETO *et al.*,2013.

```
void fazer_pid(){
    float dt = (float) 50/100;
    int erro = setpoint – entrada;
    erro_somatorio += erro;
    float p = (float) kp*(erro);
    float i = (float) ki*(erro_somatorio)*dt;
    float d = (float) kd*(erro – erro_anterior)/dt;
    erro_anterior = erro;
    saída = p + i + d;
}

void setup( ) {
    server_initi();
}

void loop ( ) {
    if (client_receive( )){
        while (client_available( )) {
            client_write(client_read( ));
        }
        client_send();
    }
}
```

ANEXO

ANEXO II – ROTINAS EXEMPLOS PARA IMPLEMENTAÇÃO

//Código da página web

```

<html>
<head>
<meta http-equiv="Content-Language" content="pt-br">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Controle de Leds com Arduino</title>
</head>
<body>
<div align="center">
  <table border="0" width="15%" cellpadding="0">
    <tr></tr><tr><td>

<!-------POST DO BOTAO LIGAR AMARELO----->
<form method="POST" action="arduino.php">
<p>
  <input type="hidden" value="Amarelo" name="estado" >
  <input type="submit" value="Ligar Led Amarelo" name="amarelo">
</p>
</form>
  </td></tr><tr>

<!-------POST DO BOTAO LIGAR AZUL----->

<form method="POST" action="arduino.php">

  <input type="hidden" value="Azul" name="estado" >
  <td><input type="submit" value="    Ligar Led Azul    " name="azul">
</td>
</form>
  </tr><tr><td>

<!-------POST DO BOTAO LIGAR LARANJA----->

<form method="POST" action="arduino.php">
<p>
  <input type="hidden" value="Laranja" name="estado" >
  <input type="submit" value="    Ligar Led Laranja" name="laranja">
</p>
</form>
  </td></tr><tr>

<!-------POST DO BOTAO LIGAR VERDE----->

<form method="POST" action="arduino.php">

  <input type="hidden" value="Verde" name="estado" >
  <td><input type="submit" value="    Ligar Led Verde    " name="verde">
</td>
</form>
  </tr>
  </table>
</div>
</body>
</html>

```

Salve com o nome de “index.html”

//Código para comunicação em PHP

```

/*****
Autor: Fernando Krein Pinheiro
Data: 18/08/2011
Linguagem: PHP
===== IMPORTANTE =====
O código esta livre para usar,
citar e compartilhar desde que
mantida sua fonte e seu autor.
Obrigado.
*****/

<?php

$port = fopen("/dev/ttyUSB0", "w");
if ($_POST['estado']=="Amarelo")
{
    echo "Ligou o Amarelo";
    fwrite($port, "a");
}
if ($_POST['estado']=="Azul")
{
    echo "Ligou o Azul";
    fwrite($port, "z");
}
if ($_POST['estado']=="Laranja")
{
    echo "Ligou o Laranja";
    fwrite($port, "l");
}
if ($_POST['estado']=="Verde")
{
    echo "Ligou o Verde";
    fwrite($port, "v");
}
fclose($port);
?>

```

Salve com o nome de “Serial.php”

**=====
///Código do arduino**

```

/*****
Autor: Fernando Krein Pinheiro
Data: 18/08/2011
Linguagem: C (Wiring)
===== IMPORTANTE =====
O código esta livre para usar,
citar e compartilhar desde que
mantida sua fonte e seu autor.
Obrigado.
*****/

int ledPin13 = 13;
int ledPin11 = 11;
int ledPin9  = 9;
int ledPin7  = 7;

```

```
void setup()  {

    Serial.begin(9600);

    pinMode(ledPin13, OUTPUT);
    pinMode(ledPin11, OUTPUT);
    pinMode(ledPin9, OUTPUT);
    pinMode(ledPin7, OUTPUT);

}

void loop()
{
    char character;
    character = Serial.read();
    //se pressionado "v" liga o led verde
    if(character == 'v')
    {
        digitalWrite(ledPin13, HIGH);
    }
    else
        //Se pressionado "z" liga o led azul
        if(character == 'z')
        {
            digitalWrite(ledPin11, HIGH);
        }
    else
        //S pressionado "l" liga o led laranja
        if(character == 'l')
        {
            digitalWrite(ledPin9, HIGH);
        }
    else
        //Se pressionado "a" liga o led amarelo
        if(character == 'a')
        {
            digitalWrite(ledPin7, HIGH);
        }

    delay(1000);
}
```