



UNIVERSIDADE FEDERAL DA GRANDE DOURADOS
FACULDADE DE ENGENHARIA
CURSO DE ENGENHARIA DE ENERGIA



Gabriel de Oliveira Alves

RASTREADOR SOLAR CONTROLADO POR ARDUINO

DOURADOS/MS

2018

Gabriel de Oliveira Alves

RASTREADOR SOLAR CONTROLADO POR ARDUINO

Trabalho de Conclusão de Curso
apresentado à banca examinadora da
Faculdade de Engenharia da Universidade
Federal da Grande Dourados para a
obtenção do título de Bacharel em
Engenharia de Energia.

Orientador: Prof. Dr. Gerson Bessa Gibelli

DOURADOS/MS

2018

Dados Internacionais de Catalogação na Publicação (CIP).

A474r Alves, Gabriel De Oliveira
Rastreador solar controlado por arduino / Gabriel De Oliveira Alves –
Dourados: UFGD, 2018.
99f. : il. ; 30 cm.

Orientador: Gerson Bessa Gibelli

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

1. Arduino. 2. Automação. 3. Controle. I. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

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

RASTREADOR SOLAR CONTROLADO POR ARDUINO

Por

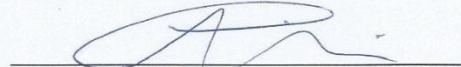
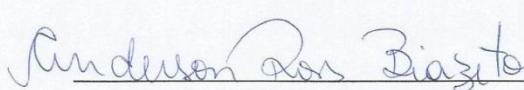
GABRIEL DE OLIVEIRA ALVES

Trabalho de Conclusão de Curso apresentado como parte dos requisitos
exigidos para obtenção do título de ENGENHEIRO DE ENERGIA

Aprovado em: 25/07/2018


Prof. Dr. Gerson Bessa Gibelli

Orientador – UFGD/FAEN


Prof. Dr. Clivaldo de Oliveira – UFGD/FAEN
Mestre Anderson Ross Biazeto – UFGD

DOURADOS
MATO GROSSO DO SUL
2018

Aos meus pais, José e Dorcilia, por todo amor e confiança que sempre dedicaram a mim.

AGRADECIMENTOS

Primeiramente, agradeço a Deus pelo dom da vida e por que sem Ele em minha vida, não poderia fazer os agradecimentos a seguir.

Aos meus pais José e Dorcília, meu irmão André Luis e às minhas tias Maria Aparecida e Maria Alzira pelo apoio incondicional ao longo da execução deste trabalho, colaborando na compra dos componentes e dando dicas para as montagens.

Ao professor Dr. Gerson Bessa Gibelli, meu orientador, por toda dedicação e pelas críticas construtivas que permitiram a este trabalho seguir o melhor caminho.

Ao professor Dr. Clivaldo de Oliveira por contribuir com materiais sobre o ScadaBR, que foram de grande ajuda no desenvolvimento do supervisório.

Aos meus amigos que me acompanham desde o início do curso, contribuindo para minha formação tanto como acadêmico, quanto como pessoa.

Às comunidades que contribuem para projetos de código aberto como Arduino e ScadaBR. Este trabalho certamente não teria ganhado vida se não existisse o esforço de pessoas que colocam a busca pelo conhecimento acima da necessidade de lucro. Muito obrigado a todos.

X

“Tecnologia não é nada. O que é importante é que se tenha fé nas pessoas, que elas são basicamente boas e inteligentes, e se derem a elas ferramentas, elas farão coisas maravilhosas com elas.”

(Steve Jobs)

RESUMO

Alves, G. O. **RASTREADOR SOLAR CONTROLADO POR ARDUINO.** 75 p.
Monografia (Trabalho de Conclusão de Curso) – Faculdade de Engenharia, Universidade Federal da Grande Dourados, 2018.

Com a expansão constante do uso de fontes renováveis para produção de energia como a solar fotovoltaica, torna-se importante a realização de estudos que possam contribuir para a maximização de sua eficiência. Neste contexto, o uso de um Rastreador Solar pode colaborar para que se colete a maior quantidade de luz solar possível. Desta forma, foi desenvolvido o protótipo de um Rastreador Solar controlado via plataforma Arduino, que por sua vez é de baixo custo e fácil uso, possuindo um supervisório baseado no *software* ScadaBR, cuja origem é brasileira, sendo também grátis e de código aberto. A combinação entre Arduino e ScadaBR se mostrou de grande valia, pois facilitou a programação do Rastreador Solar e possibilitou a execução de melhorias no protótipo durante as medições realizadas. São apresentadas dicas para trabalhos futuros, fazendo com que este trabalho possa servir de base para o desenvolvimento de outros tipos de protótipos.

Palavras-chave: Arduino, Automação, Controle.

ABSTRACT

Alves, G. O. **SOLAR TRACKER CONTROLLED VIA ARDUINO.** 75 p. Monograph (Course Completion Work) – Faculty of Engineering, Federal University of Grande Dourados, 2018.

With the constant expansion of the renewable energy sources such as the solar photovoltaic, it is important to carry out studies that may contribute to the maximization of its efficiency. In this context, the use of a Solar Tracker can contribute to collecting as much sunlight as possible. In this way, a prototype of a Solar Tracker controlled via Arduino platform was developed, which in turn is low cost and easy to use, having a supervisory system based on ScadaBR software, whose production is Brazilian, and is also free and open-source. The combination between Arduino and ScadaBR proved to be of great value, since it facilitated the programming of the Solar Tracker and made possible the execution of improvements in the prototype during the measurements performed. Tips for future works are presented, making this work as a basis for the development of other types of prototypes.

Keywords: Arduino, Automation, Control.

LISTA DE ILUSTRAÇÕES

Figura 1.1: Pulsos do Sol.....	1
Figura 1.2: Propaganda da Bell Solar Battery	2
Figura 1.3: Parque Solar Lapa.....	3
Figura 1.4: Seguidor Solar Passivo.....	4
Figura 1.5: Rastreador Solar Ativo.....	5
Figura 1.6: Representação Gráfica em ScadaBR.....	7
Figura 1.7: Comparação de Sistema de Placa Fixa e Rastreador Solar.....	8
Figura 2.1: Arduino UNO.....	11
Figura 2.2: Seleção da Placa na <i>IDE</i>	12
Figura 2.3: <i>Light Dependent Resistor</i>	13
Figura 2.4: Servomotor SG-5010.	14
Figura 2.5: Células Fotovoltaicas de Silício Monocristalino.	15
Figura 2.6: Coordenadas de Orientação da Superfície.	16
Figura 2.7: Rastreador Solar Controlado por PIC 16F877	18
Figura 2.8: Protótipo de Rastreador Solar de Baixo Custo.	18
Figura 2.9: Rastreador Solar Controlado por Arduino.	19
Figura 2.10: Rastreador Solar Controlado por Arduino UNO.....	20
Figura 2.11: <i>Interface</i> de Supervisão e Controle em MATLAB®.....	20
Figura 2.12: <i>Interface</i> de Aplicativo Usado pelo Paciente.	22
Figura 2.13: Representação Gráfica em ScadaBR Usada Pelo Cuidador.....	22
Figura 2.14: Supervisório de Temperaturas.....	23
Figura 2.15: Supervisório para Controle de Pressão em Adutora de Água.	23
Figura 2.16: Configuração de <i>Data Source Modbus IP</i>	25
Figura 2.17: Exemplo de Criação de <i>Data Point</i>	25
Figura 2.18: Exemplo de <i>Watchlist</i> em ScadaBR.	26
Figura 2.19: Configuração de Geração de Relatórios.....	27
Figura 2.20: Criação de <i>Script</i> em ScadaBR.	28
Figura 2.21: Lista de Alarmes em ScadaBR.....	28
Figura 3.1: Diagrama de Blocos do Rastreador Solar.	29
Figura 3.2: Representação da Montagem em Protoboard do Rastreador Solar.....	30
Figura 3.3: Fluxograma do Rastreador Solar.....	31

XVIII

Figura 3.4: Sensor de Tensão.....	32
Figura 3.5: Sensor de Corrente ACS712.	32
Figura 3.6: Vista Lateral da Estrutura de Madeira.....	34
Figura 3.7: Vista da Caixa de Madeira com Servomotor Encaixado.....	35
Figura 3.8: Medidas do Triângulo Retângulo.....	35
Figura 3.9: Vista do Interior do Triângulo Retângulo.	36
Figura 3.10: Montagem Final do Rastreador Solar.....	37
Figura 3.11: Montagem Real em Protoboard.	37
Figura 3.12: <i>Data Source Modbus Serial</i>	39
Figura 3.13: Lista de Data Points.	40
Figura 3.14: <i>Data Point</i> para Sensor de Tensão.	41
Figura 3.15: <i>Script</i> para Obtenção da Corrente Média.	42
Figura 3.16: <i>Script</i> para Obtenção da Potência.	43
Figura 3.17: <i>Script</i> para Obtenção da Resistência em Ohms.	43
Figura 3.18: <i>Watchlist</i> para Rastreador Solar.	44
Figura 3.19: Representação Gráfica para Rastreador Solar.....	45
Figura 3.20: Configuração do e-mail no ScadaBR.....	46
Figura 3.21: Configuração para Geração de Relatórios no ScadaBR.....	47
Figura 4.1: Resistências dos LDR no 1º Dia.	49
Figura 4.2: Tensões Obtidas no 1º Dia de Medições.	50
Figura 4.3: Corrente Média no 1º Dia de Medições.	50
Figura 4.4: Movimento do Servomotor no 1º Dia.	51
Figura 4.5: Resistências dos LDR no 2º Dia.	52
Figura 4.6: Tensões Obtidas no 2º Dia.	52
Figura 4.7: Corrente Média Obtida no 2º Dia.	53
Figura 4.8: Curva de Potência no 2º Dia.	53
Figura 4.9: Movimento do Servomotor no 2º Dia.	54
Figura 4.10: Resistências dos LDR no 3º Dia.	55
Figura 4.11: Tensão Real Obtida no 3º Dia.	55
Figura 4.12: Corrente Média Obtida no 3º Dia.	56
Figura 4.13: Curva de Potência no 3º Dia.	57
Figura 4.14: Movimento do Servomotor no 3º Dia.	57

XIX

Figura 4.15: Movimento do Servomotor no 4º Dia.....	58
Figura 4.16: Tensão Real Obtida no 4º Dia.....	59
Figura 4.17: Medição de Corrente Média no 4º Dia.....	60
Figura 4.18: Curva de Potência no 4º Dia.....	60
Figura 4.19: Resistências dos LDR no 5º Dia.....	61
Figura 4.20: Movimento do Servomotor no 5º Dia.....	61
Figura 4.21: Corrente Média Obtida no 5º Dia.....	62
Figura 4.22: Curva de Potência no 5º Dia.....	63

XX

LISTA DE SÍMBOLOS E SIGLAS

α	Altura Solar
β	Inclinação da Superfície
γ	Ângulo de Incidência
a_s	Ângulo Azimutal do Sol
a_w	Ângulo Azimutal da Superfície
ADC	<i>Analogic to Digital Converter</i>
IHM	<i>Interface Homem Máquina</i>
LDR	<i>Light Dependent Resistor</i>
PCI	Placa de Circuito Impresso
PWM	<i>Pulse Width Modulation</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>

SUMÁRIO

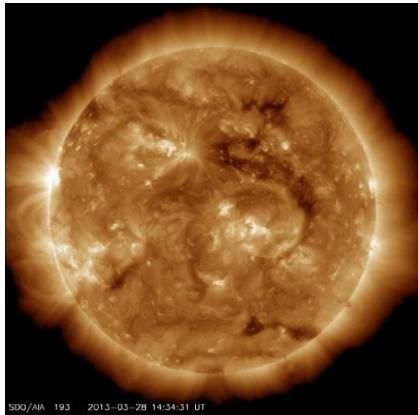
1. INTRODUÇÃO	1
1.1. OBJETIVOS	7
1.2. JUSTIFICATIVA.....	7
1.3. ESTRUTURA DO TRABALHO	8
2. REVISÃO BIBLIOGRÁFICA	11
2.1. ARDUINO	11
2.2. SENSORES LDR.....	12
2.3. SERVOMOTOR.....	13
2.4. PLACA FOTOVOLTAICA.....	14
2.5. SEGUIDOR SOLAR	15
2.5.1. Geometria Solar	15
2.5.2. Características de um Rastreador Solar.....	17
2.6. SUPERVISÓRIO	21
2.6.1. Sistemas SCADA.....	21
2.6.2. Protocolo de Comunicação.....	24
2.6.3. Funcionalidades do ScadaBR	24
3. METODOLOGIA.....	29
3.1. LÓGICA DE FUNCIONAMENTO.....	29
3.2. DESENVOLVIMENTO DO RASTREADOR SOLAR	33
3.3. DESENVOLVIMENTO DO SUPERVISÓRIO	38
3.3.1. Data Source	38
3.3.2. Data Points	39
3.3.3. Scripts	41
3.3.4. Visualização dos Dados	44
3.3.5. Geração de Relatórios	45
4. RESULTADOS E DISCUSSÕES	49
5. CONCLUSÃO.....	65
5.1. TRABALHOS FUTUROS.....	66
REFERÊNCIAS	69
APÊNDICE	73

1. INTRODUÇÃO

O Sol é a base para tudo que existe no Sistema Solar, desde rochas e bactérias menores que a cabeça de uma agulha até os animais mais complexos que habitam o planeta Terra, todos existem por conta da existência do astro-rei, cuja influência se estende por todo o Sistema Solar.

De acordo com a *National Aeronautics and Space Administration* (NASA), o Sol é uma “bola” quente de gases brilhantes, localizada no coração do nosso Sistema Solar e sem o seu intenso calor (a temperatura do seu interior ultrapassa 5500 °C), não seria possível haver vida na Terra. Processos como a fotossíntese, a evaporação e o ciclo das águas, sendo que este último permite o seu represamento e a consequente produção de eletricidade (GTES, 2014), dependem diretamente do Sol. Na Figura 1.1, pode-se ter uma visão do aspecto do Sol.

Figura 1.1: Pulsos do Sol.



Fonte: (NASA, 2017).

Em relação ao uso da energia solar para produzir eletricidade, o primeiro importante acontecimento ocorreu em 1839, quando Edmond Becquerel descobriu o efeito fotovoltaico, em que notou o surgimento de uma diferença de potencial nos terminais de uma célula eletroquímica, ao expô-la à luz solar (GTES, 2014).

Em 1873, Willoughby Smith descobriu que o Selênio possui propriedades fotocondutivas, possibilitando que em 1876, William Grylls Adams e Richard Evans Day, concluísssem que este mesmo elemento produz eletricidade ao ser exposto à luz solar. Em 1883, Charles Fritts, inventou a primeira célula solar da história, feita de Selênio (BLUESOL, 2017).

Somente em 1954, após pesquisas que se iniciaram por volta de 1952, foi apresentada ao mundo a primeira célula solar de Silício (elemento usado atualmente na produção de células fotovoltaicas), produzida em conjunto por Calvin Fuller, Daryl Chapin e Gerald Pearson, pesquisadores da Bell Labs (NREL, 2004). Esta célula, que pode ser vista na Figura 1.2, possuía uma eficiência de conversão da luz solar em eletricidade de aproximadamente 4 %, o que representa menos de um quarto da eficiência das células atuais (ENERGYSAGE, 2017).

Figura 1.2: Propaganda da Bell Solar Battery .



Fonte: (NREL, 2004).

Segundo GTES (2014), a crise do petróleo em 1973 aumentou o interesse na criação de aplicações terrestres para a energia solar fotovoltaica, sendo que seria necessário diminuir o custo de produção de células solares, para tornar o investimento neste segmento viável.

Com o passar dos anos, foram obtidos grandes avanços na produção de células fotovoltaicas, com processos de dopagem, obtendo-se rendimentos em laboratório de cerca de 25 % (GREEN *et al.*, 2013, *apud* GTES, 2014). Consequentemente, têm sido cada vez mais comum de se ver painéis solares instalados nos telhados de residências, além da criação de parques solares, onde são instalados milhares de painéis fotovoltaicos para a produção de energia elétrica em larga escala.

Um exemplo de parque solar em operação no Brasil é o Parque Solar Lapa, localizado na cidade de Bom Jesus da Lapa, estado da Bahia, sendo o maior de seu tipo no país e composto por duas usinas com capacidade total de 158 MW. O empreendimento pode suprir as necessidades de eletricidade de aproximadamente 166 mil famílias numa região que

frequentemente enfrenta períodos de seca, podendo gerar 340 GW por ano e evitando a emissão de 198 mil toneladas de dióxido de carbono (CO₂) na atmosfera (PORTAL SOLAR, 2017). Na Figura 1.3, pode-se ter um vislumbre deste parque solar.

Figura 1.3: Parque Solar Lapa.



Fonte: (PORTAL SOLAR, 2017).

A tendência é que empreendimentos como o Parque Solar Lapa se tornem cada vez mais comuns ao longo dos próximos anos, em especial num país como o Brasil, que apresenta índices solares de 1642 kWh/m², sendo superior a 1300 kWh/m², apresentado pela região de maior incidência solar da Alemanha (maior produtor mundial de energia solar fotovoltaica) (SALAMONI e RÜTHER, 2007, *apud* CABRAL, TORRES e SENNA, 2013).

Levando-se em conta um parque solar como o de Bom Jesus da Lapa, pode-se imaginar que exista um grande esforço dos engenheiros que ali trabalham para que a usina opere com a melhor eficiência possível, de modo que se evitem gastos em demasia com manutenções e que se mantenha um registro de como tudo está ocorrendo em cada ponto do parque.

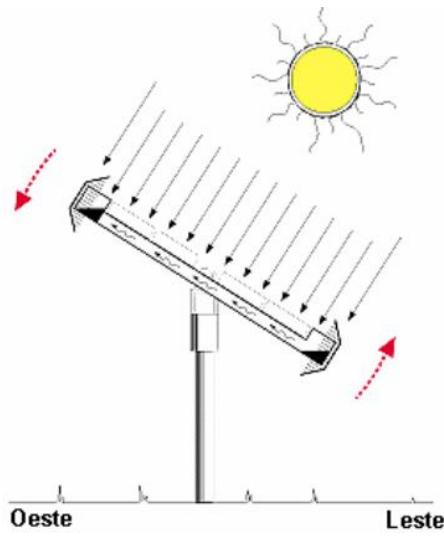
Neste cenário, podem-se buscar várias maneiras de se aumentar o rendimento de um painel solar, como adicionar um sistema de controle para que a placa literalmente possa seguir o Sol, buscando captar o máximo possível de luz solar ao longo do dia. Além disso, um sistema supervisório pode ser usado para a inspeção do sistema formado pela placa e pelo mecanismo do seguidor.

Este sistema de controle pode ser chamado de rastreador ou seguidor solar. Conforme Oliveira (2007), um módulo fotovoltaico produz eletricidade a partir da componente de radiação solar que incide perpendicular ao seu plano, fazendo com que haja uma maior

geração quando o módulo está totalmente de frente para o Sol. Com isso, o uso de um seguidor solar pode aumentar a produção de energia ao se manter os módulos fotovoltaicos perpendiculares aos raios solares durante o maior período de tempo possível.

De acordo com Castañeda (2011), existem dois tipos de rastreadores solares: os passivos e os ativos. Os passivos não utilizam nenhum controle eletrônico, sendo que sua movimentação ocorre por conta da radiação solar que incide sobre o mecanismo que possui em cada extremidade da placa, um vaso cilíndrico abastecido com um fluido mantido a certa pressão. No início do dia o painel está inclinado para a direção Oeste. Como o Sol nasce a Leste, o fluido que está no lado Leste evapora, sendo transferido para o lado Oeste. Dessa forma a placa se move através do desequilíbrio de massa provocado no sistema. Um exemplo de seguidor solar passivo pode ser visto na Figura 1.4.

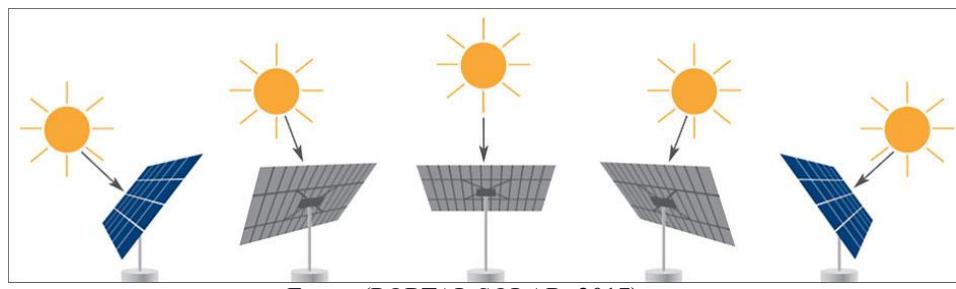
Figura 1.4: Seguidor Solar Passivo.



Fonte: (OLIVEIRA, 2007).

Os rastreadores ativos possuem um sistema eletrônico que controla a posição da placa solar em função da leitura de sensores, como os de luminosidade. O movimento pode ser feito através de atuadores ou motores de passo, por exemplo (LIRA, 2014). Oliveira (2007) ainda cita que o seguidor solar ativo pode trabalhar em um ou dois eixos. O primeiro tem seu eixo na direção Norte-Sul, com seu arranjo girando de Leste a Oeste para se alinhar com o Sol ao longo do dia, como pode ser visto na Figura 1.5. O seguidor de dois eixos possui um segundo eixo, que permite o ajuste da inclinação da placa, podendo se adaptar às inclinações do Sol ao longo do ano.

Figura 1.5: Rastreador Solar Ativo.



Fonte: (PORTAL SOLAR, 2017).

Na literatura existem vários trabalhos que descrevem protótipos de rastreadores solares, sendo que alguns destes implementam também, um supervisório, como no caso do artigo de Ozerdem e Shahin (2014), em que um microcontrolador Arduino é usado para controlar o seguidor, a partir das leituras de sensores de luz chamados LDR (*Light Dependent Resistor*) e os dados relacionados à posição do mecanismo são mostrados numa *Interface Homem Máquina* (IHM) criada em MATLAB®.

Em relação aos supervisórios, estes se baseiam no sistema SCADA (*Supervisory Control and Data Acquisition*), que corresponde a elementos de *hardware* e *software*, permitindo às indústrias controlar processos local e remotamente; monitorar, reunir e processar dados em tempo real; interagir com sensores, válvulas e bombas, através do uso de uma *Interface Homem Máquina* (IHM); e também registrar eventos (INDUCTIVE AUTOMATION, 2017).

Conforme o trabalho de Moreira (2011), o sistema SCADA teve origem a partir de sistemas de telemetria (medição remota de dados) utilizados para controlar a aproximação de trens em bifurcações, selecionando o caminho que poderiam seguir, através de redes cabeadas que possuíam switches eletrônicos nas linhas, enviando sinais para a estação central.

O desenvolvimento da indústria aeronáutica a partir dos anos 1920 provocou o surgimento da necessidade por transmissões de dados sem o uso de cabos, utilizando-se frequência de rádio. Isso ocorria por conta do fato de que muitas vezes os testes com aviões eram realizados em áreas isoladas, pois muitas vezes terminavam nas quedas dos mesmos (MOREIRA, 2011).

Mas, apesar do grande avanço na transmissão de dados sem o uso de cabos, ainda havia um problema a ser resolvido, pois a comunicação citada anteriormente no caso da indústria aeronáutica era unidirecional, ou seja, permitia apenas a coleta de informações, não sendo possível controlar o sistema remotamente (BOYER, 1999, *apud* MOREIRA, 2011).

Somente nos anos 1960, com a criação dos Controladores Lógicos Programáveis (CLP), é que foi possível realizar operações bidirecionais (PENIN, 2007, *apud* MOREIRA, 2011).

O termo SCADA surgiu por volta dos anos 1970, como referência aos sistemas de controle, que cada vez mais deixavam de ser cabeados. Com o uso de computadores, pode-se adicionar *interfaces* que mostravam dados através de valores numéricos e gráficos (BOYER, 1999, *apud* MOREIRA, 2011).

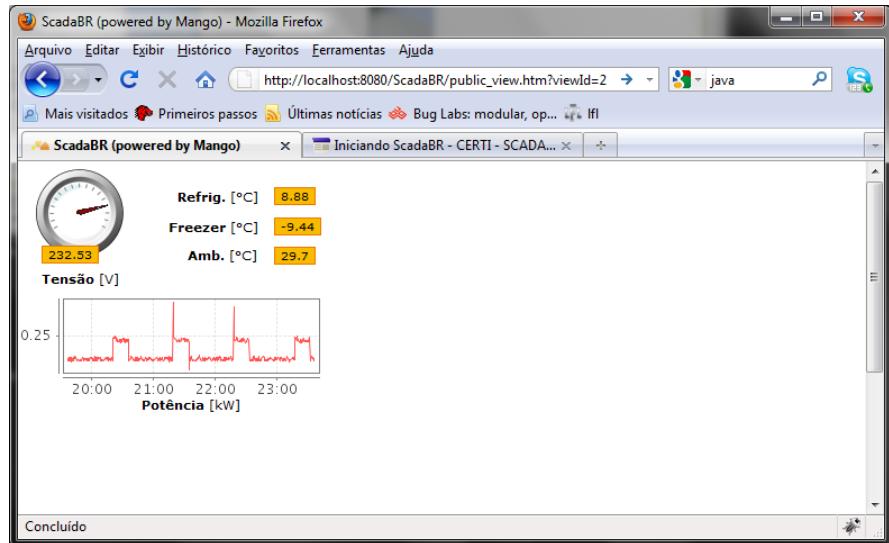
Atualmente, a comunicação via rádio está defasada, sendo que as comunicações são feitas via satélite, *bluetooth* ou *internet*, permitindo-se inclusive, o acesso remoto a SCADAS, podendo-se executar inúmeras ações simultâneas visando o controle e estabilidade de um sistema (MOREIRA, 2011).

Com a popularização do sistema SCADA, surgiram inúmeros *softwares* relacionados à supervisão e automação de processos. De acordo com trabalho de Raysaro (2012), há sistemas SCADA proprietários (pagos), como *LabVIEW®* e *Indusoft®*, e também há sistemas SCADA *open-source* (código aberto), tais como Mango, Argos e ScadaBR.

Para este trabalho, foi escolhido como supervisório, o ScadaBR, que é um *software* livre, gratuito e de código-fonte aberto, criado no Brasil, para aplicações de automação, aquisição de dados e controle, tendo como público-alvo profissionais da área de controle, universitários e também indústrias (SCADABR, 2017).

O ScadaBR é baseado no *software* Mango, permitindo a criação de uma IHM que permite a leitura, controle e aquisição de dados através de diversos protocolos de comunicação, como *Modbus Serial*, *TCP/IP*, *IEC*, etc. Na Figura 1.6, pode-se observar uma representação gráfica criada no ScadaBR, que mostra o consumo de energia e as temperaturas em um sistema de refrigeração.

Figura 1.6: Representação Gráfica em ScadaBR.



Fonte: (CERTI – SCADABR, 2009).

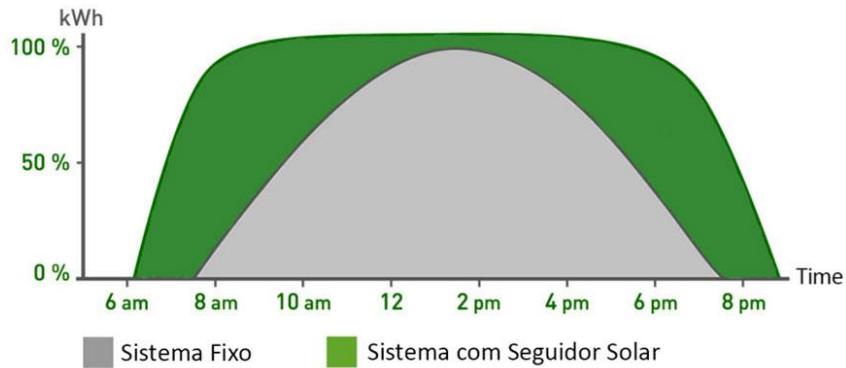
1.1. OBJETIVOS

Utilizar a plataforma Arduino para controlar um Rastreador Solar e envio de dados do mesmo para um supervisório desenvolvido em ScadaBR, de modo a se obter um protótipo funcional e de baixo custo.

1.2. JUSTIFICATIVA

Este trabalho possui importância por conta do fato de que há uma possibilidade de se aumentar a eficiência na geração de energia solar fotovoltaica através do uso de um Rastreador Solar. Conforme dados da companhia catalã Valdoreix Greenpower citados pelo site Portal Solar, o uso de seguidores solares pode acarretar num ganho de 20 a 45 % na produção de energia, se comparado a sistemas com placas fixas, conforme apresentado na Figura 1.7.

Figura 1.7: Comparação de Sistema de Placa Fixa e Rastreador Solar.



Fonte: Adaptado de (VALDOREIX GREENPOWER, 2017).

Segundo Vilela, Fraidenraich e Tiba (2003) *apud* Oliveira (2007), o uso de rastreadores solares, até mesmo em sistemas de pequeno porte, pode ter vantagens substanciais, como por exemplo, ao poder abastecer um sistema de bombeamento de água em uma fazenda.

Nesse segmento, é de grande importância o uso de um supervisório, tanto para o monitoramento de variáveis como a posição da placa solar, a quantidade de luz recebida pelos sensores e potência produzida, quanto para o controle da voltagem a ser direcionada à bateria, após uma sinalização ser mostrada na *interface* do *software*. Eis que o ScadaBR, que é um programa de fácil utilização, gratuito e produzido no Brasil, pode ser usado como alternativa à softwares comerciais, que apresentariam os mesmos resultados esperados.

Com tudo isso, o protótipo produzido neste trabalho serve de referência para aplicações reais, e também para futuros trabalhos acadêmicos, que poderão vir a fazer melhorias, podendo usar outras opções de controladores e supervisórios.

1.3. ESTRUTURA DO TRABALHO

O trabalho está dividido como descrito abaixo:

No Capítulo 2 são apresentados os componentes de um seguidor solar, assim como seus diferentes tipos, além de informações pertinentes a sistemas supervisórios.

No Capítulo 3 é descrito o desenvolvimento do Seguidor Solar e do sistema supervisório baseado no *software* ScadaBR.

No Capítulo 4 são apresentados os resultados obtidos durante os cinco dias de medições realizados.

No Capítulo 5 apresentam-se as conclusões pertinentes ao Seguidor Solar e ao sistema supervisório desenvolvidos, possibilitando apresentar ideias para melhorias a serem executadas em trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

Baseado nas literaturas consultadas, este capítulo apresenta informações pertinentes ao desenvolvimento do Seguidor Solar e seu Sistema Supervisório, de modo que se tenha uma base sólida, para produzir adequadamente o protótipo idealizado.

2.1. ARDUINO

Arduino é uma plataforma de código aberto, facilitando sua aplicação por meio do uso do *hardware* e *software*, sendo capaz de ler entradas (dados de sensores) e usá-las para acionar saídas (um alarme sonoro, por exemplo) de acordo com a programação determinada pelo usuário nas linguagens *C/C++* (ARDUINO, 2018).

Esta placa é uma boa opção para ser o controlador do sistema, dado o seu tamanho reduzido, além de ser uma plataforma de baixo custo, de fácil aprendizado e que pode ser utilizada nos mais variados sistemas operacionais, como *Windows*, *MacOS* e distribuições *Linux* em geral (ARDUINO, 2018).

É importante lembrar que existe uma variada gama de Arduino, como o UNO, DUE, Mega, Nano, Pro Mini, etc. O modelo mais popular é o UNO (que pode ser visto na Figura 2.1), por conta da sua versatilidade, facilidade de uso e aquisição, o que foi decisivo para a sua escolha em detrimento dos outros modelos. O microcontrolador do Arduino UNO é o ATmega328, da família AVR da Atmel, que possui 14 pinos digitais e 6 analógicos (ROCHA, 2017).

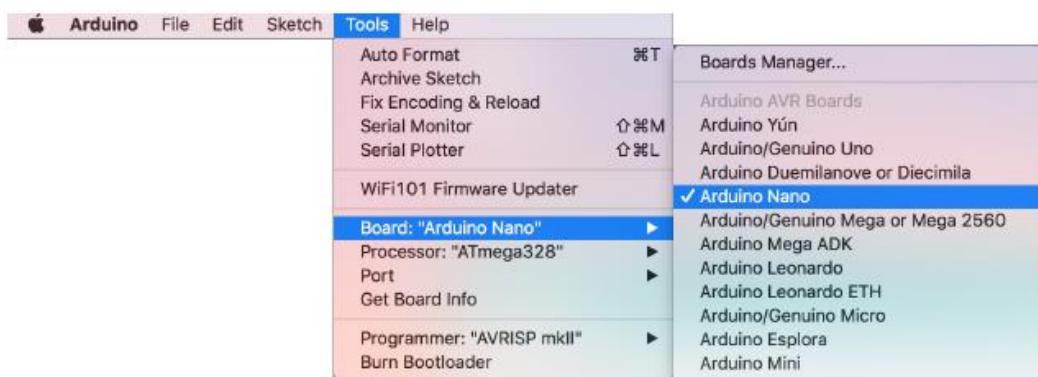
Figura 2.1: Arduino UNO.



Fonte: (ARDUINO, 2017).

A programação do Arduino é realizada por meio do Arduino *IDE*, que pode ser baixada diretamente no site oficial do Arduino gratuitamente. Ao se conectar o cabo de comunicação da placa com o computador, basta selecionar a placa correta no menu *Tools* para enviar o código à mesma, como pode ser observado na Figura 2.2.

Figura 2.2: Seleção da Placa na *IDE*.

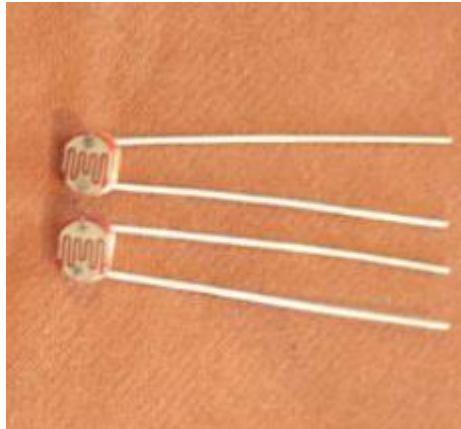


Fonte: (ROCHA, 2017).

2.2. SENSORES LDR

O componente LDR (*Light Dependent Resistor*) é um resistor que varia a sua resistência em função da luminosidade que incide sobre o mesmo, ou seja, quanto mais luz incidir sobre o LDR, menor será a sua resistência. De acordo com a sua especificação técnica, sob uma luminosidade de 10 lux, a uma temperatura de 25 °C, a sua resistência estará na faixa de 8 a 20 kΩ. No escuro (luminosidade a 0 lux), a resistência poderá chegar a 1.0 MΩ (CDS, 2018). O sensor pode ser observado na Figura 2.3.

Figura 2.3: Light Dependent Resistor.



Fonte: (CDS, 2018).

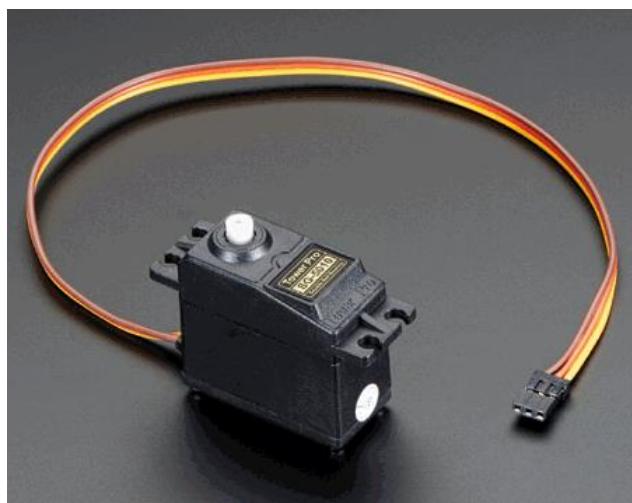
Bates (2013) explica que um LDR é semelhante ao olho humano em relação à sua resposta espectral a variações luminosas, pois pode demorar a se adaptar a uma nova intensidade luminosa a qual é exposto.

2.3. SERVOMOTOR

Segundo McRoberts (2011), um servomotor é um motor capaz de controlar com precisão a posição de seu eixo através de um sistema realimentado, ou seja, uma retroalimentação que avisa ao servo se deve manter a posição ou variá-la. De acordo com Santos (2007), *apud* Marinho (2016), este controle é feito a partir de um PWM (*Pulse Width Modulation*) no fio de controle.

No PWM é enviado um pulso máximo, que decai para um valor mínimo numa determinada taxa de repetição. O servomotor aguarda cerca de 20 milissegundos por alguma ordem de mudança de posição (um pulso), sendo este tempo o período e consequentemente a frequência será de 50 Hz. A quantidade de graus que o motor girará, será determinada pelo comprimento do pulso, que caso seja menor do que 1,5 ms a posição tenderá a 0° e caso seja maior, tenderá a 180°, para o caso de um servo que tenha uma faixa de movimento de 0° a 180°, como o modelo apresentado na Figura 2.4.

Figura 2.4: Servomotor SG-5010.



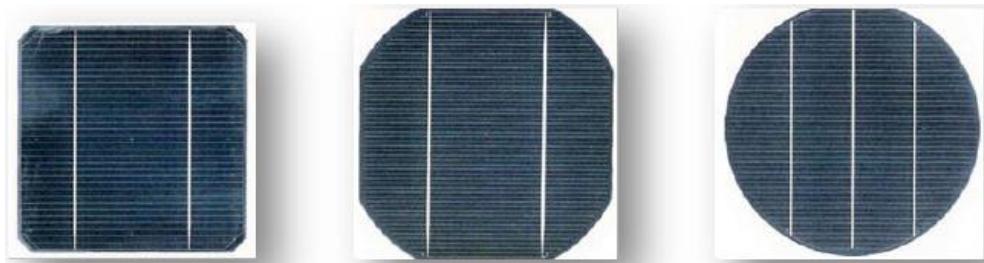
Fonte: (DATASHEETCAFE, 2015).

2.4. PLACA FOTOVOLTAICA

Uma placa fotovoltaica é um aparato que possui o objetivo de produzir eletricidade através da captação de luz solar. O processo responsável pela conversão de luz do Sol em energia elétrica é o efeito fotovoltaico, que ocorre quando um determinado material exposto a uma fonte luminosa faz com que os elétrons contidos no material sejam excitados e entrem em movimento, e dessa forma, produzindo corrente elétrica (RUTHER, 2004 *apud* MARINHO, 2016).

De acordo com o Grupo de Trabalhos de Energia Solar (2014), as células que compõem um módulo fotovoltaico são geralmente feitas de Silício, sendo que apenas uma célula não é capaz de gerar uma potência elétrica elevada, de modo que é necessário unir certa quantidade de células, que formarão um módulo fotovoltaico, que então gerará eletricidade. Na Figura 2.5 podem-se observar exemplos de células fotovoltaicas.

Figura 2.5: Células Fotovoltaicas de Silício Monocristalino.



Fonte: (BLUESOL, 2017).

Conforme publicação do Centro de Referência para Energia Solar e Eólica Sérgio de Salvo Brito (2006), a corrente produzida por um módulo fotovoltaico é diretamente proporcional à luminosidade, sendo que a tensão, por sua vez, é inversamente proporcional. A temperatura de uma célula fotovoltaica tem grande influência na produção de energia, uma vez que o seu aumento faz com que haja uma queda de eficiência. Por conta disso, painéis fotovoltaicos tendem a ter boa produção de eletricidade mesmo em dias nublados, quando a temperatura ambiente tende a ser menor.

2.5. SEGUIDOR SOLAR

2.5.1. Geometria Solar

O planeta Terra circunda o Sol ao longo do ano descrevendo uma órbita de forma elíptica num plano com inclinação de aproximadamente $23,5^\circ$ com relação ao plano equatorial. Esta inclinação é responsável pelas mudanças de elevação do Sol no horizonte ao longo do dia (MANUAL DE ENGENHARIA PARA SISTEMAS FOTOVOLTAICOS, 1999).

O ângulo existente entre os raios solares e a linha do Equador é chamado Declinação Solar (MARQUES, 2012, *apud* MORAES, 2012), podendo variar de $-23,5^\circ$ a $23,5^\circ$ ao longo do ano, sendo que a soma da declinação com a latitude local, resulta na trajetória do movimento aparente do Sol para um determinado dia, numa dada localidade na Terra (MANUAL DE ENGENHARIA PARA SISTEMAS FOTOVOLTAICOS, 1999).

A inclinação de $23,5^\circ$ do eixo terrestre influí diretamente na maneira como os raios solares atingem a superfície (MARQUES, 2012, *apud* MORAES, 2012). Isso é percebido ao se analisar as relações geométricas existentes entre os raios solares e a superfície terrestre,

como no caso do Ângulo de Incidência (γ), que é formado entre os raios do Sol e a superfície de captação.

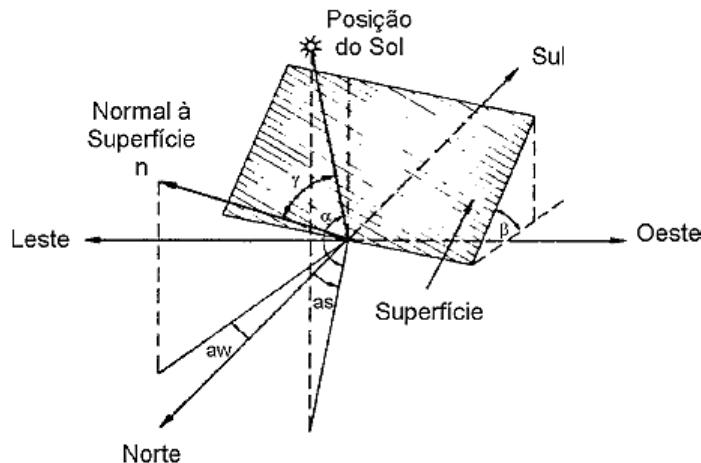
O Ângulo Azimutal da Superfície (a_w) é aquele entre a projeção da normal à superfície no plano horizontal e a direção norte-sul, podendo variar de -180° a 180° , sendo positivo quando a projeção estiver à esquerda do Norte e negativa se estiver à direita (MANUAL DE ENGENHARIA PARA SISTEMAS FOTOVOLTAICOS, 1999).

O Ângulo Azimutal do Sol (a_s) é aquele compreendido entre a projeção dos raios solares no plano horizontal e a direção Norte-Sul, podendo também variar de -180° a 180° , assim como no caso do Ângulo Azimutal de Superfície.

A Altura Solar (α) é o ângulo entre o raio solar e a projeção do mesmo sobre o plano horizontal. Tem-se ainda a Inclinação da Superfície (β), que está diretamente relacionada ao ângulo de uma placa solar, por exemplo, sendo o ângulo entre o plano da superfície e a horizontal.

Todas essas relações podem ser visualizadas na Figura 2.6.

Figura 2.6: Coordenadas de Orientação da Superfície.



Fonte: (MANUAL DE ENGENHARIA PARA SISTEMAS FOTOVOLTAICOS, 1999).

De acordo com O Manual de Engenharia para Sistemas Fotovoltaicos (1999) é recomendável que em localidades do hemisfério sul o arranjo fotovoltaico esteja direcionado para o norte geográfico com uma inclinação próxima à latitude local. Dessa forma, tanto para um sistema de placa fixa quanto para um sistema seguidor solar, poderá se obter o máximo possível de captação ao longo do ano, sendo que o Sol fará o seu caminho de leste a oeste num ângulo de 180° , estando ao meio-dia, aproximadamente na posição de 90° .

Uma inclinação próxima à latitude local além de possibilitar uma maximização na captação de luz solar, permite ainda que a chuva e o vento façam a limpeza dos módulos de maneira natural. Segundo BlueSol (2017) não é recomendável uma inclinação menor que 10°, pois a limpeza natural ficará prejudicada principalmente em sistemas isolados da rede, onde a manutenção é reduzida.

2.5.2. Características de um Rastreador Solar

Um painel solar gera eletricidade a partir da componente de radiação solar que incide perpendicularmente ao seu plano, fazendo com que ocorra uma maior geração quando o painel está literalmente de frente para o Sol (OLIVEIRA, 2007).

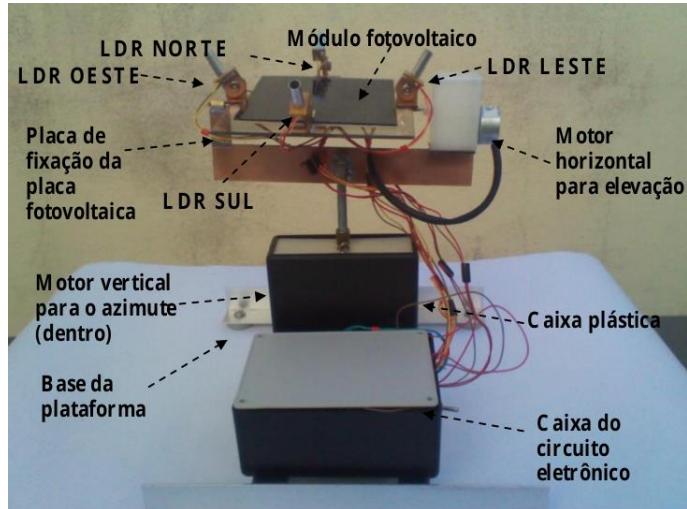
Um modo de se maximizar a produção de energia solar que pode ser aplicado é fazendo-se uso de um seguidor solar, que terá a função de manter os painéis solares sempre perpendiculares ao Sol, devendo ter uma boa precisão, além de ser capaz de operar por longos períodos de tempo.

Para a implementação de um Rastreador Solar, se faz necessário o uso de um controlador, que deverá seguir uma lógica para que o sistema opere corretamente, podendo demonstrar dados em uma *Interface Homem Máquina* (IHM) para a devida supervisão de dados como leitura de sensores, posição dos motores, potência gerada pela placa solar, etc.

A lógica seguida por Marinho (2016), por exemplo, calcula a diferença das leituras dos LDR e as compara com uma determinada tolerância. Se essa diferença for maior que a tolerância estipulada, o Rastreador Solar deverá permanecer parado. Caso contrário, deverá se mover para o lado de maior luminosidade até que o critério de parada descrito seja atingido.

Na literatura existem diversos artigos e teses sobre o uso de um Rastreador Solar. Ribeiro, Prado e Gonçalves (2012), produziram um seguidor utilizando um microcontrolador PIC 16F877, baseando-se nas leituras de 4 LDR (sensores de luz) posicionados nas posições norte, sul, leste e oeste, como pode ser observado na Figura 2.7. Foram usados dois motores de passo, um para controlar o movimento norte/sul da placa e o outro para regular a posição leste/oeste da mesma e não foi mencionado o uso de algum *software* supervisório.

Figura 2.7: Rastreador Solar Controlado por PIC 16F877.



Fonte: (RIBEIRO, PRADO E GONÇALVES, 2012).

Oliveira (2007) produziu um protótipo em que foram usados dois LDR para o rastreamento da posição do Sol, sendo que estes enviam as leituras para um circuito eletrônico de controle composto por vários microcontroladores montados em uma PCI (Placa de Circuito Impresso). Foi utilizado um motor de passo para controlar a posição da placa a partir da lógica implementada no circuito de controle. Conforme apresentado na Figura 2.8, verifica-se que o arranjo foi apoiado em blocos de concreto, de modo a se obter uma inclinação próxima à latitude da cidade de Recife, local onde o trabalho foi realizado.

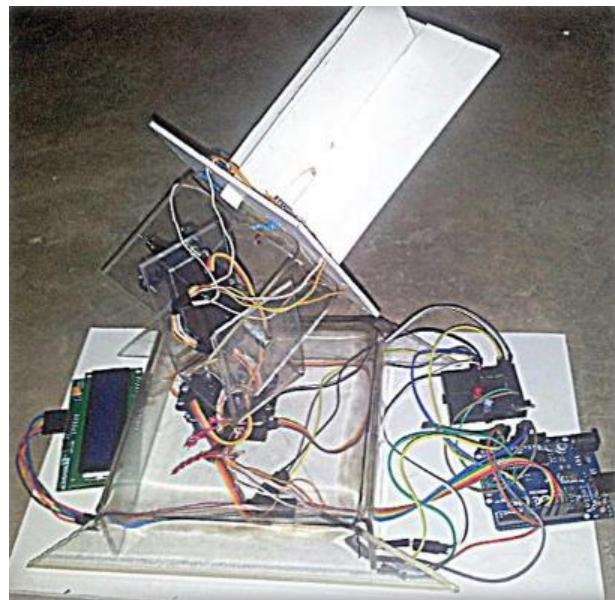
Figura 2.8: Protótipo de Rastreador Solar de Baixo Custo.



Fonte: (OLIVEIRA, 2007).

Hossain *et al.* (2015) utilizou a plataforma Arduino para o controle de um Rastreador Solar, cujos movimentos de inclinação e rotação da placa são feitos por servomotores a partir das leituras de quatro LDR. As informações coletadas pelos sensores são mostradas num *display LCD 20x4*, como apresentado na Figura 2.9.

Figura 2.9: Rastreador Solar Controlado por Arduino.



Fonte: (HOSSAIN *et al.*, 2015).

Um estudo realizado por Ozerdem e Shahin (2014) efetuou testes com diferentes técnicas de controle em um Rastreador Solar, cujo sistema é composto por um Arduino UNO, dois sensores LDR e um servomotor, além de uma miniplaca solar. A primeira técnica de controle foi chamada de tradicional, no qual utiliza o ambiente MATLAB®/Simulink na programação do Arduino para controlar o mecanismo de seguimento do Sol em função das leituras de luminosidade dos LDR.

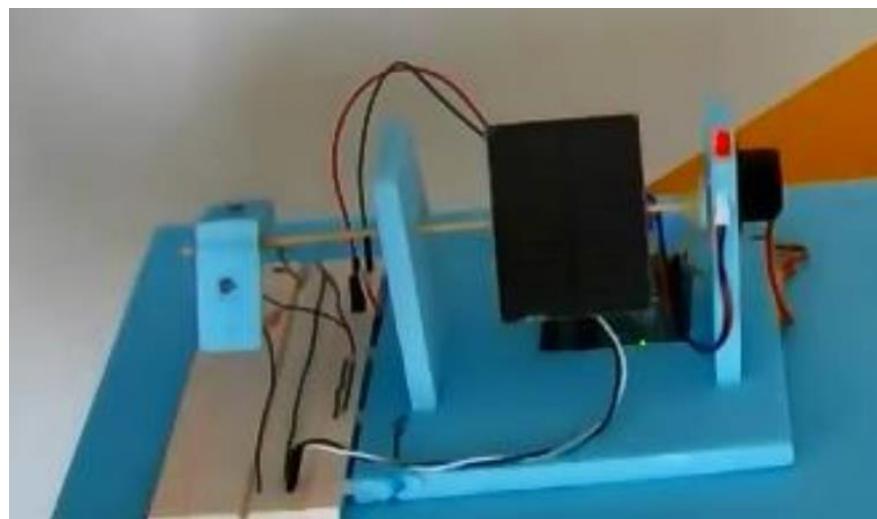
O segundo método utilizou-se do controle PID (Proporcional, Integral e Derivativo), de modo a regular o comportamento do sistema no domínio do tempo. Os autores optaram por utilizar apenas a parte integral, já que o movimento do Sol é lento ao longo do dia, podendo-se descartar as partes Derivativa e Proporcional, sendo que ambos segundo os autores são mais indicados a sistemas com mudanças mais rápidas no seu comportamento.

O terceiro método buscava a posição solar de acordo com a localização do seguidor e da hora atual. Para este caso foram utilizadas as coordenadas de latitude, longitude e altitude do Chipre, país onde o estudo foi realizado. Com base nos dados citados, foi encontrada uma

função aproximada a partir de um algoritmo desenvolvido em MATLAB®, no qual resultou que este método tecnicamente é mais eficiente dentre os métodos testados, sendo que o primeiro método, que utilizava sensores LDR, foi o menos eficiente, pois a estabilidade e o desempenho não são boas o suficiente para uma aplicação real.

O trabalho de Ozerdem e Shahin (2014) apresenta a versatilidade do Arduino, ao programá-lo via MATLAB, e os resultados dos experimentos num supervisório criado com base no mesmo *software*, sendo observado nas Figuras 2.10 e 2.11.

Figura 2.10: Rastreador Solar Controlado por Arduino UNO.



Fonte: (OZERDEM e SHAHIN, 2014).

Figura 2.11: Interface de Supervisão e Controle em MATLAB®.



Fonte: (OZERDEM e SHAHIN, 2014).

2.6. SUPERVISÓRIO

2.6.1. Sistemas SCADA

O uso de um supervisório pode ser importante tanto em um sistema fotovoltaico conectado à rede como em um sistema isolado, sendo neste necessário o uso de baterias. Dessa forma, pode-se ter uma visão do que ocorre nos sistemas mencionados, permitindo que sejam tomadas medidas adequadas para determinadas situações.

Conforme Filho (2011), *apud* Raysaro (2012), SCADA (*Supervisory Control and Data Acquisition*) é um sistema de supervisão de processos, que coleta dados através do uso de um controlador, os formata e envia para o operador das mais variadas maneiras, que visualizará as informações numa tela de computador.

De acordo com o Manual do *Software ScadaBR* (2010), um típico sistema SCADA deve oferecer ao usuário meios de comunicação com os equipamentos, um sistema para registro contínuo de dados e uma IHM (*Interface Homem Máquina*) que apresentará todos os dados pertinentes ao processo monitorado, além de elementos gráficos como ícones, botões e displays.

Da Silva e Bianchini (2014) fazem um rápido comparativo de supervisórios em seu trabalho, citando o Cimplicity HMI, cuja licença é paga, sendo que seu preço varia de acordo com o número de entradas e saídas que serão usadas no servidor. Os autores ainda citam como supervisório proprietário o Elipse E3, recomendado para empresas que usam equipamentos de diferentes fabricantes. Como opções gratuitas são colocados o Lintouch e o ScadaBR, sendo este último o escolhido para execução de seu trabalho, por ser grátis e possuir uma grande quantidade de ferramentas para se trabalhar, e dessa forma, se encaixando perfeitamente no perfil de sua pesquisa sobre o desenvolvimento de um aplicativo para supervisão de pacientes com afasia, ou seja, dificuldade para se comunicar através da fala.

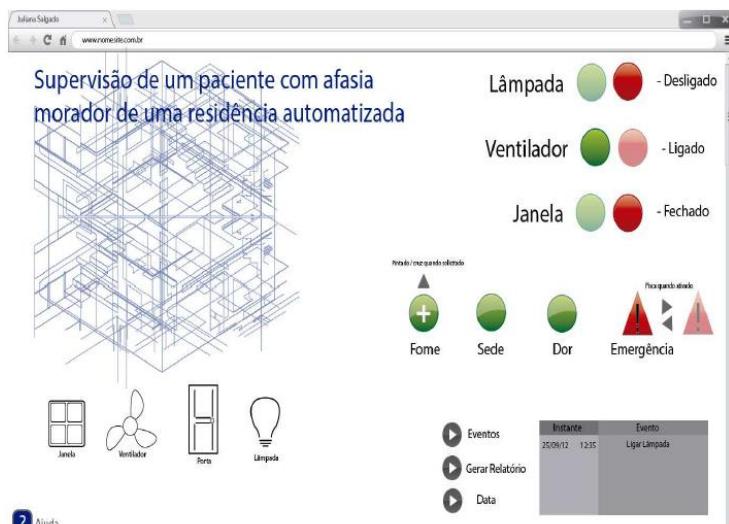
O aplicativo poderia ser instalado num smartphone utilizado pelo paciente, onde permite a interação do cuidador com o paciente doente mesmo à distância através de um supervisório em ScadaBR. Na Figura 2.12 pode-se ter uma ideia da *interface* do aplicativo para *smartphone*, enquanto na Figura 2.13 é possível ver a *interface* construída no ScadaBR.

Figura 2.12: Interface de Aplicativo Usado pelo Paciente.



Fonte: (DA SILVA e BIANCHINI, 2014).

Figura 2.13: Representação Gráfica em ScadaBR Usada Pelo Cuidador.



Fonte: (DA SILVA e BIANCHINI, 2014).

O trabalho de Raysaro (2012) se propõe a apresentar a aplicabilidade do ScadaBR no meio industrial, ao desenvolver um protótipo para teste na Heineken Brasil, fabricante de cervejas, onde são monitoradas as temperaturas de entrada e saída de dois tanques de refrigeração que usam etanol. Na Figura 2.14 pode-se conferir como ficou a IHM do processo feita em ScadaBR.

Figura 2.14: Supervisório de Temperaturas.



Fonte: (RAYSARO, 2012).

Assunção (2016) produziu um controlador *Fuzzy* para o controle de pressão numa adutora de água em Florianópolis com supervisório em ScadaBR, cuja representação gráfica pode ser vista na Figura 2.15.

Figura 2.15: Supervisório para Controle de Pressão em Adutora de Água.



Fonte: (ASSUNÇÃO, 2016).

2.6.2. Protocolo de Comunicação

Um protocolo de comunicação é uma norma de controle de transmissão de dados, sendo capaz de controlar o formato e o significado das informações trocadas. Existem diversos tipos de protocolos, sendo o *Modbus* um dos mais populares (TANEMBAUM, 1997, *apud* RAYSARO, 2012).

O protocolo *Modbus* utiliza o princípio de comunicação mestre/escravo, onde o mestre – CLP ou computador, por exemplo – envia perguntas aos seus escravos – sensores, atuadores ou controladores – para que estes enviem uma resposta, de modo que o mestre faça a leitura dos dados, podendo agir, se assim determinado for, para que se obtenha o resultado desejado.

Este protocolo utiliza os padrões RS-232, RS-485 ou *Ethernet* como meios físicos de comunicação, permitindo um mestre e até 247 escravos. Os padrões RS citados usam a transmissão serial, em que os *bits* individuais de informação são transmitidos em longas séries, sendo usado em conexões de impressoras por exemplo e até mesmo num Arduino UNO com um computador. O protocolo *Modbus* serial apresenta duas variações em função do número de bytes transmitidos, podendo ser *Modbus* – RTU ou *Modbus* – ASCII, sendo o primeiro o mais utilizado (CLARKE, 2004, *apud* RAYSARO, 2012).

2.6.3. Funcionalidades do ScadaBR

De acordo com o Manual do *Software* ScadaBR (2010), o ScadaBR é uma aplicação baseada em Java, que pode funcionar em sistemas operacionais como *Windows* e em distribuições *Linux* em geral, como o Ubuntu, podendo ser executado a partir de um servidor de aplicações, que neste caso é o Apache Tomcat. O ScadaBR pode ser acessado através de um navegador de *Internet* como *Mozilla Firefox*, *Google Chrome* ou *Opera*.

Para a construção de um supervisório no ScadaBR é necessária a criação de um *Data Source*, que possui a função de receber os dados, funcionando de acordo com as configurações de protocolo de comunicação, *baud rate* (velocidade da transmissão de dados no caso de uma comunicação serial), codificação (RTU ou ASCII) (MANUAL DO SOFTWARE SCADABR, 2010). Na Figura 2.16 é mostrada a configuração de um *Data Source* com protocolo *Modbus IP*, em que *Internet* é usada para a transmissão de dados e insere-se o *IP* na posição *Host*.

Figura 2.16: Configuração de *Data Source Modbus IP*.



Fonte: (ASSUNÇÃO, 2016).

Criado o *Data Source*, podem-se criar *Data Points*, que são pontos que receberão e armazenarão os valores históricos a serem analisados (MANUAL DO SOFTWARE SCADABR, 2010). É na configuração dos *Data Points* que podem ser feitas conversões de valores. Na Figura 2.17 pode-se observar a criação de um *Data Point*.

Figura 2.17: Exemplo de Criação de *Data Point*.



Fonte: (ASSUNÇÃO, 2016).

O monitoramento dos *Data Points* criados pode ser feito de duas maneiras: por meio de uma *Watchlist* (Figura 2.18) ou de uma Representação Gráfica (Figuras 2.13, 2.14 e 2.15). A *Watchlist* é literalmente uma lista em que pode ser acompanhada em tempo real a mudança dos valores monitorados sem a necessidade de se atualizar a página do navegador. A Representação Gráfica permite a criação de uma IHM utilizando-se imagens animadas, por exemplo, tornando a visualização dos dados mais amigável (MANUAL DO SOFTWARE SCADABR, 2010).

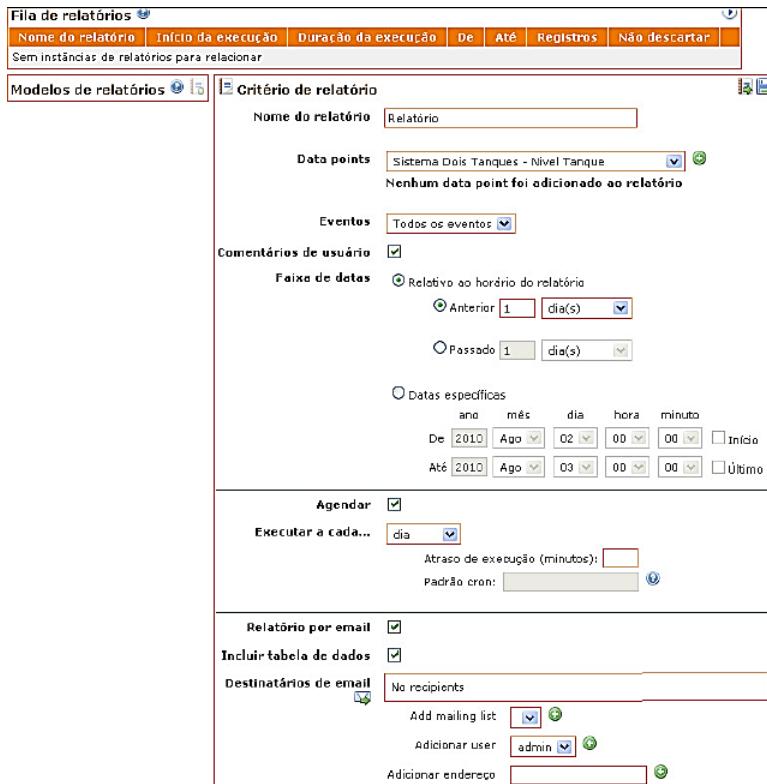
Figura 2.18: Exemplo de *Watchlist* em ScadaBR.

Watch list		(sem nome)	15:04:16			
	Primeiro Data Source - Primeiro Alfanumérico		Escreva aqui!	15:04:16		
	Primeiro Data Source - Primeiro Binário	0	15:04:16			
	Primeiro Data Source - Primeiro Multi-Estados	2	15:04:16			
	Primeiro Data Source - Primeiro Numérico	39,62	15:04:16			

Fonte: (MANUAL DO SOFTWARE SCADABR, 2010).

Para se obter uma melhor visualização dos dados após determinado período de tempo é possível a produção de relatórios. De acordo com Manual do Software ScadaBR (2010), é possível o agendamento para a geração automática de relatórios, além de se poder enviá-los para um endereço de *e-mail*, como pode ser visto na Figura 2.19.

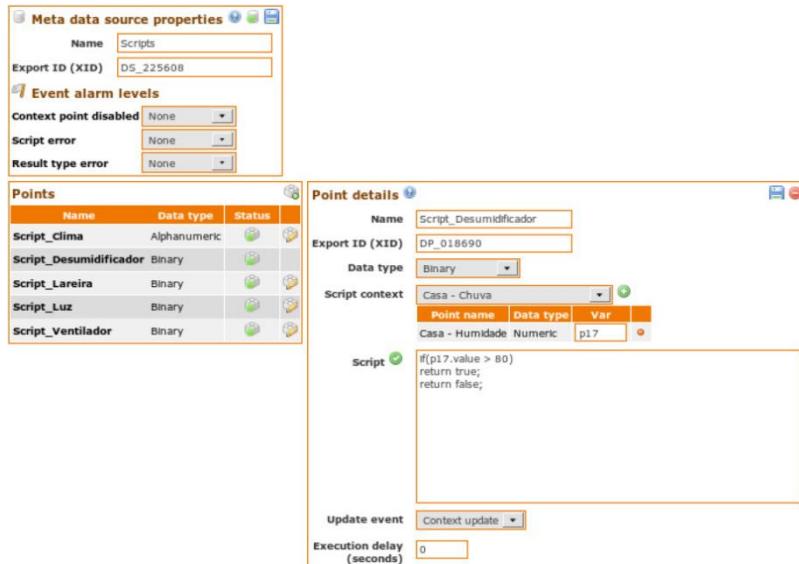
Figura 2.19: Configuração de Geração de Relatórios.



Fonte: (MANUAL DO SOFTWARE SCADABR, 2010).

Um ponto importante a ser citado sobre o ScadaBR está em relação aos *Scripts* (Figura 2.20), que permitem criar lógicas computacionais do tipo “se algo acontecer, faça algo” (RAYSARO, 2012).

Figura 2.20: Criação de Script em ScadaBR.



Fonte: (MANUAL DO SOFTWARE SCADABR, 2016).

Os alarmes também possuem grande importância no ScadaBR, pois são acionados quando ocorre alguma medição de valor fora dos parâmetros preestabelecidos. Dessa forma, se faz necessária a intervenção de um operador para a solução do problema detectado (RAY SARO, 2012). Um exemplo de lista de alarmes pode ser visto na Figura 2.21.

Figura 2.21: Lista de Alarmes em ScadaBR.

Id	Nível de alarme	Tempo	Mensagens	Reconhecer todos			Silenciar todos
					Tempo de inatividade		
693	!	17:38:17	Acima do Limite Superior	17:38:18 - Retornou ao normal	✓	🔇	✖
692	!	17:38:13	Abaixo do Limite Inferior	17:38:14 - Retornou ao normal	✓	🔇	✖
691	!	17:38:05	Acima do Limite Superior	17:38:06 - Retornou ao normal	✓	🔇	✖
690	!	17:37:51	Acima do Limite Superior	17:38:00 - Retornou ao normal	✓	🔇	✖
689	!	17:37:37	Abaixo do Limite Inferior	17:37:38 - Retornou ao normal	✓	🔇	✖
688	!	17:37:11	Acima do Limite Superior	17:37:12 - Retornou ao normal	✓	🔇	✖
687	!	17:37:01	Acima do Limite Superior	17:37:04 - Retornou ao normal	✓	🔇	✖
685	!	17:36:55	Acima do Limite Superior	17:36:56 - Retornou ao normal	✓	🔇	✖
683	!	17:36:31	Abaixo do Limite Inferior	17:36:32 - Retornou ao normal	✓	🔇	✖
684	!	17:36:41	Acima do Limite Superior	17:36:46 - Retornou ao normal	✓	🔇	✖

Fonte: (RAY SARO, 2012).

3. METODOLOGIA

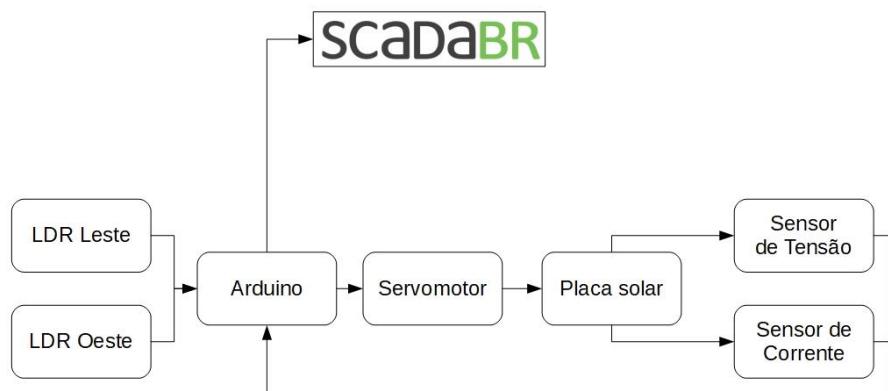
Neste capítulo, serão executados os objetivos deste trabalho, de modo que sejam usadas as referências encontradas na literatura estudada para a produção do protótipo do seguidor solar baseado em Arduino, cujos dados coletados serão demonstrados no supervisório desenvolvido em ScadaBR.

3.1. LÓGICA DE FUNCIONAMENTO

O Arduino é o responsável por fazer o seguidor solar funcionar corretamente a partir de sua programação disponibilizada no Apêndice e transmitir os dados dos sensores via comunicação serial para o ScadaBR. Por se tratar de um projeto de código aberto, existem modelos compatíveis com o Arduino original, sendo o utilizado neste trabalho chamado *Blackboard UNO* e de fabricação brasileira.

O funcionamento do sistema está representado pelo diagrama de blocos simplificado na Figura 3.1. A partir das leituras dos LDR (*Light Dependent Resistor*) efetuadas por meio do Arduino, é enviado um comando ao servomotor para se movimentar a leste, oeste ou permanecer parado. A placa solar é conectada ao servomotor e sua tensão e corrente são medidas em tempo real. Todas as leituras dos sensores, assim como a posição do servomotor são apresentadas em tempo real no supervisório desenvolvido no *software* ScadaBR.

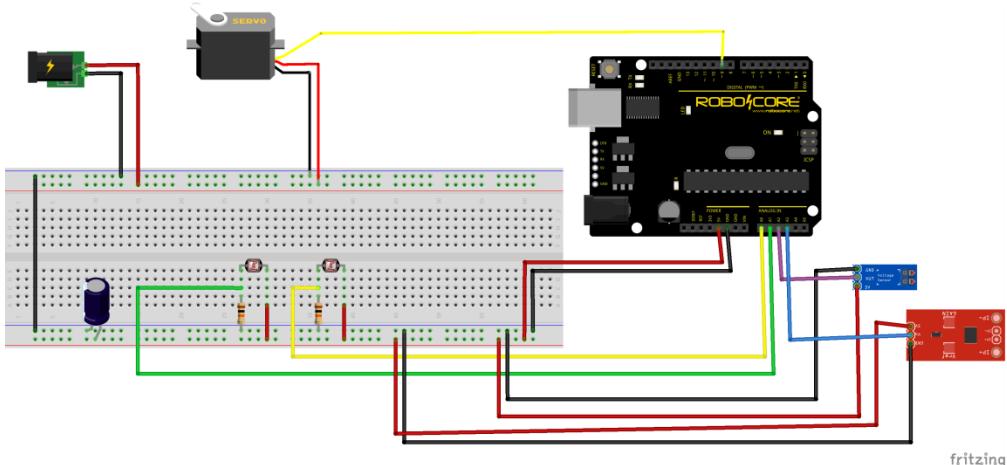
Figura 3.1: Diagrama de Blocos do Rastreador Solar.



Fonte: Próprio Autor.

A Figura 3.2 representa a montagem final em protoboard do Seguidor Solar, onde o servomotor possui alimentação externa de 5 V_{CC} estando conectado ao pino digital 9 do Arduino, sendo uma saída PWM (*Pulse Width Modulation*). O LDR Leste está conectado ao pino analógico A0 e o LDR Oeste ao pino analógico A1. Os sensores de tensão (em azul) e de corrente (em vermelho) estão conectados aos pinos A2 e A3, respectivamente. Todos os sensores são alimentados por 5 V_{CC} provenientes do próprio Arduino. Há ainda um capacitor eletrolítico de 3300 μ F para estabilização da tensão do Arduino, que por sua vez é alimentado pela porta USB do computador.

Figura 3.2: Representação da Montagem em Protoboard do Rastreador Solar.



Fonte: Próprio Autor.

Os LDR Leste e Oeste estão conectados em série com resistores de 10 k Ω aos pinos analógicos A0 e A1 do Arduino formando divisores de tensão. Isso faz com que a tensão de entrada nas portas analógicas do Arduino não exceda 5 V_{CC}, sendo que o resultado obtido para implementação da lógica de programação será mostrado em *bits*, por conta do ADC (*Analog to Digital Converter*) do Arduino, que possui uma resolução de 10 *bits*, fazendo com que valores de 0 a 1023 *bits* (0 a 5 V) representem a intensidade luminosa captada pelos LDR.

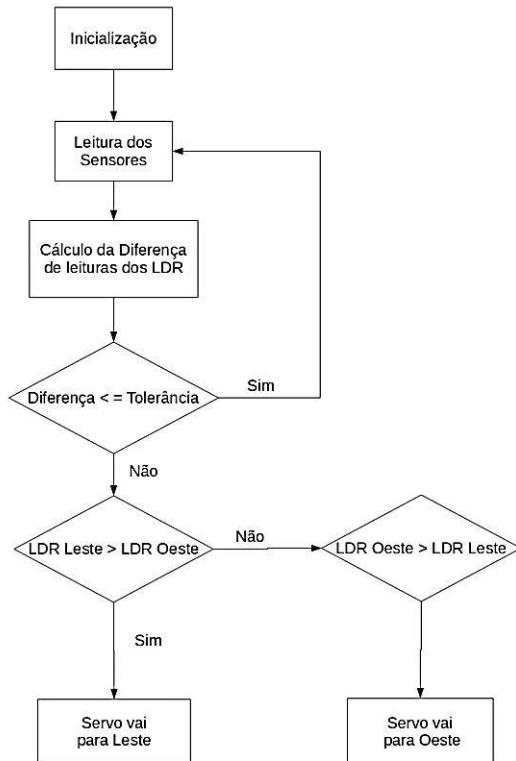
Embora possa ser feita a conversão de *bits* para Ohms (unidade de resistência), foi adotado neste trabalho os valores em *bits*, por resultar em uma faixa mais ampla de valores, facilitando desta forma, a escrita da programação. Os valores das resistências dos LDR serão posteriormente apresentados no supervisório.

A rotina de programação calcula a diferença entre as leituras dos LDR e se resultar em um valor menor ou igual a cinco *bits* (equivalente a uma tensão de 0,0244 V), o Rastreador

Solar não se move. Caso a diferença seja maior que a tolerância, será verificado qual LDR obteve a maior leitura. Se o LDR Leste obtiver valor maior que o Oeste, o servomotor se moverá na direção Leste (para 0° do servomotor). Se o Oeste obtiver o maior resultado, o sistema seguirá para Oeste (direção de 180°).

É importante salientar que o valor da tolerância de 5 *bits* do critério de parada descrito, foi utilizado levando-se em conta testes realizados ao longo do desenvolvimento do trabalho. Diferentes valores foram testados, sendo que aqueles maiores que 5 *bits* tornam a resposta do Rastreador Solar mais lenta em relação às mudanças de posição do Sol. Valores menores que 5 *bits*, fazem com que o sistema fique muito sensível às oscilações de luminosidade e consequentemente instável, se movimentando a cada pequena mudança detectada pelos LDR. O fluxograma da Figura 3.3 demonstra de maneira simplificada a lógica de funcionamento do Seguidor Solar controlado pelo Arduino.

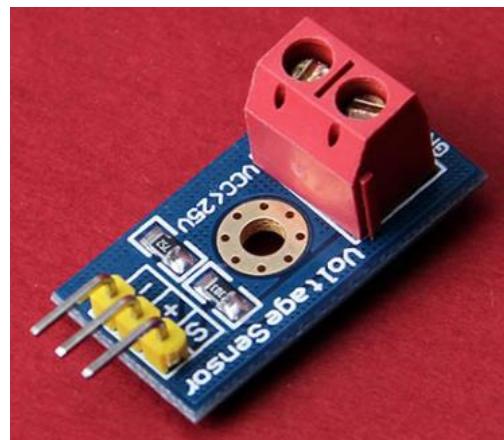
Figura 3.3: Fluxograma do Rastreador Solar.



Fonte: Próprio Autor.

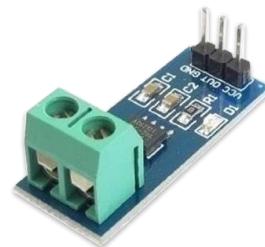
Ao se movimentar o servomotor, consequentemente tem-se o movimento das placas solares, cujos valores de tensão e corrente são medidos por sensores correspondentes a tais grandezas. Os sensores a serem usados para o monitoramento das placas são mostrados nas Figuras 3.4 (sensor de tensão) e 3.5 (sensor de corrente).

Figura 3.4: Sensor de Tensão.



Fonte: Próprio Autor.

Figura 3.5: Sensor de Corrente ACS712.



Fonte: Próprio Autor.

Com o intuito de se deixar o código de programação simplificado, aproveitou-se a capacidade do ScadaBR de converter dados na criação dos Data Points, podendo calcular facilmente a tensão e a corrente das placas fotovoltaicas.

O sensor de tensão é capaz de medir tensões contínuas até V_{CC} , sendo que este enviará no máximo 5 V_{CC} para a porta analógica do Arduino, cujo ADC converterá esta tensão

em *bits*. Para o cálculo da tensão deve-se converter o valor de *bits* para Volts novamente, e em seguida multiplicar esse resultado por 5, como mostrado na equação 1.

$$Tensão = LeituraADC * \frac{5}{1023} * 5 \quad (1)$$

Por fim, o cálculo da tensão das placas fotovoltaicas é demonstrada na equação 2.

$$Tensão = LeituraADC * 0,0244 \quad (2)$$

O sensor de corrente utilizado é o ACS712, cujo modelo pode realizar medições de -5 A até +5 A. A equação 3 apresenta o cálculo para obtenção da corrente em Ampères, devendo-se converter a leitura do ADC do Arduino em Volts e dividi-la por 0,185 mV/A, que é a sensibilidade do sensor de acordo com seu *datasheet* (ALLEGRO, 2018). Como será utilizada a faixa de 0 a 5 A por se tratar de uma corrente contínua a ser medida, deverá ser levada em conta apenas a faixa de 2,5 a 5 V_{CC} enviada à porta analógica do Arduino. Por isso, dever ser realizada uma operação de subtração na equação 3, de modo a restar somente à parte positiva da corrente.

$$Corrente = \left(LeituraADC * \frac{5}{1023} * \frac{1}{0,185} \right) - \frac{2,5}{0,185} \quad (3)$$

A equação 4 apresenta o cálculo da corrente.

$$Corrente = (LeituraADC * 0,0264) - 13,51 \quad (4)$$

3.2. DESENVOLVIMENTO DO RASTREADOR SOLAR

Para se desenvolver o protótipo do Rastreador Solar foi levado em conta o longo período de tempo exposto ao Sol. Dessa forma, se a estrutura a ser utilizada fosse de metal, esta aqueceria muito, podendo a temperatura elevada causar danos ao Arduino e aos demais componentes.

Sendo assim, deveria ser escolhido um material que não esquentasse em demasia e que fosse resistente ao mesmo tempo. Desta forma, eliminou-se o papelão e utilizou-se a Madeira, por aliar resistência e baixa condutividade de calor.

Para se evitar que o Arduino e a fiação ficasse exposta, foi projetado uma pequena caixa de madeira com 15,4 cm de comprimento, 19,2 cm de largura e 8 cm de altura, na qual o mesmo seria encaixado de modo a se ter expostas no lado externo as suas entradas USB e P4. O formato final da caixa com encaixe para o servomotor SG-5010 e para os fios provenientes dos sensores LDR e das placas ligadas em série pode ser vista nas Figuras 3.6 e 3.7.

Figura 3.6: Vista Lateral da Estrutura de Madeira.



Fonte: Próprio Autor.

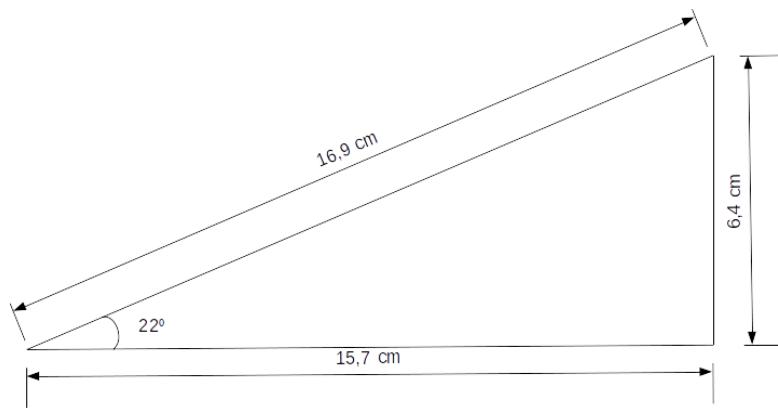
Figura 3.7: Vista da Caixa de Madeira com Servomotor Encaixado.



Fonte: Próprio Autor.

Considerando que o seguidor solar a ser desenvolvido é de eixo único, de modo que varia sua posição numa faixa de 0° a 180° , decidiu-se fixar as placas num ângulo adequado para a região de Dourados/MS. Desta forma, foi projetada uma estrutura na forma de um triângulo retângulo com uma inclinação de aproximadamente 22° , que é o mesmo valor de latitude da cidade. As medidas da estrutura podem ser vistas no desenho da Figura 3.8.

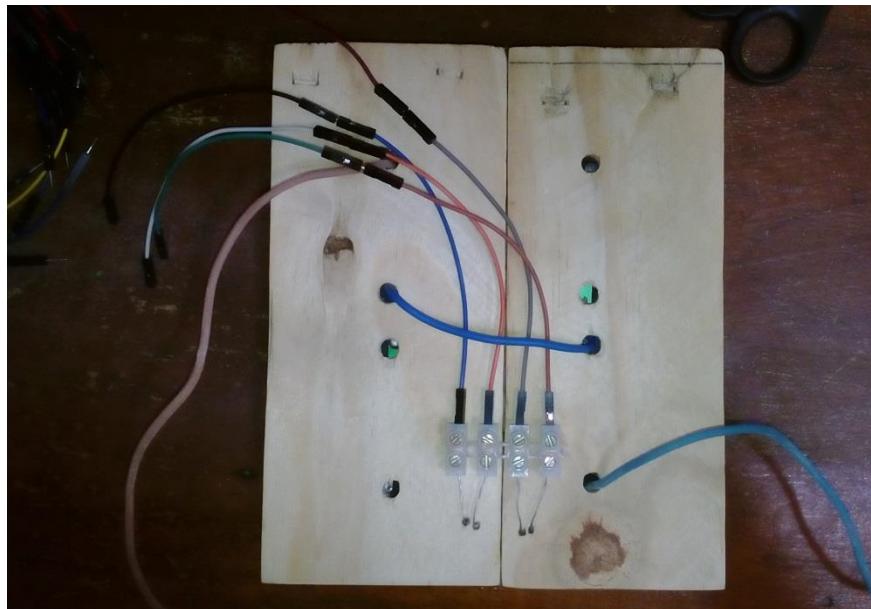
Figura 3.8: Medidas do Triângulo Retângulo.



Fonte: Próprio Autor.

Foram usadas emendas para facilitar a conexão dos LDR com os jumpers que iriam até a caixa na qual o servomotor e o Arduino estão localizados. Na Figura 3.9 pode-se ter uma visão do interior da estrutura de fixação das placas solares e dos LDR.

Figura 3.9: Vista do Interior do Triângulo Retângulo.



Fonte: Próprio Autor.

É importante lembrar que duas placas solares de 6 Vcc cada, foram conectadas em série, resultando em 12 Vcc, sendo que a corrente nominal se manteve em 200 mA. Medições realizadas com um multímetro registraram tensões acima de 12 Vcc, até mesmo ultrapassando 13 Vcc.

A estrutura de fixação foi devidamente colada à hélice do servomotor na posição de 90°, para uma maior facilidade no seu posicionamento. A montagem final da estrutura do seguidor solar pode ser conferida na Figura 3.10, podendo-se notar a presença de uma divisória entre os sensores LDR. Sua função é fazer com que cada LDR receba sua própria parcela de luz solar conforme o Sol se movimenta ao longo do dia, de modo a ocasionar as diferenças de leituras entre os LDR, para que a lógica possa ser implementada da maneira planejada e consequentemente que o movimento do Rastreador possa ocorrer.

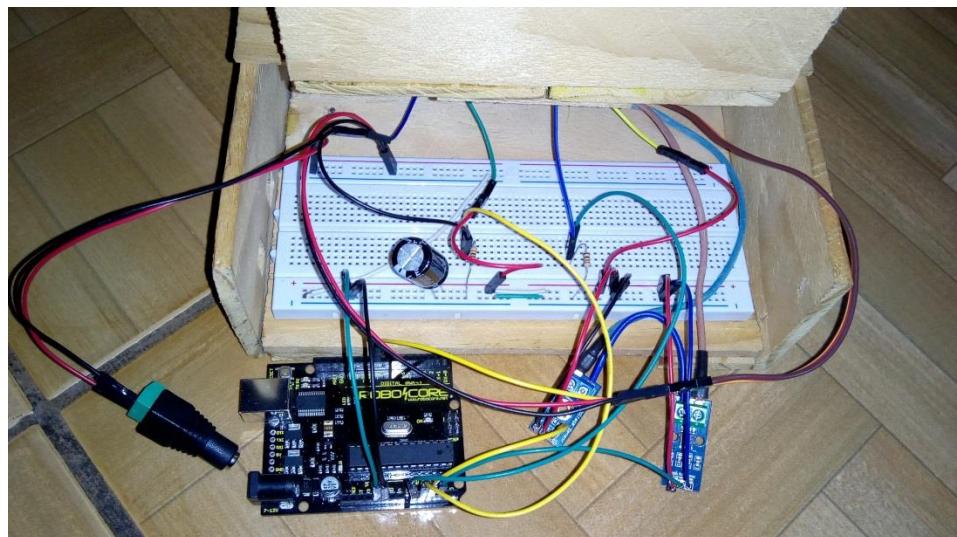
Figura 3.10: Montagem Final do Rastreador Solar.



Fonte: Próprio Autor.

A Figura 3.11, por sua vez, mostra a montagem real feita em protoboard.

Figura 3.11: Montagem Real em Protoboard.



Fonte: Próprio Autor.

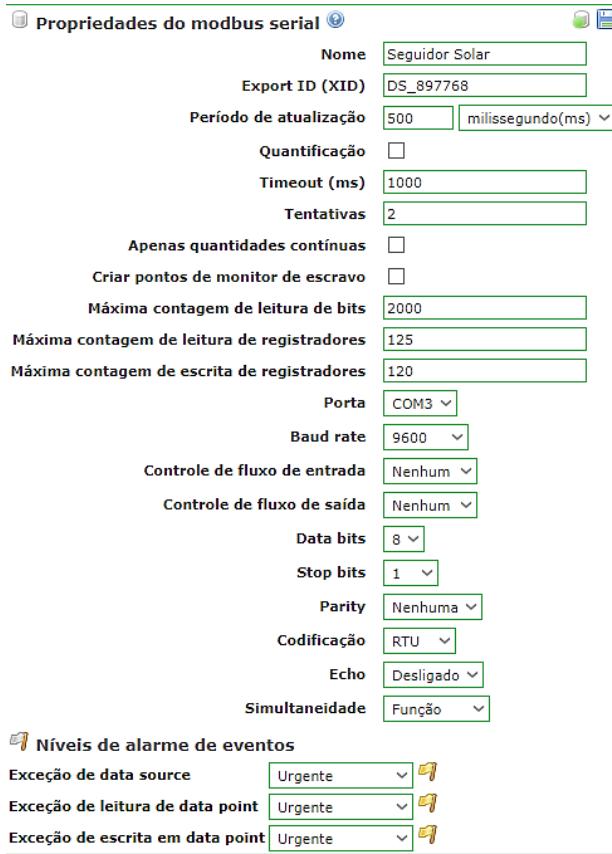
3.3. DESENVOLVIMENTO DO SUPERVISÓRIO

3.3.1. *Data Source*

Ao se trabalhar com o ScadaBR é necessário primeiramente criar um *Data Source*, que é responsável por adquirir os dados da rede de comunicação, que neste caso é a *Modbus Serial*. A transmissão dos dados é feita através de um cabo USB-AB que conecta o Arduino ao computador.

Neste trabalho o *Data Source* foi nomeado “Seguidor Solar” e configurado com uma taxa de atualização de 500 milissegundos, de modo a ser sincronizado com a interrupção da programação do Arduino. A Figura 3.12 mostra o resultado da configuração do *Data Source*, sendo importante salientar que a maioria das opções não foi alterada, como no caso dos *bits de parada* e *baud rate*.

Figura 3.12: Data Source Modbus Serial.



Fonte: Próprio Autor.

3.3.2. Data Points

Para a aquisição das medições dos sensores LDR, de corrente, de tensão e da posição do servo, é necessária a criação de *Data Points* inseridos no *Data Source* Seguidor Solar. Deste modo, foram criados 5 *Data Points*, sendo um para cada sensor (LDR Leste, LDR Oeste, sensor de tensão e sensor de corrente) e um para o servomotor.

Para que a aquisição de dados ocorra de forma adequada, deve-se ter em mente que os *Data Points* no ScadaBR possuem *offset* baseado em zero, ou seja, o primeiro *Data Point* é o número zero, o seguinte é o número um e assim sucessivamente. É importante frisar que a ordem dos *Data Points* configurados no ScadaBR deve ser a mesma daqueles listados na programação enviada ao Arduino.

No caso deste trabalho, o primeiro *Data Point* listado foi o pertinente ao LDR Leste, sendo que todos são do tipo numérico, como mostrado na Figura 3.13.

Figura 3.13: Lista de Data Points.

Data points						
Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)	
Corrente da placa	Numérico		1	Registrador holding	4	
LDR Leste	Numérico		1	Registrador holding	0	
LDR Oeste	Numérico		1	Registrador holding	1	
Servo	Numérico		1	Registrador holding	2	
Tensão da placa	Numérico		1	Registrador holding	3	

Fonte: Próprio Autor.

A faixa de registro e o tipo de dados de todos os *Data Points* criados foram configurados como Registrador *Holding* e Inteiro de 2 *bytes* sem sinal, respectivamente. O primeiro indica que serão recebidos dados que variam numa faixa hexadecimal de 0x40000 a 0xFFFFF e que poderá ser feita a leitura dos mesmos e também o envio de algum comando, se assim preferir o usuário. O segundo determina que os valores recebidos tenham um tamanho máximo de 2 *bytes*.

A maneira como os dados são mostrados no ScadaBR é definida na configuração do *Data Point*, podendo-se atribuir ou não valores aos campos Multiplicador e Aditivo de modo a se converter o valor “bruto” enviado ao *software*. Um exemplo é mostrado na Figura 3.14 para o caso do sensor de tensão, cuja leitura da porta analógica na qual está conectada varia de 0 a 1023 *bits* (0 a 5 V). Para se obter o valor em Volts, basta multiplicar a leitura por 0,02445 no próprio ScadaBR.

Figura 3.14: *Data Point* para Sensor de Tensão.



Fonte: Próprio Autor.

No caso do servomotor não foi necessário o uso de multiplicadores e aditivos. Dessa forma, o multiplicador teve valor um e o aditivo zero. O sensor de corrente por sua vez, teve multiplicador 0,0264 e aditivo de -13,51 para se obter a corrente em Amperes.

3.3.3. Scripts

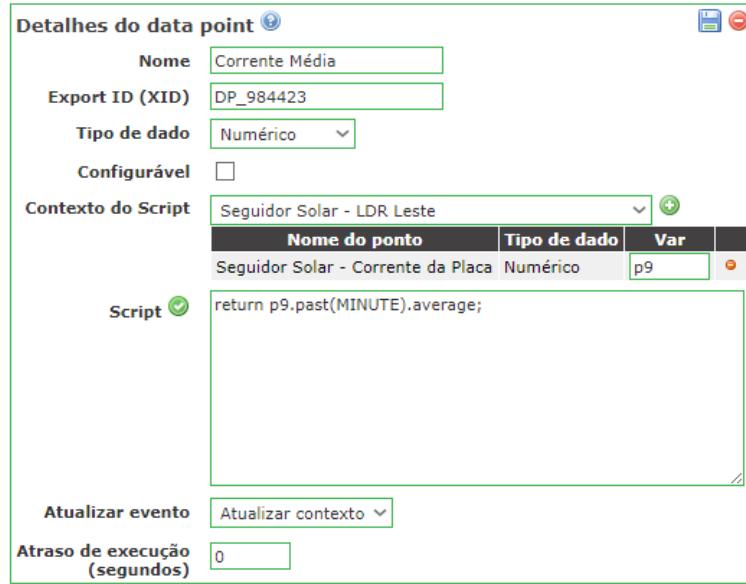
A criação de *scripts* do ScadaBR permite a construção de lógicas e a manipulação das variáveis medidas, podendo-se assim, obter outros valores sem a necessidade de calculá-los na programação do controlador. Para tal, é necessário que primeiramente se crie um novo *Data Source*, sendo este do tipo Meta. Para este trabalho, o *Data Source* Meta foi nomeado “Variáveis – Seguidor Solar”, não sendo necessária configuração alguma.

Feito isso, pode-se adicionar *Data Points* dentro do *Data Source* Meta. Neste trabalho foram criados cinco *Data Points* do tipo numérico: Corrente Média, Tensão Média, Potência, Resistência do LDR Leste e Resistência do LDR Oeste. Vale destacar que as médias calculadas pela programação em Java no ScadaBR são móveis, ou seja, soma-se os valores do período estipulado e a divide pelo número de pontos, indicando a tendência de um valor se repetir por determinado tempo.

Para obter a corrente média, por exemplo, foi inserido no campo Contexto de *Script* o *Data Point* “Corrente da Placa” criado anteriormente, que obteve o nome de variável “p9”.

Utilizou-se um *Script* na linguagem Java para obtenção da corrente média a cada minuto passado, como mostrado na Figura 3.15.

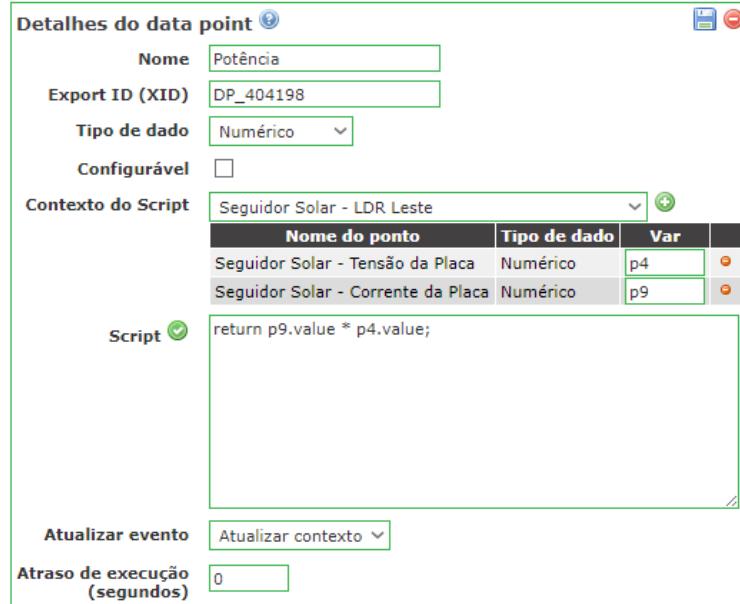
Figura 3.15: *Script* para Obtenção da Corrente Média.



Fonte: Próprio Autor.

Para a tensão média foi utilizada à mesma lógica no *Script* de corrente média, sendo apenas modificada a variável. No caso da potência foram utilizados os valores médios de tensão e corrente obtidos através de *Scripts*, por serem considerados mais precisos, de modo a diminuir a influência do ruído das portas analógicas do Arduino. A Figura 3.16 mostra o cálculo de potência através da criação de um *Script*.

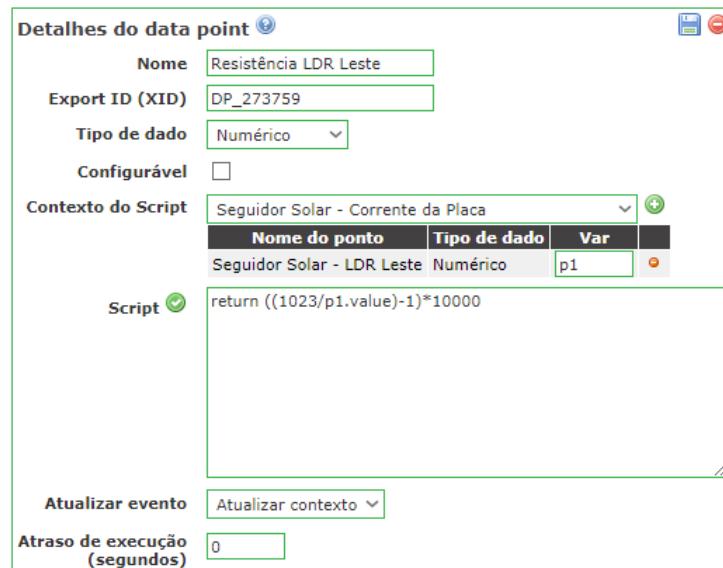
Figura 3.16: Script para Obtenção da Potência.



Fonte: Próprio Autor.

Por se fazer mais sentido, decidiu-se mostrar os valores medidos pelos LDR em Ohms, já que os mesmos são resistores, em vez de se mostrar apenas os *bits* resultantes da conversão do ADC do Arduino. O *Script* resultante da conversão pode ser visto na Figura 3.17 para o caso do LDR Leste, sendo que a mesma lógica se aplica ao LDR Oeste.

Figura 3.17: Script para Obtenção da Resistência em Ohms.



Fonte: Próprio Autor.

A lógica do *Script* para o cálculo da resistência dos LDR é obtido ao se isolar o valor da resistência 1 na fórmula de divisores de tensão (equação 5), sendo que a tensão de entrada são os 5 V_{CC} entregues pelo Arduino, a tensão de saída resulta da conversão da leitura analógica de *bits* em tensão e a resistência 2 possui valor de 10 kΩ.

$$R1 = 10000 * \left(\left(\frac{1023}{LeituraADC} \right) - 1 \right) \quad (5)$$

3.3.4. Visualização dos Dados

No ScadaBR as variáveis com que se está trabalhando podem ser analisadas de duas maneiras. A primeira é a *Watchlist*, que simplesmente apresenta os dados na forma de lista de maneira simples e direta, como mostra a Figura 3.18 para o caso deste trabalho.

Figura 3.18: *Watchlist* para Rastreador Solar.

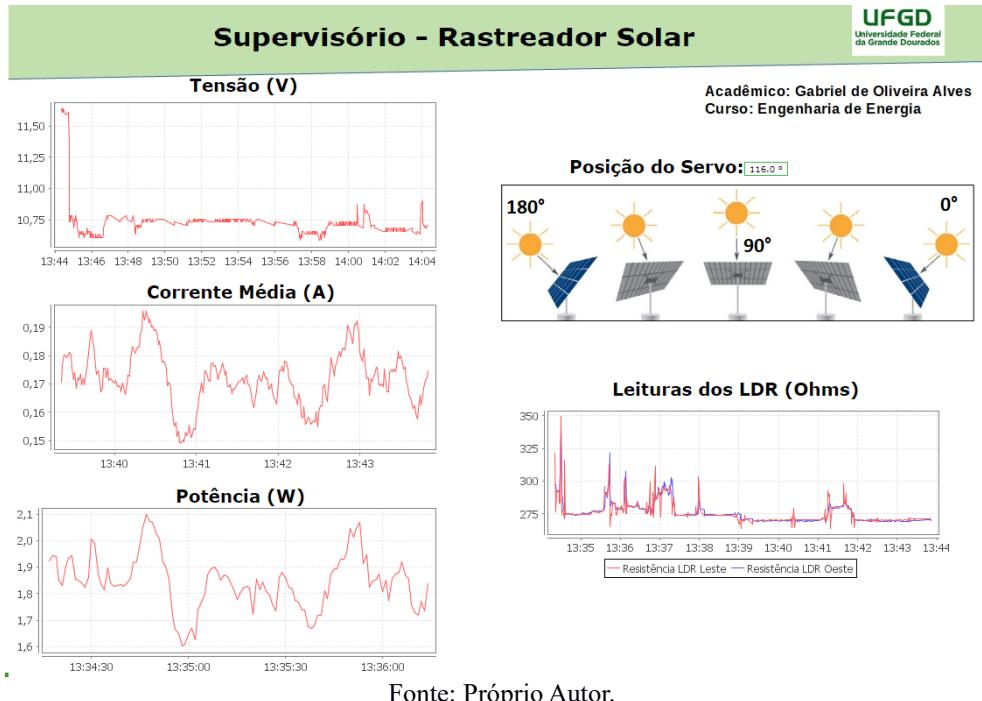
Watch list ⓘ				Seguidor Solar
Seguidor_Solar - LDR Leste	172.0	16:33:53	<input checked="" type="checkbox"/>	
Seguidor_Solar - LDR Oeste	182.0	16:33:53	<input checked="" type="checkbox"/>	
Seguidor_Solar - Servo	80.0°	16:33:53	<input checked="" type="checkbox"/>	
Seguidor_Solar - Corrente da placa	0,4028 A	10:49:21	<input checked="" type="checkbox"/>	
Seguidor_Solar - Tensão da placa	4,9389 V	16:33:53	<input checked="" type="checkbox"/>	
Variáveis - Seguidor Solar - Corrente Média	0,5266 A	10:49:21	<input checked="" type="checkbox"/>	
Variáveis - Seguidor Solar - Potência	3,1723 W	16:33:53	<input checked="" type="checkbox"/>	
Variáveis - Seguidor Solar - Tensão Média	6,0240 V	16:33:53	<input checked="" type="checkbox"/>	

Fonte: Próprio Autor.

Outra forma de visualização dos dados é pela representação gráfica, sendo esta uma maneira mais intuitiva e elaborada de demonstrar dados, sendo que gráficos e animações podem ser inseridos de modo a facilitar a sua leitura.

Para a representação gráfica de dados pertinentes ao Seguidor Solar (Figura 3.19) é interessante à apresentação de gráficos que são atualizados em tempo real com os valores dos sensores, além da posição do servomotor. É importante citar que para a visualização do valor em tempo real de uma variável gráfica, basta posicionar o cursor do *mouse* sobre o mesmo.

Figura 3.19: Representação Gráfica para Rastreador Solar.



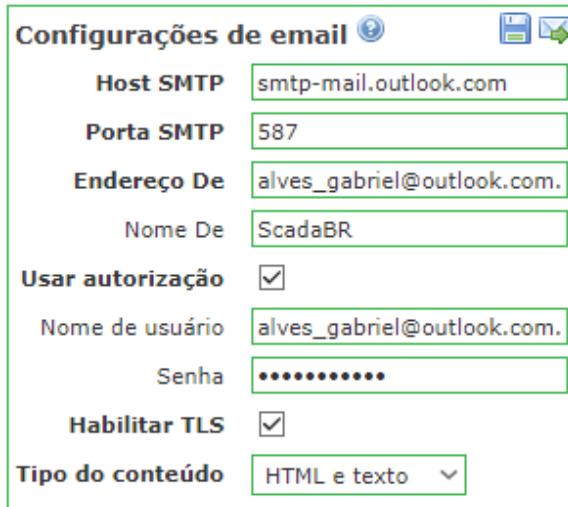
Fonte: Próprio Autor.

3.3.5. Geração de Relatórios

Um fator imprescindível num supervisório, por mais simples que este seja, é a capacidade de gerar relatórios para a visualização de resultados pertinentes a um determinado período, podendo ser do dia atual ou de algum momento passado. O ScadaBR permite inclusive o envio automático de relatórios para *e-mails*, incluindo gráficos e tabelas.

É importante salientar que para se enviar o relatório via *e-mail*, é necessária a configuração de *e-mail*, devem-se determinar os parâmetros vistos na Figura 3.20, sendo que o *e-mail* do próprio administrador do supervisório foi escolhido como “*host*”, ou seja, o responsável por enviar o *e-mail* aos destinatários.

Figura 3.20: Configuração do e-mail no ScadaBR.



Fonte: Próprio Autor.

A geração de relatórios no ScadaBR é criada a partir da inserção dos *Data Points* pertinentes e também dos destinatários, para o caso do relatório ser enviado por *e-mail*. Na Figura 3.21 pode-se ver que o único destinatário é o administrador do supervisório.

Figura 3.21: Configuração para Geração de Relatórios no ScadaBR.

Critério de relatório

Nome do relatório	Dados coletados pelo Seguidor Solar
Data points	LM35 - Média Móvel
Nome do data point	Seguidor_Solar - Tensão da placa Seguidor_Solar - Servo Variáveis - Seguidor Solar - Corrente Média Variáveis - Seguidor Solar - Tensão Média Variáveis - Seguidor Solar - Potência
Tipo de dados	Numérico Numérico Numérico Numérico Numérico
Cor	■ ■ ■ ■ ■
Gráfico consolidado	□ <input checked="" type="checkbox"/> □ <input checked="" type="checkbox"/> □ <input checked="" type="checkbox"/> □ <input checked="" type="checkbox"/> □ <input checked="" type="checkbox"/>

Eventos: Todos os eventos

Comentários de usuário:

Faixa de datas:

- Relativo ao horário do relatório
 - Anterior: 1 dia(s)
 - Passado: 1 dia(s)
- Datas específicas

De:	ano: 2018	mês: Abr	dia: 01	hora: 00	minuto: 00	<input type="checkbox"/> Início
Até:	ano: 2018	mês: Abr	dia: 02	hora: 00	minuto: 00	<input type="checkbox"/> Último

Agendar:

Relatório por email:

Incluir tabela de dados:

Dados em formato .zip:

Destinatários de email:

- Adicionar lista de envio:
- Adicionar usuário:
- Adicionar endereço:

admin

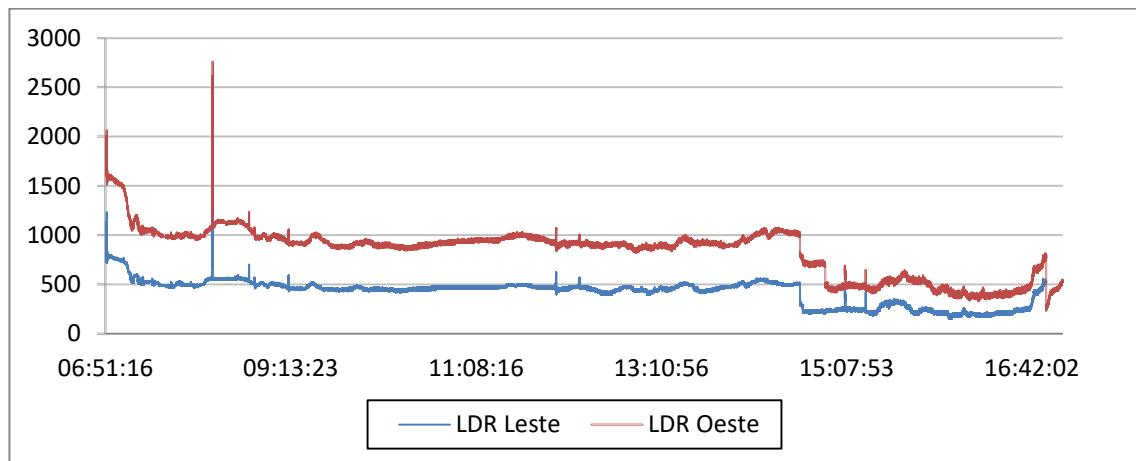
Fonte: Próprio Autor.

4. RESULTADOS E DISCUSSÕES

As medições foram realizadas durante cinco dias ao longo do mês de maio de 2018, o tempo de duração nas medições foi influenciado devido à estação de outono, pois os dias estavam mais curtos. Os resultados obtidos nas medições foram acompanhados em tempo real e no final de cada dia de medição gerou-se um relatório contendo os gráficos pertinentes aos *Data Points* e as planilhas dos dados enviados do Arduino ao ScadaBR.

O primeiro dia de experimento foi em 05 de maio de 2018, com início às 06h52min. Neste dia, o Sol estava encoberto por nuvens finas, o que permitiu a incidência de certa quantidade de raios solares à superfície das placas solares. Os valores das resistências dos LDR (*Light Dependent Resistor*) estão acima de $500\ \Omega$ em especial no início e no final do dia, já que há uma menor luminosidade, sendo que o período de maior incidência de luz solar ocorre por volta de 14h45min até 16h25min, aproximadamente, com uma faixa de valores de $250\ \Omega$ a $500\ \Omega$. De acordo com o relatório, as resistências médias dos LDR Leste e Oeste foram de $422,5215\ \Omega$ e $425,0888\ \Omega$, respectivamente. A Figura 4.1 apresenta as curva das resistências para este o 1º dia de medições.

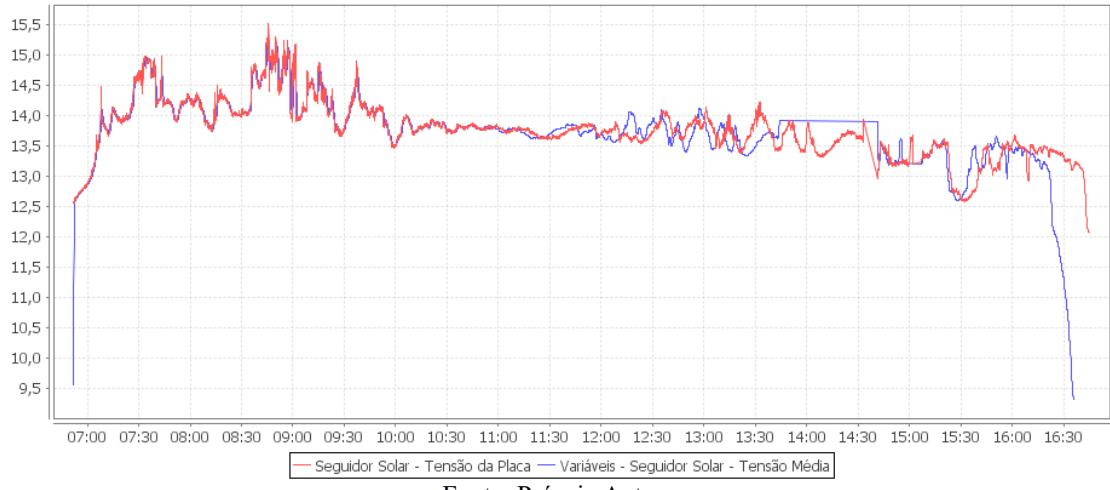
Figura 4.1: Resistências dos LDR no 1º Dia.



Fonte: Próprio Autor.

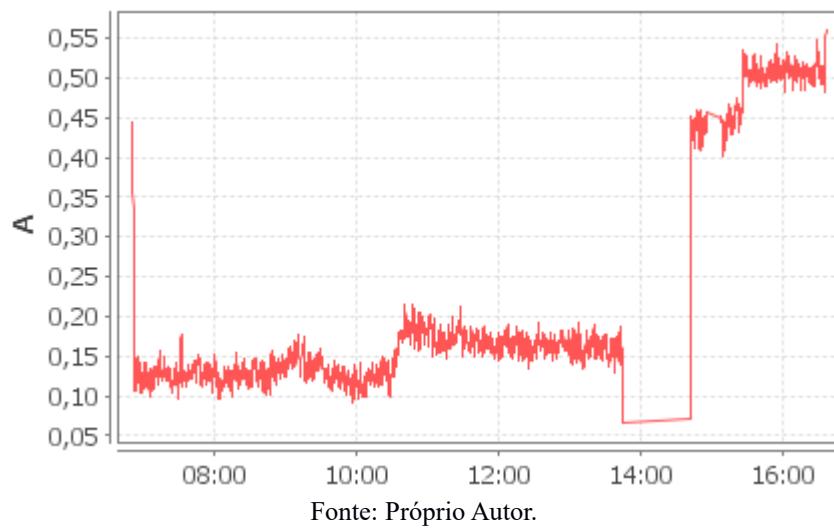
Por mais que o tempo estivesse predominantemente nublado, a tensão se manteve elevada, com média de $13,7118\ V_{CC}$. As curvas de tensão real (em vermelho) e média (em azul) podem ser observadas na Figura 4.2.

Figura 4.2: Tensões Obtidas no 1º Dia de Medições.



A corrente foi a variável mais afetada durante as medições, o que está relacionado à suscetibilidade do sensor ACS712 aos ruídos, podendo prejudicar os valores finais. Levando-se em conta que a corrente nominal das placas conectadas em série é de 200 mA, a média de 213,2 mA, por estar acima de 200 mA, confirma esta suscetibilidade. A potência cujo valor nominal é de 1 W, chegou a um máximo de 13,0371 W, sendo afetada pela corrente. A Figura 4.3 mostra o comportamento da corrente média em Ampères neste dia.

Figura 4.3: Corrente Média no 1º Dia de Medições.



É importante ressaltar que preferiu-se deixar de lado a medição da corrente real por conta da ruidosidade do sensor ACS712, sendo que o próprio fabricante recomenda em seu

datasheet o cálculo de corrente média, de modo a se obter um resultado mais estável e preciso.

Em relação ao servomotor, este moveu-se como esperado de leste para oeste, embora com algumas oscilações nos períodos em que a luminosidade do Sol esteve reduzida por conta das nuvens, que por sua vez aumentam os efeitos da luminosidade difusa. Pelo gráfico da Figura 4.4, pode-se verificar que conforme a programação, o motor inciou seus trabalhos na posição central de 90° , movendo-se em seguida para leste (posição igual 0°).

Figura 4.4: Movimento do Servomotor no 1º Dia.

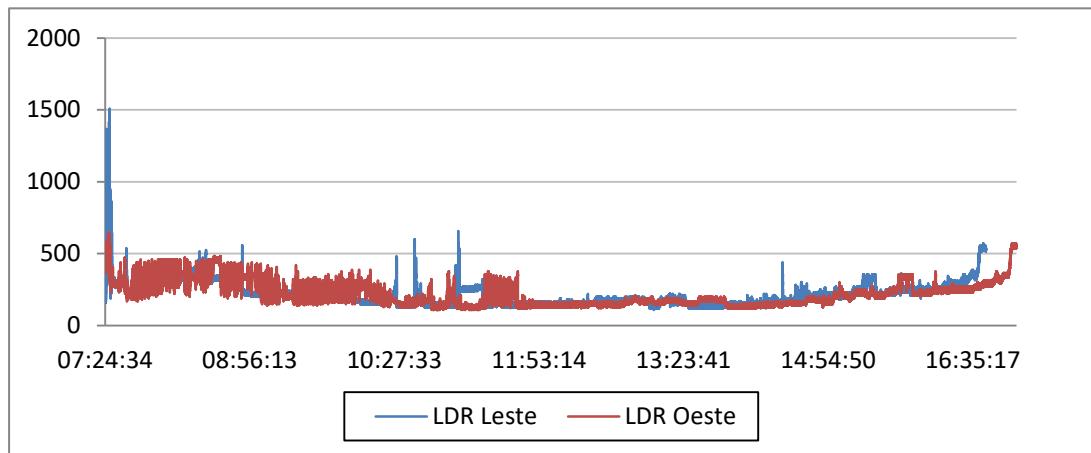


Fonte: Próprio Autor.

O Rastreador Solar, apesar das oscilações das leituras dos LDR devido ao Sol estar encoberto pelas nuvens conseguiu completar o movimento de leste a oeste, sendo que por volta das 12h00min esteve próximo de 60° , atingindo a posição central (cerca de 90°) por volta das 12h08min. Com o reinício do sistema às 14h41min, o sistema voltou a 90° , para em seguida, continuar seguindo para oeste até atingir a posição máxima de 160° , antes de começar a recuar com a redução da luminosidade, ocasionado pelo início do pôr do sol.

O segundo dia de medições foi realizado em 09 de maio de 2018, que por sua vez foi um dia com poucas nuvens e predominantemente ensolarado. A Figura 4.5 mostra as leituras das resistências dos LDR Leste (curva azul) e Oeste (curva vermelha), que obtiveram médias de 218,8899 e 221,6709 Ω respectivamente, sendo valores menores que os do primeiro dia, o que é de se esperar, já que a resistência de um LDR é inversamente proporcional à luminosidade a que está exposto.

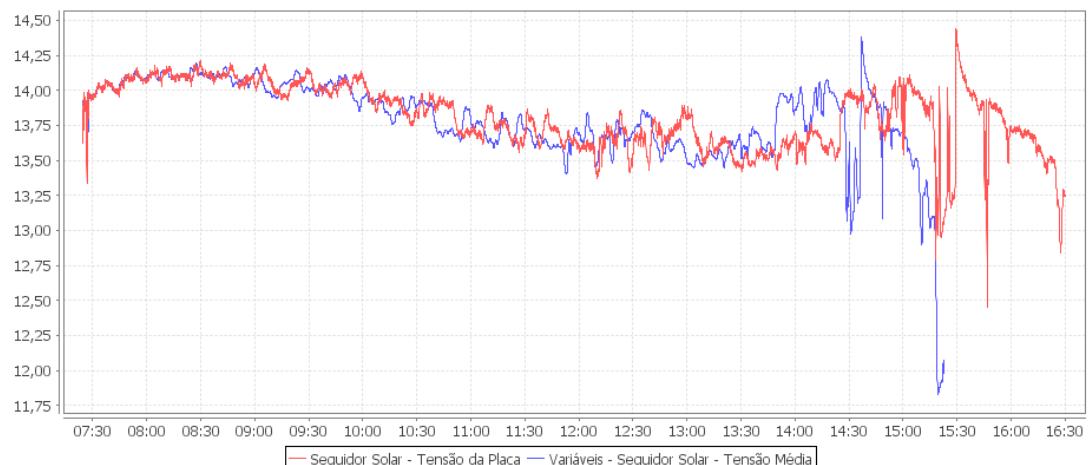
Figura 4.5: Resistências dos LDR no 2º Dia.



Fonte: Próprio Autor.

A tensão média foi de $13,8164 \text{ V}_{\text{CC}}$, um pouco maior que há do primeiro dia. Pela Figura 4.6 é possível verificar que o comportamento da tensão variou pouco ao longo do dia, tendo uma queda significativa apenas ao final do dia. Pode-se verificar que houve um atraso em relação ao cálculo da tensão média (curva azul) a partir da tensão em tempo real (curva vermelha) das placas pelo ScadaBR. Este fato, provavelmente se deve a grande quantidade de dados a serem processados ao mesmo tempo pelo *software*, o que provoca uma demora na execução dos cálculos dos *Scripts*. Com isso, na sequência dos resultados serão mostrados apenas os valores reais de tensão, sendo que não há grande diferença entre ambos.

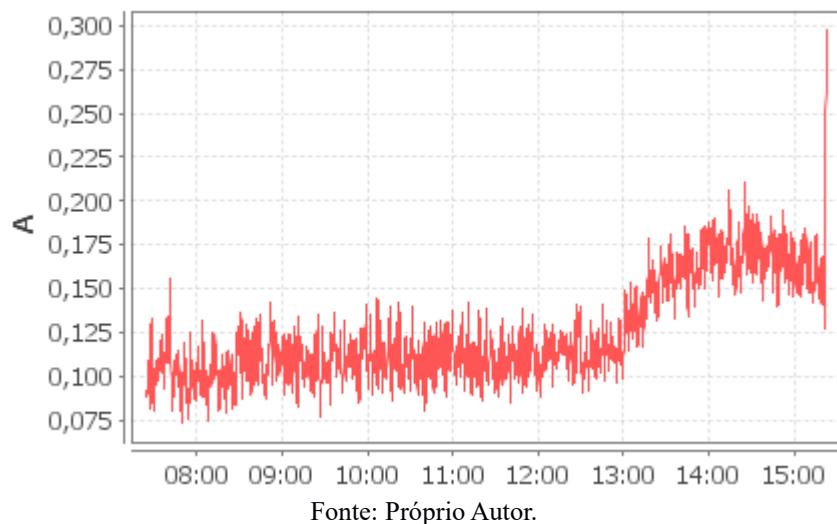
Figura 4.6: Tensões Obtidas no 2º Dia.



Fonte: Próprio Autor.

A corrente média para este dia foi de 145,6 mA, que pode ser considerado mais próxima do esperado em relação ao primeiro dia, já que não ultrapassa os 200 mA nominais das placas, embora mesmo utilizando um valor médio de corrente e o dia tendo poucas oscilações de luminosidade, a curva da Figura 4.7 mostra variações bruscas.

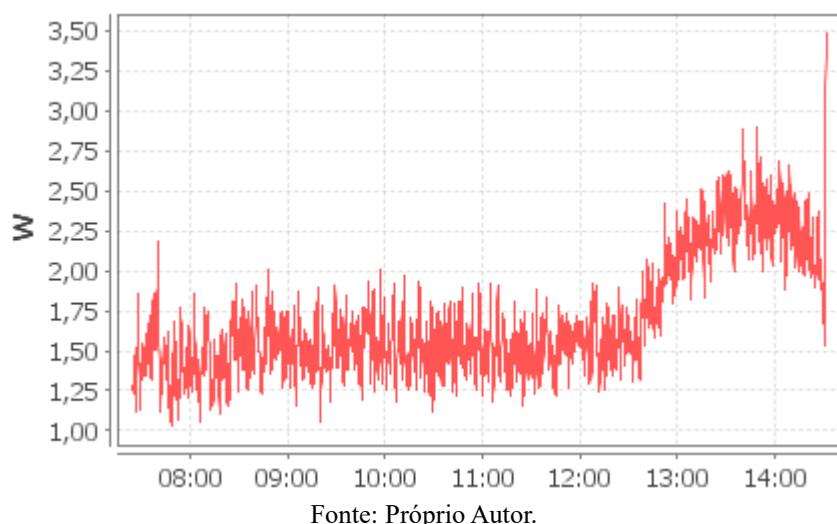
Figura 4.7: Corrente Média Obtida no 2º Dia.



Fonte: Próprio Autor.

O valores de potência ainda não são os ideais, por conta da própria instabilidade na leitura da corrente, sendo que o seu valor médio de 2,0939 W. A curva de potência é mostrada na Figura 4.8.

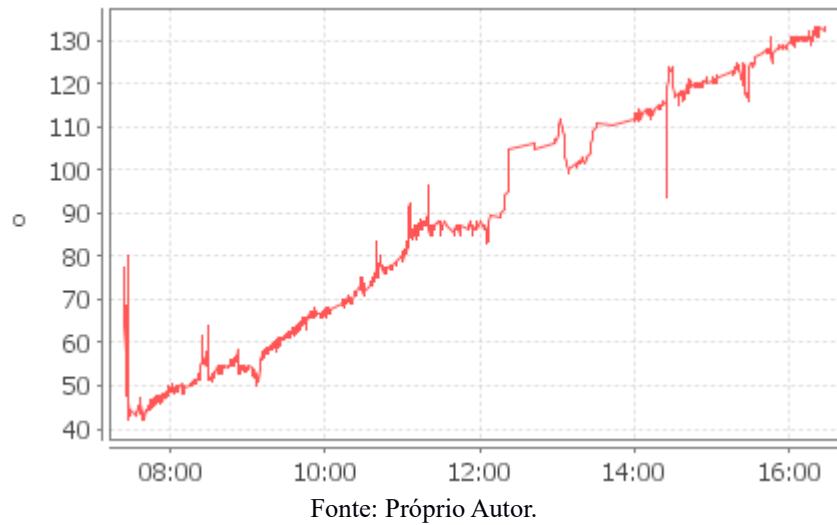
Figura 4.8: Curva de Potência no 2º Dia.



Fonte: Próprio Autor.

O servomotor mais uma vez executou o movimento de leste a oeste, embora como pode ser observado na Figura 4.9, exista uma dificuldade em manter a sua posição estável. Isso ocorre por conta da instabilidade dos LDR, que demoram a se adaptar a variações de luminosidade, sendo semelhantes aos olhos humanos neste aspecto.

Figura 4.9: Movimento do Servomotor no 2º Dia.

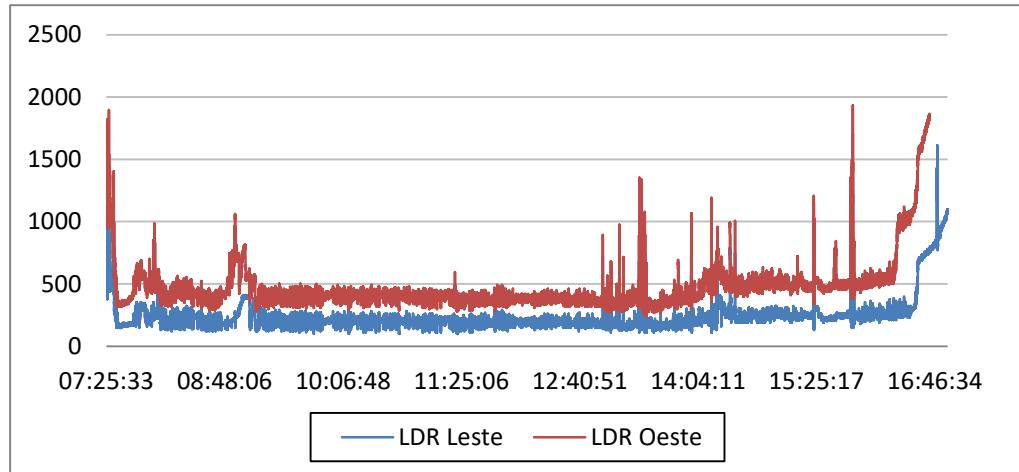


Fonte: Próprio Autor.

Observando-se a curva das resistências mostrada na Figura 4.5, pode-se perceber esta instabilidade dos LDR em especial na curva vermelha, que representa o LDR Oeste, que ao se manter distante da leitura do LDR Leste faz com que o motor fique variando em poucos graus, demorando a parar em uma determinada posição.

O terceiro dia de medições foi executado em 23/05/2018, que foi predominantemente ensolarado, havendo presenças de nuvens no início do dia. A Figura 4.10 mostra o comportamento das resistências dos LDR neste dia, que inicialmente tinham valores altos e diminuíram ao longo do dia.

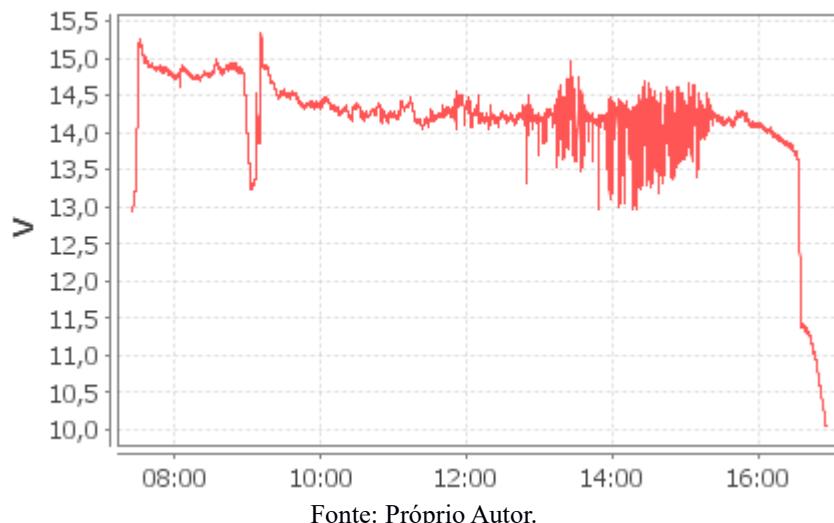
Figura 4.10: Resistências dos LDR no 3º Dia.



Fonte: Próprio Autor.

É possível afirmar que as curvas das resistências estão corretas ao compará-las com a curva de tensão, por exemplo, mostrada na Figura 4.11, em que ao final do dia, com a diminuição da luminosidade há uma queda na tensão e uma elevação dos valores das resistências.

Figura 4.11: Tensão Real Obtida no 3º Dia.



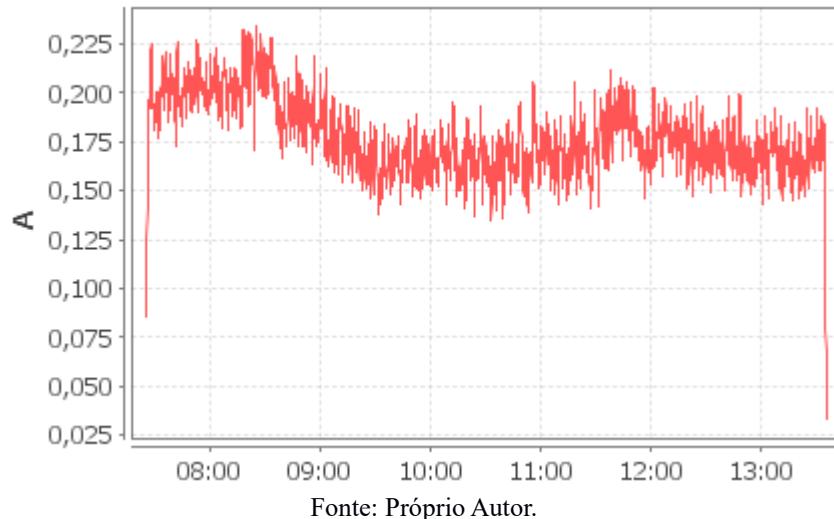
Fonte: Próprio Autor.

Na Figura 4.11 pode-se perceber que o valor de tensão de 15 Vcc registrado em alguns momentos da manhã, se mostra incoerente, já que as duas placas conectadas em série

possuem 6 V_{CC} nominais cada, totalizando 12 V_{CC}. Por mais que testes com multímetro sob Sol forte, foram registrados picos acima de 13 V_{CC}, o valor de 15 Vcc está muito acima do esperado.

A medição de corrente continuou mostrando-se pouco precisa. Por mais que houvesse coerência nos valores nos momentos em que a tensão aumentava e consequentemente a corrente caía para se manter a potência das placas, os mesmos se mostravam imprecisos, por conta do valor de corrente nominal das placas, que é de 200 mA. A Figura 4.12 mostra a curva de corrente para o dia 23 de maio de 2018.

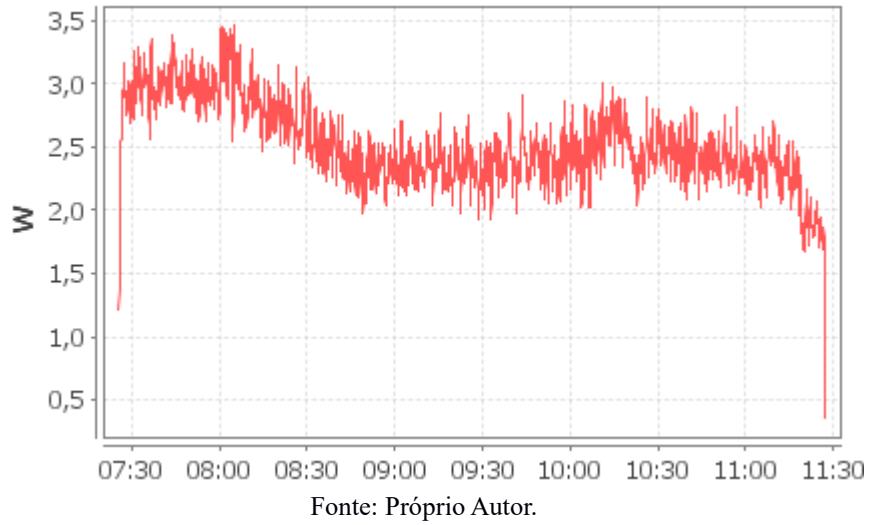
Figura 4.12: Corrente Média Obtida no 3º Dia.



Fonte: Próprio Autor.

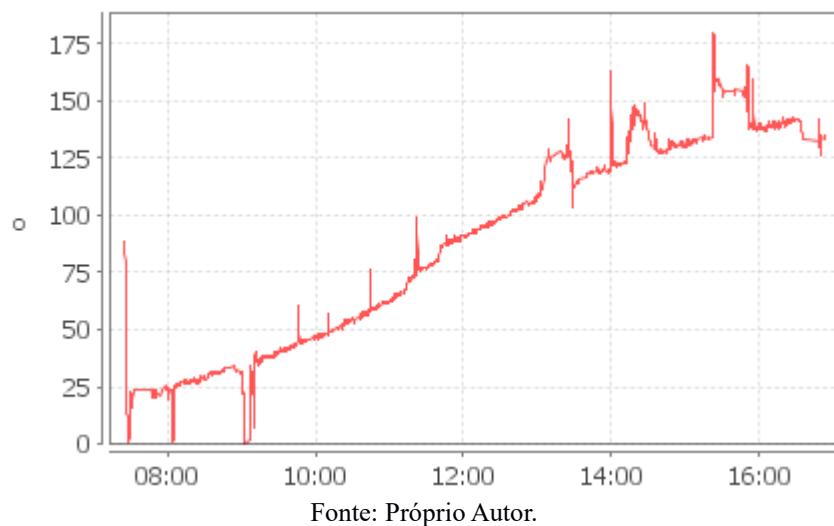
A potência, consequentemente, obteve uma curva pouco precisa, como pode ser visto na Figura 4.13.

Figura 4.13: Curva de Potência no 3º Dia.



O movimento leste-oeste do Rastreador Solar ocorreu como esperado, ao iniciar em 90° , seguir para leste em 0° , e na sequência seguindo o Sol para oeste. O aparato inclusive atingiu a posição de 180° , retornando para leste conforme o Sol se punha. A Figura 4.14 demonstra o movimento do Seguidor Solar ao longo do dia, podendo-se perceber que por conta da instabilidade dos LDR, em raros momentos este mantém uma posição fixa, oscilando poucos graus, o que pela observação dos dados, não influiu significativamente na tensão, que se manteve praticamente estável ao longo do dia, fora os momentos de pico.

Figura 4.14: Movimento do Servomotor no 3º Dia.

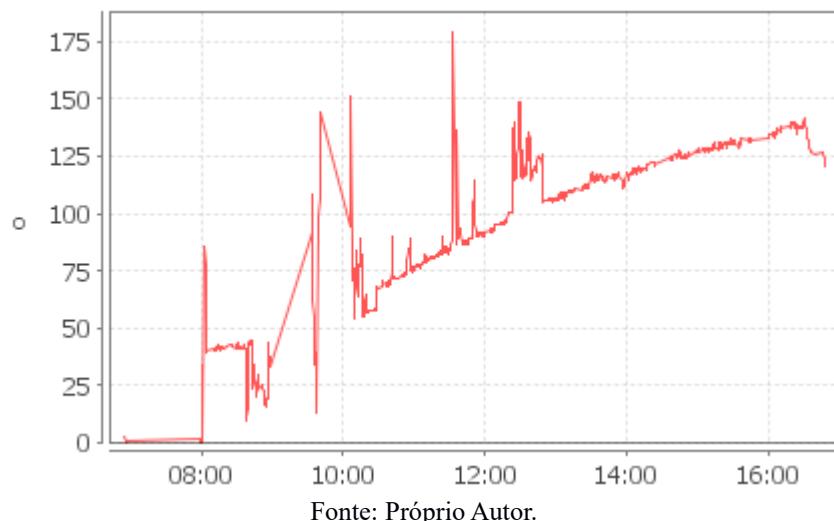


O quarto dia de medições ocorreu em 28 de maio de 2018, sendo um dia predominantemente ensolarado, embora com presença de várias nuvens no céu. Inicialmente o Seguidor Solar seguiu para leste, com o seu extremo em 0° , mas não se movimentou até às 08h02min, por conta da instabilidade dos LDR, em especial o oeste, cuja leitura não variava o suficiente, de modo a fazer o aparato seguir o Sol como esperado. A luz refletida pelo telhado certamente foi a principal causa para que tal fato ocorresse.

Como até o momento não havia problemas com a conexão dos LDR, o sistema foi reiniciado, o que não resultou em melhorias nas leituras dos LDR. Às 08h58min o sistema foi desligado para manutenção, ao se constatar que um fio havia se partido. Com isso, aproveitou-se a oportunidade para realizar mudanças na montagem do Seguidor Solar, ao se utilizar uma fonte chaveada de 5 V_{CC} exclusivamente para alimentação do servomotor e um capacitor eletrolítico de $3300\text{ }\mu\text{F}$ em paralelo com a alimentação dos sensores pelo Arduino.

O retorno da manutenção ocorreu por volta de 10h07min, com o servomotor inicialmente se posicionando adequadamente em torno de 70° , mas em seguida variando novamente. Apesar disso, no decorrer do dia o Seguidor Solar se movimentou para oeste, como pode ser visto pelo gráfico da Figura 4.15.

Figura 4.15: Movimento do Servomotor no 4º Dia.



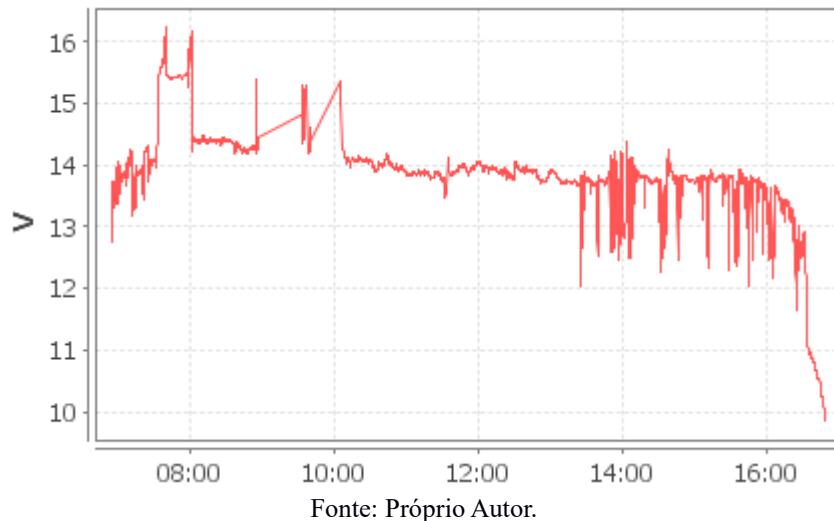
Fonte: Próprio Autor.

Com o uso de uma fonte somente para o servomotor, a tendência é de que as medições dos sensores de tensão e corrente se tornassem mais próximas dos valores nominais, pois a lógica para o cálculo de tais variáveis leva em conta uma tensão de 5 V_{CC} proporcionada pelo

Arduino, o que não ocorre na realidade, fazendo com que quando há um atuador que consuma muita energia, como um servomotor, as medições fiquem prejudicadas. O capacitor eletrolítico de $3300 \mu\text{F}$ também colaborou para a estabilização da tensão enviada do Arduino para os sensores.

Pela Figura 4.16 pode-se ver que há uma diminuição na tensão para cerca de 14 V_{CC} após o retorno da manutenção, sendo mais apropriado, já que em medições com multímetro, foram obtidos valores acima de 13 V_{CC} sob Sol forte. A média da tensão neste dia foi de $13,8771 \text{ V}_{\text{CC}}$.

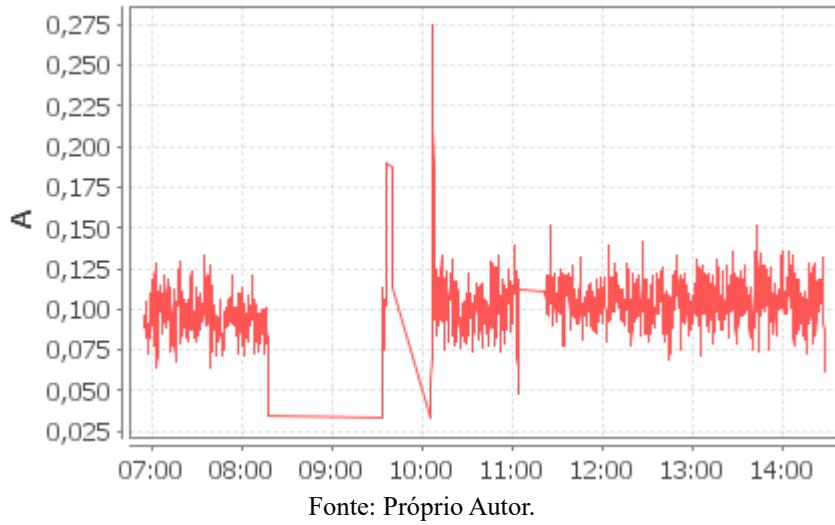
Figura 4.16: Tensão Real Obtida no 4º Dia.



Fonte: Próprio Autor.

Houve melhoria evidente na medição de corrente, que obteve média de $84,1 \text{ mA}$, sendo que o seu comportamento pode ser analisado na Figura 4.17.

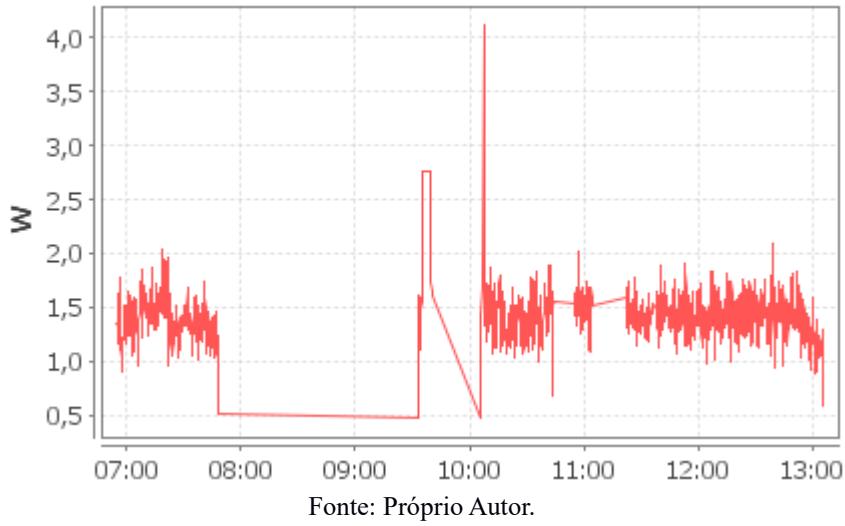
Figura 4.17: Medição de Corrente Média no 4º Dia.



Fonte: Próprio Autor.

Consequentemente, houve influência no cálculo de potência. Pela Figura 4.18 pode-se ver que houve uma leve melhoria, sendo que segundo o relatório gerado pelo ScadaBR, a potência média foi de aproximadamente 0,9610 W.

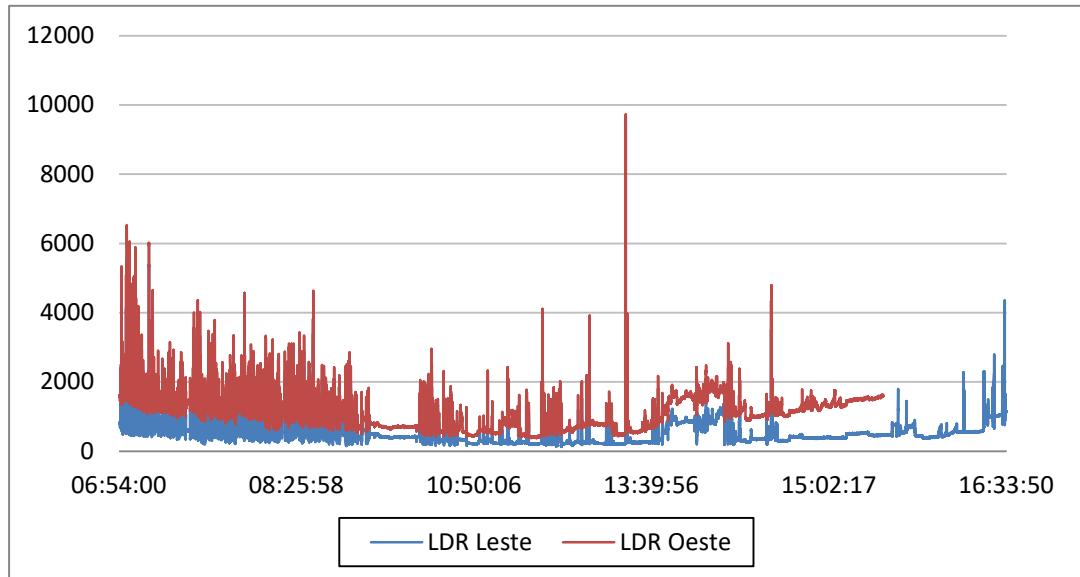
Figura 4.18: Curva de Potência no 4º Dia.



Fonte: Próprio Autor.

Para o quinto dia de medições, realizado em 31 de maio de 2018, foi mantida a montagem com uma fonte de alimentação para o servomotor e um capacitor em paralelo com a alimentação dos sensores pelo Arduino. Este foi um dia com muitas nuvens e oscilações na luminosidade, o que afetou a leitura dos LDR. As curvas das resistências dos LDR leste (em vermelho) e oeste (em azul) são mostrados na Figura 4.19.

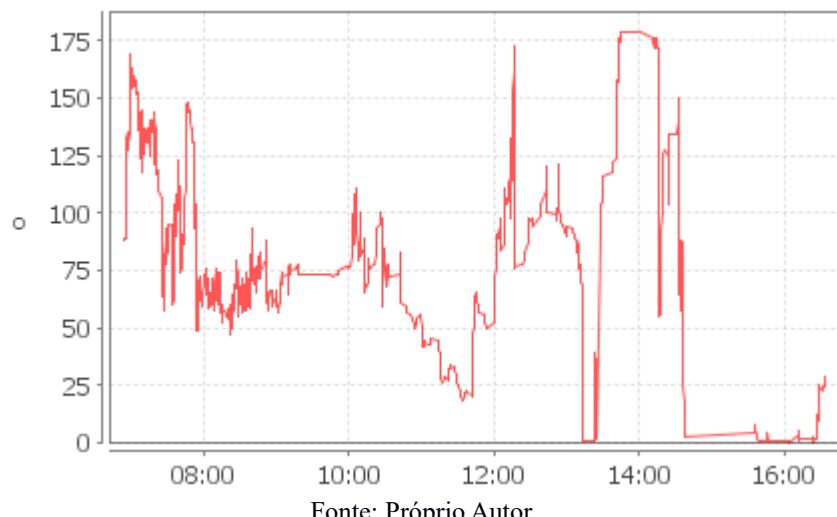
Figura 4.19: Resistências dos LDR no 5º Dia.



Fonte: Próprio Autor.

A instabilidade dos LDR refletiu diretamente no movimento leste-oeste do Seguidor Solar neste dia, que não seguiu adequadamente o padrão esperado, chegando a atingir 180° às 13h45min em vez de fazê-lo ao final do dia, próximo às 17h00min, como é mostrado na Figura 4.20.

Figura 4.20: Movimento do Servomotor no 5º Dia.



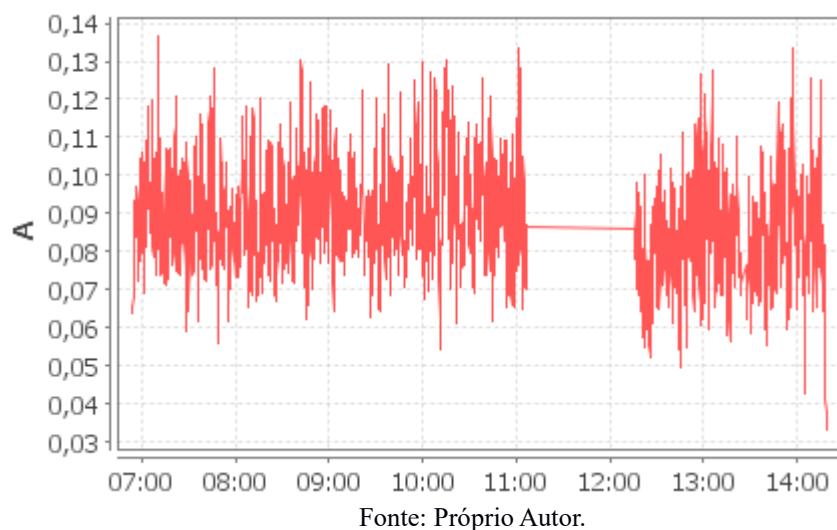
Fonte: Próprio Autor.

Apesar do comportamento inconstante e do céu nublado, a tensão obteve uma média de 14,1136 V_{CC}. Mesmo que o valor real possa ser um pouco menor, é de se esperar que uma placa fotovoltaica tenha uma boa geração mesmo com o dia nublado, pois o silício, material que compõe as células fotovoltaicas é um semicondutor, que por sua natureza, perde eficiência ao ser exposto a uma temperatura elevada.

Portanto, pelo fato de o dia estar nublado, a temperatura tende a ser mais baixa. Com isso, é possível que mesmo com a luminosidade reduzida, as placas continuem a ter uma boa geração, conforme explicado na seção 2.4.

A curva de corrente média (Figura 4.21), por mais que apresente grandes oscilações, neste dia apresentou valores condizentes com o esperado, sendo que em nenhum momento ultrapassou o valor nominal de 200 mA, obtendo uma média de 75,3 mA. Com isso, mostra-se que o fato de utilizar uma fonte própria para o servomotor colabora para a obtenção de resultados mais próximos do esperado nas medições, pois o Arduino de certa forma economiza energia, ao alimentar somente os sensores.

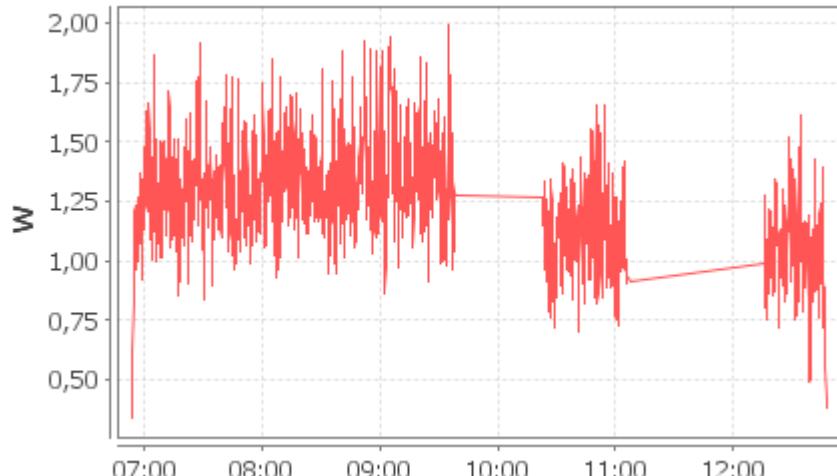
Figura 4.21: Corrente Média Obtida no 5º Dia.



Fonte: Próprio Autor.

Assim como a corrente, a curva de potência mostrada na Figura 4.22 tem um comportamento instável, mas apresenta valores mais próximos do esperado, embora ainda, em vários momentos, ultrapasse 1 W. No entanto, a média de acordo com o relatório foi de 0,88 W.

Figura 4.22: Curva de Potência no 5º Dia.



Fonte: Próprio Autor.

Com os cinco dias de medições completa observou-se comportamentos semelhantes do Seguidor Solar nestes, como a instabilidade dos LDR que de certa forma, se assemelham aos olhos humanos, pois demoram a se adaptar a mudanças de luminosidade, sendo que a luminosidade difusa interfere no comportamento dos mesmos.

A partir da análise das resistências pode-se perceber que valores abaixo de $500\ \Omega$ estão associados há um dia ensolarado, ou ao fato de o Sol não estar encoberto naquele momento. Valores superiores a $500\ \Omega$, podem indicar a presença de nuvens, sendo que geralmente encontram-se valores superiores a $1000\ \Omega$ no início do dia (quando o Sol ainda está surgindo no horizonte) e ao final (quando o Sol está se pondo, o que durante o período das medições começava a ocorrer antes mesmo das 17h00min).

É possível dizer que o efeito da luminosidade difusa pode ter sido agravado por conta da própria estrutura do Seguidor Solar que é baixa, permitindo que a luz refletida pelo telhado atingisse com maior intensidade os LDR, o que ocasionou por diversas vezes o retorno do servomotor para leste, em vez de continuar seguindo o Sol.

Melhorias na programação, que fariam o Seguidor Solar manter a posição durante momentos em que o Sol é encoberto por nuvens e a construção de uma estrutura mais alta com divisórias que envolvessem os LDR, são opções para que se possa reduzir a instabilidade das leituras dos mesmos.

Por mais que houvesse imprecisões no Rastreador Solar, este executou o movimento leste-oeste em quatro dos cinco dias de medições da maneira esperada, sendo que no quinto dia, o tempo estava predominantemente nublado, o que somado à luminosidade difusa, deixou o sistema instável.

A supervisão dos dados em tempo real a partir do ScadaBR possibilitou correções, que mesmo tardias, mostraram efeito nos cálculos de tensão, corrente e potência. Mesmo com os atrasos nos cálculos dos *Data Points* Meta, este *software* demonstrou-se preciso nos seus resultados, sendo que os valores imprecisos obtidos inicialmente, ocorreram por imperícia do próprio autor.

O fato do ScadaBR permitir a criação de *Scripts* para os cálculos de tensão, corrente, potência e resistências, mostrou-se um grande facilitador, pois o código enviado ao Arduino ficou mais enxuto e fácil de ser escrito, o que diminui consideravelmente o número de variáveis armazenadas na memória do microcontrolador e consequentemente a quantidade de informações a serem processadas.

5. CONCLUSÃO

Este trabalho demonstrou a criação de um protótipo de Rastreador Solar com Eixo Único, controlado por Arduino e com supervisão de dados via ScadaBR.

O sistema mostrou-se capaz de executar a função proposta apesar das imprecisões detectadas, sendo que algumas melhorias foram feitas durante a etapa de medições, possibilitando uma visão do que um engenheiro deve fazer ao se deparar com os mais diversos tipos de problemas.

A plataforma Arduino provou ser um sistema robusto e de fácil uso, executando o controle do Rastreador, assim como o envio das medições dos sensores para o supervisório. Um fato a ser destacado, é que o uso do Arduino no desenvolvimento do protótipo, permitiu que o mesmo tivesse um custo final de produção baixo, pois todos os materiais utilizados são em sua maioria baratos e possuem ampla disponibilidade, seja em lojas físicas ou na *internet*.

O supervisório desenvolvido em ScadaBR mostrou-se versátil, sendo que as suas inúmeras funcionalidades permitem ao usuário a manipulação de uma grande variedade de dados da maneira que for desejada. Ainda, a possibilidade de se manipular os valores medidos através de *scripts* permitiu que a programação do Arduino ficasse menor e mais fácil de ser escrita, podendo-se inclusive calcular valores médios e se obter novas variáveis.

Certamente, num ambiente industrial, por exemplo, no qual é necessário o monitoramento em tempo real de inúmeras variáveis de vários processos que ocorrem ao mesmo tempo, é de grande importância o uso de um *software* como o ScadaBR, em especial por sua gratuidade, pois possibilita uma redução de custos, ao mesmo tempo que executa corretamente aquilo que se propõe.

Seja num parque solar, ou até mesmo numa instalação fotovoltaica residencial, pode-se instalar um Rastreador Solar, de modo a se melhorar a produção de energia elétrica pelas placas. Se o sistema for controlado por um Arduino, ao mesmo tempo em que é supervisionado via ScadaBR, poderá se obter uma otimização do processo de conversão de luz do Sol em eletricidade a um baixo custo.

Dessa forma, o trabalho produzido pode ser visto como uma chave que abre a porta para inúmeras melhorias, podendo servir como base para futuras produções acadêmicas e até mesmo para uso em sala de aula, facilitando o aprendizado de conhecimentos pertinentes a conceitos como energia solar, eletrônica analógica e digital, programação e supervisão de sistemas automatizados.

Com tudo isso, abre-se ainda o caminho para a utilização de Rastreadores Solares na região de Dourados (MS), podendo ser controlados por uma plataforma de fácil uso e aquisição como o Arduino, e supervisionado por um *software* gratuito e desenvolvido no Brasil como o ScadaBR.

5.1. TRABALHOS FUTUROS

Com as imperfeições apresentadas pelo protótipo demonstrado ao longo deste trabalho, é possível dizer que este abre margem para inúmeras melhorias, além de possibilitar a criação de diferentes formas de se produzir um Rastreador Solar.

Em relação às melhorias do modelo apresentado, pode-se indicar que se produza uma estrutura mais alta com divisórias que circundem totalmente os LDR para se tentar diminuir os efeitos da luminosidade difusa. Aprimoramentos na programação poderão ser feitos, de modo a se obter um movimento ininterrupto de leste a oeste do Seguidor Solar, independentemente da situação.

Uma opção interessante seria a de se instalar uma bateria para armazenar a energia produzida pelas placas ou até mesmo usar essa energia para suprir o próprio sistema, de modo a torná-lo autossustentável.

Com o intuito de se evitar interrupções nas medições por mau contato ou por qualquer outro fator físico, pode-se implementar uma transmissão de dados do Arduino ao ScadaBR via *internet*, conectando um *shield Ethernet* à placa controladora e configurando o protocolo *Modbus TCP/IP* no supervisório.

Pode-se também, otimizar o tempo de aquisição de dados pelo supervisório, para que se evite um acúmulo desnecessário de dados no computador, o que pode prejudicar a etapa de geração de relatórios.

É válido o teste de outros *softwares* supervisórios, tais como *LabView®* ou até mesmo o ScadaLTS, recentemente lançado e baseado no ScadaBR, sendo considerado a sua versão 2.0 e contendo várias melhorias em relação ao seu antecessor utilizado neste trabalho.

Um estudo acerca da estabilidade do sistema, calculando sua função de transferência, pode ser importante para que se verifique o quão instável está o sistema, podendo assim, utilizar alguma técnica de controle para que se corrija tal instabilidade.

Uma possibilidade interessante para se testar, é desenvolver um Seguidor Solar, no qual ao invés de sensores LDR, fosse utilizado um RTC (*Real Time Clock*), para que o movimento do servomotor fosse executado em função da hora oficial. Primeiramente, um estudo sobre a variação da posição solar ao longo do dia deverá ser feito.

Neste mesmo segmento, tem-se a opção de usar um módulo ESP8266, que pode ser programado como um Arduino e que possui conexão *WiFi* em sua estrutura. Desta forma, a hora oficial poderia ser adquirida via *internet*, evitando-se o uso do RTC.

Uma análise acerca do quanto o protótipo desenvolvido colabora para uma maior geração de energia também é grande valor, pois dessa forma, pode-se confirmar os benefícios em relação ao uso de um sistema Rastreador Solar.

REFERÊNCIAS

ALLEGRO. Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor. Disponível em: <<https://s3-sa-east-1.amazonaws.com/robocore-lojavirtual/861/ACS712-Datasheet.pdf>>. Acesso em 30 de janeiro de 2018.

ARDUINO. Arduino UNO Rev3. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>. Acesso em 08 de dezembro de 2017.

ASSUNÇÃO, Fábio Nuno Oliveira. Controlador Fuzzy implementado em ScadaBR para o controle de pressão numa adutora de água em Florianópolis. TCC (graduação) - Universidade Federal de Santa Catarina, 2016. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/171567>>. Acesso em 02 de fevereiro de 2018.

BATES, Martin. Interfacing PIC Microcontrollers – Embedded Design by Interactive Simulation. Newnes, 2013.

BLUESOL. Os Sistemas de Energia Solar Fotovoltaica. Livro Digital. Disponível em: <http://programaintegradoronline.com.br/livro/?utm_source=site-bluesol&utm_medium=livro-solar&utm_campaign=leads-iscas>. Acesso em 03 de setembro de 2017.

CABRAL, Isabelle de Souza; TORRES, Adriana Cazelgrandi; SENNA, Pedro Rocha. Energia Solar – Análise Comparativa Entre Brasil e Alemanha. In: IV Congresso Brasileiro de Gestão Ambiental, Salvador (BA), 2013.

CASTAÑEDA, Cesar Eduardo Figueroa. Desenvolvimento de um rastreador solar passivo por transferência de massa. Prodetec-Programa de Pós-Graduação de Desenvolvimento Tecnológico, 2011.

CDS. Fotoconductive Cells. Disponível em: <https://www.robocore.net/upload/attachments/sensor_ldr_gl5528_145.pdf>. Acesso em 31 de janeiro de 2018.

Centro de Referência para Energia Solar e Eólica Sérgio de Salvo Brito (2006). Energia Solar: Princípios e Aplicações. Disponível em: <http://www.cresesb.cepel.br/download/tutorial/tutorial_solar_2006.pdf>. Acesso em 08 de julho de 2018.

CERTI – SCADABR. IHM Feita em ScadaBR. Disponível em:
https://sites.google.com/a/certi.org.br/certi_scadabr/home/minicursos/iniciando-scadabr. Acesso em 10 de dezembro de 2017.

DA SILVA, Juliana Salgado Oliveira; BIANCHINI, David. **Desenvolvimento De Aplicativo Para Supervisão De Um Sistema De Comunicação Que Aciona Diferentes Dispositivos Por Meio De Uma Interface IHC Via RSSF**. Anais do XIX Encontro de Iniciação Científica. Campinas: PUC, 2014.

DATASHEETCAFE. **SG-5010 Datasheet – Servo Motor – Tower Pro**. Disponível em: <<http://www.datasheetcafe.com/sg-5010-datasheet-servo-motor/>>. Acesso em 20 de março de 2018.

EKT. **Arduino Voltage Sensor Module**. Disponível em:
http://www.ekt2.com/pdf/412_ARDUINO_SENSOR_VOLTAGE_DETECTOR.pdf. Acesso em 30 de janeiro de 2018.

ENERGYSAGE. **What is the story of solar power and where were solar panels invented?**. Disponível em: <https://news.energysage.com/the-history-and-invention-of-solar-panel-technology/>. 2017. Acesso em 08 de dezembro de 2017.

FERREIRA, Maria Julita Guerra. **Inserção da energia solar fotovoltaica no Brasil**. São Paulo, Brasil: Dissertação de Mestrado. Programa de Pós-Graduação em Energia, USP, 1993. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/86/86131/tde-05122011-141720/en.php>>. Acesso em 07 de dezembro de 2017.

GRUPO DE TRABALHO DE ENERGIA SOLAR FOTOVOLTAICA (GTES). **Manual de Engenharia para Sistemas Fotovoltaicos**. Rio de Janeiro: CEPEL - CRESESB, 2014. Disponível em:
http://www.cresesb.cepel.br/publicacoes/download/Manual_de_Engenharia_FV_2014.pdf. Acesso em 07 de dezembro de 2017.

HOSSAIN, Sabir et al. **Azimuth-Altitude Dual Axis Solar Tracker**. In: International Conference on Mechanical Engineering and Renewable Energy (ICMERE), Chittagong University of Engineering & Technology, Chittagong, Vol. 2, 2017.

INDUCTIVE AUTOMATION. **What is SCADA?**. Disponível em:
<https://inductiveautomation.com/what-is-scada>. Acesso em 08 de dezembro de 2017.

LIRA, José Rômulo Vieira. **Desenvolvimento de um rastreador solar do tipo polar com ajuste do ângulo de inclinação.** Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte, 2014.

MANUAL de Engenharia para Sistemas Fotovoltaicos. Rio de Janeiro: CRESESB, 1999.

MANUAL DO SOFTWARE SCADABR. **ScadaBR 0.7 – Sistema Open Source para Supervisão e Controle.** 2010. Disponível em: <<https://sourceforge.net/projects/scadabr/postdownload?source=dlp>>. Acesso em 22 de fevereiro de 2018.

MARINHO, Thales Rafael Ronquigali. **Solar Tracker:** Rastreador Solar Utilizando a Plataforma Arduino. Trabalho de Conclusão de Curso. Instituto Federal de Mato Grosso do Sul. Ponta Porã – MS, 2016.

McROBERTS, Michael. **Arduino Básico.** São Paulo, Novatec Editora, 2011.

MORAES, Euler Paiva de. **Seguidor Solar de um Único Eixo Inclinado.** Trabalho de Conclusão de Curso. UNICEUB – Brasília – DF, 2012.

MOREIRA, Pedro Manuel Barbosa. **Sistema de Supervisão e Controlo de Estruturas.** Dissertação (Mestrado), 53 f. Faculdade de Engenharia da Universidade do Porto – FEUP. Porto, 2011.

NASA. **The Sun: The Basics.** Disponível em: <https://www.nasa.gov/sun>. 2017. Acesso em 07 de dezembro de 2017.

NREL. **The Silicon Solar Cell Turns 50.** Disponível em: <<https://www.nrel.gov/docs/fy04osti/33947.pdf>> 2004. Acesso em 08 de dezembro de 2017.

OLIVEIRA, Carlos Antonio Alves de. **Desenvolvimento de um protótipo de rastreador solar de baixo custo e sem baterias.** Programa de Pós-Graduação em Tecnologias Energéticas e Nucleares, Universidade Federal de Pernambuco, Recife, 2007. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/10013>>. Acesso em 22 de outubro de 2017.

OZERDEM, Ozgur Cemal; SHAHIN, Adnan. **A PV Solar Tracking System Controlled By Arduino/Matlab/Simulink.** International Journal on “Technical and Physical Problems of Engineering” (IJTPE). Volume 6. 5-10, 2014.

PORTAL SOLAR. Maior parque solar do Brasil começa a operar na Bahia.
Disponível em: <https://www.portalsolar.com.br/blog-solar/energia-solar/maior-parque-solar-do-brasil-comeca-a-operar-na-bahia.html>. Acesso em 08 de dezembro de 2017.

PORTAL SOLAR. Seguidor Solar - Tracker: Vantagens e Desvantagens - Parte 1.
Disponível em: <<https://www.portalsolar.com.br/blog-solar/painel-solar/seguidor-solar---tracker-vantagens-e-desvantagens-parte-1.html>>. Acesso em 02 de fevereiro de 2018.

RAYSARO, Márcio Coiado. Sistema open-source de supervisão, controle e aquisição de dados. Trabalho de Conclusão de Curso, Universidade de Cuiabá, Cuiabá (MT), 2012.

RIBEIRO, Sandro Carneiro; PRADO, Pedro Paulo Leite; GONÇALVES, João Bosco.
Projeto e Desenvolvimento de um Rastreador Solar. In: IX Seget - Simpósio de Excelência em Gestão e Tecnologia, 2012, Resende – RJ. IX Seget, 2012.

ROCHA, Helder Lima Santos da. **Introdução à Eletrônica para Artistas.** Livro digital. 2017.

SCADABR. **O que é ScadaBR?** Disponível em: <http://www.scadabr.com.br/>. Acesso em 10 de dezembro de 2017.

VALDOREIX GREENPOWER. **The Benefits of Solar Trackers.** Disponível em: <<http://www.valdoreix-gp.com/the-benefits-of-solar-trackers/>>. Acesso em 02 de fevereiro de 2018.

APÊNDICE

Programação do Arduino.

```
#include <SimpleModbusSlave.h> //Biblioteca que permite a comunicação entre Arduino e
ScadaBR
```

```
#include <Servo.h> //Biblioteca que permite o controle do servomotor
```

```
Servo servoRotacao; //Criação do objeto relacionado ao controle do servomotor
```

```
int pos; //Variável que armazenará a posição em graus do servomotor
int LDR1 = A0; //Pino em que o LDR 1 está conectado
int LeituraLDR1 = 0; //Variável que armazenará a leitura do pino A0
int LDR2 = A1; //Pino em que o LDR 2 está conectado
int LeituraLDR2 = 0; //Variável que armazenará a leitura do pino A1
int tolerancia = 5; //Diferença máxima permitida entre as leituras dos LDRs
```

```
unsigned long delay1 = 0; //Variável usada execução de uma pausa na execução do código
```

```
//Declaração dos offsets
```

```
enum {
    LDR_1, //Registrador de Offset 0
    LDR_2, //Registrador de Offset 1
    POS_SERVO, //Registrador de Offset 2
    TENSAO_PLACA, //Registrador de Offset 3
    CORRENTE_PLACA, //Registrador de Offset 4
    HOLDING_REGS_SIZE //Variável que armazenará cada offset
};
```

```
//Variável responsável por registrar os valores dos offsets e enviá-los ao ScadaBR
```

```
unsigned int holdingRegs[HOLDING_REGS_SIZE];
```

```

void setup(){
    //Configuração da comunicação entre Arduino e ScadaBR
    //((Comunicação Serial, baud rate de 9600, 1 bit de início - 8 bits de dados - 1 bit de parada, id
    //do escravo, habilitação de transmissão de 2 bytes)
    modbus_configure(&Serial, 9600, SERIAL_8N1, 1, 2, HOLDING_REGS_SIZE,
    holdingRegs);

    //Configuração da taxa de atualização
    //((baud rate de 9600, 1 bit de início - 8 bits de dados - 1 bit de parada, id do escravo)
    modbus_update_comms(9600, SERIAL_8N1, 1);

    pinMode(LDR1, INPUT); //Configura LDR Leste para entrada de dados
    pinMode(LDR2, INPUT); //Configura LDR Oeste para entrada de dados

    servoRotacao.attach(9); //Associa o servomotor ao pino digital 9

    delay(5000); //Atraso de 5s para estabilização da tensão do Arduino
}

void loop(){
    if((millis() - delay1) > 500){
        LeituraLDR1 = analogRead(LDR1); //leitura do LDR Leste
        LeituraLDR2 = analogRead(LDR2); //Leitura do LDR Oeste

        //Se as diferenças entre os valores inteiros
        //e positivos das leituras forem menor ou igual a 5 não faça nada
        if((abs(LeituraLDR1 - LeituraLDR2) <= tolerancia) || (abs(LeituraLDR2 - LeituraLDR1) <=
        tolerancia)){
            }
        else{ //Senão
    }
}

```

```
if(LeituraLDR1 > LeituraLDR2){ //Se a leitura do LDR Leste for maior que a do Oeste
    pos = --pos; //Vá para Leste
    if(pos < 0) {pos = 0;} //Ao se atingir o limite do servomotor, permaneça ali
}
if(LeituraLDR1 < LeituraLDR2){ //Se a leitura do LDR Leste for maior que a do Oeste
    pos = ++pos; //Vá para Oeste
    if(pos > 180) {pos = 180;} //Ao se atingir o limite do servomotor, permaneça ali
}

servoRotacao.write(pos); //Envio do sinal da posição para o servomotor
delay1 = millis();
}
}
//Envio de dados ao ScadaBR
modbus_update();
holdingRegs[LDR_1] = analogRead(A0);
holdingRegs[LDR_2] = analogRead(A1);
holdingRegs[POS_SERVO] = pos;
holdingRegs[TENSAO_PLACA] = analogRead(A2);
holdingRegs[CORRENTE_PLACA] = analogRead(A3);
}
```