



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Luis Filipi Garcia

**Desenvolvimento de uma Plataforma de Aquisição de Dados
Para Medição e Armazenamento da Radiação Solar**

Araranguá
2022

Luis Filipi Garcia

**Desenvolvimento de uma Plataforma de Aquisição de Dados
Para Medição e Armazenamento da Radiação Solar**

Trabalho Conclusão do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia de Computação.
Orientador: Prof. Dr. Roderval Marcelino

Araranguá

2022

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

Garcia, Luis Filipi

Desenvolvimento de uma Plataforma de Aquisição de Dados
Para Medição e Armazenamento da Radiação Solar / Luis Filipi Garcia ;
orientador, Roderval Marcelino, 2022.

22 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em
Engenharia de Computação, Araranguá, 2022.

Inclui referências.

1. Engenharia de Computação. 2. Engenharia de Computação.
3. Sistemas Embarcados. 4. Sistemas Distribuídos. 5. Sistemas Web. I.
Marcelino, Roderval. II. Universidade
Federal de Santa Catarina. Graduação em Engenharia de Computação. III.
Titulo.

Luis Filipi Garcia

Desenvolvimento de uma Plataforma de Aquisição de Dados para Medição e Armazenamento da Radiação Solar

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 21 de março de 2022.

Profa. Analucia Schiaffino Morales, Dra.
Coordenadora do Curso

Banca Examinadora:

Prof. Roderval Marcelino, Dr.
Orientador
Universidade Federal de Santa Catarina

Prof. Jim Lau, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. André Possamai Rosso, Me.
Avaliador
Instituto Federal de Santa Catarina

Prof. Fabio De La Rocha, Dr.
Avaliador Suplente
Universidade Federal de Santa Catarina

Desenvolvimento de uma Plataforma de Aquisição de Dados para Medição e Armazenamento da Radiação Solar

Luis Filipi Garcia¹

Roderval Marcelino²

2022, Março

Resumo

As usinas fotovoltaicas de grande porte podem gerar menos energia do que o esperado, por fatores que podem variar durante sua operação. Uma das maneiras da obtenção de dados de irradiância é a medição com células de referência, tendo como vantagem o reduzido custo de construção e de operação e a resposta espectral idêntica à do módulo fotovoltaico quando utilizada da mesma tecnologia. A medição da irradiância em um único ponto em usinas fotovoltaicas pode acarretar valores não compatíveis com os valores de geração de toda a usina, pois à medida que as nuvens se movem sobre uma planta fotovoltaica, cada módulo fotovoltaico é impactado em um momento diferente, dependendo da localização do mesmo e das características da nuvem. Nas usinas fotovoltaicas de grande porte, precisa-se de células de referência espalhadas para diminuir o erro na previsão da geração fotovoltaica, além de garantir que as medições de todas as células estão sendo feitas ao mesmo instante, preferencialmente de forma automatizada para mitigar erros causados por falhas humanas, como algum atraso na medição. Através da importância da medição da radiação solar não ser de maneira manual, este artigo apresenta o desenvolvimento de um sistema de aquisição e armazenamento de dados de irradiância solar. O sistema distribuído embarcado irá armazenar esses dados gerados pelas células de referência e posteriormente demonstrará esses dados através de gráficos através de uma aplicação *web*.

Palavras-chaves: Sistemas Embarcados. Célula e Módulo de Referência. Sistemas Distribuídos. Sistemas *Web*

¹ filipi.garcia7@gmail.com

² roderval.marcelino@ufsc.br

1 Introdução

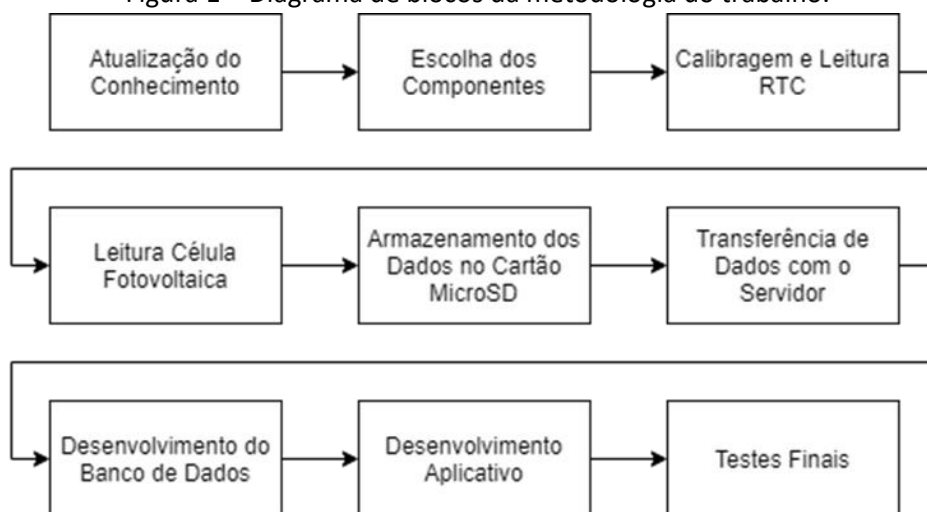
A informação sobre o recurso solar é a variável com maior importância para o desenvolvimento de um projeto de sistema de aproveitamento da energia solar. Com isso, a medição precisa da radiação solar é muito importante tanto para melhorar a tecnologia dos módulos como para prever o desempenho (KEOGH e BLAKERS, 2004).

A geração de energia através de sistemas fotovoltaicos é afetada pelas sombras das nuvens. Com isso, previsões precisas de nuvens passando sobre o campo fotovoltaico são necessárias para o gerenciamento otimizado do sistema. Redes de sensores de irradiância estão se posicionando como uma fonte promissora de dados nas escalas espacial e temporal do problema (ESPINOSA-GARIVA et al, 2020).

As previsões precisas da radiação solar fornecem informações sobre as incertezas da geração de eletricidade de uma usina solar. Através do conhecimento das incertezas, podemos gerenciar a entrada de energia elétrica na rede para não causar distúrbios de energia (BAKKER et al. 2019).

Com isso, este trabalho tem como finalidade a obtenção dos dados para o estudo das previsões precisas de uma usina de energia elétrica fotovoltaica através da construção de um sistema embarcado distribuído de aquisição de dados e um *software* de leitura e monitoramento, integrando-os a um banco de dados para análises futuras. A metodologia utilizada para o desenvolvimento do trabalho foi separada nas seguintes seções representadas na Figura 1: atualização do conhecimento através de uma revisão sobre os temas do trabalho, escolha dos componentes para a montagem do sistema embarcado, calibragem e leitura dos relógios do sistema, leitura da célula fotovoltaica pelo sistema embarcado, armazenamento dos dados para o estudo das previsões precisas em um cartão MicroSD, transferência de dados com o servidor onde estará o aplicativo para a análise da aquisição de dados, desenvolvimento de um banco de dados relacional para o armazenamento dos dados gerados, desenvolvimento de uma interface amigável para o usuário obter os dados armazenados, testar o *software* para análise da aquisição de dados.

Figura 1 – Diagrama de blocos da metodologia do trabalho.



Fonte: Do autor.

2 Fundamentação Teórica

O fluxo de energia emitida pelo sol e transmitida através de radiação eletromagnética é denominado de radiação solar. Durante muitos anos, a radiação solar era considerada uma constante. No entanto, observações desde a década de 1970 demonstram a variação da radiação solar através do tempo em escala de minutos e de décadas. (GOMEZ, J.M.R. et al, 2018).

De acordo com a ANEEL (Agência Nacional de Energia Elétrica) e a ABSOLAR (Associação Brasileira de Energia Solar Fotovoltaica), em fevereiro de 2022 o Brasil alcançou a geração de 4,6 GW através de usinas solares de grande porte, correspondendo a 2,4% de toda a matriz energética do país.

A produção de energia elétrica pela energia solar é resultado da incidência de radiação solar nas células solares fotovoltaicas. Existem diversas maneiras de obtenção de dados de irradiância, uma delas é através de células de referências, que são dispositivos usados para uma obtenção da radiação solar. As vantagens dessas células são o custo reduzido para construção e operação e a resposta espectral idêntica ao módulo fotovoltaico quando utilizadas da mesma tecnologia (ROSSO, A.P. et al, 2019).

A medição da irradiância em um único ponto de uma usina fotovoltaica pode gerar valores não compatíveis com os gerados por toda a usina. Na geração centralizada, a dispersão espacial das condições de operação da usina pode afetar os valores medidos em um único ponto da planta, podendo mudar os resultados de desempenho de toda a geração (DYRESON et al, 2014; ESPINOSA-GARIVA et al, 2020).

Com o crescimento da energia solar fotovoltaica na matriz energética, haverá um aumento significativo da importância das informações meteorológicas e climáticas em razão de seu forte impacto no planejamento e operação dos sistemas fotovoltaicos de geração de energia elétrica. Com isso, observa-se a importância da realização de medições precisas da radiação solar em campo, buscando compreender a variação da radiação solar ao longo do tempo e a sua dispersão espacial (ROSSO, A.P. et al, 2019).

A partir do crescimento da energia solar fotovoltaica na matriz energética, deverá aumentar significativamente a importância das informações meteorológicas e climáticas em razão de seu forte impacto no planejamento e operação dos sistemas fotovoltaicos de geração de energia elétrica. A disponibilidade e a variabilidade do recurso energético solar estão intrinsecamente associadas às condições de tempo e clima da região (PEREIRA et al, 2017).

Sistema distribuído é o sistema onde um conjunto de processos em um ou mais sítios, são conectados entre si através de uma rede e são executados de forma concorrente. A utilização desses sistemas vem crescendo cada vez mais conforme a evolução da área de redes de computadores (GORENDER, S. et. al, 2002).

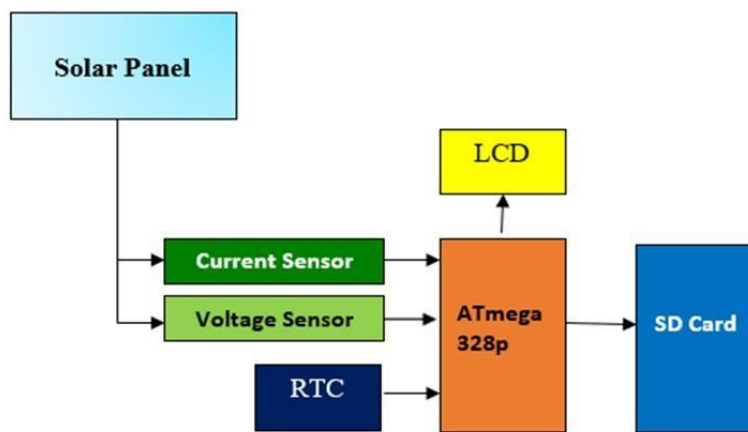
Aquisição de dados consiste em fazer medições de fenômenos físicos e registrá-los de alguma maneira no meio digital para serem analisados futuramente. Os sistemas de aquisição de dados atualmente consistem em quatro componentes essenciais: sensores, condicionadores de sinal, conversores analógico-digital e algum computador com *software* de aquisição de dados para gravação e análise (DEWESOFT, 2020).

3 Estado da Arte

Os microcontroladores são uma junção de *software* com *hardware*, contendo ao menos uma memória, um processador e interfaces de entrada e saída, como sensores, *displays*, botões e cartões de memória. Os microcontroladores são geralmente utilizados em circuitos de controle ou sistemas que irão fornecer informações para o circuito de controle. O microcontrolador precisa de um *software* para definir o que fazer com seus sinais de entrada, que é escrito ou convertido através de um compilador para linguagem de máquina, e registrado no microcontrolador. Os sistemas de aquisição de dados são utilizados em medições da radiação solar para analisar e arquivar os dados de irradiância.

Hadi et al, 2018 utilizaram um microcontrolador ATmega328p para desenvolver o *datalogger* junto com um sensor de corrente e de tensão. Utilizou um relógio de tempo real RTC (*Real Time Clock*), um LCD (*Liquid Crystal Display*) e um módulo de cartão SD para arquivar as leituras. O diagrama de bloco do *datalogger* está apresentado na Figura 2.

Figura 2 – Diagrama de blocos do *datalogger*.



Fonte: Hadi et al (2018).

Hadi et al, 2018 utilizaram um divisor de tensão como sensor de tensão para converter a tensão de saída do dispositivo fotovoltaico. A tensão de saída do dispositivo varia de 0 V até 48 V e foi dividido para uma tensão de 0 V até 5 V, ficando na faixa da tensão de entrada de leitura do microcontrolador. O sensor de corrente que foi utilizado é um sensor ACS712. O sensor ACS712 é um sensor que converte pulso elétrico para uma faixa de tensão de 0V até 5V. Para o relógio de tempo real foi utilizado um DS1307 RTC, que pode salvar o tempo em segundos, minutos, horas, dia, mês e ano, com mudança de mês de 30 ou 31 dias e anos bissextos. Além disso, o RTC possui sua própria tensão de alimentação, possibilitando deixar o relógio ligado enquanto todo o sistema está desligado. O display LCD foi utilizado para mostrar ao usuário em tempo real as informações medidas. Como cartão SD utilizaram um Vgen de 8GB de memória, para poder transportar as informações salvas para um computador.

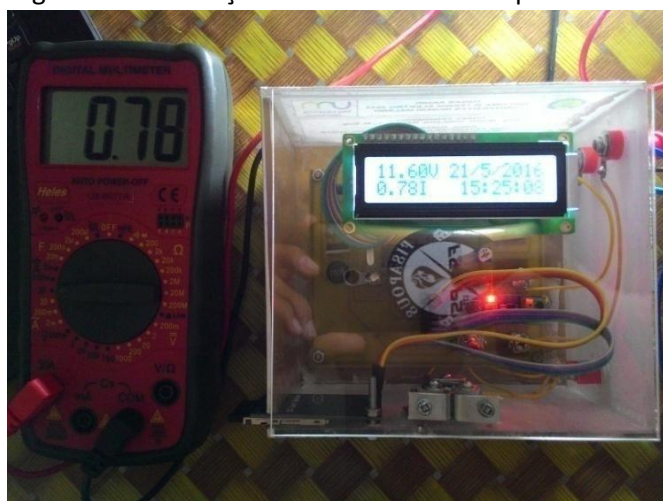
Para validar as medições, foi utilizado um multímetro para medir tanto a corrente quanto a tensão gerada. A representação dessa medida estão presentes nas Figura 3 a e b a seguir:

Figura 3a – Validação da tensão medida pelo sistema.



Fonte: Hadi et al (2018).

Figura 3b – Validação da corrente medida pelo sistema.



Fonte: Hadi et al (2018).

Pode-se observar que há uma pequena variação na medição da tensão, mas com um resultado muito satisfatório nas medidas. O estudo observou que o erro médio do sistema é de 3%, sendo causado pelo sensor de corrente.

Dyrezon et al, 2014 utilizaram um resistor *shunt* para converter a corrente de um sensor de irradiância 200S da LI-COR em tensão. A leitura em tensão foi através de um microcontrolador Propeller da Parallax Inc. Esse microcontrolador armazena os dados de irradiância, tempo, temperatura do chip e a tensão em um cartão SD de 2GB. Cada sensor de irradiância foi montado em um poste e conectado com o sistema embarcado como demonstrado na Figura 4:

Figura 4 – Sensor de irradiância e sistema embarcado.



Fonte: Dyrezon et al (2014).

A comunicação entre o computador e o sistema de aquisição de dados foi feita através de uma comunicação *half-duplex*, onde somente um dos lados transfere dados, e um programa na linguagem Turbo C++ foi desenvolvido para facilitar a comunicação.

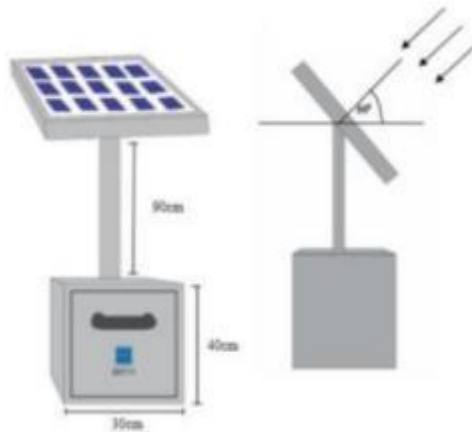
Com o trabalho, pôde-se observar que como esperado, os valores foram alterados conforme o coeficiente de velocidade das nuvens, mas com resultados consistentes para a energia medida pelos sensores.

Chowdhury et al, 2011, desenvolveram um sistema de segurança baseado em sensor infravermelho usando um microcontrolador, sendo utilizado um sensor PIR (*passive infrared*) conectado a um regulador de tensão de 5 V LM7805, um sensor de temperatura LM35 e um microcontrolador PIC 16F876A conectado a um cristal de 4 MHz, além de um alarme como saída. O sensor de temperatura fica monitorando a temperatura do ambiente, e como o sensor PIR não funciona abaixo de 15 °C, o microcontrolador só começará a monitorar o ambiente quando esta temperatura seja atingida ou superada. O *software* foi feito na linguagem Assembly e compilado através do MPLAB, e consiste em quando o sistema estiver ativo, ele irá verificar se a temperatura está igual ou acima de 15 °C, se estiver e algum movimento for captado pelo sensor PIR, o microcontrolador espera um tempo pré-definido e então busca novamente algum movimento antes de acionar o alarme, para evitar acioná-lo por algum ruído.

Devido a interface de fácil acesso, seu baixo custo e baixo consumo de energia, esse sistema teve um resultado satisfatório para ser utilizado em casas, escritórios e lojas. Ainda assim, em trabalhos futuros há algumas melhorias que podem ser feitas, como a utilização sensores mais robustos para um resultado mais satisfatório, mais sensores para o monitoramento de áreas maiores, etc.

Gusa et al, 2018 utilizaram um Arduino Atmega 2560 com um sensor ACS712 para monitorar a corrente e tensão de um painel fotovoltaico, além de um sensor de temperatura DHT11 para monitorar a temperatura ao redor do painel fotovoltaico e um módulo *Wi-Fi* ESP8266 para passar essas informações em tempo real a um *smartphone* entre as 08:00 horas da manhã e 04:00 horas da tarde durante sete dias. O *design* do protótipo está representado na Figura 5:

Figura 5 – Design do painel fotovoltaico conectado a uma caixa onde estará o sistema embarcado de monitoramento.



Fonte: Gusa et al (2018).

Também foi utilizado uma tela para monitoramento em tempo real da tensão, corrente e energia gerada pelo painel, além da temperatura do ambiente, representada na Figura 6.

Figura 6 – Tela para monitoramento. Em amarelo a tensão, em azul a corrente, branco a energia e em vermelho a temperatura. Os valores ao lado do círculo verde é a energia.



Fonte: Gusa et al (2018).

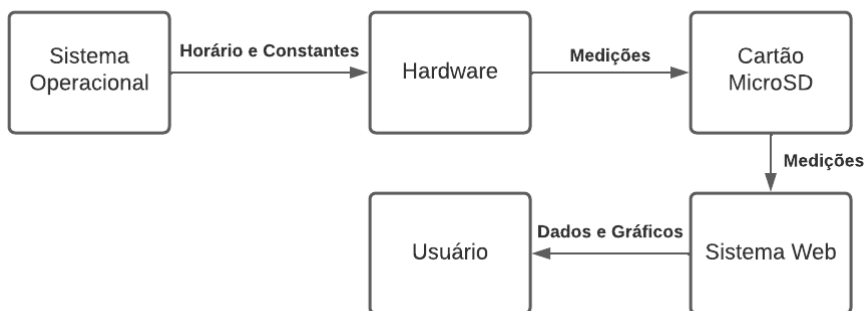
Os estudos demonstraram que a margem de erro desse sistema foi de menos de 10%, e com a representação em tempo real no *smartphone* de forma satisfatória. Com isso, concluiu-se que o sistema contempla o seu objetivo de desenvolvimento.

4 Desenvolvimento

Os componentes utilizados no trabalho foram escolhidos através das conclusões obtidas com o Estado da Arte e procura pelo melhor custo e disponibilidade no mercado. Com os componentes escolhidos e comprados, o projeto foi analisado como um todo e separado em três partes: *Hardware*, Banco de Dados e Sistema *Web*. O diagrama de blocos de todo o projeto está

representado pela Figura 7, onde pode-se observar que o sistema operacional do computador envia as informações de horários e constantes da célula para o hardware, que faz as medições e as envia para o cartão MicroSD, onde o usuário envia as informações armazenadas para o sistema *web* que retorna ao próprio usuário as medições através de dados e gráficos.

Figura 7 – Diagrama de blocos do projeto.

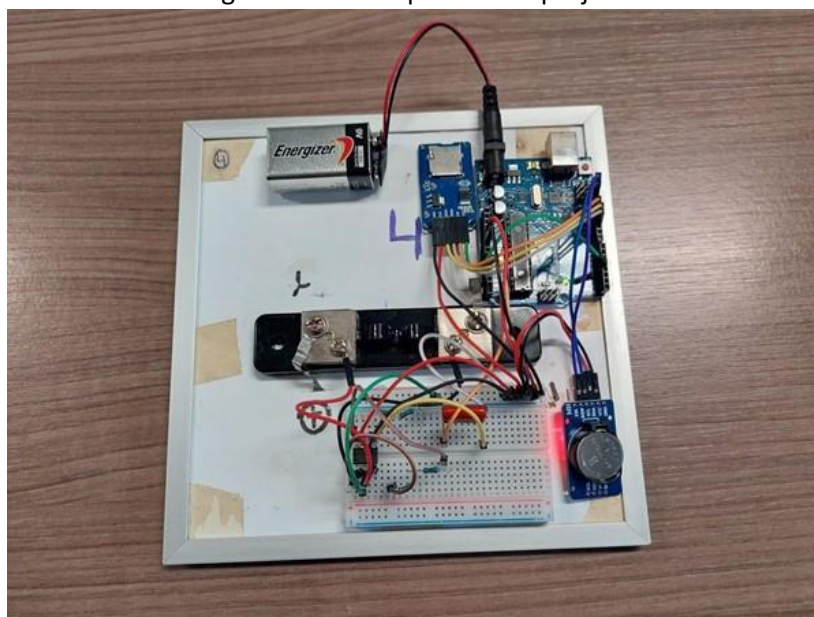


Fonte: Do autor.

4.1 Hardware

A plataforma escolhida para ser responsável pelo controle do sistema foi o Arduino Uno, responsável pela leitura da tensão do *shunt* escolhido por conta do seu baixo preço e alta precisão comparado à outras opções do mercado, amplificada por um amplificador operacional, além da calibragem e conversão desta tensão para irradiância. O Arduino também foi responsável pela calibragem e leitura do horário e temperatura registrados pelo relógio de tempo real e do armazenamento de todas essas informações no cartão MicroSD. A programação do Arduino foi toda desenvolvida através da IDE (*Integrated Development Environment*) do próprio Arduino. Podemos observar a montagem final do protótipo através da Figura 8.

Figura 8 – Protótipo final do projeto.



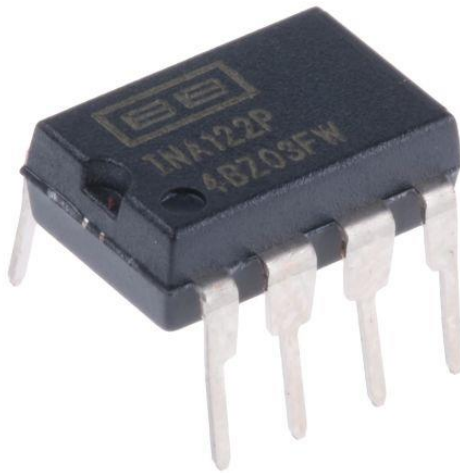
Fonte: Do autor.

O Arduino IDE é um *software* de código aberto desenvolvido para facilitar a criação de códigos e o carregamento para a placa. O *software* pode ser utilizado com qualquer placa Arduino (ARDUINO, 2021).

4.1.1 Amplificador INA122P e filtro

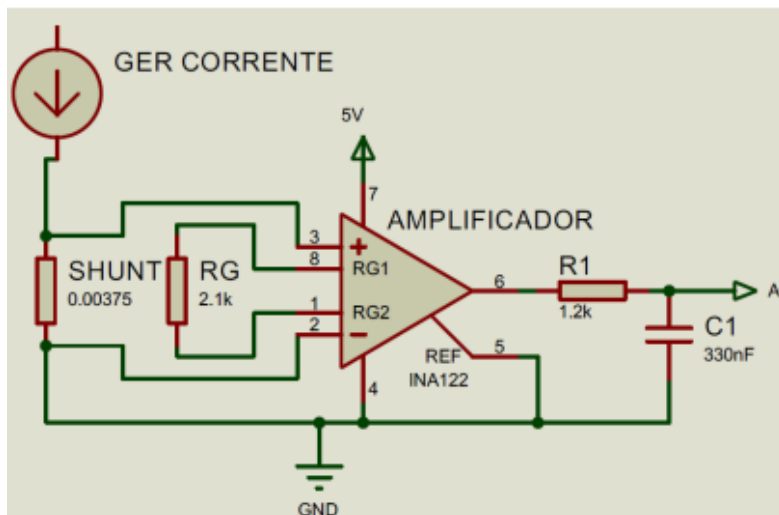
A saída do resistor *shunt* é conectada a um amplificador operacional INA122P desenvolvido pela Texas Instruments, demonstrado na Figura 9, junto a um filtro de tipo RC que elimina sinais de interferência indesejados, para converter o valor de tensão no resistor *shunt* à faixa detectada pelas portas analógicas do Arduino. O circuito elétrico da amplificação da tensão no resistor *shunt* está representado na Figura 10.

Figura 9 – Modelo de amplificador operacional utilizado no trabalho.



Fonte: octopart.com.

Figura 10 – Circuito eletrônico amplificador.



Fonte: ROSSO, A.P. et al, 2019.

O código onde o Arduino faz a leitura e conversão dos dados está representado na Figura 11.

Figura 11 – Código de leitura do amplificador operacional.

```
void setup(void)
{
    pinMode(in, INPUT);
}
void loop(void)
{
    adc_noise1 = analogRead(in);
    adc_noise2 = moving_average(adc_noise1);

    tensao = (adc_noise2*(5.0/1023.0))/ganho;
    corrente = tensao*(20/0.075);
    radiancia = (tensao/0.0285)*1000;
}
long moving_average(int sig)
{
    int i;
    long acc = 0;
    for(i = num; i > 0; i--) values[i] = values[i-1];
    values[0] = sig;
    for(i = 0; i < num; i++) acc += values[i];
    return acc / num;
}
```

Fonte: Do autor.

Três argumentos são pré-definidos para o código da Figura 11 e serão alterados para cada célula fotovoltaica. O argumento *num* representa a quantidade desejada de medições a serem feitas para ser calculado uma média e definir o valor a ser armazenado, buscando assim diminuir possíveis ruídos nas medições. O argumento *shunt* é a constante de calibração da célula, sendo medido de forma experimental junto a uma outra célula com constante já definida. Esta calibração é feita de forma a encontrar um valor em que a corrente medida pelas células seja a mais aproximada possível. Por último, o argumento *ganho* é o ganho que o amplificador operacional proporciona ao valor do *shunt*. Com estes valores, o código lê o que chega ao pino analógico do Arduino através da função *analogRead()* e calcula a média das medições definidas por *num* através da função *moving_average()*. Após o retorno da média, faz-se a conversão da tensão real dividindo o valor medido pelo ganho, e com a tensão real convertida se obtém a irradiância.

4.1.2 Relógio de tempo real (RTC)

O relógio de tempo real escolhido para o sistema foi o DS3231 devido a sua faixa de temperaturas permitidas (0°C a 70°C) e sua precisão (± 2 ppm) serem suficientes para o projeto. A Figura 12 demonstra o DS3231.

Figura 12 – Relógio de tempo real usado no trabalho.



Fonte: filipeflop.com.

Este RTC foi calibrado através do sistema operacional do computador no momento da compilação do código para o Arduino. Para mitigar erros causados pelo atraso humano entre compilar e carregar o código para o Arduino, foi desenvolvido um *script* na linguagem de programação Python que faz a compilação e carrega o código automaticamente. Este *script* está representado na Figura 13.

Figura 13 – *Script* em Python para compilação e carregamento do código.

```
import os

file = open(r"D:\xampp\htdocs\CelulasReferencia\Python\cmd.txt")
lines = file.readlines()

for line in lines:
    os.system(line)
```

Fonte: Do autor.

O *script* da Figura 13 faz a leitura de um arquivo que contém os comandos necessários para o carregamento e compilação do código do Arduino, onde o usuário final do sistema poderá modificar conforme o sistema operacional utilizado.

Para a leitura do horário e temperatura armazenado no RTC, é utilizada a biblioteca do Arduino RTCLib e suas funções *getTemperature()* e as funções *year()*, *month()*, *day()*, *hour()*, *minute()* e *second()* do objeto *now()* da mesma biblioteca, conforme demonstrado na Figura 14.

Figura 14 – Código da leitura das informações do RTC.

```
RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {"Domingo", "Segunda", "Terça", "Quarta", "Quinta", "Sexta", "Sábado"};
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
float c = rtc.getTemperature();
t = rtc.now();
float ano = t.year();
float mes = t.month();
float dia = t.day();
float hora = t.hour();
float minuto = t.minute();
float segundo = t.second();
```

Fonte: Do autor.

4.1.3 Armazenamento no cartão MicroSD

As informações são armazenadas em um cartão MicroSD através de um módulo do Arduino para leitura e escrita de dados em cartões SD e MicroSD. O código do Arduino armazena as informações no cartão através da função *grava_SD()*, que utiliza a função *abre_arquivo_gravacao()* para abrir e/ou criar o arquivo "data.txt" dentro do cartão. Com o arquivo aberto, é utilizada a função *print()* do objeto *File*, cuja escreve o que é passado como argumento no final de tudo que já está escrito no arquivo. O padrão de cada linha armazenada é o seguinte: *ID_CÉLULA;DATA("yyyy-MM-dd;HH:mm:ss");TEMPERATURA*. O código responsável por esta parte do sistema está apresentado na Figura 15.

Figura 15 – Código do armazenamento no cartão MicroSD.

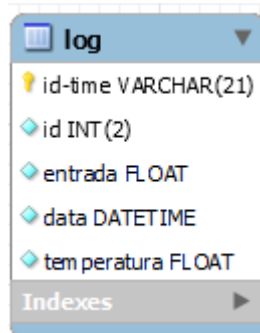
```
File file;
void grava_SD(DateTime t)
{
    float c = rtc.getTemperature();
    abre_arquivo_gravacao("data.txt");
    file.print(id_modulo);
    file.print(";");
    file.print(radiancia);
    file.print(";");
    file.print(t.year());
    file.print("-");
    if (t.month() < 10)
    {
        file.print("0");
    }
    file.print(t.month());
    file.print("-");
    if (t.day() < 10)
    {
        file.print("0");
    }
    file.print(t.day());
    file.print(";");
    if (t.hour() < 10)
    {
        file.print("0");
    }
    file.print(t.hour());
    file.print(":");
    if (t.minute() < 10)
    {
        file.print("0");
    }
    file.print(t.minute());
    file.print(":");
    if (t.second() < 10)
    {
        file.print("0");
    }
    file.print(t.second());
    file.print(";");
    file.println(c);
    fecha_arquivo();
}
int abre_arquivo_gravacao(char filename[])
{
    file = SD.open(filename, FILE_WRITE);
    if (file)
    {
        return 1;
    } else
    {
        return 0;
    }
}
void fecha_arquivo()
{
    if (file)
    {
        file.close();
    }
}
```

Fonte: Do autor.

4.2 Banco de Dados

Para o armazenamento das informações medidas e futuras consultas pelo sistema *web*, uma base de dados relacional na linguagem MySQL foi desenvolvida e hospedada no mesmo servidor que do sistema *web*. A base de dados possui apenas uma tabela de *Log* que irá armazenar uma chave primária *id-time*, uma coluna para o *id*, uma entrada que será a *irradiância*, uma *data* e outra *temperatura*. O diagrama da base está representado na Figura 16.

Figura 16 – Diagrama da Base de Dados.



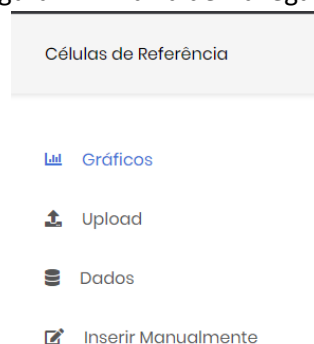
Fonte: Do autor.

4.3 Sistema Web

Para desenvolver o sistema *web* que irá armazenar e demonstrar os dados coletados, a linguagem de programação PHP foi escolhida junto ao framework Bootstrap devido ao conhecimento prévio com as duas tecnologias. Bootstrap é um framework *front-end* que fornece estruturas de *layout* para a criação de sites e aplicações responsivas de forma rápida e simples (ALURA, 2021). Como *template* do sistema foi utilizado um desenvolvido pelo site Colorlib.

O Sistema *Web* foi dividido em quatro páginas, sendo elas: *Gráficos*, *Upload*, *Dados* e *Inserir Manualmente*. Para a navegação entre as páginas, foi criada uma barra de navegação lateral responsável pela troca de páginas, que está representada na Figura 17.

Figura 17 – Barra de Navegação.

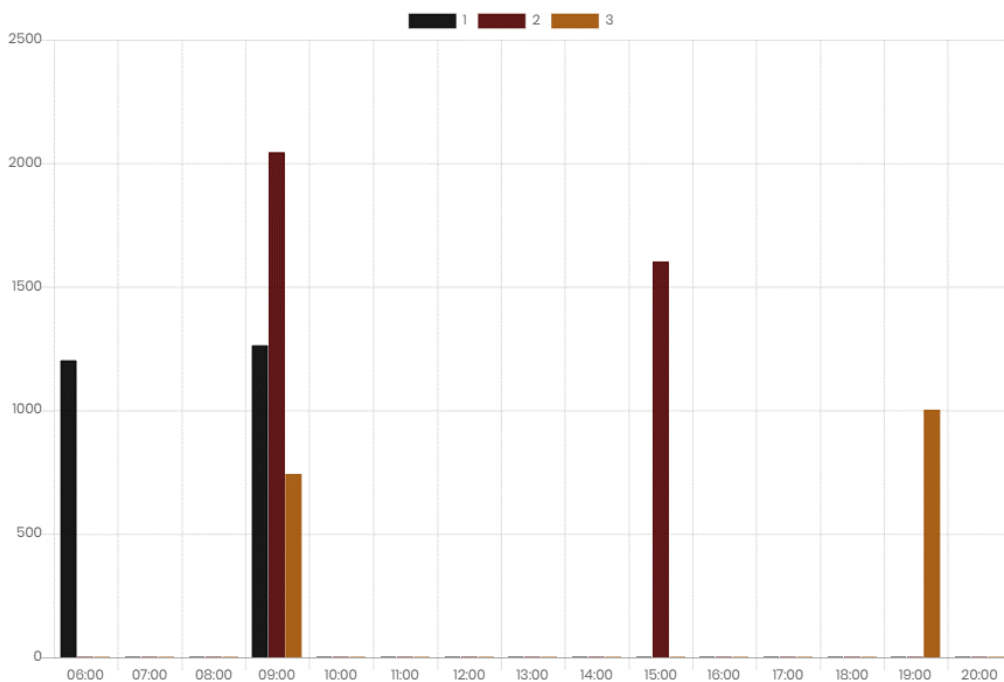


Fonte: Do autor.

A página inicial do sistema é a de gráficos, que busca demonstrar de forma mais clara as informações captadas durante todo o dia atual pelas células de referência dentro do período das 06:00hrs às 20:00hrs. Para atender este objetivo, há três gráficos diferentes com as medições

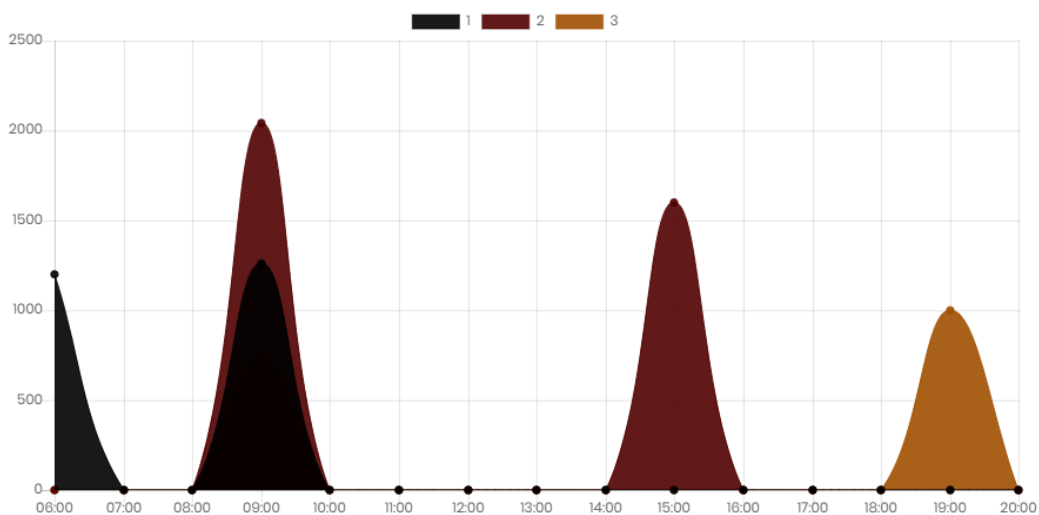
diárias: barras, linhas e pizza, representados respectivamente pelas Figuras 18, 19 e 20. Nos gráficos de barras e de linhas, o eixo horizontal representa o horário da medição em h, enquanto o eixo vertical representa a irradiância em W/m^2 , enquanto no gráfico de pizza o valor de cada fatia é a soma da irradiância em W/m^2 de todo o dia.

Figura 18 – Gráfico de barras do sistema *web*.



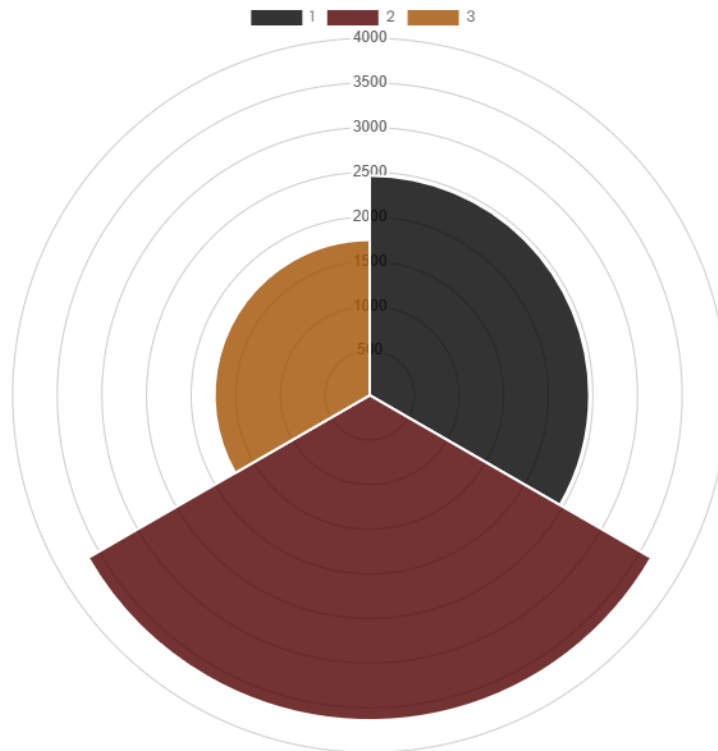
Fonte: Do autor.

Figura 19 – Gráfico de linhas do sistema *web*.



Fonte: Do autor.

Figura 20 – Gráfico de pizza do sistema *web*.



Fonte: Do autor.

Buscando mais praticidade e rapidez, foi desenvolvido uma página para fazer o *Upload* dos arquivos salvos dentro dos cartões MicroSD. Esta página consiste em um único botão de seleção para adicionar as informações contidas em todos os arquivos desejados, como demonstrado na Figura 21.

Figura 21 – Página de *upload* de arquivos.

Upload ×











Por favor, adicione o arquivo abaixo:

Nenhum ficheiro selecionado

Fonte: Do autor.

O sistema *web* possui também uma página de dados, onde todas as informações cadastradas na base de dados serão apresentadas junto a uma tabela, possuindo a opção para editar e apagar alguma informação específica. Esta página permite também que o usuário exporte todos os dados da base para um arquivo *.csv* clicando em um botão de “*Exportar*”. A página está representada na Figura 22.

Figura 22 – Página de dados.


ID	IRRADIÂNCIA	DATA	TEMPERATURA	
4	0	2022-02-15 20:04:10	19.25	 
9	14.01	2022-02-15 15:32:10	16.75	 
6	14.01	2022-02-15 09:32:30	16.75	 
7	14.01	2022-02-15 09:32:20	16.75	 
4	0	2022-02-14 20:04:00	19.25	 


Fonte: Do autor.


A última página do sistema é a de inserção manual, que consiste em um formulário simples com os campos das colunas da base de dados com o objetivo de adicionar novas linhas de forma manual à tabela, mesmo sem a linha estar armazenada no arquivo do cartão MicroSD de alguma das células. A página é representada pela Figura 23.


Figura 23 – Página de inserção manual.

Inserir Item Manualmente

ID Célula 

Irradiância 

Horário (yyyy-MM-dd HH:mm:ss) 

Temperatura 

Inserir

Fonte: Do autor.

5 Resultados e Discussões

As medições foram realizadas na cidade de Araranguá em Santa Catarina. A condição climática era nublado com chuva fraca, o que poderia prejudicar a precisão das medidas devido a uma passagem de nuvem no momento das medições.

Foram utilizadas três células de referência calibradas com suas respectivas constantes para os testes com IDs 1, 2 e 3. Todas foram colocadas apoiadas e viradas para cima lado a lado, buscando diminuir a interferência dos fatores externos. Todas as informações foram

armazenadas em seus respectivos cartões MicroSD, sendo configurado para o armazenamento ser feito a cada 10 segundos. A Figura 24 demonstra como foi montado o ambiente de testes.

Figura 24 – Ambiente de teste. Da esquerda para a direita: Célula 1, 2 e 3.



Fonte: Do autor.

Após o período de teste, todos os dados foram coletados e carregados para o sistema *web*. A Tabela 1 demonstra as medições mais relevantes.

Tabela 1 – Medições mais relevantes.

Horário (HH:mm:ss)	Entrada 1 (W/m ²)	Entrada 2 (W/m ²)	Entrada 3 (W/m ²)
09:13:30	1,75	1,73	12,19
09:13:40	1,75	1,73	12,19
09:13:50	1,75	1,73	12,19
09:14:00	1,75	1,73	12,19
09:14:10	1,75	1,73	12,19
09:14:20	1,75	1,73	12,19
09:16:20	5,25	6,93	10,45
09:16:30	5,25	6,93	10,45
09:16:40	5,25	6,93	10,45
09:16:50	5,25	6,93	10,45

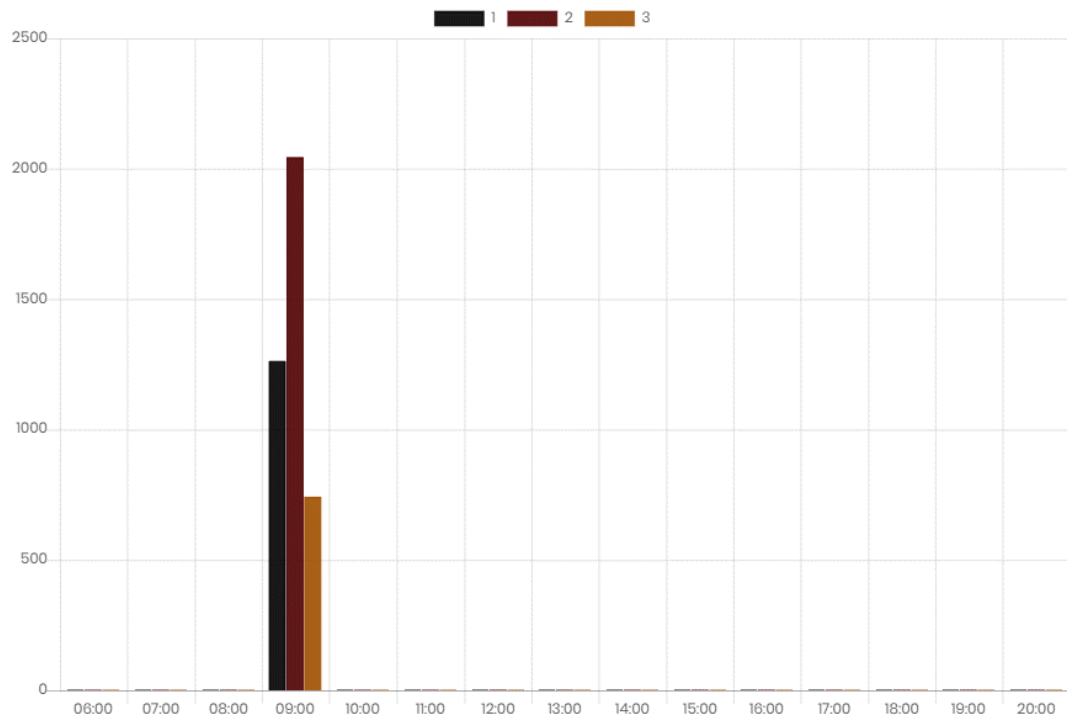
09:17:00	5,25	6,93	10,45
09:19:00	8,75	12,14	8,71
09:19:10	8,75	12,14	8,71
09:19:20	8,75	12,14	8,71
09:19:30	8,75	12,14	8,71
09:19:40	8,75	13,87	8,71
09:19:50	8,75	13,87	8,71
09:20:00	8,75	13,87	8,71
09:20:10	8,75	13,87	8,71
09:32:00	14,01	24,27	3,48
09:32:10	14,01	24,27	3,48
09:32:20	14,01	24,27	3,48
09:32:30	14,01	24,27	3,48
09:32:40	14,01	24,27	3,48
09:32:50	14,01	24,27	3,48

Fonte: Do autor.

Com estas informações da Tabela 1 observa-se que há um intervalo de calibração para as células chegarem ao seu valor real devido ao cálculo da média das últimas medições na entrada para evitar ruídos. Observa-se também que às 09:19 as células chegaram a um ponto de equilíbrio, com uma variação pequena entre elas, tendo o seu pico encontrado na célula 2. Também pode-se concluir que a partir das 09:32 as células possuem uma grande diferença entre suas medições, o que pode ter sido ocasionado devido ao aumento do volume de chuva no momento final.

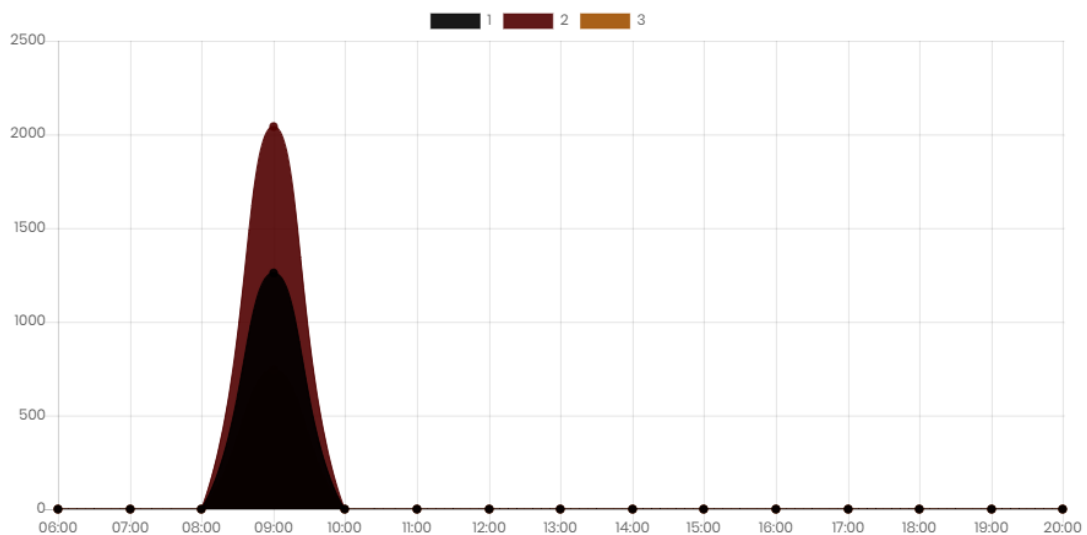
Ao carregar os dados para o sistema *web*, através dos gráficos pode-se observar que a célula 2 apresentou o maior valor registrado nas medidas devido à sua discrepância das demais nos últimos períodos da medição ao aumentar o volume de chuva. Essa informação pode ser encontrada nas Figuras 25, 26 e 27.

Figura 25 – Gráfico de barras do sistema *web* após as medições.



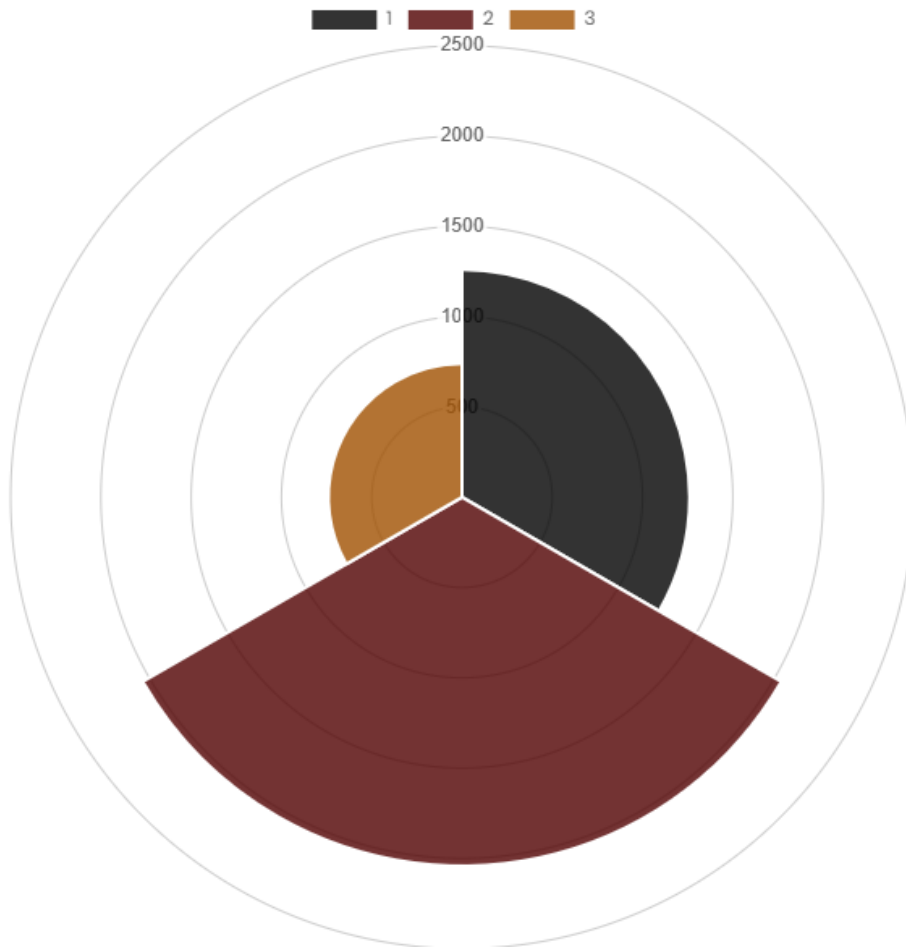
Fonte: Do autor.

Figura 26 – Gráfico de linhas do sistema *web* após as medições.



Fonte: Do autor.

Figura 27 – Gráfico de pizza do sistema *web* após as medições.



Fonte: Do autor.

6 Conclusão e Trabalhos Futuros

Este trabalho propôs o desenvolvimento de um protótipo de um sistema distribuído de aquisição de dados para armazenar os dados gerados por células fotovoltaicas de referência. Durante os testes observou-se que o sistema armazenou as informações corretamente, porém as primeiras medidas do dia devem ser desconsideradas por um usuário por conta da demora da estabilização devido ao filtro feito no *software* do sistema, além de demonstrar que em ambientes com chuva e muitas nuvens o valor medido do resistor *shunt* não é muito confiável por conta da sua alta variação.

As medições tomadas pelos sistemas não foram satisfatórias para a análise de dispersão dos dados devido ao pouco período de exposição e ao clima no dia das medições, porém consegue-se observar através da calibração que a medição de irradiância através do resistor *shunt* está correta, com seu armazenamento junto com a data e hora registrada pelo RTC acontecendo conforme planejado no cartão MicroSD.

Consegue-se concluir também que devido ao filtro do *software*, caso o valor da irradiância sofrer uma diferença brusca devido a passagem de alguma nuvem ou qualquer objeto sob a célula, haverá um período de estabilização da medida para que ela volte a ser confiável.

Pelos equipamentos serem de fácil acesso, com preço acessível e fácil montagem, o sistema se torna uma boa opção para futuras análises de dispersão, estudo com células de referência fotovoltaicas e outros estudos da área.

Para maior precisão nas informações de data e hora, seria interessante adicionar um módulo com conexão *Wi-Fi* para buscar da internet e armazenar no RTC o horário atual sempre que uma conexão conhecida estiver disponível, além de enviar para o servidor onde o sistema *web* está armazenado as informações coletadas de forma automatizada.

Também é possível a incrementação de um *Liquid Crystal Display* (LCD) para o usuário poder acompanhar os valores que estão sendo medidos pelo sistema embarcado em tempo real e poder atuar caso as medições forem discrepantes com a realidade.

Para melhorar a confiabilidade da irradiância armazenada, o Arduino Uno pode ser substituído por outra plataforma que possui maior precisão e confiabilidade para valores pequenos, além de ser feito uma placa de circuito integrado com melhor acabamento para evitar com que cabos e componentes sejam desconectados.

Outra melhoria interessante para o produto final seria aumentar a quantidade de gráficos e informações apresentados pelo sistema *web*, priorizando a criação de gráficos de desvios e adicionando análises feitas por inteligências artificiais buscando os melhores horários para a geração de energia e calibração das células, qual dia da semana seria mais interessante fazer as medições tendo como base a previsão do tempo da semana, entre outros tópicos que o usuário final poderia sugerir.

Os trabalhos futuros podem utilizar esta seção e todas as demais informações coletadas neste trabalho para estudos, além de poder alterar qualquer parte do desenvolvimento do *hardware*, banco de dados e sistema *web*.

Referências

- Hadi, M. S.; Afandi, A. N.; Wibawa, A. P.; Ahmar, A. S.; Saputra, K.H. **Standalone data logger for solar panel energy system with RTC and SD card**. Journal of Physics: Conf. Series 1028 012065, 2018.
- Dyreson, A. R.; Morgan, E. R.; Monger, S. H.; Acker, T. L. **Modeling solar irradiance smoothing for large PV power plants using a 45-sensor network and the Wavelet Variability Model**. Solar Energy, v. 110, pp. 482-495, 2014.
- Chowdhury, Z. I., Imtiaz, M. H., Azam, M. M., Sumi, M. R. A., & Nur, N. S. **Design and implementation of pyroelectric infrared sensor based security system using microcontroller**. IEEE Technology Students' Symposium, pp. 1-5, 2011.
- Gusa, R. F., Sunanda, W., Dinata, I., & Handayani, T. P. **Monitoring system for solar panel using smartphone based on microcontroller**. 2018 2nd international conference on green energy and applications (ICGEA), pp. 79-82, 2018.

Keogh, W. L.; Blakers, A. W. **Accurate measurement, using natural sunlight, of silicon solar cells.** Progress in Photovoltaics: Research and Applications, v. 12, n. 1, pp; 1-19, 2014.

Espinosa-Gavira, M. J. et al. **Cloud motion estimation from small-scale irradiance sensor networks: General analysis and proposal of a new method.** Solar Energy, v. 202, pp. 276-293, 2020.

Bakker, K. et al. **Comparison of statistical post-processing methods for probabilistic NWP forecasts of solar radiation.** Solar Energy, v. 191, pp. 138-150, 2019.

Gómez, J. M. et al. **A irradiância solar: conceitos básicos.** Revista Brasileira de Ensino de Física, v. 40, n. 3, 2018.

Rosso, A. P. et al. **Desenvolvimento de células e módulos de referência para medição da radiação solar.** 2019.

Gorender, S.; Macêdo, R. J. A. **Um modelo para tolerância a falhas em sistemas distribuídos com QoS.** Anais do Simpósio Brasileiro de Redes de Computadores, SBRC 2002, pp. 277-292, 2002.

ANEEL [Agência Nacional de Energia Elétrica]. Banco de Informações da Geração (BIG), 2017. Disponível em: <http://www.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm>. Acesso em: 04 mar. 2022.

Dewesoft. **O que é Aquisição de Dados – DAQ ou DAS?**, 2020. Disponível em: <http://www.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm>. Acesso em: 04 mar. 2022.

Alura. **Bootstrap - O que é, como e quando usar?**, 2021. Disponível em: <https://www.alura.com.br/artigos/bootstrap#:~:text=Bootstrap%20%C3%A9%20um%20framework%20front,dispositivos%20m%C3%B3veis%20da%20mesma%20forma>. Acesso em: 04 mar. 2022.

Absolar. **Energia fotovoltaica segue em plena expansão no Brasil**, 2022. Disponível em: <https://www.absolar.org.br/noticia/energia-fotovoltaica-segue-em-plena-expansao-no-brasil/>. Acesso em: 25 mar. 2022.

Arduino. **Arduino IDE**, 2021. Disponível em: <https://www.arduino.cc/en/software>. Acesso em: 04 mar. 2022.