



**UNIVERSIDADE FEDERAL DA GRANDE DOURADOS  
FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA  
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

JOÃO VINICIUS OTTONI NORMANDO  
MILTON PINTO DE ALMEIDA CASTRO

**DESENVOLVIMENTO DE UM SISTEMA DE AQUISIÇÃO DE DADOS E  
MONITORAMENTO REMOTO PARA SISTEMAS DE BOMBEAMENTO DE ÁGUA  
FOTOVOLTAICO**

DOURADOS  
2018

JOÃO VINICIUS OTTONI NORMANDO  
MILTON PINTO DE ALMEIDA CASTRO

**DESENVOLVIMENTO DE UM SISTEMA DE AQUISIÇÃO DE DADOS E  
MONITORAMENTO REMOTO PARA SISTEMAS DE BOMBEAMENTO DE ÁGUA  
FOTOVOLTAICO**

Monografia apresentada na Faculdade de Ciências Exatas e Tecnologia da Universidade Federal da Grande Dourados, como parte dos requisitos necessários para a obtenção do título de Bacharel em Engenharia de Computação.  
Área de concentração: Aquisição de Dados

Orientador: Prof. Me. Ricardo Camparim

DOURADOS  
2018

**Dados Internacionais de Catalogação na Publicação (CIP).**

C355d Castro, Milton Pinto De Almeida  
DESENVOLVIMENTO DE UM SISTEMA DE AQUISIÇÃO DE  
DADOS E MONITORAMENTO REMOTO PARA SISTEMAS DE  
BOMBEAMENTO DE ÁGUA FOTOVOLTAICO / Milton Pinto De Almeida  
Castro, João Vinicius Ottoni Normando -- Dourados: UFGD, 2018.  
65f. : il. ; 30 cm.

Orientador: Ricardo Camparim

TCC (Graduação em Engenharia de Computação) -  
Universidade Federal da Grande Dourados  
Inclui bibliografia

1. Aquisição. 2. Monitoramento. 3. Dados. 4. Bombeamento. 5.  
Fotovoltaico. I João Vinicius Ottoni Normando II. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelos autores.

**©Direitos reservados. Permitido a reprodução parcial desde que citada a fonte.**

JOÃO VINICIUS OTTONI NORMANDO  
MILTON PINTO DE ALMEIDA CASTRO

**DESENVOLVIMENTO DE UM SISTEMA DE AQUISIÇÃO DE DADOS E  
MONITORAMENTO REMOTO PARA SISTEMAS DE BOMBEAMENTO DE ÁGUA  
FOTOVOLTAICO**

Monografia apresentada na Faculdade de Ciências Exatas e Tecnologia da Universidade Federal da Grande Dourados, como parte dos requisitos necessários para a obtenção do título de Bacharel em Engenharia de Computação. Área de concentração: Aquisição de Dados

Aprovada em: \_\_\_/\_\_\_/\_\_\_

**BANCA EXAMINADORA**

---

Prof. Me. Ricardo Camparim (Orientador)  
Universidade Federal da Grande Dourados (UFGD)

---

Prof. Dr. Orlando Moreira Júnior  
Universidade Federal da Grande Dourados (UFGD)

---

Prof. Dr. Rodrigo Yoshikawa Oeiras  
Universidade Federal da Grande Dourados (UFGD)

## **RESUMO**

O presente trabalho tem como objetivo principal desenvolver e implantar um sistema capaz de obter dados como tensão, corrente elétrica, potência, temperatura e vazão. Além disso armazenar e disponibilizar os dados local e remotamente. Utilizando conhecimentos adquiridos em microcontroladores para automação industrial (*hardware*) e programação voltada para aplicações em sistemas embarcados (*software*). Neste trabalho é utilizado um sistema de bombeamento fotovoltaico localizado no Campus da FAEN da Universidade Federal da Grande Dourados (UFGD) em Dourados - MS / Brasil. Latitude: 22° 11' 56" Sul. Longitude: 54° 55' 56" Oeste. O trabalho apresenta dados obtidos através de dois microcontroladores. Estes dados são oriundos de divisores de tensão, sensores de corrente elétrica, sensores de temperatura e um medidor de vazão.

**Palavras-chave:** Aquisição, Monitoramento, Dados, Fotovoltaico, Bombeamento.

## **ABSTRACT**

The main objective of this work is to develop and implement a system capable of obtaining data such as voltage, current, power, temperature and flow. Also stored and make the data available locally and remotely. Using the knowledge acquired in microcontrollers for industrial automation (hardware) and programming directed to applications in embedded systems (software). This work uses a photovoltaic pumping system located at FAEN Campus of the Federal University of Grande Dourados (UFGD) in Dourados - MS / Brazil. Latitude: 22 ° 11 '56" South. Longitude: 54° 55 '56" West. The work presents data obtained through two microcontrollers. These are the voltage divider diodes, the electric current sensors, the temperature sensors and a flow meter.

**Keywords:** Acquisition, Monitoring, Data, Photovoltaic, Pumping.

## LISTA DE FIGURAS

Figura 1 – Diagrama de um sistema autônomo.....	30
Figura 2 – Diagrama de um sistema de bombeamento de água.....	31
Figura 3 – Anauger 175Wp (extremidades) e CanadianSolar 260Wp (centro).....	32
Figura 4 – Controlador de Bomba Anauger ( <i>Driver 100</i> ).....	33
Figura 5 – Controlador de carga EPEver <i>Tracer 2210A MPPT</i> .....	34
Figura 6 – Inversor de onda senoidal modificada Hayonic 400W.....	35
Figura 7 – Baterias estacionárias Heliar DF2500 em paralelo .....	36
Figura 8 – Medidor de vazão eletromagnético incontrol PRO1000 .....	38
Figura 9 – Bomba imersa vibratória Anauger R100 .....	39
Figura 10 – Conectores MC4 .....	39
Figura 11 – Unidade de bombeamento de água FV (FAEN-UFGD) .....	40
Figura 12 – Diagrama completo da unidade FV (FAEN-UFGD).....	41
Figura 13 – Sensores de temperatura Dallas DS18B20.....	43
Figura 14 – A imagem abaixo exemplifica o processo descrito acima .....	44
Figura 15 – Sensor de tensão $V_{cc} < 25V$ (divisor de tensão).....	45
Figura 16 – Aplicativo ElectroDroid .....	45
Figura 17 – Aplicativo ElectroDroid .....	46
Figura 18 – Extensor com sensor de tensão destacado .....	47
Figura 19 – Validação e calibração dos sensores com multímetros.....	47
Figura 20 – Leituras de tensão obtidas pelo Arduino Mega 2560 .....	48
Figura 21 – Sensor de Corrente Allegro ACS712 20A .....	49
Figura 22 – Imagem com sensor de corrente destacado .....	49
Figura 23 – Fontes de corrente em paralelo .....	50
Figura 24 – Leituras de corrente obtidas pelo Arduino Mega 2560 .....	50
Figura 25 – Fontes de corrente totalizando 9A.....	51
Figura 26 – Exemplo da saída de tensão: vazão igual a 4mA (0m3h) .....	52
Figura 27 – Exemplo da saída de tensão: vazão igual a 20mA (10m3h) .....	52
Figura 28 – Teste de comparação/equivalencia do sinal de 4 – 20mA .....	53
Figura 29 – Transceptor nRF24L01 +PA +LNA 2,4GHz.....	54
Figura 30 – Módulos transmissor e receptor RF 433MHz .....	55
Figura 31 – Módulo <i>Datalogger</i> + SD + RTC DS1307.....	56

Figura 32 – Ethernet Shield W5100 .....	57
Figura 33 – Arduino Mega 2560 .....	58
Figura 34 – Arquitetura do sistema de monitoramento: transmissor .....	59
Figura 35 – Arquitetura do sistema de monitoramento: receptor.....	60
Figura 36 – Comparação do plano gratuito do ThingSpeak com Servidor Local.....	62
Figura 37 – Aplicativo Virtuino executando o template PVmonitor .....	63
Figura 38 – Aplicativo Virtuino com alguns manômetros de dados .....	64
Figura 39 – Aplicativo Virtuino com alguns gráficos .....	64
Figura 40 – Diagrama de adaptação P2: DS18B20 .....	66
Figura 41 – DS18B20 adaptado com plug P2 e cintas plásticas para identificação ..	66
Figura 42 – Diagrama de adaptação P2: Tensão e Corrente .....	67
Figura 43 – Adaptação com extensão para os sensores de corrente e tensão.....	67
Figura 44 – Proteção das extensões criada para os sensores de corrente e tensão	68
Figura 45 – Sensor de temperatura fixado de baixo do módulo fotovoltaica .....	68
Figura 46 – Conexão da extensão e o sensor fixados com a ajuda de fita isolante ..	69
Figura 47 – Protótipo utilizando protoboard .....	70
Figura 48 – PCB adaptada para o Arduino transmissor .....	70
Figura 49 – Adaptação do botão de reset .....	71
Figura 50 – PCB adaptada para Arduino receptor .....	72
Figura 51– Esquema elétrico da placa do Arduino Transmissor .....	72
Figura 52 – Esquema elétrico da placa do Arduino Receptor .....	73
Figura 53 – PCB projetada para substituir protótipo.....	73
Figura 54 – Arduino Mega 2560 fixado dentro da caixa hermética .....	74

## LISTA DE ABREVIATURAS E SIGLAS

P&D	Pesquisa e Desenvolvimento
FV	Fotovoltaica
DAQ	Aquisição de Dados
PC	Computador Pessoal
CBENS	Congressos Brasileiros de Energia Solar
RMS	<i>Root Mean Square</i>
CC	Corrente Contínua
CA	Corrente Alternada
LAMP	<i>Linux, Apache, MySQL e PHP</i>
SFCR	Sistemas Fotovoltaicos Conectados à Rede
VI	<i>Visual Instrument</i>
PLC	<i>Power Line Communication</i>
PWM	Modulação por Largura de Pulso
RTC	Relógio de Tempo Real
ADCES	<i>Analog/Digital Converter Embedded System</i>
REMS	<i>Renewable Energy Monitoring System</i>
HID	<i>Human Interface Device</i>
CLP	Controlador Lógico Programável
SCADA	Supervisory Control and Data Acquisition
DC	<i>Direct Current</i>
MPPT	<i>Maximum Power Point Tracking</i>
AGM	Absorbed Glass Mat
VLRA	<i>Valve Regulated Lead Acid</i>
f.e.m	força eletro motriz
UV	Ultravioleta
FAEN	Faculdade de Engenharia
UFGD	Universidade Federal da Grande Dourados
UTM	Universal Transversa de Mercator
SMD	Surface Mounting Device
ISM	Industrial, Scientific and Medical Radio Bands
SPI	Interface Serial Periférica
TX	Transmissor

RX	Receptor
I2C	Inter-Integrated Circuit
LGPL	Licença Pública Geral Menor
UART	Receptor/Transmissor Assíncrono Universal
IDE	Ambiente de Desenvolvimento Integrado
CSV	valores separados por vírgula
IoT	Internet of Things (Internet das Coisas)
PVC	Policloreto de Vinila

## LISTA DE SÍMBOLOS

%	Percentual (1/100)
<i>V</i>	Tensão (Volts)
Hz	Frequência (Hertz)
W	Potência (Watts)
<i>U</i>	Tensão Induzida
K	Constante
B	Campo Magnético
V	Velocidade
D	Distância
°	Grau
'	Minuto
"	Segundo
m	Metros
M	Mega
J	Joule
$m^2$	Metros Quadrados
<i>h</i>	Horas
k	Kilo
°C	Graus Celsius
W <sub>p</sub>	Watt-pico
A	Corrente (Ampère)
Ah	Ampère hora
GND	Ground
V <sub>cc</sub>	Voltagem em corrente contínua
$\Omega$	Resistência (Ohms)
mm	Milímetros
Acc	Corrente em corrente contínua
mA	Miliampere
$m^3$	Metros Cúbicos
<i>R</i>	Resistência

KB	Kilobyte
Km	Kilômetro
dBi	Decibel
pol	Polegadas
cm	Centímetro

## SUMÁRIO

1.	<b>INTRODUÇÃO</b>	14
2.	<b>A PESQUISA COMO FERRAMENTA DE ANÁLISE SOBRE OS DAQS</b>	17
2.1	<b>Fundamentação teórica</b>	19
3.	<b>A ENERGIA FOTOVOLTAICA E OS TIPOS DE SISTEMAS</b>	28
3.1	<b>História da energia fotovoltaica</b>	28
3.1.1	<i>As células fotovoltaicas</i>	28
3.2	<b>Sistema FV autônomo</b>	29
3.3	<b>Sistema de bombeamento de água FV</b>	30
3.4	<b>Descrição dos equipamentos e suas funcionalidades</b>	31
3.4.1	<i>Módulo fotovoltaico</i>	32
3.4.2	<i>Controlador de Bomba (Driver)</i>	33
3.4.3	<i>Controlador de carga</i>	33
3.4.4	<i>Inversor</i>	34
3.4.5	<i>Bateria</i>	35
3.4.6	<i>Medidor de Vazão</i>	36
3.4.7	<i>Bomba imersa vibratória</i>	38
3.4.8	<i>Conectores MC4</i>	39
3.5	<b>Unidade FV utilizada como elemento de pesquisa deste trabalho</b>	40
4.	<b>O SISTEMA DE AQUISIÇÃO DE DADOS</b>	42
4.1	<b>Descrição dos materiais e equipamentos envolvidos</b>	42
4.1.1	<i>Sensor de temperatura</i>	42
4.1.2	<i>Sensor de Tensão</i>	44
4.1.3	<i>Sensor de Corrente</i>	48
4.1.4	<i>Medidor de Vazão</i>	51
4.1.5	<i>Transceptor nRF24L01</i>	53
4.1.6	<i>Módulo Datalogger + SD + RTC DS1307</i>	55
4.1.7	<i>Ethernet Shield W5100</i>	56
4.1.8	<i>Arduino Mega 2560</i>	57
4.2	<b>Arquitetura do sistema a nível de hardware</b>	58
4.2.1	<i>Transmissor (Arduino instalado em campo)</i>	58
4.2.2	<i>Receptor (Arduino instalado em sala no prédio da FAEN)</i>	59
4.3	<b>Arquitetura do sistema a nível de software</b>	60

<b>4.3.1 Programação do transmissor</b>	60
<b>4.3.2 Programação do receptor</b>	61
<b>4.3.3 Outras aplicações utilizadas</b>	61
<b>4.3 Problemas e soluções encontrados</b>	64
<b>4.4 Otimizações e melhorias na instalação</b>	65
<b>4.4.1 Sensores de temperatura</b>	65
<b>4.4.2 Sensores de tensão e corrente</b>	66
<b>4.4.3 Fixação dos sensores de temperatura</b>	68
<b>4.4.4 Prototipação e PCB perfurada (protoboard)</b>	69
<b>4.4.5 Placa de circuito impresso</b>	72
<b>4.4.6 Arduino em campo (transmissor)</b>	73
<b>5. CONCLUSÃO</b>	75
<b>APÊNDICE A – Código Fonte do Arduino Transmissor (TX)</b>	80
<b>APÊNDICE B – Código Fonte do Arduino Receptor (RX)</b>	85
<b>APÊNDICE C – Diagrama Sistema Fotovoltaico</b>	90
<b>APÊNDICE D – Diagrama Sistema Monitoramento: Transmissor</b>	91
<b>APÊNDICE E – Diagrama Sistema Monitoramento: Receptor</b>	92

## 1. INTRODUÇÃO

O expressivo desenvolvimento das tecnologias de informação e comunicação nos últimos anos, especialmente no setor de pesquisa e desenvolvimento (P&D) na indústria, proporcionou uma impulsão na produção de recursos sofisticados e de baixo custo para sensoriamento e comunicação. Em paralelo a esse cenário de desenvolvimento de mecanismos mais avançados com custo relativamente baixo, temos o crescimento das discussões acerca da consciência energética e ambiental.

Nessa vertente, considerando o aumento demasiado da demanda de energia elétrica no Brasil e no mundo, é de grande importância a busca por fontes alternativas de energia que não dependem de recursos não renováveis e que tenham baixo impacto no meio ambiente, como é o caso da energia eólica, geotérmica, solar térmica, fotovoltaica (FV), entre outras.

No Brasil, há uma grande dependência da energia hidroelétrica dado os volumosos recursos hídricos, porém, é considerável que esse tipo de geração de energia requer um elevado custo em planejamento e execução, além de provocar bruscas alterações ambientais no local onde é instalada, e em alguns casos, a rede de distribuição não alcança determinadas regiões remotas por conta do difícil acesso.

Uma das alternativas mais promissoras em contextos de micro geração para áreas remotas e ambientes rurais, é a fotovoltaica, por conta da sua modularidade, baixo custo com manutenção e ampla vida útil. No Plano Nacional de Energia 2030, o EPE (2007) ressalta que com a ampliação do mercado no setor fotovoltaico, o preço tem reduzido de forma abrupta, possibilitando maior acesso para uso doméstico e rural, e essa alternativa de energia é considerada a mais promissora para o enfrentamento dos desafios do novo milênio.

Cada localidade possui um índice de radiação solar específico, tal variação provoca diferentes resultados que podem ser percebidos ao analisarmos variáveis térmicas e elétricas de cada sistema de geração de energia. Para progredirmos com as pesquisas relacionadas à energia solar e obtermos resultados eficientes, precisamos adquirir dados que permitam uma profunda análise do comportamento desses sistemas. Com o propósito de solucionar essa questão da demanda de dados, precisamos de um mecanismo de aquisição de dados (DAQ), que consiste em uma rede de sensores interligados em um dispositivo capaz de condicionar e converter

sinais e se comunicar com computadores e smartphones através de uma interface de comunicação.

Um sistema de aquisição de dados simplificado pode ser definido por três componentes, sensores ou transdutores, **hardware** de aquisição e dispositivo hospedeiro. Os sensores e transdutores são instrumentos que fornecem parâmetros reais sobre os fenômenos físicos observados, eles são capazes de converter esses comportamentos em sinais elétricos que podem ser interpretados de acordo com a grandeza física analisada. O *hardware* de aquisição consiste em um dispositivo de entrada e saída que realiza processamento, conversão de sinais e quando necessário o condicionamento, com a amplificação e filtragem de ruído. Já o dispositivo hospedeiro representa o receptor desses dados, como um computador, *smartphone*, *tablet* entre outros dispositivos computacionais.

O problema de estudo deste trabalho consiste na aquisição de dados para o monitoramento remoto de uma unidade FV para bombeamento de água, com o uso de elementos de *hardware* e *software* capazes de obterem, armazenar e disponibilizar dados de maneira objetiva e amigável para o usuário. Para solucionar o problema proposto, foi definido o seguinte objetivo geral: Desenvolver e implantar um sistema de aquisição e armazenamento de dados o qual permite monitorar remotamente uma unidade FV para bombeamento de água.

Para alcançar tal objetivo, são propostos os seguintes objetivos específicos:

- Revisar a bibliografia de um dos eventos técnico-científico mais importantes no cenário nacional na área de energia solar, com a finalidade de fortalecer a importância dessa problemática;
- Elaborar um projeto utilizando microcontrolador em conjunto com sensores para obter os seguintes dados relevantes: tensão, corrente elétrica, potência, temperatura e vazão;
- Implementar uma comunicação que utilize a tecnologia de rede sem fio (*wireless*) para proporcionar o envio destes dados para um servidor;
- Implementar uma aplicação que se comunique com a base de dados obtidos em campo ou em tempo real, permitindo o acesso por qualquer dispositivo conectado à internet, seja ele móvel como *smartphones*, *tablets* e *notebooks*, ou *desktop* como um computador pessoal (PC).

É importante ressaltar que apesar do foco deste trabalho ser o preenchimento de uma lacuna específica de uma unidade FV, ele deverá permitir também a

articulação para o desenvolvimento de diferentes contextos de sistema e além disso, a base para pesquisa e análise de dados dessa unidade fotovoltaica em que será implantado.

## 2. A PESQUISA COMO FERRAMENTA DE ANÁLISE SOBRE OS DAQS

Na busca de compreender as questões envolvidas na discussão deste trabalho, e investigar como estão sendo desenvolvidos os sistemas de DAQs aplicados em fontes de energia renovável, foi realizada uma revisão de literatura nos anais publicados e apresentados nos Congressos Brasileiros de Energia Solar (CBENS), entre os anos de 2008 a 2018. A escolha deste evento se deve a sua representatividade para os pesquisadores da área e a sua abrangência nacional. Entretanto, vale ressaltar que, o CBENS, não é o único evento da área de energia solar que permite esse entendimento sobre o tema investigado.

Para a seleção dos trabalhos, utilizou-se os mecanismos da consulta *online*, consultando, ano a ano nos índices e barras de pesquisas nas páginas dos eventos, por meio dos títulos, palavras-chave e resumos que relacionassem aos respectivos assuntos: “Aquisição”, “Monitoramento” e “Coleta”, visto que, essa busca foi realizada por pares, sendo necessária a aparição de dois termos, no mínimo. Foram analisados os últimos 7 congressos, os quais haviam diversos trabalhos, porém, só alguns condiziam com os critérios pré-estabelecidos de busca.

Embora alguns congressos já não possuam suas páginas virtuais disponíveis, consequentemente, tornando uma consulta limítrofe de alguns trabalhos, foram selecionados 15 artigos, os quais apresentam inicialmente, maiores possibilidades de discussões e estimulam a reflexão acerca do tema de interesse, conforme pode ser visto no Quadro 1.

**Quadro 1 - Distribuição dos artigos selecionados nos anais dos CBENS.**

Ano do evento	Título do trabalho	Autor (a) (es)
2010	Desenvolvimento de um sistema de aquisição de dados para monitoração de um sistema fotovoltaico conectado à rede	Manito et al.
2012	Desenvolvimento de um sistema de supervisão e aquisição de dados para bancada de ensaios de motobombas utilizando labview	Jesus et al.
2012	Sistema de automação, controle e aquisição de dados para fontes renováveis descentralizadas	Pinheiro Neto et

		al.
2012	Sistema de hardware e software para a aquisição de dados automática para painéis fotovoltaicos	Freitas, Moraes e Silva.
2012	Supervisório de código aberto e baixo custo para monitoramento remoto de geração fotovoltaica conectada à rede	Torelli et al.
2014	Implementação de um sistema de monitoração em um sistema híbrido solar-eólico instalado no flutuante base do instituto de desenvolvimento sustentável mamirauá	Cavalcant e et al.
2016	Aquisição de dados de um sistema fotovoltaico conectado à rede com a plataforma arduino	Dias, Dias e Prá.
2016	Desenvolvimento de um sistema de aquisição de dados para aquecedores solares planos	Grando et al.
2016	Desenvolvimento de um sistema de monitoração para avaliação de desempenho operacional de um sistema fotovoltaico conectado à rede de baixa tensão	Carvalho et al.
2016	Desenvolvimento de um sistema de monitoramento fotovoltaico baseado em power line communication	Coelho et al.
2016	Sistema de aquisição para painéis fotovoltaicos com armazenamento de dados em servidor remoto utilizando plataformas open source Raspberry pi e arduino	Oliveira Filho e Maia.
2016	Sistema embarcado linux aplicado ao monitoramento em nuvem de planta de microgeração fotovoltaica	Dupont et al.
2018	Aplicativo android para monitoramento remoto de módulos fotovoltaicos	Piombini, Melo e Campos.
2018	Desenvolvimento de um sistema de monitoração para análise comparativa de dois sistemas fotovoltaicos conectados à rede com diferentes configurações	Teixeira Junior, Blasques e Nascimento.
2018	Desenvolvimento de uma plataforma experimental para aquisição de dados de geração fotovoltaica	Cagnini, Grando e Costa.

Fonte: elaborado pelos autores (2018).

## 2.1 Fundamentação teórica

A partir da revisão de literatura realizada nos CBENS, conforme citadas no quadro 1, apresentamos neste tópico, as reflexões e articulações entre tais pesquisas, buscando compreender os elementos fundamentais de um sistema de aquisição de dados (DAQ) voltado para energia solar fotovoltaica sob perspectivas de outros estudiosos.

No sistema fotovoltaico proposto por Manito et al. (2010), onde tinham como objetivos, a implementação de um DAQ com a capacidade de verificar em tempo real raízes do valor quadrático médio, do inglês *root mean square* (RMS) e as formas de onda, de tensão e corrente, tanto para CC, quanto para CA, além disso, coletar outros parâmetros importantes, tais como o potencial energético produzido e consumido no sistema, a temperatura da face posterior do painel solar e a irradiação.

Segundo os autores, a implementação de um programa proprietário para solucionar problemas de coleta de dados do laboratório, proporciona uma maior flexibilidade no que diz respeito a compatibilidade com instrumentos de medição de diferentes fabricantes e um custo reduzido na aquisição e manutenção em *software*. Este DAQ foi desenvolvido utilizando a linguagem de programação gráfica LabVIEW, que segundo os pesquisadores, provê ferramentas específicas para interfaces de aquisição de dados, com comandos simplificados e amigáveis ao usuário, já em termos de *hardware*, foi utilizada uma placa de aquisição de dados PCI 1002 L. Sobre o DAQ, os autores concluem que

O novo sistema de aquisição de dados torna a monitoração da edificação muito mais prática e acessível. O programa de monitoração possibilitará acompanhar o desempenho do prédio e dos seus sistemas em tempo real, tanto no Laboratório quanto pela Internet [...] (p. 8).

A respeito do LabVIEW, Jesus et al. (2012) descrevem suas principais características e benefícios, tais como, permitir o desenvolvimento personalizado de aplicações de maneira rápida e eficaz, incluir diferentes instrumentos virtuais de análises, além de obter dados a partir da utilização de diversos dispositivos. Para o desenvolvimento de tal trabalho, inicialmente os autores identificaram um obstáculo, o qual, motivou-os a buscar uma solução.

[...] o procedimento de realização do ensaio não permite que o técnico visualize as informações em tempo real e possa realizar uma análise imediata, tendo em vista descartar ensaios mal sucedidos, evitando assim o trabalho de analisar dados inválidos. Uma forma de solucionar este problema é integrando à bancada de ensaios um sistema de supervisão e aquisição de dados. Todavia, dificilmente se encontra no mercado um sistema que atenda a esta aplicação específica (p. 1).

Nesse sentido, os autores buscaram desenvolver um sistema supervisório de aquisição de dados para bancadas de ensaios de motobombas, cujo LabVIEW proporcionou a visualização dos parâmetros desejados durante a realização da pesquisa. Além disso, os autores afirmam que o uso do sistema com gerador fotovoltaico possibilita variadas curvas de desempenho e também, armazenam uma grande quantidade de dados, sendo estes, acessados remotamente e possibilita o envio automático dos mesmos por e-mail.

Na busca de elaborar um sistema simples, produtivo e com custo reduzido, Torelli et al. (2012) apresentaram a elaboração de um protótipo para monitorar de modo *on-line* e remoto diversas grandezas elétricas. Sobre tal aspecto, os autores afirmam que “O monitoramento remoto via *internet* é uma característica interessante e não necessita da instalação de qualquer *software* no computador que acessará o servidor” (p. 7).

A respeito da motivação em desenvolverem seu próprio *software*, os autores consideraram que existem “[...] outras opções para realizar esta tarefa, mas a preferência por elaborar um *software* sob medida prevaleceu pelo fato de nenhuma das alternativas abordadas apresentarem fácil e direta personalização, acesso remoto e que fossem gratuitas” (p. 5). Então, os pesquisadores desenvolveram um sistema supervisório composto por elementos de *hardware* e *software* que melhor atenderiam os quesitos de simplicidade, flexibilidade, mobilidade, portabilidade e disponibilidade.

Para isso, a nível de *hardware*, os pesquisadores optaram por utilizar o Arduino para realizar a aquisição de dados, cálculos e o envio das variáveis coletadas para um computador, o qual desempenha o papel de servidor *Linux*, *Apache*, *MySql* e *PHP* (*LAMP*), onde os dados devem ser armazenados e organizados para posterior consulta através de um navegador *web*. Já a nível de *software* os autores implementaram um programa para manipular o banco de dados do servidor e oferecer uma interface amigável para o usuário. Entre as vantagens observadas, como por

exemplo, o baixo custo, ressalta-se a importância do uso do microcontrolador Arduino, pois através dele foi possível simplificar o processamento e proporcionar um circuito em tamanho portátil.

Similarmente, Dias, Dias e Prá (2016) utilizaram também como base de uma DAQ para um sistema fotovoltaico em conexão com a rede, uma plataforma de desenvolvimento livre, denominada Arduino. Tal semelhança com o trabalho anteriormente citado, deve-se a plataforma, entretanto, nesta pesquisa, realizou medições de variáveis diferentes, sendo elas: temperatura e irradiação.

Com a proposta de contribuir com os estudos e análises a respeito de sistemas fotovoltaicos conectados à rede (SFCR), Carvalho et al. (2016), desenvolveram um protótipo de sistema de monitoração também utilizando a plataforma Arduino, nesse sistema o microcontrolador funciona como uma central de aquisição e tratamento de dados, nesse intuito, foram definidos sensores e transdutores específicos para cada grandeza a ser mensurada. Segundo os autores,

O uso da plataforma de prototipagem Arduino mostrou-se uma boa escolha para realizar as funções de controle do sistema de monitoração, uma vez que é uma ferramenta de baixo custo, fácil programação e com uma grande capacidade de adaptação às necessidades do projeto (p. 7).

Embora, ainda haja um fragmento de aprimoramentos que podem ser realizados no sistema de modo a torná-lo mais eficaz e influente na execução de suas tarefas de monitoramento.

Nessa perspectiva, Cavalcante et al. (2014), afirmam que só é possível realizar um diagnóstico preciso do funcionamento de um sistema gerador, e realizar as devidas avaliações de forma detalhada, se forem obtidas as grandezas fundamentais envolvidas com tal sistema. Para isso, os autores utilizaram uma placa de aquisição de dados, modelo NI25018, da *National Instruments* conectada a um computador, e elaboraram um programa com interface gráfica utilizando o LabVIEW (também da *National Instruments*) para usufruir do conceito instrumentos visuais, do inglês *visual instrument* (VI), o qual permite a criação lógica de instrumentos de medição e análise de sinais como se esses existissem no meio físico.

Os pesquisadores, ressaltam ainda, a importância das características individuais de cada local onde um sistema de geração é instalado, quando dizem que

o “[...] monitoramento é ferramenta de grande valor científico, devido às características singulares do local de instalação” (p. 2).

Ao pensar no mercado empresarial, Oliveira Filho e Maia (2016) relatam a existência de inúmeros sistemas de aquisição de dados e, desde a última década, surgiram vários trabalhos com o intuito de baratear e melhorar a eficácia desses sistemas, entretanto, em sua maioria, utilizam a conexão física para executar suas transferências de dados. Para viabilizar esse sistema, os autores propuseram desenvolver um sistema que não dependa de um computador externo para processar os dados, e que a comunicação seja feita completamente sem fio.

Os resultados cujo autores apontaram, mostra um investimento financeiro reduzido dos sistemas já existentes, uma vez que não é preciso utilizar um computador convencional para o armazenamento e tratamento dos dados, mas sim, um módulo Raspberry Pi. Notaram também que houve uma melhoria na infraestrutura de transmissão de dados, mediante a sua execução sem utilizar os fios. Em relação ao desenvolvimento deste trabalho, os autores descrevem,

O sistema proposto faz uso de um servidor administrado por um Raspberry Pi, à medida que os sistemas existentes no mercado utilizam um computador mais caro, superdimensionado para as necessidades do projeto. Isso proporciona uma diminuição de custos na implementação do projeto, haja vista o investimento inferior necessário para aquisição de um módulo Raspberry Pi, quando comparado a um computador convencional (p. 4).

Ainda sobre um módulo para monitoramento, Coelho et al. (2016) realizaram um experimento cujo objetivo era desenvolver um sistema para monitorar painéis fotovoltaicos com base em *power line communication* (PLC), e utilizou três módulos, dois para modulação por largura de pulso (PWM), e um para prover a comunicação com o computador via interface RS-232. Durante o experimento,

Foi observado que fontes de alimentação instáveis podem levar a erros de medida, uma vez que o sensor tem uma tensão de offset sensível à mudanças na tensão de alimentação, assim, a curva de calibração deve ser ajustada de acordo com a fonte utilizada no projeto (p. 3).

Contudo, a rede PLC mostrou-se eficaz e que pode ser uma excelente opção para o desenvolvimento de ferramentas que auxiliem nas construções e pesquisas no âmbito de energias renováveis.

Em função de uma conexão e transmissão de dados via rede sem fio, Piombini, Melo e Campos (2018), desenvolveram um aplicativo com um sistema operacional Android, cuja plataforma livre era um Arduino com sensores, afim de obter e enviar dados. Conforme avaliação dos autores, o aplicativo mostrou-se confiável e estável, além de permitir aquisição de dados. No projeto, também foi utilizado um cartão micro SD e este demonstrou grande eficácia, visto que, com ele foi possível adquirir de modo constante, valores e dados sobre como o sistema de aquisição e monitoria está funcionando.

Entretanto, “[...] esse desenvolvimento necessitou de um estudo profundo sobre os sistemas operacionais e seus componentes para que o monitoramento remoto fosse feito com qualidade” (p. 9).

A dificuldade em medir e registrar manualmente uma grande quantidade de dados com periodicidade, em curtos intervalos de tempo, ao longo de horas ou dias, motivou Grando et al. (2016) a desenvolverem um sistema automatizado que realize essas tarefas. Os autores propõem uma definição bem consolidada a respeito dos sistemas de aquisição de dados, em que para eles existe uma espécie de receita, chamada por eles de estrutura genérica. Tal definição compreende que

A aquisição de dados (DAQ) é o processo pelo qual um fenômeno físico é transformado em um sinal elétrico proporcional e convertido em formato digital, sendo processado, analisado e armazenado através do uso de um computador. Todo processo de aquisição de dados é formado por blocos funcionais básicos, sendo eles: sensores ou transdutores, condicionamento de sinais, hardware de medição e um computador com software programável (p. 2).

Os pesquisadores definiram a utilização de uma placa de aquisição proprietária, projetada pela *National Instruments*, modelo NI USB-6259, que conta com recursos considerados ideais para tarefas de múltiplas aquisições de vários tipos de sinais, que além de atender aos requisitos atuais de instrumentação, permite expansão de novos sensores e instrumentos de medição.

Indo no sentido da utilização de soluções proprietárias, Teixeira Junior, Blasques e Nascimento (2018) utilizaram uma placa PCI DAQ para compor o seu

sistema de aquisição de dados, e devido ao fato dessa placa possuir número de canais menor do que o número de sensores e transdutores, foi preciso utilizar um circuito de multiplexação de sinal, além desse circuito, foram desenvolvidos outros dois, um para amplificação e conversão, e outro para filtragem e interface entre os instrumentos de campo e a placa de aquisição PCI. Os autores conseguiram demonstrar a eficácia de seu sistema ao fazer comparação dos dados coletados pelo DAQ com aferições de um osciloscópio, o que validou a coerência dos dados obtidos.

Entre as pesquisas citadas, ao falar sobre as exigências de *hardware* e *software* para um DAQ, Grando et al. (2016) se destacam ao afirmarem que

A necessidade de adaptar as funcionalidades de cada sensor e aparelho de medição, às funcionalidades da placa de aquisição, constituem um desafio inerente em qualquer projeto de aquisição de dados. Por outro lado, a capacidade da placa e o potencial do software LabVIEW, permitiram simplificar o hardware construído no condicionamento de sinal, reduzindo custos de material e aumentando a confiabilidade do sistema, uma vez que, quanto mais hardware utilizado em um projeto, mais sujeito a erros estará o sistema. Por isso, buscou-se atribuir a responsabilidade dos ajustes técnicos, o máximo possível, ao software (p. 7).

Com o intuito de analisar os fatores que afetam o custo de investimento em equipamentos para geração fotovoltaica e tempo de retorno, Cagnini, Grando e Costa (2018) propõem o desenvolvimento de uma plataforma para realizar a coleta e armazenamento de dados referentes a um sistema de geração composto por painéis fotovoltaicos e outros fatores que também influenciam na geração de energia, permitindo ainda comparar dois sistemas fotovoltaicos simultaneamente e de forma isolada.

Nesse sistema, assim como no projeto de um DAQ voltado para aquecedores solares planos, apresentado por Grando et al. (2016), foi utilizado o *hardware* proprietário NI USB-6259 para aquisição, discretização e comunicação com o computador. Os pesquisadores ressaltam a importância comercial do DAQ desenvolvido, pois como

A plataforma construída tem a capacidade de analisar dois sistemas fotovoltaicos distintos e de forma totalmente isolada. Isso permite a comparação entre dois sistemas fotovoltaicos expostos as mesmas condições, ou painéis de diferentes fabricantes ou diferentes tecnologias de fabricação. Do ponto de vista comercial isso é

importante para encontrar o sistema ou painel fotovoltaico com o melhor custo/benefício (p. 9).

Na busca por inovação e portabilidade, Dupont et al. desenvolveram um complexo sistema DAQ que pode ser visualizado como dois subsistemas, o primeiro de aquisição, composto por um microcontrolador PIC18Fxx5x conectado a uma memória externa e a um relógio de tempo real (RTC), possibilitando a transmissão dos dados coletados para um segundo subsistema de supervisão, através de uma comunicação via interface serial MODBUS. Este segundo subsistema, é composto por um minicomputador Raspberry Pi, que realiza o armazenamento no banco de dados e a transmissão em tempo real dos dados para um servidor na nuvem.

Vale ressaltar que os autores demonstram preocupação em tornar o sistema eficaz e de fácil manutenção, pois

Outra vantagem da plataforma proposta é que usando o serviço online como um ponto de acesso, não é necessário desbloquear as portas e/ou firewall do roteador. O firmware ADCES também pode ser baixado online e atualizado pelo REMS via USB através do protocolo HID (Human Interface Device). Vale ressaltar que a comunicação entre o usuário e o sistema de monitoramento é feita de maneira remota, sem a necessidade de um computador dedicado no local (p. 3).

Contudo, os pesquisadores deste trabalho concluíram que o sistema proposto possui elevada eficiência, devido à possibilidade de realizar consulta em tempo real dos dados relacionados a planta de geração, além disso, tal sistema proporciona acessibilidade e maior interação com os usuários, pois apresenta baixo custo de implementação.

Foi observado um caso em particular, da utilização de um controlador lógico programável (CLP) para monitorar e controlar os componentes de um laboratório de energias alternativas, composto por sistemas de plantas de dessalinização via osmose reversa, de geração de energia fotovoltaica e de biogás, controlador de carga, de bombeamento e biodigestor (Pinheiro Neto et al., 2012). Os autores demonstram afinidade para o uso de CLP devido a sua flexibilidade, visto que, “Este painel proporciona a modificação do *hardware* e *software* de forma simples, alterando conforme a necessidade o controle, monitoramento e interação entre as plantas” (p. 2).

Nesse DAQ, os pesquisadores utilizaram um computador para fazer a interface com usuário e para armazenar os dados coletados, além disso, foi utilizado o *software* de sistema supervisório ELIPSE SCADA que atua como mestre na comunicação serial MODBUS. A escolha por este *software* proprietário foi motivada pela sua agilidade e versatilidade no desenvolvimento das aplicações, e por sua confiabilidade e disponibilidade de suporte técnico, considerado pelos autores um *software* de supervisão completo.

Os autores tiveram uma importante conclusão no que tange a viabilidade da aplicação deste sistema em diferentes portes de plantas, já que

A aplicação de um sistema tão completo é viável para integração de plantas de médio/grande porte ou em laboratórios, facilitando o gerenciamento, manutenção, análise e estudo dos sistemas, porém seu custo final e necessidade de mão-de-obra especializada para manutenção, dificultam sua aplicação em pequenas plantas de geração (p. 8).

Para Freitas, Moraes e Silva (2012), estratégias de monitoramento e aquisição de dados são de grande relevância para sistemas geradores de energia elétrica, principalmente para o seccionamento em regiões onde a energia primária é temporária, dependente de fatores climáticos. Além disso, esses sistemas oferecem recursos que auxiliam na manutenção preditiva, que é fundamental para gerenciar a manutenção da planta geradora.

Os pesquisadores deste trabalho, expressam sua preocupação com a viabilidade/custo e a eficiência/consumo, reconhecendo a existência de diversas soluções comerciais para tal finalidade, mas ressaltando o seu custo, visto que

Existem vários sistemas comerciais para realizar a aquisição de dados tanto em aplicações de campo como laboratório, mas em geral eles são relativamente caros. O sistema de aquisição de dados desenvolvido neste trabalho tem um baixo custo e baixo consumo de energia elétrica, caracterizando-se também como boa alternativa para acompanhamento de sistemas embarcados (p. 1).

O sistema desenvolvido neste trabalho, foi projetado para a coleta de valores de corrente, tensão de armadura, e a temperatura do painel solar, possibilitando a exibição destes dados na forma de diversos gráficos comparando essas variáveis. O armazenamento é realizado em um banco de dados e o controle e aquisição dos

dados é automatizado via comunicação serial em interface RS-232 de um computador. Este DAQ é constituído por uma placa de comunicação que utiliza o PIC16F877A para estabelecer contato entre as placas dos sensores e o *software* desenvolvido para o computador do laboratório.

Consequentemente, os autores chegaram à conclusão de que o DAQ desenvolvido tem grande importância e utilidade nos sistemas de geração de energia fotovoltaica, por conta da sua rapidez e precisão na aquisição e tratamento das variáveis coletadas, além disso, ressaltam que na instrumentação eletrônica automatizada, não ocorrem erros de cálculos e leitura, diferentemente da aferição manual.

### 3. A ENERGIA FOTOVOLTAICA E OS TIPOS DE SISTEMAS

Este capítulo tem o intuito de oferecer um panorama histórico da energia solar FV, descrever suscintamente os elementos básicos de um sistema FV autônomo, de um sistema de bombeamento de água FV, o funcionamento de cada elemento destes sistemas e por fim apresentar o sistema FV em que este trabalho foi aplicado.

#### 3.1 História da energia fotovoltaica

A energia fotovoltaica é a conversão da luz em eletricidade. Alguns materiais exibem propriedade conhecida como efeito fotoelétrico que faz com que absorvam fótons de luz e liberem elétrons. Quando esses elétrons livres são capturados ocorre a conversão direta da potência associada à radiação solar em potência eléctrica DC. O efeito fotoelétrico foi notado pela primeira vez por um físico francês, Edmund Bequerel, em 1839, que descobriu que certos materiais produziriam pequenas quantidades de corrente elétrica quando expostos à luz. Em 1905, Albert Einstein descreveu a natureza da luz e o efeito fotoelétrico no qual a tecnologia fotovoltaica é baseada, para o qual ele mais tarde ganhou um prêmio Nobel de Física.

O primeiro módulo fotovoltaico foi construído pela Bell Laboratórios em 1954. Ele foi anunciado como uma bateria solar, criado apenas por curiosidade, por conta do altíssimo custo de produção não era viável a comercialização. Na década de 1960, a indústria espacial começou a fazer o primeiro uso sério da tecnologia para fornecer energia para as espaçonaves. Através dos programas espaciais, a tecnologia avançou e sua confiabilidade foi estabelecida, o custo começou a diminuir conforme o tempo e a demanda foi aumentando.

Durante a crise de energia no Nos anos de 1970, a tecnologia fotovoltaica ganhou reconhecimento como fonte de energia para aplicações não espaciais.

##### 3.1.1 As células fotovoltaicas

Conforme Castro (2002) a célula é o menor elemento de um sistema fotovoltaico e individualmente não produz muita energia, para isso são necessárias várias para atender demandas de maior consumo, uma célula gera em torno de 1,5W de potência sendo aproximadamente 0.5V e 3A. No mercado encontramos tipicamente

painéis fotovoltaicos com potência em torno de 50W a 100W, também encontrando painéis com maior capacidade, mas esses não são comuns.

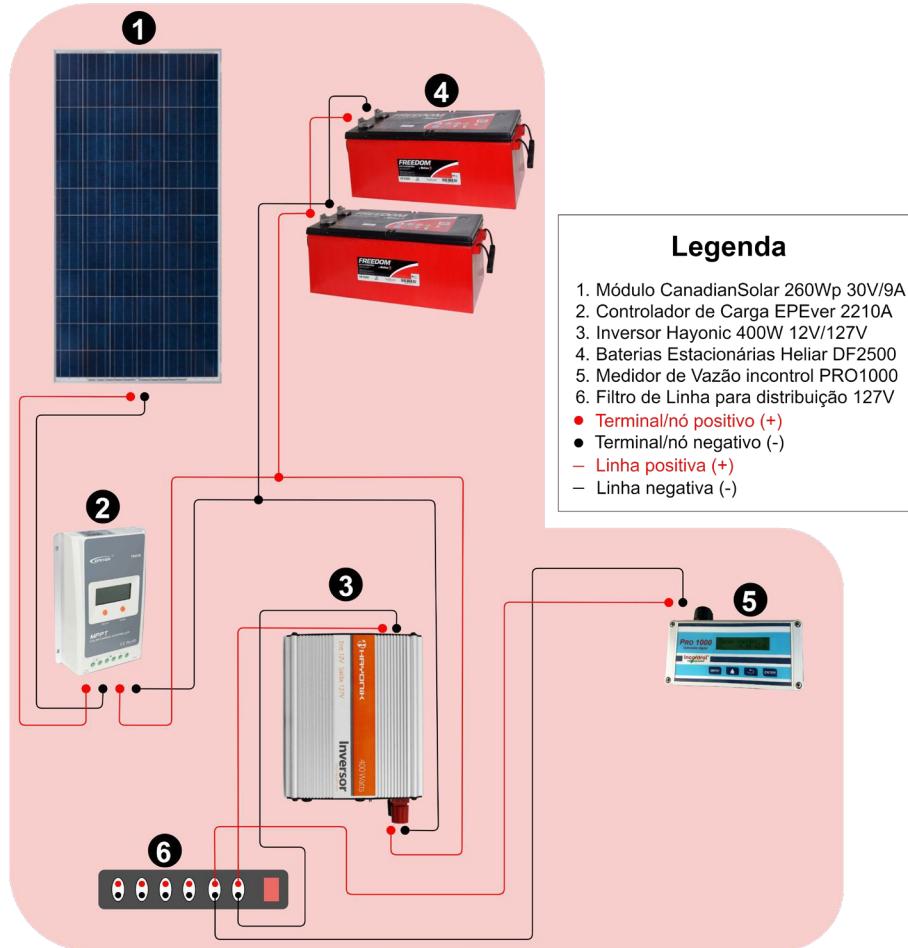
As células fotovoltaicas são formadas por um material semicondutor (silício) no qual são adicionadas substâncias dopantes, na eletrônica consiste num procedimento de adição de impurezas químicas a um elemento semicondutor para transformá-lo em um elemento de maior condutância, porém de forma controlada, assim criando um meio adequado ao estabelecimento do efeito fotovoltaico.

### 3.2 Sistema FV autônomo

São sistemas isolados capazes de garantir sua demanda energética, ou seja, ter autonomia do seu consumo elétrico. Geralmente são compostos por cinco elementos, um ou mais módulos fotovoltaicos, um controlador de carga, uma ou mais baterias (preferencialmente estacionárias), um inversor de tensão contínua para tensão alternada e equipamentos alimentados por essa carga armazenada na bateria. Esse tipo de sistema é amplamente empregado em regiões de difícil acesso a rede elétrica, como locais remotos da Amazônia, interior de grandes propriedades rurais, mas também utilizado em diversas finalidades do nosso cotidiano, tais como iluminação pública, sinalização e radares de estradas, estacionamentos com vagas para recarga de carros elétricos, parquímetros, entre diversas outras finalidades, que variam de pequenos aparelhos eletrônicos até grandes sistemas aeroespaciais. Além de ser uma forma simples no que se refere a quantidade de equipamentos de geração de energia elétrica, esse tipo de geração não polui o meio ambiente, exige pouca manutenção e não necessita do abastecimento constante de combustível, tendo como ponto negativo o seu elevado custo financeiro e energético durante a fabricação das células e módulos.

O sistema apresentado na Figura 1, é composto por um módulo FV, um controlador de carga, duas baterias ligadas em paralelo e um inversor, no caso exemplificado temos um filtro de linha que pode ser uma régua de tomadas, e um medidor de vazão alimentado pelo sistema.

**Figura 1 – Diagrama de um sistema autônomo**



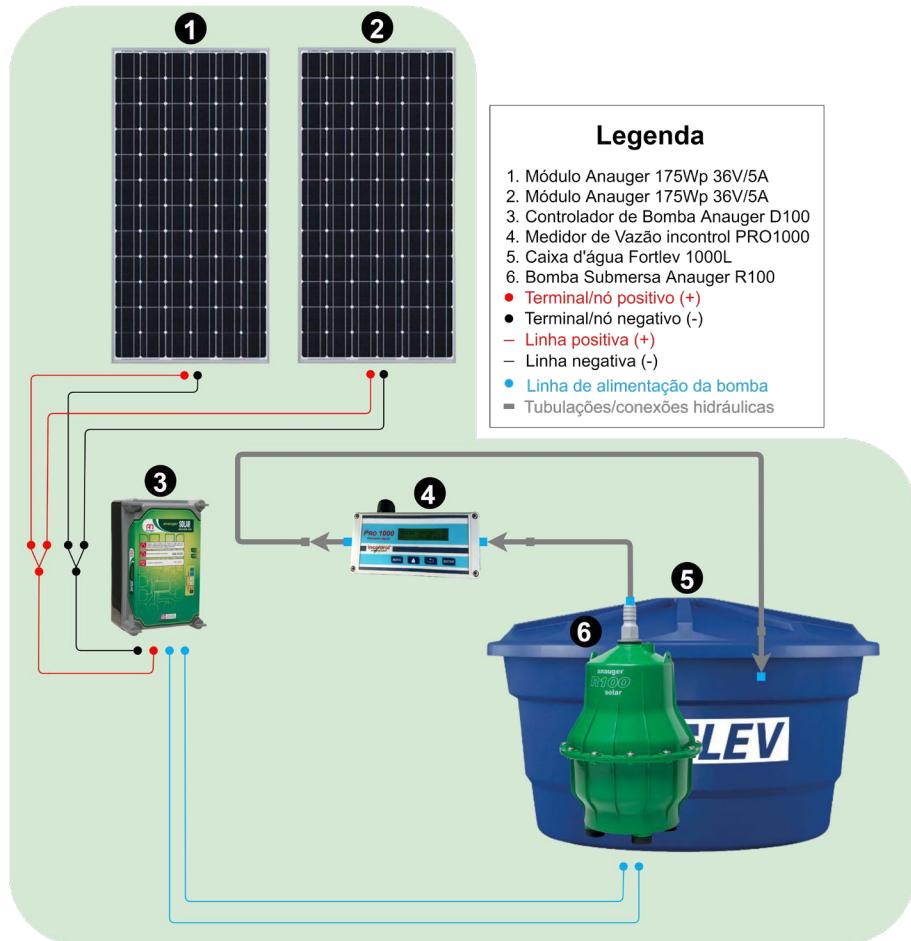
Fonte: elaborado pelos autores (2018).

### 3.3 Sistema de bombeamento de água FV

Os sistemas FV para bombeamento de água também são isolados, e são capazes de garantir o funcionamento de uma bomba d'água durante o dia, equipamentos mais modernos conseguem manter uma boa vazão até em condições de baixa radiação solar. Esses sistemas são compostos por um ou mais módulos fotovoltaicos, um controlador de bomba (*driver*), também conhecido como condicionador de potência, um conjunto motobomba, um ou mais tanques de armazenamento e um ou mais elementos de consumo hídrico abastecidos por esse sistema. Geralmente esses sistemas são usados na zona rural para abastecimento doméstico, irrigação, gotejamento, criação de animais entre outras finalidades. Existem diversos tipos de bombas e sua implantação varia com a necessidade de uso.

O sistema apresentado na Figura 2, é composto por dois módulos FV ligados em paralelo, um controlador de bomba (*driver*), uma bomba submersa vibratória, uma caixa d'água e um medidor de vazão com alimentação externa.

**Figura 2 – Diagrama de um sistema de bombeamento de água**



Fonte: elaborado pelos autores (2018).

### 3.4 Descrição dos equipamentos e suas funcionalidades

Nesta subseção, serão descritas as funcionalidades dos quatro componentes básicos de um sistema FV, módulos FV, controladores de carga, inversores e baterias, e outros que também são encontrados com certa frequência em propósitos específicos.

Uma interessante analogia que pode ser feita com os quatro elementos básicos, é a associação deles com importantes órgãos do corpo humano, como o importante papel do coração de bombear o sangue pelo corpo, que no sistema FV é desempenhado pelos módulos FV que bombeiam energia para o sistema, as válvulas

cardíacas que controlam o fluxo sanguíneo, e que tem função semelhante a dos controladores de carga, que evitam sobrecargas e descargas elétricas no sistema, o cérebro que no sistema FV pode ser compreendido pelos inversores, que realizam transformações de tensões e operam o sincronismo da rede, e por fim as baterias, que como os pulmões, armazenam energia para o posterior consumo na ausência de luz.

### **3.4.1 Módulo fotovoltaico**

O módulo fotovoltaico consiste em um conjunto de células geralmente ligadas em série, montadas sobre uma espécie de armação moldurada. Villalva e Gazoli (2012), com muita propriedade explicam que o princípio do funcionamento dessas células é conhecido como efeito fotovoltaico, onde na presença de energia solar, elétrons de uma camada mais fina de material semicondutor  $N$ , ganham energia suficiente para saltar a barreira entre camadas e se movem para a camada mais espessa de material semicondutor  $P$ , e ao estarem em circuito fechado, geram corrente elétrica. Existem diversos tipos de células fotovoltaicas, como as de silício monocristalino, policristalino, silício amorfo, microcristalino, entre outras. A principal diferença entre essas tecnologias de materiais está associada ao seu custo de fabricação e a sua eficiência energética, sendo o monocristalino o mais eficiente, o policristalino com uma eficiência intermediária, e o amorfo, também empregado na tecnologia de filme fino, o de menor eficiência.

**Figura 3 – Anauger 175Wp (extremidades) e CanadianSolar 260Wp (centro)**



Fonte: elaborado pelos autores (2018).

### **3.4.2 Controlador de Bomba (Driver)**

Placa eletrônica com circuito programado para comandar a energia fornecida pelo módulo FV para a bomba, o funcionamento consiste em armazenar a energia em capacitores e converter essa energia em impulsos elétricos proporcionais ao nível de radiação solar e proteger contra curto circuito ou interrupção de energia, evitando variações bruscas no funcionamento da bomba, o que por sua vez promove maior vida útil ao equipamento (ANAUGER S.A, 2011).

**Figura 4 – Controlador de Bomba Anauger (Driver 100)**



Fonte: elaborado pelos autores (2018).

### **3.4.3 Controlador de carga**

Quando utilizamos uma ou mais baterias em um sistema fotovoltaico existe a necessidade de adicionar um controlador de carga no projeto afim de realizar o controle ideal de carga para as baterias, evitando tanto a sobrecarga ou descarga excessiva.

O controlador mais simples tem as funções de desconectar o módulo fotovoltaico quando as baterias estão completamente cheias, e cessar a alimentação dos circuitos consumidores, que geralmente são possíveis cargas de eletrodomésticos residenciais, maquinários rurais, entre outros, quando os níveis das baterias estão

baixos, assim evitando que elas danifiquem por mal funcionamento. Esses controladores são conhecidos como controladores de liga/desliga.

Já os mais sofisticados possuem um controle eletrônico com PWM e MPPT, onde no lugar da utilização de relés ou chaves utilizam um controle um preciso de correntes e cargas. O MPPT (*Maximum Power Point Tracking*), em português, rastreamento do ponto de máxima potência, permite que o módulo fotovoltaico opere sempre com a tensão necessária pra atingir o ponto de melhor eficiência. Os controladores MPPT em geral melhoram a produção de energia em torno de 30%,  apesar de mais caro pode ser um fator decisivo na hora da compra, podendo até utilizar menos módulos solares, assim economizando recursos como espaço e dinheiro.

**Figura 5 – Controlador de carga EPEver Tracer 2210A MPPT**



Fonte: elaborado pelos autores (2018).

### 3.4.4 Inversor

O inversor é necessário no sistema pois a geração de energia dos módulos FV é em CC e os equipamentos que utilizamos em grande maioria trabalham em CA. Ele tem o papel de realizar a conversão CC para CA, que geralmente é encontrada em

tensões de 127V ou 220V com uma frequência de 60Hz para operação em países como Brasil, Canadá, Estados Unidos. Esses dispositivos, são encontrados com diversas especificações no mercado, normalmente as tensões de entrada são de 12V, 24V ou 48V. Existem dois tipos principais de inversores disponíveis para compra no mercado, o inversor de onda quadrada e o inversor de onda senoidal modificada.

Os inversores de onda senoidal modificada produzem tensões de saída com formato de ondas semisenoidais, as ondas semisenoidais tem menos distorções harmônicas do que as ondas totalmente quadradas, porém ainda são muito distorcidas se comparadas com a onda senoidal pura. O ideal é a utilização de um inversor PWM de onda senoidal pura que funciona com o princípio da modularização de largura de pulso (PWM), onde as ondas geradas por pulsos não são perfeitas, mas possuem baixíssima distorção harmônica, esses inversores são os ideais para alimentar os equipamentos da grande maioria dos consumidores.

**Figura 6 – Inversor de onda senoidal modificada Hayonic 400W**



Fonte: elaborado pelos autores (2018).

### **3.4.5 Bateria**

As baterias são necessárias para fornecer uma demanda constante de energia, em um sistema FV nem sempre temos energia constantes para atender as necessidades de consumo e também não é todo o momento que estamos a consumir, assim para evitar desperdício quando a geração é maior que o consumo as baterias podem ser carregadas, e quando existe a falta de energia as baterias podem compensar a deficiência.

Se utilizar mais de uma bateria as ligações entre elas podem ser em série, paralelo ou série paralelo. A escolha do tipo de agrupamento deve ser levantada levando em conta a tensão e corrente que se deseja alcançar.

Para o uso em sistemas FV, recomenda-se as baterias estacionárias, AGM (*Absorbed Glass Mat*), do português manta de fibra de vidro, OpzS, ou ventiladas e VRLA (*Valve Regulated Lead Acid*). Não é recomendado o uso de baterias automotivas em sistemas FV, isso porque elas não suportam ciclo de descarga completo e dependem do movimento do veículo para otimizar o seu funcionamento realizando a homogeneização da sua solução, diferentemente da estacionária que foi projetada para ficar imóvel e suportar centenas de ciclos de descarga e recarga.

**Figura 7 – Baterias estacionárias Heliar DF2500 em paralelo**



Fonte: elaborado pelos autores (2018).

### **3.4.6 Medidor de Vazão**

O medidor de vazão é um instrumento que consegue medir o fluxo ou vazão de um determinado líquido. No caso do incontrol PRO 1000, ele é um medidor de vazão do tipo eletromagnético, ou seja, ele não possui peças móveis e tem seu funcionamento baseado na lei de Faraday, que deduz a velocidade do condutor a partir da tensão induzida por ele através de seu movimento em ângulos retos ao longo de um campo magnético, pois o movimento ao longo das linhas de campo magnético resultam em uma força eletro motriz (*f.e.m*), e essa força é proporcional a velocidade do condutor, no caso a água. Nesse tipo de medidor, por se tratar de um princípio eletromagnético, ele se restringe a fluidos eletricamente condutivos (INCONTROL, [S.d]).

Para descobrir a tensão induzida  $U$ , utiliza-se a expressão referente a segunda lei de Faraday:

- $U = K \times B \times V \times D$

Onde:

- $U = f.e.m$
- $K$  = constante do instrumento
- $B$  = intensidade do campo magnético
- $V$  = velocidade média do fluxo
- $D$  = distância entre os eletrodos

**Figura 8 – Medidor de vazão eletromagnético incontrol PRO1000**



Fonte: elaborado pelos autores (2018).

### **3.4.7 Bomba imersa vibratória**

A bomba d'água, tem a função de exercer uma força sobre a água , com o propósito de realizar um deslocamento para um determinado lugar, geralmente, tubos e conexões hidráulicas realizam essa delimitação de deslocamento, esses equipamentos podem ser utilizados para diferentes finalidades, como transferência de água de um poço artesiano para um reservatório ou caixa d'água, ou para sucção de um grande volume de líquido, como água da chuva empoçada. As bombas vibratórias, funcionam com o princípio do magnetismo para realizar vibrações, essas vibrações ocorrem quando um eletroímã atrai e repele uma estrutura metálica fixada a um diafragma, com essa movimentação, válvulas realizam a sucção da água e ejetam ela do reservatório onde a bomba se encontra para outra unidade de armazenamento.

**Figura 9 – Bomba imersa vibratória Anauger R100**



Fonte: elaborado pelos autores (2018).

### **3.4.8 Conectores MC4**

São os conectores padronizados para módulos FV, desenvolvimento da empresa alemã, Multi-Contact. O nome MC4 é um acrônimo do nome da fabricante mais a identificação com o número 4 que dá referência ao diâmetro do pino de contado interno. O grande diferencial destes conectores, é a sua robustez, travamento automático, e resistência a intempéries, radiação ultravioleta (UV) e umidade.

**Figura 10 – Conectores MC4**



Fonte: elaborado pelos autores (2018).

### **3.5 Unidade FV utilizada como elemento de pesquisa deste trabalho**

O sistema de monitoramento e aquisição de dados implementado neste trabalho foi desenvolvido com base na análise da unidade fotovoltaica instalada na Faculdade de Engenharia (FAEN) da Universidade Federal da Grande Dourados (UFGD), localizada em Dourados-MS, com coordenadas UTM de latitude 22°11'56" Sul e longitude 54°55'56" Oeste, com elevação de 437m. Segundo COORD. TIBA (2001), a região de Dourados possui uma média anual de índice de radiação solar global diária de 18MJ/m<sup>2</sup>. dia e média anual de insolação diária de 6 h. No centro-oeste brasileiro, abrangendo os estados de Goiás, Mato Grosso e Mato Grosso do Sul, o índice de radiação global média é de 5,7kWh/m<sup>2</sup>, já quanto a temperatura, Dourados possui uma temperatura média anual de 24°C e humidade média anual de 78% (PEREIRA et al., 2006).

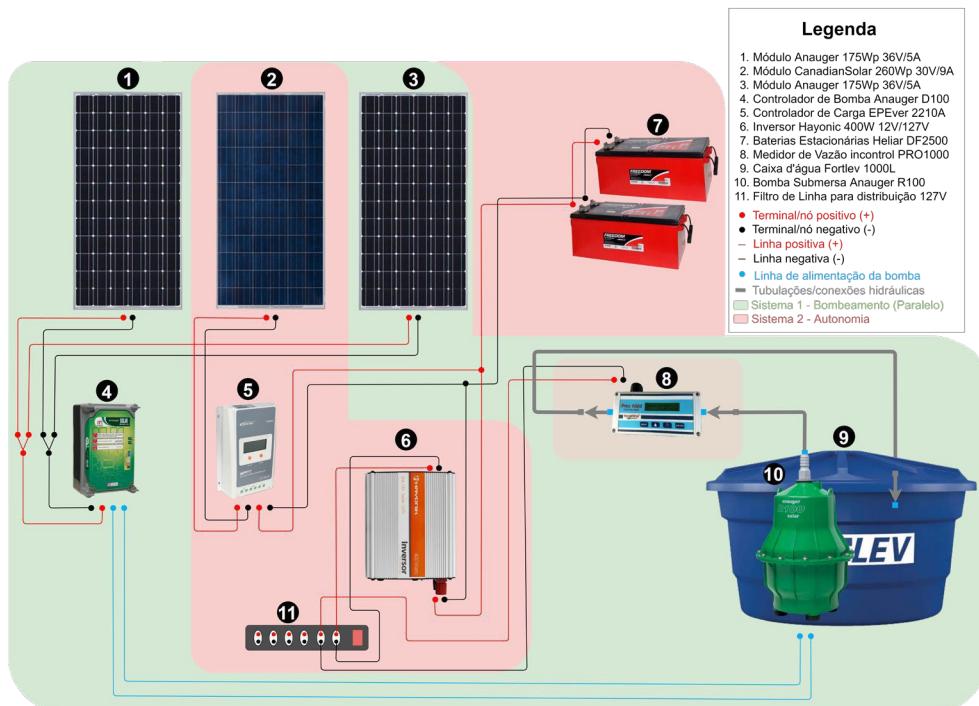
**Figura 11 – Unidade de bombeamento de água FV (FAEN-UFGD)**



Fonte: elaborado pelos autores (2018).

O diagrama da Figura 12, representa de forma ilustrativa a composição da unidade FV utilizada como objeto de estudo para o desenvolvimento do sistema de aquisição de dados. Tal unidade é composta por dois sistemas fotovoltaicos, um sistema autônomo e o outro de bombeamento de água, ambos explicados nas subseções anteriores.

**Figura 12 – Diagrama completo da unidade FV (FAEN-UFGD)**



Fonte: elaborado pelos autores (2018).

Especificação do sistema ilustrado na Figura 11:

1. Módulo FV de silício monocristalino Anauger 175Wp 35,7V/4,9A (SF 125X125-72-M);
2. Módulo FV de silício policristalino CanadianSolar 260Wp 30,4V/8,56A (CS6P-260P);
3. Módulo FV de silício monocristalino Anauger 175Wp 35,7V/4,9A (SF 125X125-72-M);
4. Controlador de Bomba Anauger Driver 100;
5. Controlador de Carga EPEver Tracer 2210A MPPT;
6. Inversor de Onda Modificada Hayonic 400W 12V/127V;
7. Banco com duas baterias estacionárias Hiliar Freedom FD2500 12V (C100-160Ah, C20-150Ah, e C10-130Ah);
8. Medidor de Vazão Eletromagnético de plástico incontrol PRO1000;
9. Caixa d'água Fortlev 1000L;
10. Bomba submersa vibratória Anauger Solar R100;
11. Filtro de linha ou régua de tomadas.

## 4. O SISTEMA DE AQUISIÇÃO DE DADOS

Nesta seção, está descrito as especificações e funções dos principais instrumentos de medição e comunicação, ou seja, instrumentos disponíveis para a aquisição dos dados requisitados para ambos os sistemas FV (bombreamento de água e autônomo), tais como, temperatura, tensão, corrente e vazão. Além disso, abordamos os seguintes assuntos:

- Justificativa da escolha de equipamentos específicos;
- Validação e calibragem de sensores quando necessário;
- A arquitetura do sistema proposto, a nível de *hardware* e *software*;
- Problemas encontrados e as respectivas soluções;
- Otimizações empregadas para o bom uso e facilitação durante instalação.

### 4.1 Descrição dos materiais e equipamentos envolvidos

A seguir serão explicados os diversos componentes que fazem parte da aquisição de dados do sistema de bombeamento fotovoltaico, da FAEN - UFGD.

#### 4.1.1 Sensor de temperatura

Para aferirmos as temperaturas dos módulos fotovoltaicos utilizamos os sensores de temperatura DS18B20 da Dallas que operam na faixa de medição de  $-55^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$  e possui precisão de  $\pm 0,5^{\circ}\text{C}$  entre as faixas de temperatura de  $-10^{\circ}\text{C}$  e  $+85^{\circ}\text{C}$ . O fato de ser prova d'água e possuir uma conexão de dados através de um único fio, onde cada sensor pode ser identificado assim não necessitando utilizar uma entrada analógica para cada sensor, nos poupar tempo e espaço na hora de realizar as conexões entre sensores e micro controlador.

O sensor possui três fios sendo eles GND, Vcc e Sinal, operando na fixa de 3 a 5V.

Figura 13 – Sensores de temperatura Dallas DS18B20



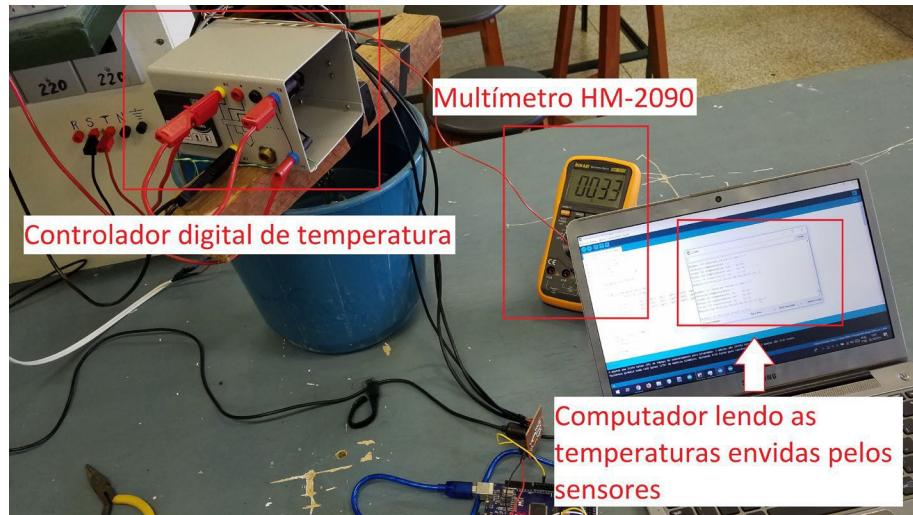
Fonte: adaptado do AliExpress (2018).

Para verificar a precisão dos sensores e se os mesmos estavam calibrados utilizamos o processo de calibração em banho-maria, onde mergulhamos os três sensores na água e fomos esquentando-a gradativamente até certa temperatura e deixávamos na mesma por um período de tempo até estabilizar.

Como não tínhamos os equipamentos profissionais para realizar tal feito utilizamos uma resistência para água de 1000W mergulhada em um balde juntamente com os três sensores DS18B20 mais o sensor de temperatura do multímetro HM-2090 para fins de confirmação das leituras. Assim as temperaturas lidas nos sensores eram enviadas para o computador em tempo real, onde pudemos ter uma comparação rápida e precisa das leituras informadas pelos sensores.

Para realizar o controle de temperatura no banho-maria utilizamos um controlador de temperatura digital onde foi possível configurar a temperatura desejada, assim quando seu sensor que introduzimos no balde junto com os demais sensores atingisse a temperatura configurada a resistência era desligada, e quando fosse detectado que a temperatura começou a diminuir, ele reativada a resistência.

**Figura 14 – A imagem abaixo exemplifica o processo descrito acima**



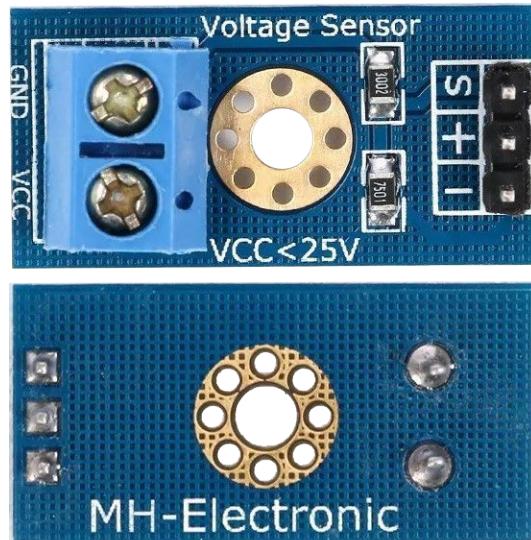
Fonte: elaborado pelos autores (2018).

Ao final de 1h com temperatura a 60°C foi possível validar a precisão dos sensores, nenhuma das três unidades apresentou temperatura fora do normal durante os testes.

#### **4.1.2 Sensor de Tensão**

Este sensor funciona como um divisor de tensão com proporção de 5:1 com dois resistores do tipo *SMD* com tolerância de 1%, sendo um resistor de  $30\text{k}\Omega$  e o outro de  $7,5\text{k}\Omega$ , assim quando a tensão for de 25V teremos um valor na saída de 5V, portanto a tensão pode ser lida pelo Arduino em sua entrada analógica. Para adquirirmos as grandezas de tensão gerada pelos dois sistemas FV, referente aos três módulos, optamos por utilizar o sensor de tensão DC 0 – 25V igual a imagem abaixo.

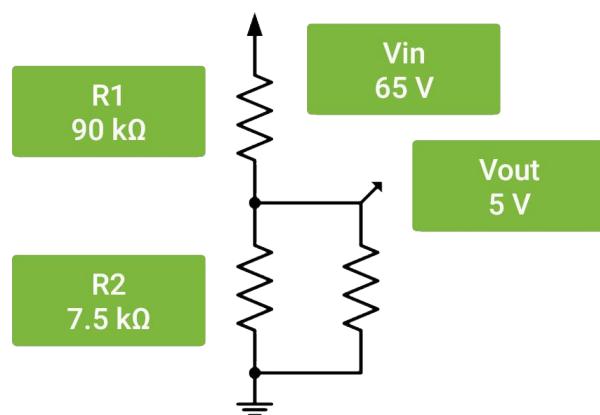
**Figura 15 – Sensor de tensão Vcc < 25V (divisor de tensão)**



Fonte: adaptado do MercadoLivre (2018).

Até o momento da instalação dos sensores em campo, tínhamos a informação de que o sistema (1) de bombeamento fotovoltaico estava associado em série, logo a soma das tensões dos dois módulos FV seria maior do que poderíamos medir, assim foi necessário modificar o sensor para fazer leituras de maior tensão, acrescentando uma resistência de  $60\text{k}\Omega$  em série com uma de  $30\text{k}\Omega$ , totalizando  $90\text{k}\Omega$ , assim poderíamos realizar leituras de tensão de até 65V, como mostrado na imagem a seguir.

**Figura 16 – Aplicativo ElectroDroid**



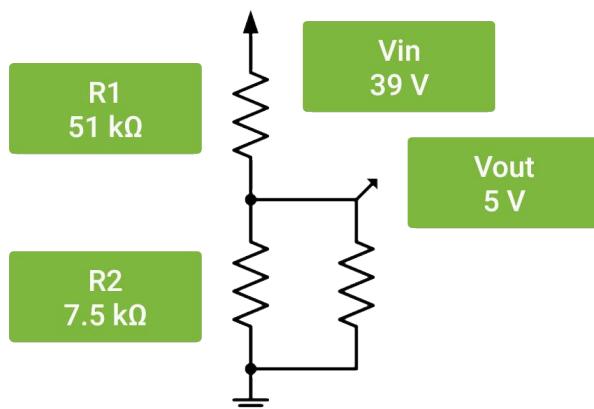
Fonte: adaptado pelos autores a partir do aplicativo ElectroDroid (2018).

Para obtermos uma saída mais próxima de 5V, procuramos valores de tensão e resistência onde o Vout fosse o mais próximo possível de 5V.

No entanto, constatamos que os dois módulos não estavam em série, mas sim em paralelo, logo a tensão permaneceria a mesma e a corrente seria duplicada, já que os módulos são idênticos. Como já havíamos montado os sensores com as resistências modificadas e não haveria problema para realizar as leituras, optamos por manter o divisor de tensão para entradas de até 65V, assim ele suportará ambas configurações, em série ou paralelo.

Também foi necessário fazer ajustes para leitura do sistema (2) fotovoltaico autônomo, isso porque a tensão de potência máxima deste módulo FV pode chegar a 30,4V. Neste caso foi utilizado dois resistores em série, um com resistência de 20kΩ Ohms e outro de 1kΩ totalizando 21kΩ mais a associação em série com o resistor já proveniente do sensor, um resistor de 30kΩ, com isso resultamos em 51kΩ que permitem assim conseguimos realizar leituras de até 39V para este sistema (2).

**Figura 17 – Aplicativo ElectroDroid**

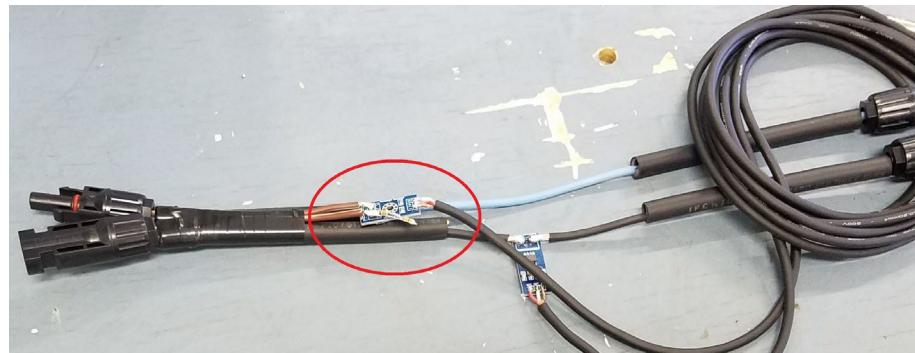


Fonte: adaptado pelos autores a partir do aplicativo ElectroDroid (2018).

Os sensores foram instalados em um cabo PP de 4mm com conectores MC4 que utilizamos tanto para realizar as leituras de tensão, quanto de corrente.

Foi utilizado um par de fios soldados na parte metálica dos conectores MC4 que serviram de extensor até a entrada do sensor. A extensão posteriormente foi inserida em campo para realizar as leituras. Para melhor organização e compactação do cabo extensor foi necessário remover os bornes de conexão e soldar os fios diretamente no módulo FV. Logo a saída dos sinais dos sensores foi adicionado conectores do tipo P2 para facilitar a conexão e ou possível substituição em campo.

**Figura 18 – Extensor com sensor de tensão destacado**



Fonte: elaborado pelos autores (2018).

Para validação dos sensores realizamos testes em laboratório afim de aferir a precisão dos sensores e caso necessário realizar os ajustes que cabíveis a cada sensor. Para isso identificamos os sensores de 1 a 3 (Realizamos a validação de três sensores, mas só utilizamos dois no projeto, o terceiro calibraramos para caso um dos sensores viesse a falhar ou apresentar problemas).

Utilizamos para a calibração dos sensores um multímetro Hikari HM-2090 e um multímetro *Instrutherm* MDA-235 juntamente com uma fonte de bancada Minipa MPL-3303M de dois canais, onde foi adicionado 30V no canal 1 e 9V no canal 2, totalizando 39V.

Utilizamos o multímetro MDA-235 para aferir a soma dos dois canais de tensão da fonte de bancada e o HM-2090 para fazer a leitura da saída do sensor de corrente que estava sendo enviada para o Arduino Mega 2560 que estava configurado para receber os valores lidos e converter para 0 a 39V.

**Figura 19 – Validação e calibração dos sensores com multímetros**



Fonte: elaborado pelos autores (2018).

**Figura 20 – Leituras de tensão obtidas pelo Arduino Mega 2560**

```
Voltage: 38.96V
Voltage: 39.00V
Voltage: 38.84V
Voltage: 39.00V
Voltage: 39.00V
Voltage: 39.00V
Voltage: 39.00V
Voltage: 38.92V
```

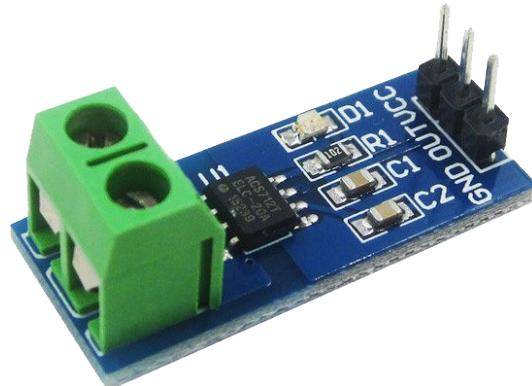
Fonte: elaborado pelos autores (2018).

Os três sensores calibrados apresentaram os mesmos resultados de precisão, tanto para baixas tensões quanto para tensão máxima de operação, logo não foi necessário realizar nenhuma alteração ou ajuste para corrigir possíveis imprecisões dos sensores de tensão.

#### **4.1.3 Sensor de Corrente**

Para realizarmos as leituras de corrente utilizamos dois sensores de corrente modelo ACS712 que operam de  $-20\text{A}$  a  $+20\text{A}$  e possui precisão de 1,5% a  $25^\circ\text{C}$ , como saída emitem uma tensão de 0 a 5V proporcional a entrada. Pelo fato dos módulos FV produzirem corrente **continua** não foi possível utilizar sensores não invasivos. O limite máximo de 20A é mais que o suficiente para realizar a leitura dos dois módulos, já que a corrente máxima individual para cada painel fotovoltaico é de 4,9A.

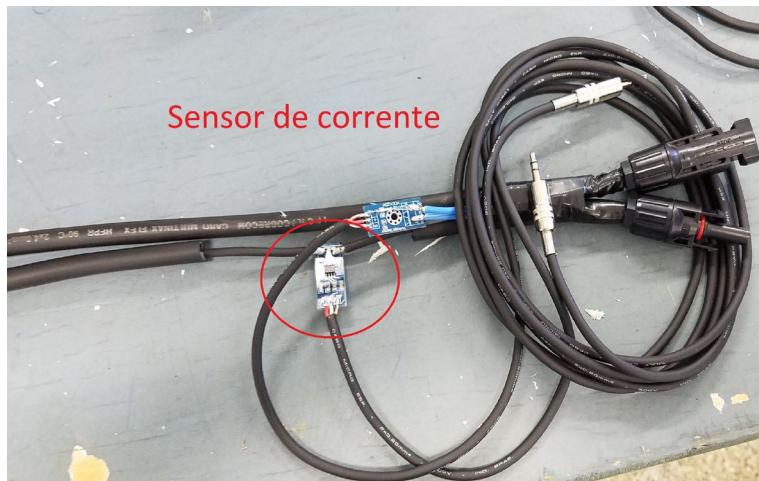
**Figura 21 – Sensor de Corrente Allegro ACS712 20A**



Fonte: adaptado do AliExpress (2018).

Os dois sensores foram adicionados na mesma extensão que adicionamos o sensor de tensão, porém para aferir a corrente foi necessário abrir o circuito no cabo positivo e soldar a placa do sensor sem os bornes de conexões originais, e na saída do sensor foi adicionado um conector do tipo P2 para facilitar a instalação e possível substituição em campo.

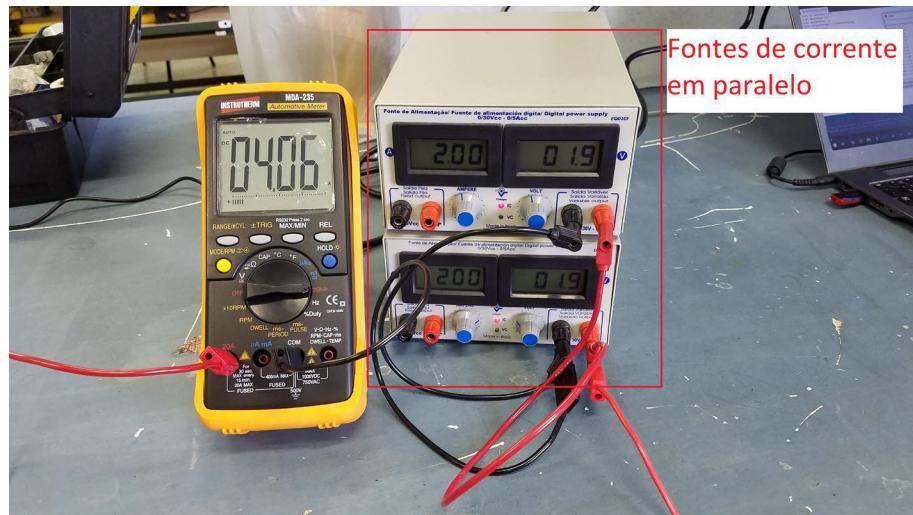
**Figura 22 – Imagem com sensor de corrente destacado**



Fonte: elaborado pelos autores (2018).

Para certificarmos que os sensores de corrente estavam devidamente calibrados realizamos testes em laboratório para simular diversos valores de leituras, desde correntes mais baixas até mais elevadas. Para isso utilizamos um multímetro da *Instrutherm* modelo MDA-235 e duas fontes de bancadas digitais 0 – 30Vcc e 0 – 5Acc em paralelo para somarmos as correntes, já que com apenas uma fonte ficaríamos limitados a apenas 5A.

**Figura 23 – Fontes de corrente em paralelo**



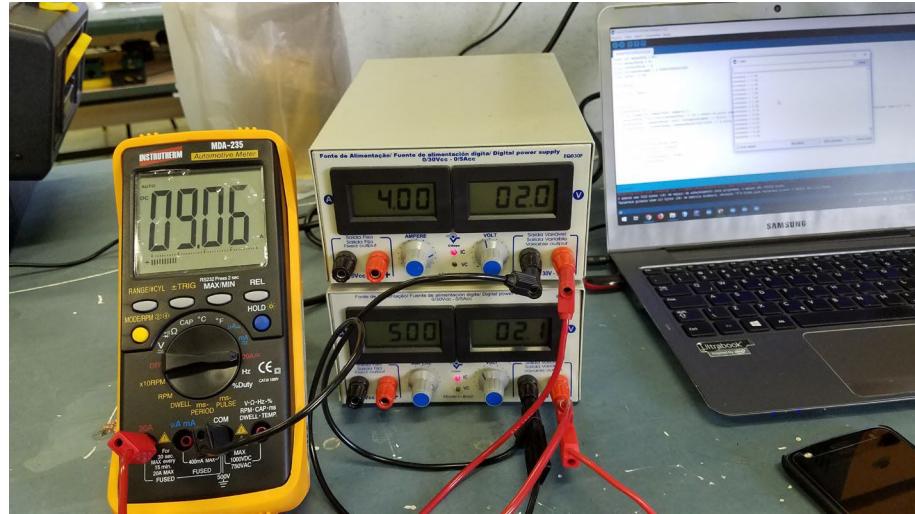
Fonte: elaborado pelos autores (2018).

**Figura 24 – Leituras de corrente obtidas pelo Arduino Mega 2560**

```
corrente = 2.52
corrente = 3.00
corrente = 3.34
corrente = 3.56
corrente = 4.04
corrente = 4.05
corrente = 4.06
corrente = 4.04
corrente = 4.03
corrente = 4.03
corrente = 4.03
corrente = 4.03
corrente = 4.04
corrente = 4.05
corrente = 4.04
```

Fonte: elaborado pelos autores (2018).

**Figura 25 – Fontes de corrente totalizando 9A**



Fonte: elaborado pelos autores (2018).

Juntamente com o multímetro para auxiliar as aferições das leituras comparamos os valores com o que era apresentado pelo Arduino Mega 2560 no computador.

Para melhor precisão dos sensores foi necessário realizar um conjunto de aferições e realizar a média das mesmas, neste caso realizamos 500 medições e dividimos por 500, como recomendações encontradas em fóruns na internet. Assim garantímos maior precisão nas leituras, como pode ser visto nas imagens a cima. Assim como os sensores de tensão não foi necessário realizar outros tipos de ajustes para os dois sensores de corrente testados.

#### **4.1.4 Medidor de Vazão**

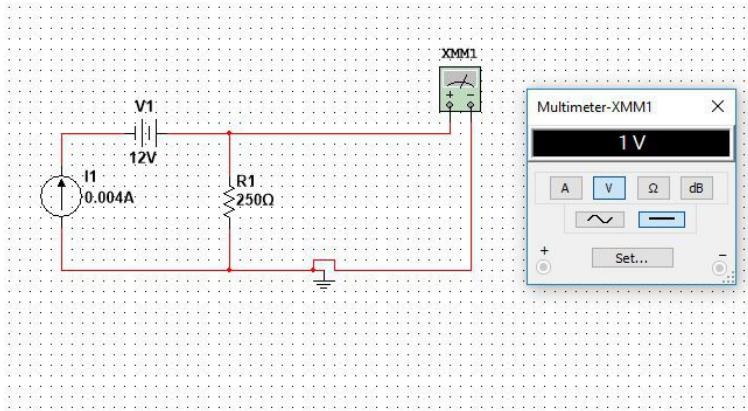
O medidor (computador) de vazão disponível na unidade FV da FAEN, é o modelo PRO1000 da incontrol, no qual para realizarmos as leituras do medidor utilizamos a sua saída de 4 – 20mA, onde configuramos o medidor seguindo a instrução do fabricante. Após seguir os passos de configuração proposto pelo fabricante ajustamos o nível máximo de vazão para 10 m<sup>3</sup>/h, já que em teste a vazão não ultrapassou os 5 m<sup>3</sup>/h. Por segurança foi decidido adotar 10 m<sup>3</sup>/h para não ter falhas nas medições.

Como o Arduino não realiza leituras de corrente, precisamos montar um pequeno circuito onde fosse possível realizar as leituras em Volt, para isso foi necessário acrescentar uma resistência de 250Ω em paralelo a saída de corrente do

medidor, desta forma como sabemos que a corrente sempre será de 4mA quando a vazão for 0 m<sup>3</sup>/h e 20mA quando for 10 m<sup>3</sup>/h. Com uma resistência de 250Ω sempre teremos 1V a 4mA e 5V quando a 20mA.

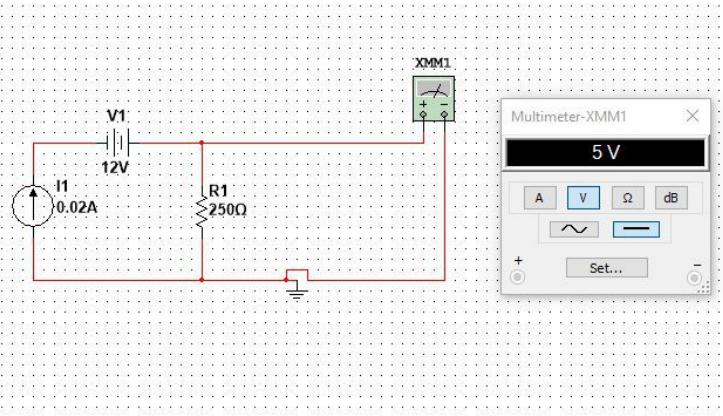
Independente da tensão da fonte a saída será sempre de 1 a 5V de acordo com a lei de Ohm, onde Tensão = Resistência × Corrente ( $U = R \times I$ ), logo  $V = 250\Omega \times 0,004A = 1V$  e  $V = 250\Omega \times 0,02A = 5V$ .

**Figura 26 – Exemplo da saída de tensão: vazão igual a 4mA (0 m<sup>3</sup>/h)**



Fonte: elaborado pelos autores (2018).

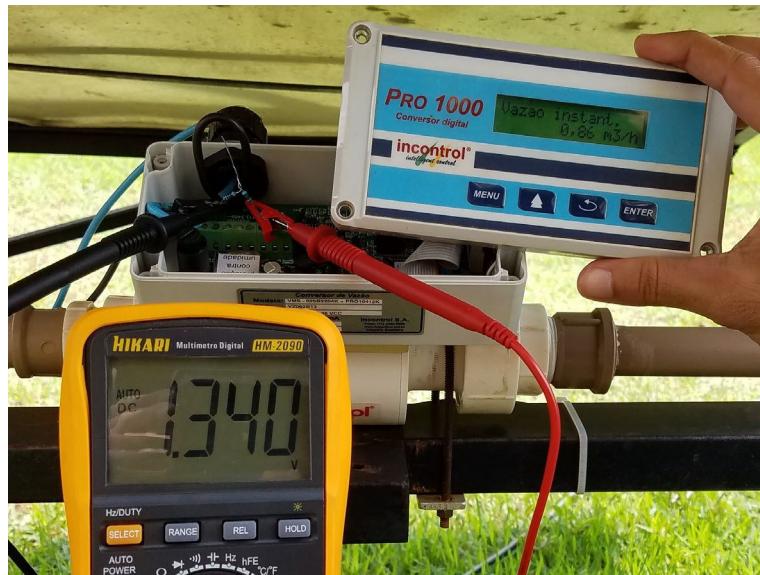
**Figura 27 – Exemplo da saída de tensão: vazão igual a 20mA (10 m<sup>3</sup>/h)**



Fonte: elaborado pelos autores (2018).

A nível de código, foi necessário realizar uma regra de proporcionalidade, relacionando os 4mA a 0 m<sup>3</sup>/h, e os 20mA aos 10 m<sup>3</sup>/h, no Arduino foi possível utilizar a função map para realizar a proporcionalidade, para obtermos as leituras com precisão conforme a imagem obtida em campo.

**Figura 28 – Teste de comparação/equivalência do sinal de 4 – 20mA**



Fonte: elaborado pelos autores (2018).

A função map ficou da seguinte forma:

- vazao = map(analogVazao, 204.6, 1023.0, 1023.0, 0.0, 10.0);

Sabendo que as entradas analógicas do Arduino trabalham com 10bits, sendo 0bits equivalente a 0V e 1023bits a 5V, cada 1V representa 204.6bits, logo 1V está para 0 m<sup>3</sup>/h assim como 5V estão para 10 m<sup>3</sup>/h.

No medidor de vazão também não foi necessário realizar ajustes de correção nas medidas, já que os valores apresentados estão muito próximos do valor real, assim dispensando algum tipo de ajuste por deslocamento, do inglês *offset*, caso as medidas diferissem do esperado.

#### **4.1.5 Transceptor nRF24L01**

O nRF24L01 é um dispositivo de comunicação transceptor, ou seja, ele é capaz de operar tanto para transmissão quanto para recepção, característica que lhe torna bidirecional. Ele foi projetado para trabalhar sob a faixa de frequência de 2,4GHz de acordo com a *ISM (Industrial, Scientific and Medical Radio Bands)*, em português, bandas de rádio industriais, científicas e médicas. Sua configuração de comunicação é baseada no protocolo de interface serial periférica (*SPI*), que permite inúmeros dispositivos se comunicarem através de quatro barramentos (*MISO*, *MOSI*, *SCK* e

SS), fazendo uso do princípio de mestre e escravo. A taxa de dados deste dispositivo é de 2Mbps (250KB/s), o comprimento máximo de mensagem corresponde a 32 bytes, ou seja, 32 caracteres, entretanto 1 byte destes 32 bytes é reservado para o fim da mensagem que corresponde a um campo em branco, já o seu alcance máximo, sob condições ideais e sem obstáculos pode chegar a 1Km utilizando antena com ganho de 2dBi. Quanto a alimentação, sua faixa de trabalho é de 1,9 a 3,6V, possuindo regulador de tensão integrado, e por ser um dispositivo sensível a tensão, para melhores alcances deve trabalhar próximo ao seu limite máximo de tensão. A média de consumo deste dispositivo enquanto transmite ou recebe mensagens é de 8,2mA, e enquanto está em modo ocioso a média é de 300 $\mu$ A (NORDIC, [S.d]).

**Figura 29 – Transceptor nRF24L01 +PA +LNA 2,4GHz**



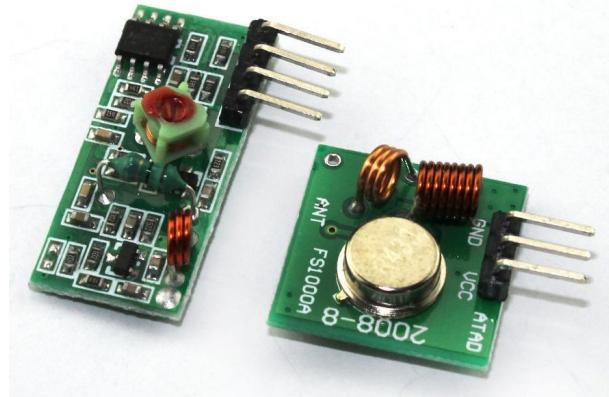
Fonte: adaptado do AliExpress (2018).

Para a finalidade de comunicação dos dados, também estudamos um outro dispositivo que é popularmente conhecido como RF 433Mhz, nome que é dado pela sua operação sob a frequência de 433,92MHz, ele consiste em um conjunto de 2 elementos, o transmissor (TX) e o receptor (RX), o que lhe classifica como dispositivo unidirecional. A sua taxa de transferência de dados corresponde a 16kbps/s (2KB/s), já o alcance máximo, sob condições ideais e sem obstáculos pode chegar a 200m. Quanto a alimentação, a faixa de trabalho dele é de 3,5 a 12V, e para melhores alcances também deve trabalhar próximo ao seu limite máximo de tensão. A média de consumo deste dispositivo é de 10mW (MANTECH, [S.d]).

Em nossos testes de alcance, o nRF24L01 conseguiu a marca de 300m, já o RF 433MHz chegou no máximo a 100m. Por termos acesso e possibilidade de uso de ambos os dispositivos, constatamos que além de melhor alcance, o nRF24L01 possui

maior estabilidade e confiabilidade do que o RF 433Mz, o que nos levou a optar pelo uso do primeiro.

**Figura 30 – Módulos transmissor e receptor RF 433MHz**



Fonte: adaptado do AliExpress (2018).

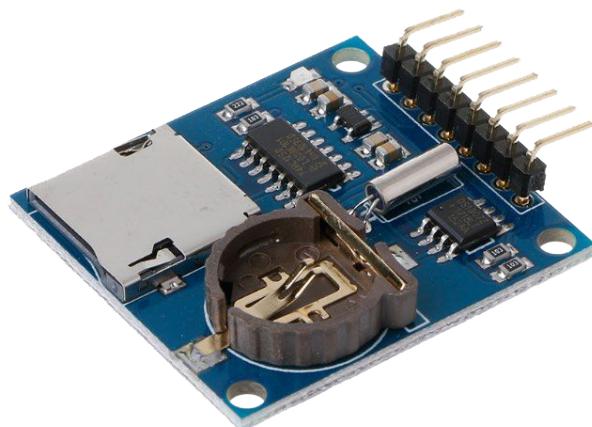
#### **4.1.6 Módulo Datalogger + SD + RTC DS1307**

O módulo *datalogger* consiste em uma proposta que une os dois recursos básicos para fazer registro de dados, uma interface de armazenamento e um relógio de tempo real (*RTC*), que mantém salva as propriedades de data e hora por anos com o uso de uma bateria de 3V. Nesse módulo temos um controlador de cartão de memória que se comunica com o Arduino utilizando o protocolo *SPI*, assim como o nRF24L01 e permite o uso de cartões SD formatados em FAT16 e FAT32, e trabalha com alimentação de 3,3V, para fazer o controle do SD esse módulo utiliza geralmente o controlador de barramento quadruplo (*Quad Bus Buffer*) 74HC125D. O *RTC* utilizado é o DS1307, um relógio muito utilizado e conhecido que trabalha com tensão de 5V e é projetado para funcionar com bateria de 3V, que no caso deste módulo é do formato CR1220. O relógio se comunica com o Arduino utilizando o protocolo I2C, que consiste basicamente em controlar múltiplos dispositivos utilizando somente dois barramentos (SDA, SCL).

Para a finalidade de armazenamento de dados, também estudamos o uso de um conjunto de outros dois dispositivos, o módulo SD convencional que utiliza o controlador de barramento LVC125A e o RTC DS3231, porém devido a uma falha de projeto no comutador de nível (tensão) do módulo SD convencional, ele mantinha o

controlador de barramento sempre ativo e gerava uma incompatibilidade funcional com o nRF24L01 que também utiliza o barramento *SPI*, ou seja, ele provocava algo semelhante a um congestionamento do canal de comunicação, impedindo o funcionamento do transceptor com o impedimento do pino seletor (*chip select*). Por conta disso, optamos pelo uso do módulo *datalogger*, que faz uso do controlador 74HC125D e funcionou normalmente em conjunto com o nRF24L01.

**Figura 31 – Módulo Datalogger + SD + RTC DS1307**

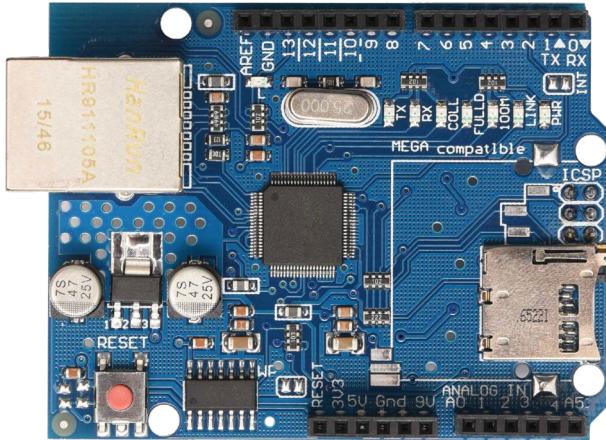


Fonte: adaptado do AliExpress (2018).

#### **4.1.7 Ethernet Shield W5100**

Com a finalidade de permitir conectividade com a internet, optamos pela *Ethernet Shield* W5100, que consiste em uma adaptação feita no molde dos conectores do Arduino Uno, mas que possui compatibilidade com o Arduino Mega 2560. O grande diferencial deste dispositivo é a integração do módulo leitor de cartão de memória (*SD card*) e do controlador de rede Wiznet W5100 que suporta até quatro conexões sockets simultâneas. Tanto o leitor de cartão quanto a conexão *Ethernet* se comunicam com o Arduino pelo protocolo *SPI*, o mesmo do nRF24L01, e no caso dessa *Ethernet Shield*, não foi observado problemas de projeto eletrônico que provocasse incompatibilidades (MOUSER, [S.d]).

**Figura 32 – Ethernet Shield W5100**



Fonte: adaptado do AliExpress (2018).

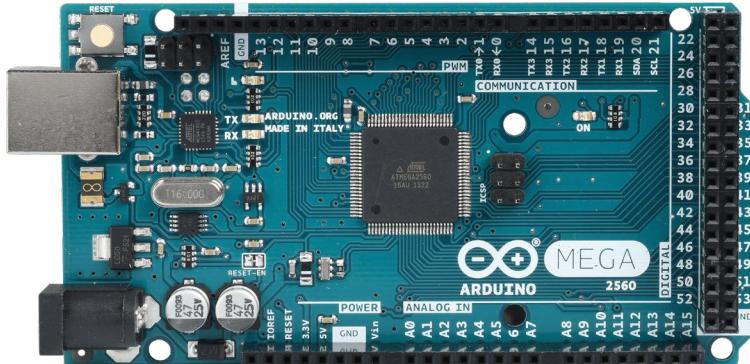
#### **4.1.8 Arduino Mega 2560**

O Arduino consiste em uma placa de prototipagem que utiliza na maioria dos seus modelos microcontroladores Atmel e o conceito de *software e hardware livre*, utilizando como licenças LGPL (Licença Pública Geral Menor) e Creative Commons.

No caso do Arduino Mega, temos um microcontrolador ATmega2560 com 54 pinos digitais, 16 pinos analógicos e quatro portas seriais UART (Receptor/Transmissor Assíncrono Universal), sua capacidade de armazenamento de programação (*flash*) é de 256KB. Ele é programado utilizando uma *IDE* (ambiente de desenvolvimento integrado) própria, que reconhece C/C++.

Para utilização como transmissor e dispositivo de aquisição de dados, optamos por utilizar o Arduino Mega 2560 devido ao seu grande número de portas analógicas e a sua capacidade de armazenamento *flash*, que para códigos-fonte extensos e pesados é um recurso fundamental. Já para utilização como receptor e dispositivo de envio dos dados para nuvem (servidores virtuais), inicialmente optamos pelo Arduino Uno, entretanto devido a sua capacidade oito vezes inferior à do Arduino Mega 2560, tivemos de mudar a nossa escolha, visto que o código-fonte do dispositivo receptor extrapolou os 32KB de memória *flash*.

**Figura 33 – Arduino Mega 2560**



Fonte: adaptado do AliExpress (2018).

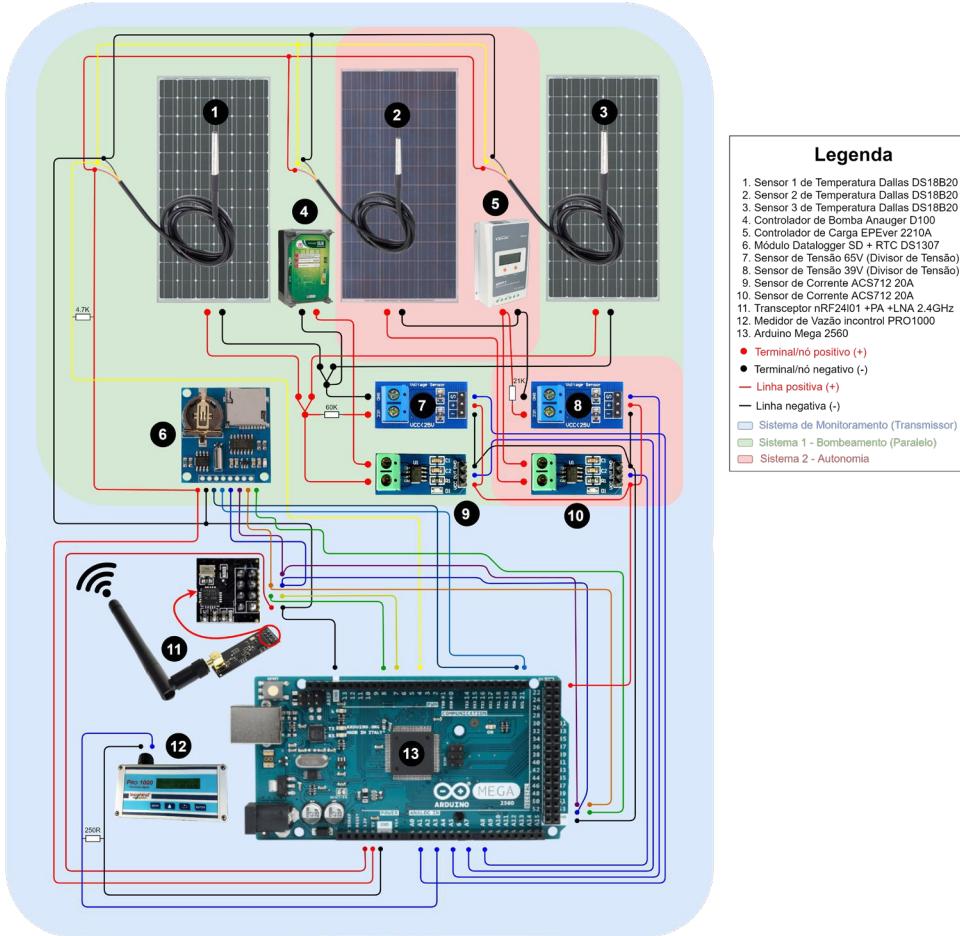
## 4.2 Arquitetura do sistema a nível de hardware

A seguir, será narrado o funcionamento do sistema de aquisição e monitoramento proposto, na perspectiva de *hardware*.

### 4.2.1 Transmissor (*Arduino instalado em campo*)

O sistema transmissor é representado pela Figura 33, ele consiste em um Arduino Mega 2560 que realiza aquisição das temperaturas dos três módulos FV através do sinal analógico oriundo dos sensores DS18B20, para a correta aquisição desse sinal referente a temperatura, é preciso inserir um resistor de  $4,7\text{k}\Omega$  entre a linha de sinal e o Vcc, esse sensor possui somente um barramento para comunicação dos sensores, trata-se do protocolo *OneWire*, onde cada sensor possui um código identificador único que deve ser usado a nível de código para referenciá-lo. Além da aquisição de temperatura, esse Arduino faz a leitura de dois conjuntos de sensores de corrente e tensão, cada um com variáveis elétricas referente aos sistemas FV de bombeamento e autônomo, referenciados como sistema 1 e 2 respectivamente. Por fim o Arduino faz a leitura do sinal de 4 – 20mA do medidor de vazão e registra todas essas informações no cartão de memória inserido no módulo *datalogger*, para finalizar o ciclo, ele faz comunicação com o outro Arduino, o receptor, e envia os dados em forma de uma *String*, cadeia de caracteres.

**Figura 34 – Arquitetura do sistema de monitoramento: transmissor**

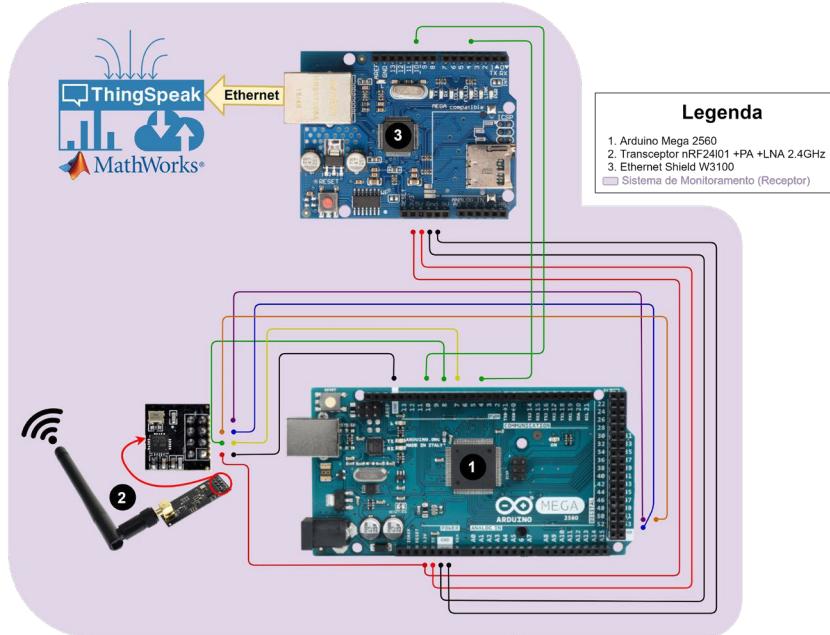


Fonte: elaborado pelos autores (2018).

#### 4.2.2 Receptor (Arduino instalado em sala no prédio da FAEN)

O sistema receptor, esquematizado na Figura 31, consiste em um Arduino Mega 2560 que fica escutando o canal de comunicação com o outro Arduino aguardando a chegada de uma mensagem contendo os dados adquiridos, após o recebimento destes dados, o Arduino faz a redundância dos dados, realizando um novo registro em um cartão de memória inserido no *Ethernet Shield W5100*, após os dados estarem salvos, o ciclo se encerra com o envio dos dados para o servidor do ThingSpeak, que permitirá o acesso remoto a tudo o que foi coletado.

**Figura 35 – Arquitetura do sistema de monitoramento: receptor**



Fonte: elaborado pelos autores (2018).

#### 4.3 Arquitetura do sistema a nível de software

Adiante será explicado brevemente o funcionamento do código-fonte (*software*) implementado para ambos os sistemas, transmissor e receptor. Ambos os códigos estão disponíveis nos Apêndices deste trabalho, A e B respectivamente.

##### 4.3.1 Programação do transmissor

O código-fonte do transmissor pode ser resumido em sete funções fundamentais, a primeira referente a aquisição do dia da semana, data e hora, a segunda referente a aquisição das três temperaturas e tensão do sistema FV 1, a terceira referente a aquisição da tensão do sistema FV 1, tensão e corrente do sistema FV 2 e vazão, a quarta referente ao envio dos dados para o outro Arduino, a quinta referente a separação dos dados do vetor de caracteres para variáveis independentes para calcular a potência (resultante da multiplicação da tensão pela corrente), a sexta função calcula a média dos últimos 15 segundos, e enfim a sétima função faz o registro dos dados em um arquivo de planilha eletrônica no formato CSV (valores separados por vírgula).

Nomes das funções supracitadas:

- 1<sup>a</sup> Função: getData();
- 2<sup>a</sup> Função: getSensores1();
- 3<sup>a</sup> Função: getSensores2();
- 4<sup>a</sup> Função: sendLog();
- 5<sup>a</sup> Função: separaDados(&Log);
- 6<sup>a</sup> Função: calculaMedia(&contadorMed, &variável\_auxiliar, variável);
- 7<sup>a</sup> Função: writeLog(LogMed);

#### **4.3.2 Programação do receptor**

O código-fonte do receptor é composto por cinco funções fundamentais, a primeira consiste em manter o Arduino escutando o canal de comunicação com o transmissor afim de receber uma mensagem contendo a cadeia de caracteres referente as leituras dos sensores, a segunda é responsável por separar os dados em variáveis independentes para o posterior cálculo da potência e envio para a nuvem IoT, a terceira função faz o cálculo da média dos últimos 15 segundos, a quarta faz o armazenamento dos dados no cartão de memória, também em formato CSV, por fim a sétima função faz o envio dos dados para um canal no servidor do ThingSpeak.

Nome das funções supracitadas:

- 1<sup>a</sup> Função: receiveLog();
- 2<sup>a</sup> Função: separaDados(&Log);
- 3<sup>a</sup> Função: calculaMedia(&contadorMed, &variável\_auxiliar, variável);
- 4<sup>a</sup> Função: writeLog(LogMed);
- 5<sup>a</sup> Função: sendThingSpeak(&contador, temp1Med, temp2Med, ..., pot2Med)

#### **4.3.3 Outras aplicações utilizadas**

Para o acesso remoto, Optamos em utilizar um servidor gratuito de armazenamento de dados voltado para internet das coisas (IoT), onde fosse possível o envio de dados de maneira simples e que apresentasse os dados de maneira clara e amigável ao usuário, e disponibilizasse formatos de gráficos ajustados por período de tempo ou quantidade de amostras. Como os serviços da plataforma ThingSpeak

atendeu todas nossas principais necessidades para realização do projeto optamos por utilizá-lo.

Para fins de comparação a tabela a baixo demonstra uma comparação do ThingSpeak que foi a plataforma escolhida, com um servidor local para armazenamento de dados *IoT*.

**Figura 36 – Comparação do plano gratuito do ThingSpeak com Servidor Local**

	<b>ThingSpeak Free</b>	<b>Servidor Local</b>
<b>Escalável para projetos maiores</b>	Não	Sim
<b>Mensagens</b>	3 milhões/ano (~8.200/dia)	Ilimitado
<b>Intervalo de tempo</b>	Mínimo a cada 15 segundos	Ilimitado
<b>Suporte técnico</b>	Fórum	Nenhum
<b>Hardware extra</b>	Não	Sim
<b>Conexão com internet</b>	Sim	Sim
<b>Tempo de implementação</b>	Baixo	Alto
<b>Limite total de dados</b>	10 milhões de mensagens	Tamanho do disco
<b>Custo</b>	Nenhum	Alto
<b>Interface de acesso</b>	Sim	Necessário implementação
<b>Manutenção</b>	Nenhuma	Sim

Fonte: elaborado pelos autores e baseado no site do ThingSpeak (2018).

O ThingSpeak é uma plataforma de análise de dados providos de *IoT*, permitindo a visualização e processamento em nuvem. Essa plataforma fornece a possibilidade de análise instantânea dos dados enviados ao seu servidor, com a capacidade de executar códigos MATLAB, realizar a análise e o processamento online dos dados conforme eles são recebidos. O ThingSpeak é frequentemente usado para prototipagem em sistemas *IoT* que exigem algum tipo de análise.

A vantagem de utilizar o ThingSpeak se deve ao fato de existir a possibilidade da utilização da plataforma sem custos desde que não seja utilizada para fins comerciais, o que atende a nossa necessidade de uso, existindo apenas a limitação de quantidade de envios e armazenamento de dados 3 milhões/ano ou 8200 mensagens/dia o que nos dá a possibilidade de enviar mensagens para o servidor com intervalo de tempo de 15 segundos, oito sensores simultâneos e armazenar até 10 milhões de dados, o que equivale a mais de 3 anos de

armazenamento contínuo. Também ficando limitado a não utilização em projetos governamentais.

No sentido de acesso remoto, foi realizada uma breve pesquisa sobre as ferramentas existentes que fornecem esse tipo de acessibilidade, e foi encontrado um aplicativo móvel chamado Virtuino.

O Virtuino é uma plataforma IHM (interface humano máquina) com recursos especiais para IoT, oferecendo suporte a canais do ThingSpeak, onde o usuário consegue sincronizar os dados enviados para o servidor da nuvem com o aplicativo móvel. Realizamos testes com essa aplicação e de forma geral foi muito satisfatório.

Figura 37 – Aplicativo Virtuino executando o template PVmonitor



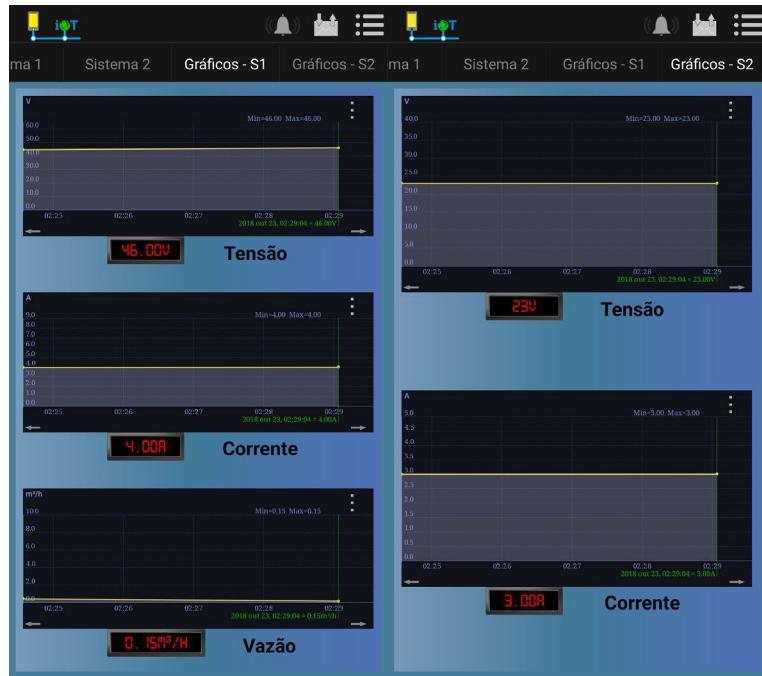
Fonte: elaborado pelos autores (2018).

**Figura 38 – Aplicativo Virtuino com alguns manômetros de dados**



Fonte: elaborado pelos autores (2018).

**Figura 39 – Aplicativo Virtuino com alguns gráficos**



Fonte: elaborado pelos autores (2018).

#### 4.3 Problemas e soluções encontrados

1. Os dados do log devem ser redundantes por motivos de segurança (backup na ausência de energia no Arduino receptor), sendo assim, os dois Arduinos devem estar com os campos sincronizados.
- Enviar informações de data e hora junto com a *String* de dados, pois assim cada linha dos dados será encabeçada por uma data e hora.
2. A mensagem que o transmissor (TX) envia deve atender ao tamanho máximo de 32 caracteres - 1, e a *String* que tínhamos inicialmente era de 69 caracteres.
  - Enviar o volume mínimo necessário de informações, ou implementar uma função que trunca a mensagem em pedaços (*chunks*) de até 32 bytes - 1.
3. Incompatibilidade devido a uma falha de compartilhamento do barramento SPI (SCK-clock, MISO-escravo, MOSI-mestre) entre o transceptor nRF24L01 e o módulo leitor de cartão SD convencional.
  - Aprofundar os conhecimentos na comunicação SPI e no esquema elétrico dos módulos leitores SD, afim de identificar o real problema. Após um certo período, foi constado que havia um defeito de projeto no módulo e a solução viável encontrada foi a substituição dele pelo módulo *datalogger*.

## **4.4 Otimizações e melhorias na instalação**

Adiante apresentamos algumas melhores e otimizações que implementamos neste trabalho, em geral com o propósito de oferecer melhor acabamento e proporcionar maior simplicidade e facilidade nas conexões físicas dos dispositivos.

### **4.4.1 Sensores de temperatura**

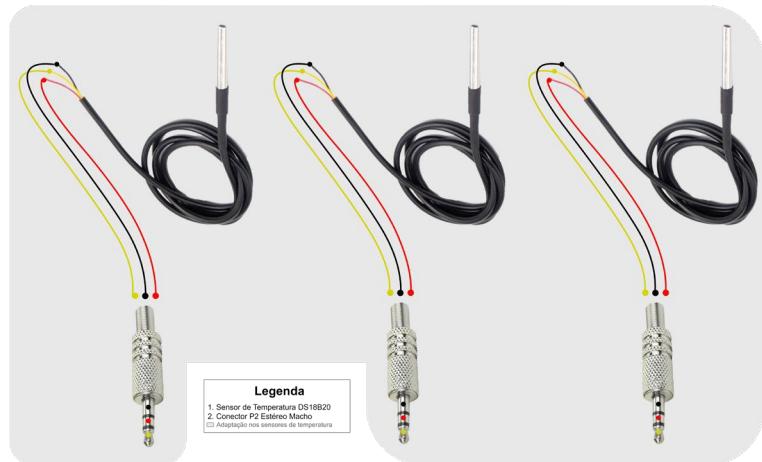
A instalação e conexão com o Arduino e os sensores de temperatura na unidade FV foi facilitada pelo fato de utilizarmos conexões do tipo P2 tanto nos sensores como nas extensões que foram necessárias.

Para identificação dos sensores foram utilizadas cintas plásticas, também conhecidas como abraçadeiras de nylon. Cada sensor recebeu uma, duas ou três cintas próximo a sua base de medição e também a mesma quantidade na outra extremidade. O que facilitou bastante na hora de conectar corretamente nas respectivas entradas/portas.

A fim de melhorar a condução da temperatura e ampliar a área de contato do sensor e o módulo fotovoltaico, utilizamos uma interface térmica entre o sensor e o módulo FV.

Sensor de temperatura com conector do tipo P2 e cintas plásticas para identificação.

**Figura 40 – Diagrama de adaptação P2: DS18B20**



*Fonte: elaborado pelos autores (2018).*

**Figura 41 – DS18B20 adaptado com plug P2 e cintas plásticas para identificação**

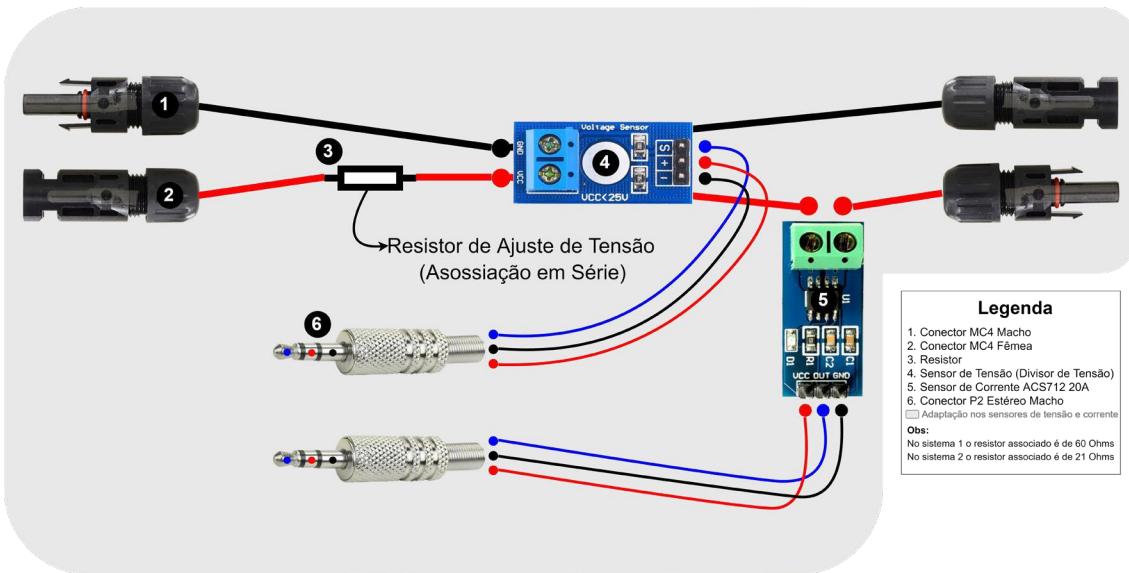


*Fonte: elaborado pelos autores (2018).*

#### 4.4.2 Sensores de tensão e corrente

Os sensores de corrente e tensão foram instalados em uma extensão com pontas MC4 que foram conectadas nos circuitos dos módulos fotovoltaicos.

**Figura 42 – Diagrama de adaptação P2: Tensão e Corrente**



Fonte: elaborado pelos autores (2018).

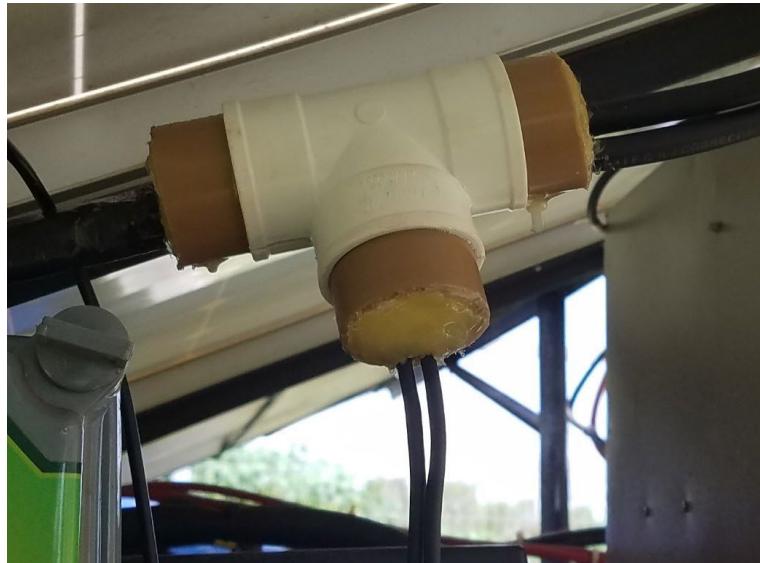
**Figura 43 – Adaptação com extensão para os sensores de corrente e tensão**



Fonte: elaborado pelos autores (2018).

Por necessidade de proteger os sensores, evitando que ficassem expostos e que por algum motivo o manuseio pudesse vir a quebrá-los, decidimos adicionar uma conexão em T de 1"pol para conexão de tubulações de esgoto e nas três extremidades colamos com cola para canos PVC (policloreto de vinila) cerca de 3cm de um tubo hidráulico também de 1"pol para aumentar a área de proteção e ajudar na fixação. Para evitar a entrada de água cortamos um círculo de isopor no tamanho necessário para encaixar dentro do tubo, e inserimos cerca de 1cm para posteriormente preencher com cola quente, assim vedando as três entradas.

**Figura 44 – Proteção das extensões criada para os sensores de corrente e tensão**



Fonte: elaborado pelos autores (2018).

Os cabos necessários para conexão no Arduino saíram pela conexão perpendicular do T.

#### **4.4.3 Fixação dos sensores de temperatura**

A fixação em baixo dos módulos fotovoltaicos foi feita através de uma barra de alumínio com perfil em L fixada em suas extremidades, juntamente uma borracha presa por cintas plásticas no centro da barra, os sensores de temperatura ficaram presos devido a pressão exercida pela barra de alumínio juntamente a borracha.

**Figura 45 – Sensor de temperatura fixado de baixo do módulo fotovoltaica**



Fonte: elaborado pelos autores (2018).

Para garantir que por acidente a conexão dos sensores não fosse desfeita, foi reforçado com fita isolante após o encaixe, ainda garantindo o propósito de facilitar a troca dos sensores e possível manutenção posteriormente, além de proteger contra curto e mal contato.

**Figura 46 – Conexão da extensão e o sensor fixados com a ajuda de fita isolante**

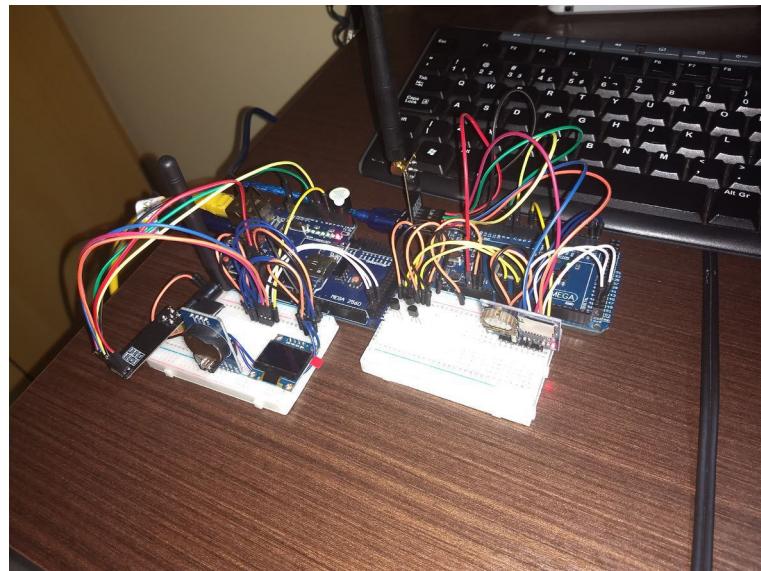


Fonte: elaborado pelos autores (2018).

#### **4.4.4 Prototipação e PCB perfurada (*protoboard*)**

Os testes dos componentes, cálculo de resistores e toda prototipação, além de diversos testes e correção de código foram feitos utilizando uma placa de prototipação, do termo em inglês *protoboard*, e com uso de fios de prototipação, *jumpers*, para ligar dois pontos em um circuito, o que proporcionou mais agilidade nos testes.

**Figura 47 – Protótipo utilizando protoboard**

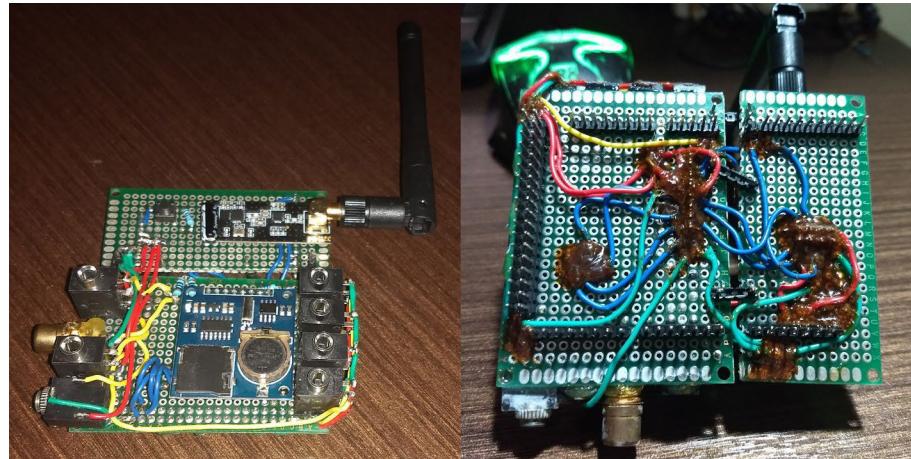


Fonte: elaborado pelos autores (2018).

Após realizarmos todos testes e ajustes necessários utilizamos uma placa de circuito impressa com furos, realizando todas as conexões necessárias com fios e solda, junto aos resistores para cada componente. Para facilitar a conexão com o Arduino foram utilizados pinos que foram soldados sobrepostos as entradas do Arduino, assim o módulo criado com os componentes e conexões podem ser encaixados e desencaixado quando necessário.

Para fortificar as soldas e realizar a isolação foi utilizado cola para junta de motores da 3M, o ideal seria a utilização de fita isolante líquida, mas infelizmente não obtivemos acesso a esse material.

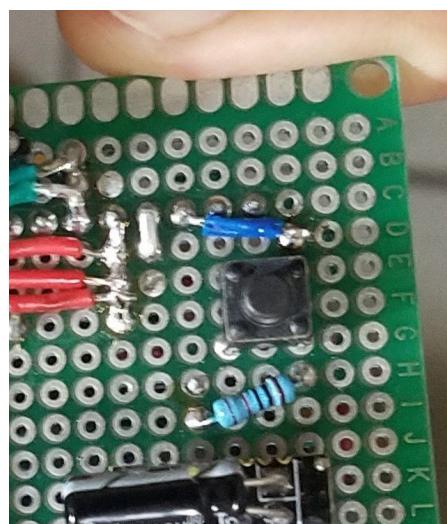
**Figura 48 – PCB adaptada para o Arduino transmissor**



Fonte: elaborado pelos autores (2018).

Como o módulo desenvolvido ocupa toda a parte superior do Arduino Mega 2650, o botão de reiniciar o microcontrolador, *reset*, ficou inacessível, e caso precisássemos utilizá-lo, seria necessário desconectar a fonte de alimentação ou retirar o módulo. Para solucionar esse problema, foi criada uma extensão do botão de *reset*.

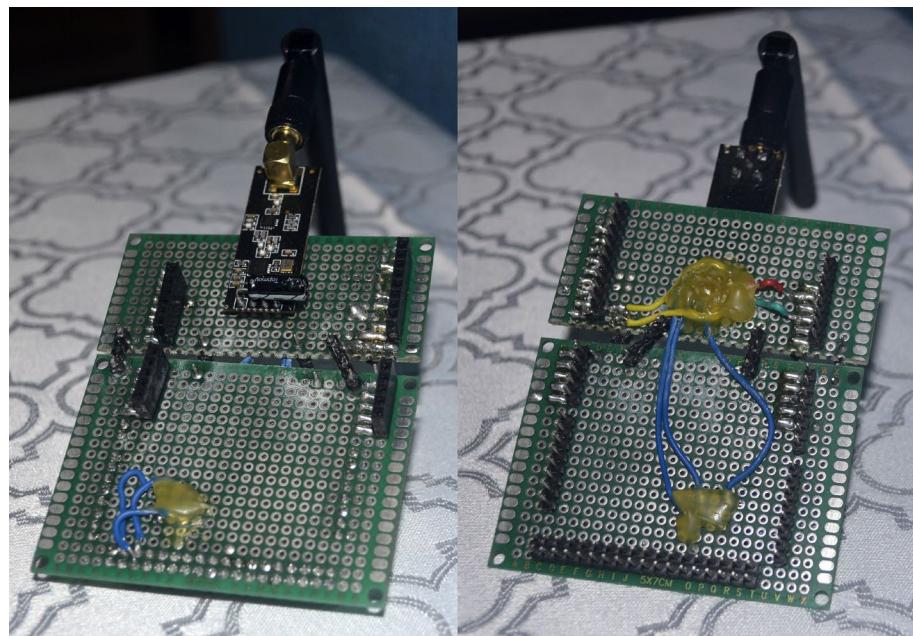
**Figura 49 – Adaptação do botão de *reset***



Fonte: elaborado pelos autores (2018).

Além da PCB *Shield* de aquisição de dados, referente ao Arduino transmissor, também foi montada uma *Shield* para adaptar o Arduino receptor, proporcionando uma conectividade das portas dos dispositivos com o Arduino mais fácil e intuitiva, com conceito semelhante ao *PnP* (*plug and play*).

**Figura 50 – PCB adaptada para Arduino receptor**

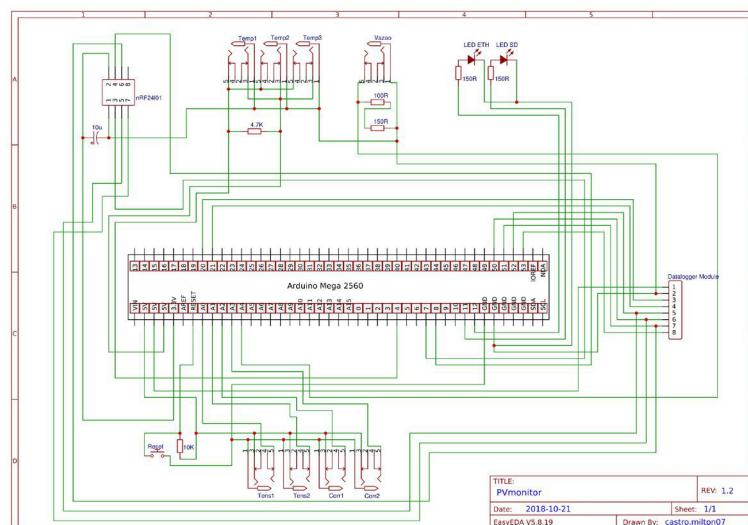


Fonte: elaborado pelos autores (2018).

#### **4.4.5 Placa de circuito impresso**

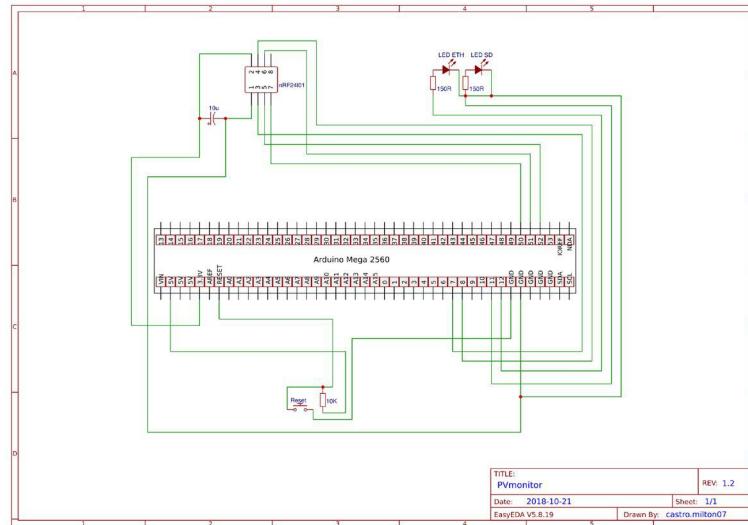
Com intuído de promover um aperfeiçoamento dos protótipos confeccionados para os Arduinos de transmissão e recepção, foram projetadas placas de circuito impresso exclusivas para este projeto, até o momento desta documentação as placas ainda não ficaram prontas, mas assim que estiverem disponíveis, serão devidamente testadas, instaladas e dependendo do resultado, substituirão as atuais.

**Figura 51– Esquema elétrico da placa do Arduino Transmissor**



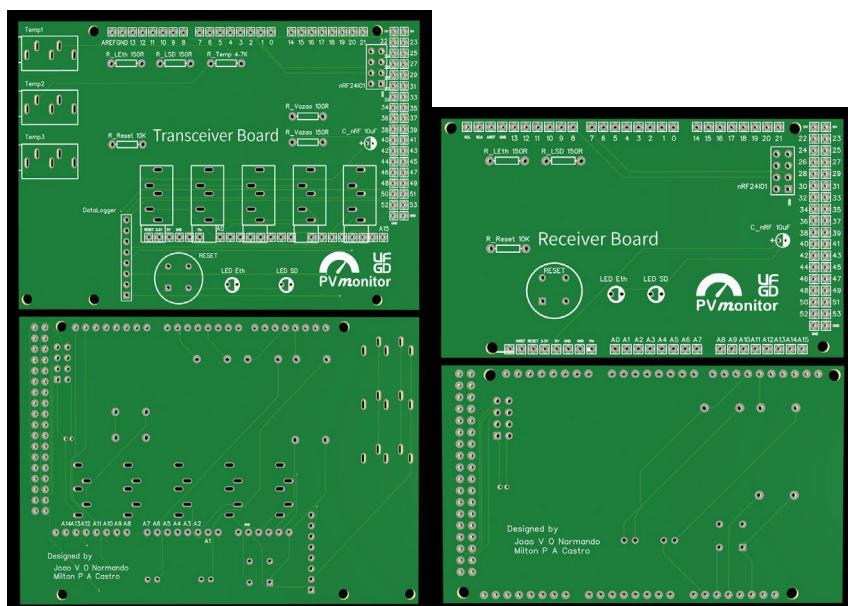
Fonte: elaborado pelos autores (2018).

**Figura 52 – Esquema elétrico da placa do Arduino Receptor**



Fonte: elaborado pelos autores (2018).

**Figura 53 – PCB projetada para substituir protótipo**



Fonte: elaborado pelos autores (2018).

#### 4.4.6 Arduino em campo (transmissor)

O Arduino que fica em campo foi fixado em baixo dos módulos FV junto aos outros componentes do sistema. Para evitar que o mesmo ficasse exposto a condições climáticas, foi necessário a adesão de uma caixa hermética com proteção IP67 que foi presa com dois parafusos tipo Phillips com porca e arruela na estrutura metálica que sustenta os três módulos fotovoltaicos.

Para a passagem dos cabos de alimentação do Arduino e dos sensores foram instaladas três prensas cabos na lateral da caixa, sendo dois para passagem dos fios dos sensores e um exclusivamente para alimentação. Para fixação do Arduino na caixa hermética utilizamos parafusos com porcas.

A disposição do Arduino na caixa foi pensada no intuito de facilitar todas as conexões, por isso o mesmo não ficou paralelo as paredes da caixa.

**Figura 54 – Arduino Mega 2560 fixado dentro da caixa hermética**



Fonte: elaborado pelos autores (2018).

## 1. CONCLUSÃO

O trabalho desenvolvido buscou realizar leituras de grandezas como tensão, corrente, temperatura e vazão, todos referentes a um sistema de bombeamento de água utilizando energia FV.

Uma das finalidades deste trabalho, foi possibilitar estudos mais aprofundados na área, visto que, anteriormente para realizar o levantamento de tais dados, era necessário realizar um deslocamento até o ambiente físico onde estavam instalados os módulos FV e realizar a coleta de dados manualmente.

Conforme execução deste trabalho, todos os dados citados acima, podem ser acessados remotamente, facilitando assim, o monitoramento e a produção de estudos relacionados ao bombeamento de água utilizando energia FV.

No desenvolvimento deste projeto alguns desafios estiveram presentes, um deles, foi a dificuldade para identificar uma falha de projeto no comutador de nível (tensão) do módulo SD tradicional, onde tal dificuldade demandou muito tempo, a qual não se tratava de uma falha de montagem ou programação, mas sim uma falha de projeto do dispositivo. Um outro fator a ser superado foi a comunicação com o Arduino que estava no bloco da FAEN cujo desafio envolvia conhecimentos de programação e após identificou-se que a solução era trivial, entretanto, após a criação do modelo de comunicação, a ideia de enviar dados do campo para o bloco, o qual possui conexão com a internet, mostrou-se muito eficiente.

Desse modo e com o uso do ThingSpeak, o qual se mostrou eficaz e preciso, também foi possível obter a mesma exatidão de apresentação dos dados coletados no projeto do app Virtuino.

Contudo, os resultados obtidos com a instalação do sistema DAQ proposto foram positivos, as conexões utilizadas encurtaram significativamente o tempo de montagem, uma vez que não houve necessidade de utilizar emendas de fios ou solda, logo, todos esses ajustes de cabos extensores e conectores MC4 puderam ser feitos em laboratório ou em casa, de modo natural, os quais serão necessários passarem por pequenos ajustes que poderão ser realizados conforme a necessidade da correção ou na medida em que as melhorias forem surgindo.

Como trabalhos futuros recomenda-se a inserção de um sistema de redundância resiliente a falhas dos dados armazenados no cartão de memória, o qual se encontra no sistema interno, localizado em uma sala dentro do prédio do campus.

Essa integração será de grande aplicabilidade, uma vez que, se a energia elétrica acabe ou por algum outro motivo o sistema interno pare de funcionar, os dados continuarão sendo coletados e armazenados no sistema externo, localizado no campus. Quando o sistema interno estiver ligado novamente, os dados salvos no cartão de memória não estarão atualizados com as leituras realizadas no período que o mesmo esteve indisponível, neste caso, a adição de uma bateria recarregável para suprir a falta de energia quando necessário, poderia ser uma solução para uma possível falta de energia elétrica no bloco.

Recomenda-se também, que os dados que não foram enviados a nuvem neste período de inatividade do sistema, termo popular do inglês *downtime*, também fossem enviados, assim, garantindo que todos os dados coletados em campo, sem exceção, estivessem disponíveis para consulta e estudos futuros.

Mediante o curto prazo de desenvolvimento do presente projeto, também não foi possível a elaboração de um sistema capaz de gerar gráficos com médias diárias, semanais e mensais da potência gerada pelos dois sistemas, o qual fosse viável relacionar com outras grandezas, tais como, a vazão da bomba d'água ou a temperatura dos painéis FV.

## REFERÊNCIAS

- AN AUGER S.A. Sistema anauger solar.** Itupeva: xkdesigner, 2011. Disponível em: <[https://www.neosolar.com.br/loja/fileuploader/download/download/?d=0&file=custo\\_m%2Fupload%2FFile-1455121282.pdf](https://www.neosolar.com.br/loja/fileuploader/download/download/?d=0&file=custo_m%2Fupload%2FFile-1455121282.pdf)>. Acesso em 19 nov. 2018.
- ARDUINO. Arduino Mega 2560 Rev3.** Disponível em: <<https://store.arduino.cc/usa/arduino-mega-2560-rev3>>. Acesso em 20 nov. 2018.
- ARDUINO. Compare board specs.** Disponível em: <<https://www.arduino.cc/en/products.compare>>. Acesso em 20 nov. 2018.
- BENGHANEM, M.; MAAFI, A. Data acquisition system for photovoltaic systems performance monitoring.** In: Instrumentation and Measurement Technology Conference, 1997. IMTC/97. Proceedings. Sensing, Processing, Networking., IEEE., 1997. p. 1030-1033.
- CAGNINI, P. R.; GRANDO, F. L.; COSTA, J. P. Desenvolvimento de uma plataforma experimental para aquisição de dados de geração fotovoltaica.** Gramado: VII Congresso Brasileiro de Energia Solar, 2018.
- CARVALHO, L. H. L. et al. Desenvolvimento de um sistema de monitoração para avaliação de desempenho operacional de um sistema fotovoltaico conectado à rede de baixa tensão.** Belo Horizonte: VI Congresso Brasileiro de Energia Solar, 2016.
- CASTRO, R M. G. Introdução à energia fotovoltaica – Energias renováveis e produção descentralizada.** Lisboa: Universidade Técnica de Lisboa – Instituto Superior Técnico (DEEC), 2002.
- CAVALCANTE, R. L et al. Implementação de um sistema de monitoração em um sistema híbrido solar-eólico instalado no flutuante base do instituto de desenvolvimento sustentável mamirauá.** Recife: V Congresso Brasileiro de Energia Solar, 2014.
- COELHO, F. J. S. C. et al. Desenvolvimento de um sistema de monitoramento fotovoltaico baseado em power line communication.** Belo Horizonte: VI Congresso Brasileiro de Energia Solar, 2016.
- COORD. TIBA. Atlas Solarimétrico do Brasil:** banco de dados terrestres. Recife: UFPE, 2001.
- DIAS, R. M.; DIAS, J. B.; PRÁ, L. B. D. Aquisição de dados de um sistema fotovoltaico conectado à rede com a plataforma arduino.** Belo Horizonte: VI Congresso Brasileiro de Energia Solar, 2016.
- DUPONT, I. M. et al. Sistema embarcado linux aplicado ao monitoramento em nuvem de planta de microgeração fotovoltaica.** Belo Horizonte: VI Congresso

Brasileiro de Energia Solar, 2016.

DUPONT, I. M. **Sistema embarcado Linux para monitoramento em nuvem aplicado a bombeamento fotovoltaico.** 2017. Dissertação (Mestrado em Engenharia Elétrica) - Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, 2017.

EPE, Empresa de Pesquisa Energética. **Plano Nacional de Energia 2030:** eficiência energética. Rio de Janeiro: EPE/MME, Brasil, 2007. Disponível em: <<http://epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-165/topico-173/PNE%202030%20-%20Efici%C3%A3ncia%20Energ%C3%A9tica.pdf>>. Acesso em 19 nov. 2018.

FORERO, N.; HERNÁNDEZ, J.; GORDILLO, G. **Development of a monitoring system for a PV solar plant.** Energy Conversion and Management, v. 47, n. 15-16, p. 2329-2336, 2006.

FREITAS, E. D. G.; MORAES, M. M. B.; SILVA, J. H. R. **Sistema de hardware e software para a aquisição de dados automática para painéis fotovoltaicos.** São Paulo: IV Congresso Brasileiro de Energia Solar, 2012.

GRANDO, F. L. et al. **Desenvolvimento de um sistema de aquisição de dados para aquecedores solares planos.** Belo Horizonte: VI Congresso Brasileiro de Energia Solar, 2016.

INCONTROL. **Manual de Operação e Instalação:** Conversor, Indicador e Transmissor de Vazão Eletromagnético PRO1000. Disponível em: <<D:\Usuários\MiltonCastro\Downloads\Documents\pro10-fev15-6812354568.pdf>>. Acesso em 17 nov. 2018.

INCONTROL. **Medidor de Vazão:** Eletromagnético. Disponível em: <<http://192.163.198.161/~incontrol/admin/public/img/magnetico-6133439447.pdf>>. Acesso em 17 nov. 2018.

JESUS, L. E. M. de et al. **Desenvolvimento de um sistema de supervisão e aquisição de dados para bancada de ensaios de motobombas utilizando Labview.** São Paulo: IV Congresso Brasileiro de Energia Solar, 2012.

KNIER, G. **How do photovoltaics work.** Science@ NASA, 2002.

KOUTROULIS, E.; KALAITZAKIS, K. **Development of an integrated data-acquisition system for renewable energy sources systems monitoring.** Renewable Energy, v. 28, n. 1, p. 139-152, 2003.

MANITO, A. R. A. et al. **Desenvolvimento de um sistema de aquisição de dados para monitoração de um sistema fotovoltaico conectado à rede.** Belém: III Congresso Brasileiro de Energia Solar, 2010.

**MANTECH. 433Mhz RF TX&RX.** Disponível em: <[http://www.mantech.co.za/datasheets/products/433mhz\\_rf-tx&rx.pdf](http://www.mantech.co.za/datasheets/products/433mhz_rf-tx&rx.pdf)>. Acesso em 21 nov. 2018.

MCEVOY, A. et al. (Ed.). **Practical handbook of photovoltaics: fundamentals and applications.** Elsevier, 2003.

**MOUSER. Arduino Ethernet Shield Datasheet.** Disponível em: <[https://www.mouser.com/catalog/specsheets/A000056\\_DATASHEET.pdf](https://www.mouser.com/catalog/specsheets/A000056_DATASHEET.pdf)>. Acesso em 20 nov. 2018.

**NORDIC, S. nRF24L01+:** Product Specification v1.0. Disponível em: <[https://www.nordicsemi.com/eng/nordic/download\\_resource/8765/2/24332246/2726](https://www.nordicsemi.com/eng/nordic/download_resource/8765/2/24332246/2726)>. Acesso em 21 nov. 2018.

**OLIVEIRA FILHO, J. I.; MAIA, W. C.** *Sistema de aquisição para painéis fotovoltaicos com armazenamento de dados em servidor remoto utilizando plataformas open source Raspberry pi e arduino.* Belo Horizonte: VI Congresso Brasileiro de Energia Solar, 2016.

**OMEGA, E. Medidores de Vazão Magnéticos:** Introdução aos Medidores de Vazão Magnéticos. Campinas: Omega Engineering, 2015. Disponível em: <<https://br.omega.com/prodinfo/medidores-de-vazao-magneticos.html>>. Acesso em 22 nov. 2018.

**PARK, J.; MACKAY, S.** *Practical data acquisition for instrumentation and control systems.* Newnes, 2003

**PEREIRA, E. B. et al.** *Atlas brasileiro de energia solar.* São José dos Campos: Inpe, 2006.

**PINHEIRO NETO, J. S. et al.** *Sistema de automação, controle e aquisição de dados para fontes renováveis descentralizadas.* São Paulo: IV Congresso Brasileiro de Energia Solar, 2012.

**PIOMBINI, E. S.; MELO, W. W. M.; CAMPOS, G. A. do P.** *Aplicativo android para monitoramento remoto de módulos fotovoltaicos.* Gramado: VII Congresso Brasileiro de Energia Solar, 2018.

**TEIXEIRA JUNIOR, L. R.; BLASQUES, L.; NASCIMENTO, A. C.** *Desenvolvimento de um sistema de monitoração para análise comparativa de dois sistemas fotovoltaicos conectados à rede com diferentes configurações.* Gramado: VII Congresso Brasileiro de Energia Solar, 2018.

**TORELI, G. de A. et al.** *Supervisório de código aberto e baixo custo para monitoramento remoto de geração fotovoltaica conectada à rede.* São Paulo: IV Congresso Brasileiro de Energia Solar, 2012.

**VILLALVA, M. G.; GAZOLI, J. R.** *Energia solar fotovoltaica: conceitos e aplicações.* São Paulo: Érica, 2012.

## APÊNDICE A – Código Fonte do Arduino Transmissor (TX)

```

1 #include <DallasTemperature.h>
2 #include <DS3231.h>
3 #include <nRF24L01.h>
4 #include <RF24.h>
5 #include <SD.h>
6 #include <SPI.h>
7 #include <OneWire.h>
8
9 #define pinNRFce 7
10 #define pinNRFcs 8
11 #define pinSDcs 53
12 #define pinTemp 4
13 #define pinTemp1 10
14 #define pinTemp2 11
15 #define pinTemp3 12
16 #define pinVazao 3
17
18 const int pinTens1 = 1;
19 const int pinTens2 = 5;
20 const int pinCorr1 = 7;
21 const int pinCorr2 = 8;
22 const byte address[6] = "000001";
23 int contadorMed = 0;
24 double temp1Med = 0;
25 double temp2Med = 0;
26 double temp3Med = 0;
27 double tens1Med = 0;
28 double tens2Med = 0;
29 double corr1Med = 0;
30 double corr2Med = 0;
31 double vazaoMed = 0;
32 double pot1Med = 0;
33 double pot2Med = 0;
34
35 DS3231 rtc(SDA, SCL);
36 RF24 radio(pinNRFce,pinNRFcs);
37 OneWire oneWire(pinTemp);
38 DallasTemperature sensores(&oneWire);
39 File file;
40
41 DeviceAddress Sensor02 = { 0x28, 0x83, 0x17, 0x43, 0x98, 0x04, 0x00, 0xDE };
42 DeviceAddress Sensor03 = { 0x28, 0x62, 0x15, 0x43, 0x98, 0x07, 0x00, 0x16 };
43 DeviceAddress Sensor04 = { 0x28, 0xFF, 0x86, 0xD5, 0x80, 0x14, 0x02, 0x96 };
44
45 void setup() {
46   Serial.begin(9600);
47
48   rtc.begin();
49
50   sensores.begin();
51   sensores.setResolution(Sensor02, 10);
52   sensores.setResolution(Sensor03, 10);
53   sensores.setResolution(Sensor04, 10);
54
55   SPI.begin();
56   pinMode(pinNRFcs, OUTPUT);
57   pinMode(pinSDcs, OUTPUT);
58   digitalWrite(pinNRFcs, HIGH);
59   digitalWrite(pinSDcs, HIGH);
60

```

```

61   digitalWrite(pinNRFcs, LOW);
62   radio.begin();
63   radio.openWritingPipe(address);
64   radio.setPALevel(RF24_PA_MAX);
65   radio.setDataRate(RF24_250KBPS);
66   radio.stopListening();
67   digitalWrite(pinNRFcs, HIGH);
68
69   digitalWrite(pinSDcs, LOW);
70   Serial.print("Iniciizando SD Card...");
71   if (!SD.begin(pinSDcs)) {
72     Serial.println("ERRO!! SD card falhou na inicialização, cartão com
defeito ou não está conectado!");
73     while (1);
74   }
75   Serial.println("SD card está pronto para uso.");
76   digitalWrite(pinSDcs, HIGH);
77
78   contadorMed = 0;
79   temp1Med = 0;
80   temp2Med = 0;
81   temp3Med = 0;
82   tens1Med = 0;
83   tens2Med = 0;
84   corr1Med = 0;
85   corr2Med = 0;
86   vazaoMed = 0;
87   pot1Med = 0;
88   pot2Med = 0;
89
90   String cabecalho =
" DiaSemana, Data, Hora, Temp1, Temp2, Temp3, Tens1, Tens2, Corr1, Corr2, Pot1, Pot2, Vazao";
91   Serial.println(cabecalho);
92   Serial.println();
93 }
94
95 void loop() {
96   String temp1, temp2, temp3, tens1, tens2, corr1, corr2, pot1, pot2,
vazao;
97
98   String dados1 = getData();
99   String dados2 = getSensores1();
100  String dados3 = getSensores2();
101
102  String Log = dados1 + "," + dados2 + "," + dados3;
103
104  dados1 += "-1-";
105  dados2 += "-2-";
106  dados3 += "-3-";
107
108  sendLog(dados1);
109  sendLog(dados2);
110  sendLog(dados3);
111
112  Log.remove(0, 24);
113
114  Log += ",";
115
116  temp1 = separaDados(&Log);
117  temp2 = separaDados(&Log);
118  temp3 = separaDados(&Log);
119  tens1 = separaDados(&Log);
120  tens2 = separaDados(&Log);
121  corr1 = separaDados(&Log);
122  corr2 = separaDados(&Log);
123  vazao = separaDados(&Log);
124

```

```

125     dados1.remove(23);
126
127     pot1 = String((double)tens1.toFloat()*(double)corr1.toFloat(), 2);
128     pot2 = String((double)tens2.toFloat()*(double)corr2.toFloat(), 2);
129
130     temp1Med = calculaMedia(&contadorMed, &temp1Med, temp1);
131     temp2Med = calculaMedia(&contadorMed, &temp2Med, temp2);
132     temp3Med = calculaMedia(&contadorMed, &temp3Med, temp3);
133     tens1Med = calculaMedia(&contadorMed, &tens1Med, tens1);
134     tens2Med = calculaMedia(&contadorMed, &tens2Med, tens2);
135     corr1Med = calculaMedia(&contadorMed, &corr1Med, corr1);
136     corr2Med = calculaMedia(&contadorMed, &corr2Med, corr2);
137     vazaoMed = calculaMedia(&contadorMed, &vazaoMed, vazao);
138     pot1Med = calculaMedia(&contadorMed, &pot1Med, pot1);
139     pot2Med = calculaMedia(&contadorMed, &pot2Med, pot2);
140
141     Log = dados1 + "," + temp1 + "," + temp2 + "," + temp3 + "," + tens1 +
142     "," + tens2 + "," + corr1 + "," + corr2 + "," + pot1 + "," + pot2 + "," + vazao;
143
144     if(contadorMed == 11) {
145         String LogMed = dados1 + "," + temp1Med + "," + temp2Med + "," +
146         temp3Med + "," + tens1Med + "," + tens2Med + "," + corr1Med + "," +
147         corr2Med + "," + pot1Med + "," + pot2Med + "," + vazaoMed;
148         writeLog(LogMed);
149         contadorMed = -1;
150         temp1Med = 0;
151         temp2Med = 0;
152         temp3Med = 0;
153         tens1Med = 0;
154         tens2Med = 0;
155         corr1Med = 0;
156         corr2Med = 0;
157         vazaoMed = 0;
158         pot1Med = 0;
159         pot2Med = 0;
160     }
161 }
162
163 String getData() {
164     uint8_t format = FORMAT_SHORT;
165     String dataTemp = rtc.getDateStr();
166     String data = dataTemp.substring(0,2) + "/" + dataTemp.substring(3,5) +
167     "/" + dataTemp.substring(6,10);
168     return (String)rtc.getDOWStr(format) + "," + data + "," +
169     (String)rtc.getTimeStr();
170 }
171
172 String getSensores1() {
173     sensores.requestTemperatures();
174     String temp1 = String((double)sensores.getTempC(Sensor02), 2);
175     if(temp1.length() < 5) temp1 = corrigeTamString(temp1);
176     if(temp1.length() > 5) temp1 = "99.99";
177     if(temp1.startsWith("-")) temp1 = "00.00";
178     String temp2 = String((double)sensores.getTempC(Sensor03), 2);
179     if(temp2.length() < 5) temp2 = corrigeTamString(temp2);
180     if(temp2.length() > 5) temp2 = "99.99";
181     if(temp2.startsWith("-")) temp2 = "00.00";
182     String temp3 = String((double)sensores.getTempC(Sensor04), 2);
183     if(temp3.length() < 5) temp3 = corrigeTamString(temp3);
184     if(temp3.length() > 5) temp3 = "99.99";
185     if(temp3.startsWith("-")) temp3 = "00.00";
186     String tens1 = String((double)getTensao(pinTens1), 2);
187     if(tens1.length() < 5) tens1 = corrigeTamString(tens1);
188     if(tens1.length() > 5) tens1 = "99.99";

```

```

187     if(tens1.startsWith("-")) tens1 = "00.00";
188     return temp1 + "," + temp2 + "," + temp3 + "," + tens1;
189 }
190
191 String getSensores2() {
192     String tens2 = String((double)getTensao(pinTens2), 2);
193     if(tens2.length() < 5) tens2 = corrigeTamString(tens2);
194     if(tens2.length() > 5) tens2 = "99.99";
195     if(tens2.startsWith("-")) tens2 = "00.00";
196     String corr1 = String((double)getCorrente(pinCorr1), 2);
197     if(corr1.length() < 5) corr1 = corrigeTamString(corr1);
198     if(corr1.length() > 5) corr1 = "99.99";
199     if(corr1.startsWith("-")) corr1 = "00.00";
200     String corr2 = String((double)getCorrente(pinCorr2), 2);
201     if(corr2.length() < 5) corr2 = corrigeTamString(corr2);
202     if(corr2.length() > 5) corr2 = "99.99";
203     if(corr2.startsWith("-")) corr2 = "00.00";
204     String vazao = String(double(mapFloat(analogRead(pinVazao), 204.6,
1023.0, 0.0, 10.0)), 2);
205     if(vazao.length() < 5) vazao = corrigeTamString(vazao);
206     if(vazao.length() > 5) vazao = "99.99";
207     if(vazao.startsWith("-")) vazao = "00.00";
208     return tens2 + "," + corr1 + "," + corr2 + "," + vazao;
209 }
210
211 double getTensao(int pinTensao) {
212     double tensao = 0;
213     int pin = 0;
214     int leituraTensao = analogRead(pinTensao);
215     if(pinTensao == 1) {
216         tensao = map(leituraTensao, 0, 1023, 0, 6075);
217         pin = 1;
218     }
219     else if(pinTensao == 5) {
220         tensao = map(leituraTensao, 0, 1023, 0, 3640);
221         pin = 2;
222     }
223     tensao /= 100;
224     return tensao;
225 }
226
227 double getCorrente(int pinCorrente) {
228     double corrente = 0;
229     double leituraCorrente = 0;
230     double voltsPorUnidade = 0.0048828125;
231     double ruido = 0;
232     int pin = 0;
233     if(pinCorrente == 7) {
234         pin = 1;
235         ruido = -0.0362;
236     }
237     else if(pinCorrente == 8) {
238         pin = 2;
239         ruido = -0.0090;
240     }
241     for(int i = 0; i<450; i++) {
242         leituraCorrente = analogRead(pinCorrente);
243         leituraCorrente = (leituraCorrente-510)*voltsPorUnidade;
244         corrente = corrente+(leituraCorrente/100)*1000;
245         delay(1);
246     }
247     corrente /= 450;
248     corrente -= ruido;
249     return corrente;
250 }
251
252 String corrigeTamString(String sensor) {

```

```

253     String aux = sensor;
254     sensor = "0" + aux;
255     return sensor;
256 }
257
258 void sendLog(String tempSendLog) {
259     const char text[32];
260     tempSendLog.toCharArray(text, 32);
261     digitalWrite(pinNRFcs, LOW);
262     radio.write(&text, sizeof(text));
263     digitalWrite(pinNRFcs, HIGH);
264 }
265
266 void writeLog(String tempLog) {
267     digitalWrite(pinSDcs, LOW);
268     file = SD.open("log.csv", FILE_WRITE);
269     if(file) {
270         file.println(tempLog);
271         file.close();
272     }
273     else {
274         Serial.println("ERRO!! O arquivo não pode ser aberto.");
275     }
276     digitalWrite(pinSDcs, HIGH);
277 }
278
279 double calculaMedia(int *contador, double *mediaTemp, String novaLeitura) {
280     if(*contador == 11) {
281         double media = (*mediaTemp+novaLeitura.toFloat()) / 12;
282         return media;
283     }
284     else {
285         double somatorio = *mediaTemp + (double) novaLeitura.toFloat();
286         return somatorio;
287     }
288 }
289
290 String separaDados(String *Log) {
291     String c, valor;
292     double aux;
293     for (int i = 0; i < Log->length(); i++) {
294         c = Log->charAt(i);
295         if (c == ",") {
296             valor = Log->substring(0, i);
297             Log->remove(i-i, i+1);
298             break;
299         }
300     }
301     aux = valor.toFloat();
302     valor = String(aux, 2);
303     return valor;
304 }
305
306 float mapFloat(float x, float in_min, float in_max, float out_min, float
out_max) {
307     return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
308 }

```

## APÊNDICE B – Código Fonte do Arduino Receptor (RX)

```

1 #include <Ethernet.h>
2 #include <nRF24L01.h>
3 #include <RF24.h>
4 #include <SD.h>
5 #include <SPI.h>
6 #include "ThingSpeak.h"
7
8 #define pinNRFce 7
9 #define pinNRFcs 8
10#define pinSDcs 4
11#define pinETHcs 10
12
13 EthernetClient cliente;
14 RF24 radio(pinNRFce,pinNRFcs);
15 File file;
16
17 const byte address[6] = "00001";
18 const byte mac[] = {0x86,0x16,0xF9,0xB9,0x2E,0xD3};
19 const unsigned long canal1 = 629468;
20 const char * APIKey1 = "XXXXXXXXXXXXXXXXXX";
21 const unsigned long canal2 = 629496;
22 const char * APIKey2 = "XXXXXXXXXXXXXXXXXX";
23 int contador = 0;
24 double temp1Med = 0;
25 double temp2Med = 0;
26 double temp3Med = 0;
27 double tens1Med = 0;
28 double tens2Med = 0;
29 double corr1Med = 0;
30 double corr2Med = 0;
31 double vazaoMed = 0;
32 double pot1Med = 0;
33 double pot2Med = 0;
34
35 void setup() {
36     Serial.begin(9600);
37
38     SPI.begin();
39     pinMode(pinNRFcs, OUTPUT);
40     pinMode(pinSDcs, OUTPUT);
41     pinMode(pinETHcs, OUTPUT);
42     digitalWrite(pinNRFcs, HIGH);
43     digitalWrite(pinSDcs, HIGH);
44     digitalWrite(pinETHcs, HIGH);
45
46     digitalWrite(pinNRFcs, LOW);
47     radio.begin();
48     radio.openReadingPipe(0,address);
49     radio.setPALevel(RF24_PA_MAX);
50     radio.setDataRate(RF24_250KBPS);
51     radio.startListening();
52     digitalWrite(pinNRFcs, HIGH);
53
54     digitalWrite(pinSDcs, LOW);
55     Serial.print("Inicializando SD Card...");
56     if (!SD.begin(pinSDcs)) {
57         Serial.println("ERRO!! SD card falhou na inicialização, cartão com
58 defeito ou não conectado!");
59     }
60     Serial.println("SD card está pronto para uso.");
61     digitalWrite(pinSDcs, HIGH);
62

```

```

63
64
65 digitalWrite(pinETHcs, LOW);
66 Serial.print("Inicializando ETH..."); 
67 if (!Ethernet.begin(mac)) {
68     Serial.println("ERRO!! ETHERNET falhou na inicialização, cabo com
defeito ou não conectado!");
69     while (1);
70 }
71 Serial.println("ETHERNET está pronta para uso.");
72 ThingSpeak.begin(cliente);
73 digitalWrite(pinETHcs, HIGH);
74
75 digitalWrite(pinETHcs, LOW);
76 Ethernet.begin(mac);
77 ThingSpeak.begin(cliente);
78 digitalWrite(pinETHcs, HIGH);
79
80 contador = 0;
81 temp1Med = 0;
82 temp2Med = 0;
83 temp3Med = 0;
84 tens1Med = 0;
85 tens2Med = 0;
86 corr1Med = 0;
87 corr2Med = 0;
88 vazaoMed = 0;
89 pot1Med = 0;
90 pot2Med = 0;;
91
92 String cabecalho =
" DiaSemana,Data,Hora,Temp1,Temp2,Temp3,Tens1,Tens2,Corr1,Corr2,Vazao";
93 Serial.println(cabecalho);
94 Serial.println();
95 }
96
97 void loop() {
98     String dataHora, temp1, temp2, temp3, tens1, tens2, corr1, corr2, pot1,
pot2, vazao;
99     int verificador = 0;
100    double tempoRestante;
101
102    String Log = receiveLog();
103
104    dataHora = Log;
105    dataHora.remove(23);
106
107    Log.remove(0, 24);
108
109    Log += ",";
110
111    temp1 = separaDados(&Log);
112    temp2 = separaDados(&Log);
113    temp3 = separaDados(&Log);
114    tens1 = separaDados(&Log);
115    tens2 = separaDados(&Log);
116    corr1 = separaDados(&Log);
117    corr2 = separaDados(&Log);
118    vazao = separaDados(&Log);
119
120    pot1 = String((double)tens1.toFloat() * (double)corr1.toFloat(), 2);
121    pot2 = String((double)tens2.toFloat() * (double)corr2.toFloat(), 2);
122
123    temp1Med = calculaMedia(&contador, &temp1Med, temp1);
124    temp2Med = calculaMedia(&contador, &temp2Med, temp2);
125    temp3Med = calculaMedia(&contador, &temp3Med, temp3);
126    tens1Med = calculaMedia(&contador, &tens1Med, tens1);

```

```

127     tens2Med = calculaMedia(&contador, &tens2Med, tens2);
128     corr1Med = calculaMedia(&contador, &corr1Med, corr1);
129     corr2Med = calculaMedia(&contador, &corr2Med, corr2);
130     vazaoMed = calculaMedia(&contador, &vazaoMed, vazao);
131     pot1Med = calculaMedia(&contador, &pot1Med, pot1);
132     pot2Med = calculaMedia(&contador, &pot2Med, pot2);
133
134     Log = dataHora + "," + temp1 + "," + temp2 + "," + temp3 + "," + tens1
+ "," + tens2 + "," + corr1 + "," + corr2 + "," + pot1 + "," + pot2 + "," +
vazao;
135
136     if(contador == 19) {
137         String LogMed = dataHora + "," + temp1Med + "," + temp2Med + "," +
temp3Med + "," + tens1Med + "," + tens2Med + "," + corr1Med + "," + corr2Med +
"," + pot1Med + "," + pot2Med + "," + vazaoMed;
138         writeLog(LogMed);
139     }
140
141     verificador = sendThingSpeak(&contador, temp1Med, temp2Med, temp3Med,
tens1Med, tens2Med, corr1Med, corr2Med, vazaoMed, pot1Med, pot2Med);
142
143     if(verificador == 1) {
144         Serial.println("Dados foram enviados ao servidor ThingSpeak com
SUCESSO!");
145     }
146     else if(verificador == -1) {
147         tempoRestante = 15-(double)contador*0.75;
148         Serial.print("Intervalo de 15 segundos ainda não foi
atingido.\nFaltam: ");
149         Serial.print(tempoRestante);
150         Serial.println(" segundos para fazer o proximo envio.");
151         Serial.println();
152     }
153     else Serial.println("Erro!!!");
154     delay(750);
155     contador++;
156 }
157
158 String receiveLog() {
159     char text[32] = "";
160     String tempLog = "";
161     String tempReceiveLog = "";
162
163     while (!tempLog.endsWith("---")) {
164         digitalWrite(pinNRFcs, LOW);
165         if (radio.available()) {
166             radio.read(&text, sizeof(text));
167             tempReceiveLog = String(text);
168             digitalWrite(pinNRFcs, HIGH);
169         }
170         //else Serial.println("Erro: Sem Comunicacao!!!");
171
172         if ((tempReceiveLog.endsWith("-1-")) && (tempLog.length() == 0)) {
173             tempReceiveLog.remove(23);
174             tempLog = tempReceiveLog;
175         }
176
177         if ((tempReceiveLog.endsWith("-2-")) && (tempLog.length() > 20)) {
178             tempReceiveLog.remove(23);
179             tempLog += "," + tempReceiveLog;
180         }
181
182         if ((tempReceiveLog.endsWith("-3-")) && (tempLog.length() > 40)) {
183             tempReceiveLog.remove(23);
184             tempLog += "," + tempReceiveLog;
185             tempLog += "---";
186         }

```

```

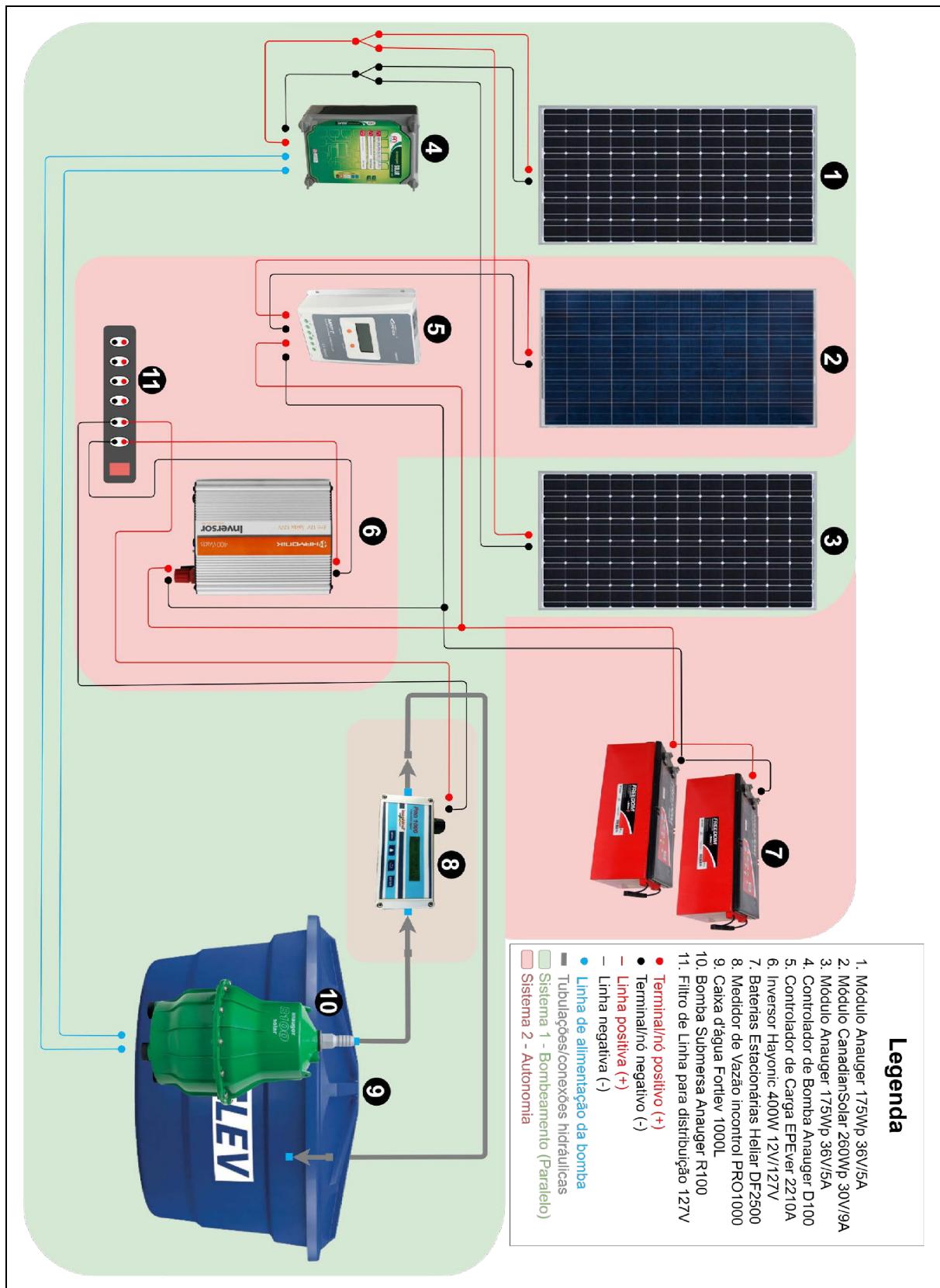
187     }
188     tempLog.remove(71);
189     if (tempLog.length() > 50) {
190         return tempLog;
191     }
192 }
193
194 void writeLog(String tempLog) {
195     digitalWrite(pinSDCs, LOW);
196     file = SD.open("log.csv", FILE_WRITE);
197     if (file) {
198         file.println(tempLog);
199         file.close();
200     }
201     else {
202         Serial.println("ERRO!! O arquivo não pode ser aberto");
203     }
204     digitalWrite(pinSDCs, HIGH);
205 }
206
207 double calculaMedia(int *contador, double *mediaTemp, String novaLeitura)
{
208     if(*contador == 19) {
209         double media = (*mediaTemp+novaLeitura.toFloat()) / 20;
210         return media;
211     }
212     else {
213         double somatorio = *mediaTemp+(double) novaLeitura.toFloat();
214         return somatorio;
215     }
216 }
217
218 int sendThingSpeak(int *contador, float temp1, float temp2, float temp3,
219 float tens1, float tens2, float corr1, float corr2, float vazao, float pot1,
220 float pot2) {
221     if(*contador == 19) {
222         ThingSpeak.setField(1, temp1);
223         ThingSpeak.setField(2, temp3);
224         ThingSpeak.setField(3, tens1);
225         ThingSpeak.setField(4, corr1);
226         ThingSpeak.setField(5, pot1);
227         ThingSpeak.setField(6, vazao);
228         ThingSpeak.writeFields(canall, APIKey1);
229         ThingSpeak.setField(1, temp2);
230         ThingSpeak.setField(2, tens2);
231         ThingSpeak.setField(3, corr2);
232         ThingSpeak.setField(4, pot2);
233         ThingSpeak.writeFields(canal2, APIKey2);
234         *contador = -1;
235         return 1;
236     }
237     else return -1;
238 }
239
240 String separaDados(String *Log) {
241     String c, valor;
242     double aux;
243     for (int i = 0; i < Log->length(); i++) {
244         c = Log->charAt(i);
245         if (c == ",") {
246             valor = Log->substring(0, i);
247             Log->remove(i-i, i+1);
248             break;
249         }
250         aux = valor.toFloat();
251         valor = String(aux, 2);

```

```
251     return valor;  
252 }
```

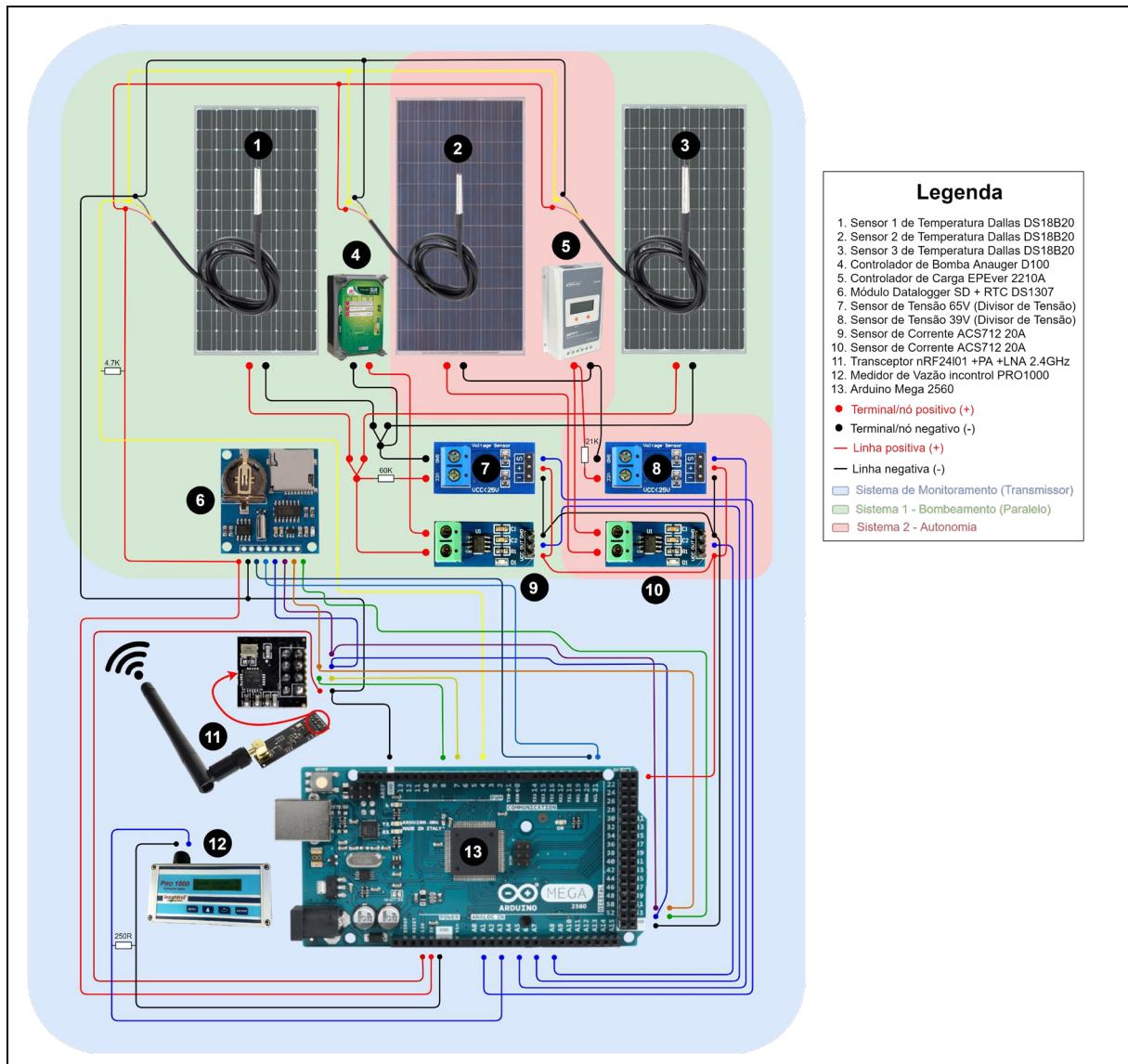
Documento alterado pela última vez no dia 21 nov. 2018.

## APÊNDICE C – Diagrama Sistema Fotovoltaico



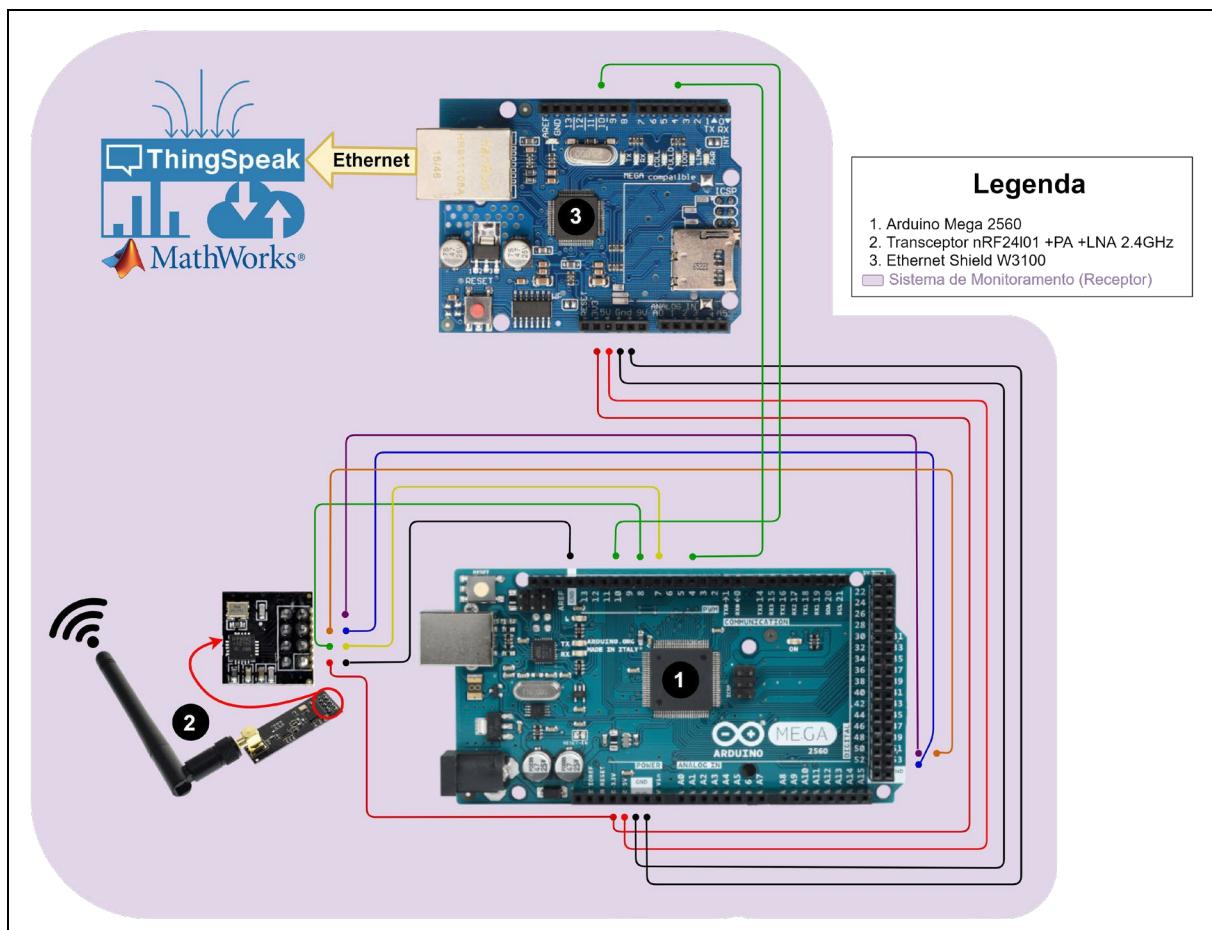
Fonte: elaborado pelos autores (2018).

## APÊNDICE D – Diagrama Sistema Monitoramento: Transmissor



Fonte: elaborado pelos autores (2018).

## APÊNDICE E – Diagrama Sistema Monitoramento: Receptor



Fonte: elaborado pelos autores (2018).