

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

**LEONARDO DEIVID PIERRI E SABRINA PITZ LIMA**

**DESENVOLVIMENTO DE UM EXPERIMENTO REMOTO BASEADO EM  
SISTEMA DE GERAÇÃO ALTERNATIVA HÍBRIDO**

**Araranguá, 27 de Junho de 2016**

LEONARDO DEIVID PIERRI E SABRINA PITZ LIMA

DESENVOLVIMENTO DE UM EXPERIMENTO REMOTO BASEADO EM SISTEMA DE GERAÇÃO  
ALTERNATIVA HÍBRIDO

Trabalho de Curso submetido à  
Universidade Federal de Santa Catarina  
como parte dos requisitos necessários  
para a obtenção do Grau de Bacharel em  
Tecnologias da Informação e  
Comunicação. Sob a orientação do  
Professor Dr. Roderval Marcelino.

**Araranguá, 2016**

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da  
UFSC.

Pierri e Lima, Leonardo Deivid e Sabrina Pitz  
DESENVOLVIMENTO DE UM EXPERIMENTO REMOTO BASEADO EM  
SISTEMA DE GERAÇÃO ALTERNATIVA HÍBRIDO / Leonardo Deivid  
Pierri e Sabrina Pitz Lima ; orientador, Roderval  
Marcelino - Florianópolis, SC, 2016.  
126 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, . Graduação em  
Tecnologias da Informação e Comunicação.

Inclui referências

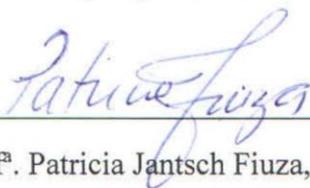
1. Tecnologias da Informação e Comunicação. 2.  
Experimentação Remota. 3. Sistema Híbrido. 4. Sistema  
Embarcado. 5. Educação. I. Marcelino, Roderval. II.  
Universidade Federal de Santa Catarina. Graduação em  
Tecnologias da Informação e Comunicação. III. Título.

Leonardo Deivid Pierri e Sabrina Pitz Lima

**DESENVOLVIMENTO DE UM EXPERIMENTO REMOTO BASEADO EM  
SISTEMA DE GERAÇÃO ALTERNATIVA HÍBRIDO**

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de Bacharel em Tecnologias da Informação e Comunicação, e aprovado em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

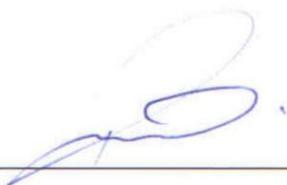
Araranguá, junho de 2016.



Prof. Patricia Jantsch Fiuza, Dra.

Coordenadora do Curso

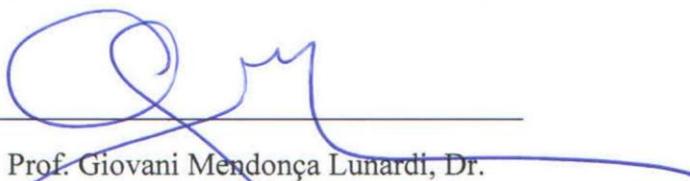
**Banca Examinadora:**



Prof. Roderval Marcelino, Dr.

Orientador

Universidade Federal de Santa Catarina



Prof. Giovani Mendonça Lunardi, Dr.

Examinador

Universidade Federal de Santa Catarina



Prof. Wilson Gruber, Dr.

Examinador

Universidade Federal de Santa Catarina

*Dedicamos aos nossos pais,  
Luci e David, Inalva e Samir, pois sem  
esse alicerce de amor, apoio e incentivo,  
jamais seria possível essa conquista.*

## **AGRADECIMENTOS**

Agradecemos sempre a Deus, porque ele é a base que nos sustenta e a luz que nos guia em todos os momentos, por abrir portas em nossos caminhos nos dando saúde e força para superar todas as dificuldades.

Às nossas famílias, por estar sempre presente e nos apoiando em todos os momentos dessa caminhada e em especial aos nossos pais, pela educação e orientação que nos deram, pela compreensão a nossa ausência e pelo apoio nos momentos difíceis.

Aos nossos colegas e amigos, pelo auxílio e companheirismo.

A todos os mestres que contribuíram de alguma forma para a construção do nosso conhecimento e formação acadêmica.

À Universidade Federal de Santa Catarina pela oportunidade proporcionada e ao nosso orientador prof. Roderval Marcelino pelas suas sábias orientações cujas quais foram fundamentais para que este trabalho fosse concluído.

*Talvez não tenhamos conseguido fazer o melhor, mas lutamos para que o melhor fosse feito. Não somos o que deveríamos ser, não somos o que iremos ser, mas graças a Deus, não somos o que éramos.*

*Martin Luther King*

## RESUMO

Este trabalho retrata o desenvolvimento de um experimento remoto baseado em um sistema de iluminação com geração de energia alternativa híbrido eólico-fotovoltaico. Um grande fator que contribui para a degradação do meio ambiente é o uso acentuado de energia não renovável, devido a este fato, aumentaram-se os estudos na área de energias renováveis na busca por soluções que reduzam os impactos causados pelo consumo de energia. A implantação de sistemas híbridos de energia nas instituições de ensino é deveras importante, tanto pelo fato da sustentabilidade quanto pela possibilidade de experiência prática dos alunos através do acesso ao mesmo, pois a experiência prática reforça o processo de aprendizagem. Porém, o custo de implantação de um sistema deste é alto e nem toda instituição pode adquiri-lo, outro fator de impasse é a distância geográfica entre o sistema e os interessados, pois o acesso ao sistema fica restrito aquela localidade. O experimento remoto desenvolvido possui um sistema de aquisição de dados que é formado por um hardware e um software. O hardware realiza a medição das informações do estado de operação do sistema híbrido, sendo elas, corrente do carregador solar, corrente do carregador eólico, corrente do carregador elétrico, tensão da bateria, carregador elétrico ligado ou desligado e luminária LED acesa ou apagada. O software é responsável por armazenar os dados da medição das informações do estado de operação do sistema híbrido realizada pelo hardware. Esses dados são disponibilizados em uma página web que fora desenvolvida, onde são disponibilizadas também as imagens do sistema híbrido que são capturadas por uma câmera IP. Desta forma qualquer usuário que possuir um dispositivo com navegador web e acesso a internet poderá usufruir das informações do sistema híbrido sempre atualizadas e visualizar as imagens capturadas do mesmo online, através da página web. Sendo assim, o experimento remoto apresentado neste trabalho é considerado uma boa alternativa, pois a experimentação remota possui recursos que permitem o acesso por meio virtual a um experimento real, que ficará disponível 24 horas por dia, 7 dias por semana, onde o usuário pode suprir a necessidade de experiência prática independente da sua localização geográfica, adquirindo e agregando conhecimento.

**Palavras-chave:** Experimentação Remota. Sistema Híbrido. Sistema Embarcado. Educação. Desenvolvimento Web.

## ABSTRACT

This work depicts the development of a remote experiment based on a lighting system with alternative power generation wind-photovoltaic hybrid. A major factor that contributes to environmental degradation is the heavy use of non-renewable energy, due to this fact, we increased the studies in the area of renewable energy in the search for solutions to reduce the impacts caused by energy consumption. The introduction of hybrid power systems in educational institutions is quite important, both because of sustainability as the possibility of practical experience for students through access to it, because the practical experience enhances the learning process. However, the cost of implementation of such a system is high and not every institution can acquire it, another impasse factor is the geographical distance between the system and interested, because access to the system is restricted that location. The assay developed has a remote data acquisition system consists of hardware and software. The hardware performs the measurement of the information of the hybrid system's operating status, as follows, current solar charger, wind charger current electric charger current, battery voltage, electric charger on or off and on or off LED light fixture. The software is responsible for storing the data measuring the information the operating status of the hybrid system performed by hardware. These data are available on a web page that had been developed, which are also made available the images of the hybrid system are captured by an IP camera. Thus any user who has a device with a web browser and Internet access can take advantage of the hybrid system is current and view the captured images of it online, through the website. Thus, the remote experiment presented in this paper is considered a good alternative for remote experimentation has features that allow access through virtual to a real experiment, which will be available 24 hours a day, 7 days a week, where the user can meet the need for independent practical experience of their geographical location, acquiring and aggregating knowledge.

**Key words:** Remote Experimentation. Hybrid System. Embedded System. Education. Web development.

## LISTA DE ABREVIATURAS E SIGLAS

ABNT - Associação Brasileira de Normas Técnicas  
AC - Alternating Current  
ADSL - Asymmetric Digital Subscriber Line  
ARP - Address Resolution Protocol  
ASP - Active Server Pages  
BOOTP - Bootstrap Protocol  
CGI - Common Gateway Interface  
CSS - Cascading Style Sheets  
DC - Direct Current  
DHCP - Dynamic Host Configuration Protocol  
DMA - Direct Memory Access  
DNS - Domain Name System  
DRR - Double Data Rate  
EPROM - Erasable Programmable Read-Only Memory  
FTP - File Transfer Protocol  
FTPS - File Transfer Protocol Secure  
Full HD - Full High Definition  
GB - Gigabyte  
GHZ - Gigahertz  
GPU - Graphics Processing Unit  
HDMI - High-Definition Multimedia Interface  
HTML - Hypertext Markup Language  
HTTP - Hypertext Transfer Protocol  
IBM - International Business Machines  
ICSP - In-Circuit Serial Programming  
IDE - Integrated Development Environment  
ICMP - Internet Control Message Protocol  
IGMP - Internet Group Management Protocol  
ILR - Infraestrututa de Laboratório Remoto  
I/O - Input/Output  
IP - Internet Protocol

IPv4 - Internet Protocol version 4  
ISS - Internet Information Services  
I2C - Inter-Integrated Circuit  
JPG - Joint Photographic Experts Group  
JSP - JavaServer Pages  
Kg - Quilograma  
KM/H - Quilômetros por Hora  
LED - Light Emitting Diode  
LCD - Liquid Crystal Display  
LTDA - Limitada  
mA - MiliAmpère  
Mac OS - Macintosh Operating System  
MHZ - Megahertz  
mm - Milimetro  
m/s - Milisegundos  
NTP - Network Time Protocol  
NTSC - National Television Systems  
PAL - Phase Alternate Line  
PHP - Hypertext Preprocesso  
PPoE - Point-to-Point Protocol over Ethernet  
PWM - Pulse Width Modulation  
RAD - Rapid Application Development  
RAM - Random Access Memory  
RTC - Real Time Clock  
SeTIC - Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação  
SD - Secure Digital  
SFTP - Secure Shell File Transfer Protocol  
SMTP - Protocolo de Transferência de Correio Simples  
SPI - Serial Peripheral Interface  
SRAM - Static Random Access Memory  
SSI - Server Side Include  
TCP - Transmission Control Protocol  
TI - Tecnologia da Informação

TICs - Tecnologias da Informação e Comunicação

TV's - Televisões

USB - Universal Serial Bus

UART - Universal Asynchronous Receiver/Transmitter

UDP - User Datagram Protocol

UFSC - Universidade Federal de Santa Catarina

UPnP - Universal Plug and Play

URL - Uniform Resource Locator

UTP - Unshielded Twisted Pair

Vcc - Volts Corrente Continua

WTP - Web Tools Platform

XNU - X is Not Unix

# SUMÁRIO

<b>I. INTRODUÇÃO.....</b>	<b>15</b>
<b>I.I OBJETIVOS.....</b>	<b>16</b>
I.I.I Geral.....	16
I.I.II Específicos.....	16
<b>I.II JUSTIFICATIVA E CONTRIBUIÇÃO.....</b>	<b>17</b>
<b>I.III METODOLOGIA.....</b>	<b>17</b>
<b>I.IV ORGANIZAÇÃO DO TRABALHO.....</b>	<b>18</b>
<b>1. SISTEMA HÍBRIDO.....</b>	<b>20</b>
<b>1.1 Luminária LED.....</b>	<b>22</b>
1.1.1 Definições do LED.....	22
1.1.2 Dissipador para os LEDs.....	24
<b>1.2 Controlador da Fonte de Corrente.....</b>	<b>25</b>
1.2.1 Controlador MBI6661.....	26
1.2.1.1 Principais Parâmetros do Controlador MBI6661.....	26
1.2.2 Aplicação no Projeto .....	27
<b>1.3 Layouts dos Circuitos Eletrônicos .....</b>	<b>27</b>
1.3.1 Placa do Controlador de Corrente.....	27
1.3.2 Placa do Módulo de LEDs.....	28
<b>1.4 Montagem da Luminária.....</b>	<b>29</b>
1.4.1 Estrutura da Luminária.....	30
<b>1.5 Sistema de Armazenamento.....</b>	<b>30</b>
<b>1.5.1 Gerador Eólico.....</b>	<b>31</b>
1.5.1.1 Aerogerador Air Breeze.....	31
1.5.1.2 Funcionamento do Aerogerador.....	32
<b>1.5.2 Placa Fotovoltaica.....</b>	<b>33</b>
<b>1.5.2.1 Placa Utilizada.....</b>	<b>34</b>
1.5.2.1.1 Características Térmicas.....	34
1.5.2.1.2 Informações Adicionais.....	35
<b>1.5.2.2 Controlador de Carga Solar.....</b>	<b>36</b>
1.5.2.2.1 Especificações Técnicas do Controlador.....	37
<b>1.5.3 Bateria.....</b>	<b>38</b>

<b>1.6 Sistema Embarcado .....</b>	<b>38</b>
<b>1.6.1 Propriedades do Sistema Embarcado.....</b>	<b>39</b>
1.6.1.1 Hardware.....	39
1.6.1.2 Software.....	41
<b>2. EXPERIMENTAÇÃO REMOTA.....</b>	<b>43</b>
<b>2.1 Infraestruturas de Laboratórios Remotos.....</b>	<b>50</b>
2.1.1 Camada Física (Hardware) .....	52
<b>2.1.1.1 Câmera Web.....</b>	<b>54</b>
<b>2.1.1.2 Plataformas de Prototipagem Eletrônica.....</b>	<b>55</b>
2.1.1.2.1 Beaglebone Black.....	56
2.1.1.2.2 Raspberry Pi.....	57
2.1.1.2.3 Intel Galileo.....	59
2.1.1.2.4 Arduino.....	61
2.1.2 Camada Lógica (Software).....	62
<b>3. TECNOLOGIAS PARA DESENVOLVIMENTO DE AMBIENTE WEB.....</b>	<b>65</b>
3.1 Introdução à Web.....	65
<b>3.2 Recursos Necessários para Desenvolvimento de Ambientes Web.....</b>	<b>66</b>
3.2.1 Servidor Web.....	66
3.2.2 IDEs para Desenvolvimento de Ambientes Web.....	68
3.2.2.1 Eclipse.....	68
3.2.2.2 Aptana Studio.....	69
3.2.2.3 Visual Studio.....	70
3.2.2.4 NetBeans.....	71
<b>3.3 Linguagens de Programação para Web Sites.....</b>	<b>72</b>
3.3.1 Linguagens de Programação para páginas Dinâmicas – Server Side.....	73
3.3.1.1 ASP.....	73
3.3.1.2 PHP.....	75
3.3.1.3 JSP.....	75
3.3.2 Linguagens de Programação para páginas Dinâmicas – Client Side.....	77
3.3.2.1 HTML.....	77
3.3.2.2 CSS.....	78

3.3.2.3 Java Script.....	79
<b>3.4 Hospedagem de Páginas Web.....</b>	<b>80</b>
<b>3.5 Responsividade de Páginas Web.....</b>	<b>80</b>
<b>4. EXPERIMENTO REMOTO BASEADO EM UM SISTEMA HÍBRIDO DE GERAÇÃO DE ENERGIA.....</b>	<b>81</b>
<b>4.1 Desenvolvimento da Camada Física do Experimento Remoto - Controle e Funcionamento.....</b>	<b>82</b>
4.1.1 Placa de Aquisição de Dados.....	83
4.1.2 Plataforma de Prototipagem Utilizada no Sistema de Aquisição de Dados.....	86
4.1.3 Placas de Expansão Arduino ( <i>Shields</i> ).....	87
4.1.4 Arduino Ethernet Shield.....	88
4.1.5 Software do Sistema de Aquisição de Dados.....	91
4.1.6 Câmera IP D-Link DCS-950.....	95
<b>4.2 Desenvolvimento do Ambiente Web para o Experimento Remoto.....</b>	<b>103</b>
4.2.1 Linguagens e Ferramentas Utilizadas para o Desenvolvimento da Página Web.....	104
4.2.2 A Página Web.....	109
4.2.3 Estrutura que compõe a Página Web.....	113
<b>5 CONSIDERAÇÕES FINAIS.....</b>	<b>115</b>
<b>REFERÊNCIAS.....</b>	<b>117</b>

## I INTRODUÇÃO

Com a degradação do meio ambiente e as suas lamentáveis consequências, principalmente pelo uso exacerbado de energia, alavancaram-se os interesses e estudos na área de energias renováveis e limpas, que não agridem o meio ambiente. O objetivo é buscar soluções que aproveitem os recursos naturais para gerar energia sem causar danos ao meio ambiente.

Uma das soluções é a criação de sistemas híbridos de energia eólico-fotovoltaico que utilizam como fonte geradora o vento e o sol. Desta forma, é possível utilizar da energia gerada pelas fontes alternativas e ter um sistema para ser utilizado nas experimentações práticas para fins educacionais. Havendo poucos sistemas deste tipo, muitas vezes, os estudantes não têm acesso físico a algum para experimentação prática, pois o custo de aquisição é elevado e nem toda Universidade ou instituição de ensino pode adquiri-lo.

Por este motivo, devido à importância dessa área nos dias atuais, seria significativa a disponibilização da experimentação remota baseado em um sistema híbrido de energia, com a finalidade de suprir essa necessidade de experiência.

De acordo com os argumentos explicitados acima, este trabalho caracteriza a criação e disponibilização de um experimento remoto baseado em um sistema de iluminação com energia alternativa híbrida eólico-fotovoltaico. Para que possa ser realizado este experimento extraem-se dados do sistema híbrido e disponibiliza-os remotamente por meio de uma página web da qual fora criada no decorrer deste trabalho. Esses dados extraídos do sistema híbrido são informações do estado do mesmo, como, corrente do carregador solar, corrente do carregador eólico, corrente do carregador elétrico, tensão da bateria, carregador elétrico ligado ou desligado e luminária LED acesa ou apagada. Serão disponibilizadas também, através de uma câmera web, imagens capturadas do sistema híbrido.

Com o desenvolvimento deste trabalho propõe-se que qualquer pessoa que possuir um dispositivo com acesso à internet possa acessar a página Web onde o experimento estará disponível e realizar suas experiências com dados reais a um custo zero. O sistema ficará disponível 24 horas por dia, 7 dias por semana, sendo que a tecnologia utilizada na experimentação remota é de baixo custo comparado aos benefícios que a mesma proporciona.

## **I.I OBJETIVOS**

Esta sessão apresenta o objetivo geral e os objetivos específicos deste trabalho de conclusão de curso.

### **I.I.I Geral**

Este trabalho tem como objetivo estudar, projetar e disponibilizar o acesso ao experimento remoto baseado no sistema de iluminação com energia alternativa híbrido eólico-fotovoltaico, com a finalidade de obter informações técnicas e estatísticas do sistema, on-line.

### **I.I.II Específicos**

Para atingir o objetivo proposto, alguns objetivos específicos são requeridos, entre eles:

1. Estudar o Sistema Híbrido num todo, bem como seus componentes físicos e lógicos, e seu funcionamento;
2. Estudar o conceito de experimentação remota, o que é comum entre as experimentações remotas e o que é necessário para disponibilização da mesma;
3. Criar um sistema de aquisição de dados do sistema híbrido, para captar dados tais como: Corrente do carregador eólico, corrente do carregador solar, corrente do carregador elétrico e tensão da bateria.
4. Estudar as linguagens de programação, frameworks e ferramentas necessárias para o desenvolvimento da página Web;
5. Desenvolver a página Web;

6. Capturar os dados que são obtidos pelo sistema de aquisição do sistema híbrido e disponibilizá-los na página Web on-line;
7. Disponibilizar o acesso remoto da experimentação do sistema híbrido através da página Web.

## **I.II JUSTIFICATIVA E CONTRIBUIÇÃO**

Devido à importância do uso de fontes de energias limpas e renováveis nos dias atuais e o custo de um sistema de geração dessas energias ser relativamente alto dificultando a aquisição do mesmo pelas instituições de ensino, a criação e disponibilização de um experimento remoto baseado em um sistema de geração alternativa híbrido, foi considerado um fator de grande utilidade e importância, pois através dele qualquer pessoa que possuir um dispositivo com acesso à internet poderá acessá-lo através de uma página Web.

O experimento proposto contribuirá para fins educacionais, pois o usuário terá acesso a dados reais do estado do sistema híbrido em uma página web e poderá visualizar as imagens capturadas online na mesma página, o que dispensa o contato físico do usuário com esse tipo de sistema. O Sistema de Geração Alternativa Híbrido disponibilizado como experimento remoto trata-se de um sistema integrado de iluminação baseado em LEDs de alto brilho com alimentação através das energias alternativas geradas pelo mesmo. O sistema utiliza como fonte geradora de energia o sol, vento e a energia elétrica proveniente da rede de abastecimento concessionário local, esta última é utilizada caso a energia gerada pelas demais fontes alternativas não dê conta de manter a carga da bateria.

## **I.III METODOLOGIA**

Este trabalho foi desenvolvido com base em uma pesquisa qualitativa, bibliográfica e documental. A pesquisa qualitativa compreende um conjunto de técnicas interpretativas que visam descrever e a decodificar os componentes de um sistema complexo de significados (LAKATOS, 2010). A pesquisa bibliográfica abrange toda obra científica já tornada pública em relação ao tema de estudo, desde publicações avulsas, boletins, jornais, revistas, livros, pesquisa, monografias, teses, etc. Sua finalidade é colocar o pesquisador em contato direto com tudo o que foi escrito, dito ou

filmado sobre determinado assunto (LAKATOS, 2010). A análise documental constitui uma técnica importante na pesquisa qualitativa, seja complementando informações obtidas por outras técnicas, seja desvelando aspectos novos de um tema ou problema. (LUDKE; ANDRÉ, 1986).

As pesquisas realizadas possibilitaram o desenvolvimento de um experimento remoto baseado no sistema de iluminação com energia alternativa híbrida eólico-fotovoltaico. A fundamentação teórica é apresentada nos capítulos 1, 2 e 3. No capítulo 1 é abordado um estudo sobre o funcionamento e os componentes existentes em um sistema de iluminação com energia alternativa híbrida eólico-fotovoltaico, este sistema encontra-se instalado no campus da Universidade Federal de Santa Catarina na cidade de Araranguá. O capítulo 2 corresponde à experimentação remota, esta serviu para identificar os recursos necessários para o desenvolvimento de um experimento remoto. Constatou-se a necessidade de instalar uma web câmera para capturar as imagens do sistema e a criação de uma página web. O estudo realizado para o desenvolvimento da página web é apresentado no capítulo 3. A partir da fundamentação teórica dos capítulos 1, 2 e 3 foi possível adquirir conhecimento para desenvolver o experimento remoto proposto apresentado no capítulo 4.

O experimento remoto possui um sistema de aquisição de dados que tem como componente principal a plataforma de prototipagem eletrônica Arduino UNO R3 juntamente com um software, desenvolvido em linguagem Arduino, esta linguagem é semelhante ao C/C++. Os dados provenientes do sistema de aquisição e as imagens capturadas pela web câmera do sistema de iluminação com energia alternativa híbrida eólico-fotovoltaico foram disponibilizados na página web. A IDE Aptana Studio foi utilizada para o desenvolvimento da página web e o código implementado nas linguagens HTML, PHP, CSS e JavaScript.

## **I.IV ORGANIZAÇÃO DO TRABALHO**

Além desta introdução, este trabalho contém mais 4 (quatro) capítulos.

O **capítulo 1** apresenta uma breve introdução sobre sistemas híbridos, em seguida, descreve com detalhes o sistema híbrido específico, que será transformado em um experimento remoto, num todo, tal como seus componentes, estrutura lógica e física, e seu funcionamento.

O **capítulo 2** apresenta uma introdução sobre experimentação remota, definição de conceitos, suas vantagens e benefícios, sua importância no ensino e aprendizagem, e um estudo sobre a infraestrutura de laboratórios remotos (camadas física e lógica).

No **capítulo 3** faz-se um estudo das tecnologias disponíveis e necessárias, utilizadas no desenvolvimento de páginas web, linguagens de programação web, plataformas de desenvolvimento, frameworks etc. Enfim, contém um apanhado do estudo realizado sobre o referido argumento, para a possível construção da página web através da qual o experimento remoto será disponibilizado.

O **capítulo 4** descreve a proposta do experimento remoto, bem como o desenvolvimento do mesmo e seu funcionamento ilustrado e descrito por meio de um diagrama de bloco, sendo dividido em duas etapas. A primeira etapa será o desenvolvimento da camada física do experimento remoto - controle e funcionamento e a segunda etapa será o desenvolvimento do ambiente web para disponibilização do experimento remoto.

O **capítulo 5** apresenta as considerações finais.

## 1. SISTEMA HÍBRIDO

Atualmente, a procura de energia tem se concentrado fundamentalmente nas fontes de energia não renováveis, as quais têm tecnologia difundida, mas possuem um elevado impacto ambiental. Devido à situação ambiental global, e o uso exacerbado de energia cada vez maior, é de grande importância que esta tendência seja invertida, o que tem despertado o interesse pela exploração de fontes de energia renováveis.

Das fontes de energia renováveis, a energia solar e a eólica são as mais promissoras, pois são fontes abundantes e limpas, e também pelo fato de que seus crescimentos estão intimamente relacionados ao uso das novas tecnologias, o que minimiza os custos de instalação e de utilização das mesmas. Não obstante, essas duas fontes renováveis podem ser conciliadas formando assim um sistema híbrido eólico-solar, mantendo a confiabilidade e qualidade da energia fornecida, não danificando o meio ambiente (MONTEIRO; CABREIRA, 2007).

Os sistemas de energia híbridos possuem uma grande viabilidade, utilizando fontes alternativas de energia e de dispositivos de armazenamento para integrá-los (DALI et al, 2010). Esse tipo de sistema comparado com as usinas convencionais, possuem um porte inferior e opcionalmente são instalados mais próximos aos centros de carga.

Figura 1 – Configuração do sistema híbrido eólico-solar - rede elétrica.



Fonte: CEPEL (2011)

Considera-se um sistema híbrido de energia aquele que utiliza conjuntamente mais de uma fonte de energia, localmente disponíveis, para geração de energia elétrica, de modo que uma fonte possa complementar a eventual falta da outra (BARBOSA et al, 2004). Outra característica deste tipo de sistema é a existência de um bloco inteligente responsável pelo controle e condicionamento de potência (CARDIM, 2012). A Figura 1 mostra o modelo de sistema híbrido que retrata o sistema físico localizado no Campus da UFSC em Araranguá, o qual adiante será disponibilizado para experimento remoto através do acesso via Web.

A unidade de controle e condicionamento de potência (representada na Figura 1), possui a função de coordenar a operação do sistema e fornecer eletricidade adequada ao uso. Já o sistema de armazenamento tem a função de acumular energia gerada pelas fontes renováveis para utilização em períodos onde essa não é suficiente para atender a carga (CARDIM, 2012).

Um sistema de energia híbrido pode ser auto-independente ou conectado a rede de fornecimento de energia concessionário. No cenário independente, torna-se necessário que o sistema tenha capacidade de armazenamento suficiente para operar com as variações de alimentações de ambas as fontes envolvidas. Desta forma o mesmo pode ser considerado como uma micro-rede contendo fontes de geração de energia e de cargas próprias, sendo equipada com dispositivos eletrônicos de potência para o ajuste da tensão e frequência de forma correta que irá garantir a partilha de carga adequada entre as diversas fontes. Na configuração conectada à rede, as fontes de energia podem fornecer energia para às cargas e à rede elétrica. É interessante que este modo seja utilizado como backup do sistema de geração, pois ao ser conectado a rede a capacidade de aproveitamento do mesmo é menor (MARCELINO, 2013).

A construção de um sistema híbrido tem como objetivo aproveitar o máximo dos recursos naturais de uma determinada região, sendo necessário um estudo prévio afim de obter informações se o mesmo poderá ser viável no local onde irá ser implantado.

Uma boa estratégia de operação irá proporcionar em um ótimo custo-benefício, tornará o sistema confiável, maximizará o aproveitamento de energia bem como a vida útil do sistema, além de minimizar os custos de implantação, operação e manutenção do mesmo (GRUPO DE TRABALHO DE ENERGIA SOLAR - GTES (ORG.), 2014).

O Sistema Híbrido que adiante será disponibilizado como experimento remoto é constituído por uma luminária LED de alta eficiência, um mini aerogerador, uma célula solar, sistema de bateria e um controle central inteligente. O sistema controla a luminária para acender e apagar conforme a luminosidade através de um sensor de luz e o controle inteligente. A luminária recebe a energia da bateria, que por sua vez é carregada via célula solar ou energia eólica. Quando não houver vento ou luminosidade suficiente, a carga da bateria acontece via rede da concessionária, por isto o sistema é denominado híbrido. O controle inteligente acompanha estas situações em tempo real e responde conforme a necessidade e a programação realizada. Os tópicos seguintes descreverão os vários componentes do sistema híbrido, bem como o funcionamento e interligação dos mesmos.

### **1.1 Luminária LED**

O LED (Diodo Emissor de Luz) é um componente eletrônico capaz de emitir luz através de um fenômeno chamado eletroluminescência responsável por transformar a energia elétrica em radiação visível (BLEY, 2012). Inicialmente o LED foi criado para suportar aplicações pequenas. Com a evolução desta tecnologia nos últimos anos foi possível aprimorar ainda mais a sua capacidade de uso, podendo ser encontrado facilmente em diversos sistemas como de automóveis, televisores e de iluminação em larga escala.

O LED opera como um semicondutor e quando polarizado diretamente é capaz de conduzir corrente elétrica liberando fótons de luz. Sua construção ocorre a partir de um condutor ao qual são acrescentadas impurezas, a luz irá variar de acordo com o material utilizado (MARCELINO, 2013).

#### **1.1.1 Definições do LED**

Ao escolher um LED, uma das principais características a ser observada é a cor que a lâmpada utilizada irá possuir, pois não adianta projetar uma boa luminária com materiais de qualidade se a cor transmitida pelo LED não for propícia para aquele determinando ambiente onde a mesma irá iluminar. Além da cor, existem outras características que são fundamentais para a escolha do LED tais como a temperatura da cor, a potência, a luminância, temperatura de junção, o custo entre outros (MARCELINO, 2013).

A partir de um estudo sobre um corpo negro radiador, onde se elevou este material a altas temperaturas e assim ele emitiu uma luz com determinada cor, se estabeleceu temperaturas para as cores, sendo assim a cor do LED é representada em Kelvin. A fim de manter um índice de reprodução de cor elevado, decidiu-se utilizar uma cor branca para os LEDs. Como opções de branco têm o branco quente com temperaturas de cor de aproximadamente 3300K, o branco neutro com temperaturas de cor de aproximadamente 4200K e o branco frio com temperaturas de cor de aproximadamente 7500K. Pode haver LEDs com diferentes tons de branco (MARCELINO, 2013).

Para a lâmpada utilizada no projeto foram utilizados LEDs com potência de 1W, já que se trabalhará com módulos e com um grupo maior de LEDs. Procurou-se com isto uma melhor distribuição luminosa e um sistema com portabilidade variando de acordo com o ambiente que for iluminar.

Tabela 1 – Comparações envolvendo LEDs de 1W

Característica	Cor	Apollo	Cromateck	Seoul
Potência Dissipada	Branco Quente	1,25W	-	4W
	Branco Neutro	-	-	4W
	Branco Frio	1,25W	-	4W
Temperatura de Junção	Branco Quente	110°C	125°C	145°C
	Branco Neutro	-	125°C	145°C
	Branco Frio	110°C	125°C	145°C
Fluxo Luminoso	Branco Quente	60lm	60lm	85lm
	Branco Neutro	-	70lm	100lm
	Branco Frio	75lm	90lm	120lm
Resistência Térmica	Branco Quente	9°C/W	13°C/W	6,2°C/W
	Branco Neutro	-	13°C/W	6,2°C/W
	Branco Frio	9°C/W	13°C/W	6,2°C/W

Fonte: Cromax Eletrônica, Seoul e SunLED (2010)

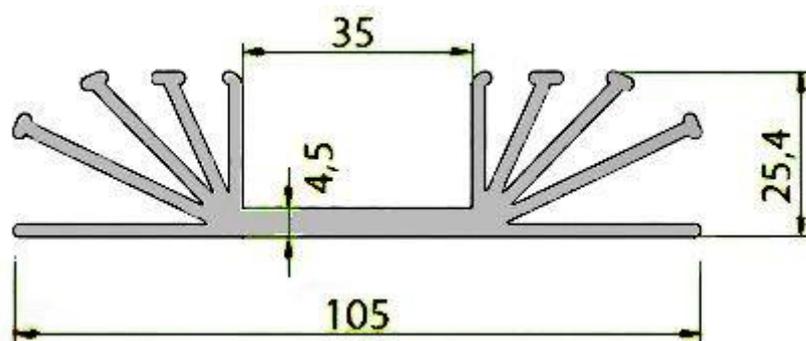
A temperatura de junção influencia diretamente na luz emitida pelo LED, como também na vida útil. Pois se o mesmo trabalhar a temperaturas extremas, este fator pode interferir diretamente no correto funcionamento do LED. Sendo esta outra característica importante a considerar na escolha do LED (MARCELINO, 2013).

Através de pesquisas realizadas foi possível chegar a três tipos de fabricantes de LEDs de auto brilho que se adequaram ao projeto, sendo eles: Apollo, Cromateck e Seoul. Após comparações, que podem ser analisadas na Tabela 1, optou-se por utilizar o LED Apollo, pois devido as suas características foi o que obteve um melhor funcionamento, comparado com os outros, para ser utilizado na luminária. Tendo temperatura de junção de  $110^{\circ}\text{C}$ , potência dissipada de 1,25W, fluxo luminoso de 60lm a 75lm e resistência térmica de  $9^{\circ}\text{C/W}$ . Através de dados técnicos viu-se que atingirá uma temperatura menor em funcionamento, pois a resistência térmica é a menor, se comparada com a potência do LED, isto faz com que sua vida útil se prolongue e tenha um melhor fluxo luminoso (MARCELINO, 2013).

### 1.1.2 Dissipador para os LEDs

Para que parte do calor gerado pelos LEDs seja eliminada tornou-se necessário a fixação de dissipadores de calor na placa de alumínio do LED. Após estudos e cálculos realizados optou-se pelo perfil de dissipador, ilustrado na Figura 2, como sendo adequado.

Figura 2 – Perfil de dissipador de calor.



Fonte: Hs Beneficiamento de Alumínio LTDA (2008/2009)

Este perfil possui a resistência térmica de  $2,47^{\circ}\text{C/W}$  para 4'' (quatro polegadas) de comprimento. Para ajudar a conduzir o calor da placa para o dissipador utilizou-se

pasta térmica, alojando o LED no dissipador de forma plana afim de obter uma melhor condução térmica.

## 1.2 Controlador da fonte de corrente

A operabilidade das lâmpadas de LED deve possuir um alto grau de confiabilidade, pois as mesmas deverão se manter acesas durante todo o período noturno, contribuindo na iluminação da Universidade.

Para um bom funcionamento do LED é de grande importância que se utilize uma fonte de corrente adequada, garantindo um valor de corrente constante. Esse valor constante é necessário e muito importante, pois faz com que a intensidade de brilho permaneça contínua, evitando oscilações que possam danificar o LED (MARCELINO, 2013). O responsável por essa função, de manter o valor de corrente constante, é o controlador da fonte de corrente.

É necessário que a luminária LED possua uma boa eficiência. Desta forma algumas características foram observadas na escolha do controlador. As principais são Tensão (V) de entrada, a Eficiência ( $\eta$ ) e a Corrente (A) máxima de saída (MARCELINO, 2013). A Tensão mínima que o controlador irá operar é 12 Vcc (Vcc = Tensão em Corrente Contínua), pois a bateria utilizada é de 12 Vcc.

Depois de observadas algumas características foi escolhido o controlador que mais se adequou ao projeto. A Tabela 2 ilustra as características do controlador escolhido.

Tabela 2 – Características do controlador escolhido.

Conversor DC-DC							
Descrição	Topologia	Vin. (V)	Isaídamáx (mA)	Eficiência	Dimmer, Isolação, MOSFET, PFC etc.	PKG	Fabricante
MBI6661	Buck	9~60	1000	Higher sustaining Voltage, OCP, 97% $\eta$	PWM Dimmer, intgMOSFET	TO252, SOP	Macroblock (MBI)

Fonte: Excelpoint (2011)

Optou-se por utilizar o controlador MBI6661, por este ser o mais adequado para o uso na fonte de corrente da lâmpada LED. O mesmo opera numa faixa de tensão de 12 Vcc e possibilita o uso de três LEDs em série que operam com 3Vcc cada, na saída.

### **1.2.1 Controlador MBI6661**

A escolha do controlador MBI6661 se deu pelo fato de o mesmo operar dentro das limitações impostas pela luminária. Fazendo o uso do mesmo pode-se garantir a corrente necessária para a luminária LED.

“O controlador MBI6661 é um conversor DC/DC Buck 60V e 1A de alta eficiência, para utilizações em diversas aplicações como iluminação externa com lâmpadas LED e iluminação de túneis. A tensão de entrada tem um grande intervalo variando de 9V a 60V e uma alta eficiência de aproximadamente 97%. A máxima corrente de saída é 1A com um erro de  $\pm 5\%$ ” (MARCELINO, 2013).

O MBI6661 pode operar com até 14 LEDs de 1W ligados em série, recebendo uma tensão de entrada de 60V controlando a corrente de saída para a corrente nominal do LED. Uma grande vantagem deste tipo de controlador é que o mesmo possui um baixo custo e ocupa um pequeno espaço físico na placa (MARCELINO, 2013).

Projetado com diversas proteções dentre elas, proteção contra pico proteção contra corrente de pico, baixa tensão de entrada, térmica, sobre corrente, LED aberto e curto-circuito no LED. O MBI6661 também possui um pino para conectar a um dissipador térmico (MARCELINO, 2013).

#### **1.2.1.1 Principais Parâmetros do Controlador MBI6661**

- Corrente Máxima na Saída: 1A;
- Tensão de Entrada: 9~60V;
- Corrente de Saída Ajustável por uma entrada PWM (Nível Alto=2,5V, Nível Baixo=0,8V);
- Proteções: Proteção térmica / Circuito-aberto / Corrente e tensão de pico / Baixa tensão de entrada / Sobre corrente / LED aberto / Curto no LED.

### 1.2.2 Aplicação no projeto

O sistema trabalha com o máximo de corrente de saída, 1A, sendo assim desnecessário o uso de um circuito *PWM* para controlar a corrente de saída, por consequência um número menor de controladores MBI6661. A corrente em cada ramo tem o valor de 333,33mA. Assim o projeto opera com 1 controlador MBI6661 para 9 LEDs.

### 1.3 Layouts dos Circuitos Eletrônicos

O software utilizado na criação do projeto das placas foi o Proteus com sua extensão, o Ares.

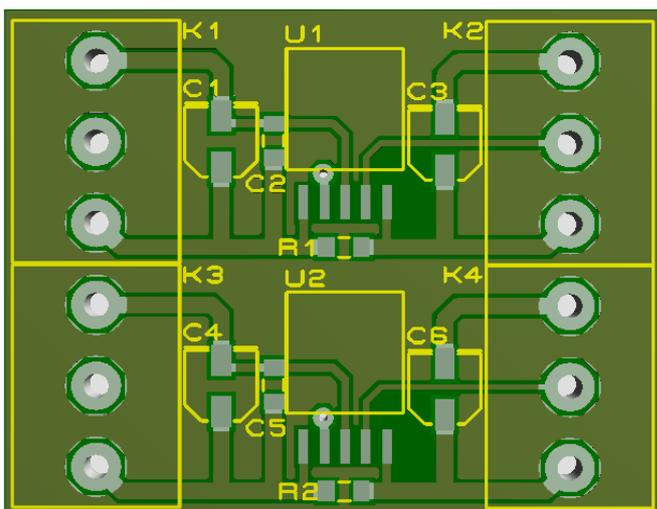
A placa do módulo da lâmpada de LED possui um módulo de 18W com 18 LEDs. A placa é de face simples e para uma melhor transferência de calor entre os LEDs e o dissipador a mesma é constituída de alumínio.

#### 1.3.1 Placa do Controlador de Corrente

A placa do controlador de corrente foi projetada com base nos valores nominais do LED Apollo, a fim de se obter um melhor desempenho dos LEDs. Para o controle do mesmo, é utilizado o CI MBI6661 modelo GSD.

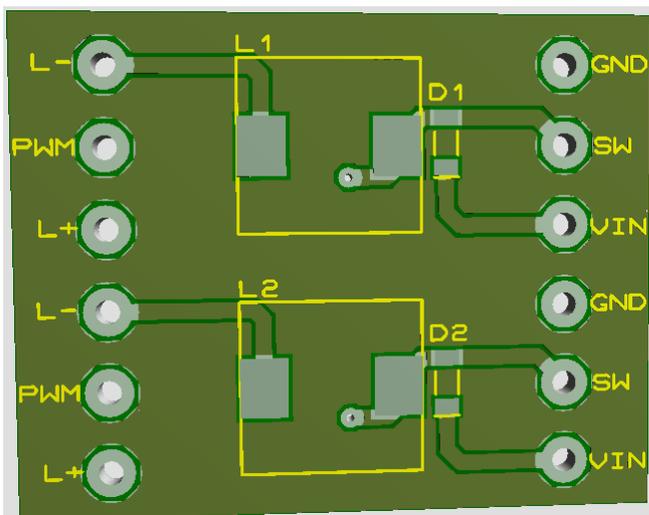
O layout da placa foi desenhado baseado nos componentes externos e no modelo do CI. Na Figura 3 e Figura 4 pode-se analisar uma demonstração 3D do layout.

Figura 3 – Vista Superior da Placa do Controlador de Corrente



Fonte: Marcelino (2013)

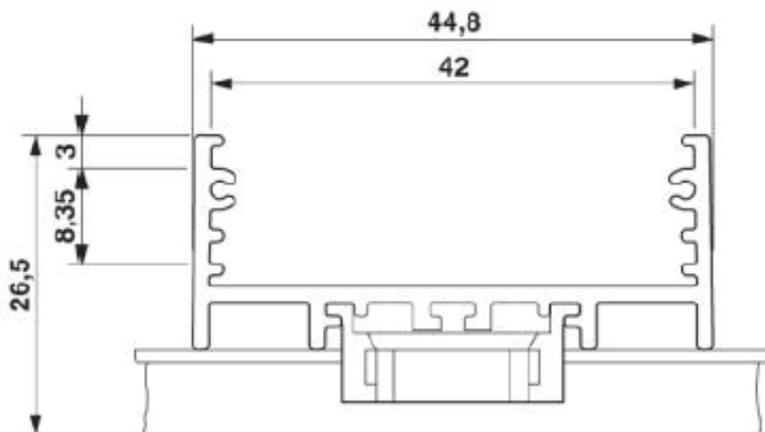
Figura 4 – Vista Inferior da Placa do Controlador de Corrente



Fonte: Marcelino (2013)

As placas foram projetadas com dimensões apropriadas para entrar no perfil de 42. Na Figura 5 pode-se visualizar o desenho do perfil.

Figura 5 – Perfil do Perfil 42

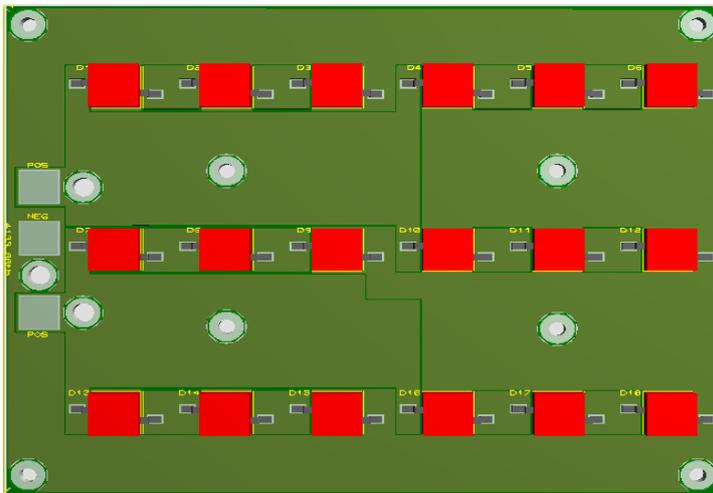


Autor: Phoenix Contact (2004)

### 1.3.2 Placa do Módulo de LEDs

A placa foi construída com base nas dimensões do dissipador utilizado e na quantidade de LEDs por módulo. A Figura 6 ilustra a modelagem em 3D, onde os pontos vermelhos representam a posição que o LED estará disposto na placa.

Figura 6 – Vista Superior da Placa do módulo de LEDs

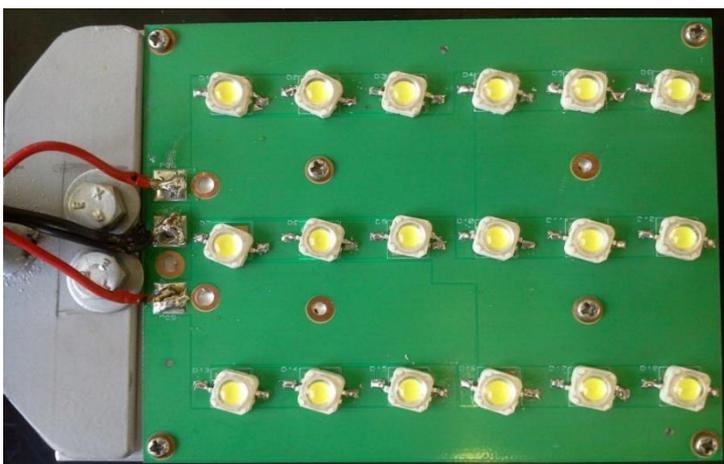


Fonte: Marcelino (2013)

#### 1.4 Montagem da Luminária

As placas foram produzidas em alumínio com as trilhas em cobre. Para a soldagem dos componentes foi utilizado a estação de retrabalho e solda. A Figura 7 ilustra a placa inteira com os LEDs e cabos de alimentação já fixados pela soldagem. Optou-se por não utilizar conector na alimentação do módulo, a fim de evitar mau contato no sistema.

Figura 7 – LEDs Soldados na Placa



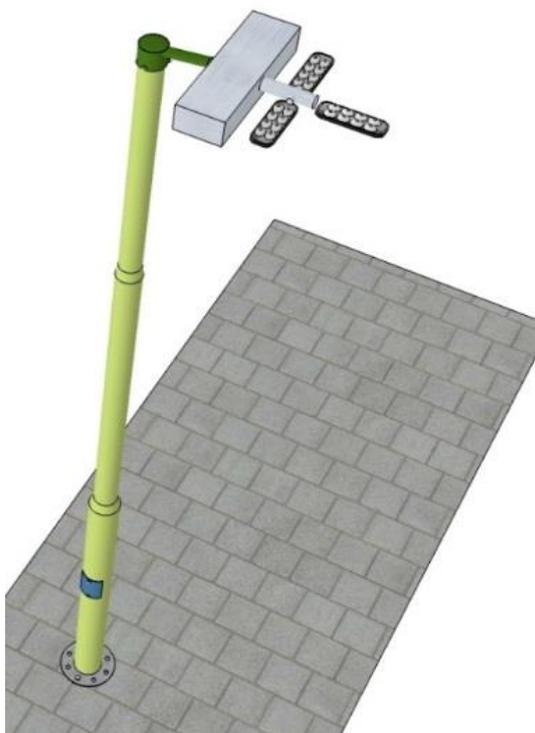
Fonte: Marcelino (2013).

### 1.4.1 Estrutura da Luminária

A estrutura da luminária foi projetada para suportar de 1 a 5 módulos. De acordo com a necessidade de iluminância do ambiente onde a mesma foi aplicada, optou-se por utilizar três módulos (ou pétalas), sendo que cada módulo possui cerca de 20W, totalizando 60W de potência. Foi preciso também um alojamento para acomodar as fontes e demais componentes necessários. Este bloco possui uma espera para encaixe em tubulação padrão de luminárias públicas.

A estrutura mostrou ser muito robusta e flexível, podendo operar com diversas regulagens fim de obter-se o melhor ângulo de funcionamento. A Figura 8 ilustra a estrutura física da luminária já fixada e instalada no poste de sustentação.

Figura 8 – Visão da luminária com poste



Fonte: MARCELINO (2013)

### 1.5 Sistema de Armazenamento

O sistema de armazenamento de energia elétrica tem papel fundamental em um sistema híbrido. Um sistema robusto e confiável evita desperdícios, aproveita ao máximo dos recursos naturais disponíveis, aumenta o rendimento e provê um ótimo custo-benefício. Para que se obtenham bons resultados é necessário que a escolha dos

equipamentos que compõe o sistema, bem como o modo em que estes operam estejam dispostos da melhor maneira possível, por isso cabe à equipe de projeto utilizar da melhor estratégia com base em estudos e dados técnicos.

### **1.5.1 Gerador Eólico**

A escolha de um gerador eólico em um sistema híbrido é de grande importância, pois este deve se adequar as condições de projeto do sistema e ambiente no qual será implantado.

“Os geradores eólicos têm como funcionalidade a produção de energia elétrica a partir da energia mecânica proveniente da turbina eólica. Os requisitos necessários de um gerador prendem-se, sobretudo, com a simplicidade do uso, da longa duração, da baixa manutenção e do baixo custo de investimento. Cumpridos estes requisitos, é escolhido o gerador apropriado” (FERREIRA, 2011).

Conforme características citadas acima, o aerogerador (ou gerador eólico) escolhido foi o Air Breeze, considerado como suficiente e adequado. Nos subtópicos seguintes, citam-se as especificidades do mesmo, bem como seu funcionamento.

#### **1.5.1.1 Aerogerador Air Breeze**

Modelo: Land;

Número Serial: BR0014453;

Fabricante: Southwest Windpower;

Representante no Brasil: Energia Pura Empreendimentos LTDA;

Tensão de Saída: 12V;

Corrente: 25A;

Peso: 6Kg;

Velocidade do vento no arranque: 2,7m/s aproximadamente 10Km/h;

KWh/mês: 38. Considerando ventos com velocidade média de 5,4 m/s;

Velocidade máxima de geração: 15,6 m/s;

Velocidade máxima do vento: 49,5 m/s;

Gama de temperatura: -10°C a 40°C;

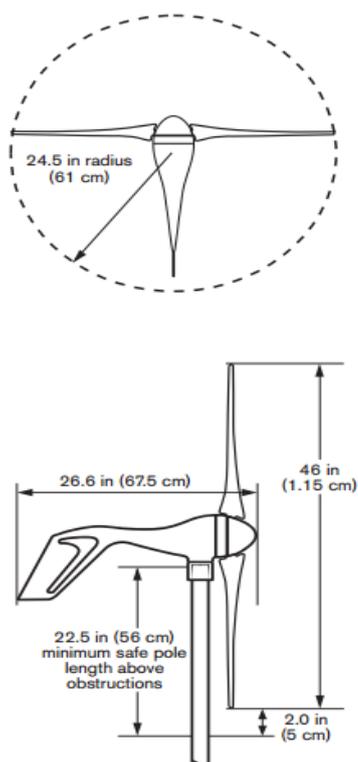
Sistema 12V, tensão de regulação: 15V;

Gama de ajuste do regulador: 13,6 a 17,0V;

Fusível: 20A Lento.

A Figura 9 mostra as dimensões do aeorogerador Air Breeze.

Figura 9 – Dimensões do Aerogerador



Fonte: Air Breeze Owner's Manual (2008)

### 1.5.1.2 Funcionamento do Aerogerador

As turbinas eólicas capturam a energia cinética gerada pelos ventos e utilizam esta energia para gerar o movimento de rotação que aciona um alternador com objetivo de produzir energia elétrica (REIS et al, [s.d]). A energia elétrica produzida deverá ser

estabilizada a um nível de tensão que possa carregar as baterias existentes no sistema. É imprescindível que se faça a utilização de um sistema que evite sobrecargas nas baterias, além disso, o sistema deverá retomar o carregamento na medida em que a tensão diminuir. É importante o uso de um mecanismo que proteja a turbina contra a força dos ventos externos que possam causar algum dano a mesma.

O aerogerador utilizado cumpre todos os requisitos citados acima, faz uso de um alternador trifásico com ímã permanente sem escovas e componentes eletrônicos controlados por um microprocessador com intuito de aperfeiçoar a produção de energia (Marcelino, 2013). O microprocessador estabiliza a carga do alternador de modo a manter que as pás operem no seu melhor ângulo estratégico. Os resultados obtidos são alta produção de energia, alta eficiência das pás com um baixo nível de ruído.

### **1.5.2 Placa Fotovoltaica**

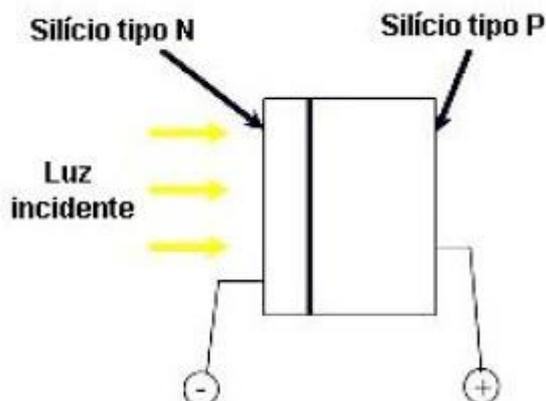
As placas fotovoltaicas são compostas de células solares de silício. Elas são semicondutoras de eletricidade porque o silício é um material com características intermediárias entre um condutor e um isolante (NASCIMENTO, 2004).

O silício apresenta-se normalmente como areia. Através de processos adequados se obtém o silício na sua forma pura. O cristal de silício na sua pureza não possui elétrons livres, sendo um mau condutor elétrico. Para alterar esta composição se acrescentam outros elementos, processo denominado como técnica de dopagem. A dopagem do silício com o fósforo trás como resultado materiais portadores de cargas negativas (silício tipo N). Acrescentando-se Boro ao invés de Fósforo se obtém um material com características inversas, que possuem cargas positivas livres (Silício tipo P). Cada célula solar é composta por uma camada mais fina tipo N e outra com maior espessura do tipo P (NASCIMENTO, 2004).

Como ilustra a Figura 10.

Ambas as camadas de forma individual são eletricamente neutras, mas ao se unirem formam um campo elétrico. Ao ocorrer a incidência de luz sobre as células fotovoltaicas os fótons chocam-se com os outros elétrons existentes na estrutura da célula fornecendo-lhes energia e transformando-os em condutores. Os elétrons fluem da camada “P” para a camada “N” gerando um campo elétrico quando estão ligados por um condutor externo, enquanto houver incidência de luz o fluxo é mantido. A intensidade de corrente elétrica irá variar conforme a incidência de luz.

Figura 10 – Ilustração básica de uma célula de silício.



Fonte: NASCIMENTO (2004)

Uma célula fotovoltaica não é capaz de armazenar energia elétrica, apenas mantém o fluxo de elétrons estabelecidos (NASCIMENTO, 2004). Para que a energia gerada seja armazenada é necessário um sistema de baterias.

### 1.5.2.1 Placa Utilizada

Optou-se por utilizar o modelo de placa fotovoltaica Sun module - SW85 poly RNA. A Tabela 3 mostra algumas características do material que compõem as células.

Tabela 3 – Características dos materiais da placa fotovoltaica

Células por módulo	36
Tipo da célula	Policristalino
Dimensões da célula	94mm x 156mm
Frontal	Vidro temperado (EN 12150)

Fonte: SolarWorld (2011)

#### 1.5.2.1.1 Características Térmicas

Na Tabela 4 podem-se observar as características térmicas do módulo solar e a variação das grandezas técnicas em relação à temperatura.

Tabela 4 – Características térmicas.

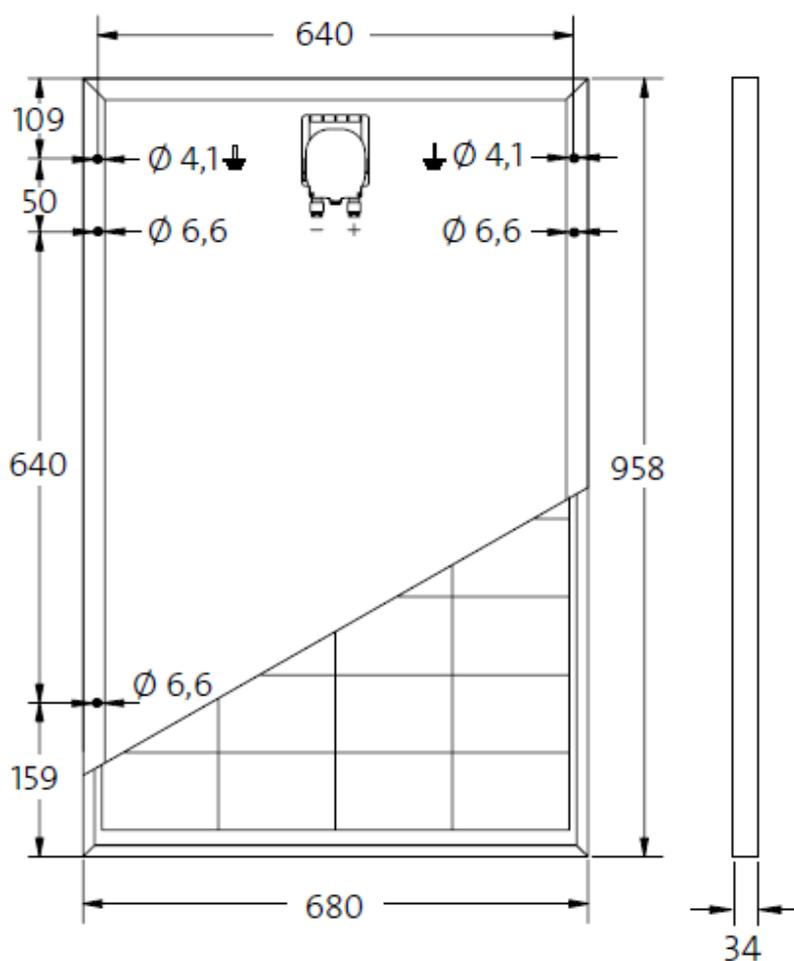
NOCT	46°C
TC Icc	0,034% / K
TC Uca	-0,34% / K
TC Ppcf	-0,48% / K

Fonte: SolarWorld (2011)

### 1.5.2.1.2 Informações Adicionais

Logo abaixo na Tabela 5, têm-se algumas informações adicionais a respeito da placa solar fotovoltaica. E, na Figura 11, mostram-se as dimensões da mesma.

Figura 11 – Dimensões da placa solar fotovoltaica



Fonte: SolarWorld (2011)

Tabela 5 – Informações adicionais

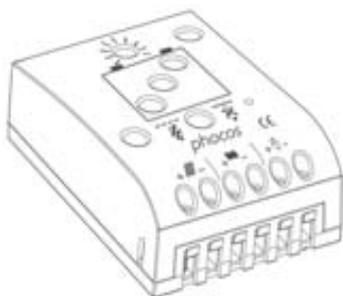
Tolerância da Potência	-5/10%
Caixa de conexões	Proteção IP65
Máximo diâmetro do cabo	6,9mm
Máxima bitola do cabo	4mm <sup>2</sup>

Fonte: Solar World (2011)

### 1.5.2.2 Controlador de Carga Solar

É necessário o uso de um controlador para que o fluxo de energia na placa seja armazenado na bateria do sistema. Optou-se por utilizar o controlador CML20 – 2.2.A. Figura 12 mostra um desenho do mesmo.

Figura 12 – Imagem ilustrativa do controlador



Fonte: Phocos (2009)

Dentre os recursos, que equipam o controlador, destacam-se:

- Mostrador do estado de carga claro e legível por LEDs.
- Sinal acústico quando houver mudança no estado de carga.
- Função desconectar baixa tensão (LVD) regulada pelo estado de carga ou tensão.
- Grampos de conexão de 16mm<sup>2</sup>.
- Proteção eletrônica completa.

O circuito possui um microcontrolador que permite uma alta eficiência no carregamento da bateria do sistema, além de um display que exhibe o estado de carga, erros, e outras funções.

“O controlador utiliza um método de carregamento por PWM, com compensação de temperatura e três estágios (boost – equalization – float), sendo ajustável para baterias de gel (eletrólito sólido) ou eletrólito líquido. O controlador possui ajuste do LVD (Desconectar por Baixa Tensão) para operar por estado de carga controlado ou tensão controlada. Ainda possui a opção de desligamento da função sinal acústico” (MARCELINO, 2013).

#### **1.5.2.2.1 Especificações Técnicas do Controlador**

Corrente Nominal: 20A;

Tensão do sistema: 12V / 24V;

Tensão de reforço: 14,5V / 29V;

Tensão de equalização: 14,8V / 29,6V;

Tensão Flutuante: 13,7V / 27,4V;

Função desconectar baixa tensão: 11,4 – 11,9V / 22,8V – 23,8V controlado pelo estado da carga; 11V / 22V controlado pela tensão;

Tensão para reconexão da carga: 12,8V / 25,6V;

Compensação de temperatura: -4mV/célula x K;

Máxima carga do painel solar: 20A;

Máxima carga da bateria: 20A;

Peso: 180g;

Temperatura de operação: -40 a 50°C;

Corrente de consumo do controlador: <4mA;

Dimensões (LxCxH): 80x100x32mm;

Tipo de proteção: IP22.

### 1.5.3 Bateria

A Bateria utilizada no Projeto foi a Freedom, modelo DF2000, do fabricante Johnson Controls. Conforme pode ser observada na Figura 13.

Figura 13 – Baterias Freedom



Fonte: Marcelino (2013)

Características Físicas da Bateria:

- Baterias de chumbo-ácido, estacionária;
- Dimensões: Largura (172mm), Comprimento (330mm) e Altura (240mm);
- Peso: 27,1 Kg;
- Capacidade a temperatura ambiente de 25°C: 115Ah.

### 1.6 Sistema Embarcado

Sistemas embarcados são sistemas que se comportam como um computador, porém de maneira independente e são projetados para operar sobre uma determinada tarefa. Neste tipo de sistema o usuário final normalmente não tem acesso ao programa que foi embutido no dispositivo (CUNHA, 2013). Para o Projeto foi necessário a criação de um sistema embarcado. Nas subseções seguintes, segue-se o detalhamento sobre o mesmo.

#### 1.6.1 Propriedades do Sistema Embarcado

O sistema embarcado criado tem como função principal realizar a escolha da fonte de energia, seja esta placa fotovoltaica, gerador eólico ou conectado a rede

concessionária de fornecimento de energia. Todas estas têm como propósito realizar a carga da bateria para que o painel de LED possa ser acionado quando necessário. Para que essas funções sejam realizadas um hardware microcontrolado foi desenvolvido contendo uma placa de circuito impresso com os devidos componentes e um microcontrolador.

### 1.6.1.1 Hardware

Devido ao fato de que a programação do sistema não demandará de muita memória de processamento e nem de armazenamento, optou-se por utilizar o microcontrolador PIC16F628A. Na Tabela 6 podem-se conferir as principais características do microcontrolador utilizado.

Tabela 6 – Parâmetros do PIC16F628A

Parâmetro	Valor
Tipo de memória programável	Flash
Tamanho da memória programável	3,5KB
Velocidade da UPC	5 MIPS
Tamanho da memória RAM	224 Bytes
Tamanho da EEPROM	128 Bytes
Periféricos da comunicação digital	1-A/E/USART
Captura / Comparação / PWM periféricos	1 CCP
Quantidade de Timer	2 x 8 bit, 1 x 16 bit
Quantidade de comparadores	2
Faixa de temperatura de operação	-45 °C até 125 °C
Faixa de tensão de operação	2 V até 5,5 V
Número de pinos	18

Fonte: Marcelino (2013)

Para o acionamento de cargas de maior potência foram necessários componentes que dão suporte a corrente solicitada. Desta forma optou-se por utilizar os relés de acionamento do fabricante Metaltex, modelo MA1RC2, que funcionam com uma tensão de bobina de 12 Vcc e corrente de 12 mA. Estes relés podem acionar cargas de até 220 Vca e 16 A.

Foi instalado na placa de circuito um módulo de lâmpadas LEDs para facilitar os testes realizados no sistema e indicar alguma anomalia no mesmo. Cada LED possui uma cor diferente representando as informações pré-determinadas.

O sistema opera com uma bateria de 12Vcc, porém alguns componentes como o microcontrolador operam em 5Vcc. Desta forma faz-se uso de reguladores de tensão. O mais indicado é o LM7805, ele regula a tensão de saída para 5Vcc, a partir de uma tensão de entrada de no mínimo 7Vcc. A corrente que este componente suporta é de 1,5A, logo ele suporta cinco vezes mais do que o circuito consome, não havendo problema. A Tabela 7 contém as principais características do regulador de tensão utilizado.

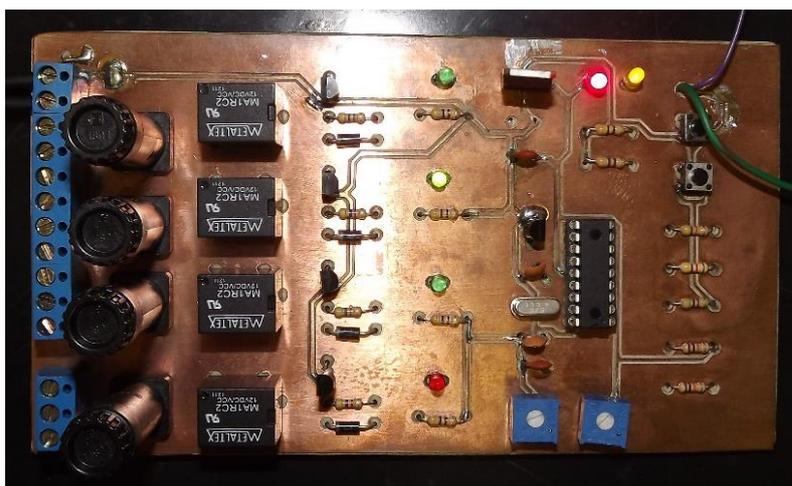
Tabela 7 – Parâmetros do regulador de tensão LM7805

Parâmetro	Mínimo	Máximo
Tensão de entrada	7 V	25 V
Corrente	-	1,5 A
Temperatura de operação	0°C	125 °C

Fonte: Marcelino (2013)

A placa de circuito impresso foi desenvolvida para que se possa fazer o controle inteligente das fontes de energia utilizadas pelo sistema a fim de carregar a bateria. A mesma possui componentes eletrônicos que possibilitam este controle inteligente. A Figura 14 mostra a placa de circuito impresso desenvolvida.

Figura 14 – Placa do protótipo montada



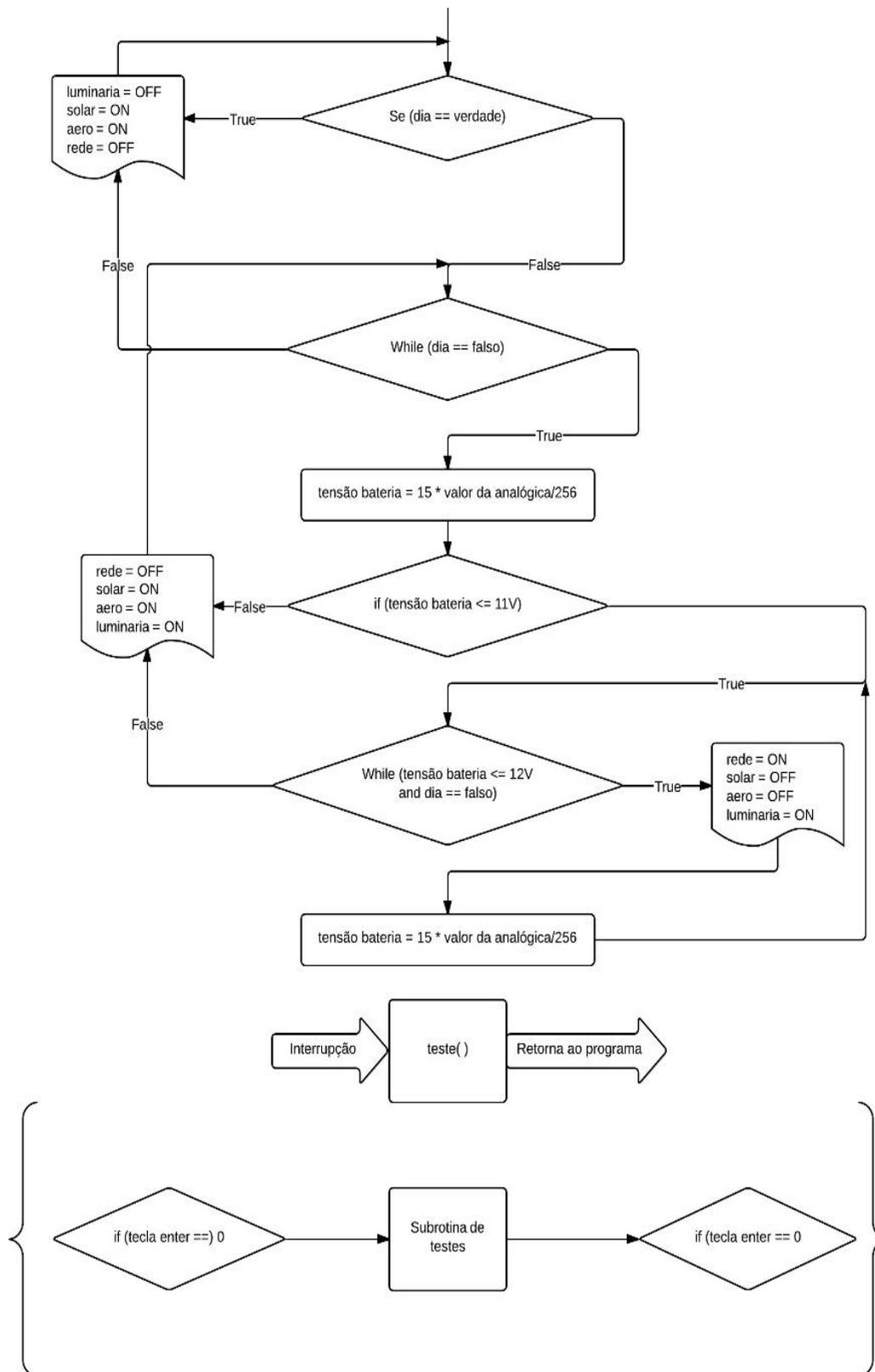
Fonte: Marcelino (2013)

### 1.6.1.2 Software

Como o controlador utilizado pertence à família PIC, o mesmo pode ser programado em linguagem *Assembly* ou linguagem C. A programação foi realizada no compilador PCW C++.

Como a bateria poderá ser carregada por três fontes de energia diferentes o programa armazena a tensão da bateria para saber qual fonte utilizar para o carregamento. No período em que a luminária estiver desligada ele deverá manter a placa fotovoltaica e o gerador eólico captando energia para o carregamento da bateria. Quando a luminária entrar em funcionamento, neste caso no período da noite, inicia-se a medição do valor de tensão da bateria que só será carregada pela rede concessionária caso a tensão atingir um valor menor que 11 Vcc. A intenção é utilizar ao máximo as fontes alternativas evitando o uso da rede concessionária. Na Figura 15 pode-se observar o fluxograma, ele auxiliará para um melhor entendimento sobre o funcionamento do sistema embarcado.

Figura 15 – Fluxograma do Funcionamento do Sistema Embarcado



Fonte: Marcelino (2013)

## 2. EXPERIMENTAÇÃO REMOTA

A experimentação prática é um dos métodos de ensino que reforça o processo de aprendizagem, sendo fundamental nas diversas áreas técnicas, principalmente nas áreas da ciência e engenharia. É importante que os alunos possam conhecer e se familiarizarem com as aplicações práticas de conceitos, fórmulas, para que obtenham conhecimentos teóricos a um sólido saber prático (SOUSA, 2008).

Os alunos aprendem mais ou menos informação da forma que a mesma é apresentada. A Figura 16 mostra uma relação entre vários tipos de métodos de ensino e a porcentagem de informação aprendida (SOUSA, 2008).

Figura 16 – Pirâmide de aprendizagem



Fonte: Sousa (2008)

É evidente a importância da interação do aluno com a prática no aprendizado, além disso, o mesmo tem a oportunidade de conhecer novas tecnologias que são as principais aliadas na construção de um experimento. Cabe às instituições disponibilizarem de laboratórios para que essa forma de ensino possa ser repassada aos interessados.

Um laboratório de ensino onde possam ser ministradas aulas práticas é essencial, através dele os alunos podem associar a teoria que aprenderam em sala de aula com a prática proposta, manuseando e interagindo com experimentos e materiais que

complementam o processo de aprendizagem. Porém nem sempre é possível dispor de um espaço físico para a realização das aulas práticas ou até mesmo ocorre falta de recursos na aquisição de equipamentos para a montagem de um laboratório de ensino. Mesmo um laboratório físico com diversos experimentos que possibilitam a interação sendo o ideal, muitas vezes este recurso só fica disponível para aquela instituição onde o mesmo está localizado ou aos arredores. A distância geográfica e a disponibilidade de tempo para visitas impede que se tenha acesso ao recurso, desta forma, muitos laboratórios permanecem fechados e os experimentos sem uso. Além disso, um laboratório físico necessita de um profissional qualificado para orientar os alunos no manuseio dos experimentos. Um grupo muito grande pode ser difícil de supervisionar e dependendo da experiência realizada podem acontecer acidentes, colocando em risco a segurança.

Um meio alternativo para o aprendizado prático tem sido os laboratórios virtuais que dispõem de simuladores, porém simulações possuem dados pré-programados trazendo resultados que nem sempre se aproximam da realidade. É de grande importância no aprendizado prático a interação com os aspectos reais (SILVA *et al.*, 2013).

“Os termos laboratório virtual e remoto podem gerar alguma confusão entre si, mas basicamente, laboratório virtual trata-se de um software de computador com capacidade de realizar (ou de simular) leituras de dados em instrumentos. Por outro lado, um laboratório remoto é constituído por uma estrutura de hardware e software para aquisição de dados em instrumentos reais, com experimentos também reais, sendo a manipulação do instrumento realizada necessariamente através de uma rede de computadores” (NEDIC *et al.*, 2003<sup>1</sup> apud OGIBOSKI 2007).

Vendo a disponibilização de experimentos reais para que alunos ou interessados tenham acesso e possam interagir, analisar, realizar suas experiências e aplicar na prática seus conhecimentos obtidos em sala de aula, alguns laboratórios têm disponibilizados seus experimentos para acesso remoto através da internet (SILVA *et al.*, 2013).

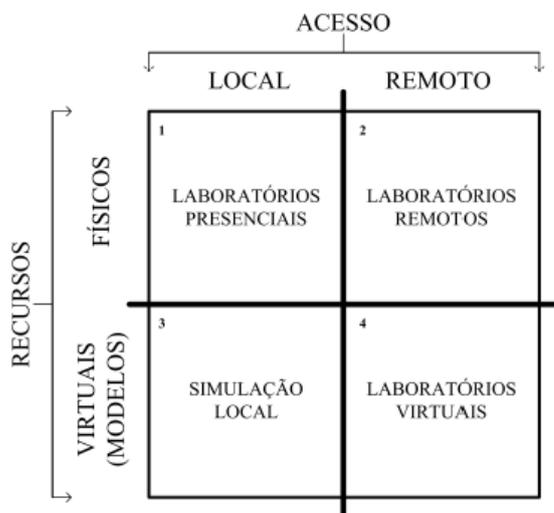
Experimentos Remotos são representações de dispositivos reais conectados por circuitos atuadores, sua interação é dada através da internet. Portanto lidar com

---

<sup>1</sup>NEDIC, Z., J. MACHOTKD, *et al.* Remote Laboratories versus Virtual and Real Laboratories.33rd ASEE/IEEE Frontiers in Education Conference. Boulder, Colorado, USA: IEEE, 2003. T3E p.

experimentos remotos é uma experiência real, pois os mesmos possuem elementos físicos que interagem por comandos virtuais. A interação com os experimentos é de forma direta e os resultados obtidos são imediatos online (SILVA, 2012). A Figura 17 mostra a relação entre os recursos e acessos dos diferentes meios de aprendizado prático.

Figura 17 – Relação entre recursos e acesso



Fonte: Sousa (2008)

A experimentação remota tem sido considerada um elemento de grande importância na área das TICs (Tecnologias da Informação e Comunicação) e da educação. Pois, permite por meio da automatização de experimentos práticos de laboratórios didáticos, o acesso remoto a estudantes que não têm acesso físico a tais experimentos, em qualquer hora e em qualquer lugar.

O primeiro passo para entender uma experimentação remota, é diferenciá-la de simulações, que não trazem resultados reais, visto que não interagem diretamente com o hardware e não podem garantir que os resultados obtidos estejam realmente corretos (ZANOTTO, 2006).

De um modo geral, um laboratório remoto propicia o acesso a experimentos do laboratório por meio de um computador conectado à Internet como se o usuário estivesse no próprio local. Este recurso tem sido aplicado em várias instituições de Ensino Superior. Outra possibilidade é que as universidades podem compartilhar seus laboratórios com outras universidades, aumentando assim flexibilidade entre os cursos

ou disciplinas oferecidas além de minimizar os custos (SILVA, 2007). A aplicação das aulas práticas tem sido realizada através de laboratórios de experimentação remota, desta forma os estudantes podem observar fenômenos que em grande parte são difíceis de serem entendidos por material escrito (SILVA *et al.*, 2013). A figura 18 ilustra os componentes fundamentais de um laboratório remoto, para que possa ocorrer uma experimentação remota.

Figura 18 – Esquema de um laboratório de Experimentação Remota



Fonte: Cardoso e Takahashi (2011)

O usuário através do seu computador com acesso à Internet pode acessar o servidor Web, buscar por um experimento e executá-lo. O servidor Web permite o que o usuário tenha acesso ao laboratório, conseguinte, ao controle dos dispositivos e a obtenção dos resultados do experimento. A interface programável possui basicamente duas funções: interpretar os dados obtidos dos experimentos para que o servidor Web possa repassar para o usuário, e interpretar o comando do usuário para que ele possa ser executado no aparato experimental. Muitas vezes, são incluídas câmeras para a visualização do experimento (CARDOSO; TAKAHASHI, 2011).

O laboratório de experimentação remota permite a interação com processos reais possibilitando a quem dele faz uso uma reflexão e conclusão sobre problemas do mundo real. Segundo Silva (2013), “estes laboratórios são aqueles em que os elementos são reais, seu acesso é virtual e as suas experiências reais”. Na prática trata-se de um laboratório real, porém que pode ser acessado por qualquer pessoa que possua um computador ou dispositivo com acesso à internet.

Segundo Cardoso e Takahashi (2011) esse tipo de laboratório vem sendo utilizado por empresas, no treinamento de pessoal especializado e no ensino, em particular, nas disciplinas que usam práticas experimentais. Cardoso e Takahashi

(2011) destacam ainda que “o uso de um laboratório remoto para o ensino pode suprir muitas dificuldades e carências relacionadas ao uso da experimentação remota nas escolas ou universidades”. Além disso, os laboratórios remotos podem servir de suporte para cursos de Educação a Distância (EaD), tornando os mesmos mais interativos e dinâmicos.

De acordo com Silva (2006, pág. 135), a implantação e uso de um laboratório de experimentação remoto em instituições de ensino tem as seguintes vantagens:

- Maior utilização dos equipamentos do laboratório. Pelo fato de os equipamentos estarem disponíveis 24 horas por dia, 365 dias ao ano. Não havendo restrição de tempo nem de espaço.
- Organização de laboratórios. Os laboratórios podem manter-se fechados, contanto, basta que estejam operacionais.
- Organização do trabalho dos alunos. Com os laboratórios remotos os alunos e professores podem organizar melhor seu tempo, de maneira similar aos horários de aulas.
- Aprendizagem autônoma. Os laboratórios remotos estimulam o trabalho autônomo, que é fundamental no modelo atual de educação superior.
- Abertura a sociedade. Os laboratórios remotos podem ser colocados à disposição da sociedade.
- Cursos não presenciais. Possibilitam a organizar cursos totalmente não presenciais, evitando muitos dos problemas atuais.
- Inserção dos usuários em um contexto real.

Silva (2006) enfatiza ainda que os laboratórios de experimentação remota buscam resolver os problemas de acesso aos laboratórios clássicos, com o objetivo de:

- Incrementar as atividades práticas em um curso (de forma que os alunos possam acessar a eles em qualquer horário, não somente quando esteja aberto o centro para temas docentes),
- Reduzir os custos de gestão e manutenção dos laboratórios (ao aumentar o uso em qualquer horário aos mesmos com um pessoal menor),

- Permitir o uso dos mesmos desde qualquer ponto geográfico de forma que se reduzam ou minimizem os custos de deslocamento, assim como a qualquer hora, permitindo desta forma resolver o problema dos fusos horários com outras zonas geográficas, e,
- Integrar em um mesmo ambiente as aplicações docentes das práticas, experimentação e trabalho no laboratório, com as atividades propriamente docentes mediante a integração de materiais, simulações e acesso a equipamentos e dispositivos.

No entanto, construir um sistema de experimentação remota não é considerado uma tarefa trivial, pois requer que existam condições físicas, monetárias e até administrativas para que seja implantado. Além disso, trata-se de um sistema crítico, em que o hardware é sujeito a fatores externos, como, as condições meteorológicas ou a perturbações da rede elétrica (TEIXEIRA, 2010).

Segundo Teixeira (2010):

“O fato de existir uma interface virtual com a qual o usuário interage, faz com que haja uma grande preocupação na eliminação das barreiras físicas previamente mencionadas. Isto é, há a necessidade de reproduzir virtualmente o laboratório físico através de fotografias, vídeos e outros elementos que dêem a percepção de um laboratório real, de forma a transmitir ao usuário um sentimento de estar presente e mais familiarizado com o sistema”. Não são todas, mas muitas áreas necessitam ter o laboratório virtual dessa forma, como, na medicina, onde os profissionais têm contato direto com as pessoas e necessitam de um ambiente que simule o que existe na realidade (HANSON et al., 2009<sup>2</sup> apud TEIXEIRA, 2010).

As principais funcionalidades ou requisitos que um laboratório remoto deve ter são (LEWIS et al., 2009<sup>3</sup> apud TEIXEIRA 2010):

- Disponibilidade: O sistema deve estar sempre disponível para responder aos pedidos dos usuários, mesmo quando existirem situações de acessos

---

<sup>2</sup>HANSON, Bem. “ReLOAD: Real laboratories Operated At Distance”. University of Leeds, 2009.

<sup>3</sup> LEWIS, N. et al. “A Distance Measurement Platform Dedicated to Electrical Engineering”. Bordeaux University, 2009.

simultâneos. Normalmente sugere-se a implementação de uma fila de espera FIFO (*First In First Out* em inglês, que significa o primeiro pedido a chegar é o que é tratado em primeiro lugar e assim, conseqüentemente) para gerenciar corretamente os pedidos de acesso;

- Segurança: Com foco em três pontos fortes: a validação dos dados introduzidos pelo usuário; o protocolo de autenticação do usuário; e o controle de acesso indevido aos dados;
- Flexível: Capacidade de se adaptar a novos ambientes, como, mudar alógica da experiência, suportar diversas ligações em simultâneo com cenários diferentes ou capacidade de mudar de laboratório (caso existam vários);
- Portabilidade: Capacidade de acessar o(s) experimento(s) do laboratório remoto de qualquer sistema operacional ou *Web Browser*;
- Custo de manutenção reduzido: Se possível, devem-se usar linguagens *open source* na implementação, tendo em vista uma manutenção futura ou evitar a compra de licenças.

Segundo Johnston e Agarwal (1995)<sup>4</sup> citado por Silva (2007): “um laboratório de experimentação remota deve cumprir os seguintes requisitos:

- Controle remoto e monitoramento dos experimentos.
- Comunicações multimídia entre os usuários.
- Um caderno de notas digital com todas as facilidades para introdução de dados, arquivos, figuras, buscas, etc.
- Gestão dos recursos, para decidir adequadamente que usuário ou usuários podem acessar a cada um dos experimentos disponíveis.
- Segurança tanto no aspecto de permitir e negar acesso como nos recursos para gerir possíveis falhas do sistema.
- Diversos tipos de comunicação: Voz, imagem, dados, resultado de experimentos, estado dos experimentos.
- Largura de banda: Adequada para permitir as distintas comunicações de dados científicos como de imagens ou vídeo”.

---

<sup>4</sup> JOHNSTON, W.; AGARWAL, D.. The Virtual Laboratory: Using Networks to enable Widely Distributed Collaboratory Science. Ernest Orlando Lawrence Berkeley National Laboratory, University of California, 1995.

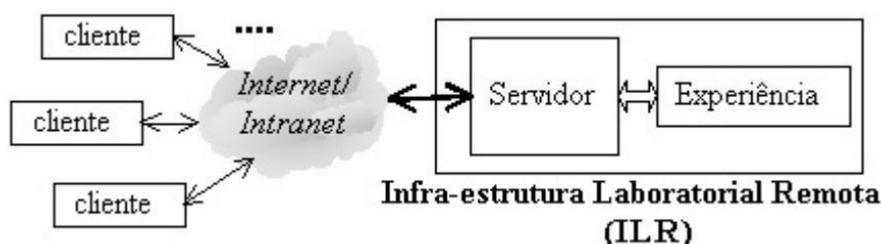
## 2.1 Infraestruturas de Laboratórios Remotos

O campo de desenvolvimento dos laboratórios remotos tem sido muito ativo na atualidade permitindo a sua implementação com diversas técnicas. Ao escolher as técnicas, é interessante primeiramente conhecer três cenários de desenvolvimento de laboratórios remotos descritos por Paladini (2008):

- A maioria dos laboratórios remotos em uso foram desenvolvidos numa instituição de ensino superior, isto normalmente lhes dá uma forte conotação de componentes de hardware deixando em segundo plano o design e a gestão dos serviços, priorizando os serviços de hardware. Este perfil de laboratório tem como premissa básica que os alunos possam acessar os experimentos no laboratório desprezando outros aspectos.
- Simetricamente ao anterior está o projetista especializado em software e em comunicações que vê o laboratório remoto como um típico problema de comunicação, níveis, objetivos, disponibilidade etc. Colocando em segundo plano o objetivo principal: tornar acessível o laboratório via Internet.
- Em outros casos os projetistas levam o laboratório remoto para seu domínio de conhecimento, ou seja, não se parte de um estudo das possibilidades tecnológicas, e sim simplesmente do uso de uma tecnologia conhecida.

A criação de uma infraestrutura que possibilita realizar experimentações de laboratórios remotamente via internet necessita de uma arquitetura que suporte as funcionalidades para realizar a interação com os componentes envolvidos. Com o propósito de obter o maior número de usuários adota-se para implementação da arquitetura o modelo cliente-servidor. Apesar de ser um modelo mais complexo de implementar, apresenta vantagens referentes a flexibilidade de acesso, permitindo a conexão de vários usuários ao servidor sem a necessidade de realizar modificações na infraestrutura do laboratório remoto. A Figura 19 ilustra a arquitetura cliente-servidor utilizada no acesso à infraestrutura de um laboratório remoto.

Figura 19 – Arquitetura cliente-servidor utilizada em uma infraestrutura de laboratório remoto



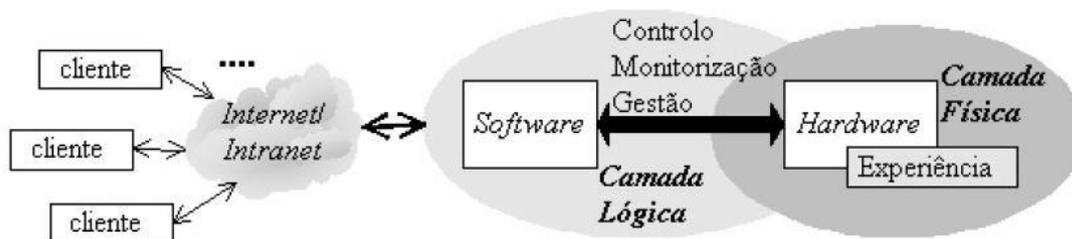
Fonte: Costa (2003)

O usuário tem a possibilidade de enviar comandos para o controle dos experimentos remotos além de dados a fim de configurar os mesmos. O servidor retorna para o usuário o resultado das experiências realizadas, que poderá ser por gráficos, tabelas, imagens, relatórios, entre outros (GILLET et al, 1999).

O servidor faz a ligação entre o experimento remoto e a rede informática responsável por manter o canal de comunicação para o usuário, além disso, deve fornecer o suporte para a realização das experiências, isso inclui interfaces, que permitem o controle e observação dos parâmetros envolvidos (COSTA, 2003).

Para implementar de fato uma infraestrutura de laboratório remoto é necessária uma camada física de hardware, que se refere a equipamentos, servidor, experimentos, e uma camada lógica de software responsável por controlar, monitorar e gerenciar os recursos da infraestrutura. Na Figura 20 é possível observar o esboço entre a camada física e lógica para a criação de uma infraestrutura de laboratório remoto.

Figura 20 – Camada física e camada lógica utilizadas no desenvolvimento de uma infraestrutura de laboratório remoto



Fonte: Costa (2003)

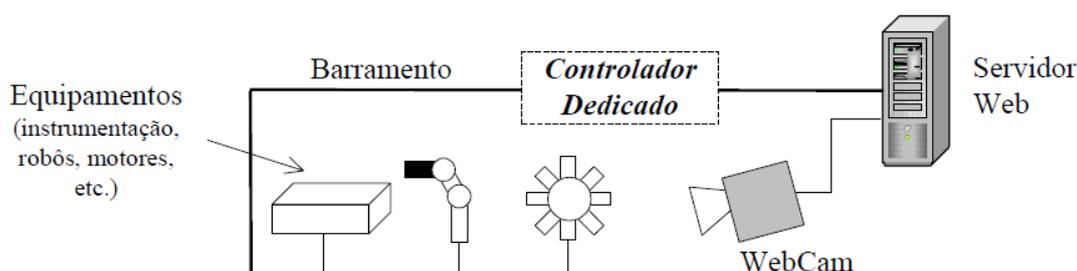
Nos subtópicos seguintes faz-se uma abordagem geral sobre a infraestrutura da camada física e lógica de um laboratório remoto, e dentro de cada camada, um estudo realizado sobre as possíveis tecnologias a serem utilizadas na construção do experimento remoto do sistema alternativo híbrido de energia.

### 2.1.1 Camada Física (Hardware)

A camada física de um laboratório remoto deve ser constituída por equipamentos capazes de controlar localmente as experiências, mediante a recepção de um conjunto de comandos dos usuários remotos (COSTA, 2003). Alguns desses equipamentos deverão ainda ser responsáveis pela disponibilização dos resultados das ações realizadas pelos usuários sobre a experiência, ou seja, conceder um retorno para os usuários. Este retorno, para melhor entendimento e aproveitamento da experimentação remota, deve conter resultados visuais. Isto pode ser feito através da utilização de webcams, por exemplo.

Além de um servidor Web, para que os requisitos apontados sejam cumpridos, é necessário que seja escolhido de forma conveniente todo o hardware desta camada. “Todos os equipamentos usados para controlar as experiências deverão ter um processamento dedicado, de modo que possam realizar as suas tarefas autonomamente, sem ser necessário qualquer tipo de processamento no servidor ou nos computadores dos usuários remotos” (COSTA, 2003). Desta forma, conforme representa a Figura 21, todos os equipamentos necessitam estar interligados em rede, através de um barramento com ou sem controle dedicado, para que os dados possam percorrer, neste caso, os dados são os comandos.

Figura 21 – Interligação de equipamentos num laboratório remoto



Fonte: Costa (2003)

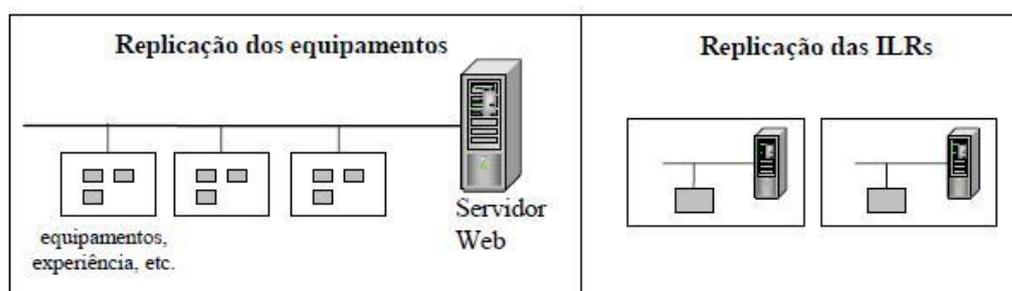
A conexão dos equipamentos com a rede é realizada pelo servidor Web, que permite o acesso remoto aos diferentes usuários. Este tipo de arquitetura flexibiliza a inclusão e retirada de componentes (COSTA, 2003). Segundo Alves (2000), nesse tipo de arquitetura são considerados dois tipos de barramentos:

- Barramento de controle. Utilizado para interligar controles e atuadores, através de conexões físicas simples e específicas para a troca de pequenas quantidades de dados a uma velocidade elevada (COSTA, 2003).
- Barramento de instrumentação. Tem a intenção de ligar os vários componentes envolvidos, controlados por um computador, permite a troca de dados complexos com velocidade (COSTA, 2003).

O controle de forma síncrona tem algumas implicações na disponibilidade de acesso aos experimentos remotos, vendo que é necessária a marcação prévia. A solução encontrada para resolver essa limitação é replicar os experimentos e seus respectivos equipamentos, conectando-os a um servidor, ou fazer a replicação da estrutura conforme é ilustrado na Figura 22.

Essa alternativa possibilita o aumento da confiabilidade, garantindo que falhas futuras ocorram na infraestrutura. Porém a aplicação desta solução tornar o custo elevado, devido aos equipamentos envolvidos, tornando o projeto inviável. Um outro meio é utilizar a forma assíncrona, onde o processamento remoto é disponibilizado conforme a requisição de usuários, através de uma fila de pedidos.

Figura 22 – Replicação dos equipamentos envolvidos e suas experiências ou de toda a infraestrutura laboratorial remota (ILR)

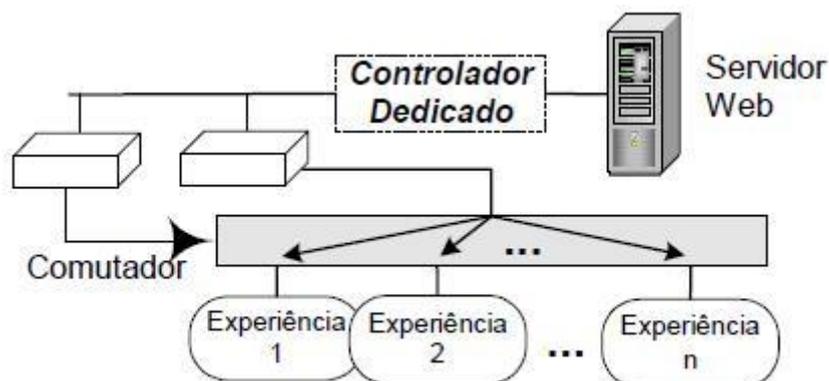


Fonte: Costa (2003)

É desejável que uma infraestrutura laboratorial remota suporte e seja adaptável a diversas experiências (COSTA, 2003). Para que se obtenha essa robustez é necessário a implementação de um sistema que permita alternar automaticamente entre os diversos experimentos, sem que haja a intervenção de um técnico. Este tipo de sistema aplica-se através de uma comutação automática, é utilizado um controlador que permite alterar as

conexões dos equipamentos aos experimentos. A Figura 23 exibe uma representação para o melhor entendimento de como funciona este sistema.

Figura 23 – Sistema de comutação que permite a partilha de vários experimentos utilizando uma só infraestrutura



Fonte: Costa (2003)

Caso não haja o interesse em ter vários experimentos disponíveis simultaneamente, esta é uma solução muito interessante, pois as ligações físicas entre os equipamentos e os experimentos são feitas de forma remota e automática, evitando a intervenção de um técnico no local para fazer a mudança (COSTA, 2003). A redução dos custos é evidente uma vez que é possível utilizar os mesmos equipamentos em diversas experiências. A escolha de um sistema deverá ser analisada a fim de se obter um bom custo-benefício. E a implantação deverá satisfazer as necessidades do projeto e dos beneficiados com ele (usuários e instituição).

### 2.1.1.1 Câmera Web

Um componente de grande importância na experimentação remota é a câmera Web. Esta é responsável por capturar as imagens do experimento para que possam ser transmitidas em tempo online ao usuário. Existem diversos tipos de câmeras web e interfaces por onde os dados, no caso as imagens capturadas, irão trafegar. As câmeras web podem utilizar como meio de comunicação a rede intranet/internet, sendo assim chamadas de câmeras IP. As câmeras IP têm suas configurações definidas em um computador e após a realização desta configuração a câmera deverá estar conectada à rede, tornando possível o acesso as imagens transmitidas por ela através de um

navegador Web. A transferência de dados entre a câmera IP e o ponto de rede ou roteador pode ser feita através da rede Wireless ou pela rede cabeada utilizando cabo UTP, isso dependerá da tecnologia utilizada na câmera que pode variar de acordo com o modelo e fabricante.

Existem modelos de câmeras que utilizam para a transferência de dados outros tipos de conexões, a mais comum entre elas é a USB. As câmeras USB estão disponíveis no mercado em maior número, possuem desde modelos simples, como as câmeras Web utilizadas nos computadores convencionais até modelos profissionais com alta resolução e alta taxa de transmissão de dados. Uma desvantagem deste tipo de câmera é que esta precisa estar conectada fisicamente a um computador ou servidor para que o mesmo possa receber as imagens e redistribuí-las na rede.

### **2.1.1.2 Plataformas de Prototipagem Eletrônica**

Atualmente são notáveis os elos formados entre a computação e a eletrônica, eles estão presentes nos computadores, celulares, eletrodomésticos, automóveis, entre outros. E ao ver o produto final constituído por essa junção tem-se a impressão de que a comunicação de sistemas e a lógica de programação são tarefas complexas para definir rotinas e funções para um hardware (COLATTO, 2011). Porém existem ferramentas corretas, de fácil acesso e utilização com o propósito de minimizar a complexidade entre os sistemas de hardware e software no que diz respeito ao desenvolvimento. Desta forma é possível que qualquer pessoa que tenha conhecimentos básicos em computação e eletrônica possa desenvolver sistemas que envolvam automação, hardware e software.

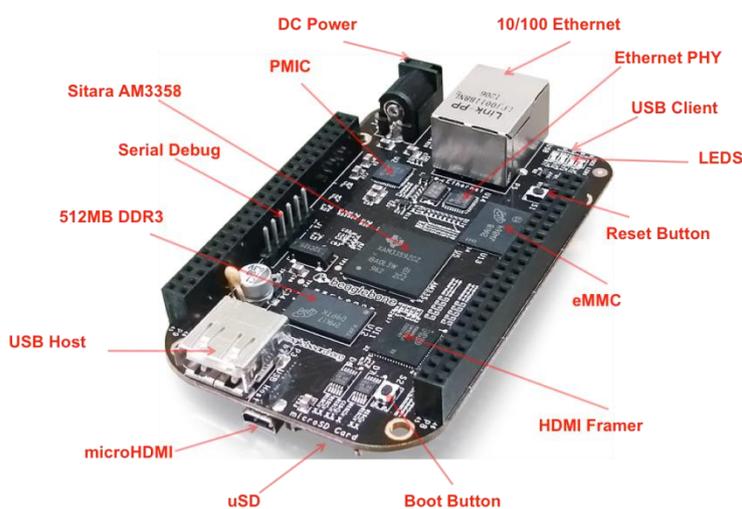
As plataformas de prototipagem eletrônica são ótimas ferramentas para quem tem pouca noção de computação e eletrônica, mas quer desenvolver ou automatizar um sistema. São muito utilizadas nas escolas e universidades onde há o ensino de eletrônica e programação, mas também podem servir de base para projetos de pequeno e médio porte. Consiste em uma placa única com entradas e saídas, um microcontrolador ou microprocessador, uma interface de programação e demais componentes eletrônicos. Basicamente as plataformas de prototipagem eletrônica são um conjunto de ferramentas que possibilitam o desenvolvimento de dispositivos eletrônicos (CAMARGO, 2013). Existem diversas placas de prototipagem de diferentes fabricantes. Nos próximos

subtópicos serão apresentadas algumas placas de prototipagem eletrônica disponíveis no mercado.

### 2.1.1.2.1 Beaglebone Black

O projeto da Beaglebone Black é dirigido pela BeagleBoard Foundation, uma corporação sem fins lucrativos que tem como objetivo promover a utilização de hardware e software open source. Atualmente a placa Beaglebone Black é produzida pela CircuitCo e utiliza um processador da Texas Instruments (SUEIRO, 2014). A Figura 24 ilustra a placa Beaglebone Black e seus componentes característicos.

Figura 24 – Características da Beaglebone Black



Fonte: Janeiro (2015).

O valor de aquisição desta placa no Brasil está em torno de 250 reais. A mesma pode operar como um computador de baixo custo, porém existem algumas limitações como uma saída HDMI na qual podem ser conectados monitores e TV's, porém não é possível utilizar aplicações em Full HD, pois o processador não acompanha no processamento dos frames. Uma grande vantagem é a flexibilidade, a placa possui diversas entradas e saídas além de suporte a placas de expansão, o usuário pode empilhar até quatro placas de expansão. Segue abaixo características da Beaglebone Black.

Características físicas (hardware):

- Processador AM3358BZCZ100 ARM Cortex-A8 32-Bits 1GHz;
- On-board HDMI para ligar diretamente em TV's e monitores;
- Memória RAM de 512MB SDRAM 400 MHz DDR3, 4 GB de armazenamento interno;
- Slot para cartão micro SD;
- Suporte para placas de plug-in Cape existentes (placas de expansão);
- Regulador TPS65217C CIMP e uma LDO adicional;
- Apoio Debug: opcional onboard de 20 pinos CTI JTAG;
- HS USB 2.0 Cliente e portas Host;
- Porta serial: acesso UART0 via de 6 pinos cabeçalho 3.3V TTL;
- 10/100 RJ45 Ethernet.

Características lógicas (software):

Sistemas operacionais suportados:

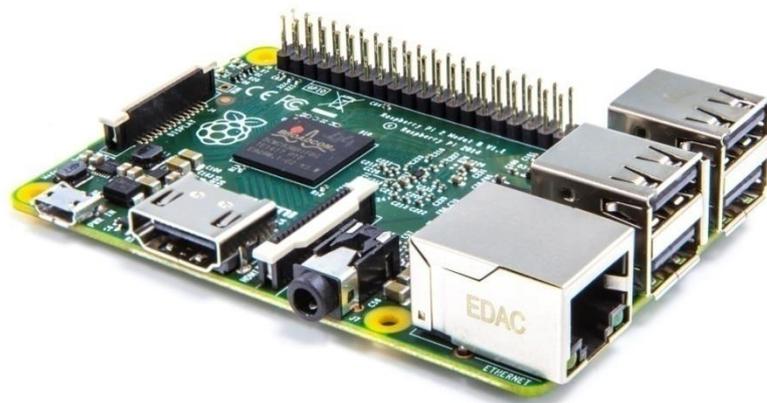
- Debian (oficial);
- Angstrom (oficial – de fábrica);
- Ubuntu;
- Android;
- Nintendo;
- Gentoo;
- ArchLinux;
- LinuxCNC;
- Minix;
- XNU;
- Asterisk;
- TI EZSDK;
- Bleaglenmt.

#### **2.1.1.2.2 RaspberryPi**

Em 2006 surgiram os primeiros conceitos para o RaspberryPi. O engenheiro Eben Upton e sua equipe tinham como base o microcontrolador Atmel ATmega. No ano de 2009 os membros criaram a RaspberryPi Foundation, porém somente após três anos

o RaspberryPi passou a ser comercializado oficialmente (HEIN, 2013). A Figura 25 ilustra a placa RaspberryPi 2, modelo comercializado atualmente.

Figura 25 – RaspberryPi 2



Fonte: raspberrypi.org

Apesar do RaspberryPi ser desenvolvido com o foco em projetos de eletrônica, robótica, projetos educacionais e entre outros, o mesmo pode operar como um computador portátil, pois suas características físicas permitem que o mesmo possa ser utilizado como um desktop. Uma das principais vantagens é o processador QuadCore com um clock de 900MHZ permitindo um bom desempenho com baixo aquecimento. Outra grande vantagem é o acelerador gráfico GPU embutido que faz com que o RaspberryPi 2 processe aplicações de vídeo em Full HD. Uma desvantagem é a de não possuir memória de armazenamento interno, porém conta com um slot de expansão para cartão micro SD. O valor de aquisição no Brasil está em torno de 250 reais. Segue abaixo características do RaspberryPi 2.

Características físicas (hardware):

- Processador de Broadcom BCM2836 900MHz ARM Cortex-A7 quad-core com GPU dual core de VideoCore IV GPU fornece Open GL ES 2.0, OpenVG acelerado por hardware, e 1080p 30 H.264 decodificação de alto perfil;
- GPU é capaz de 1Gpixel/s, 1.5Gtexel / s ou 24GFLOPs com textura filtragem e infra-estrutura DMA;
- 1GB RAM;

- Saída de vídeo HD 1080p;
- Saída de vídeo composto (PAL/NTSC);
- Saída de áudio estéreo;
- Soquete de BaseT RJ45 Ethernet 10/100;
- HDMI 1.3 & 1.4 tomada de vídeo/áudio;
- 3,5 mm vídeo fora tomada de jack áudio/composto de 4 pólos;
- 4 x soquetes USB 2.0;
- Conector de CSI MPI-2 15-entrada para a câmara de vídeo;
- Conector de Interface Serial Display 15 vias;
- Soquete do cartão MicroSD;
- 40-pinos GPIO (compatível com RaspberryPi 1 26-pinos);
- Fonte de alimentação: + 5V @ 2A via micro USB soquete;
- Dimensões: 86 x 56 x 20 mm.

Características lógicas (software):

- Compatível com distribuições Linux;
- Microsoft Windows 10.

### **2.1.1.2.3 Intel Galileo**

Criada pela Intel e lançada em 2013 a placa de prototipagem eletrônica Galileo foi projetada para dar suporte a projetos de estudantes, educadores e entusiastas da eletrônica. A placa Galileo possui pinagem padrão Arduino, porém a arquitetura do hardware é desenvolvida pela Intel. O processador é o Intel Quark SoC X1000 de 400MHZ Single Core. A Figura 26 ilustra a placa de prototipagem eletrônica Intel GalileoGen2 esta que é comercializada atualmente pelo fabricante.

O Intel GalileoGen2 possui como principal vantagem a compatibilidade de hardware e software com o Arduino. A pinagem é padrão Arduino UNO R3 que permite que o Galileo suporte placas de expansão utilizadas no Arduino. A interface de programação é compatível com os sistemas operacionais Linux, Windows e Mac OS, pois o simulador é o mesmo do Arduino UNO R3. A distribuição Linux que vem instalada por padrão na placa é o Yocto 1.4 Poky. O Galileo possui slot para cartão de memória até 32GB e conta com 8MB de armazenamento interno, uma porta Ethernet de

100Mb e um slot mini PCI Express. A principal desvantagem do Galileo comparado com as demais plataformas de prototipagem eletrônica é o valor de aquisição. No Brasil o modelo atual GalileoGen2 custa em torno de 450 reais. Segue abaixo características da plataforma de prototipagem eletrônica Intel GalileoGen2.

Figura 26 – Intel GalileoGen2



Fonte: Lehrbaum (2014).

#### Características físicas (hardware):

- Processador: Intel Quark SoC X1000 400MHz, 32-bit, cache: 16 KBytes L1;
- Memória RAM DDR3 de 256 MB;
- Memória de armazenamento Flash de 8 MB e EEPROM padrão de 8 KB on-board, além de suporte para cartão MicroSD com até 32 GB;
- Slot mini-PCI Express;
- Porta Ethernet de 100 Mb;
- Conector host USB 2.0 ;
- Conector client USB 2.0;
- JTAG padrão 10-pinos para debugging;
- Suporta I2C, SPI, UART e ICSP.

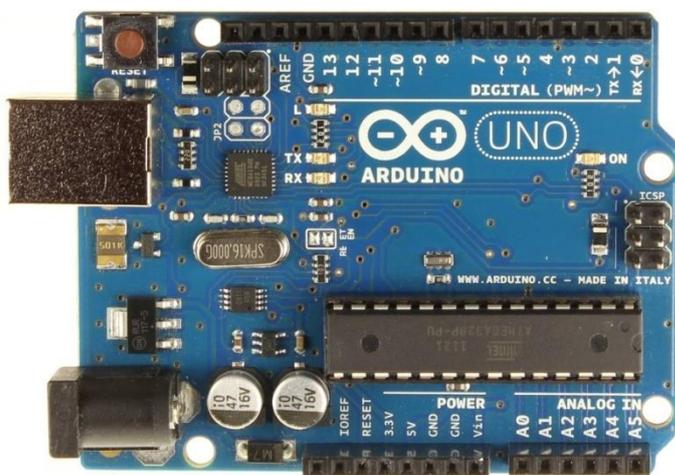
#### Características lógicas (software):

- Programável via ambiente de desenvolvimento integrado Arduino que é compatível com os sistemas operacionais Microsoft Windows, Mac OS e Linux;
- Suporte para a versão Linux Yocto 1.4 Poky.

### 2.1.1.2.4 Arduino

O projeto do Arduino teve início em 2005 na cidade de Ivrea na Itália com o propósito de facilitar a prática de estudantes em trabalharem com a tecnologia (EVANS *et al.*, 2013). O Arduino é uma plataforma de prototipagem eletrônica de código aberto que possui um software e hardware flexível, sendo bastante utilizada em projetos de robótica e eletrônica embarcada. Todo o projeto eletrônico e a plataforma de desenvolvimento de software de controle possuem acesso público e gratuito. A Figura 27 ilustra a plataforma de prototipagem eletrônica Arduino UNO R3.

Figura 27 – Arduino UNO R3



Fonte: <http://www.forefront.io> (2013)

O componente principal da placa Arduino UNO R3 é o microcontrolador ATMEL ATmega328P de 16 MHz este conta com 32 KB de memória Flash, porém 512 Bytes são utilizados para a inicialização, possui 2 KB de memória RAM e 1KB de EPROM. A placa pode ser alimentada por uma fonte externa de 7V até 12V (estes valores de tensão ideal, porém seus limites são de 6V até 20V) ou através da conexão USB. O Arduino UNO R3 possui 14 pinos digitais e 6 pinos analógicos, permitindo que o usuário possa utilizar placas de expansão potencializando a capacidade de hardware do mesmo.

Em relação ao software o Arduino possui uma camada simples implementada na placa que é um *bootloader*, e uma interface de programação que é instalada no computador (GOMES; TAVARES, 2013). A IDE de desenvolvimento do Arduino utiliza uma linguagem de programação que é semelhante a linguagem C/C++.A

vantagem de programar para Arduino é que o mesmo conta com uma grande quantidade de bibliotecas que facilitam a implementação do código, além de vários materiais e projetos que servem de base para estudos disponíveis na internet. Segue abaixo as características do mesmo.

#### Características físicas (hardware):

- Microcontrolador: ATmega328P;
- Tensão operacional: 5V;
- Tensão de entrada (recomendado): 7-12V;
- Tensão de entrada (limite): 6-20V;
- Pinos digitais I/O: 14 (dos quais 6 oferecem saída PWM);
- Pinos PWM Digital I/O: 6;
- Pinos analógicos de entrada: 6;
- Pino de Corrente DC I/O: 20 mA;
- Pino de Corrente DC 3.3V: 50 mA;
- Memória flash: 32 KB (ATmega328P), dos quais 0,5 KB utilizado pelo bootloader;
- SRAM: 2 KB (ATmega328P);
- EEPROM: 1 KB (ATmega328P);
- Velocidade do clock: 16 MHz;
- Comprimento: 68,6 mm;
- Largura: 53,4 mm;
- Peso: 25 g.

#### Características lógicas (software):

- IDE programável compatível com Windows, Linux e Mac OS utilizando a linguagem Arduino que semelhante a linguagem C/C++.

### **2.1.2 Camada Lógica (Software)**

A camada lógica é a responsável por efetuar todo o controle, monitoração e gestão de uma infraestrutura de laboratório remoto. As aplicações de software que são necessárias servem para conectar o dispositivo com o computador local, o computador local com o remoto, e o computador remoto com o usuário. Para isso, são usados

diferentes tipos de tecnologias, diferentes linguagens de programação, para a implementação da lógica da aplicação.

As tecnologias utilizadas para a implementação de um laboratório de experimentação remota, independentemente de sua finalidade, divergem de acordo com sua necessidade (SANTOS, 2009). Por exemplo, na etapa local, onde a experiência é desenvolvida e os dados são processados, podem ser utilizadas linguagens tradicionais como JAVA, PHP, como também podem ser utilizadas linguagens proprietárias, que é o caso dos softwares comerciais Matlab/Simulink e LabView, e o gratuito Scilab/Scicos, que são softwares que possuem este objetivo; na etapa de disponibilização da experiência para acesso através da Internet, a mesma pode ser feita através do uso de um servidor Web, existem distintos servidores Web, como os gratuitos Apache e Jetty, ou comerciais como o Microsoft IIS; e na etapa de desenvolvimento de uma interface que permita que o usuário possa interagir com a experiência podem ser utilizadas linguagens de programação, como o HTML, PHP e JSP, assim como ferramentas disponibilizadas por softwares de desenvolvimento, como o LabView (SANTOS, 2009).

Os sistemas de experimentação costumam ser bem aceitos pelos utilizadores por serem sistemas inovadores, entusiasmantes, motivantes, inspiradores e fáceis de usar (TEIXEIRA, 2010). No caso específico da informática, e para uma maior disponibilidade, convém que esse tipo de sistema seja acessível de qualquer local, preferencialmente através de um navegador Web, e o mais multiplataforma possível, para que usuários com sistemas operativos diferentes possam acessar nas mesmas condições (SOUSA et al., 2010<sup>5</sup> apud TEIXEIRA, 2010).

O controle e a monitorização da camada lógica poderão ser efetuados de forma síncrona ou assíncrona. Embora ambas as soluções permitam o controle remoto do equipamento real, a escolha por cada uma deverá ser de acordo com a aplicação pretendida. Costa (2003) caracteriza as duas soluções:

“Controlar sincronamente instrumentos remotos, permite aos utilizadores efetuar um controle em tempo real, muito semelhante ao que realizariam localmente num laboratório. No entanto, e embora a experimentação remota tenha como principal objetivo permitir aos utilizadores efetuar experiências com instrumentos reais a partir de qualquer lugar e em qualquer altura, o

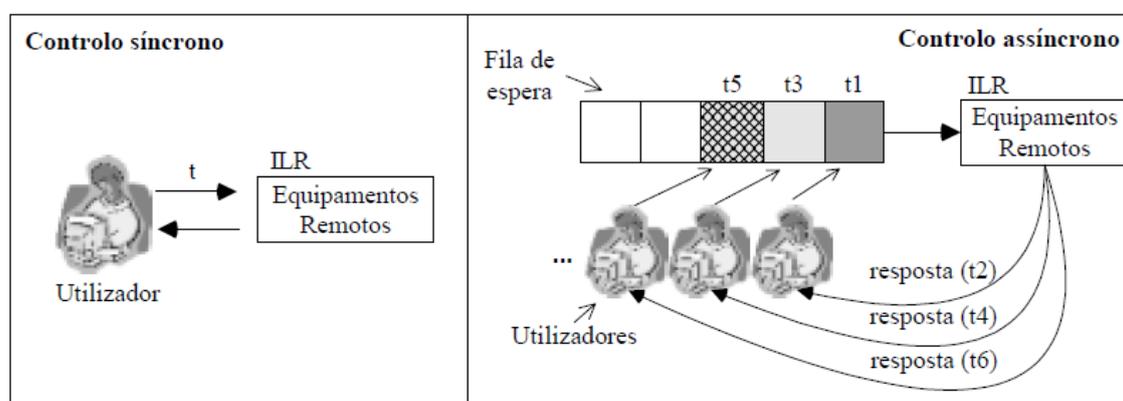
---

<sup>5</sup>Sousa, N.; Alves, A.R.; Gericota, M.G., 2010. “Na integrated reusable remote laboratory to complement electronics teaching”, IEEE Transactions on learning technologies, Vol. 3, Nº 3, July - September 2010, pp. 265-271.

controle síncrono põe em causa a permanente disponibilidade do laboratório, na medida em que o acesso dos utilizadores só é possível quando o laboratório está livre. Neste pressuposto, haverá a necessidade de gerir os acessos ao laboratório através da marcação prévia do horário pretendido. Por outro lado, a interação assíncrona com o equipamento remoto é gerida de forma mais eficiente, permitindo a submissão de dados para as experiências em qualquer instante, sem ter que efetuar reservas prévias. Nesta situação, as experiências solicitadas terão que ser geridas por filas de espera, para as quais os utilizadores remotos enviam as configurações ou dados necessários à sua realização. Estes dados poderão ficar em fila de espera se a ILR (Infraestrutura Laboratorial Remota) estiver ocupada, ou em caso contrário ser imediatamente transferidos, possibilitando a obtenção imediata dos resultados da experiência.”

A Figura 28 ilustra a forma de controle e monitorização síncrona e assíncrona, mostrando a diferença entre estas duas aproximações que permitem a realização de experiências remotas, salientando-se a sequência temporal das operações no controle assíncrono, possivelmente através dos tempos indicados.

Figura 28 – Controle síncrono e assíncrono de uma infraestrutura laboratorial remota



Fonte: Costa (2003)

### **3. TECNOLOGIAS PARA DESENVOLVIMENTO DE AMBIENTE WEB**

Para o desenvolvimento de páginas web é necessário o entendimento de alguns conceitos, bem como a realização de um estudo para conhecimento das linguagens de programação web, plataformas de desenvolvimento, frameworks etc. Enfim, tudo o que for fundamental para o desenvolvimento da mesma, levando em consideração que quanto maior o estudo aplicado, maior será conhecimento de recursos que poderão facilitar o desenvolvimento. Os subtópicos seguintes contêm um apanhado do estudo realizado sobre o referido argumento, para a possível construção da página web através da qual o experimento remoto será disponibilizado.

#### **3.1 Introdução à Web**

A internet é um conjunto de serviços, e a Web é um deles, sendo o mais popular. A web é então, um serviço que consiste em um imenso conjunto de páginas conectadas umas às outras por um sistema de links. O sistema no qual está construída a web chama-se hipertexto e é um emaranhado de páginas conectadas com links. A web não só se limita a apresentar textos e links, como também pode nos oferecer imagens, vídeos, áudio e todo o tipo de apresentações, chegando a ser o serviço mais completo em meios que tem a Internet. Por esta razão, para nos referirmos ao sistema que implementa a web (hipertexto) se cunhou um novo termo que é a hipermídia, fazendo referência a que a web permite conteúdos multimídia (ALVAREZ, 2004).

Quando se vê em um navegador ou cliente web, uma página web parece uma só entidade, mas não é assim, ela está composta por uma série de arquivos diferentes, como as imagens, os possíveis vídeos e o mais importante, o código de origem que diz aonde colocar cada texto, cada imagem ou cada vídeo e a forma que estes terão ao serem colocados na página. Não é um problema que as webs estejam compostas por tantos elementos, já que rapidamente percebe-se que sua organização é fácil e que não os perderá.

Como se pode ver de forma resumida, para publicar na Internet é necessário construir alguns documentos hipertextos ou hipermídia, com seus correspondentes arquivos de imagens ou vídeos e colocá-los em alguns computadores que se tornarão servidores de páginas web. Isto será explicado por partes nos próximos tópicos.

## 3.2 Recursos Necessários para Desenvolvimento de Páginas Web

Basicamente, os recursos necessários para a construção de páginas web são um editor de texto, no qual se programam as páginas web, um cliente da web, que são os navegadores (browsers), como o Google Chrome, Mozilla Firefox e Internet Explorer, e um Servidor Web. Para o desenvolvimento de algo simples, pode ser utilizado um editor de texto simples, como, o Notepad do Windows. Existem editores de texto como Gedit ([www.gnome.org](http://www.gnome.org)), Sublime (<http://www.sublimetext.com/>) e Notepad++ (<http://notepad-plus-plus.org>), que possuem realce de sintaxe e outras ferramentas para facilitar o desenvolvimento de páginas. Para a construção projetos mais sofisticados, é necessário um editor mais robusto, no caso uma IDE<sup>6</sup>, que oferece recursos como autocompletar e pré-visualização.

### 3.2.1 Servidor Web

Conforme ilustrado na Figura 29, o Servidor Web é responsável por aceitar requisições HTTP dos clientes, geralmente os navegadores, respondendo essas requisições com respostas HTTP, incluindo opcionalmente dados, que geralmente são páginas web, tais como documentos HTML com objetos embutidos (imagens, vídeos, gráficos etc.). Um Servidor Web pode ser um software que implementa o protocolo de comunicação HTTP, atuando no papel de servidor, ou um computador que executa um programa que provê a funcionalidade descrita anteriormente (MARCHIORO, 2003). Exemplos de servidores Web são o Apache (software livre), o mais popular e mais utilizado no mundo, o iPlanet, o ISS da Microsoft, dentre outros (TELEPAC 2003<sup>7</sup> apud MARCHIORO, 2003).

Segundo Marchioro (HESSEL 2001 apud MARCHIORO 2003), “um cliente HTTP (*browser* cliente Web) se comunica com um servidor HTTP (Servidor Web) requisitando arquivos. Normalmente estes arquivos encontram-se no

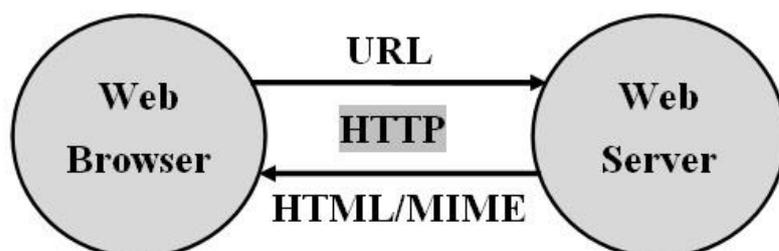
---

<sup>6</sup>IDE (Integrated Development Environment): No português, Ambiente de Desenvolvimento Integrado é uma aplicação de software que fornece recursos abrangentes para programadores de computador para desenvolvimento de software. Um IDE normalmente consiste de um editor de código-fonte, ferramentas para construção de automação e um depurador. A maioria dos IDEs modernos têm código de complementação inteligente (WIKIPÉDIA, 2015).

<sup>7</sup>TELEPAC (Org.). **Glossário**. 2003. Disponível em: <[https://www.ptempresas.pt/pme/internet/internet-fixa/netplusglossario/glossario\\_W.html](https://www.ptempresas.pt/pme/internet/internet-fixa/netplusglossario/glossario_W.html)>. Acesso em: 13 jul. 2015.

formato HTML (*Hypertext Markup Language*) e, ao receber o arquivo HTML, o cliente verifica cada linha do arquivo, solicitando ao servidor HTTP os arquivos necessários. Esse é o modelo básico e estático de apresentação de páginas na Internet, ou seja, o usuário não faz nenhuma interação, apenas as aplicações cliente e servidor é que são encarregadas do processo como um todo”.

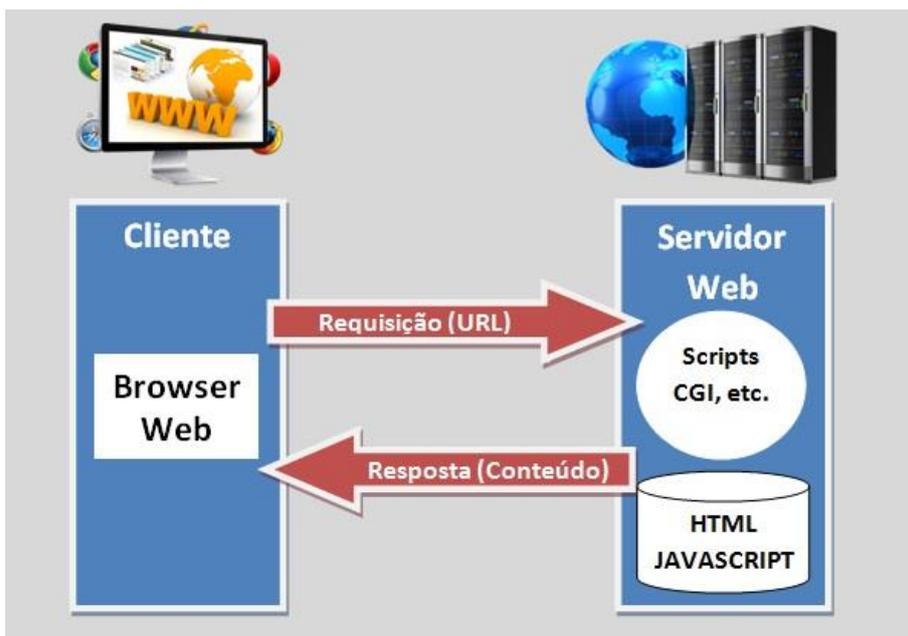
Figura 29 – Modelo Genérico Cliente-Servidor



Fonte: Marchioro (2003).

Além das páginas estáticas existe a possibilidade de interação e dinamismo executado sobre as páginas. Neste grupo encontram-se páginas baseadas em *script* como o Microsoft ASP (*Active Server Pages*) e o PHP (*Hypertext Preprocessor*). Através desses sistemas tornou-se possível um servidor Web receber uma informação do cliente, processá-la, e retornar uma página em HTML construída em tempo real no servidor Web. O modelo de páginas dinâmicas é ilustrado na Figura 30.

Figura 30 – Modelo de um Servidor WEB com páginas dinâmicas.



Fonte: Arquivo Pessoal.

### 3.2.2 IDEs para Desenvolvimento de Ambientes Web

O uso de um Ambiente de Desenvolvimento Integrado (do inglês *Integrated Development Enviroment* - IDE) adequado para o desenvolvimento de software é primordial para um desenvolvimento bem sucedido, principalmente quando o desenvolvimento é focado para Web, que geralmente apresenta uma maior complexidade que os sistemas *stand alone*<sup>8</sup> (CHAVES E SILVA, 2008).

Um ambiente de desenvolvimento integrado pode ser considerado como um conjunto de ferramentas utilizadas com a intenção de prestar suporte ao desenvolvimento de software. O principal objetivo das IDE's é agilizar este processo de desenvolvimento. A maioria das IDE's utiliza a técnica RAD (do inglês *Rapid Application Development*), que permite que os desenvolvedores tenham um aproveitamento maior, para que estes possam desenvolver códigos com maior agilidade e facilidade. O ambiente de desenvolvimento é considerado integrado quando envolve fundamentalmente os recursos de editor, compilador e depurador (CHAVES E SILVA, 2008).

---

<sup>8</sup>**Stand Alone:** São chamados *stand alone*, ou *stand-alone* (literalmente "ficam em pé por si só") os sistemas completamente autossuficientes: para seu funcionamento não necessitam de um software auxiliar, como um interpretador, sob o qual terão de ser executados (Wikipedia, 2016).

### 3.2.2.1 Eclipse

O Eclipse é um IDE de código aberto desenvolvido em Java, o projeto Eclipse teve início no ano de 2001 pela empresa norte-americana IBM. Atualmente é considerada a IDE Java mais popular e utilizada do mundo (FARIA et AL., 2010). A plataforma Eclipse pode suportar diversas linguagens, pois o software possibilita trabalhar com uma estrutura de *plug-ins*. Essa estrutura permite que o usuário possa adicionar funções, compiladores ou até mesmo outros programas para serem utilizados pela IDE de acordo com a linguagem que o mesmo necessitar em seu projeto.

O Eclipse é uma IDE bastante utilizada para o desenvolvimento da linguagem Java, mas também pode trabalhar com outras linguagens de programação graças a sua arquitetura baseada em *plug-ins*, que torna o software multilinguagem. Baixando os devidos *plug-ins* o usuário pode trabalhar com linguagens tais como C/C++, Java, PHP, Python e entre outras. Além disso, a plataforma fornece diversos pacotes de desenvolvimento, tais como Eclipse JDT que serve de base para *plug-ins* na linguagem Java, o Eclipse SDK que é o pacote de distribuição da IDE Java, O Eclipse WTP (*Web Tools Platform*) que é utilizado para o desenvolvimento de aplicações em linguagem Web e o seu próprio compilador Java de código aberto chamado de JDT (FARIA et al., 2010).

O Eclipse permite ao usuário trabalhar com a refatoração do código, desta forma o código pode ser reestruturado de forma organizada. Porém a IDE apresenta algumas desvantagens, como a deficiência de não possuir uma versão online, é um software que exige da máquina uma capacidade de processamento maior e a arquitetura de *plug-ins* pode dificultar no início a utilização do usuário, pois uma má instalação dos mesmos poderá interferir nas implementações (FARIA et al., 2010).

### 3.2.2.2 Aptana Studio

A Aptana Studio é uma IDE de código aberto utilizado para o desenvolvimento de aplicações web, baseado no software Eclipse que também é utilizado para desenvolvimento web, porém com foco no desenvolvimento de Java. A Aptana Studio é voltado para desenvolvimento com suporte a linguagens PHP, HTML, CSS, JavaScript, Python entre outras. A IDE pode ser distribuído tanto como software independente ou

como *plug-in* para o Eclipse. O software independente pode operar em plataformas de sistemas operacionais Windows, Linux e Mac OS X (ALVARES, 2007).

Segundo o site oficial do software Aptana Studio o mesmo possui recursos tais como:

- Indicação do nível de suporte para cada elemento nos principais navegadores Web;
- Suporta os principais protocolos, incluindo FTP, SFTP e FTPS;
- Capacidade de publicar automaticamente as aplicações desenvolvidas nos serviços de hospedagem;
- Possui *debugger* integrado, onde é possível definir os pontos de interrupção e inspecionar as variáveis;
- Permite o controle de execução do código através dos depuradores integrados que auxiliam a identificar a fonte de possíveis erros;
- Terminal de acesso rápido para a execução de comandos do sistema operacional;
- Capacidade de personalização do ambiente de desenvolvimento por meio de comandos e *scripts*;
- Assistente de Código: O desenvolvedor digita uma *tag* e a IDE oferece a possibilidade de auto completar essa *tag*, além de apresentar em qual navegador ela está disponível para ser renderizada corretamente.

A Aptana Studio pode parecer uma IDE menos robusta comparado a outras opções de software disponíveis que suportam diversos tipos de linguagens de programação. A Aptana Studio é voltada para o desenvolvimento com linguagens Web, desta forma projetos que não necessitam fazer uso de outras linguagens têm suas necessidades atendidas pelo software. Isso evita desperdício de recursos do programa e diminui a saturação da capacidade de processamento da máquina.

### 3.2.2.3 Visual Studio

O Visual Studio é um ambiente de desenvolvimento integrado desenvolvido pela Microsoft. O software possui suporte para as linguagens C#, Visual Basic, F#, C++, Python, Node.js, CSS e HTML/JavaScript, mas também pode trabalhar com outras linguagens como PHP, basta o usuário baixar o pacote que deseja utilizar em seu projeto

de desenvolvimento. A IDE é multiplataforma podendo operar com sistemas operacionais Windows, Linux, Mac OS X e plataformas de sistemas para dispositivos móveis como Windows Phone, Android e IOS.

Segundo o site oficial da Microsoft Visual Studio, a IDE possui recursos básicos de depuração, refatoração do código, correção de erros de sintaxe do código, gerenciador de código-fonte hospedado em qualquer provedor, possibilidade de exportar bibliotecas, entre outros.

O Visual Studio é um software bastante robusto e completo, oferece uma gama enorme de possibilidades de desenvolvimento aos usuários, porém somente as versões básicas estão disponíveis gratuitamente e as licenças podem ser utilizadas somente para uso particular. As versões completas e de uso corporativo possuem custo de aquisição.

A grande quantidade de funções e suporte para diversas linguagens pode ser uma desvantagem para usuários que desejam trabalhar com projetos específicos onde é utilizado somente um tipo de linguagem, isto acaba gerando um desperdício de recursos do programa, além de exigir que a máquina possua requisitos de hardware mais robustos para poder processar o mesmo.

### **3.2.2.3 NetBeans**

O NetBeans é um ambiente de desenvolvimento integrado gratuito e de código aberto. O IDE é bastante utilizado no desenvolvimento de aplicações Java, principalmente quando se trabalha com orientação a objetos. Este motivo se dá pelo fato de existir uma grande quantidade de bibliotecas e materiais em meio digital que auxiliam no desenvolvimento. Além do Java o NetBeans possui suporte para C/C++ e para as linguagens voltadas ao desenvolvimento Web, estas HTML, CSS, PHP, JSP e JavaScript. Atualmente são 6 distribuições que iniciam na versão básica até a versão completa.

Segundo o site oficial do NetBeans o mesmo possui recursos tais como os descritos abaixo:

- Modelos e exemplos de desenvolvimento, importante para iniciantes na compreensão da linguagem de programação.

- Indentação automática, recuo de linhas, associa palavras e colchetes e realça códigos-fonte sintática e semanticamente.
- Refatoração e depuração do código.
- Possui editor extensível, desta forma é possível adicionar suporte para outras linguagens.
- Executa em plataformas Windows, Linux e Mac OS X.
- É possível acrescentar *plug-ins* expandido a capacidade de recursos da IDE.
- Inclui integração com banco de dados.

Apesar do NetBeans atender usuários de níveis básico, intermediário e avançado, desenvolvedores iniciantes podem ter uma certa dificuldade nas primeiras utilizações, pois alguns recursos como compiladores são acrescentados após a instalação da IDE, assim como o suporte a demais linguagens que não estão inclusas no pacote de instalação. Outra desvantagem é o alto consumo de recursos da máquina (memória e processamento), causando lentidão caso a capacidade seja reduzida, exigindo que o usuário possua um hardware mais robusto para utilizar a IDE sem travamentos.

### 3.3 Linguagens de Programação para Web Sites

As linguagens de programação para a web são os *scripts* que compõem os códigos ou sequências de códigos, que formam os arquivos existentes em um web site. Cada linguagem oferece recursos e algumas funções que uma outra não possui. Por isso é necessário saber quais os recursos que serão necessários a um web site para escolher o que será usado ao invés de fazer uma opção sem critérios por uma linguagem em detrimento da outra. Entre tantas linguagens existentes é difícil determinar qual é a melhor. O que se pode determinar é qual é a melhor para uma aplicação específica. Como não é possível saber com precisão quais recursos existem na máquina dos usuários, uma saída é reduzir ao máximo os recursos que dependem da máquina do usuário, utilizando recursos que dependem da servidora (DZENDZIK, 2005).

Dzendzik (2005) cita alguns critérios que devem ser considerados ao escolher uma linguagem:

- A linguagem escolhida é suportada pela máquina servidora disponível para a aplicação em questão?

- As sequências de comandos (*scripts*) destinam-se a máquina cliente ou a servidora?
- Que navegadores ou servidores possuem os recursos que serão necessários para a exibição das páginas?

É muito importante saber também quais são as técnicas e padrões mais utilizados e recomendados quando se assume o papel de programador front-end<sup>9</sup> para atingir o objetivo de transformar imagens estáticas, os layouts, em código que os navegadores entendem e exibem como páginas da Web. A única linguagem que o navegador consegue interpretar para a exibição de conteúdo é o HTML.

Um navegador, por padrão, pode não exibir caracteres acentuados corretamente e não exibir quebras de linha. Para que se possam exibir as informações desejadas com a formatação, é necessário que cada trecho de texto tenha uma marcação indicando qual é o significado dele. Essa marcação também influencia a maneira com que cada trecho do texto será exibido. As marcações do HTML são chamadas de tags, e elas basicamente dão significado ao texto contido entre sua abertura e fechamento.

Apesar de serem corretamente marcadas, usando apenas o HTML, as informações não apresentam nenhum atrativo estético e, nessa deficiência do HTML, reside o primeiro e maior desafio do programador front-end. O HTML foi desenvolvido para exibição de documentos científicos. Para ter-se uma comparação, é como se a web fosse desenvolvida para exibir monografias redigidas e formatadas pela Metodologia do Trabalho Científico da ABNT. Porém, com o tempo e a evolução da Web e de seu potencial comercial, tornou-se necessária a exibição de informações com grande riqueza de elementos gráficos e de interação. Para suprir essa deficiência do HTML foram criadas as linguagens de programação para páginas dinâmicas, que são as linguagens Server Side, e também as linguagens de programação ClientSide. Nos subtópicos dessa seção, há uma abordagem sobre esses dois tipos de linguagens e exemplos das mais utilizadas, no desenvolvimento de páginas web.

### **3.3.1 Linguagens de Programação para Páginas Dinâmicas- Server Side**

---

<sup>9</sup>**Front-end** é toda a parte da apresentação visual de um site. A forma como o conteúdo se apresenta na tela, a estrutura hierárquica das informações e a aplicação do design para a exibição das informações (MobGeek, 2015).

As Linguagens de Programação Server Side, são códigos de programa que são processados no servidor. Devido a este fato, não é necessário preocupar-se com a linguagem que o código foi criado, pois o servidor é quem se encarrega de interpretá-lo e de devolver uma resposta para o cliente (DZENDZIK, 2005).

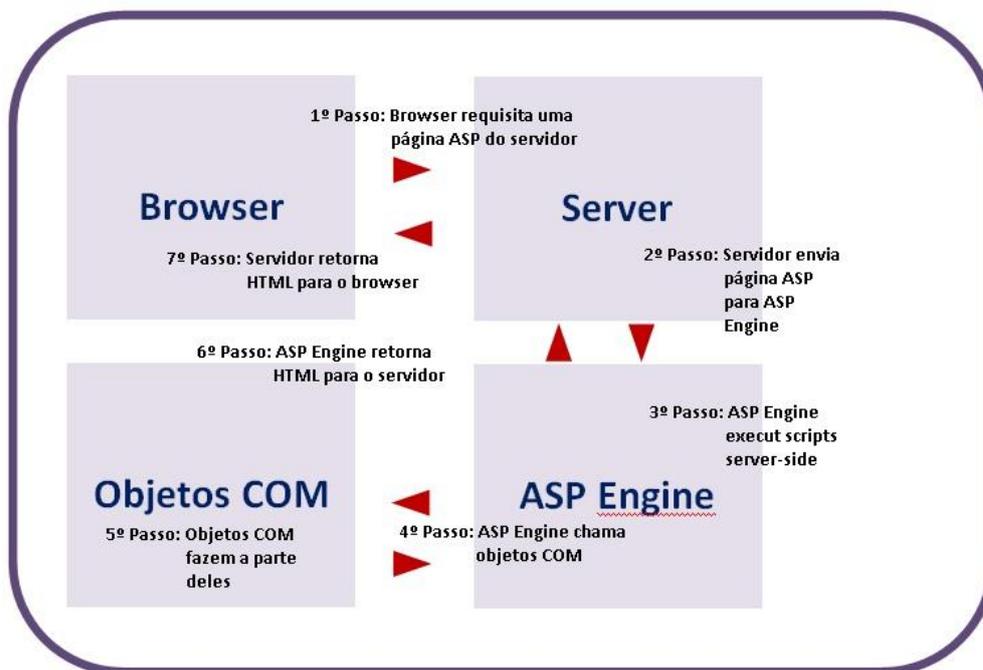
### 3.3.1.1 ASP

A *Active Server Pages* (ASP) é uma tecnologia orientada a objetos, criada pela Microsoft, utilizada para desenvolver páginas HTML dinamicamente. A ASP trabalha com linguagem de scripts VBScript baseada no Visual Basic da própria Microsoft. Permite escrever scripts que serão executados no servidor, produzindo páginas dinâmicas e interativas. O servidor é que se encarrega de transformar os scripts em páginas HTML, fazendo com que qualquer navegador do mercado seja capaz de ler um site que utilize ASP (MACHADO, 1999).

A tecnologia ASP dispõe do recurso de *Server Side Include* (SSI) que é um processo em que o servidor utiliza as informações de um arquivo e as inclui como parte de outro. A tecnologia ASP já é em si uma espécie de SSI, na qual o servidor utiliza um arquivo HTML e procura por comandos que precisam ser executados e inseridos antes de retornar uma página como resultada dos scripts incorporados (DZENDZIK, 2005).

Uma página ASP pode incluir chamadas para objetos intrínsecos ou para outros componentes ativos de servidor. Quando uma página ASP é chamada em um browser, o servidor Web passa a requisição para a máquina ASP – ASP Engine (ao oposto das requisições HTML, que o servidor Web mesmo as processa). A máquina ASP processa o script e insere os resultados dentro do fluxo HTML, que é então retornado para o browser requisitante como se pode verificar na Figura 31.

Figura 31 – Chamada de uma página ASP pelo Browser.



Fonte: Zeilmann (2002).

Segundo Zeilmann (VIRTUAL SCAPE 2002<sup>10</sup> apud ZEILNANN 2002), não é necessário nenhum software especial para escrever as páginas ASP, porque a tecnologia ASP integra seus códigos especiais com códigos HTML, logo, qualquer ferramenta de autoria web que permite edição de HTML (como o FrontPage, Page Mill ou até mesmo o bloco de notas do Windows) pode ser utilizada. Entretanto, se forem integrados controles Activex com ASP, é necessário o uso de um ambiente apropriado de desenvolvimento assim como Visual Basic ou Visual C++. Pode ser utilizado também o Visual Interdev da Microsoft para auxiliar na construção das páginas ASP.

### 3.3.1.2 PHP

PHP (PHP.NET) é uma linguagem de script open source, interpretada, muito utilizada para o desenvolvimento de aplicações web. Incorporada ao HTML, permite a criação de sites dinâmicos, possibilitando uma interação com o usuário. Esta linguagem trabalha em conformidade com modelo cliente-servidor, onde o servidor é responsável por interpretar e processar os scripts que contém o documento solicitado pelo cliente,

<sup>10</sup> VIRTUAL SCAPE (Org.). **Active Server Pages Hosting**. 2002. Disponível em: <<http://www.virtualscape.com/services/asp.html>>. Acesso em: 12 fev. 2015.

transformando-os em código HTML e retornando o resultado, em HTML, ao cliente que fez a solicitação (CHAVES E SILVA, 2008).

O PHP pode ser utilizado em quase todos os sistemas operacionais e é suportado pela maioria dos servidores Web atuais, principalmente pelo Apache. A característica principal, e mais forte, da linguagem é o suporte a uma grande variedade de bancos de dados. O PHP acessa diretamente os principais bancos de dados utilizados atualmente, como dBase, Interbase, MySQL, Oracle, PostgreSQL, SyBase e outros (br.php.net, 2016).

A tecnologia PHP incorpora a linguagem PHP que é baseada nas linguagens C, Java e Perl e ainda pode ser vista como uma combinação de linguagem de programação e servidores de aplicações (CHAVES E SILVA, 2008). A linguagem base da PHP é a JavaScript que quando usada fora de um software servidor de PHP é uma linguagem interpretada na máquina cliente. Quando utilizada em um servidor de PHP é chamada de “linguagem PHP” e é interpretada no servidor (DZENDZIK, 2005).

### 3.3.1.3 JSP

A Java Server Page (JSP) é uma linguagem de script desenvolvida para possibilitar a criação de páginas dinâmicas. Ela é totalmente orientada a objetos, baseada na linguagem Java, possui em sua estrutura tags HTML, entre as quais são inseridas as sentenças Java a serem executadas no Servidor (SOUSA, 2009). Pode-se fazer basicamente qualquer coisa com JSP, desde coletar dados de formulários até gerar páginas dinâmicas oriundas de fontes de dados quaisquer (BARNABÉ, 2010).

Toda página JSP durante seu primeiro acesso é transformada em um Servlet, o que faz com que esta etapa demore mais para ser respondida. A partir do segundo acesso a resposta é mais rápida, pois o servidor acessa diretamente o Servlet criado no primeiro acesso (SOUSA, 2009). Segundo Sousa (LOPES 2008<sup>11</sup> apud SOUSA 2009), Servlets são classes Java para a WEB.

A tecnologia JSP não oferece nada além do que pode obter-se através de Servlets, entretanto, oferece a vantagem de ser facilmente codificado, o que facilita a

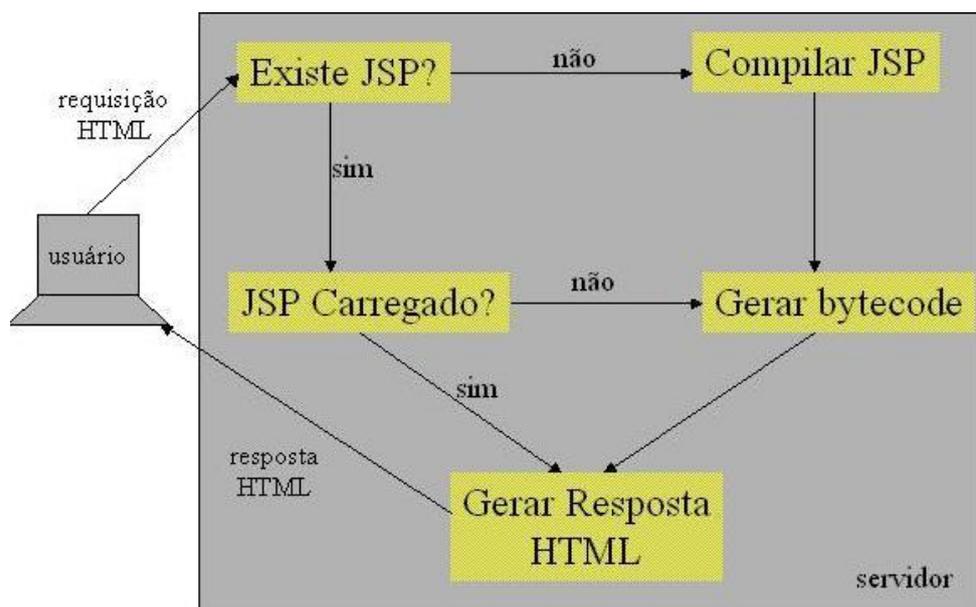
---

<sup>11</sup> LOPES, Camilo. **Entendendo Servlet**. 2008. Disponível em: <<https://camilolopes.wordpress.com/2008/06/09/entendendo-servlet/>>. Acesso em: 06 mar. 2015.

elaboração e manutenção de um Web Site. Além disso, a JSP permite que a programação lógica (parte dinâmica) seja separada da programação visual (parte estática), facilitando assim o desenvolvimento de aplicações mais robustas, portáteis, seguras e eficientes, em que o designer e o programador podem trabalhar juntamente, em um mesmo projeto, de forma independente (BARALE, 2007).

O Funcionamento de Web Sites que utilizam a tecnologia JSP se dá através de requisições efetuadas pelos usuários através de um browser, a página então requisitada, será processada pelo servidor. Se for a primeira vez que a página JSP for acessada, ela será transformada em um Servlet que é compilado e gera um bytecode (conhecido por .class), a partir da qual é gerada uma página HTML que é enviada ao cliente através do browser. A partir da segunda vez que essa página for acessada, apenas é verificado se ocorreram ou não quaisquer mudanças, caso negativo, apenas o bytecode é chamado para gerar a página HTML (BARALE, 2007). A Figura 32 ilustra esse Funcionamento.

Figura 32 - Funcionamento JSP



Fonte: Barale (2007)

### 3.3.2 Linguagens de Programação para páginas Dinâmicas – Client Side

As Linguagens de Programação Client Side, são interpretadas na máquina cliente, onde o conteúdo gerado é exibido conforme os recursos disponíveis no

navegador utilizado. Se o navegador não tiver os recursos que estão nos *scripts*, parte do conteúdo não poderá ser visualizada pelo usuário (DZENDZIK, 2005). A seguir, serão abordadas as linguagens Cliente Side mais utilizadas atualmente.

### 3.3.2.1 HTML

HTML (*Hyper Text Markup Language*) não se trata de uma linguagem de programação propriamente dita, mas sim de uma linguagem de marcação utilizada para estruturar e representar o conteúdo de páginas web (ANACLETO, 2012). Possui um sistema de anotação, através de um conjunto de tags, que permite distinguir diversos elementos através da sua sintaxe, afetando a maneira como os textos e os outros elementos são exibidos pelo navegador. É com recurso a esta linguagem que se definem todos os elementos, mais ou menos complexos, de uma página Web (ROCHA, 2014).

O HTML foi criado com o objetivo de dar à Internet um aspecto gráfico, em conjunto com o protocolo HTTP e com os navegadores, foi a responsável pela popularidade da Internet, pois até a sua criação, as ferramentas existentes, tais como *ftp*, *gopher* e *telnet*, rodavam em terminais de caracteres e, apesar de muito úteis e populares no meio acadêmico eram pouquíssimas atrativas ao grande público (DZENDZIK, 2005). O HTML se criou sem dar respostas a todos os possíveis usos que lhe dariam posteriormente e a todo coletivo de gente que o utilizariam no futuro, pois não se pensou que a web chegaria a ser uma área de ócio com caráter multimídia. Entretanto, frente a este deficiente planejamento, com o tempo, foram se incorporando modificações as quais são os padrões do HTML. Numerosos padrões já se apresentaram.

O HTML5 trata-se da quinta versão da linguagem HTML, na qual estão estabelecidos os padrões mais recentes da linguagem. Esta nova versão vem adaptar o HTML às novas exigências da Web com novas capacidades, maior simplicidade e melhor interação com os mais recentes tipos de multimídia (ROCHA, 2015). O HTML5 deriva da cooperação entre duas organizações, a World Wide Web Consortium (W3C) e a Web Hypertext Application Technology Working Group (WHATWG). E vem substituir a anterior especificação HTML4.01, que já contava com uma década de existência desde que foi publicada pela W3C em 1999 (ANACLETO, 2012).

### 3.3.2.2 CSS

Tags HTML para estilo são má prática hoje em dia e jamais devem ser usadas. Em seu lugar, surgiu a CSS (*Cascading Style Sheets*), objetivo único de cuidar da estilização da página. Nessa tecnologia são definidas folhas de estilo para personalização das páginas web, ou seja, CSS é uma linguagem de layout padrão para a Web que define cores, tipografia, tamanho e disposição de conteúdo (REIS, 2007). A CSS pode ser utilizada em meios aos códigos de HTML ou de forma separada, porém os padrões Web sugerem como norma que seja separado (OLIVEIRA et. al., 2014).

Segundo Dzendzik (2005) “as folhas de estilo facilitam o trabalho de um *web design* quando se trata de garantir uma formatação homogênea e mais criativa por todas as páginas de um *web site* e ainda permite mais interatividade com o usuário. Mesmo que se deseje mudar todo o *design*, ou um certo grupo de formatação, as folhas de estilo permitem que uma alteração possa repercutir em todas as páginas do *web site*. As folhas de estilo podem ser comparadas a um gabarito de formatação de páginas HTML. Ela permite que se alterem quase todas as *tags* da linguagem HTML. Sua limitação está na falta de reconhecimento por algumas versões de navegadores, sendo utilizada, portanto, como uma linguagem complementar ao HTML”.

A partir das diversas características citadas e sendo bem utilizada, Reis (2007) cita os inúmeros benefícios práticos que a CSS proporciona, como:

- “Conservação da largura de banda do usuário, acelerando os tempos de carga da página, especialmente através de uma conexão discada;
- Redução da sobrecarga do servidor e da largura de banda do proprietário, assim economizando dinheiro;
- Redução do tempo de design e desenvolvimento;
- Aumento da interoperabilidade aderindo aos padrões web;
- Aumento da acessibilidade removendo alguns, vários ou todos os elementos de apresentação da marcação.”

### 3.3.2.3 JavaScript

JavaScript é uma linguagem de scripts voltada para executar operações no lado do cliente. Ela não é uma linguagem compilada, ela é interpretada e é inserida junto com o código fonte HTML das páginas web. Com ela, é possível fazer validações de campos, abertura de janelas, controle da utilização de botões, mensagens de alertas e confirmações. Sua principal função é criar uma interatividade maior do usuário com a página utilizada (LIMEIRA, 2006).

A linguagem JavaScript possibilita o uso de diversos objetos na composição de uma página. Todos eles possuem propriedades que podem ser alteradas, além disso, os objetos fornecem eventos que possibilitam que uma página execute uma ação conforme instrução de um usuário. JavaScript é uma linguagem estruturada que usa objetos, mas não é uma linguagem de programação orientada a objetos, nela os objetos são usados para representar algum aplicativo (DZENDZIK, 2005). Em Java Script apenas encontram-se alguns conceitos de orientação a objetos, como o conceito de objeto, herança implícita, encapsulamento e o polimorfismo. Porém, não possui alguns conceitos importantes, como o conceito de objeto, herança implícita, encapsulamento e o polimorfismo (LIMEIRA, 2006).

A linguagem JavaScript é processada diretamente no navegador, dispensando a ajuda de um servidor. Ao contrário da HTML que é uma linguagem estática, com a JavaScript se fazem animações com textos e imagens e diversas interatividades com usuários, sendo assim considerada um acessório da HTML (DZENDZIK, 2005).

Segundo Rodrigues (2014), “o Java Script permite editar criar ou remover informação contida numa página HTML de uma maneira dinâmica e fluída, como:

- Introduzir código HTML na página de uma forma dinâmica, como, para validar um formulário e, caso existam inconsistências, alertar o utilizador. Este tipo de técnicas evita assim chamadas desnecessárias ao servidor.
- Atribuir eventos a elementos, permitindo, por exemplo, que com um simples clique num botão seja colocado conteúdo disponível ao utilizador.
- Criar cookies, de modo a que os utilizadores possam ter uma experiência mais personalizada nas próximas vezes que usufruírem da aplicação web.”

### **3.4 Hospedagem de Páginas Web**

Qualquer serviço que se quer oferecer na Internet tem que provê-lo um servidor, que é um computador que se encontra ligado 24 horas por dia e também, conectado na Internet permanentemente. No caso de uma página web, existem alguns servidores que são os encarregados de mandá-la sempre que é solicitada, estes são os servidores web. As páginas têm que estar hospedadas em um servidor web para que possam estar acessíveis na Internet.

### **3.5 Responsividade de Páginas Web**

Além do acesso por meio do navegador dos computadores, hoje o acesso à Internet a partir de dispositivos móveis representa um grande avanço da plataforma, mas também implica em um pouco mais de atenção ao trabalho que um programador front-end tem que realizar. É preciso conhecer e utilizar os recursos disponíveis para atender a essa nova necessidade que é a responsividade das páginas webs, ou seja, a capacidade de redimensionamento do conteúdo das páginas em diferentes tamanhos de telas.

#### **4. EXPERIMENTO REMOTO BASEADO EM UM SISTEMA HÍBRIDO DE GERAÇÃO DE ENERGIA**

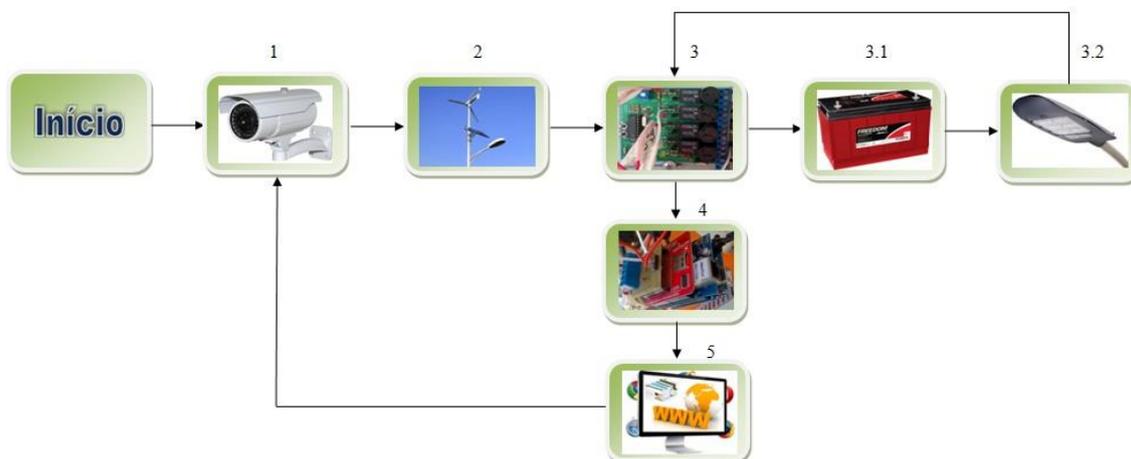
No capítulo 1 foi apresentado o sistema híbrido, suas características, componentes que o formam e a sua finalidade que é aproveitar o máximo da energia gerada pelas fontes alternativas para manter uma luminária LED acesa durante o período noturno, porém ao estudar este sistema foi possível observar o potencial que o mesmo propõe para fins de pesquisa. Mais do que manter uma luminária LED acesa durante o período noturno, pode-se utilizar deste para extrair informações sobre o seu estado de operação e o acúmulo destas informações durante um período de tempo pode servir de base para pesquisa, como verificar os valores de corrente e tensão elétrica gerados para entendimento do funcionamento de sistemas alternativos.

Além de extrair as informações do estado de operação do sistema híbrido, é interessante que essas informações estejam disponíveis em meio público para fins educacionais. Baseado nesta ideia viu-se a necessidade de aperfeiçoar as capacidades do sistema híbrido, propondo a criação de um experimento remoto do mesmo. Basicamente este experimento remoto é composto de um sistema de aquisição de dados que é formado por um hardware e um software, uma página web no qual estas informações serão disponibilizadas e um sistema de monitoramento remoto que é composto por uma web câmera que irá exibir o funcionamento do sistema híbrido, assim o usuário poderá visualizar o mesmo operando em tempo online através da página web. Esta página web pode ser acessada por qualquer dispositivo que possua um navegador com acesso à internet.

A Figura 33 ilustra o esquema de funcionamento do experimento remoto, iniciando pelo primeiro bloco que representa a web câmera. A web câmera está posicionada de forma a monitorar a estrutura do sistema híbrido (figura do bloco 2). Estrutura essa que é formada por um poste, no qual são fixados os elementos que serão monitorados em tempo online, sendo eles: gerador eólico, placa solar e luminária LED. Esses elementos estão diretamente ligados ao sistema de controle (figura do bloco 3), pois é este que irá selecionar a melhor fonte de energia para efetuar a carga da bateria, sendo elas: solar, eólica ou da rede concessionária de energia elétrica (bateria figura do bloco 3.1). A bateria possuindo carga irá manter a luminária LED acesa durante o período noturno (luminária LED figura do bloco 3.2). O fluxo retorna ao sistema de

controle, pois este processo entre o sistema de controle, bateria e luminária LED permanecerá sempre em *loop*.

Figura 33 – Diagrama de Bloco: Esquema de funcionamento do experimento remoto



Fonte: Arquivo pessoal

O sistema de controle (bloco 3) também possui ligação com o sistema de aquisição de dados, representado pelo bloco 4. O sistema de aquisição de dados realiza a medição das variáveis do sistema híbrido, estas já citadas em tópicos anteriores. E as informações extraídas são disponibilizadas em uma página web (bloco 5). A página web além de disponibilizar as informações de estado do sistema híbrido irá permitir que o usuário visualize o experimento real, através da captura de vídeo da web câmera. Por este motivo o fluxo do diagrama retorna ao primeiro bloco, representado pela web câmera, formando um *loop*.

Nos próximos tópicos deste capítulo, faz-se a apresentação das camadas física e lógica do experimento remoto descrito acima. Sendo que na camada física haverá um software embutido responsável por disponibilizar os dados necessários que serão captados pela página web através do código de programação implementado na mesma, o que forma a camada lógica do experimento.

#### 4.1 Desenvolvimento da Camada Física do Experimento Remoto- Controle e Funcionamento

O sistema embarcado apresentado no tópico 1.6 do capítulo 1 opera como um sistema inteligente, sendo responsável por controlar o sistema de geração de energia

híbrido. É ele que seleciona qual a melhor fonte de energia para realizar a carga da bateria, monitora o nível de tensão da mesma e aciona a luminária LED quando se aproxima o período noturno. Porém, esse sistema não foi projetado para trazer informações de estado do sistema híbrido, apenas mantém o mesmo em operação realizando as rotinas que foram implementadas no software embarcado, com a finalidade de controlar o sistema da melhor forma possível.

Como visto na proposta deste trabalho o objetivo é criar e disponibilizar um experimento remoto baseado em um sistema híbrido de energia alternativa eólico-solar, para que o usuário possa acessar remotamente dados do sistema híbrido on-line. Porém, para que esses dados sejam extraídos do sistema e disponibilizados em uma página Web para acesso remoto, torna-se necessário a criação de um sistema de aquisição de dados, que irá extrair dados em tempo real do estado em que o sistema híbrido se encontra. Este sistema de aquisição de dados é formado por um conjunto de hardware e um software embutido no mesmo, e serão apresentados nos subtópicos seguintes.

#### **4.1.1 Placa de Aquisição de Dados**

O microcontrolador implantado no sistema híbrido pode ser utilizado para extrair os dados do sistema, porém o mesmo só recebe valores de tensão em suas portas analógicas, com uma variação de 0 à 5V corrente contínua, e o sistema híbrido opera com valores de corrente contínua em Ampères. Sendo assim, foi necessária a criação de uma placa de aquisição de dados, que faz a conversão da corrente obtida do sistema híbrido em tensão, para que o microcontrolador possa então recebê-la em forma de sinais. Estes sinais são interpretados pelo microcontrolador, que mede os mesmos e armazena-os dentro de variáveis. As variáveis que o microcontrolador recebe do sistema híbrido, são quatro analógicas e duas digitais. Assim sendo, as variáveis analógicas armazenarão os valores:

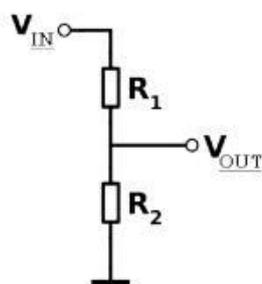
1. da corrente do carregador, cuja fonte de energia é proveniente do sistema concessionário de abastecimento de energia elétrica;
2. da corrente do carregador do sistema de energia eólico;
3. da corrente do carregador do sistema de energia solar;
4. da tensão da bateria, para monitoramento.

E as variáveis digitais, que armazenam um valor binário de um dígito (0 ou 1) informando:

1. se a luminária LED está acesa (1) ou apagada (0);
2. se o carregador elétrico, que utiliza a energia da concessionária, está ligado (1) ou desligado (0).

A placa de aquisição de dados possui em seu circuito alguns capacitores e resistores necessários. Os resistores operam como um divisor de tensão, que consiste em um método utilizado para criar uma tensão de saída ( $V_{out}$ ), que é proporcional a uma tensão de entrada ( $V_{in}$ ), que pode operar como um sinal atenuador de baixa frequência. Desta forma, é possível monitorar a tensão da bateria (MARTINEZ, 2013). Pode-se analisar o esquema de um divisor de tensão ilustrado na Figura 34, onde  $R_1$  e  $R_2$  representam os resistores.

Figura 34 – Esquema de um divisor de tensão



Fonte: Martinez (2013).

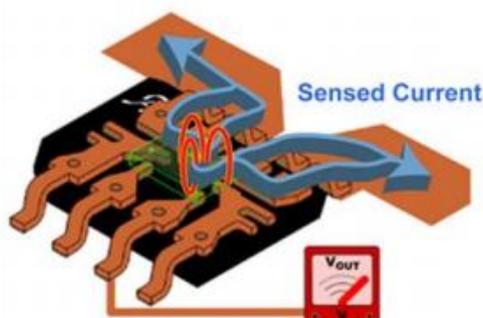
No caso do carregador da concessionária não é necessário o monitoramento da tensão, pois este já é por padrão 12V que irá alimentar a bateria na falta de energia alternativa do sistema híbrido.

O componente principal da placa de aquisição de dados é o sensor de corrente ACS712 fabricado pela Allegro Micro Systems. A leitura do sensor é feita por efeito Hall, que para sua medição consiste em um material ferromagnético e uma placa semicondutora, de forma que o campo magnético gerado pela corrente que percorre o material em questão, e fica posicionado perpendicularmente a esta placa e isolados eletricamente entre si. Esta placa é percorrida por uma corrente constante e conhecida. A corrente primária a ser mensurada gera então um campo magnético que causa uma força de arrasto

nos elétrons conduzidos na placa semicondutora, gerando assim uma diferença de potencial proporcional a corrente a primária (MARTINEZ, 2013).

A Figura 35 ilustra o sensor de efeito Hall. É possível observar o campo magnético gerado ao centro do circuito e o sentido da corrente e tensão indicado pelas setas.

Figura 35 – Sensor de efeito Hall com encapsulamento em circuito integrado



Fonte: Martinez (2013).

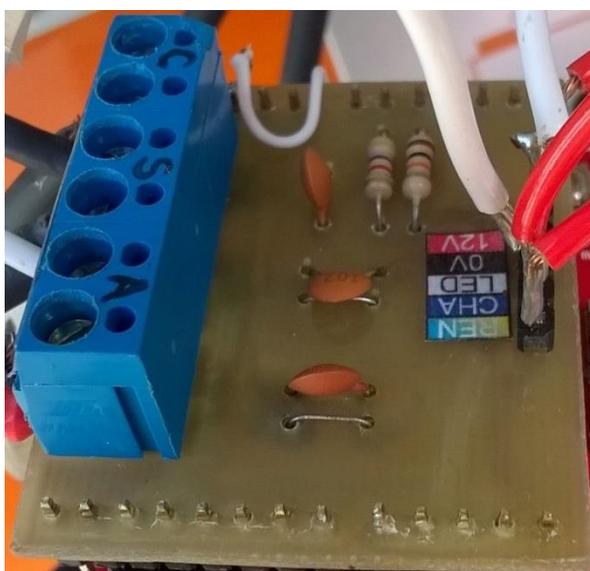
O ACS712 é o componente responsável por converter a corrente em tensão, para que o microcontrolador possa receber as informações do estado em que o sistema híbrido se encontra. O sensor possui três pinos principais: VCC, GND e OUT (sinal). O VCC é o pino de alimentação do sensor, este é alimentado com 5V (corrente contínua) que pode receber do próprio micro controlador, assim como o GND. O pino OUT trata-se de uma saída analógica e através dela é possível receber a leitura da corrente. O sensor ACS712 é bidirecional e pode efetuar leituras que vão de -30A à +30A com equivalente a 66mV/A. A leitura realizada pelo componente é feita através do efeito HALL, desta forma não existem ligações físicas entre os bornes de leitura do sensor e a saída analógica (DUTRA, 2013).

Como citado anteriormente o ACS712 é bidirecional, devido a isto, quando ocorrer de a corrente lida pelo sensor for igual a zero, ou seja, quando nenhuma corrente for detectada no sensor, a tensão será de 2,5V ( $VCC/2$ ) e a leitura do microcontrolador será algo em torno de 512. Assim conforme a corrente for positiva ou negativa a leitura irá aumentar ou diminuir respectivamente. Isto ocorre pelo fato de o microcontrolador

usar em suas portas analógicas uma lógica que vai de 0 a 1023, sendo 0 = 0V e 1023 = 5V (DUTRA, 2013).

A Figura 36 mostra a placa de aquisição de dados que foi desenvolvida com o intuito de converter corrente em tensão para que o microcontrolador da plataforma de prototipagem possa realizar a leitura dos dados do sistema híbrido. A placa foi projetada para ser acoplada nos pinos do microcontrolador como se fosse uma extensão do mesmo, uma espécie de “*shield* conversor de corrente”.

Figura 36 – Placa de Aquisição de Dados (vista superior)



Fonte: Arquivo Pessoal

#### 4.1.2 Plataforma de Prototipagem Utilizada no Sistema de Aquisição de Dados

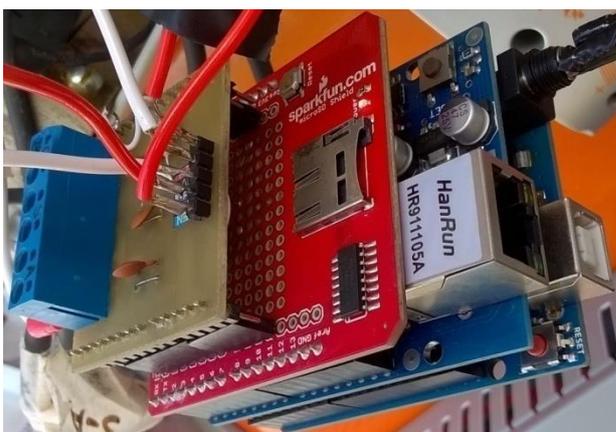
Após o estudo realizado, no tópico 2.1.1.1 deste trabalho, sobre algumas plataformas de prototipagem eletrônica disponíveis no mercado, optou-se por utilizar a plataforma Arduino UNO R3 no sistema de aquisição de dados do sistema híbrido de energia alternativa eólico-solar. O mesmo possui uma grande quantidade de expansões, entradas e saídas que podem ser conectadas a ele, dispõe de um bom processamento para o uso no projeto proposto, além da facilidade em encontrar referências que auxiliam na programação. Analisando o custo de implementação e as possibilidades que a plataforma permite ao usuário o Arduino UNO R3 foi a que apresentou um melhor custo-benefício.

O grande diferencial da plataforma de prototipagem eletrônica Arduino é a flexibilidade em relação ao hardware. A mesma possui em uma placa única os componentes básicos para a prototipagem, porém conforme a necessidade do projeto é possível acrescentar placas de expansão específicas que irão atender um determinado projeto. Isso evita o desperdício de recursos de hardware da plataforma e reduz o custo de aquisição da mesma.

#### 4.1.3 Placas de Expansão Arduino (*Shields*)

As placas de expansão do Arduino, também chamadas de *shields*, são placas de circuito que podem ser conectadas nos pinos da parte superior do Arduino, encaixando-se perfeitamente e expandindo a capacidade de hardware da plataforma. Existem diversos tipos de *shields* disponíveis no mercado, estas placas podem conter displays de LCD, sensores, controladores, dispositivos de comunicação entre outros (LEMOS, 2013). O usuário também pode criar o seu próprio *shield* caso necessite de algo mais específico em seu projeto, como foi o caso da placa de aquisição de dados apresentada no tópico 3.1.1 que opera como um *shield* conversor de corrente no sistema de aquisição de dados do sistema híbrido de energia. A Figura 37 ilustra a plataforma de prototipagem Arduino UNO R3 utilizada no sistema de aquisição de dados do sistema híbrido e suas placas de expansão conectadas na parte superior e inferior.

Figura 37 – Arduino UNO R3 e suas placas de expansão (*shields*)

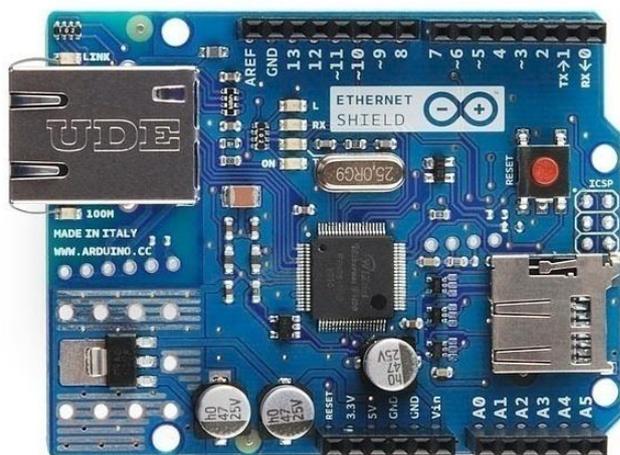


Fonte: Arquivo Pessoal

#### 4.1.4 Arduino Ethernet Shield

O *Shield Ethernet* é uma placa de expansão que permite a conexão do Arduino com a rede *Ethernet*. Desta forma é possível aumentar as possibilidades de utilização da plataforma de prototipagem. Com o uso do *Shield Ethernet* o Arduino pode operar como um micro servidor web, permitindo que as informações obtidas através do código implementado no Arduino possam ser disponibilizadas na rede, como é o caso do projeto proposto onde o Arduino operará como um micro servidor web, facilitando a captura das informações de estado do sistema híbrido para a disponibilização na página web. A Figura 38 ilustra a placa de expansão Arduino *Shield Ethernet*.

Figura 38 – Arduino *Shield Ethernet*



Fonte: (arduino.cc)

A placa é baseada no chip Wiznet ethernet W5100 fornecendo uma pilha (TCP/IP). O Arduino *Shield Ethernet* suporta até quatro conexões simultâneas. A pinagem exposta é a 1.0 em revisão 3 da placa Arduino UNO (Autor desconhecido). Segue abaixo as características da placa de expansão Arduino *Shield Ethernet*.

- Suporte a protocolos TCP/IP: TCP, UDP, ICMP, ARP IPv4, IGMP, PPPoE, Ethernet;
- 10BaseT/100BaseTX Ethernet PHY integrado;
- Suporte (Full-duplex e half-duplex);
- Suporte Auto MDI/MDIX;

- Conexão ADSL Support (com suporte ao protocolo PPPoE com PAP/CHAP modo de autenticação);
- Suporta 4 conexões independentes em simultâneo - Não suporta fragmentação de IP;
- Memória Interna 16Kbytes para Tx/Rx Buffers;
- Operação de 3.3V com 5V tolerância sinal I / O;
- Suporte Interface Serial (SPI modo 0, 3);
- Saídas função Multi-LED (TX, RX, Full/Half duplex, Colisão, Speed Link);
- O chip W5100 Wiznet fornece uma rede (IP) com suporte a TCP e UDP;

O *shield* contém uma série de LEDs informativos:

- PWR: indica que a placa *shield* está alimentada;
- LINK: indica a presença de uma ligação de rede e pisca quando o *shield* transmite ou recebe dados;
- FULLD: indica que a conexão de rede é full duplex;
- 100M: indica a presença de uma conexão de rede 100 Mb/s (em oposição a 10 Mb/s);
- RX: pisca quando o *shield* recebe dados;
- TX: pisca quando o *shield* envia os dados;
- COLL: pisca quando são detectadas colisões de rede.

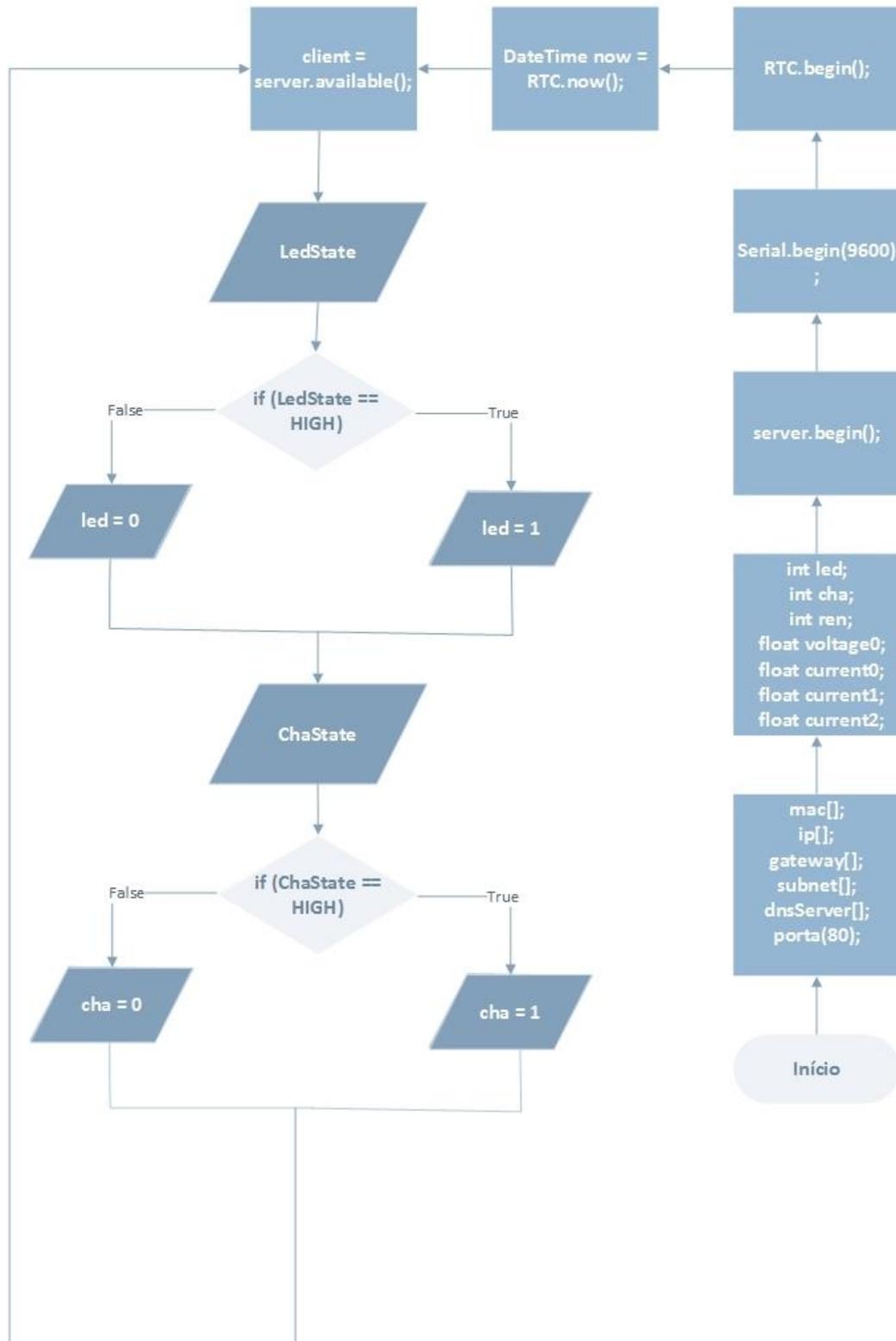
#### 4.1.5 Software do Sistema de Aquisição de Dados

A placa de aquisição de dados obtém os sinais elétricos do sistema de controle embarcado. A mesma converte esses sinais que operam com valores de corrente em uma tensão que o Arduino possa suportar nas suas portas analógicas e digitais. As variáveis que serão disponibilizadas na página Web devem demonstrar valores de estado real do sistema híbrido. Desta forma foi necessário o desenvolvimento de um software que irá realizar as rotinas necessárias para a aquisição de dados, de tal forma que esses dados possam ser capturados e disponibilizados em uma página web. Para a construção do mesmo foi utilizada a IDE de desenvolvimento do Arduino que é compatível com os sistemas operacionais Windows, Linux e Mac OS. E o código desenvolvido em linguagem Arduino que é semelhante à linguagem C/C++.

O Arduino UNO R3 possui 6 portas analógicas denominadas A0, A1, A2, A3, A4 e A5. A porta A0 recebe o sinal de corrente do carregador elétrico que é alimentado pela energia da rede concessionária de abastecimento. A porta A1 recebe o sinal de corrente que está sendo gerada pelo carregador do sistema de energia solar. A porta A2 recebe o sinal de corrente que está sendo gerada pelo carregador do sistema de energia eólica. A porta A3 recebe o sinal com nível de tensão que a bateria possui no momento. As portas A4 e A5 estão desativadas, pois não estão sendo utilizadas.

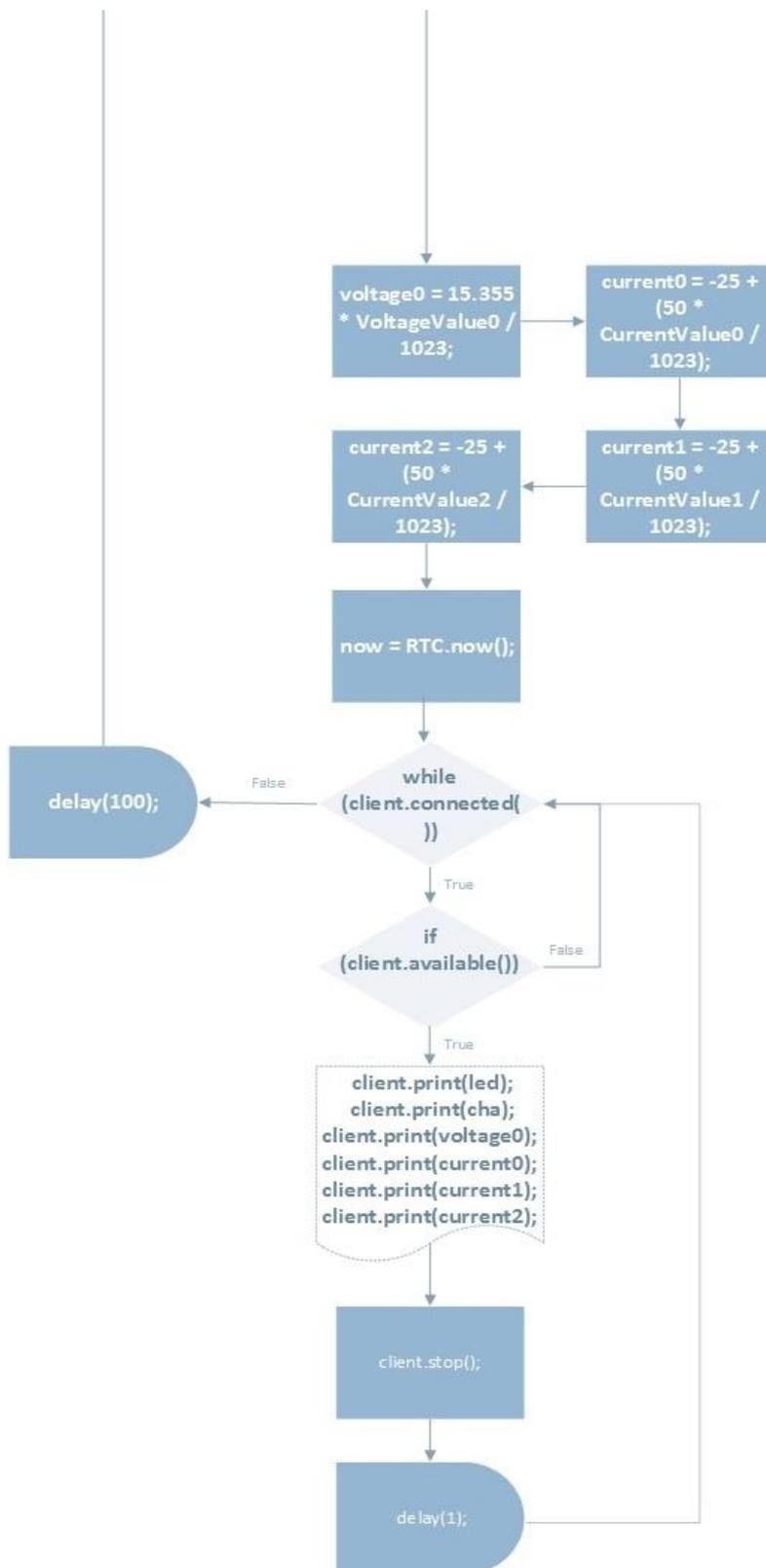
As portas digitais do Arduino UNO R3 são 14 denominadas de 0 até 13. A porta 2 recebe o sinal de estado do LED, se o mesmo está aceso ou apagado, onde 0 indica LED apagado e 1 indica LED aceso. A porta 3 recebe o sinal do estado de operação do carregador elétrico, que é alimentado pela energia da rede concessionária de abastecimento elétrico. Onde o sinal 0 indica que o carregador está desligado e 1 indica ligado. As portas 10, 11, 12 e 13 estão sendo utilizadas pelo *shield* ethernet e as demais estão desativadas por ausência de uso. A Figura 39 ilustra o fluxograma das rotinas realizadas pelo software desenvolvido para a aquisição de dados.

Figura 39 – Fluxograma: Código do software do sistema de aquisição de dados, parte 1



Fonte: Arquivo Pessoal

Figura 40 – Fluxograma: Código do software do sistema de aquisição de dados, parte 2



Fonte: Arquivo Pessoal

A primeira rotina a ser executada é a definição das configurações de rede ethernet, onde será definido o endereço MAC, IP do servidor, endereço *Gateway*, máscara de sub-rede, servidor DNS e a porta HTTP padrão. Após isto são declaradas as variáveis que recebem os valores correspondentes às portas analógicas e digitais do Arduino. O servidor é inicializado e abre uma conexão no qual o canal de comunicação é a rede cabeada. São liberadas as portas do Arduino para a leitura dos sinais elétricos recebidos da placa de aquisição de dados. Inicializa-se o RTC (relógio de tempo real) buscando a hora e data atual. Declara uma variável chamada “now” que irá receber a data e hora atual do RTC. O servidor “avisa” para o cliente que está disponível “escutando” por conexões de entrada ethernet. Após isto se inicia o processo de leitura das variáveis digitais do sistema híbrido. É verificado o estado da luminária LED, se o estado for ligado ( $LedState == HIGH$ ) a variável “led” recebe o valor 1, caso contrário a luminária está desligada e a variável “led” recebe o valor 0. O estado do carregador elétrico que é alimentado pela energia proveniente do sistema de abastecimento concessionário também é verificado. Se o estado for ligado ( $ChaState == HIGH$ ) a variável “cha” recebe o valor 1, caso contrário o estado do carregador elétrico é desligado e a variável “cha” recebe o valor 0. O próximo passo é a leitura e cálculo das variáveis analógicas. É verificada a tensão da bateria, o quanto de carga a mesma possui no momento da leitura. O valor máximo de tensão de carga é 15.355V e como citado no tópico 3.1.1 O Arduino utiliza uma lógica em suas portas analógicas que vai de 0 a 1023, sendo  $0 = 0V$  e  $1023 = 5V$ , desta forma multiplica-se a tensão máxima de carga que a bateria pode suportar pelo valor de tensão que o Arduino está recebendo em suas portas analógicas ( $VoltageValue0$ ) e dividi-se o resultado da multiplicação por 1023. O resultado final é armazenado na variável “voltage0” ( $voltage0 = 15.355 * VoltageValue0 / 1023$ ). As demais variáveis analógicas são calculadas de forma diferente, pois receberão valores que representam a corrente medida de cada fonte geradora de energia. Como citado no tópico 3.1.1 quando nenhuma corrente estiver sendo detectada no sensor da placa de aquisição de dados, ou seja, quando o valor da corrente for igual a zero a tensão será de 2,5V e a leitura no Arduino terá o valor de 512. Os valores máximos de corrente gerada pelas fontes de energia do sistema híbrido é de 25A. Desta forma multiplica-se 50 pelo valor que o Arduino está recebendo, este valor é proveniente da fonte geradora de energia ( $CurrentValueX$ ) e dividi-se o resultado da multiplicação por 1023. Do resultado da divisão diminui-se 25 e obtém-se o valor final

que é armazenado em uma variável “currentX” ( $\text{currentX} = -25 + (50 * \text{CurrentValueX} / 1023)$ ). Após o cálculo das variáveis digitais e analógicas é salvo o horário e data atual na variável “now”, desta forma é possível saber o horário em que foram realizadas as medições. O próximo passo é informar os dados que foram medidos. É necessário verificar primeiramente se há cliente conectado, caso não haja é executado um *delay* de 100 milissegundos e o servidor retorna a “escutar” por conexões de entrada ethernet. Caso contrário, se o cliente estiver conectado e enquanto permanecer conectado passa para o próximo passo que é verificar se o cliente está disponível, caso esteja indisponível o servidor retorna a requisitar se tem cliente conectado. Caso o cliente esteja disponível os dados são informados ao mesmo, a conexão com o cliente é encerrada, ocorre um *delay* de 1milissegundo e o servidor retorna a verificar se há cliente conectado.

#### **4.1.6 Câmera IP D-Link DCS-950**

No capítulo 2 tópico 2.1 (Infraestruturas de Laboratórios Remotos) é feito uma abordagem da estrutura necessária para suportar um laboratório de experimentação remota, onde foi possível identificar os principais elementos que compõe um experimento remoto. Conforme citado no subtópico 2.1.1.1 (Câmera Web) a câmera web é um componente primordial na experimentação remota, pois esta irá capturar as imagens do experimento para que os usuários possam visualizá-las em um ambiente web.

Com base no estudo realizado optou-se pelo tipo de câmera IP por ser de fácil configuração e por não depender de um servidor para receber as imagens capturadas, sendo possível acessar essas imagens através da rede de internet/intranet. O modelo de câmera IP utilizado para capturar as imagens do sistema de iluminação com energia alternativa híbrida eólico-fotovoltaico foi o DCS-950 do fabricante D-Link. A grande vantagem deste modelo de câmera é a compatibilidade com a maioria dos roteadores e softwares de monitoramento remoto disponíveis o que facilita a configuração e utilização da mesma. A DCS-950 envia as imagens capturadas através da rede cabeada UTP, na Figura 41 é possível observar uma imagem do modelo de câmera IP utilizada.

Figura 41 - Câmera IP D-Link DCS-950



Fonte: <http://shop.by>

A D-Link DCS-950 apresenta um ótimo custo-benefício e pode ser encontrada no mercado custando em torno de 500 reais. O grande destaque fica por conta do software de gerenciamento que permite ao usuário gravar as imagens capturadas do monitoramento em um computador, basta selecionar qual diretório deseja gravar. Abaixo seguem as especificações técnicas da câmera IP D-Link DCS-950.

- Resolução: 0.3MP; 640x480 pixels;
- Memória: Incluso 32MB; interna / Incluso 4MB; Flash;
- Aplicação: Segurança / Web Cam;
- Interface: 1 RJ45 Interface;
- 1 Porta RJ-45 auto-sensing Fast Ethernet 10/100Mbps Hardware e Sistema;
- Ram: 32 MB;
- Flash: 4 MB;
- Sistemas Operacionais Suportados: Windows 98 SE, ME, 2000, XP, 7;
- Navegador Web: Internet Explorer 6 ou superior;
- Lente (Focal Length, Aperture): 4.5mm, f2.8;
- Resolução máxima: 640 x 480;
- Sensor de Imagem: 1/4color CMOS sensor, Progressive Scan Mode;
- Tipo de Lente: Fixa;
- Iluminação: Mín. 0.5 lux;

- Suporte à Controle Automático de Ganho (AGC): 20dB;
- Suporte de Exposição Automática (AE);
- Suporte de Balanço Automático de Branco (AWB);
- Disparador Eletrônico;
- Velocidade do Disparador: 1/60 a 1/15000 seg. Vídeo;
- Imagem Frame Rate: Até 30 fps;
- Compressão de Imagem: MPEG4;
- Resoluções Suportadas: 160 x 120 30fps, 320 x 240 30fps, 640 x 480 30fps;
- Viewing System Required Protocol: ActiveX;
- Suporte a Gravação de Vídeo;
- Controle de Brilho;
- Protocolos Suportados: HTTP, TCP/IP, UDP, SMTP client, DHCP client, ARP, ICMP, BOOTP, NTP, client, DNS, DDNS, UPnP, PPPoE;
- Text, Date & Time Overlay;
- Playback Schemes: Date, Time Interval, Event Triggering;
- View Snapshots;
- Save Snapshots;
- Motion Detection Alarm (Three motion-detection windows);
- Email Notificaction;
- Ajuste Flip e Mirror;
- Led's Indicadores;
- Link/Activity Administração e Firmware Updates;
- Management Site: Local e Remoto;
- IP View Lite SE (Visualiza somente 4 câmeras);
- Máx. de câmeras em uma tela: 16 (Com D-ViewCam);
- Botão Reset;
- Máximo de usuários acessando a câmera simultaneamente: 10;
- Alimentação Elétrica 5 VDC 2.5A;
- External universal power adapter (100 to 240 VAC, 50/60Hz) Temperatura de Operação;
- 0° C a 50° C Temperatura de Armazenamento;
- -25° C a 60° C Umidade;

- 5% a 95% Não Condensado Emissão (EMI);
- FCC Class B;
- CE Class B;
- C-Tick;
- IC Segurança;
- LVD (EN60950);
- Segurança - LVD (EN60950).

Após a escolha e aquisição da câmera deu-se início a instalação e configuração da mesma. Foi definido um ponto estratégico de instalação, o objetivo é proteger a câmera contra danos provocados pelos fenômenos naturais, como, sol, chuva e vento. Por este motivo optou-se por fixar a câmera no beiral da cobertura do prédio da Universidade, além de protegê-la contra danos, a posição em que está localizada possibilita a captura de imagens satisfatórias.

Para dar início a configuração da DCS-950 foi necessário primeiramente alimentá-la, conectando a mesma a uma fonte de energia e em seguida instalar o cabo de rede UTP. Em uma extremidade do cabo de rede é conectado a câmera e na outra extremidade conecta-se o roteador ou ponto de rede. O cabo de rede UTP é responsável por fazer a transferência de dados, neste caso os dados tratam-se das imagens capturadas pela câmera que serão disponibilizadas na rede de internet/intranet.

Inicialmente a câmera IP DCS-950 precisa ser configurada na rede local, a mesma possui um software de configuração que deve ser instalado em um computador. É indispensável que o computador e a câmera IP estejam conectados na mesma rede local para que o software possa fazer a identificação da câmera na rede. A instalação do software de configuração é simples e expressa, e após a conclusão basta iniciá-lo para dar início a configuração da câmera IP.

A Figura 42 ilustra a tela inicial do software de configuração da câmera IP D-Link DCS-950. Ao clicar no botão “Search” o software inicia a busca pela câmera na rede. Assim que o software identifica a câmera na rede local, este define o Endereço MAC e um Endereço de IP que será utilizado para acessar as imagens capturadas pela câmera após a finalização da configuração.

Figura 42 – Identificando a câmera DCS-950 na rede local



Fonte: Arquivo Pessoal

Figura 43 – Alterando as configurações da DCS-950



Fonte: Arquivo Pessoal

Conforme visto anteriormente, assim que o software identifica a câmera na rede local o mesmo já lhe atribui um Endereço MAC e um Endereço de IP, porém é possível alterar essas informações. Conforme a Figura 43 para alterar as configurações foi necessário escolher a primeira opção “Wizard” no menu do software.

Por questões de segurança o software exige que seja criado um usuário e senha para acesso as imagens capturas da câmera. Como este modelo de câmera é bastante utilizado em sistemas de monitoramento de segurança, essa necessidade de autenticação inicial é significativa para evitar que qualquer pessoa realize acesso remoto as imagens capturadas no monitoramento. Conforme a Figura 44 foi definido um nome de usuário “admin” e senha “admin”. Optou-se por esses nomes de credenciais, pois os mesmos são de fácil memorização, o que facilita o acesso na realização de testes e configurações posteriores.

Figura 44 – Criando as credenciais de acesso remoto à câmera DCS-950



The screenshot shows the D-Link Securicam configuration page. At the top, there is the D-Link logo with the tagline "Building Networks for People" and the Securicam logo. The main heading reads "Set up an Admin ID and Password to secure your camera. Click Next to continue." Below this, there are two input fields: "Admin ID:" with the value "admin" and "Password:" with "xxxxx". Underneath each field is a "Change" checkbox. Below the "Change" checkboxes are two columns of input fields: "New ID:" and "Reconfirm:" on the left, and "New Password:" and "Reconfirm:" on the right. At the bottom right, there are three buttons: "Back" (left arrow), "Next" (right arrow), and "Exit" (stop sign).

Fonte: Arquivo Pessoal

As configurações de rede foram alteradas com os parâmetros da rede local da Universidade. Conforme a Figura 45 ilustra, foi necessário alterar o Endereço de IP da câmera e o Gateway Padrão. Foi atribuída uma faixa de Endereço IP alta para evitar que o mesmo entre em conflito com os demais IP's na rede. O Gateway Padrão alterado é definido pela rede da Universidade. Os demais parâmetros foram mantidos.

Figura 45 – Definição das configurações de rede da câmera DCS-950



D-Link Building Networks for People

SECURICAM

### Set IP Address

IP Address	<input type="text" value="150.162.232.134"/>
Subnet Mask	<input type="text" value="255.255.255.0"/>
Default Gateway	<input type="text" value="150.162.232.254"/>
DNS 1	<input type="text" value="8.8.8.8"/>
DNS 2	<input type="text" value="8.8.4.4"/>

Back Next Exit

Fonte: Arquivo Pessoal

A Figura 46 mostra as configurações definidas para acessar remotamente as imagens da câmera. Foi selecionada a opção “Restart” para salvar as informações e concluir o processo de configuração.

Figura 46 – Finalização das configurações da câmera DSC-950



D-Link Building Networks for People

SECURICAM

Admin ID :	<input type="text" value="admin"/>
Password :	<input type="password" value="*****"/>
IP Address :	<input type="text" value="150.162.232.134"/>
Subnet :	<input type="text" value="255.255.255.0"/>
DNS 1 :	<input type="text" value="8.8.8.8"/>
DNS 2 :	<input type="text" value="8.8.4.4"/>

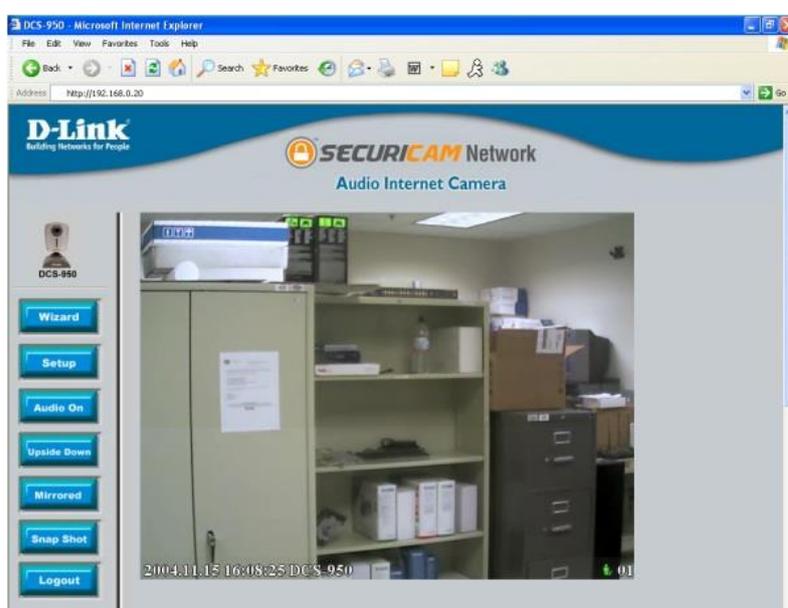
The Setup Wizard has completed. Click on **Back** to modify your settings. Click **Restart** to save your current settings and reboot the DCS-950 Series Camera.

Back Restart

Fonte: Arquivo Pessoal

A câmera IP foi configurada na rede local, sendo assim qualquer usuário que estiver conectado nesta rede ao informar o seguinte endereço: <http://150.162.232.134/> no navegador web, poderá monitorar as imagens que estão sendo capturadas pela câmera. O software da câmera IP DSC-950 possui um ambiente web que é vinculado ao endereço de IP definido na configuração da câmera. Neste ambiente web é possível habilitar ou modificar os recursos fornecidos, como, habilitar a captura de áudio, alterar a resolução e a qualidade da imagem, inverter ou espelhar a imagem, entre outros recursos disponíveis. A Figura 47 é uma imagem que serve de exemplo para ilustrar o ambiente web do software da câmera IP DSC-950.

Figura 47 – Ambiente web do software da câmera IP DCS-950



Fonte: Manual do usuário da câmera DCS-950

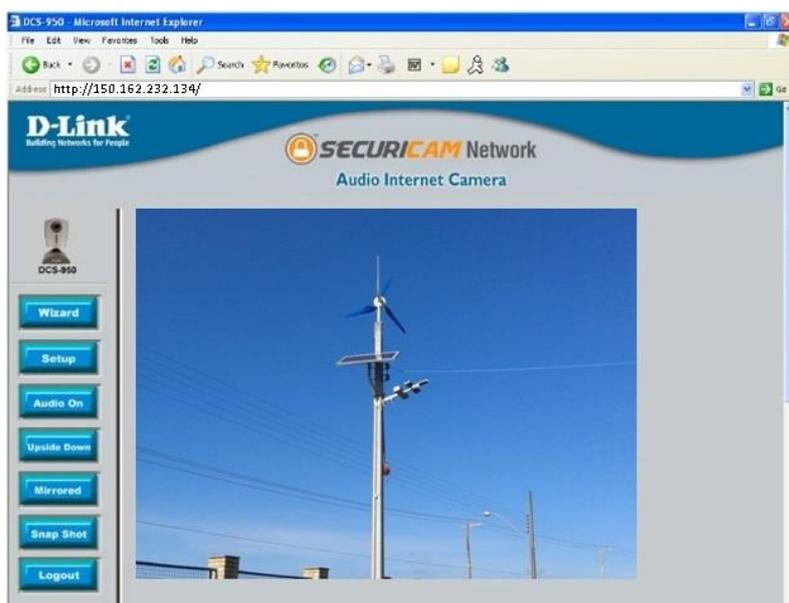
A câmera IP foi configurada para ser acessada remotamente na rede local, porém o objetivo deste trabalho é disponibilizar as imagens do sistema de iluminação com energia alternativa híbrida capturadas pela câmera IP em uma página web. De modo que qualquer pessoa que possuir acesso à internet e um navegador web poderá visualizar as imagens do sistema online disponíveis na página web, independente de qual rede de internet o usuário esteja conectado ou da sua localização geográfica.

Assim foi necessário realizar a configuração para visualização remota via internet. Para isso, primeiramente, foi preciso verificar no manual de configuração da câmera quais as portas utilizadas para transmitir tráfego de áudio e vídeo. A DCS-950

utiliza como padrão de fábrica a porta 80 para TCP e 5000, 5001 e 5100 para UDP. Após a identificação das portas utilizadas, foi necessário configurar o roteador de internet da UFSC, para que este efetue o redirecionamento dessas portas para o IP local da câmera.

O acesso e configuração no roteador da Universidade são feitos pela SETIC (Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação). A SETIC é um setor da Universidade localizado no campus de Florianópolis, sendo responsável por prestar suporte de TI à Universidade Federal de Santa Catarina. Foi aberto um chamado de atendimento com a SETIC para os responsáveis realizar o redirecionamento das portas para o IP local da câmera. Desta forma é possível acessar remotamente a câmera IP de qualquer rede de internet por um navegador web com o seguinte endereço: <http://150.162.232.134/>. A Figura 48 ilustra o acesso remoto à câmera IP realizado de uma rede externa.

Figura 48 – Acesso remoto à câmera IP DCS-950



Fonte: Arquivo Pessoal

## 4.2 Desenvolvimento do Ambiente Web para o Experimento Remoto

O sistema de aquisição de dados apresentado no tópico 4.1 é composto basicamente por um hardware e um software. O hardware é responsável por obter os sinais elétricos do sistema de controle embarcado e convertê-los em tensão para que o

Arduino possa receber em suas portas analógicas e digitais. O software realiza a conversão dos valores medidos em tensão para valores de estado real, estes valores são armazenados em variáveis que contêm os dados do estado do sistema híbrido sempre em tempo real. Neste trabalho um dos principais objetivos foi a disponibilização destas informações em uma página web.

Sendo assim, foi necessário o desenvolvimento do código de programação que realize a leitura destas variáveis provenientes do sistema de aquisição, mais especificamente do Arduino, a fim de disponibilizar essas informações em um ambiente web. Para tal, foi realizado um estudo para conhecimento das linguagens de programação web, plataformas de desenvolvimento, frameworks, dentre outros elementos necessários para o desenvolvimento do ambiente web. No ambiente web desenvolvido fez-se uso de várias ferramentas e linguagens que se tornaram indispensáveis à concretização da mesma.

Atualmente existem várias ferramentas disponíveis na internet, muitas delas gratuitas, outras não, o que acaba interferindo na escolha de uma em detrimento da outra. Optou-se por ferramentas compatíveis com o sistema operacional Windows de fácil entendimento e utilização. As escolhas poderiam certamente ter sido outras, que possivelmente o resultado seria algo semelhante, pois há um leque de ferramentas e linguagens que levam há um mesmo resultado. Porém a priori deu-se sobre recursos de fácil compreensão, que permitem moldagens posteriores sem muitos contratempos. A seguir abordam-se as ferramentas que foram escolhidas e utilizadas no decorrer do trabalho.

#### **4.2.1 Linguagens e Ferramentas Utilizadas para o Desenvolvimento da Página Web**

Para o desenvolvimento da página web foi necessário aprofundar conhecimentos em HTML, recorrendo-se em determinados aspectos a mais recente versão – HTML 5. A HTML5 foi a base da construção da página, e foi sendo adaptada para se tornar numa página dinâmica, conforme houveram necessidades.

Para a estilização da página web foi utilizada a CSS. Pois, esta linguagem foi criada com o único objetivo de cuidar da estilização de páginas web. Através da CSS foram definidas cores, tipografia, tamanho e disposição do conteúdo na tela. O uso da CSS garantiu que a página web tivesse uma formatação homogênea. Com o propósito

de facilitar ainda mais o desenvolvimento do layout da página, foi utilizado o *Framework CSS Zurb Foundation*, que facilitou bastante a criação da tabela, o posicionamento das imagens, e especialmente, por possibilitar que a página web se tornasse responsiva, ou seja, adaptável a diversos tamanhos de telas. Na Figura 49 é possível analisar a flexibilidade da página web, sendo visualizada através de um celular, tablet ou desktop.

Figura 49 – Responsividade da Página Web



Fonte: Arquivo Pessoal

Como dito anteriormente, precisava-se desenvolver um código de programação que capturasse as informações adquiridas do sistema híbrido através do sistema de aquisição de dados, essas informações estariam então disponíveis no Arduino, que é um dos componentes que integram o sistema de aquisição de dados. Para tal, foi utilizada a linguagem de script PHP. No código PHP desenvolvido, foram criadas variáveis que posteriormente receberam as informações do sistema híbrido, capturadas do Arduino, através do seu IP, pois o Arduino juntamente com o *Shield Ethernet* funciona como um microservidor web.

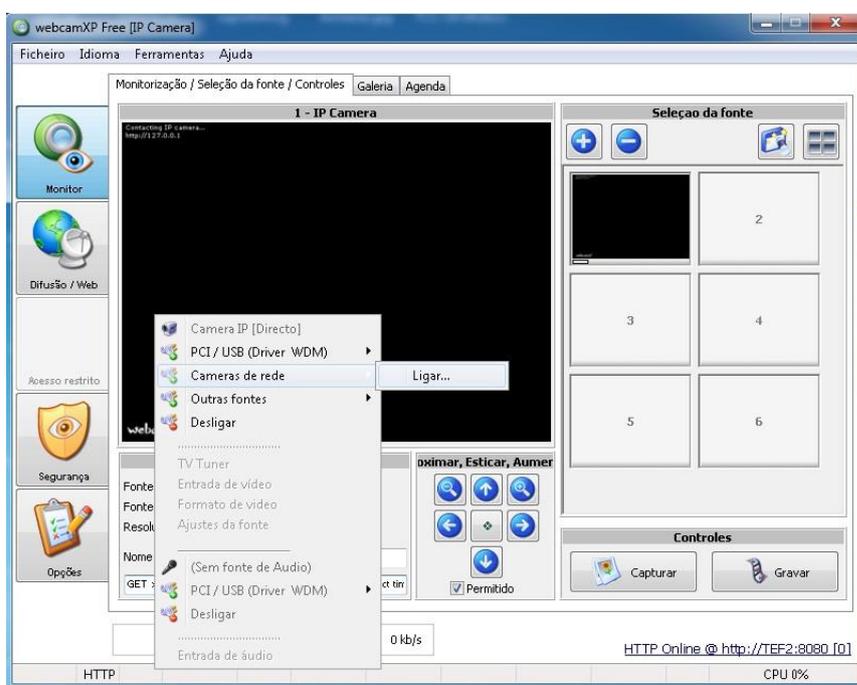
O código, em geral, da página foi desenvolvido utilizando-se o IDE Aptana Studio. Foi escolhido este IDE por ser voltado especificamente para desenvolvimento com suporte as linguagens PHP, HTML, CSS e JavaScript. Além de ser uma

ferramenta leve comparada a outros softwares de desenvolvimento, justamente por ser voltada especificamente para desenvolvimento web, evitando assim desperdício de recursos do programa e diminuindo a saturação da capacidade de processamento da máquina utilizada para o desenvolvimento. O Aptana Studio se mostrou uma ferramenta muito eficiente, que supriu as necessidades do projeto.

Como servidor local independente, escolheu-se usar o Wamp Server, de modo que se pudesse construir a página e visualizá-la sem a necessidade de estar hospedada, on-line, operação esta que foi efetuada mais tarde após a página web estar pronta.

Conforme apresentado no subtópico 4.1.6 foi utilizada uma câmera para captação das imagens do sistema híbrido na página Web, a câmera utilizada é uma câmera IP de modelo DCS-950 do fabricante D-Link. A mesma está conectada a um roteador e configurada na rede de internet para acesso remoto através de um navegador web, pois o software da câmera disponibiliza uma interface web, na qual é possível monitorar as imagens capturadas pela mesma. Porém, este trabalho propõe que as imagens do sistema híbrido, capturadas pela câmera, estejam disponíveis em uma página web juntamente com as demais variáveis de estado do sistema híbrido.

Figura 50 – Execução do assistente de configuração do software Webcam XP 5

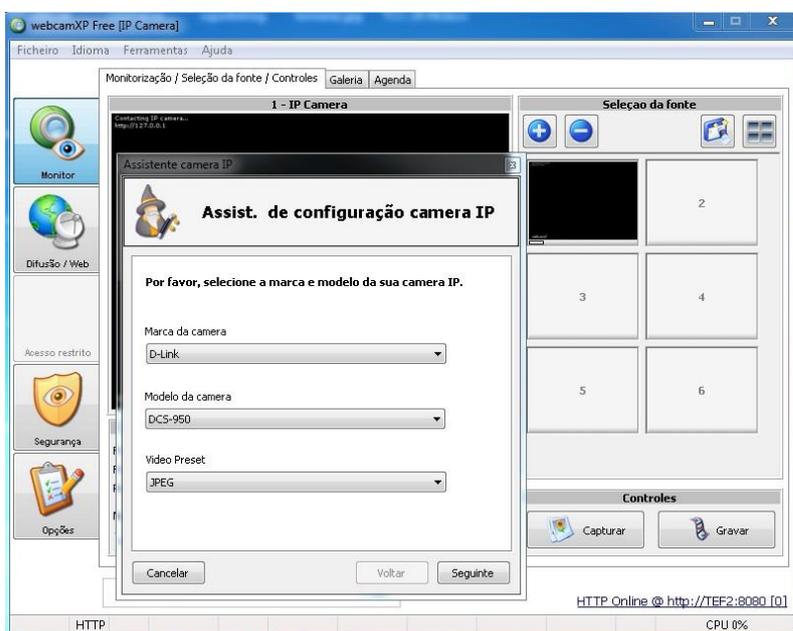


Fonte: Arquivo Pessoal

Deste modo, foi necessário buscar uma solução para a integração da câmera IP, e que esta fosse compatível com a página web desenvolvida. Optou-se pela utilização do software Webcam XP 5, esta é uma ferramenta que permite a configuração de câmeras web na rede. O usuário pode realizar o monitoramento remoto diretamente pelo software, além disso, a ferramenta indica um caminho de imagem com a extensão .jpg para que possa ser implementado em uma página web, assim, o desenvolvedor pode integrar as imagens capturadas pela câmera em um ambiente web próprio. Para realizar a integração foi necessário primeiramente configurar a câmera IP DCS-950 no software Webcam XP 5. A Figura 50 ilustra a execução do assistente de configuração do software Webcam XP 5 no modo de configuração para câmeras de rede (IP).

Após a execução do assistente de configuração da câmera de rede, foi necessário selecionar o fabricante da câmera, modelo e a extensão do arquivo, que neste modelo de câmera só permite a utilização da extensão .jpg. A Figura 51 ilustra a seleção das opções para a configuração.

Figura 51 – Seleção do fabricante, modelo e extensão de imagem para a configuração da DCS-950

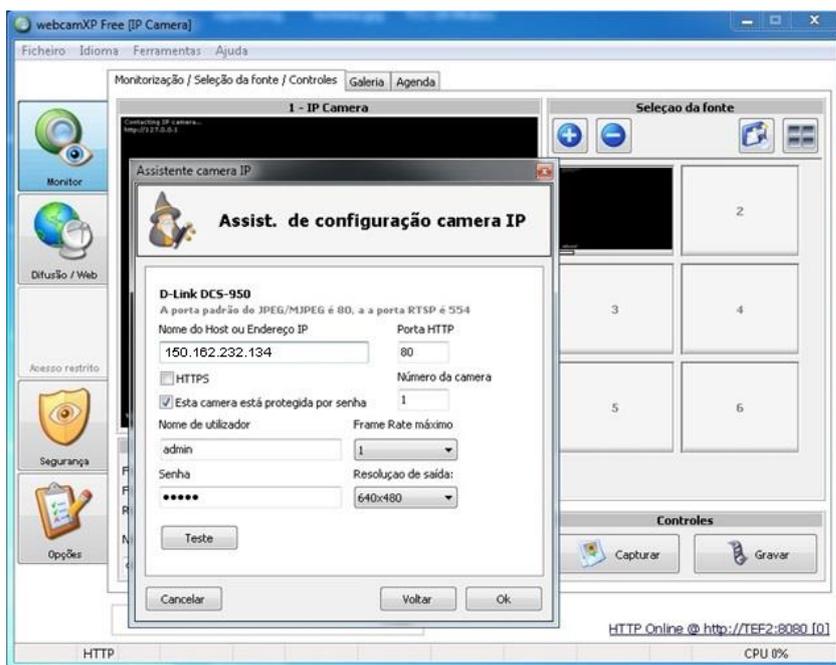


Fonte: Arquivo Pessoal

Para a finalização do assistente de configuração foi necessário informar alguns parâmetros, entre eles, acrescentar o IP da rede. Este IP é o mesmo que a SETIC da

UFSC configurou para o acesso remoto externo através do navegador web. Como a câmera está protegida com um usuário e senha para acesso remoto, foi necessário informar estas credenciais. A resolução foi mantida no padrão 640x480 pixels conforme configurado no software da câmera DCS-950. As demais informações não foram alteradas. Na Figura 52 é possível observar os parâmetros de configuração informados.

Figura 52 – Finalização do assistente de configuração do software Webcam XP 5



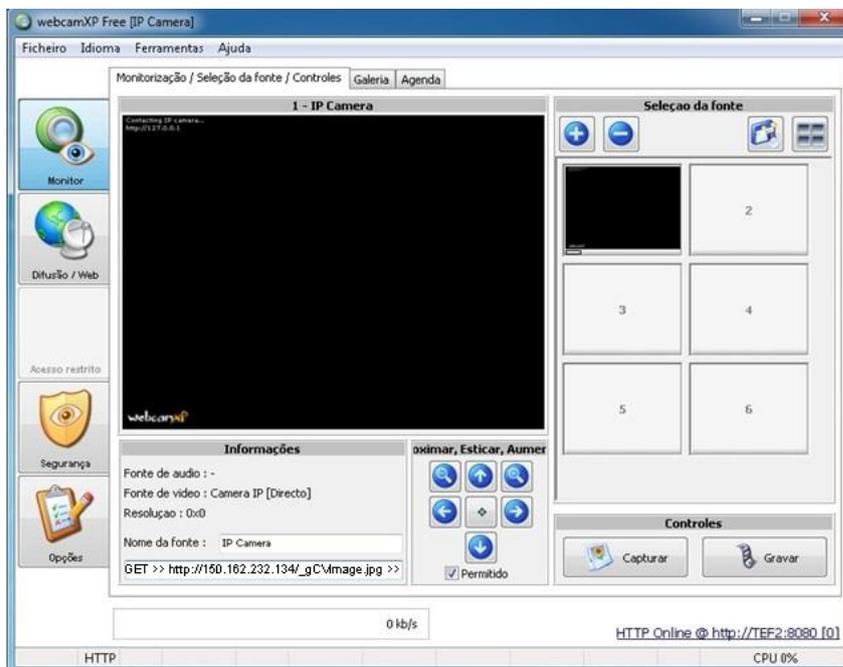
Fonte: Arquivo Pessoal

Após a câmera IP DCS-950 ter seus parâmetros definidos no software Webcam XP 5, é possível obter o caminho da imagem capturada pela câmera com a extensão .jpg, para que possa ser acrescentado no código da página web, a fim de possibilitar a aparição das imagens capturadas pela câmera na página. Este caminho indica o IP externo utilizado para acessar a câmera na rede e uma extensão.jpg. Na Figura 53 é possível observar no final da sub-janela “Informações” o caminho com extensão de imagem .jpg.

O caminho obtido no software Webcam XP 5 é descrito por: `http://150.162.232.134/_gCVimage.jpg`. Ao integrar o caminho na página web, percebeu-se a necessidade de implementar uma solução que faça a atualização da imagem. Assim foi desenvolvido um código de programação em JavaScript que realiza

o refresh da imagem obtida através da webcam, dispensando o carregamento manual da página por parte do usuário, essa atualização é feita a cada 15 segundos.

Figura 53 – Obtenção do caminho de imagem da câmera IP DCS-950 no software Webcam XP 5



Fonte: Arquivo Pessoal

#### 4.2.2 A Página Web

A página web desenvolvida está disponível para acesso no endereço web através da URL: [http://novo.more.ufsc.br/projeto\\_silah](http://novo.more.ufsc.br/projeto_silah). Este endereço foi fornecido pela UFSC e o código fonte da página web está hospedado em um servidor da própria Universidade. Ao acessar a página é possível observar o que foi desenvolvido na proposta deste trabalho, ou seja, as informações do sistema híbrido e imagens do mesmo captadas em tempo online, disponíveis virtualmente. Na figura 54, pode-se visualizar o template da página web assim que a mesma é acessada.

Na criação do layout da página foram utilizadas poucas cores, com o intuito de apresentar uma interface clara e limpa. Optou-se por este modelo de layout a fim de destacar as informações do sistema híbrido e as imagens capturadas do mesmo. O logo utilizado na página foi adaptado do projeto SILAH (Sistema de Integrado de Iluminação

por LED com Alimentação Híbrida), assim como o título. Conforme se pode observar na Figura 54 as informações do estado do sistema híbrido estão dispostas em uma tabela, o intuito foi melhorar a organização e facilitar a visualização destas.

Figura 54 – Template da Página Web

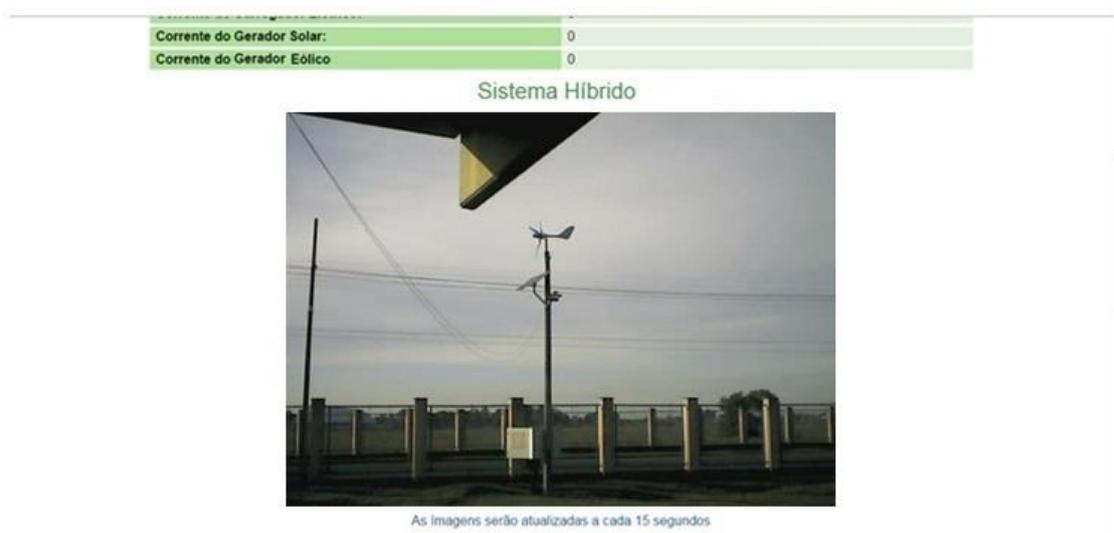


Fonte: Arquivo Pessoal.

A tabela expõe as informações, como, a data e a hora do acesso, e os dados do sistema híbrido naquele momento. Na segunda linha da tabela pode-se ver se a Luminária Led encontra-se acessa (1) ou apagada (0). Na terceira, se o carregador elétrico, que utiliza a energia da rede de abastecimento concessionário, está ligado (1) ou desligado (0). Na quarta linha, tem-se o valor da tensão da bateria, para monitoramento. Na quinta linha pode-se ver o valor da corrente do carregador, cuja fonte de energia é proveniente do sistema concessionário de abastecimento de energia elétrica; Na sexta linha tem-se o valor da corrente do gerador do sistema de energia eólico; Na sétima e última linha, mostra-se o valor da corrente do gerador do sistema de energia solar.

Na sequência, abaixo da tabela, conforme mostra a Figura 55, pode-se visualizar a imagem do experimento remoto em tempo online, essa imagem é atualizada na página web a cada 15 segundos.

Figura 55 – Sistema Híbrido em Tempo Online



Fonte: Arquivo Pessoal.

No rodapé da página, encontram-se as organizações parceiras do projeto SILAH, BAESA, Enercam, SATC, A Vero Domino, UFSC e P&D ANEEL.

Figura 56 – Monitoramento remoto durante o período diurno (presença de sol e vento)



Fonte: Arquivo Pessoal.

A Figura 56 é uma imagem capturada da página web durante o período diurno. Na página é possível observar os valores de estado do sistema de iluminação com energia híbrida e a imagem capturada do sistema através da câmera. Como a imagem foi

capturada no período diurno a Luminária LED está inoperante e o valor demonstrado na tabela para esta variável é zero. Naquele momento havia sol e vento, assim o carregador elétrico que recebe alimentação da rede concessionária de abastecimento de energia elétrica estava desligado, recebendo o valor zero em sua variável. A corrente deste carregador também é zero, pois o mesmo está desligado. A corrente do gerador solar e do gerador eólico é demonstrada, assim como a tensão da bateria que estava recebendo carga destas duas fontes de energia naquele momento. Conforme citado no capítulo 1 a tensão da bateria é medida em Volts e a corrente das fontes de energia em Ampères.

A Figura 57 também mostra uma imagem do monitoramento remoto do sistema de iluminação com energia híbrida durante o período diurno, porém no momento da captura havia apenas sol no ambiente. Assim, a bateria estava recebendo carga apenas do gerador solar, a corrente deste gerador e tensão da bateria é demonstrado na tabela e as demais variáveis do sistema permanecem com valores em zero por estarem ociosos.

Figura 57 – Monitoramento remoto durante o período diurno (ausência de vento)

Valores atuais do Experimento:

Data/Hora:	25 May 2016 13:40:44
LED (Lâmpada):	0
CHA (Carregador Elétrico):	0
Tensão da Bateria:	10.12
Corrente do Carregador Elétrico:	0
Corrente do Gerador Solar:	4.1
Corrente do Gerador Eólico:	0

Sistema Híbrido

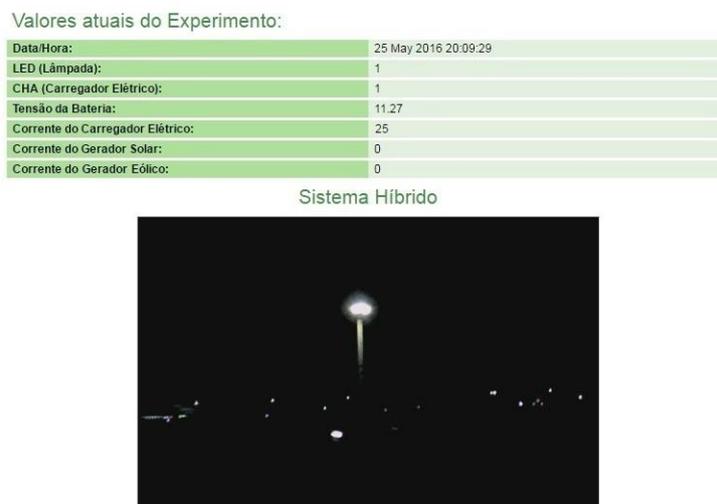


Fonte: Arquivo Pessoal.

A Figura 58 é uma imagem que foi capturada do monitoramento remoto do sistema de iluminação com energia híbrida durante o período noturno. É possível observar que a luminária LED está operando, desta forma a variável “LED (Lâmpada)” possui o valor 1. Como o monitoramento foi realizado no período noturno não havia a presença de sol no ambiente e nem de vento, assim os valores de corrente do gerador solar e gerador eólico permaneceram em zero. Desta forma a bateria está sendo

alimentada pelo carregador elétrico com a energia proveniente do sistema de abastecimento concessionário local. A corrente deste carregador pode ser observada na tabela assim como a tensão da bateria.

Figura – 58 Monitoramento remoto durante o período noturno



Arquivo Pessoal.

### 4.2.3 Estrutura que Compõe a Página Web

A página web é composta pelos seguintes ficheiros e pastas, que se estrutura da seguinte maneira (não estão contempladas as imagens):

- css
  - app.css
  - foundation.css
  - foundation.min.css
  - lightbox.css
  - normalize.css
- img
  - logo.jpg
  - luminaria.jpg
  - luminaria1.jpg
- js
  - foundation.min.js

- lightbox.min.js
- foundation
  - foundation.abide.js
- vendor
  - modernizr.js
  - jquery.js
- source
  - Data\_Logger\_UNO2
    - Data\_Logger\_UNO2.ino
  
- index.php

## 5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento de um experimento remoto baseado em um sistema de iluminação com energia alternativa híbrida eólico-fotovoltaico. No decorrer do desenvolvimento da fundamentação teórica apresentada nos primeiros capítulos, pôde-se perceber que os sistemas híbridos que utilizam da energia eólica e fotovoltaica são primordiais para a redução dos impactos ambientais causados pelo consumo excessivo de energia não renovável. As fontes geradoras desse tipo de sistema híbrido são limpas e abundantes em nosso meio.

Devido aos motivos citados acima, os estudos envolvendo sistemas híbridos de energias limpas e renováveis vem crescendo, porém existem poucos sistemas destes disponíveis nas instituições de ensino e Universidades para que os interessados possam ter acesso a fim de realizar suas experiências práticas. Uma alternativa para dar suporte a essa necessidade de experiência é a aplicação da experimentação remota nas instituições. A experimentação remota possui recursos que permitem o usuário ter acesso virtual a um experimento real, possibilitando ao mesmo, êxito na utilização tanto quanto no acesso real ao experimento.

Assim, com a base de conhecimento adquirido nos estudos realizados na área de sistemas híbridos e dos recursos necessários para a aplicação da experimentação remota, foi desenvolvido um experimento remoto baseado em um sistema de iluminação com energia alternativa híbrida eólico-fotovoltaico. O experimento remoto possui um sistema de aquisição de dados que foi desenvolvido para realizar a medição das informações de estado do sistema híbrido, informações estas que foram disponibilizadas na página web criada, a qual também disponibiliza imagens do sistema híbrido capturadas em tempo online, adquiridas através da câmera IP que fora instalada em local privilegiado.

A experimentação prática é um dos métodos de ensino que reforça o processo de aprendizagem, sendo fundamental nas diversas áreas técnicas, o experimento remoto possibilita que isso aconteça sem impedimentos físicos e geográficos. Sendo assim, a solução aqui descrita é uma grande aliada no que diz respeito a fins educacionais, pois o usuário pode suprir a necessidade de experiência sem estar fisicamente em um laboratório, tendo acesso a informações de estado do sistema híbrido, além de imagens

do mesmo disponíveis on-line que podem ser acessadas através de um ambiente web utilizando qualquer dispositivo que possua acesso à internet e um navegador web.

Um grande benefício que o experimento desenvolvido traz também, é que as informações disponíveis na página web podem ser utilizadas por outros estudantes e instituições que desejarem aperfeiçoar os resultados obtidos, tendo como base os resultados obtidos na utilização do experimento remoto do sistema de iluminação com energia alternativa híbrida eólico-fotovoltaico. Logo, constata-se que o custo de implantação do experimento remoto desenvolvido é relativamente baixo visto a porção de benefícios que o mesmo proporciona.

A partir dessa perspectiva, propõe-se como trabalhos futuros a otimização das possibilidades de utilização do sistema híbrido através do experimento remoto, como, a implementação de um banco de dados para armazenamento das informações acumuladas do estado do sistema híbrido. Atualmente, essas informações encontram-se disponíveis on-line em uma página web, e ter um histórico disponível a fim de consultar essas informações seria bastante relevante para realizar comparações e estudos. É interessante a disponibilização dessas informações em uma base pública para que os alunos e interessados possam usufruir e agregar às suas pesquisas.

## REFERÊNCIAS

ALVAREZ, Miguel Angel. **Introdução à Web**. 2004. Disponível em: <<http://www.criarweb.com/artigos/introducao-a-web.html>>. Acesso em: 05 fev. 2016.

ALVAREZ, Miguel Angel. **Uma IDE para Desenvolvimento de Aplicações Web, com Foco no Ajax e Web 2.0** Disponível em: <<http://www.criarweb.com/artigos/aptana-studio.html>>. Acesso em: 01 mar. 2016.

ALVES G., 2000. D2.1 **Review of the Foundation Technologies, Technical and Standard Issues**. Deliverable 2.1 PEARL, Setembro de 2000, 79 p.

ANACLETO, Joaquim Alberto da Costa. **Desenvolvimento de uma Aplicação Web para Dispositivos Móveis - Monitorização e Controlo de uma Rede de Digital Signage**. 2012. 112 f. Tese (Doutorado) - Curso de Engenharia Informática, Departamento de Informática, Universidade do Minho Escola de Engenharia, Minho, 2012. Disponível em: <<http://wiki.di.uminho.pt/twiki/pub/Research/APEX/Publications/tese.pdf>>. Acesso em: 01 mar. 2016.

APPCELERATOR (Ed.). **Site Oficial do Aptana**. Disponível em: <<http://www.aptana.com/>>. Acesso em: 01 mar. 2016.

ARDUINO.CC (Org.). **Arduino Ethernet Shield**. Disponível em: <<https://www.arduino.cc/en/Main/ArduinoEthernetShield>>. Acesso em: 06 out. 2015.

BARALE, Rafael Ferreira. **Desenvolvimento de um Sistema de Vendas na Web Utilizando JSP**. 2007. 78 f. TCC (Graduação) - Curso de Sistemas de Informação, Faculdade de Ciências Aplicadas de Minas, Uberlândia, 2007. Disponível em: <<http://www.si.lopesgazzani.com.br/TFC/monografias/TFC-BARALE.pdf>>. Acesso em: 05 fev. 2016.

BARBOSA, Claudomiro Fábio de Oliveira et al. **Situação da Geração Elétrica Através de Sistemas Híbridos no Estado do Pará e Perspectivas Frente à Universalização da Energia Elétrica**. In: ENCONTRO DE ENERGIA NO MEIO RURAL, 5., 2004, Campinas. **Proceedings online**. Campinas: ScieloProceedings, 2015. p. 1 - 10. Disponível em: <<http://www.proceedings.scielo.br/pdf/agrener/n5v2/116.pdf>>. Acesso em: 03 fev. 2015.

BARNABÉ, Grasiela. **Um Estudo Comparativo Entre as Linguagens de Programação PHP, ASP E JSP**. 2010. 108 f. Monografia (Especialização) - Curso de Didático-pedagógica, Centro Universitário Para O Desenvolvimento do Alto Vale do Itajaí, Rio do Sul, 2010. Disponível em: <<http://docplayer.com.br/477541-Um-estudo-comparativo-entre-as-linguagens-de-programacao-php-asp-e-jsp.html>>. Acesso em: 05 fev. 2016.

CAMARGO, Daniel. **Arduino: Introdução à Prototipagem Eletrônica**. 2013. Disponível em: <[https://www.academia.edu/4975136/Arduino\\_Introdu%C3%A7%C3%A3o\\_%C3%A0\\_Prototipagem\\_Eletr%C3%B4nica\\_Palestrante](https://www.academia.edu/4975136/Arduino_Introdu%C3%A7%C3%A3o_%C3%A0_Prototipagem_Eletr%C3%B4nica_Palestrante)>. Acesso em: 06 jun. 2015.

CARDIM, Ostend Batista. **Energia Fotovoltaica em Estruturas Híbridas**. 2012. 62 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012. Disponível em: <<http://monografias.poli.ufrj.br/monografias/monopoli10005163.pdf>>. Acesso em: 03 fev. 2015.

CARDOSO, Dayane Carvalho; TAKAHASHI, Eduardo Kojy. Experimentação remota em atividades de ensino formal: um estudo a partir de periódicos Qualis A. **Revista Brasileira de Pesquisa em Educação em Ciências**, ISSN 1806-5104 / e-ISSN 1984-2486, Uberlândia, v. 11, n. 3, p.185-208, out. 2011.

CEPEL (Org.). **Energia Solar: Princípios e Aplicações**. Disponível em: <<http://www.cresesb.cepel.br>>. Acesso em: 01 maio 2015.

CHAVES, Aline Martins; SILVA, Gabriel da. I Jornada Científica e VI FIPA do CEFET Bambuí 1, 2008, Bambuí-MG. **Proposta de uma Arquitetura e Software e Funcionalidades para Implementação de um Ambiente Integrado de Desenvolvimento para a Linguagem PHP**. Bambuí: Cefet, 2008. p. 1 - 5. Disponível em: <[http://www.cefetbambui.edu.br/str/artigos\\_aprovados/informatica/68-CO-5.pdf](http://www.cefetbambui.edu.br/str/artigos_aprovados/informatica/68-CO-5.pdf)>. Acesso em: 04 fev. 2016.

COLLATO, Uélyson. **Arduino– Plataforma de Prototipagem Eletrônica**. 2011. Disponível em: <<http://esseconhece.com.br/arduino-plataforma-de-prototipagem-eletronica/>>. Acesso em: 06 jun. 2015.

CROMAX ELETRÔNICA LTDA. **Power LED Branco 1W sem Dissipador**. Guarulhos - SP, 2010.

COSTA, Ricardo Jorge Guedes da Silva Nunes da. **Infra-Estrutura Laboratorial para Experimentação Remota**. 2003. 2013 f. Dissertação (Mestrado) - Curso de Engenharia

Electrotécnica e de Computadores, Universidade do Porto Faculdade de Engenharia - Feup, Porto, 2003. Disponível em: <<http://repositorio-aberto.up.pt/handle/10216/11198>>. Acesso em: 03 mar. 2015.

CUNHA, Alessandro F. **O que são Sistemas Embarcados?** Disponível em: <<http://www.techtraining.eng.br/files/uploads/2013/04/19/artigo-sist-emb.pdf>>. Acesso em: 19 fev. 2015.

DALI, Mehdi; BELHADJ, Jamel; ROBOAM, Xavier. **Hybrid Solar-Wind System With Battery Storage Operating in Grid-Connected and Standalone Mode: Control and Energy Management - Experimental Investigation.** Elsevier: Energy, França, p. 1-9. 08 abr. 2010. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360544210001234>>. Acesso em: 03 fev. 2015.

DUTRA, Leonardo. **Sensor de Corrente ACS712 30A.** 2013. Disponível em: <<https://dutrleo.wordpress.com/2013/01/29/sensor-de-corrente-ac712-30a/>>. Acesso em: 25 maio 2015.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em Ação.** São Paulo: Novatec, 2013. Camila Paduan. Disponível em: <<http://www.novateceditora.com.br/livros/arduino-em-acao/capitulo9788575223734.pdf>>. Acesso em: 07 set. 2015.

DZENDZIK, Isolete Teresinha. **Processo de Desenvolvimento de Web Sites com Recursos da UML.** 2005. 181 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2004. Disponível em: <<http://mtc-m16.sid.inpe.br/col/sid.inpe.br/jeferson/2004/12.10.09.14/doc/publicacao.pdf>>. Acesso em: 04 fev. 2016.

EXCELPOINT. **LED Lighting: Products & Solutions.** Tailândia, 2011.

FERREIRA, Abel António de Azevedo. **Sistema de Produção de Energia Eólica.** 2011. 165 f. Dissertação (Mestrado) - Curso de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, 2011. Disponível em: <[http://paginas.fe.up.pt/~ee06206/images/docs2/MSc\\_AbelFerreira.pdf](http://paginas.fe.up.pt/~ee06206/images/docs2/MSc_AbelFerreira.pdf)>. Acesso em: 16 fev. 2015.

FARIA, Fernanda B. et al. **Evolução e Principais Características do IDE Eclipse.** Catalão: Universidade Federal de Goiás, 2010. 8 p. Disponível em:

<[http://www.enacomp.com.br/2010/cd/artigos/completos/enacomp2010\\_23.pdf](http://www.enacomp.com.br/2010/cd/artigos/completos/enacomp2010_23.pdf)>.  
Acesso em: 01 mar. 2016.

FOREFRONT (Org.). **The Absolute Beginner's Guide to Arduino**. 2013. Disponível em: <<http://forefront.io/a/beginners-guide-to-arduino/>>. Acesso em: 12 out. 2015.

GILLET D., Salzmann C. e Crisalle O., 1999. **Hands-on Laboratory Sessions in Distance Learning**. Swiss Federal Institute of Technology, Lausanne, Suíça, Agosto de 1999, 15 p.

GOMES, Evandro Luís Brandão; TAVARES, Luís Antônio. **Uma Solução com Arduino para Controlar e Monitorar Processos Industriais**. **Revista Controle & Instrumentação: Elétrica e Automação**, São Paulo, v. 1, n. 185, p.77-79, 10 abr. 2013. Mensal.

GRUPO DE TRABALHO DE ENERGIA SOLAR - GTES (ORG.). **Manual de Engenharia para Sistemas Fotovoltaicos**. Rio de Janeiro: Cepel - Cresesb, 2014. 530 p. (1). Disponível em: <[http://www.cresesb.cepel.br/publicacoes/download/Manual\\_de\\_Engenharia\\_FV\\_2014.pdf](http://www.cresesb.cepel.br/publicacoes/download/Manual_de_Engenharia_FV_2014.pdf)>. Acesso em: 04 fev. 2015.

HANSON, Bem. “ReLOAD: Real laboratories Operated At Distance”. University of Leeds, 2009. LEWIS, N. et al. “A Distance Measurement Platform Dedicated to Electrical Engineering”. Bordeaux University, 2009.

HEIN, Werner. **RaspberryPi Aplicado a Projetos do Mundo Real**. 2013. Revista Linux Magazine, 2013. Disponível em: <[http://www.linuxnewmedia.com.br/images/uploads/pdf\\_aberto/LM\\_100\\_60\\_65\\_06\\_tut\\_raspberry\\_pi.pdf](http://www.linuxnewmedia.com.br/images/uploads/pdf_aberto/LM_100_60_65_06_tut_raspberry_pi.pdf)>. Acesso em: 05 set. 2015.

HS BENEFICIAMENTO DE ALUMÍNIO LTDA. **Catálogo HS Dissipadores**. São Paulo, 2008/2009.

INC, Southwest Windpower (Org.). **Air Breeze Owner's Manual**. Flagstaff: Southwest Windpower, 2008. 32 p. Disponível em: <[http://www.reps.fi/datasheetsandmanuals/air\\_breeze\\_manual.pdf](http://www.reps.fi/datasheetsandmanuals/air_breeze_manual.pdf)>. Acesso em: 14 jul. 2015.

JANEIRO, Luiz. **Instalando O Ubuntu Na BeagleBone Black Em SD Card**. 2015. Disponível em: <<http://www.decom.ufop.br/imobilis/instalando-o-ubuntu-na-beaglebone-black-em-sd-card/>>. Acesso em: 23 abr. 2015.

JOHNSTON, W.; AGARWAL, D.. **The Virtual Laboratory**: Using Networks to enable Widely Distributed Collaboratory Science. Ernest Orlando Lawrence Berkeley National Laboratory, University of California, 1995.

LAKATOS, Eva Maria. **Fundamentos de Metodologia Científica**, 7ª ed. São Paulo: Atlas, 2010.

LEHRBAUM, Rick. **Intel unveils second generation Galileo hacker SBC**. 2014. Disponível em: <<http://hackerboards.com/intel-unveils-second-generation-galileo-hacker-sbc/>>. Acesso em: 11 abr. 2015.

LEMOS, Manoel. **Conheça os Shields e Incremente seu Arduino com Eles**. 2013. Disponível em: <<http://blog.fazedores.com/conheca-os-shields-e-incremente-seu-arduino-com-eles/>>. Acesso em: 19 set. 2015.

LIMEIRA, José Luiz Silveira. **Utilização de AJAX no desenvolvimento de sistemas Web**. 2006. 44 f. Monografia (Especialização) - Curso de Web e Sistemas de Informação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006. Disponível em: <[http://www.limeira.eti.br/monografia\\_ajax.pdf](http://www.limeira.eti.br/monografia_ajax.pdf)>. Acesso em: 01 mar. 2016.

LOPES, Camilo. **Entendendo Servlet**. 2008. Disponível em: <<https://camilolopes.wordpress.com/2008/06/09/entendendo-servlet/>>. Acesso em: 06 mar. 2015.

LÜDKE, Menga; ANDRÉ, Marli. **Pesquisa em educação: abordagens qualitativas**. São Paulo: EPU, 1986.

MACHADO, Rafael Gomes. **Rafael Gomes Machado Sistema de Informação Baseado numa Intranet para a Gestão Pedagógica de uma Escola do Ensino Secundário**. 1999. 241 f. Dissertação (Mestrado) - Curso de Gestão de Informação, Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, 1999. Disponível em: <<http://www.esev.ipv.pt/tear/Recursos/17/Dissertacao-Rafael.PDF>>. Acesso em: 04 fev. 2016.

MARCELINO, Roderval. **Sistema Integrado de iluminação por LED com Alimentação Híbrida: Sistema Integrado de Iluminação Baseado em LEDs de Alto Brilho com Alimentação por Sistema Híbrido: Placas Fotovoltaicas, Miniaerogeradores e Energia Elétrica da Rede de Distribuição.** Araranguá: Universidade Federal de Santa Catarina, 2013. 168 p.

MARCHIORO, Eliete. **Um Estudo Sobre Rejuvenescimento de Software em Servidores Web Apache.** 2003. 82 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2003. Disponível em:

<<https://repositorio.ufsc.br/bitstream/handle/123456789/84584/201481.pdf?sequence=1>>. Acesso em: 04 fev. 2016.

MARTINEZ, Pedro. **Wattímetro.** 2013. Relatório Final da Disciplina de Instrumentação Eletrônica - Curso de Engenharia Eletrônica, Universidade Católica de Pelotas, Centro Politécnico, Pelotas, Rio Grande do Sul, 2013. Disponível em: <[http://olaria.ucpel.tche.br/ie2013/lib/exe/fetch.php?media=relatorio\\_final\\_wattmetro.pdf](http://olaria.ucpel.tche.br/ie2013/lib/exe/fetch.php?media=relatorio_final_wattmetro.pdf)> Acesso em: 25 maio. 2015.

MICROSOFT (Org.). **Site Oficial do Visual Studio.** Disponível em: <<https://www.visualstudio.com/>>. Acesso em: 01 mar. 2016.

MONTEIRO, Rogério Rodrigues; CABREIRA, Maria de Fátima R. Raia. **Análise Técnica da Implantação de um Sistema Híbrido Eólico-Solar para Alimentar ERBSem Lugares Isolados: O CASO DA ESTAÇÃO ANTONINA/PR.** 2007. 7 f. TCC (Graduação) - Curso de Engenharia, Universidade Tecnológica Federal do Paraná, Curitiba, 2007. Disponível em: <[http://www.daelt.ct.utfpr.edu.br/engenharia/tcc/artigo\\_eolico-solar\\_2007.pdf](http://www.daelt.ct.utfpr.edu.br/engenharia/tcc/artigo_eolico-solar_2007.pdf)>. Acesso em: 02 fev. 2015.

NASCIMENTO, Cássio Araújo do. **Princípio de Funcionamento da Célula Fotovoltaica.** 2004. 21 f. Monografia (Especialização) - Curso de Fontes Alternativas de Energia, Universidade Federal de Lavras, Lavras, 2004. Disponível em: <[http://www.solenerg.com.br/files/monografia\\_cassio.pdf](http://www.solenerg.com.br/files/monografia_cassio.pdf)>. Acesso em: 18 fev. 2015.

OGIBOSKI, Luciano. **Laboratório Remoto Baseado em Software Livre para Realização de Experimentos Didáticos.** 2007. 89 f. Dissertação (Mestrado) - Curso de Engenharia - Sistemas Eletrônicos, Universidade de São Paulo, São Paulo, 2007. Disponível em: <[www.teses.usp.br/teses/.../3/3142/.../Luciano\\_Ogiboski\\_DissertacaoOK.pdf](http://www.teses.usp.br/teses/.../3/3142/.../Luciano_Ogiboski_DissertacaoOK.pdf)>. Acesso em: 06 mar. 2015.

OLIVEIRA, Ederson dos Santos Cordeiro de; PIPERNO, Tiago Bonetti; GERMANO, Ricardo. **Padrões para o Desenvolvimento na Web**. 2014. 5 f. Artigo, Universidade Paranaense - Unipar, Paranavaí, 2014. Disponível em: <[http://web.unipar.br/~seinpar/2014/artigos/graduacao/Ederson dos Santos.pdf](http://web.unipar.br/~seinpar/2014/artigos/graduacao/Ederson_dos_Santos.pdf)>. Acesso em: 01 mar. 2016.

ORACLE (Org.). **Site Oficial do NetBeans**. Disponível em: <<https://netbeans.org/>>. Acesso em: 01 mar. 2016.

PALADINI, Suenoni. **Experimentação Remota como Suporte a Ambientes de Aprendizagem de Física**. 2008. 89 f. Dissertação (Mestrado) - Curso de Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis, 2008. Disponível em: <<http://btd.egc.ufsc.br/wp-content/uploads/2010/06/Suenoni-Paladini.pdf>>. Acesso em: 04 mar. 2015.

PHOCOS (Alemanha). CML-V2. Magirus-deutz-straße, 2009. 125 p.

PHOENIX CONTACT (Brasil). **Extract from the Online Catalog**. Branca, 2004.

REIS, Mônica M.; OLIVEIRA JUNIOR, Demercil S.; CARVALHO, Paulo C. M. de. **Estudo de Viabilidade Econômica de Geradores Eólicos de Pequeno Porte no Modo Autônomo**. Mossoró, p.1-4, [s. d] Disponível em: <[http://www2.ufersa.edu.br/portal/view/uploads/setores/178/arquivos/Fontes Alternativas/analise\\_custo\\_ger\\_eolico\\_1kW.pdf](http://www2.ufersa.edu.br/portal/view/uploads/setores/178/arquivos/Fontes_Alternativas/analise_custo_ger_eolico_1kW.pdf)>. Acesso em: 17 fev. 2015.

REIS, Renata Tibiriçá dos. **Desenvolvimento Web com o Uso de Padrões: Tecnologias e Tendências**. 2007. 74 f. Monografia (Especialização) - Curso de Ciência da Computação, Ciência da Computação, Universidade Federal de Juiz de Fora Instituto de Ciências Exatas, Juiz de Fora, 2007. Disponível em: <<http://www.renatatr.com/monografia/monografia.pdf>>. Acesso em: 01 mar. 2016.

ROCHA, Roberto Oliveira. **WebRTC - Evolução na Web**. 2014. 111 f. Dissertação (Mestrado) - Curso de Sistemas e Tecnologias de Informação Para As Organizações, Escola Superior de Tecnologia e Gestão de Viseu, Viseu, 2014. Disponível em: <[http://repositorio.ipv.pt/bitstream/10400.19/2554/1/ROBERTO ROCHA 10286 - Dissertação MSTIO 2012-2014.pdf](http://repositorio.ipv.pt/bitstream/10400.19/2554/1/ROBERTO_ROCHA_10286_-_Dissertação_MSTIO_2012-2014.pdf)>. Acesso em: 01 mar. 2016.

RODRIGUES, Samuel da Costa. **Estudo e Implementação de Interfaces Web em HTML5**. 2014. 91 f. Dissertação (Mestrado) - Curso de Engenharia de Informática, Departamento de Informática, Universidade do Minho Escola de Engenharia, Minho, 2014. Disponível em:

<[https://repositorium.sdum.uminho.pt/bitstream/1822/36800/1/eeum\\_di\\_dissertacao\\_pg20200.pdf](https://repositorium.sdum.uminho.pt/bitstream/1822/36800/1/eeum_di_dissertacao_pg20200.pdf)>. Acesso em: 01 mar. 2016.

SANTOS, Diego Lima. **LABEXP – Laboratório de Experimentação Remota em Tempo Real**. 2009. 115 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Federal do Pará, Belém, 2009. Disponível em: <[http://www.repositorio.ufpa.br/jspui/bitstream/2011/3413/1/Dissertacao\\_LaboratorioExperimentacaoRemota.pdf](http://www.repositorio.ufpa.br/jspui/bitstream/2011/3413/1/Dissertacao_LaboratorioExperimentacaoRemota.pdf)>. Acesso em: 05 mar. 2015.

SILVA, Juarez Bento da. **A Utilização da Experimentação Remota como Suporte Para Ambientes Colaborativos De Aprendizagem**. 2006. 196 f. Tese (Doutorado) - Curso de Engenharia de Gestão do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis, 2006. Disponível em: <<http://btd.egc.ufsc.br/?p=512>>. Acesso em: 19 fev. 2015.

SILVA, Juarez Bento da. **A Utilização da Experimentação Remota como Suporte à Ambientes Colaborativos de Aprendizagem**. Florianópolis 2007.

SILVA, Juarez Bento da et al. **Uso de Dispositivos Móveis para Acesso a Experimentos Remotos na Educação Básica**. *VAEP-RITA*, Araranguá, v. 1, n. 2, p.129-134, 2 maio 2013. Disponível em: <<http://www.rexlab.net/downloads/artigos/21.pdf>>. Acesso em: 23 fev. 2015.

SILVA, Maurício Samy. **JQuery Mobile: Desenvolva Aplicações Web Para Dispositivos Móveis com HTML5, CSS3, AJAX, jQuery e jQuery UI**. São Paulo: Novatec, 2012. 352 p.

SHOP BY (Ed.). **IP-камера D-Link DCS-950**. Disponível em: <[http://shop.by/dom\\_i\\_dacha/bezopasnost/kameri\\_videonablyudeniya/d\\_link\\_dcs\\_950/](http://shop.by/dom_i_dacha/bezopasnost/kameri_videonablyudeniya/d_link_dcs_950/)>. Acesso em: 02 mar. 2016.

SOLARWORLD (Estados Unidos da América). Sunmodule SW85 poly RNA. Hillsboro, 2011. 2 p.

SOUSA, Alisson Rodrigues de. **Aplicação de Padrões de Projeto com a Linguagem PHP**. 2009. 68 f. Monografia (Especialização) - Curso de Análise de Sistemas, Ciências da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2009. Disponível em: <<http://www.bibliotecadigital.ufmg.br/dspace/bitstream/handle/1843/BUOS-94NMN6/alissonrodrigues.pdf?sequence=1>>. Acesso em: 05 fev. 2016.

SOUSA, N.; Alves, A.R.; Gericota, M.G., 2010. “**Na integrated reusable remote laboratory to complement electronics teaching**”, IEEE Transactions on learning technologies, Vol. 3, Nº 3, July - September 2010, pp. 265-271.

SOUSA, Nuno Miguel Moreira de. **Bancada Laboratorial Remota para o Ensino da Eletrotécnica**. 2008. 75 f. Tese (Doutorado) - Curso de Engenharia Electrotécnica e de Computadores, Instituto Superior de Engenharia do Porto, Porto, 2008.

SOUZA, Anderson R. de et al. **A Placa Arduino: uma Opção de Baixo Custo para Experiências de Física Assistidas pelo PC**. *Revista Brasileira de Ensino de Física*, Rio de Janeiro, v. 33, n. 1702, p.02-05, 21 mar. 2011. Disponível em: <<http://www.sbfisica.org.br/rbef/pdf/331702.pdf>>. Acesso em: 04 fev. 2016.

SOUZA, Fábio. **Arduino UNO**. 2013. Disponível em: <<http://www.embarcados.com.br/arduino-uno>>. Acesso em: 17 set. 2015.

SUEIRO, Diego. **Apresentação Beaglebone Black - O que Existe de Open por aí?**2014. Disponível em: <<http://www.embarcados.com.br/apresentacao-beaglebone-black-o-que-existe-de-open-por-ai-7masters-openhardware/>>. Acesso em: 30 ago. 2015.

TEIXEIRA, Pedro José Lima. **Construção de Interfaces em Flex para Sistemas de Experimentação Remota**. 2010. 93 f. Dissertação (Mestrado) - Curso de Engenharia Informática, Instituto Superior de Engenharia do Porto, Porto, 2010. Disponível em: <[http://recipp.ipp.pt/bitstream/10400.22/2717/1/DM\\_PedroTeixeira\\_2010\\_MEI.pdf](http://recipp.ipp.pt/bitstream/10400.22/2717/1/DM_PedroTeixeira_2010_MEI.pdf)>. Acesso em: 25 fev. 2015.

VIRTUAL SCAPE (Org.). **Active Server Pages Hosting**. 2002. Disponível em: <<http://www.virtualscape.com/services/asp.html>>. Acesso em: 12 fev. 2015.

TELEPAC (Org.). **Glossário**. 2003. Disponível em: <[https://www.ptempresas.pt/pme/internet/internet-fixa/netplusglossario/glossario\\_W.html](https://www.ptempresas.pt/pme/internet/internet-fixa/netplusglossario/glossario_W.html)>. Acesso em: 13 jul. 2015.

ZANOTTO, Nader. **Protótipo de Laboratório de Experimentação Remota Multiplataforma Cliente**. 2006. 61 f. TCC (Graduação) - Curso de Ciências da Computação, Universidade Regional de Blumenau, Blumenau, 2006. Disponível em: <<http://dsc.inf.furb.br/tcc/index.php?cd=9&tcc=977>>. Acesso em: 17 fev. 2015.

ZEILMANN, Rafael Pereira. **Uma Estratégia para Controle e Supervisão de Processos Industriais Via Internet**. 2002. 159 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002. Disponível em: <[www.ufrgs.br/ppgee/tese-eng-0362006.pdf.gz](http://www.ufrgs.br/ppgee/tese-eng-0362006.pdf.gz)>. Acesso em: 05 fev. 2016.