

UNIVERSIDADE DO ESTADO DO AMAZONAS  
ESCOLA SUPERIOR DE TECNOLOGIA  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Mateus de Azevedo Ramos

**ANÁLISE DE EFICIÊNCIA ENERGÉTICA DE UM MÓDULO FOTOVOLTAICO  
COM RASTREADOR SOLAR COMPARADO A UM MÓDULO FIXO**

MANAUS  
2021

UNIVERSIDADE DO ESTADO DO AMAZONAS  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Mateus de Azevedo Ramos

**ANÁLISE DE EFICIÊNCIA ENERGÉTICA DE UM MÓDULO FOTOVOLTAICO  
COM RASTREADOR SOLAR COMPARADO A UM MÓDULO FIXO**

Trabalho de Conclusão de Curso submetido  
à Coordenação do curso de Engenharia de  
Controle e Automação da Universidade do  
Estado do Amazonas como parte dos  
requisitos necessários para obtenção do grau  
de Engenharia em conformidade com as  
normas ABNT

Orientador: Me. Moisés Pereira Bastos

MANAUS  
2021

Mateus de Azevedo Ramos

**ANÁLISE DE EFICIÊNCIA ENERGÉTICA DE UM MÓDULO FOTOVOLTAICO  
COM RASTREADOR SOLAR COMPARADO A UM MÓDULO FIXO**

Trabalho de Conclusão de Curso submetido  
à Coordenação do curso de Engenharia de  
Controle e Automação da Universidade do  
Estado do Amazonas como parte dos  
requisitos necessários para obtenção do grau  
de Engenharia em conformidade com as  
normas ABNT

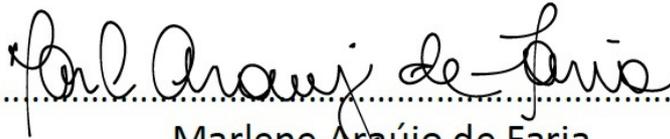
Trabalho aprovado. Manaus, 29 de julho de 2021:



Moisés Pereira Bastos



Israel Mazaira Morales



Marlene Araújo de Faria



Rodrigo Farias Araújo

MANAUS  
2021

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por me dar a capacidade de sonhar e poder buscar os meus objetivos. Sem Ele eu nada seria.

Aos meus professores, que durante a minha graduação não mediram esforços para transmitir os melhores conteúdos com métodos claros e eficazes.

À Universidade do Estado do Amazonas por, apesar das dificuldades, me permitir cursar a graduação dos meus sonhos sem custos.

Aos meus amigos, que me ajudaram com palavras de incentivo, companheirismo, auxílio nos momentos difíceis e felizes. Dentre os vários, destaco: Mayara Lima, Viviane Karine, Giovana Brandão, Matheus Souza, Isabela Zanotto e, as minhas amigas de graduação, Silvia Lima, Giovanna Bezerra e Heloísa Ribeiro.

Ao meu irmão, André Ramos, meu psicólogo particular, que sempre se faz presente nos meus piores e melhores momentos, estando constantemente apto a me escutar e aconselhar.

Ao meu pai, Julio de Oliveira Ramos, que me ensinou desde criança que nada vem de graça e que, onde há esforço, conseqüentemente haverá recompensa. Com o seu exemplo de vida, me ensinou a ser corajoso e sempre buscar capacitação.

À minha maior incentivadora, amiga e conselheira, minha mãe, Elizabete Souza de Azevedo, por jamais desistir de mim e sempre acreditar no meu potencial mesmo quando eu não acreditava. Foi a primeira a comemorar a minha realização de conseguir ingressar nessa Universidade e esteve comigo durante toda essa árdua trajetória, eu não teria conseguido chegar até aqui sem ela.

## RESUMO

A energia solar vem em constante crescimento nos últimos anos como uma alternativa para a geração de energia tradicional. Estudos vêm sendo feitos ao redor do mundo para aprimorar técnicas de montagem dos sistemas de geração de energia solar, aumentando a eficiência dos módulos fotovoltaicos, a fim de garantir o máximo de geração das placas. A eficiência dos módulos foi a base dessa pesquisa. Utilizando um módulo junto a um rastreador solar, verificamos se houve melhora na eficiência do módulo fotovoltaico comparado a eficiência de um módulo fixo. O rastreador foi montado com peças de madeira, canos e uma base de MDF. Os movimentos executados pela estrutura foram possíveis por intermédio de dois servomotores, um disposto para garantir o movimento no eixo azimutal e outro no de altitude. Para que pudessemos controlar os movimentos e ler as grandezas geradas pelos módulos, utilizamos dois módulos de prototipagem Arduino UNO e sensores de corrente, potência e tensão INA219. Uma interface foi desenvolvida no *software Processing* para permitir uma supervisão em tempo real das eficiências de ambos os módulos. Além disso, a interface possibilitava ao usuário executar um controle manual sobre os movimentos da estrutura que também se movimentava de forma automática. O modo automático de operação foi possível graças a utilização de sensores fotorresistivos, LDRs, que são capazes de alterar a sua resistividade quando há incidência luminosa. Para que a interface controlasse os movimentos dos servos e interpretasse as leituras dos sensores fotorresistivos, utilizamos a biblioteca Firmata em um dos Arduino utilizados. Como resultados obtivemos, em um dos dias analisados, uma melhora de 2,2% na eficiência do módulo acoplado ao rastreador, tal resultado poderia ser melhor se tivéssemos utilizado painéis nominalmente mais eficientes, mas ainda assim, houve comprovação no ganho de eficiência.

**Palavras-chave:** Eficiência Energética, Energia Solar, Módulos Fotovoltaicos, Rastreador Solar.

## ABSTRACT

Solar energy has been constantly growing in recent years as an alternative to traditional energy generation. Studies have been carried out around the world to improve assembly techniques for solar energy generation systems, increasing the efficiency of photovoltaic modules, in order to guarantee the maximum generation of the plates. The efficiency of the modules was the basis of this research, using a module together with a solar tracker, checking if there is an improvement in the efficiency of the photovoltaic module and comparing with the efficiency of a fixed module. The tracker was assembled with wooden parts, pipes and an MDF base. The movements performed by the structure were made possible by means of two servomotors, one arranged to ensure movement in the azimuthal axis and the other in the altitude axis. So that we could control the movements and read the quantities generated by the modules, we used two Arduino UNO prototyping modules and INA219 current, power and voltage sensors. An interface was developed in the Processing software to allow real-time supervision of the efficiencies of both modules. In addition, the interface allowed the user to perform manual control over the structure's movements, which could also move automatically. The automatic mode of operation was possible thanks to the use of photoresistive sensors, LDRs, which are able to change their resistivity when there is light incidence. In order for the interface to control the movements of the servos and interpret the readings of the photoresistive sensors, we used the Firmata library in one of the Arduino used. As a result, in one of the days analyzed, we obtained an improvement of 2.2% in the efficiency of the module coupled to the tracker, this result could have been better if we had used nominally more efficient panels, but even so, there was proof of efficiency gain.

**Key-words:** Energy Efficiency, Photovoltaic Modules, Solar Energy, Solar Tracker.

## LISTA DE ILUSTRAÇÕES

|   |    |
|---|----|
| Figura 1 - Protótipo de rastreador solar.....   | 17 |
| Figura 2 - Comparação de eficiência do rastreador e o módulo posicionado horizontalmente. | 19 |
| Figura 3 - Protótipo de rastreador solar.....   | 21 |
| Figura 4 - Sistema fotovoltaico.....  | 25 |
| Figura 5 - Tipos de módulos fotovoltaicos.....  | 26 |
| Figura 6 - Circuito equivalente às placas solares.....                                    | 26 |
| Figura 7 - Controlador de carga.....  | 28 |
| Figura 8 - Bateria estacionária.....  | 29 |
| Figura 9 - Eixos de movimentação referenciados ao sol.....                                | 30 |
| Figura 10 - Sensor Fotorresistor.....   | 31 |
| Figura 11 – Microcontrolador.....   | 32 |
| Figura 12 - Kit de prototipagem Arduino.....  | 33 |
| Figura 13 - Sensor de corrente, tensão e potência INA219.....                             | 34 |
| Figura 14 - Servo motor internamente.....   | 35 |
| Figura 15 - Controle de posição por PWM.....  | 35 |
| Figura 16 - Script desenvolvido no Processing.....  | 36 |
| Figura 17 - Fluxograma do sistema fotovoltaico fixo.....                                  | 38 |
| Figura 18 - Fluxograma do sistema com rastreador solar.....                               | 38 |
| Figura 19 - Módulo fotovoltaico de 10W.....   | 39 |
| Figura 20 - Servomotor utilizado no projeto.....  | 40 |
| Figura 21 - Circuito dos servomotores simulado no Tinkercad.....                          | 41 |
| Figura 22 - Trecho do circuito real dos motores.....                                      | 41 |
| Figura 23 - Base do protótipo do rastreador.....  | 42 |
| Figura 24 - Peça de metal acoplada ao servomotor.....                                     | 43 |
| Figura 25 - Peça de madeira posta sobre a de metal.....                                   | 43 |
| Figura 26 - Estrutura sem o módulo e sem a peça com LDR's.....                            | 44 |
| Figura 27 - Estrutura com módulo inserido.....  | 44 |
| Figura 28 - Caminho para gerenciar bibliotecas do Arduino.....                            | 45 |
| Figura 29 - Instalação da biblioteca Firmata.....   | 46 |
| Figura 30 - Caminho para incluir o script StandardFirmata.....                            | 46 |

|   |    |
|---|----|
| Figura 31 - Carregando a StandardFirmata no Arduino.....  | 47 |
| Figura 32 - Instalação biblioteca Adafruit INA219.....  | 47 |
| Figura 33 - Sketch do Arduino para os sensores de corrente e tensão.....                            | 48 |
| Figura 34 - Circuito projetado dos sensores e Arduino.....  | 49 |
| Figura 35 - Circuito dos sensores montado numa mini protoboard.....                                 | 50 |
| Figura 36 - Sensores ligados às lâmpadas.....   | 50 |
| Figura 37 - Peça suporte para LDR.....  | 51 |
| Figura 38 - Circuito dos LDRs com resistores e jumpers de alimentação.....                          | 51 |
| Figura 39 - Circuito já com jumpers de sinal e alimentação.....                                     | 52 |
| Figura 40 - Circuito dos LDRs montado no Tinkercad.....   | 53 |
| Figura 41 - Circuito LDR real.....  | 53 |
| Figura 42 - Bibliotecas utilizadas no Processing.....   | 54 |
| Figura 43 - Código de declaração dos pinos e posições dos servos.....                               | 54 |
| Figura 44 - Trecho do código dos posicionamentos do joystick.....                                   | 56 |
| Figura 45 - Parte do código de controle manual.....   | 56 |
| Figura 46 - Trecho do código para armazenar os dados lidos na porta serial.....                     | 56 |
| Figura 47 - Interface final do sistema supervisorio.....  | 57 |
| Figura 48 - Sistemas instalados para coleta.....  | 58 |
| Figura 49 - Sistema fixo de coleta de energia solar.....  | 59 |
| Figura 50 - Gráfico da eficiência dos módulos no primeiro dia.....                                  | 60 |
| Figura 51 - Gráfico dos dados coletados no dia 10 sem o intervalo onde houve erro nas medições..... | 60 |
| Figura 52 - Gráfico de eficiência dos módulos no dia 11.....  | 62 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 - Dados medidos no intervalo de 12:00 às 13:00.....               | 61 |
| Tabela 2 - Médias das tensões medidas no primeiro dia.....                 | 62 |
| Tabela 3 - Dados medidos no intervalo de 12:01 à 13:02 no segundo dia..... | 63 |
| Tabela 4 - Médias das tensões medidas no segundo dia.....                  | 64 |

## SUMÁRIO

|  |    |
|--|----|
| <b>1 INTRODUÇÃO</b> .....  | 12 |
| <b>1.1 Tema</b> .....  | 13 |
| <b>1.2 Problemática</b> .....  | 13 |
| <b>1.3 Hipótese</b> .....  | 14 |
| <b>1.4 Justificativa</b> .....   | 14 |
| <b>1.5 Objetivos</b> .....   | 15 |
| 1.5.1 Objetivo Geral .....   | 15 |
| 1.5.2 Objetivos Específicos .....  | 15 |
| <b>2 TRABALHOS RELACIONADOS</b> .....  | 16 |
| <b>2.1 Solar Tracking System using Microcontroller</b> .....   | 16 |
| <b>2.2 Eficiência energética de um painel fotovoltaico com rastreador solar baseado em sensores LDR comparado a um painel em ângulo fixo</b> ..... | 17 |
| <b>2.3 An Automated Intelligent Solar Tracking Control System With Adaptive Algorithm for Different Weather Conditions</b> .....                   | 18 |
| <b>2.4 Performance Comparison of Dual Axis Solar Tracker with Static Solar System in Ural Region of Russia</b> .....                               | 20 |
| <b>2.5 Effect of temperature on the I-V characteristics of a Polycrystalline Solar Cell</b> .....  | 22 |
| <b>3 REFERENCIAL TEÓRICO</b> .....   | 24 |
| <b>3.1 A Energia Solar</b> .....   | 24 |
| <b>3.2 Sistema Solar Fotovoltaico</b> .....  | 24 |
| 3.2.1 Componentes Básicos de um Sistema Solar Fotovoltaico .....   | 25 |
| 3.2.1.1 Placas Fotovoltaicas .....   | 25 |
| 3.2.1.2 Controlador .....  | 28 |
| 3.2.1.3 Bateria .....  | 29 |
| 3.2.1.4 Inversor de Frequência .....   | 30 |
| <b>3.3 Rastreador Solar</b> .....  | 30 |
| <b>3.4 Sensores Fotoresistores (<i>Light Dependent Resistor</i> - LDR)</b> .....   | 31 |
| <b>3.5 Microcontroladores</b> .....  | 32 |
| 3.5.1 Arduino .....  | 32 |
| 3.5.1.1 Arduino <i>data</i> logger .....   | 34 |
| <b>3.6 Servo Motor</b> .....   | 34 |
| <b>3.7 Processing</b> .....  | 36 |
| <b>4 METODOLOGIA</b> .....   | 38 |
| <b>4.1 Sistema Fotovoltaico</b> .....  | 39 |

|  |    |
|--|----|
| <b>4.2 Rastreador Solar</b> .....                          | 40 |
| 4.2.1 Motores.....   | 40 |
| 4.2.2 Estrutura.....                                       | 42 |
| 4.2.3 Sistema de Controle.....                             | 45 |
| 4.2.3.1 Sketchs dos Arduinos.....                          | 45 |
| 4.2.3.2 INA219.....  | 49 |
| 4.2.3.3 LDRs.....  | 51 |
| <b>4.3 Interface de Supervisão e Controle</b> .....        | 53 |
| <b>5 RESULTADOS E DISCUSSÕES</b> .....                     | 58 |
| <b>5.1 Instalação dos Sistemas Fixo e Rastreador</b> ..... | 58 |
| <b>5.2 Primeiro dia de coleta de dados</b> .....           | 59 |
| <b>5.3 Segundo dia de coleta de dados</b> .....            | 62 |
| <b>6 CONSIDERAÇÕES FINAIS</b> .....                        | 65 |
| <b>6.1 Trabalhos Futuros</b> .....                         | 65 |
| <b>REFERÊNCIAS</b> .....                                   | 67 |
| <b>APÊDICE A – <i>SCRIPT PROCESSING</i></b> .....          | 69 |

## 1. INTRODUÇÃO

A nível mundial, o consumo de energia elétrica aumenta exponencialmente a cada ano que se passa, o crescimento populacional, a demanda nas indústrias, são fatores que contribuem consideravelmente para esse aumento. Segundo Rosa e Gasparin (2016) esse contexto faz com que as nações sempre busquem meios para a geração de energia que continuem sendo rentáveis e que não causem danos ao meio ambiente, uma vez que o método mais utilizado atualmente para a geração de energia é o hidrelétrico, que por si só causa um impacto gigantesco na natureza.

Uma alternativa para a amenização desse problema é a utilização de meios naturais e inesgotáveis para a geração de energia elétrica. A geração eólica e solar são exemplos de métodos eficientes e que não causam danos a natureza se comparados às hidrelétricas. A energia solar, em especial, utiliza o sol como principal fonte de geração. A estrela fornece, aproximadamente, 4 milhões exajoules ( $1\text{EJ} = 10^{18}\text{J}$ ) durante um ano em toda a superfície terrestre (Kabir et al. 2018), porém ainda não há tecnologia suficiente investida para a utilização desse benefício gratuito.

Quando se trata de energia solar, o Brasil tem um grande potencial pois a radiação incidente no país é maior se comparado à Europa por exemplo, uma vez que o território brasileiro tem tamanho continental, a não utilização dessa vantagem para a geração de energia significa perda de dinheiro. A Bahia é o estado com o maior índice de radiação do país, cerca de  $2.350\text{kWh/m}^2$  incidem sobre a região anualmente. Um dos países líderes em geração de energia solar, a Alemanha possui uma incidência que varia entre  $900$  a  $1.250\text{kWh/m}^2$  ao ano, essa sendo menor que a incidência mais baixa registrada no Brasil, localizada no litoral norte do Estado de Santa Catarina, onde incidem cerca de  $1500\text{kWh/m}^2$ . O Amazonas possui um potencial fotovoltaico de cerca de  $3.679$  GWh/ano, porém o consumo, segundo a EPE, no ano de 2013 foi de, somente,  $1.784$  GWh (DA ROSA; Gasparin 2016).

Para a geração de energia elétrica com a utilização do sol, o método mais comum é por meio de módulos fotovoltaicos ou placas solares, responsáveis por transformar os fótons emitidos pelo sol em corrente elétrica, transformação esta causada pelo efeito fotovoltaico. Existem diversos tipos, tamanhos e preços de placas solares, quando montadas juntas têm a capacidade de transformar a luz incidente do sol em energia elétrica de grande escala, podendo ser utilizada para utilização doméstica ou até industrial (Sampaio; González, 2017).

A eficiência dos módulos fotovoltaicos depende de diversos fatores. O tamanho, por exemplo, influencia na medida que mais fótons poderão ser absorvidos e, conseqüentemente,

transformados em corrente elétrica. Outro fator a ser observado é o material do qual a placa é feita, segundo Karki (2015), módulos feitos à base de silício policristalino perdem eficiência quando expostos a altos índices de temperatura.

Diversos estudos são feitos ao redor do mundo para trazer ganhos em eficiência para as placas. Um dos mais estudados atualmente é a montagem dos módulos fotovoltaicos ligados a rastreadores solar. Comprovadamente, este método influencia diretamente na melhoria da eficiência, uma vez que sua proposta é direcionar as placas diretamente para o sol durante a maior parte do dia, uma maior quantidade de fótons será absorvida se comparado ao método onde as placas ficam fixas. O sol se move durante o dia e, dessa forma, quando utilizamos placas fixas, a incidência solar direta nos módulos tende a ocorrer somente no intervalo de tempo em que a estrela está posicionada perpendicularmente ao módulo (Masih; Odinaev, 2019).

Ainda que aumente a eficiência dos módulos, existem regiões onde não é viável utilizar o método com rastreadores solar. O custo para a implementação do sistema, dependendo do tamanho do mesmo, é elevado e o retorno do investimento pode demorar a ponto de não ser vantajoso (Freitas 2018). Contudo, dependendo da região, da incidência solar, que em algumas regiões do mundo chega a ser de 15 horas diárias, e dos módulos utilizados, a melhora na eficiência é tamanha a ponto de que a não utilização dos rastreadores significa uma perda enorme de retorno, tanto energético quanto financeiro (Masih; Odinaev, 2019)

Sendo assim, faz-se necessário a análise da melhora de eficiência dos módulos solares juntos aos rastreadores solar na região amazônica, esse estudo pode contribuir para a maior utilização desse método na região e também para o desenvolvimento da mesma, além de oferecer alternativas para geração de energia.

## **1.1 Tema**

Análise de eficiência energética de um painel fotovoltaico com rastreador solar comparado a um painel fixo.

## **1.2 Problemática**

A necessidade de utilização de fonte renováveis de energia vem aumentando gradativamente devido ao crescimento dos custos de energia tradicional. A energia solar é uma dessas fontes mais difundidas pelo mundo, sendo uma das que oferece um melhor custo

benefício. Porém, não é possível garantir que uma placa solar irá operar com a sua eficiência máxima de forma constante sendo esta fixa. Como resposta a esse problema, os rastreadores solar (*Solar Trackers*) fazem com que a eficiência desses sistemas melhore de forma notória. Ainda assim, não se pode afirmar, principalmente na região Amazônica sendo está localizada próxima a Linha do Equador, o quão eficiente esses sistemas são em comparação aos de placas fixas.

### **1.3 Hipótese**

A comprovação do aumento da eficiência das placas solares quando instaladas juntas aos rastreadores solar, na Região Amazônica, pode influenciar no aumento do uso desta aplicação.

### **1.4 Justificativa**

Sendo a Linha do Equador o paralelo incidente na Região Amazônica, torna assim esta parte do país mais quente ao longo do ano em comparação a outras regiões do Brasil. Nela, a incidência do sol é mais direta independente da estação do ano, o que torna mais viável a utilização de placas fotovoltaicas como geradores de energia. (F. P. da Silva et al., 2016)

Atualmente, o estado brasileiro que mais utiliza a energia solar é Minas Gerais, porém, o Amazonas vem em crescente nessa área há alguns anos. Segundo o Sebrae, muitos pesquisadores veem o Estado do Amazonas como uma promissora potência no âmbito da geração de energia solar e, com isso, muitos investimentos serão voltados para esta aplicação na região. (Portal Solar 2019)

Conforme muitas pesquisas feitas ao redor do mundo, a utilização de placas fotovoltaicas com dois graus de liberdade aumenta consideravelmente a eficiência de geração, o fato de fazer com que as placas se movimentem de acordo com a posição do sol, traz um ganho considerável de energia gerada. Com isso, aderir a esse método significa almejar um retorno energético superior se compararmos ao método de placas fixas.

Sendo assim, um estudo voltado para a comprovação dessa eficácia na Região Amazônica faz-se necessário, a utilização desse método pode colaborar para o desenvolvimento social e econômico do Estado, e uma vez comprovado, o Amazonas pode crescer exponencial no âmbito da geração de energia solar.

Poderemos então observar se a eficiência encontrada nos módulos com rastreadores solar é superior aos fixos, a ponto de desconsiderarmos os gastos maiores para o desenvolvimento e aplicação de um sistema automatizado em relação a um não móvel.

## **1.5 Objetivos**

### **1.5.1 Objetivo Geral**

Desenvolver um sistema capaz de se movimentar de forma automática ou manual com dois eixos de liberdade podendo assim acompanhar a trajetória do sol. Juntamente com módulos fotovoltaicos, esse sistema servirá para que um estudo seja feito ao se comparar a eficiência energética do módulo orientado ao sol em relação aos sistemas fixos de geração de energia solar. Assim, poderemos levantar dados capazes de comprovar quais das situações, automática ou fixa, é a mais adequada, tanto em eficiência quanto em custo benefício, para a região amazônica, assim podendo contribuir para o desenvolvimento dessa área no estado.

### **1.5.2 Objetivos Específicos**

1. Reunir referências bibliográficas sobre o tema.
2. Desenvolver o mecanismo para movimento automático ou manual da placa.
3. Montar o protótipo do rastreador solar.
4. Coletar dados de geração de energia das placas por meio de microcontroladores.
5. Elaborar um código para tratar os dados e estipular padrões.
6. Estipular tabelas comparativas.
7. Realizar testes e ajustes no sistema.

## 2. TRABALHOS RELACIONADOS

### 2.1 *Solar Tracking System using Microcontroller*

No artigo de Priyanjan Sharma et al. [2014], os autores desenvolveram um sistema capaz de acompanhar o movimento do sol utilizando um Microcontrolador (AT89S52), dois LDRs (*Light Dependent Resistors*) ou resistores dependentes de luz em português e um motor de passo, além da placa solar e toda a estrutura de suporte.

Partindo da premissa de que, dessa forma, uma maior eficiência seria obtida das placas solares, elaboraram um sistema do qual a tensão elétrica fornecida pela rede era reduzida à 9V por um transformador, seguido por uma ponte retificadora responsável pela alternância de corrente alternada para corrente contínua. Depois disso, utilizaram de um regulador de tensão para que todo o sistema fosse alimentado com 5V. Todo o processo do sistema também foi representado por um diagrama de blocos exposto pelos pesquisadores.

Os LDRs, por sua vez, foram dispostos a fim de que pudessem ser captadas a posição do sol e a condição, se era dia ou noite, para que dessa forma o sistema correspondesse da maneira mais adequada. O resistor incumbido de rastrear a posição do sol foi disposto junto a placa solar de uma forma onde seja possível se manter sempre perpendicular aos raios solares, dessa maneira o sistema se movimentaria de acordo com o sinal transmitido pelo sensor, podendo permanecer na posição que, segundo os autores, elevaria a eficiência da placa fotovoltaica.

O motor de passo, que nesse sistema se encarrega de viabilizar a movimentação em somente um eixo, funciona de acordo com o sinal transmitido pelo sensor acima mencionado. Uma vez que o LDR emite um sinal, respeitando as regras que foram estipuladas em sua programação, o controlador encaminha um comando para o motor de passo, seja para o seu movimento ou parada. Tais ações dependem de como os autores codificaram o reconhecimento do resistor de luz, uma vez que este diminui a sua resistência com o aumento da intensidade de luz recebida. Em outras palavras, quando há pouca luz incidente, o sensor informa ao controlador e este aciona o motor de passo, a fim de que este altere a posição do sistema até o ponto onde os sensores se estabilizem novamente.

Já o outro LDR, que não possui uma posição necessariamente obrigatória de disposição, funciona como um informante para o sistema, seu objetivo é identificar se está de dia ou de noite. Uma vez identificada a ausência de uma fonte luminosa, que nesse caso é o sol, o sensor

comunica isso ao sistema e este automaticamente se posiciona para o leste, posição esta referente ao nascer do sol do dia seguinte.

Realizando a coleta de dados tanto em dias ensolarados quanto em dias com muitas nuvens, os autores concluíram que houve um acréscimo de, mais ou menos, 13% na média de potência gerada pelas placas solares em comparação aos sistemas fixos. Com isso, comprovaram que há uma vantagem, ainda que não tão superior, quando possuímos sistemas capazes de acompanhar a trajetória solar.

Esta pesquisa servirá de grande colaboração para o desenvolvimento deste projeto, uma vez que ambos se assemelham na utilização de microcontroladores para a comunicação e controle do sistema. Além disso, podemos ter como base o modelo elaborado em diagrama de blocos que os autores utilizaram do sistema desenvolvido por eles. No geral, os projetos estão bem próximos quando nos referimos ao saber se existe uma eficácia superior dos sistemas solares automáticos, porém, se diferenciam a partir da premissa de que, dependendo da localidade terrestre, pode existir uma maior incidência solar durante o dia.

## **2.2 Eficiência energética de um painel fotovoltaico com rastreador solar baseado em sensores LDR comparado a um painel em ângulo fixo**

A dissertação feita por Gilberto Freitas (2018) teve por objetivo o estudo de eficácia de placas fotovoltaicas associadas a sistemas capazes de rastrear o posicionamento do sol em comparação a sistemas com placas fixas e, em consequência do resultado, concluir se existe um bom custo benefício que justifique a implementação desse método na região. A coleta de dados foi realizada em São Paulo entre os dias 24 e 27 de outubro do ano de 2017.

Figura 1: Protótipo de rastreador solar



Fonte: Eficiência energética de um painel fotovoltaico com rastreador solar baseado em sensores LDR comparado a um painel em ângulo fixo, Gilberto Freitas [2018]

A Figura 1 mostra o protótipo desenvolvido pelo autor, que conta com duas placas solares, uma dedicada coleta dos raios solares enquanto permanece fixa e outra interligada ao sistema rastreador. O pesquisador também utilizou LDR's (resistores dependentes de luz) para fazer o reconhecimento da posição solar, a transmissão de informação e sinais, estes ficaram distribuídos de forma que quando estiverem com a mesma intensidade luminosa significa que os raios estão em posição perpendicular à placa.

Além disso, o autor utilizou um Arduino Uno para fazer a recepção dos sinais e interpretação. Dois servomotores ficaram responsáveis por fazer o rastreador se movimentar de acordo com a posição estipulada pelos sensores. Para a coleta de dados, o pesquisador utilizou dois sistemas Arduino *datalogger* com sensores de corrente, um para a coleta referente ao sistema fixo e o outro ao sistema com rastreador.

Como resultado, o pesquisador pôde concluir que existe um ganho em eficiência ao utilizar o sistema rastreador, mas que esse ganho não compensa o investimento haja vista que boa parte da energia gerada foi utilizada para manter o próprio sistema em funcionamento. Quanto ao custo-benefício, o autor expôs que também não há vantagem uma vez que seu projeto automático teve um custo 216,60% superior ao sistema fixo.

Com a dissertação de Gilberto, temos um exemplo real de que manter os servos energizados durante todo o tempo de coleta, mesmo que o sistema não exija isso, é prejudicial para a eficiência resultante. Com a implementação do método de acionarmos os servos somente depois de intervalos de tempo, mesmo que a pesquisa de Gilberto também tenha sido realizada no Brasil, por se tratarem de regiões diferente, podemos esperar resultados mais promissores.

### ***2.3 An Automated Intelligent Solar Tracking Control System With Adaptive Algorithm for Different Weather Conditions.***

A pesquisa de Nurzhigit Kuttybay et al. [2019], realizada na Malásia, traz a discussão sobre o posicionamento das placas solares quando há alterações no tempo, no caso do trabalho em questão, quando o tempo está nublado ou com muitas nuvens de maneira a encobrir o sol.

A proposta dos autores visava comprovar que, quando se está em tempo nublado, a posição em que se obtém a maior eficiência das placas, é a horizontal. Para isto, utilizaram-se de 3 placas solares, uma principal que era a maior e outras duas auxiliares, que por sua vez eram menores.

Os módulos menores se dividiam em funções específicas, um permanecia sempre na posição horizontal e o outro estava ligado a um sistema de rastreamento solar. Ambos se

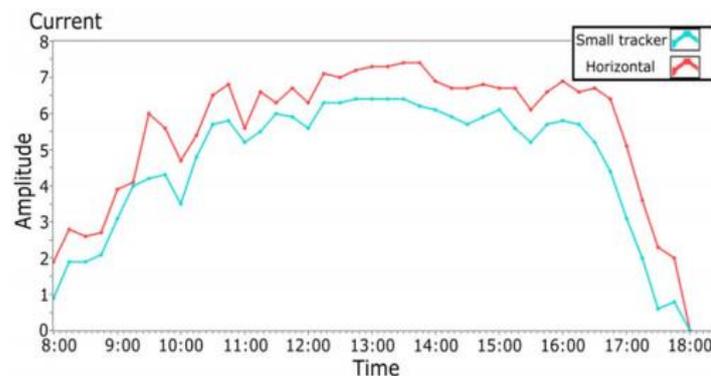
encontravam ligados a baterias, estas responsáveis por armazenar a energia gerada por eles. Transmissores *wireless* faziam a comunicação com um computador que guardava as leituras feitas pelos mesmos.

Com base nos dados recebidos pelos transmissores, o computador reconhecia qual das placas estava com uma maior eficiência naquele momento. Com isso, uma vez que a bateria do módulo fixo na horizontal recebia uma maior carga, o computador enviava uma informação, também via *wireless*, para o sistema com a placa fotovoltaica maior, dessa forma os motores de passo utilizados pelos desenvolvedores eram acionados de modo que posicionassem a placa na posição horizontal.

Por outro lado, quando a placa menor, que se encontrava ligada a um rastreador solar, abastecia a bateria com uma maior carga, os motores funcionavam de uma forma com que tornasse um sistema com o módulo maior, também em um rastreador solar. Dessa forma, a placa maior poderia copiar a posição ou movimento que faria com que uma maior eficiência fosse obtida naquele exato momento.

Durante o dia do qual os pesquisadores fizeram a coleta dos dados, o tempo se alternou de nublado para com neblina pela parte da manhã. Já no começo da tarde, nuvens encobriram boa parte do céu e, conseqüentemente, o sol, fazendo com que não houvesse uma grande incidência de luz solar durante o dia. Sendo assim, o módulo maior permaneceu durante toda a coleta de dados na posição horizontal, seguindo o posicionamento da placa menor fixada horizontalmente. A Figura 2 mostra o gráfico que os pesquisadores obtiveram como resultado, observando a comparação entre o módulo fixado e o móvel.

Figura 2: Comparação de eficiência do rastreador e o módulo posicionado horizontalmente



Fonte: An Automated Intelligent Solar Tracking Control System With Adaptive Algorithm for Different Weather Conditions, Nurzhigit Kuttybay et al. [2019].

Com a análise dos dados feitas posteriormente, foi possível testificar que durante todo o dia em questão, o módulo menor que estava na horizontal abasteceu a sua bateria com uma maior carga se comparado ao módulo com o rastreador. Além da carga, os pesquisadores também expuseram comparações de corrente, tensão e potência, estas grandezas também mostraram que houve diferença entre os dois módulos, o fixo também prevaleceu nessas comparações. De modo geral, notou-se uma diferença em torno de 18% a mais de eficiência no módulo adaptativo.

Os resultados expostos nesse artigo servirão de grande auxílio para o desenvolvimento do nosso trabalho. Uma vez que o principal objetivo é o aumento de eficiência, faz-se importante incluímos no desenvolvimento do projeto mecanismos que permitam ao sistema se adaptar às condições climáticas relatadas na pesquisa mencionada acima. Tendo em vista que serão utilizados sensores ópticos, poderão ser incluídas, na programação do controlador, regras para o melhor posicionamento da placa fotovoltaica de acordo com o tempo climático, assim podendo aumentar a sua eficiência energética.

#### ***2.4 Performance Comparison of Dual Axis Solar Tracker with Static Solar System in Ural Region of Russia.***

O trabalho desenvolvido por Adven Masih et. al [2019], tem por objetivo estudar a eficiência dos Rastreadores Solar de Dois Eixos, ou no inglês *Dual Axis Solar Tracker (DAST)*, em comparação aos sistemas fixos de geração de energia solar. A pesquisa foi desenvolvida na região de Ural na Rússia, localidade esta tem por características índices baixos de temperaturas e incidência solar diárias por cerca de 15 horas, o que faz ser muito promissor o estudo da energia solar nessa região.

Para o desenvolvimento da pesquisa, os autores utilizaram uma placa fotovoltaica, motores DC e sensores responsáveis por receber, processar e transmitir os sinais. Montaram um protótipo, como mostrado na Figura 3, que foi utilizado para ambos os tipos de coletas, seja em modo fixo ou rastreando a posição solar.

Os pesquisadores desenvolveram algoritmos que eram capazes de interpretar a condição climática e posicionar a placa de forma a garantir uma melhor eficiência. Como no artigo de Nurzhigit Kuttybay et al. [2019] mencionado anteriormente, nessa pesquisa os autores também incluíram na programação a condição de manter a placa na horizontal em dias com muitas nuvens ou nublados, ocasiões onde não há incidência solar direta com a placa, e, com isso, poderiam obter melhores resultados mesmo em condições adversas. Dentro da programação, os

pesquisadores também estipularam que a leitura dos sensores seria feita a cada 30 minutos, uma vez que o sol não se movimenta de forma rápida, os sensores poderiam localizar a nova posição e assim fazer com o que o sistema se reajustasse.

Figura 3: Protótipo de rastreador solar



Fonte: Performance Comparison of Dual Axis Solar Tracker with Static Solar System in Ural Region of Russia, Adven Masih et. al [2019].

A coleta dos dados foi feita durante o mês de agosto de 2019, durante todo o período de incidência solar, das 6 da manhã até às 21 da noite, e por fim, foram levantadas médias diárias para que as informações pudessem ser melhor estudadas. Com isso, ficou comprovada a maior eficácia do sistema com o rastreador solar, tanto em dias com muitas nuvens quanto em dias limpos, ou seja, somente com incidência solar na maior parte do dia, com cerca de 86,52% de superioridade se comparada ao sistema fixo.

É proveitoso que utilizemos da pesquisa citada acima a condição de fazer com que o sistema se movimente a cada 30 minutos, assim podemos poupar energia e poupar desgastes desnecessários. Vale também mencionar que, os resultados obtidos pelos pesquisadores só foram possíveis devido incidência solar ocorrer por mais de 12 horas, condição esta que não se aplica a nossa região.

## 2.5 Effect of Temperature on the I-V Characteristics of a Polycrystalline Solar Cell

O trabalho realizado por Indra Karki [2015] tem por objetivo fazer uma análise de eficiência das placas fotovoltaicas de silício policristalino quando submetidas a temperaturas elevadas. Módulos fabricados com este tipo de material são comumente mais utilizados por possuírem um menor preço.

Para realizar o seu experimento, primeiramente, Indra esfriou as placas à uma temperatura abaixo da temperatura ambiente, resultando em aproximadamente 10°C. Com isso, poderiam ser estabelecidos gráficos com parâmetros mais abrangentes, que demonstrassem a performance das placas indo de temperaturas baixas a temperaturas mais elevadas.

A pesquisa foi realizada durante um dia inteiro, que se alternou entre ensolarado a com poucas nuvens. Para a coleta e observação de dados, o autor utilizou dois piranômetros, que foram dispostos no mesmo plano onde as placas estavam, para medir a radiação solar. Utilizou também termopares para medir a temperatura das placas. Um *data logger* foi utilizado para fazer o armazenamento da radiação e da temperatura, informações estas vindas dos piranômetros e termopares respectivamente. As placas solares utilizadas foram de silício policristalino, além de um computador para que os dados fossem trabalhados.

O autor utilizou-se também de um software da *Daystar* para traçar curvas comparativas, seja corrente x temperatura ou tensão x temperatura. Este software é próprio para mensurar dados provenientes de sistemas com placas solares, além de fazer comparações diretas, pode também transferir os resultados para sejam retrabalhados da melhor maneira.

Finalizada a coleta e o trabalho com os dados, o autor concluir que, com o aumento da temperatura, a corrente de curto circuito, a corrente máxima do módulo fotovoltaico, aumentou linearmente seguindo o acréscimo da temperatura. Em outras palavras, quando há uma maior incidência de luz, há um aumento considerável de corrente pelo módulo. Entretanto, a tensão de circuito aberto sofre um decréscimo também relevante com a elevação da temperatura. Dessa forma, há a necessidade de estipular bem a utilização desse tipo de placa fotovoltaica, uma vez que a corrente e a tensão sofrem essas alterações, pode ser prejudicial para o sistema em que sejam aplicadas.

O Fator de Forma (FF), que é proporcional ao pico de potência real sobre o pico nominal, também sofre um decréscimo oriundo da alteração das grandezas medidas. Esse fator é importante pois determina a qualidade dos módulos fotovoltaicos e, de acordo com o mostrado por Indra, as placas utilizadas nesta pesquisa não correspondem bem a alteração de temperatura

uma vez que o fator sofre essa queda e, mesmo que isso não cause grande impacto na eficiência, é importante frisar que essas características podem ser úteis na hora de montarmos um projeto.

Podemos utilizar da pesquisa de Indra o aprendizado sobre o comportamento da eficiência das placas solares do tipo de silício policristalino, haja vista que uma placa semelhante às utilizadas por ele será utilizada no nosso trabalho. Importante notar também que as temperaturas na cidade de Manaus são comumente elevadas, então saber que vamos coletar dados provavelmente já abaixo da eficiência nominal dos módulos será útil para o desenvolvimento desta pesquisa.

### 3. REFERENCIAL TEÓRICO

#### 3.1 A Energia Solar

Hoje, umas das fontes limpas de energia mais utilizada ao redor do mundo, a energia solar tem sua história iniciada por meados dos anos de 1800. O físico francês Alexandre Edmond Becquerel (1820-1891) foi o primeiro a conseguir realizar, acidentalmente, o efeito fotovoltaico quando expôs ao sol duas placas de latão depois de submersas em eletrólito líquido. Anos depois, em 1954, com estudos voltados para a absorção mais eficiente da luz solar, cientistas desenvolveram a primeira placa feita a base de silício, essa possuindo 6% de eficiência, sendo assim, superior aos materiais anteriormente utilizados (Machado; Miranda, 2015).

Com a elevação da demanda de energia devido ao aumento populacional, a geração por energia solar vem em ascensão gradativa e o Brasil tem um grande potencial para esse ramo, uma vez que o país apresenta um dos maiores índices de irradiação solar do mundo, destacando-se por ter grande parte de seu território próximo à linha do Equador. Um dos sistemas mais utilizados para a geração da energia solar, tanto no Brasil quanto ao redor do mundo, é o Sistema Solar Fotovoltaico (Barbosa Filho et al., 2015).

#### 3.2 Sistema Solar Fotovoltaico

Esse tipo de sistema pode ser elaborado, principalmente, de duas formas, pela montagem de painéis ou módulos solares. O principal material para o seu desenvolvimento é o silício, este podendo ser Monocristalino, que é o tipo mais eficiente, Policristalino, uma vez que comparado ao Mono perde no quesito eficiência, porém é mais barato, e, por último, Amorfo ou Filme fino, este geralmente mais utilizado em pequenas aplicações, como calculadoras de bolso, por exemplo (CARVALHO, 2014).

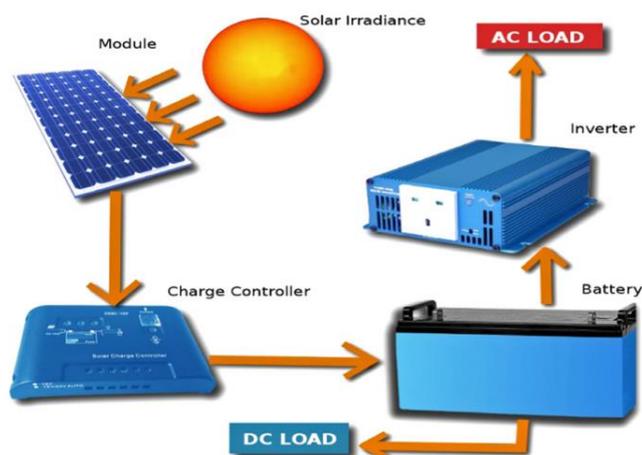
O efeito fotovoltaico ocorre quando há incidência solar sobre as placas, os fótons emitidos pelo sol são absorvidos pelas células fotovoltaicas, onde a energia proveniente dos fótons é transferida para os elétrons que, com isso, passam a se movimentar gerando corrente elétrica.

Existem dois tipos de aplicações para os sistemas fotovoltaicos, são eles: Sistemas Isolados, conhecidos também como *Off-Grid*, e Sistemas Conectados à Rede, *On-Grid* ou *Grid-tie*. Os isolados são mais comuns em sistemas de iluminação, casas de campo onde o custo para

que se tenha acesso à rede elétrica é muito elevado e em pequenas aplicações. Já os *On-grid* são comumente vistos nas cidades, estes complementam a energia gerado pela rede tradicional (SCHERER et al., 2015).

A Figura 4 demonstra um sistema fotovoltaico *Off-Grid* completo, desde a absorção da energia solar até o direcionamento da energia.

Figura 4: Sistema fotovoltaico



Fonte: Design and economics analysis of an off-grid PV system for household electrification, Ghafoor et al. [2014]

### 3.2.1 Componentes Básicos de um Sistema Solar Fotovoltaico

Para a montagem de um sistema solar fotovoltaico existem 4 itens que são necessários: o painel solar; um controlador; uma bateria e o inversor de frequência.

#### 3.2.1.1 Placas Fotovoltaicas

Como mencionado anteriormente, os painéis mais comuns são feitos à base de silício Monocristalino (m-SI), Policristalino (p-SI) ou Filme Fino (CdTe), dos quais o Monocristalino se destaca por ter mais eficiência garantida em comparação aos outros, porém possui um preço de mercado maior. A Figura 5 exemplifica, visualmente, os formatos de diversos tipos de placas. O Policristalino tem por característica perder rendimento quando há um alto índice de temperatura, mas ainda assim, garante boa eficiência e seu preço de mercado é inferior se comparado ao Monocristalino (Karki, 2015). Já os de Filme Fino, são os mais baratos, porém tendem a se degradar mais rapidamente.

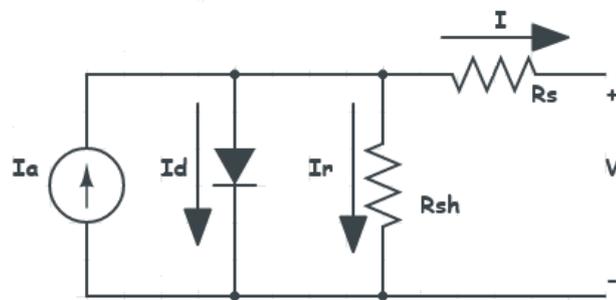
Figura 5: Tipos de módulos fotovoltaicos



Fonte: Analysis of the Conversion Efficiency of Five Types of Photovoltaic Modules During High Relative Humidity Time Periods, Burduhos et al. [2018]

A eficiência de uma placa solar não é determinada exclusivamente pelo material da qual é feita, as suas dimensões, localidade e posicionamento são importantes aspectos que contribuem para que os módulos possam funcionar com um alto índice de eficiência. Além disso, existem equações para determinar os seus parâmetros e, para que isso seja possível, é necessária a idealização de um circuito equivalente ao funcionamento das placas, o circuito é mostrado na Figura 6.

Figura 6: Circuito equivalente às placas solares



Fonte: autor

Onde  $I_a$  é a corrente gerada pelo painel solar;  $I_d$  a corrente no diodo;  $I_r$  é a corrente na resistência;  $I$  é a corrente de saída do módulo e o  $V$  a tensão de saída. As resistências,  $R_s$  e  $R_{sh}$ , são a resistência própria do material, em série com o circuito, e a resistência de *Shunt*, em paralelo, respectivamente, componente este importante para a melhor medição dos parâmetros.

Com o circuito mostrado anteriormente, é possível estipular que a corrente de saída,  $I$ , será fornecida pela equação:

$$I = I_a - I_d - I_r \quad (1)$$

A partir dessa equação, sabendo que a corrente no diodo pode ser determinada por:

$$I_d = I_o \left( e^{\frac{q(V+iR_s)}{KT}} - 1 \right) \quad (2)$$

E que a corrente no resistor de *Shunt* é dada por:

$$I_r = \frac{(V+iR_s)}{R_{sh}} \quad (3)$$

Sendo assim, substituindo (3) e (2) em (1) temos que a corrente fornecida pelo módulo será:

$$I = I_a - I_o \left( e^{\frac{q(V+iR_s)}{KT}} - 1 \right) - \frac{(V+iR_s)}{R_{sh}} \quad (4)$$

Onde  $I_o$  é a corrente de saturação do diodo;  $q$  é a carga do elétron ( $1,6 \times 10^{-19}$  C);  $i$  é a corrente no módulo;  $K$  é a constante de Boltzmann ( $1,38 \times 10^{-23}$  J/C) e  $T$  é a temperatura do módulo. (Kabalci et al., 2015)

Com esta equação podemos determinar, no caso ideal, o quanto de corrente elétrica pode ser fornecida pelos módulos fotovoltaicos e podemos fazer comparações com a obtenção real.

Além disso, a posição em que se montam as placas em relação ao sol contribui grandemente para o aumento da eficiência de geração de energia. Podemos calcular a radiação captada pela placa através da seguinte equação:

$$E = E_{max} \cos(Alt) \cos\left(\frac{\pi}{2} - Azi\right) \quad (5)$$

Temos que  $E$  é a radiação coletada pelas placas;  $E_{max}$  é a radiação máxima que pode ser captada;  $Alt$  é o ângulo de altitude e  $Azi$  é o ângulo azimutal (Salama e Taha 2018).

Por essa equação, note-se que o posicionamento das placas, no eixo de altitude e azimute, influencia diretamente na eficiência das placas solares, uma vez que a geração depende da recepção da radiação emitida pelo sol, quanto maior a incidência dos fótons solares, maior será a eficiência das placas.

A máxima eficiência pode também, em casos ideais, ser determinada por meio da seguinte equação.

$$Ef = \frac{P_{max}}{P_{in}} = \frac{P_{max}}{C \times A} \quad (6)$$

Onde  $C$  é a constante solar, aproximadamente  $1,353 \text{ kW/m}^2$  e  $A$  é a área da superfície de recepção das placas. Além destes, temos que a potência máxima de saída,  $P_{max}$ , pode ser determinada por:

$$P_{max} = Voc \times Isc \times F.F. \quad (7)$$

$Voc$  sendo a tensão máxima que pode ser fornecida pela placa, tensão de circuito aberto;  $Isc$  é a corrente máxima de saída dos módulos, corrente de curto circuito; o Fator de Forma,  $F.F.$ , é a proporção entre a potência máxima gerada pela placa e a potência nominal, seu valor pode se alterar entre 0 a 1, e é um dos fatores que testificam a eficácia de uma placa (Salama; Taha, 2018).

### 3.2.1.2 Controlador

O controlador de carga, exemplificado na Figura 7, é o dispositivo responsável por impedir a sobrecarga do sistema, bem como da bateria. Quando não há um controlador presente, a vida útil da bateria tende a diminuir consideravelmente uma vez que os módulos solares não possuem meios de determinar o momento em que a bateria atingiu a carga máxima.

Figura 7: Controlador de carga



Fonte: EPEVER, Epsolar Technology – Datasheet LandStar/E series

Esse dispositivo possui 3 principais funcionalidades: a proteção de carga, quando detectasse que a bateria está com sua carga máxima, o controlador restringe a conexão do módulo com a bateria evitando a sobrecarga; proteção de descarga excessiva, este funciona quando há um baixo nível de carga sendo fornecida, interrompendo o consumo de energia; gerenciamento da carga da bateria, alguns tipos de controladores permitem que seja feita a regulagem de alimentação na bateria (CARVALHO, 2014).

### 3.2.1.3 Bateria

A bateria tem por função armazenar a energia gerada pelo sistema para que assim seja utilizada futuramente. É de grande importância para sistema dos quais é exigido fornecimento elétrico durante a noite, quando não há sol. Ainda podem suportar cargas e descargas, fazendo com que a sua reutilização seja possível e, se mantida em condições adequadas, possuem uma garantia longa de vida.

Em sua grande maioria, são constituídas por elementos tóxicos, como o cádmio e chumbo, porém existem as feitas à base de chumbo-ácido, estas mais adequadas para os sistemas fotovoltaicos (Machado; Miranda 2015). A Figura 8 traz um exemplo de baterias estacionárias.

Figura 8: Bateria estacionária



Fonte: <https://www.heliar.com.br/pt-br/produtos/estacionaria/heliar-freedom>

As baterias podem energizar dispositivos que funcionam com corrente contínua (CC) diretamente, como microcontroladores, por exemplo, mas como a grande maioria das aplicações exigem corrente alternada (CA), faz-se necessária a utilização de um inversor de frequência.

### 3.2.1.4 Inversor de Frequência

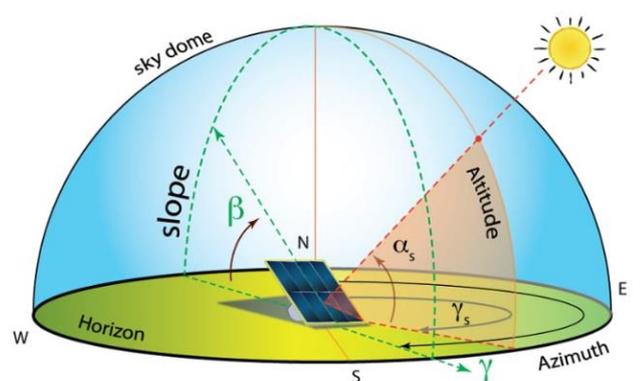
Os inversores são os responsáveis por transformar em corrente alternada a energia que é fornecida pela bateria em forma de corrente contínua, assim podendo energizar eletrodomésticos e etc, além de ampliar a tensão de saída, uma vez que as baterias, geralmente, só são capazes de fornecer até 12V (Almeida et al., 2016). A utilização de um inversor pode ser descartada quando o sistema fotovoltaico não prevê energizar dispositivos que necessitem de corrente alternada ou altas tensões, exemplos disso são os sistemas de iluminação pública, microcontroladores e etc.

## 3.3 Rastreador Solar

Na intenção de aumentar a eficiência de um módulo solar, diversos trabalhos ao redor do mundo são focados na pesquisa e desenvolvimento de rastreadores solares. Muitos desses estudos concluem que existe vantagem quando possuímos uma placa solar acoplada a um sistema capaz de rastrear o posicionamento do sol (Masih; Odinaev, 2019).

A técnica mais comum utilizada para que o sistema siga o sol é a técnica de *Maximum Power Point Tracking* (MPPT), esta visa manter o módulo fotovoltaico perpendicular ao sol o máximo de tempo possível. Por intermédio de sensores fotorresistores que percebem a posição momentânea do sol e de motores que movimentam as estruturas, o sistema é capaz de se movimentar em dois graus de liberdade fazendo com que a placa permaneça posicionada perpendicularmente ao sol.

Figura 9: Eixos de movimentação referenciados ao sol



Fonte: <http://biblus.accasoftware.com/ptb/desenho-de-uma-usina-fotovoltaica/>

A utilização de sistemas capazes de se movimentar em dois graus de liberdade é o, comprovadamente, método com melhores resultados, pois assim permite com que os módulos se ajustem da melhor forma à receber a radiação solar durante a maior parte, se não toda, do dia (Salama; Taha, 2018).

Os eixos de altitude e o azimutal, mostrados na Figura 9, são os principais a serem levados em consideração nesse tipo de sistema, dessa forma o sistema poderá acompanhar o posicionamento do sol independente da hora do dia ou época do ano.

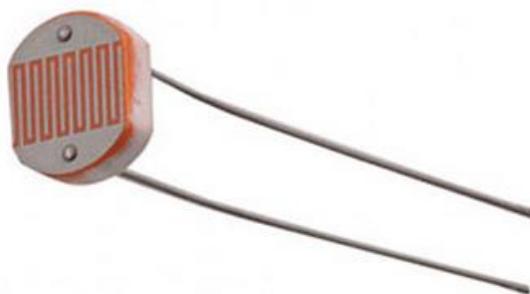
### 3.4 Sensores Fotorresistores (*Light Dependent Resistor - LDR*)

Esse tipo de sensor altera a sua resistência de acordo com a incidência de luz em sua superfície, quanto maior a luminosidade incidente, menor a resistência, isso se deve ao fato de possuir, em sua parte superior, um material semiconductor com uma alta resistência elétrica. A Figura 10 mostra um sensor fotorresistor e fica evidente que a sua superfície possui um formato próprio para a recepção da luz.

Em geral, esses sensores são utilizados em série com outro resistor comum, montando um divisor de tensão, em casos onde se faz necessário o controle da tensão que sobre o LDR. São muito utilizados em sistemas de iluminação pública, sistemas estes que detectam a ausência de luz e, com isso, acionam lâmpadas e etc.

A resistência desses sensores pode variar de  $100k\Omega$ , em ambientes mais escuros, até  $10k\Omega$ , em locais com maior iluminação. Sua aplicação é mais comum somada a microcontroladores, pois estes trabalham com tensões não muito elevadas (McRoberts, 2018).

Figura 10: Sensor Fotorresistor



Fonte: <https://components101.com/lldr-datasheet>

### 3.5 Microcontroladores

O microcontrolador é um circuito integrado (CI) que possui um microprocessador e alguns outros componentes que o permitem se assemelhar a microcomputadores, podendo armazenar e processar dados. É um componente extremamente utilizado em sistemas embarcados, onde há autonomia de funcionamento, mas com funcionalidades predefinidas.

Possuem portas de entradas e saídas (I/O), temporizadores, conversor digital analógico e etc., dessa forma ampliando a sua gama de aplicações, estas que vão desde sistemas pequenos até grandes controles de dados e informações.

Por serem baratos, são comumente utilizados em módulos de prototipagem, estes que auxiliam no ensino de automação ou controle em escolas, tendo como o mais comum o Arduino. Os microcontroladores utilizados pelos Arduinos são da marca *Atmel*, que hoje pertence a empresa Microchip. (da Silva; Araujo; Cavalcante, 2019). A Figura 11 mostra um microcontrolador modelo ATmega328P desenvolvido pela Microchip.

Figura 11: Microcontrolador



Fonte: <https://www.microchip.com/wwwproducts/en/atmega328p>

#### 3.5.1 Arduino

Quando existe a necessidade de controlarmos um processo, é necessário que tenhamos a possibilidade de recebermos os sinais de comando e tratá-los da forma desejada. Sendo assim, trabalharmos somente com os microcontroladores de forma isolado torna qualquer projeto mais difícil. Para isso existem os módulos de prototipagem, dos quais o mais comum é o Arduino.

Como mostrado na Figura 12, este módulo possui diversas entradas e saídas (I/O) que, por si só, podem acionar LEDs, motores, sensores, interruptores e etc.

O Arduino possui a sua própria IDE (*Integrated Development Environment*), uma plataforma onde podemos programar os códigos responsáveis por comandar as suas funcionalidades. A programação do Arduino, por padrão, deve ser feita linguagem C/C++, porém também há suporte pela IDE para desenvolvimento de códigos em Java (da Silva; Araujo; Cavalcante, 2019).

Figura 12: Kit de prototipagem Arduino



Fonte: <https://store.arduino.cc/usa/arduino-uno-rev3>

Existem vários modelos, tanto originais quanto clones, de Arduino e por possuir fonte aberta, é muito comum que periféricos não originais funcionem normalmente com os originais. O módulo mais utilizado ultimamente é o Arduino UNO, este possui um chip USB (*Universal Serial Bus*) que facilita a comunicação com computadores se comparado à outros modelos mais antigos.

Além das entradas e saídas disponíveis de fábrica nos módulos, existe a possibilidade de conectarmos *Shields* ao Arduino a fim de estendermos as suas funcionalidades. Os *Shields* são placas de circuito que contém outros dispositivos capazes de auxiliar em alguma funcionalidade, podem ser módulos de Ethernet, receptores GPS e etc. Caso não haja a possibilidade de utilizar um *Shield*, um protoboard pode facilmente substituí-lo, uma vez que com ele diversos circuitos podem ser elaborados (McRoberts, 2018).

Os códigos desenvolvidos para o Arduino são chamados de *Sketchs* (rascunho ou esboço), possuem um padrão em sua estrutura que para a elaboração de um programa funcional são necessários dois comandos básicos, o *setup()*, este responsável por habilitar as portas que

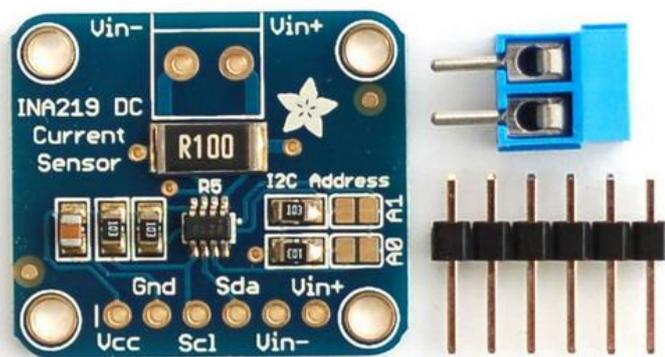
serão utilizadas no processo, e o *loop()*, que é a função capaz de repetir uma camada de instruções e aguardar até que alguma ação seja tomada (da Silva; Araujo; Cavalcante, 2019).

### 3.5.1.1 Arduino data logger

Um sistema *data logger* é capaz de registrar dados por um determinado período de tempo ou enquanto existir memória de armazenamento disponível. Uma das aplicações do Arduino é o monitoramento de sistemas, porém, para isso, faz-se necessária a utilização de módulos capazes de ler os dados e de armazená-los. Um periférico muito utilizado é o módulo INA219, Figura 13, este é capaz de fazer a leitura de corrente, tensão e potência.

Este módulo é próprio para leitura de corrente contínua e possui uma calibração que pode ser programada. Seu funcionamento ocorre com uma tensão que pode variar entre 3 a 5.5 V e é capaz de medir correntes de até 3,5 A. Além disso, possui comunicação do tipo I<sup>2</sup>C, esta permite com que façamos associação de vários sensores na mesma entrada do Arduino, um sensor será o mestre, responsável por direcionar as informações, e os demais serão os escravos. (Texas Instruments 2008)

Figura 13: Sensor de corrente, tensão e potência INA219



Fonte: <https://learn.adafruit.com/adafruit-ina219-current-sensor-breakout>

## 3.6 Servo Motor

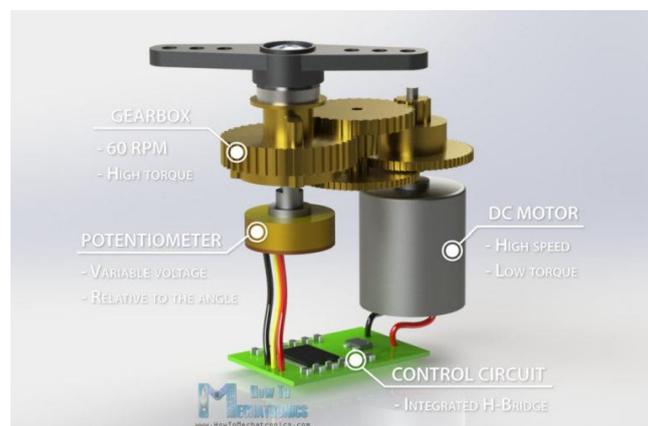
Um dos tipos de motores mais utilizados no mercado, o servo motor tem como uma de suas principais características a precisão nos movimentos e, além disso, é possível regular a sua

velocidade e torque. Um servo motor pode ser robusto ou pequeno, dependendo da sua aplicação.

Os servos motores convencionais pequenos, geralmente utilizados com projetos de Arduino, possuem um sistema interno relativamente simples, a figura 14 mostra um servo motor internamente. Por possuir engrenagens que distribuem o torque, é possível que, ainda que pequenos, estes servos suportem grandes cargas. (Tong 2014)

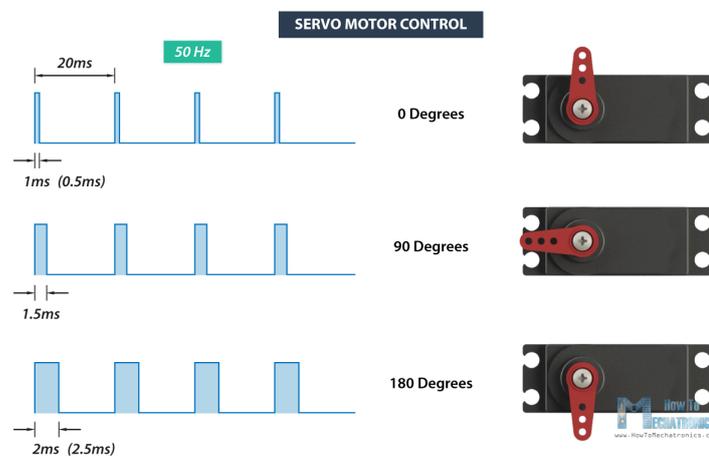
A alimentação elétrica dos servos comumente necessita ser externa, principalmente nos casos onde as cargas exercidas sobre os motores são elevadas.

Figura 14: Servo motor internamente



Fonte: <https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>

Figura 15: Controle de posição por PWM



Fonte: <https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>

Os servos de pequeno porte geralmente são alimentados com corrente DC (*direct current*), têm os sinais PWM (*Pulse Width Modulation*) como seus controladores de posição e sua velocidade depende da potência fornecida, por conta disso muitos servos possuem potenciômetros em sua estrutura. A figura 15 exemplifica o posicionamento dos servos de acordo com o sinal de entrada.

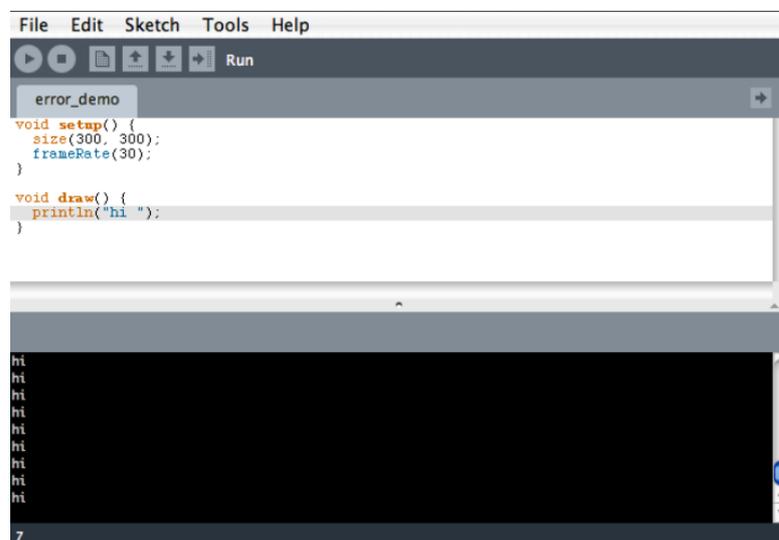
Os sinais PWM podem ser fornecidos pelo Arduino como comandos por código, este possui saídas digitais o que torna possível o controle preciso do posicionamento (Pinckney 2006).

### 3.7 Processing

O *Processing* é um software aberto desenvolvido especificamente para o desenvolvimento de interfaces gráficas. Por possuir métodos simples, qualquer pessoa, mesmo aquela não habituado com programação, pode desenvolver uma interface interativa e intuitiva para o seu sistema.

Por ser uma aplicação que foi desenvolvida e é programada em Java, suas aplicações são inúmeras e podem se comunicar com diversos sistemas, dentre estes o Arduino. Por conta disso, o *Processing* se torna uma ferramenta capaz de fornecer uma interface apta a controlar os comandos executados pelo Arduino e expor as grandezas medidas pelos sensores acoplados ao módulo de prototipagem (Noble 2009).

Figura 16: Script desenvolvido no *Processing*

The image shows a screenshot of the Processing IDE. At the top, there is a menu bar with 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for running, saving, and other functions. The main window is titled 'error\_demo' and contains the following code:

```
void setup() {  
  size(300, 300);  
  frameRate(30);  
}  
  
void draw() {  
  println("hi ");  
}
```

The output window at the bottom shows the text 'hi' printed repeatedly on multiple lines. The status bar at the bottom left shows the number '7'.

O *Processing* possui a sua própria IDE e os *scripts* são desenvolvidos de forma semelhante aos do Arduino. No *Processing*, a função *setup()* funciona da mesma maneira que a, também nomeada *setup()*, na IDE do Arduino, nela declaramos o tamanho da tela que será utilizada para a interface, sua coloração, além dos botões e outros objetos que estarão dispostas, ou seja, tudo o que utilizaremos na visualização e interação da interface.

Já a função *draw()*, que é semelhante a *loop()* na IDE do Arduino, tem por objetivo “desenhar” na tela os objetos que foram declarados anteriormente, nela é onde declaramos a localização exata dos componentes que queremos expor, sejam eles botões, imagens, gráficos e etc. (Noble 2009). A figura 16 mostra um *script* de exemplo desenvolvido no *Processing*.

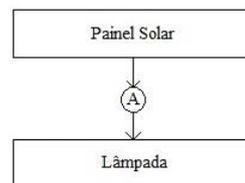
#### 4. METODOLOGIA

O sistema proposto para este projeto foi dividido em duas partes, uma direcionada a geração de energia solar e a outra responsável pelo controle dos movimentos da estrutura e armazenamento dos dados a serem estudados, juntas resultaram no sistema rastreador solar.

Para a geração de energia solar, foram montados dois pequenos sistemas fotovoltaicos, ambos contendo uma pequena placa solar de 10W que foi responsável por transformar os fótons emitidos pelo sol em energia. Como a corrente gerada pelos módulos não é elevada, não se fez necessária a utilização de baterias e, conseqüentemente, nem de controladores de carga. Os módulos energizaram diretamente duas lâmpadas de 9W cada.

Os fluxogramas expostos nas figuras 17 e 18 representam, respectivamente, a montagem desses sistemas fotovoltaicos, do quais um permanecerá fixo e o outro se movimentará graças aos rastreador solar.

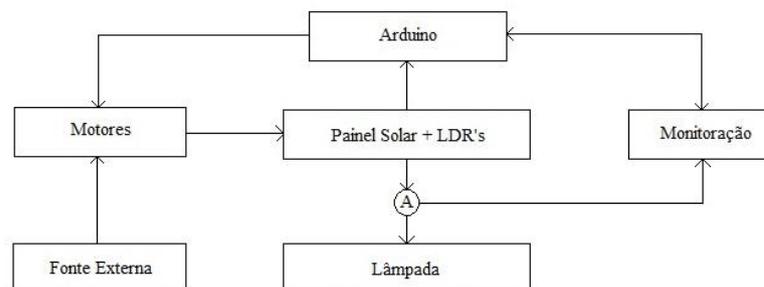
Figura 17: Fluxograma do sistema fotovoltaico fixo



Fonte: Autor

Em ambos sistemas, foram dispostos sensores de corrente e tensão, representados pela letra A, que possibilitou a mensura dessas grandezas geradas por cada módulo de forma individual.

Figura 18: Fluxograma do sistema com rastreador solar



Fonte: Autor

Como visto no fluxograma do rastreador na figura 18, temos um bloco nomeado monitoração, sendo este um sistema que supervisionou em tempo real o funcionamento, estando disposto para que fosse possível analisarmos a eficiência do módulo acoplado ao rastreador comparado ao módulo fixo. Além disso, também foi possível executar o controle manual do rastreador por intermédio de uma interface.

#### 4.1 Sistema Fotovoltaico

Neste projeto, utilizamos dois kits de energia solar adquiridos pela *internet* na loja online da NeoSolar. Ambos dispõem de um módulo fotovoltaico do tipo policristalino fabricados pela Sinosola. Segundo o fabricante, cada um dos módulos é capaz de fornecer 10W (Watts) de potência máxima, 22,5V em circuito aberto e uma corrente de curto circuito de 0,6A (Amperes). Além disso, possui dimensões aproximadas de 35cm de comprimento, 25cm de altura e 1.8cm de profundidade. A figura 19 mostra uma das placas que foram utilizadas nesse projeto.

Figura 19: Módulo fotovoltaico de 10W



Fonte: Autor

Como mencionado anteriormente, as correntes geradas pelos módulos não são altas e, sendo assim, ambas foram ligadas diretamente a lâmpadas de 9W. A conexão foi feita diretamente das placas por intermédio de fios condutores.

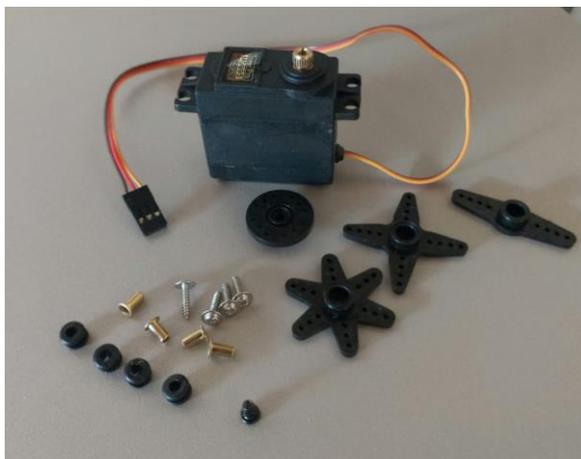
## 4.2 Rastreador Solar

O Rastreador Solar é um sistema que pode ser dividido em três etapas para a sua montagem: motores, a estrutura e o controle.

### 4.2.1 Motores

Para que pudéssemos exercer movimentos nos eixos azimutal e altitude, utilizamos dois servomotores neste projeto. Segundo o *datasheet*, os MG995 Tower Pro, figura 20, são servos com rotação de até 180°, tensão de operação na faixa de 4.8V a 7.2V e podem exercer um torque de 13kg.cm (4,8V) a 15kg.cm (6V) (MPJA [s.d.]). O torque dos motores foi suficiente para suportar as cargas exercidas sobre os mesmos.

Figura 20: Servomotor utilizado no projeto



Fonte: Autor

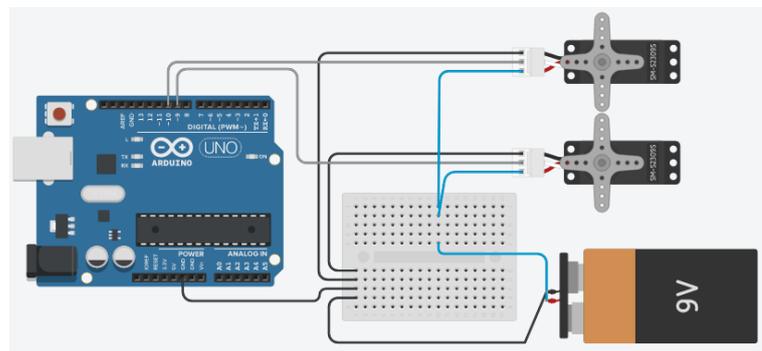
Como os servos exigem uma tensão consideravelmente alta se comparados a circuitos de pequeno porte energizado pelo próprio Arduino, fez-se necessária a utilização de uma alimentação externa para ambos os motores. Uma bateria de 9V é suficiente para fazer a energização dos servos. Ligados entre si em paralelo, a tensão fornecida pela bateria foi suficiente para que os motores exercessem o máximo de sua capacidade de torque, uma vez que a bateria não forneceu precisamente os 9V nominais. Um botão foi inserido no circuito dos motores somente para controlar a energização, pois, caso fosse necessário fazer algum reajuste na programação dos servos, poderíamos assim desenergizar os mesmos e poupar energia.

Os sinais digitais PWM, responsáveis por alterar as posições dos eixos dos servomotores, serão provenientes do Arduino, este que possui 6 portas digitais capazes de enviar sinais deste tipo, sendo mais do que suficiente para o comando dos servos que utilizamos.

Haja vista que nesse projeto utilizamos servos com alimentação externa e os sinais de comandos são oriundos do Arduino, o negativo destes componentes, tanto motores quanto da bateria, necessitam estar no mesmo nó do aterramento do Arduino para que não haja interferência de sinal (Geddes 2017).

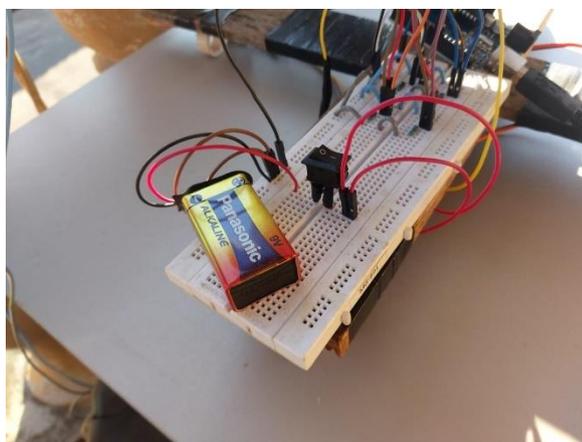
As figuras 21 e 22 mostram, respectivamente, os circuitos dos servo motores projetados no *Tinkercad* e um trecho do circuito final montado no rastreador solar.

Figura 21: Circuito dos servomotores simulado no Tinkercad



Fonte: autor

Figura 22: Trecho do circuito real dos motores



Fonte: Autor

#### 4.2.2 Estrutura

Para iniciar a montagem da nossa estrutura, primeiro fez-se necessária a aquisição de um suporte capaz de comportar um servomotor, este que será especificado mais a frente e fixado na parte superior do suporte, e todos os circuitos responsáveis pelo funcionamento do sistema.

Uma base feita em MDF (*Medium Density Fiberboard*, ou em português, placa de fibra de média densidade) com dimensões de 50cm de comprimento, 50cm de largura e 10cm de altura, em formato de gaveta sem parte dianteira e traseira, porém fechada na parte superior, foi adquirida para a montagem.

Figura 23: Base do protótipo do rastreador

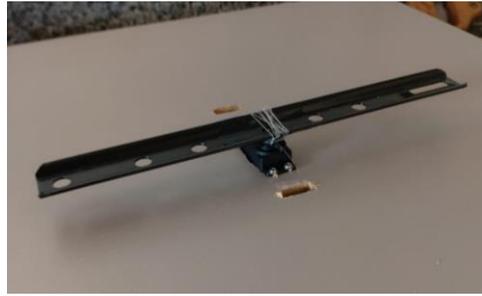


Fonte: Autor

A figura 23 exibe a base utilizada com 3 furos na sua parte superior. Um furo foi feito para que um dos servomotores fosse inserido, este ficando responsável por exercer o movimento no eixo azimutal do rastreador. Os outros dois furos foram feitos para que as fiações dos sensores que utilizamos pudessem acessar o protoboard que foi colocado na parte interna da base.

No eixo do servomotor, fixou-se, com auxílio de linhas de nylon e fita isolante, uma estrutura metálica com cerca de 25cm de comprimento e 3cm de largura, e sobre esta, também com auxílio de fita isolante, uma peça de madeira que possuía 50cm de comprimento e 7cm de largura, como mostrado nas figuras 24 e 25.

Figura 24: Peça de metal acoplada ao servomotor



Fonte: Autor

Figura 25: Peça de madeira posta sobre a de metal



Fonte: Autor

Nas duas extremidades da madeira, como visto na figura 25, foram fixadas, por intermédio de pregos de 1 polegada, peças menores de madeira com 2cm de comprimento, 5cm de largura e 1cm de altura. Essas peças tiveram por objetivo auxiliar na fixação de duas peças de canos PVC (*Polyvinyl chloride*, que em português significa Policloreto de polivilina), estas com 4,5cm de diâmetro e 23cm de altura. A peça de cano da direita foi fixada numa posição que permita a passagem dos cabos do servomotor que ainda serão inseridos.

Nas pontas superiores de cada cano, foram feitas aberturas nas laterais, sendo uma voltada para a outra. No cano que suportará o servomotor, a abertura foi feita seguindo as medidas do próprio servo a fim de que este ficasse bem fixado. Por outro lado, no cano da outra extremidade, a abertura foi feita dimensionada com base em outro cano que será responsável por suportar o módulo, este será o eixo de movimentação da placa solar. A figura 26 mostra a estrutura quase finalizada.

Figura 26: Estrutura sem o módulo e sem a peça com LDR's



Fonte: Autor

Fixado ao eixo do servomotor acoplado ao cano, encontra-se um outro cano de 1,5cm de diâmetro e aproximadamente 40cm de comprimento. A sua fixação ao servo foi possível graças a um prego de 1 polegada inserido em sua outra extremidade e este amarrado ao eixo do servomotor por fios de nylon. O prego inserido na outra ponta do cano também serviu como ferramenta para impedir que o eixo saísse de posição.

Neste cano foram fixadas duas outras peças de cano de 3.5cm de diâmetro e 33cm de comprimento, estas cortadas em formato de calha. Um corte mais fundo foi feito em ambas as peças a fim de que estas sejam capazes de fixar um dos módulos fotovoltaicos. Estes canos foram fixados de forma a ficarem com uma angulação de 45° em referência ao solo.

A figura 26 também mostra uma peça de madeira que foi fixada junto à outra madeira acoplada ao servomotor da base. Esta, que possui 7cm de largura, 35cm de comprimento e 1cm de altura, foi inserida para evitar que a estrutura balançasse quando os movimentos fossem exercidos. A figura 27 mostra a estrutura com um dos módulos inseridos nela.

Figura 27: Estrutura com módulo inserido



Fonte: Autor

### 4.2.3 Sistema de Controle

Para o nosso projeto, utilizamos dois módulos de prototipagem Arduino do tipo UNO R3. A utilização de dois fez-se necessária por conta de utilizarmos o protocolo de comunicação Firmata em um dos módulos, assim ocupando constantemente a porta serial, e utilizarmos sensores de corrente que também necessitam transmitir informações de forma serial.

Por conta disso, utilizamos um Arduino embarcado com um *script* pronto da biblioteca Firmata, focado somente no controle dos servos e leitura dos sensores fotorresistivos e outro responsável por tratar as informações dos sensores de corrente, este levando um *script* desenvolvido especificamente para a leitura das grandezas medidas.

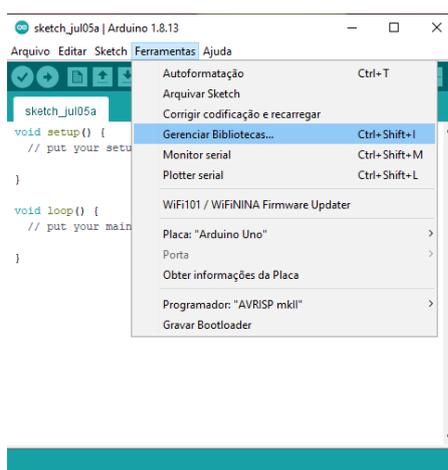
#### 4.2.3.1 *Sketchs* dos Arduinos

O protocolo de comunicação Firmata que utilizamos em um dos módulos Arduino é uma biblioteca que permite transmitir o controle total das portas, tanto analógicas quanto digitais, e funções do Arduino para qualquer sistema capaz de exercer esse controle, seja através de uma interface ou qualquer outro tipo de sistema desenvolvido para essa finalidade, e que esteja ligado ao módulo de prototipagem por algum meio de comunicação, nesse caso o serial.

O *software* que utilizamos para a criação da interface e, conseqüentemente, controle do Arduino foi o *Processing*, ferramenta que será explicada mais a frente nessa pesquisa.

Para instalarmos a biblioteca Firmata, utilizamos a própria IDE do Arduino por meio do gerenciamento de bibliotecas. A figura 28 mostra como chegar ao menu de gerenciamento.

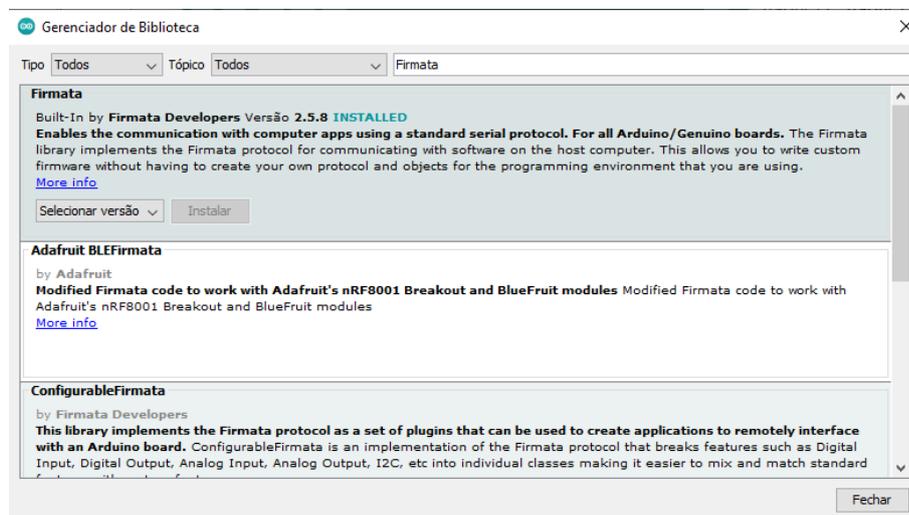
Figura 28: Caminho para gerenciar bibliotecas do Arduino



Fonte: Autor

Após clicar no menu “Gerenciar Bibliotecas...”, basta inserir o nome da biblioteca, nesse caso Firmata, no campo de pesquisa como mostrado na figura 29. Em seguida, clicar em instalar. Após esse procedimento, a biblioteca já está pronta para ser utilizada.

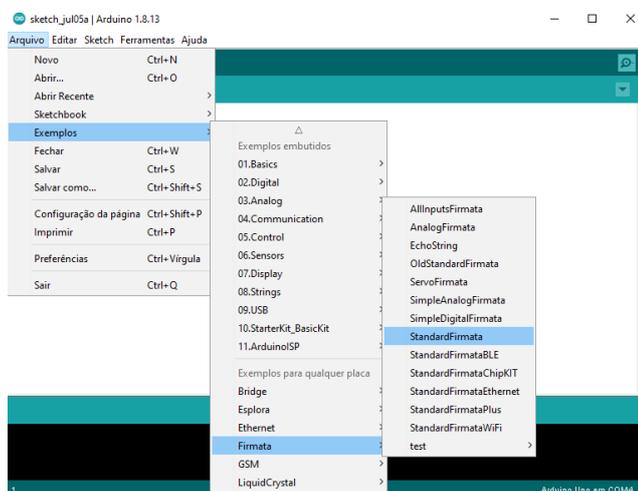
Figura 29: Instalação da biblioteca Firmata



Fonte: autor

Instalada a biblioteca, precisamos somente embarcar no Arduino um dos *scripts* exemplo chamado *StandardFirmata* já disponível no momento em que instalamos a biblioteca. A figura 30 mostra o caminho para incluirmos o *script* no *sketch* do Arduino.

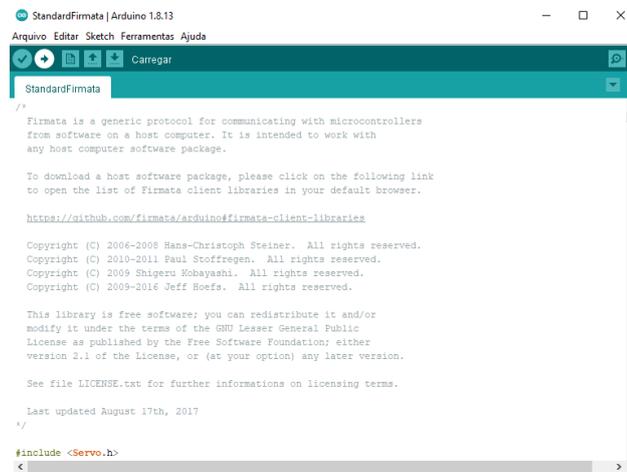
Figura 30: Caminho para incluir o *script* *StandardFirmata*



Fonte: autor

Nenhuma alteração no *sketch* foi necessária haja vista que o código inserido já transfere todo o controle do Arduino para o outro sistema ligado ao computador. Então, apenas carregamos o *sketch* no Arduino, figura 31, este ocupando a porta serial COM4.

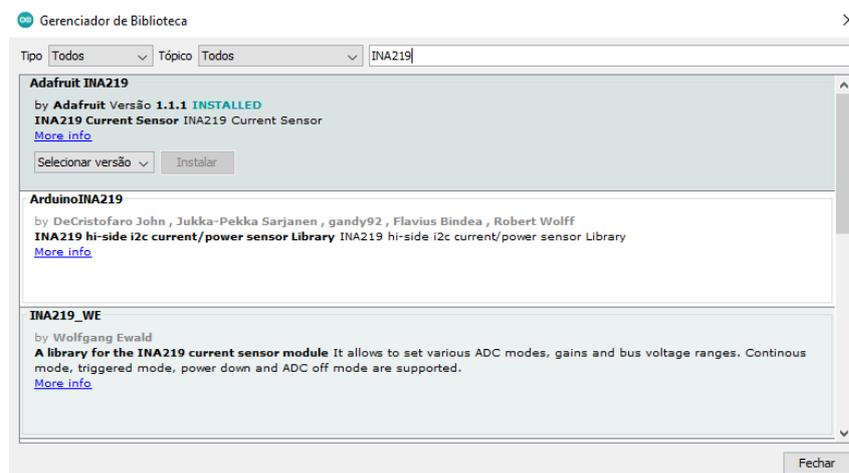
Figura 31: Carregando a *StandardFirmata* no Arduino



Fonte: autor

Por outro lado, o Arduino responsável por tratar as informações dos sensores de corrente utilizados não utilizou um *sketch* já pronto, porém foi necessário instalarmos a biblioteca própria dos sensores utilizados, os INA219. A figura 32 mostra o caminho para a instalação da biblioteca *Adafruit INA219*.

Figura 32: Instalação biblioteca Adafruit INA219



Fonte: autor

Como já explicado anteriormente, o Arduino responsável pelo tratamento dos INA219 precisou ocupar exclusivamente uma porta serial, a COM5, para poder transmitir as informações para o *Processing* através da comunicação serial.

O *sketch* desenvolvido para o Arduino dos sensores está exposto na figura 33. O endereçamento dos sensores foi determinado de acordo com as especificações dos mesmos. Tendo em vista a necessidade de considerar um deles o mestre e outro o escravo, requisitos para a comunicação I2C, um dos sensores, este com solda aplicada no endereço A0, foi declarado com o endereço 0x41. Sendo assim, podemos utilizar os dois sensores de forma simultânea no mesmo *sketch*.

Dentro da função *loop()* inserimos duas funções *delays*. A primeira ocorre para que o *Processing* tenha tempo suficiente para ler a informação que o Arduino transmitiu de forma serial, método esse que ocorre graças a função *Serial.print()*.

O outro *delay* foi inserido para que houvessem leituras de corrente somente a cada 5 minutos, dessa forma não sobrecarregaremos o sistema com dados repetidos.

Figura 33: Sketch do Arduino para os sensores de corrente e tensão

```
#include <Wire.h> //Biblioteca responsável por permitir a comunicação I2C
#include <Adafruit_INA219.h> //Biblioteca dos sensores

Adafruit_INA219 ina219_A; //Declaração do sensor mestre
Adafruit_INA219 ina219_B(0x41); //Declaração do sensor escravo com endereçamento (0x41)

void setup(void)
{
  Serial.begin(9600);
  ina219_A.begin(); //Energiza o sensor mestre (Endereço padrão 0x40)
  ina219_B.begin(); //Energiza o sensor escravo com o endereço 0x41

  ina219_A.setCalibration_32V_1A(); //Calibra os sensores para tensões de até 32V e correntes de 1A
  ina219_B.setCalibration_32V_1A();
}
void loop(void)
{
  float busvoltageA = 0; //Declaração das variáveis
  float busvoltageB = 0;

  busvoltageA = ina219_A.getBusVoltage_V(); //Armazena a resultado instantâneo das tensões medidas
  busvoltageB = ina219_B.getBusVoltage_V();

  Serial.println(busvoltageA); //Imprime na porta serial a grandeza lida no respectivo sensor
  delay(500); //Espera de 500 milissegundos
  Serial.println(busvoltageB);
  delay(300000); //Espera de 5 minutos para uma nova leitura
}
```

Fonte: autor

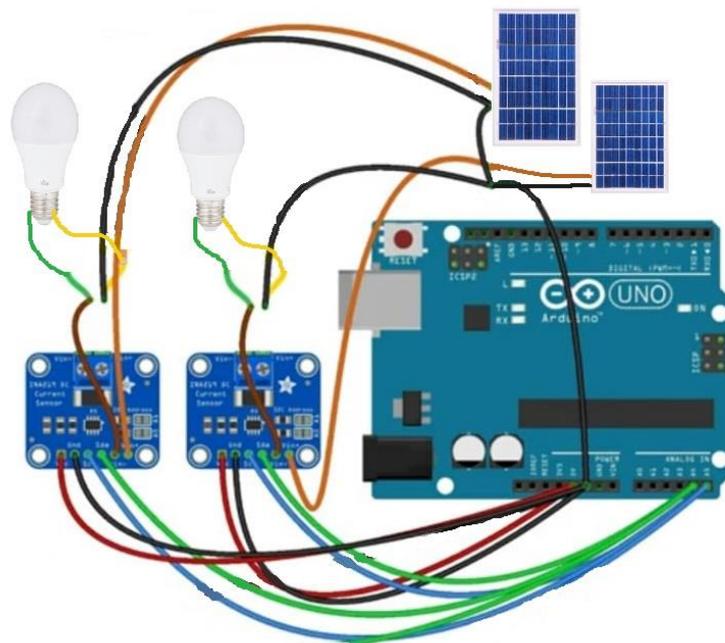
#### 4.2.3.2 INA219

Por utilizarmos módulos de baixa potência, fez-se necessária a utilização de sensores capazes de fazer a leitura adequada de correntes consideravelmente pequenas. Os INA219, da *Texas Instruments*, foram adequados para o projeto pois possuem uma alta precisão e a sua leitura permanece precisa mesmo para baixas correntes e tensões.

Esses sensores necessitam de uma alimentação que pode variar entre 3,3V a 5V, o que torna o Arduino perfeitamente capaz de energizá-los e possuem duas formas de entrada do sinal a ser medido, ambos métodos invasivos, ou seja, sendo a necessário fazer o corte nos fios. Neste projeto, o método invasivo foi adequado pois garante uma precisão maior de leitura dos dados, haja vista que não há perdas de energia.

Além disso, os módulos INA219 possuem duas portas para a comunicação I2C que, no caso deste projeto, serviu para associar os sensores, uma vez que dois foram necessários. A figura 34 mostra o circuito dos sensores projetado.

Figura 34: Circuito projetado dos sensores e Arduino



Fonte: Adaptado pelo autor

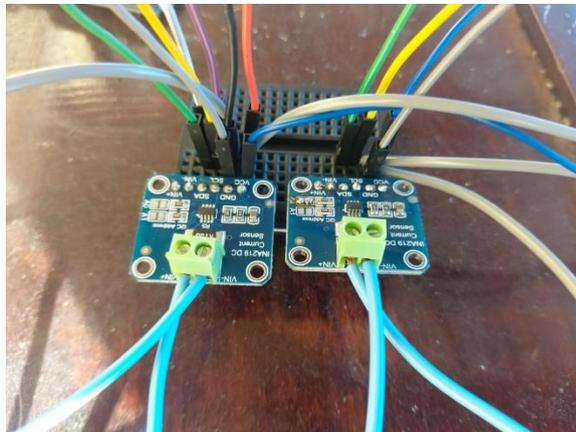
Foi necessário aplicar solda no endereço A0 de um dos sensores, desta forma esse sensor passou a ser o escravo e o outro, o mestre. Isso permite que, a partir do *sketch* desenvolvido no

Arduino mostrado na seção anterior, as leituras de ambos os sensores sejam recebidas pelas mesmas portas analógicas, que neste caso foram a A4 e A5, saídas Sda e Scl do sensor mestre respectivamente.

A figura 35 mostra os sensores montados em uma mini *proto-board* que foi adquirida justamente para esta finalidade. *Jumpers* foram utilizados para as conexões uma vez que os mesmos trazem um fácil manuseio e oferecem aos circuitos a possibilidade de reajustes rápidos.

O sensor da esquerda é o mestre e o da direita, o escravo. No sensor escravo, como mencionado anteriormente, aplicamos solda no endereço A0, dessa forma, esse sensor passou a se comunicar com o Arduino pelo endereço 0x41, respeitando os padrões estipulados pelo fabricante (Texas Instruments 2008). Já a figura 36 mostra o circuito já com as lâmpadas inclusas no mesmo.

Figura 35: Circuito dos sensores montado numa mini *proto-board*



Fonte: Autor

Figura 36: Sensores ligados às lâmpadas

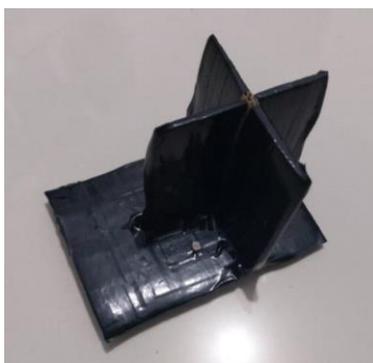


Fonte: Autor

#### 4.2.3.3 LDRs

Para que pudéssemos obter uma melhor referência da posição por meio da luminosidade, utilizamos 4 sensores fotorresistivos, LDRs. Estes ficaram dispostos na parte de cima do módulo fixado ao rastreador e centralizados, pois assim, no momento em que os sensores estivessem perpendiculares ao sol, a placa solar também estaria. Uma peça, que foi feita com papelão e revestida com fita isolante para caso pegue chuva não se desfaça, serviu como suporte para os LDRs, mostrada na figura 37.

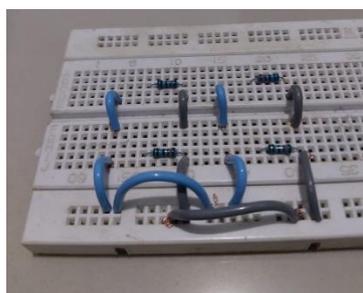
Figura 37: Peça suporte para LDR



Fonte: Autor

A peça possui este formato pois, com base em trabalhos semelhantes que dentre os quais está a pesquisa realizado por Gilberto Freitas (2018), essa mencionada no capítulo de trabalhos relacionados, podemos utilizar a alteração no valor de saída dos LDRs causados pela penumbra incidente sobre eles. Assim, os LDRs que informarem um maior valor estarão de fato com maior incidência solar, caso contrário, poderíamos ter dificuldade na montagem das condições na programação uma vez que a alteração entre os sensores não seria muito alta.

Figura 38: Circuito dos LDRs com resistores e jumpers de alimentação

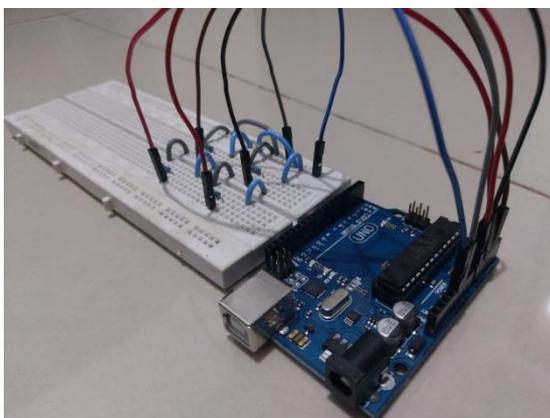


Fonte: Autor

Na montagem do circuito dos sensores fotorresistivos, fez-se necessária a utilização de 4 resistores de  $10k\Omega$  em série com cada sensor para que houvesse uma atenuação nos valores de saída e, dessa forma, pudessem ser melhor trabalhados.

A figura 38 mostra os circuitos montados de forma que sejam do tipo *pull over*, ou seja, quanto maior a incidência solar, maior a corrente de saída do circuito. Circuitos desse tipo necessitam que o terminal de maior potência da tensão, representados na figura pelos *jumpers* azuis, proveniente do Arduino, esteja ligado direto ao LDR e o terminal de menor potência, *jumpers* cinza, ligado ao resistor. A figura 39 mostra o circuito dos LDRs já com *jumpers* para transmissão do sinal de saída dos sensores para o Arduino e também os *jumpers* positivo e negativo, estes provenientes das portas de 5V do Arduino e da porta *GND* respectivamente.

Figura 39: Circuito já com jumpers de sinal e alimentação



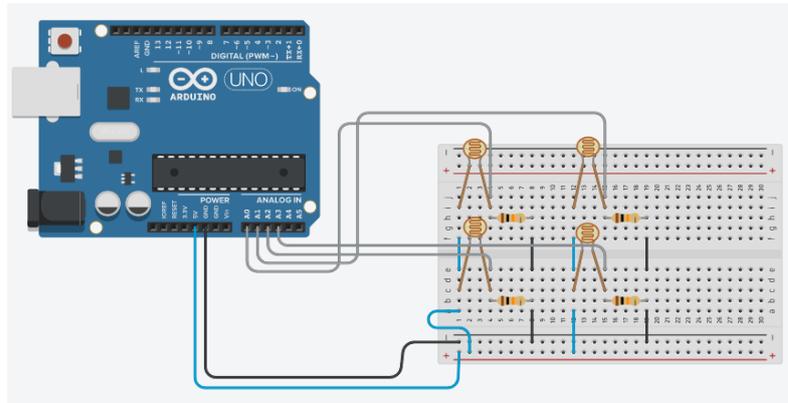
Fonte: autor

Os valores de saída de cada LDR, provenientes dos nós onde cada sensor e seus respectivos resistores se encontravam, precisaram ser ligados a uma porta analógica do Arduino cada, as portas foram as A0, essa recebendo o sinal proveniente do sensor superior do lado esquerdo, A1, que tratou o sinal do LDR superior do lado direito, A2, que por sua vez recebeu o sinal inferior esquerdo e, por último, A3, ficando responsável pelo sinal inferior direito.

As portas analógicas do Arduino possuem leitura que pode variar de 0 a 1023, no caso dos circuitos dos LDRs do tipo *pull over*, quanto mais próximo de 1023 o sinal de saída for, a tensão de saída estará mais próxima de 5V. Caso contrário, quanto menor a saída do circuito LDR, mais próximo a 0V estará a saída do circuito.

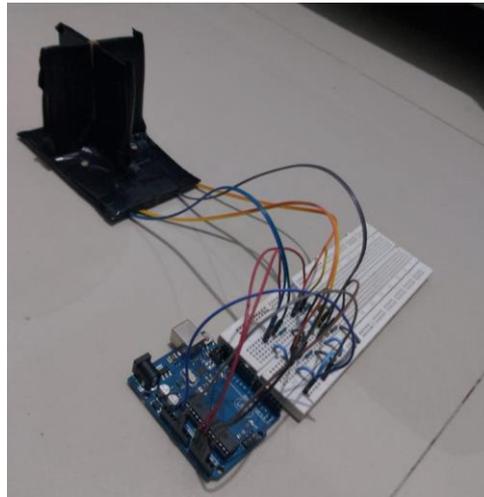
A figura 40 mostra o circuito simulado no site *Tinkercad.com*, enquanto a figura 41 exibe o circuito real já com os LDRs inseridos na peça suporte e conectados ao restante do circuito.

Figura 40: Circuito dos LDRs montado no *Tinkercad*



Fonte: autor

Figura 41: Circuito LDR real



Fonte: Autor

### 4.3 Interface de Supervisão e Controle

A interface que utilizamos nesse projeto foi desenvolvida no *Processing* e, juntamente à biblioteca Firmata, inserida no Arduino dos servos e LDRs, podemos controlar as funcionalidades do Arduino totalmente pela interface. Por ser um *script* grande, o mesmo encontra-se em sua íntegra no apêndice A.

Uma das partes mais importantes do *script* é o trecho de *import* das bibliotecas utilizadas. A figura 42 mostra quais bibliotecas utilizamos na nossa interface. As bibliotecas *controlP5* e *g4p\_controls* nos permitem inserir botões e mecanismos de controle para que possamos definir condições, como por exemplo, se o rastreador funcionará de forma automática ou manual.

Figura 42: Bibliotecas utilizadas no Processing

```

1 //Importação das bibliotecas utilizadas
2 import controlP5.*;
3 import g4p_controls.*;
4 import processing.serial.*;
5 import grafica.*;
6 import cc.arduino.*;

```

Fonte: Autor

As demais bibliotecas, *processing.serial*, *grafica* e *cc.arduino* servem, respectivamente, para permitir a comunicação serial do *Processing* com o Arduino, gerar os gráficos e fazer com que o *Processing* controle as funcionalidades do Arduino.

Um exemplo de controle do *Processing* sobre o Arduino se encontra na figura 43, nela estipulamos os pinos que serão responsáveis por enviar os sinais PWM para os servos e também posicionamos os motores nas posições iniciais, esse procedimento ocorre em todas as vezes que o sistema é inicializado para que assim o posicionamento seja iniciado a partir das posições iniciais dos eixos.

Figura 43: Código de declaração dos pinos e posições dos servos

```

84 //Declarando os pinos dos Arduino e posicionamento inicial dos servos
85 arduino.pinMode(pin_azimute, Arduino.SERVO);
86 arduino.pinMode(pin_altitude, Arduino.SERVO);
87 arduino.servoWrite(pin_azimute, 0);
88 arduino.servoWrite(pin_altitude, 0);

```

Fonte: Autor

Como mencionado anteriormente nessa seção, o sistema poderá funcionar de duas formas: manual ou automático. Para o modo manual utilizamos um controle em formato de *joystick*. Esse controle possui 4 posições possíveis, estas executadas pelo usuário da interface.

Cada posição, cima, baixo, esquerda ou direita, transmite um sinal PWM para o respectivo servo motor.

Nos casos de posicionamento do *joystick* para cima ou para baixo, o servo que recebeu os sinais foi o disposto para o posicionamento do eixo de altitude. Por outro lado, quando houver os posicionamentos para direita ou esquerda, o servo que se movimenta é o do eixo azimutal. A figura 44 mostra uma parte do *script* em que tratamos os posicionamentos do *joystick*.

Figura 44: Trecho do código dos posicionamentos do joystick

```

180   if (stickX < 0){
181     if (azimute_pos < 180){
182       azimute_pos+=1;
183       arduino.servoWrite(pin_azimute, azimute_pos);
184       delay(delay_servo);
185     }
186   }
187
188   if (stickY > 0){
189     if (altitude_pos > 0){
190       altitude_pos-=1;
191       arduino.servoWrite(pin_altitude, altitude_pos);
192       delay(delay_servo);
193     }
194   }

```

Fonte: Autor

Quando o usuário optar por utilizar o modo automático, o *Processing* trabalhará com as informações vindas dos LDRs. Para que pudéssemos estipular o posicionamento solar através do sinal recebido, desenvolvemos o *script* que tem parte mostrado na figura 45. As comparações entre os LDRs foram feitas com base nas médias obtidas, por exemplo, no momento em que a média superior dos LDRs for maior que a inferior, o *Processing* entenderá que o sol está posicionado acima da posição atual do sistema e, com isso, acionará o servo do eixo de altitude e este se movimentará até que esteja posicionado perpendicularmente ao sol.

Podemos observar na figura 45 a utilização de uma tolerância nos momentos dos cálculos. Esta medida se dá pelo fato de que durante o desenvolvimento percebemos que quando o sol estava em uma posição diferente das que correspondiam diretamente às medias, ou seja, posicionado ao nordeste, noroeste, sudoeste ou sudeste, a precisão não era satisfatória. Para resolver isso utilizamos a tolerância, dessa forma os cálculos se fizeram mais precisos e o posicionamento do rastreador mais eficaz.

Figura 45: Parte do código de controle manual

```

223 //Declaração da tolerância
224 float tol = 50;
225
226 //Se a média dos LDRs superior for maior que a inferior somado à tolerância
227 if (avg_up > avg_down + tol){
228
229     //Se a altitude for menor que 90°, acresce no eixo de altitude
230     if (altitude_pos < 90){
231         altitude_pos+=1;
232         arduino.servoWrite(pin_altitude, altitude_pos);
233         delay(delay_servo);
234     }
235
236 //Se a média dos LDRs da esquerda for maior que a da direita somado à tolerância
237 }else if (avg_left > avg_right + tol){
238
239     //Se a posição azimutal for maior que 0°, decresce no eixo azimutal
240     if (azimute_pos > 0){
241         azimute_pos-=1;
242         arduino.servoWrite(pin_azimute, azimute_pos);
243         delay(delay_servo);
244     }

```

Fonte: Autor

A interface mostra graficamente e em tempo real as tensões medidas em cada um dos sistemas de forma individual. Os gráficos são atualizados a cada 5 minutos e reiniciados a cada hora, dessa forma podemos ter uma visualização mais adequada dos gráficos.

Figura 46: Trecho do código para armazenar os dados lidos na porta serial

```

114 //Inicia a leitura da portal serial, informação enviada pelo Arduino dos sensores de tensão
115 if (port.available() > 0 ) {
116     String value = port.readString();
117     if ( value != null ) {
118         num = float(value);
119         if (aux == 0){
120             //Armazena no documento a hora, minuto e o dado lido
121             dataReport.print(hour());
122             dataReport.print(":");
123             dataReport.print(minute());
124             dataReport.print("-> Tensão Rastreador: ");
125             dataReport.println( num );
126             aux = 1;
127             voltage_automatic = num;
128             delay(500);
129             data = 1;
130         } else if (aux == 1){
131             dataReport.print(hour());
132             dataReport.print(":");
133             dataReport.print(minute());
134             dataReport.print("-> Tensão Fixo: ");
135             dataReport.println( num );
136             voltage_fixed = num;
137             aux = 0;
138         }
139     }
140 }

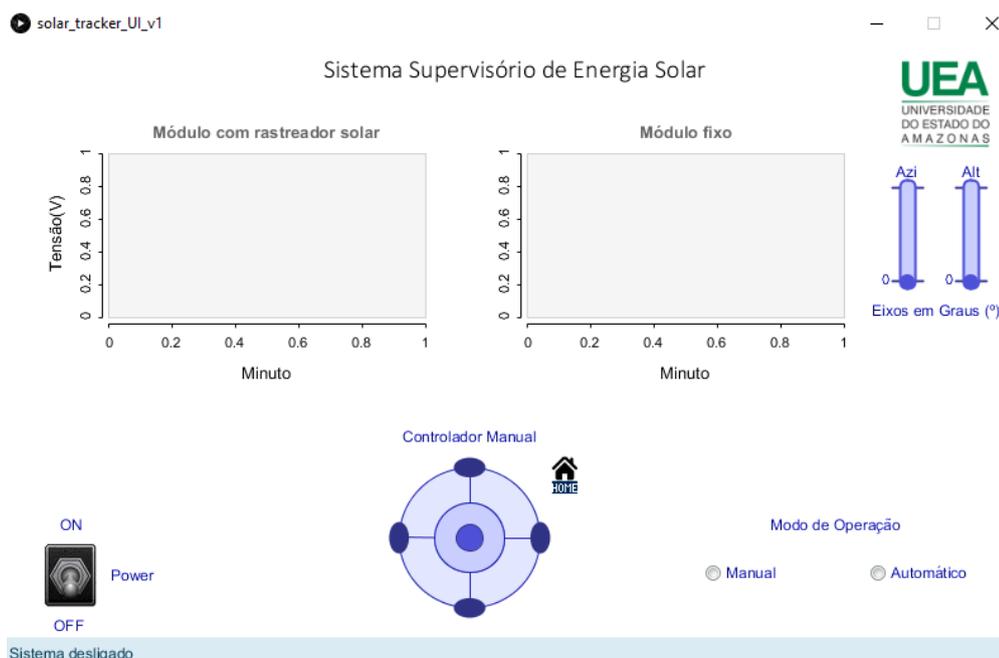
```

Fonte: Autor

Além disso, o sistema registra todos os dados lidos e armazena os mesmo em um documento externo. Os dados armazenados foram necessários para montarmos os gráficos utilizados nos resultados desta pesquisa. A figura 46 mostra o trecho do código que armazena os dados no relatório.

A figura 47 mostra a interface final projetada ainda sem os dados nos gráficos pois estes dependem do funcionamento do sistema como um todo.

Figura 47: Interface final do sistema supervisorio



Fonte: Autor

## 5. RESULTADOS E DISCUSSÕES

As coletas de dados foram feitas nos dias 10 e 11 de julho de 2021. Este capítulo mostrará os resultados que foram obtidos nessa pesquisa e a instalação dos sistemas fixo e rastreador.

### 5.1 Instalação dos Sistemas Fixo e Rastreador

Os sistemas foram instalados na cidade de Manaus, Amazonas, mais especificamente no bairro Lírio do Vale 2, localizado na zona Centro Oeste da cidade. Ambos permaneceram posicionados próximos durante toda a coleta dos dados, dessa forma, quando houvesse incidência sobre um, necessariamente também ocorreria sobre o outro. A figura 48 mostra os sistemas instalados.

Figura 48: Sistemas instalados para coleta



Fonte: Autor

O sistema fixo, mostrado na figura 49, foi instalado orientado ao Norte e angulado em aproximadamente 3° (graus), angulação esta referente a distância aproximada em graus da cidade de Manaus à linha do Equador. Esse método de instalação dos sistemas fixos segue estudos que identificam que quando os sistemas estão posicionados seguindo essas condições,

orientação ao Norte e angulação de acordo com a latitude, possuem uma eficiência mais elevada e evita sombreamento (Alonso 2016).

Figura 49: Sistema fixo de coleta de energia solar



Fonte: Autor

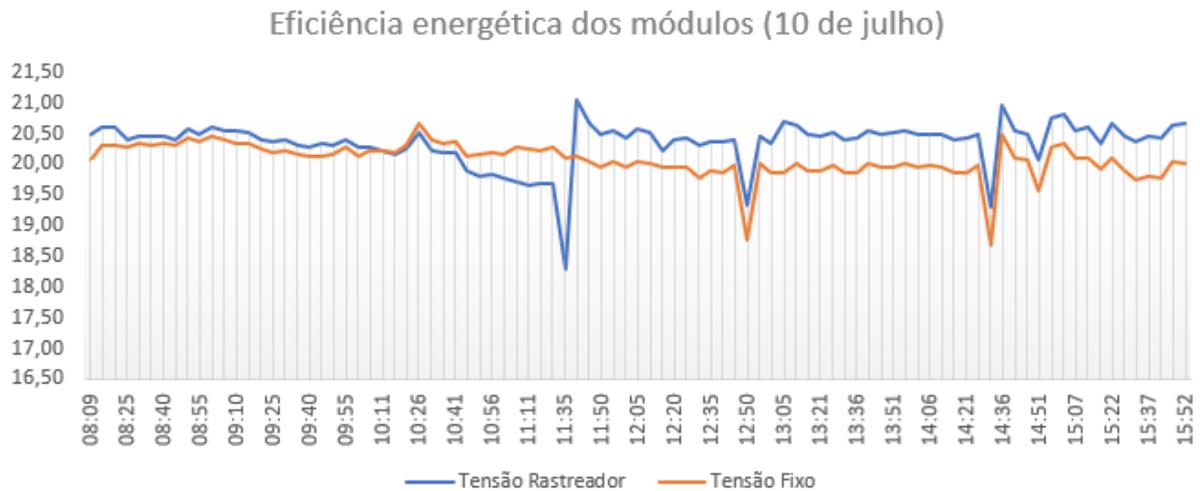
As coletas foram realizadas entre os dias 10 e 11 de julho de 2021. Em ambos os dias as análises iniciaram por volta de 8 horas da manhã. O dia 10 foi completamente ensolarado e com pouquíssimas nuvens, permitindo assim uma análise mais eficiente dos módulos. Dessa forma, as medições foram possíveis até às 16 horas sem interrupções causadas pelo clima. A partir desse horário fortes ventos passaram a incidir sobre o rastreador o que prejudicou a continuação da coleta.

Já no dia 11, a coleta precisou ser interrompida diversas vezes por conta de chuva e fortes ventos. Por grande parte da manhã, nuvens encobriram o sol, o que impossibilitou uma análise mais completa durante o dia. A partir das 10 horas, chuvas iniciaram seguidas de ventos que causaram dificuldade na continuação da análise. Dessa forma, a coleta foi interrompida no dia 11 por volta das 14 horas.

## 5.2 Primeiro dia de coleta de dados

Com os dados obtidos no dia 10 de julho, podemos criar um gráfico para uma melhor visualização. A figura 50 mostra o gráfico resultante das medidas obtidas no primeiro dia de coleta.

Figura 50: Gráfico da eficiência dos módulos no primeiro dia

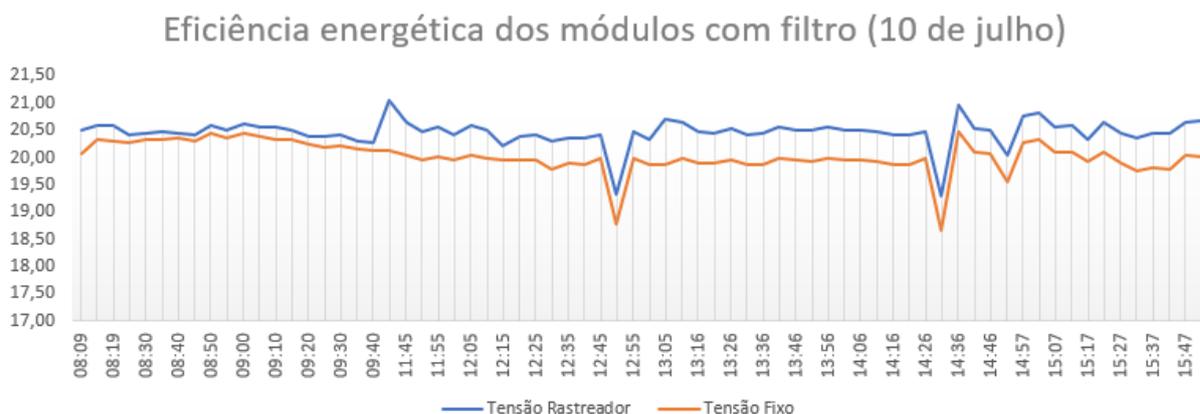


Fonte: Autor

Pelo gráfico exposto, destaco um trecho entre às 9:40 e 11:50. Nesse específico intervalo observamos uma grande perda de eficiência no módulo acoplado ao rastreador. Isso se deu ao fato de que o rastreador estava em modo automático de operação e o sistema não foi capaz seguir o sol, um problema na energização dos motores causou essa falha.

A partir de 11:50, o sistema foi alterado para o controle manual. Assim, tivemos que movimentar periodicamente o sistema para que o mesmo se encontrasse perpendicular ao sol durante o período de coleta.

Figura 51: Gráfico dos dados coletados no dia 10 sem o intervalo onde houve erro nas medições



Fonte: Autor

Se desconsiderarmos esse erro podemos gerar um outro gráfico para analisar a eficiência dos módulos enquanto estavam em pleno funcionamento. A figura 51 mostra o gráfico sem os dados coletados no intervalo de 9:45 à 11:40.

Com base no gráfico sem os dados do intervalo onde houve o erro, podemos observar que o módulo acoplado ao rastreador se manteve mais eficiente do que o módulo fixo durante todo o dia. Nas primeiras horas do dia observamos que a eficiência não foi tão superior, o módulo fixo se manteve eficiente de forma a quase se equiparar ao acoplado ao rastreador. Ainda assim, ao chegar ao meio dia, vemos que a eficiência do módulo com o rastreador se manteve consideravelmente superior ao fixo. Essa característica se manteve, como observado no gráfico, durante todo o restante das medições.

Vemos que os valores de tensão ficaram alternando entre 19V e 20V em ambos os módulos. A tensão de curto circuito dos módulos utilizados é de 22,5V, como mencionado na seção Metodologia. A proximidade da tensão mensurada à tensão de circuito aberto se dá pelo fato de que as lâmpadas que utilizamos para fecharmos os circuitos não eram apropriadas, as tensões para o funcionamento das mesmas eram maiores do que as fornecidas por cada uma das placas solares, porém isso não afeta a análise de eficiência dos módulos uma vez que eles se mantêm gerando energia independente disso.

A tabela 1 mostra as grandezas medidas no intervalo de 12:00 às 13:00, intervalo este onde o módulo com o rastreador passa a fornecer uma energia maior que o módulo fixo e se mantém assim no decorrer do dia.

Tabela 1: Dados medidos no intervalo de 12:00 às 13:00

| Hora  | Tensão Rastreador (V) | Tensão Fixo (V) |
|-------|-----------------------|-----------------|
| 12:00 | 20,42                 | 19,96           |
| 12:05 | 20,58                 | 20,05           |
| 12:10 | 20,52                 | 20,00           |
| 12:15 | 20,22                 | 19,96           |
| 12:20 | 20,40                 | 19,96           |
| 12:25 | 20,42                 | 19,95           |
| 12:30 | 20,31                 | 19,79           |
| 12:35 | 20,36                 | 19,89           |
| 12:40 | 20,37                 | 19,86           |
| 12:45 | 20,41                 | 19,98           |
| 12:50 | 19,34                 | 18,78           |
| 12:55 | 20,47                 | 20,00           |
| 13:00 | 20,33                 | 19,86           |

Fonte: Autor

As medições expostas no horário de 12:50 são menores se comparadas às restantes pois o sol se manteve coberto por uma nuvem durante esse horário.

De modo geral, podemos calcular médias das medidas com base nos dados armazenados para melhor análise. Duas médias foram estabelecidas, uma relacionada ao dia completo, contanto com o intervalo de erro na leitura, e outra média sem o intervalo onde houve erro na medição. A tabela 2 traz as médias obtidas.

Tabela 2: Médias das tensões medidas no primeiro dia

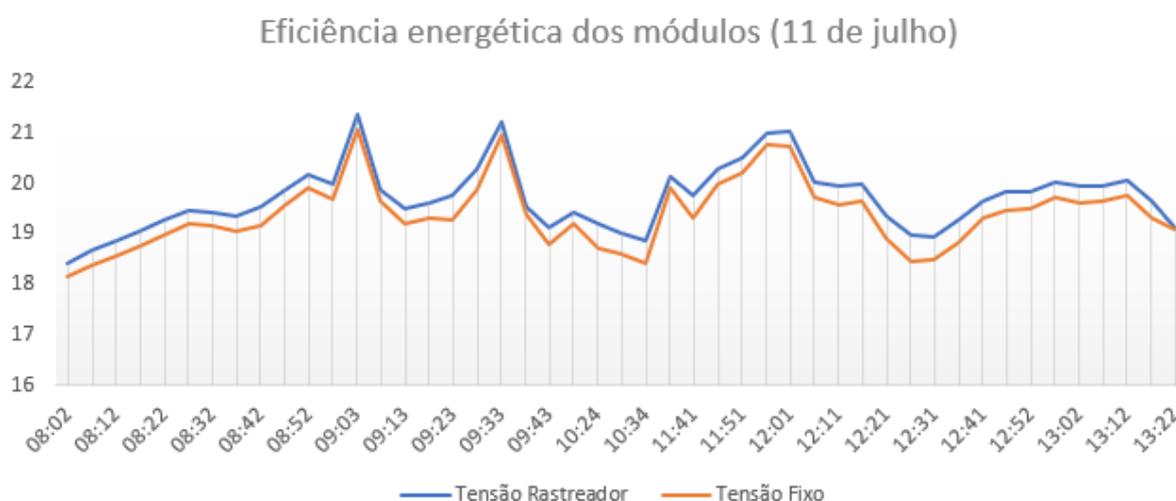
|                     | Tensão Rastreador (V) | Tensão Fixo (V) |
|---------------------|-----------------------|-----------------|
| Média total         | 20,36                 | 20,08           |
| Média sem intervalo | 20,46                 | 20,02           |

Fonte: Autor

Observamos que mesmo sem o intervalo onde houve o erro, as médias das tensões do módulo com o rastreador se mantiveram maiores que às médias do módulo fixo, porém não foi um ganho consideravelmente grande. Se analisarmos percentualmente, a média total da tensão no rastreador foi de aproximadamente 1.3% maior que a do módulo fixo, enquanto a média sem o intervalo de erro foi 2.2% superior.

### 5.3 Segundo dia de coleta de dados

Figura 52: Gráfico de eficiência dos módulos no dia 11



Fonte: Autor

No segundo dia, o clima foi diferente se comparado ao primeiro. O sol permaneceu encoberto por nuvens durante quase toda a manhã com breves exceções. A figura 52 mostra o gráfico de eficiência com base nos dados coletados durante o período do tempo em que foi possível realizar a coleta.

Com base no gráfico do dia 11, vemos que a geração de energia foi consideravelmente baixa na primeira hora de medição, por conta das nuvens que encobriam o sol, as tensões medidas foram baixas se comparadas às medidas no dia anterior.

Ainda assim, houve um ganho de eficiência no módulo junto ao rastreador que, podemos observar, permaneceu durante todo o período analisado. Fortes chuvas e ventos impediram de fazermos uma coleta mais duradoura e de forma constante durante o dia, pois a estrutura do rastreador não suportava essas circunstâncias adversas. Contudo, a tabela 3 mostra os dados coletados no intervalo de 12:01 à 13:02, horário este que o sol tende a estar posicionado bem acima dos módulos.

Tabela 3: Dados medidos no intervalo de 12:01 à 13:02 no segundo dia

| Hora  | Tensão Rastreador (V) | Tensão Fixo (V) |
|-------|-----------------------|-----------------|
| 12:01 | 21,04                 | 20,73           |
| 12:06 | 20,01                 | 19,74           |
| 12:11 | 19,96                 | 19,59           |
| 12:16 | 20                    | 19,66           |
| 12:21 | 19,37                 | 18,92           |
| 12:26 | 18,98                 | 18,46           |
| 12:31 | 18,96                 | 18,5            |
| 12:36 | 19,27                 | 18,83           |
| 12:41 | 19,66                 | 19,32           |
| 12:46 | 19,82                 | 19,48           |
| 12:52 | 19,84                 | 19,49           |
| 12:57 | 20,04                 | 19,72           |
| 13:02 | 19,94                 | 19,6            |

Fonte: Autor

Pela tabela 3, podemos ver que a tensão gerada pelo módulo com o rastreador se manteve superior em todo o intervalo especificado, garantindo assim a melhora na eficiência do mesmo ainda que sob circunstâncias climáticas adversas.

A tabela 4 traz as médias estabelecidas no segundo dia de acordo com todos os dados armazenados.

Tabela 4: Médias das tensões medidas no segundo dia

|       | Tensão Rastreador (V) | Tensão Fixo (V) |
|-------|-----------------------|-----------------|
| Média | 19,71                 | 19,38           |

Fonte: Autor

Vemos que a média do módulo do rastreador foi superior em 1.6% à média do fixo, houve um ganho de melhoria, porém não foi elevado. Ainda assim, podemos dizer que o rastreador solar torna o módulo mais eficiência mesmo quando não há incidência solar direta, quando o sol está coberto por nuvens.

## 6. CONSIDERAÇÕES FINAIS

Este projeto teve como principal objetivo o desenvolvimento de um sistema capaz de se movimentar de acordo com o posicionamento do sol e, somado a um módulo fotovoltaico, estudamos a melhoria de eficiência desse módulo tendo como parâmetro de comparação a eficiência de um outro módulo, com as mesmas características, porém este esteve fixo.

Durante o desenvolvimento do rastreador, diversos testes foram executados sobre o mesmo para garantirmos o funcionamento em ambos os modos de funcionamento. Ainda assim, vimos que durante a coleta o modo automático apresentou falhas, o que prejudicou o pleno funcionamento do sistema. Tal erro ocorreu pois, depois de um certo período de tempo energizados, os servos motores perdiam a referência, problema este que pode ter sido causado pela não energização adequada dos mesmos, uma tensão menor do que a necessária para o amplo funcionamento era fornecida ou também por conta de alguma conexão ter se soltado ou perdido o contato direto, isto podendo ter sido causado pelos fortes ventos que colidiram sobre a estrutura. Esse comportamento não pôde ter sido observado durante os testes uma vez que os tempos de utilização durante os experimentos não foram tão longos.

Ainda assim, a eficiência que o módulo acoplado ao rastreador mostrou foi superior ao módulo fixo, porém, nos dois dias analisados, essa superioridade não se mostrou tão alta. Observamos também que, mesmo com condições climáticas adversas, nuvens encobrando o sol e até mesmo com chuva, o módulo do rastreador se manteve mais eficiente.

Contudo, os gastos necessários para o desenvolvimento dessa pesquisa, para a aquisição dos módulos por exemplo, não se justificaram, uma vez que a melhoria não foi tão superior de modo que houvesse um retorno no investimento. Se analisarmos a longo prazo, os 2.2% de melhoria encontrado no primeiro dia e os 1.3% no segundo não compensam o investimento na montagem do rastreador, haja vista que o gasto seria bem maior por necessitar de motores, estrutura e etc, se compararmos ao investimento em um sistema fixo.

### 6.1 Trabalhos Futuros

Destaco alguns pontos que podem ser levados em consideração para trabalhos futuros baseados na temática desenvolvida nesta pesquisa. São eles:

- Utilizar um algoritmo capaz de identificar o posicionamento do sol pela internet, dessa forma não se faria necessária a utilização dos sensores fotorresistivos o que tornaria o

sistema mais autônomo e capaz de adaptar o seu posicionamento em qualquer condição climática;

- Desenvolver um rastreador com uma estrutura mais firme, capaz de suportar ventos e chuvas;
- Utilizar módulos fotovoltaicos com maiores potência e eficiência nominais, pois assim as grandezas medidas no módulo do rastreador podem ser consideravelmente melhores;
- Realizar as medições durante toda a incidência solar do dia, desde as primeiras horas do dia até o pôr do sol;
- Utilizar as correntes geradas pelos módulos como parâmetro de análise de eficiência.

## REFERÊNCIAS

- Almeida, Eliane, A Rosa, FCLS DIAS, K Braz, L Lana, O Santo, e T Sacramento. 2016. “Energia solar fotovoltaica: revisão bibliográfica”. *Artigo científico*, in: <http://www2.aneel.gov.br/arquivos/pdf/atlas3ed.pdf>, Outubro.
- Alonso, Rafael Herrero. 2016. “Posicionamento eficiente de módulos fotovoltaicos em plantas solares no ambiente urbano.” Universidade de São Paulo.
- Barbosa, Wilson Pereira, Wemerson Rocha Ferreira, Abílio César Soares de Azevedo, Antonella Lombardi Costa, e Ricardo Brant Pinheiro. 2015. “Expansão da energia solar fotovoltaica no Brasil: impactos ambientais e políticas públicas”. *Revista Gestão & Sustentabilidade Ambiental* 4: 628–42.
- Carvalho, Clázia Ramayana Freitas de. 2014. “Sistema fotovoltaico isolado: uma aplicação prática no projeto xapuri”.
- Freitas, Gilberto de Andrade. 2018. “Eficiência energética de um painel fotovoltaico com rastreador solar baseado em sensores LDR comparado a um painel em ângulo fixo”. Universidade de São Paulo.
- Geddes, Mark. 2017. *Manual de projetos do Arduino: 25 Projetos Práticos para começar*. Novatec Editora.
- Kabalci, Ersan, Alper Gorgun, Yasin Kabalci, e Goksel Gokkus. 2015. “A modelling case study of an island mode solar power plant”. In *2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, SG-29. IEEE.
- Kabir, Ehsanul, Pawan Kumar, Sandeep Kumar, Adedeji A Adelodun, e Ki-Hyun Kim. 2018. “Solar energy: Potential and future prospects”. *Renewable and Sustainable Energy Reviews* 82: 894–900.
- Karki, Indra Bahadur. 2015. “Effect of temperature on the IV characteristics of a polycrystalline solar cell”. *Journal of Nepal Physical Society* 3 (1): 35–40.
- Machado, Carolina T, e Fabio S Miranda. 2015. “Energia Solar Fotovoltaica: uma breve revisão”. *Revista virtual de química* 7 (1): 126–43.
- Masih, Adven, e Ismoil Odinaev. 2019. “Performance Comparison of Dual Axis Solar Tracker with Static Solar System in Ural Region of Russia”. In *2019 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBREIT)*, 375–78. IEEE.
- McRoberts, Michael. 2018. *Arduino básico*. Novatec Editora.
- MPJA. [s.d.]. “MG995 Datasheet(PDF) - List of Unclassified Manufacturers”. Acessado 19 de junho de 2021. <https://www.alldatasheet.com/datasheet-pdf/pdf/1132435/ETC2/MG995.html>.
- NeoSolar. [s.d.]. “NeoSolar”. Acessado 18 de junho de 2021. <https://www.neosolar.com.br/loja/>.
- Noble, Joshua. 2009. *Programming interactivity: a designer’s guide to Processing, Arduino, and OpenFrameworks*. “O’Reilly Media, Inc.”
- Pinckney, Nathaniel. 2006. “Pulse-width modulation for microcontroller servo control”. *IEEE potentials* 25 (1): 27–29.
- Portal Solar. 2019. “Amazonas quer ser potência em energia solar no País”. 2019. <https://www.portalsolar.com.br/blog-solar/energia-solar/amazonas-quer-ser-potencia-em-energia-solar-no-pais.html>.
- Rosa, Antonio Robson Oliveira Da, e Fabiano Perin Gasparin. 2016. “Panorama da energia solar fotovoltaica no Brasil”. *Revista Brasileira de Energia Solar* 7 (2): 140–47.
- Salama, Hend Abd El-monem, e Adel Taha Mohamed Taha. 2018. “Practical implementation of dual axis solar power tracking system”. In *2018 Twentieth International Middle East*

- Power Systems Conference (MEPCON)*, 446–51. IEEE.
- Sampaio, Priscila Gonçalves Vasconcelos, e Mario Orestes Aguirre González. 2017. “Photovoltaic solar energy: Conceptual framework”. *Renewable and Sustainable Energy Reviews* 74: 590–601.
- Scherer, Lara Almeida, Maria Eduarda Donatto Sessegolo, Taylana Borba Barcarolo, e Marco Antonio Ribeiro Edler. 2015. “Fonte Alternativa de Energia: energia solar”. *XX Seminário Interinstitucional de ensino, pesquisa e extensão. Universidade de cruz Alta/RS*.
- Silva, Fellipe Pereira Da, José Ivaldo Barbosa de Brito, e Gabrielle Bezerra Oliveira. 2016. “Investigação da variabilidade interanual e interdecenal do clima do estado do Amazonas”. *Ciência e Natura* 38 (3): 1426–39.
- Silva, Rogério Oliveira Da, Warley Monteiro Araujo, e Maxwell Machado Cavalcante. 2019. “Visão Geral Sobre Microcontroladores e Prototipagem com Arduino”. *Tecnologias Em Projeção* 10 (1): 36–46.
- Sinosola. 2006. “Sinosola”. 2006. <https://www.sinosola.cn/>.
- Texas Instruments. 2008. “INA219”. [www.ti.com](http://www.ti.com).
- Tong, Wei. 2014. *Mechanical design of electric motors*. CRC press.

## APÊNDICE A – *SCRIPT PROCESSING*

```

//Importação das bibliotecas utilizadas
import controlP5.*;
import g4p_controls.*;
import processing.serial.*;
import grafica.*;
import cc.arduino.*;

//Declaração das variáveis globais utilizadas
int facing = 6, pin_azimute = 9, pin_altitude = 10;
int azimute_pos = 0, altitude_pos = 0, stickX, stickY;
int delay_servo = 100;
PrintWriter dataReport;
int aux = 0, data;
float num, voltage_fixed, voltage_automatic;
PFont font;
PImage img_uea, img_home;
boolean systemON, manual = false, automatic = false;
Arduino arduino;
Serial port;
ControlP5 cp5;
GPlot plot, plot1;
GStick joystick;
GSlider sdrAzi, sdrAlt;
GImageToggleButton btnToggle0, btnToggle1;
GLabel labelPower, labelOperation, labelON, labelOFF, labelCon, labelMan, labelOUT,
labelGraus, labelAzi, labelAlt, labelFixo, labelAut;
GToggleGroup tg = new GToggleGroup();
GOption opt0, opt1;

ArrayList<GTextIconBase> puppets = new ArrayList<GTextIconBase>();

void setup(){

    size(800, 500);

    //Declaração o Arduino responsável por controlar os servo motores e tratar os sinais dos
    LDRs
    arduino = new Arduino(this, "COM4", 57600);

    //Declaração da porta serial que receberá os dados do Arduino responsável pelos sensores de
    tensão
    port = new Serial(this, "COM5", 9600);

    //Criação do documento que registro as leituras
    dataReport = createWriter("Dia1.txt");

    //Fonte dos textos utilizados na interface

```

```

font = createFont("calibri light", 20);

//Declaração das imagens utilizadas na interface
img_home = loadImage("home.png");
img_home.resize(20, 20);

img_uea = loadImage("logo_UEA.png");
img_uea.resize(100, 100);

//Declaração do botão de "home"
cp5 = new ControlP5(this);

cp5.addButton("home")
  .setPosition(430, 350)
  .setSize(20, 10)
  ;

//Declaração do switch ON/OFF
btnToggle0 = new GImageToggleButton(this, 30, 400);
btnToggle0.tag = "Power";

//Declaração dos gráficos
plot = new GPlot(this, 10, 50);
plot.setTitleText("Módulo com rastreador solar");
plot.setDim(250, 130);
plot.getXAxis().setAxisLabelText("Minuto");
plot.getYAxis().setAxisLabelText("Tensão(V)");

plot1 = new GPlot(this, 340, 50);
plot1.setTitleText("Módulo fixo");
plot1.setDim(250, 130);
plot1.getXAxis().setAxisLabelText("Minuto");

//Declaração do joystick de controle manual do rastreador
float ss = 130;
joystick = new GStick(this, 300, 330, ss, ss);
joystick.setMode(G4P.X4);
strokeWeight(1.5f);

//Declarando os pinos dos Arduino e posicionamento inicial dos servos
arduino.pinMode(pin_azimute, Arduino.SERVO);
arduino.pinMode(pin_altitude, Arduino.SERVO);
arduino.servoWrite(pin_azimute, 0);
arduino.servoWrite(pin_altitude, 0);

//Chamando funções desenvolvidas no script
makePuppets();
makeSliders();
createLabels();

```

```

}

void draw(){

//Declarando as características da interface
background(255);
fill(0);
textFont(font);
text("Sistema Supervisor de Energia Solar", 250, 30);

image(img_uea, 690, 1);
image(img_home, 430, 330);

GPoint lastPoint = plot.getPointsRef().getLastPoint();

//Se o sistema estiver ligado
if (systemON == true){

//Inicia a leitura da portal serial, informação enviada pelo Arduino dos sensores de tensão
if (port.available() > 0 ) {
String value = port.readString();
if ( value != null ) {
num = float(value);
if (aux == 0){
//Armazena no documento a hora, minuto e o dado lido
dataReport.print(hour());
dataReport.print(":");
dataReport.print(minute());
dataReport.print("-> Tensão Rastreador: ");
dataReport.println( num );
aux = 1;
voltage_automatic = num;
delay(500);
data = 1;
} else if (aux == 1){
dataReport.print(hour());
dataReport.print(":");
dataReport.print(minute());
dataReport.print("-> Tensão Fixo: ");
dataReport.println( num );
voltage_fixed = num;
aux = 0;
}
}
}

//Insera nos gráficos os dados lidos na porta serial
if (lastPoint == null) {
plot.addPoint(minute(), voltage_automatic);
}
}

```

```

    }
    else if (!lastPoint.isValid() || sq(lastPoint.getX() - mouseX) + sq(lastPoint.getY() +
mouseY) > 2500) {
        plot.addPoint(minute(), voltage_automatic);
    }

    if (lastPoint == null) {
        plot1.addPoint(minute(), voltage_fixed);
    }
    else if (!lastPoint.isValid() || sq(lastPoint.getX() - mouseX) + sq(lastPoint.getY() +
mouseY) > 2500) {
        plot1.addPoint(minute(), voltage_fixed);
    }

    //Reinicia os gráficos quando chegar nos 59 minutos
    if (minute() == 59){
        plot.setPoints(new GPointsArray());
        plot1.setPoints(new GPointsArray());
    }
}

//Executa os gráficos
plot.defaultDraw();
plot1.defaultDraw();

//Se o controle manual estiver ativado
if (manual == true){

    //Tratamento das posições executadas no joystick
    if (stickX > 0){
        if (azimute_pos > 0 ){
            azimute_pos-=1;
            arduino.servoWrite(pin_azimute, azimute_pos);
            delay(delay_servo);
        }
    }

    if (stickX < 0){
        if (azimute_pos < 180){
            azimute_pos+=1;
            arduino.servoWrite(pin_azimute, azimute_pos);
            delay(delay_servo);
        }
    }

    if (stickY > 0){
        if (altitude_pos > 0){
            altitude_pos-=1;
            arduino.servoWrite(pin_altitude, altitude_pos);
            delay(delay_servo);
        }
    }
}

```

```

    }
  }

  if (stickY < 0){
    if (altitude_pos < 90){
      altitude_pos+=1;
      arduino.servoWrite(pin_altitude, altitude_pos);
      delay(delay_servo);
    }
  }
}

//Se o controle automático estiver ativado
if (automatic == true){

  //Declaração de variáveis locais com dados do Arduino
  int LDR_up_left = arduino.analogRead(0);
  int LDR_up_right = arduino.analogRead(1);
  int LDR_down_left = arduino.analogRead(2);
  int LDR_down_right = arduino.analogRead(3);

  //Declaração das médias
  float avg_up = (LDR_up_left + LDR_up_right)/2;
  float avg_down = (LDR_down_left + LDR_down_right)/2;
  float avg_left = (LDR_up_left + LDR_down_left)/2;
  float avg_right = (LDR_up_right + LDR_down_right)/2;

  //Declaração da tolerância
  float tol = 50;

  //Se a média dos LDRs superior for maior que a inferior somado à tolerância
  if (avg_up > avg_down + tol){

    //Se a altitude for menor que 90°, acresce no eixo de altitude
    if (altitude_pos < 90){
      altitude_pos+=1;
      arduino.servoWrite(pin_altitude, altitude_pos);
      delay(delay_servo);
    }

    //Se a média dos LDRs da esquerda for maior que a da direita somado à tolerância
  }else if (avg_left > avg_right + tol){

    //Se a posição azimutal for maior que 0°, decresce no eixo azimutal
    if (azimute_pos > 0 ){
      azimute_pos-=1;
      arduino.servoWrite(pin_azimute, azimute_pos);
      delay(delay_servo);
    }
  }
}

```

```

//Se a média dos LDRs da direita for maior que a da esquerda somado à tolerância
}else if (avg_right > avg_left + tol){

    //Se a posição azimutal for menor que 180°, acresce no eixo azimutal
    if (azimute_pos < 180){
        azimute_pos+=1;
        arduino.servoWrite(pin_azimute, azimute_pos);
        delay(delay_servo);
    }

    //Se a média dos LDRs de baixo for maior que a de cima somado à tolerância
    }else if (avg_down > avg_up + tol){

        //Se a altitude for maior que 0°, decresce no eixo de altitude
        if (altitude_pos > 0){
            altitude_pos-=1;
            arduino.servoWrite(pin_altitude, altitude_pos);
            delay(delay_servo);
        }
    }

    delay(delay_servo);

}

//Atualiza os valores nos sliders dos eixos
sdrAzi.setValue(azimute_pos);
sdrAlt.setValue(altitude_pos);

//Se o sistema for desligado
if (systemON == false && data > 0){

    //Encerra a gravação dos dados no documento
    dataReport.flush();
    dataReport.close();
}

}

//Função que lida com os eventos do botão ON/OFF
public void handleToggleButtonEvents(GImageToggleButton button, GEvent event) {

    if (button.tag == "Power" && button.getState() > 0){
        systemON = true;
        labelOUT.setText("Sistema ligado");
    }
    if (button.tag == "Power" && button.getState() == 0){
        labelOUT.setText("Sistema desligado");
        systemON = false;
        opt0.setSelected(false);
    }
}

```

```

    opt1.setSelected(false);
}

}

//Função que cria e posiciona as opções de controle
public void makePuppets(){
    opt0 = new GOption(this, 550, 400, 90, 46, "Manual");
    opt1 = new GOption(this, 680, 400, 90, 46, "Automático");
    tg.addControls(opt0, opt1);

    puppets.add(opt0);
    puppets.add(opt1);
}

//Função que lida com os eventos das opções de controle manual ou automático
public void handleToggleControlEvents(GToggleControl option, GEvent event){
    if (systemON == true && opt0.isSelected()){
        labelOUT.setText("Sistema ligado e em Modo Manual de operação");
        manual = true;
        automatic = false;
    }
    if (systemON == true && opt1.isSelected()){
        labelOUT.setText("Sistema ligado e em Modo Automático de operação");
        automatic = true;
        manual = false;
    }
}

//Função que cria os textos utilizados na interface
public void createLabels() {
    labelPower = new GLabel(this, 80, 400, 110, 50);
    labelPower.setText("Power");

    labelON = new GLabel(this, 40, 360, 110, 50);
    labelON.setText("ON");

    labelOFF = new GLabel(this, 35, 440, 110, 50);
    labelOFF.setText("OFF");

    labelOperation = new GLabel(this, 600, 360, 110, 50);
    labelOperation.setText("Modo de Operação");

    labelCon = new GLabel(this, 310, 290, 110, 50);
    labelCon.setText("Controlador Manual");

    labelGraus = new GLabel(this, 680, 190, 110, 50);
    labelGraus.setText("Eixos em Graus (°)");
}

```

```

labelAlt = new GLabel(this, 751, 80, 110, 50);
labelAlt.setText("Alt");

labelAzi = new GLabel(this, 699, 80, 110, 50);
labelAzi.setText("Azi");

labelOUT = new GLabel(this, 0, height - 26, width, 26,
    "Sistema desligado");
labelOUT.setOpaque(true);
labelOUT.setLocalColorScheme(G4P.CYAN_SCHEME);

}

//Função que trata o clique no botão home
void home(){

    if (azimute_pos > 0) {
        for (azimute_pos--; azimute_pos >= 0; azimute_pos -= 1){
            arduino.servoWrite(pin_azimute, azimute_pos);
            delay(100);
            arduino.servoWrite(pin_azimute, 0);
        }
    }

    if (altitude_pos > 0) {
        for (altitude_pos++; altitude_pos >= 0; altitude_pos -= 1){
            arduino.servoWrite(pin_altitude, altitude_pos);
            delay(100);
            arduino.servoWrite(pin_altitude, 0);
        }
    }
    delay(10);
}

//Função que lida com os eventos do joystick
public void handleStickEvents(GStick stick, GEvent event) {

    if (joystick == stick) {
        int pos = stick.getPosition();
        if (pos < 0){
            stickX = 0;
            stickY = 0;
        }else{
            stickX = stick.getStickX();
            stickY = stick.getStickY();
        }
    }
}
}

```

```
//Função que cria e posiciona os sliders
public void makeSliders(){

    sdrAzi = new GSlider(this, 670, 200, 90, 80, 12);
    sdrAzi.setLimits(0, 0, 180);
    sdrAzi.setRotation(-PI/2);
    sdrAzi.setTextOrientation(G4P.ORIENT_RIGHT);
    sdrAzi.setShowValue(true);
    sdrAzi.setShowTicks(true);

    sdrAlt = new GSlider(this, 720, 200, 90, 80, 12);
    sdrAlt.setLimits(0, 0, 90);
    sdrAlt.setRotation(-PI/2);
    sdrAlt.setTextOrientation(G4P.ORIENT_RIGHT);
    sdrAlt.setShowValue(true);
    sdrAlt.setShowTicks(true);

}
```