



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE TECNOLOGIA



Lucas Barreto de Jesus
Lucas Tadeu de Carvalho Moreira

APLICABILIDADE DO MONITORAMENTO
REMOTO DE UTILIDADES EM UM AMBIENTE CONTROLADO

Limeira, SP

2023

Lucas Barreto de Jesus
Lucas Tadeu de Carvalho Moreira

**APLICABILIDADE DO MONITORAMENTO
REMOTO DE UTILIDADES EM UM AMBIENTE CONTROLADO**

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Engenheiro de Telecomunicações pela Faculdade de Tecnologia da Universidade Estadual de Campinas.

Orientador: Prof. Dr. Marco Aurélio Soares de Castro

Limeira, SP

2023

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Faculdade de Tecnologia
Mariana Xavier - CRB 8/9615

J499a Jesus, Lucas Barreto de, 2000-
Aplicabilidade do monitoramento remoto de utilidades em um ambiente controlado / Lucas Barreto de Jesus, Lucas Tadeu de Carvalho Moreira. – Limeira, SP : [s.n.], 2023.

Orientador: Marco Aurélio Soares de Castro.
Trabalho de Conclusão de Curso (graduação) – Universidade Estadual de Campinas, Faculdade de Tecnologia.

1. Telemetria. 2. Internet das coisas. 3. Sistemas embarcados (Computadores). 4. Estufas. I. Moreira, Lucas Tadeu de Carvalho, 1999-. II. Castro, Marco Aurélio Soares de, 1975-. III. Universidade Estadual de Campinas. Faculdade de Tecnologia. IV. Título.

Informações adicionais, complementares

Título em outro idioma: Applicability of remote monitoring of utilities in a controlled environment

Palavras-chave em inglês:

Telemetry

Internet of things

Embedded computer systems

Greenhouses

Titulação: Engenheiro de Telecomunicações

Banca examinadora:

Marco Aurélio Soares de Castro [Orientador]

Talia Simões dos Santos Ximenes

Marta Siviero Guilherme Pires

Data de entrega do trabalho definitivo: 30-11-2023

Este trabalho é dedicado àqueles que estiveram conosco ao longo dos anos de graduação, familiares, amigos e professores. Vocês têm sido fontes constantes de apoio, fornecendo-nos energia, confiança e conhecimento ao longo dessa jornada.

AGRADECIMENTOS

Expressamos aqui a nossa mais profunda gratidão a todos que estiveram ao nosso lado até aqui e que nos permitiram concluir esta importante etapa de nossas vidas.

Às nossas famílias, que sempre estiveram ao nosso lado com apoio constante e palavras de incentivo, dedicamos nossos mais sinceros agradecimentos. Vocês foram a nossa fonte de constante de força, iluminando os caminhos mais desafiadores.

À Universidade Estadual de Campinas, agradecemos sinceramente por proporcionar um ambiente acadêmico estimulante e pelos recursos fundamentais que tornaram possível a conclusão deste estudo.

Aos amigos e colegas que compartilharam ideias, ofereceram insights valiosos e estiveram presentes durante os desafios, nossa sincera gratidão. Cada conversa e colaboração contribuíram significativamente para a riqueza deste trabalho.

Em especial, queremos expressar nosso profundo agradecimento ao nosso professor orientador, Marco Aurélio, por aceitar a responsabilidade de guiar este trabalho. Suas contribuições valiosas foram cruciais para o seu desenvolvimento, e sua orientação foi fundamental para nosso crescimento acadêmico.

Por fim, estendemos nosso agradecimento a todos que desempenharam papéis vitais em nossas jornadas acadêmicas. Estamos sinceramente gratos pela colaboração e apoio de cada um de vocês.

*“If I know only one thing it's that everything that I see of the world outside is so
inconceivable. Often I barely can speak.” Robin Pecknold*

RESUMO

Este trabalho apresenta o desenvolvimento, a implementação e a análise de um sistema de telemetria que utiliza placas microcontroladoras ESP32 e uma rede de sensores para monitorar ambientes controlados, especificamente uma estufa. O principal objetivo do projeto é aprimorar a precisão e a eficiência na gestão das condições do ambiente a ser monitorado, coletando dados em tempo real sobre corrente elétrica, temperatura, umidade e luminosidade. Esses dados são transmitidos via internet e visualizados remotamente em uma dashboard. A revisão bibliográfica oferece uma análise de estudos anteriores relacionados e soluções de monitoramento baseadas em *Internet of Things* existentes para ambientes controlados. Através dessa revisão, identificamos tendências, avanços tecnológicos e lacunas no corpo de conhecimento atual, contextualizando assim a importância deste estudo. A análise de estudos semelhantes destaca a necessidade e diferenciais do sistema proposto, ressaltando as contribuições únicas e inovações que ele traz para o campo. A fase de implementação do projeto envolveu a integração de microcontroladores ESP32 com uma seleção de sensores disponíveis no mercado e de baixo custo. Os dados coletados são enviados para uma planilha em nuvem via integração com os serviços do Google, onde são processados e disponibilizados por meio de uma dashboard amigável desenvolvida no *Looker Studio*. Esta pesquisa enfatiza a importância de aproveitar a tecnologia *IoT* para melhorar ambientes controlados e apoiar o crescimento contínuo desse campo. Ela não apenas oferece uma solução prática para o monitoramento de um ambiente, mas também contribui para o amplo debate sobre aplicações de *IoT* em diversos campos, desde a pesquisa, ambientes residenciais até a indústria e agricultura. Oportunidades futuras de pesquisa e melhorias são destacadas, incentivando uma maior exploração e inovação neste domínio.

Palavras-chave: Telemetria. Internet das Coisas. Sistemas embarcados. Estufa de germinação.

ABSTRACT

This work presents the development, implementation, and analysis of an IoT telemetry system that utilizes ESP32 microcontrollers and a network of sensors to monitor controlled environments, specifically a greenhouse. The primary goal of the project is to enhance the accuracy and efficiency in managing the conditions of the monitored environment by collecting real-time data on electrical current, temperature, humidity, and luminosity. This data is transmitted via the internet and remotely visualized on a dashboard. The literature review provides an analysis of related previous studies and existing IoT-based monitoring solutions for controlled environments. Through this review, we identify trends, technological advancements, and gaps in the current body of knowledge, thereby contextualizing the significance of this study. The analysis of similar studies highlights the need and unique features of the proposed system, emphasizing its distinct contributions and innovations to the field. The project's implementation phase involved the integration of ESP32 microcontrollers with a selection of readily available and cost-effective sensors. The collected data is sent to a cloud-based spreadsheet through integration with Google services, where it is processed and made available through a user-friendly dashboard developed in Looker Studio. This research emphasizes the importance of leveraging IoT technology to improve controlled environments and support the continuous growth of this field. It not only offers a practical solution for monitoring a controlled environment but also contributes to the broader discussion about IoT applications in various fields, from research and residential settings to industry and agriculture. Future research opportunities and enhancements are highlighted, encouraging further exploration and innovation in this domain.

Keywords: Telemetry. Internet of Things. Embedded systems. Germination greenhouse.

LISTA DE ABREVIATURAS E SIGLAS

ARM	<i>Acorn RISC Machine</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
GPRS	<i>General Packet Radio Services</i>
GSM	<i>Global System for Mobile Communication</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IOT	<i>Internet of Things</i>
JSON	<i>JavaScript Object Notation</i>
LDR	<i>Light Dependent Resistor</i>
LTE	<i>Long Term Evolution</i>
M2M	<i>Machine to Machine</i>
MIMO	<i>Multiple Input Mutiple Output</i>
NTC	<i>Negative Temperature Coefficient</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
SD	<i>Secure digital</i>
SOC	<i>System on Chip</i>
SQL	<i>Structured Query Language</i>
SRAM	<i>Static Randon-aces Memory</i>
TC	<i>Transformador de Corrente</i>
VCC	<i>Common Collector Voltage</i>
WLAN	<i>Wireless LAN</i>

LISTA DE FIGURAS

Figura 1 - Esquema de funcionamento básico de um sistema de telemetria.....	16
Figura 2 - Módulo ESP32	17
Figura 3 - Sensor de corrente SCT-013	18
Figura 4 - Funcionamento de um transformador de corrente.....	19
Figura 5 - Sensor de temperatura DHT11	20
Figura 6 - Módulo LDR.....	20
Figura 7 - Topologia do protocolo ESP-NOW	21
Figura 8 - Espectro de canais da banda de 2,4 GHz.....	22
Figura 9 - Módulo TP4056	26
Figura 10 - Circuito desenvolvido para o protótipo	27
Figura 11 - Ligação circuito sensor DHT11	28
Figura 12 - Ligação circuito Sensor SCT-013	28
Figura 13 - Primeiro teste Sensor SCT-013	29
Figura 14 - Diagrama de blocos da programação do ESP32	30
Figura 15 - Rural Max.....	32
Figura 16 - Diagrama ER da base de dados desenvolvida	35
Figura 17 - Dashboard desenvolvida	36
Figura 18 - Estufa de germinação	37
Figura 19 - Protótipo instalado na 1º estufa.....	38
Figura 20 - Protótipo instalado na 2º estufa.....	39
Figura 21 - Dashboard da 1ª estufa em 21 de outubro de 2023	41
Figura 22 - Dashboard da 1ª estufa em 28 de outubro de 2023	41
Figura 23 - Dashboard da 2ª estufa em 04 de novembro de 2023.....	42
Figura 24 - Comparação de corrente da estufa 1 (cima) e estufa 2 (baixo)	43

SUMÁRIO

1 INTRODUÇÃO	11
2 OBJETIVOS	13
3 REVISÃO BIBLIOGRÁFICA	14
3.1 Internet das Coisas	14
3.1.1 Telemetria	15
3.1.2 Sistemas embarcados	16
3.1.3 ESP32	17
3.2 Métodos de medidas (sensores)	18
3.2.1. SCT -013	18
3.2.2 DHT11	19
3.2.3 Módulo LDR	20
3.3 Protocolos de comunicação.....	21
3.3.1 ESP-NOW	21
3.3.2 Wi-Fi (IEEE 802.11 b/g/n)	22
3.4 Interface do usuário	22
3.4.1 Google Spreadsheets	23
3.4.2 Looker Studio	23
3.5 Trabalhos semelhantes	23
4 MATERIAIS E MÉTODOS	26
4.1 Estrutura do sistema.....	26
4.1.1 Sensor de Temperatura e umidade	27
4.1.2 Sensor de Corrente	28
4.1.3 Programação do sistema	29
4.1.4 Conexão com a internet	31
4.1.5 Armazenamento dos dados	32
4.1.6 Desenvolvimento da Dashboard	35
4.2 Prova de conceito	36
4.2.1 O ambiente a ser monitorado	37
4.2.2 Montagem do sistema no ambiente	38
5 RESULTADOS E DISCUSSÕES	40
6 CONCLUSÃO	45
BIBLIOGRAFIA	47
ANEXO A - Código de programação do ESP32	51

1 INTRODUÇÃO

Diante de crises econômicas eclodindo mundo afora, crise energética, guerras, aumento populacional e aquecimento global, a humanidade vê-se diante de uma crescente necessidade de controlar, monitorar e entender como tem sido consumido os recursos energéticos e as suas respectivas perdas. Não à toa é observado uma busca incessante por eficiência em todas as áreas imagináveis, desde a eficiência energética, otimização de produção, redução dos poluentes emitidos, evolução na ciência dos materiais, entre outros (LUCON; GOLDEMBERG, 2009).

As tecnologias emergentes possuem um papel essencial nessa busca por renovação na maneira que a humanidade consome seus recursos (MARTINS, 1999). Sistemas embarcados conectados à Internet, hoje, são peças-chave em grandes indústrias, havendo uma grande extensão para outras vertentes como o uso residencial, mobilidade urbana, ensino e na agricultura. Entretanto, essas aplicações normalmente possuem um viés de atuação, de executar uma determinada tarefa e muitas vezes um fator extremamente importante é deixado de lado: a coleta, armazenamento e análise dos dados que tais objetos/sensores são capazes de gerar.

William Edwards Deming, estatístico amplamente conhecido pelos seus estudos na área de qualidade e de melhoria de processos, possui estudos que visam a aplicação de métricas para avaliação de desempenho, processos de melhorias de estratégia e qualidade (SILVA; BERNARDO *et al*, 2016). Seus estudos iniciam-se na década de 1950, portanto, é um campo estudado há várias décadas, mas que vem se adaptando ainda mais a realidade dos dias atuais enquanto há uma crescente demanda para a implantação de monitoramento e de processos de melhoria em todas as vertentes da sociedade (serviço, indústria, mobilidade, esporte, educação, etc).

No atual mercado existem inúmeras aplicações para soluções *de Internet of Things* (DATTA; SHARMAN, 2017) e cada aplicação possui seus parâmetros limitantes que devem ser aplicados para cada cenário ou segmento, por exemplo, existe um segmento de automação voltada para a comunicação entre máquinas, denominada *M2M (Machine-to-Machine)* (VERMA; VERMA; AGRAWAL *et al*, 2016). De modo geral, essa tecnologia possui protocolos desenvolvidos para realizar interface e integração entre as aplicações desenvolvidas por uma mesma marca, ou seja, de certa forma as empresas criam protocolos de comunicação próprios, ditos fechados, que só os dispositivos desenvolvidos por ela própria reconhecem o

sistema aumentando o nível de interação do sistema. Por outro lado, no mercado de microcontroladores e sistemas embarcados podemos observar várias plataformas que possuem seu próprio sistema de integração de protocolo aberto (*Open Source*) como o Arduíno e *Raspberry Pi*, não limitando as soluções a protocolos exclusivos de determinados fornecedores (SOURCEFORGE, 2022).

Na borda de todo sistema de monitoramento encontram-se sensores, dispositivos essenciais que realizam a conversão de grandezas físicas em grandezas elétricas, que posteriormente serão transformadas em bits. Uma vez conectado a qualquer dispositivo capaz de concentrar os sinais coletados, os sensores darão o primeiro passo ao coletar um dado que posteriormente será tratado e armazenado em uma base de dados (SHEN; LIU, 2011). A comunicação com a base de dados pode ser em tempo real ou os dados coletados podem ser encapsulados por um tempo pré-definido para que, após esse tempo, um grande pacote de dados seja enfim gravado na base de dados. Ambos os cenários são plausíveis em sistemas de monitoramento e a escolha dependerá das exigências definidas pelos projetistas de sistemas. Uma vez que esses dados estejam gravados e disponíveis, é possível gerar uma dashboard capaz de apresentar os números consolidados, visões gráficas, relatórios históricos e outras ferramentas que facilitem a análise de dados e a geração de *insights* para os processos de otimização em geral.

Dessa forma, guiado pelo contexto exposto, este trabalho visa aprofundar em termos práticos e teóricos em um sistema de telemetria IoT capaz de embarcar diversos sensores que oferecerão dados a serem publicados em um ambiente Web. Toda essa aplicação será destinada a um ambiente controlado, onde é necessário haver um controle minucioso acerca da variação de grandezas físicas como potência elétrica, temperatura, luminosidade e umidade.

2 OBJETIVOS

A partir do contexto apresentado, nasce a motivação para o desenvolvimento de um trabalho de pesquisa que possa abordar o papel das telecomunicações e gestão de recursos para o aprimoramento da estrutura de monitoramento de recursos e/ou grandezas visando o ganho de eficiência e redução de gastos, tanto com o viés teórico quanto o viés prático.

O objetivo deste trabalho foi o aprofundamento acerca de sistemas de telemetrias, suas aplicações em diferentes segmentos e como pode ser usado como uma ferramenta de muita importância na diminuição de gastos de energia elétrica, otimização de produção e outros benefícios. Também foi proposto e testado um protótipo de sistema de telemetria capaz de monitorar parâmetros operacionais de uma estufa de cultivo de microrganismos para controle toxicológico de solos através de uma dashboard em ambiente *Web*.

3 REVISÃO BIBLIOGRÁFICA

Visando o desenvolvimento e entendimento teórico do sistema que esse trabalho propõe, é necessário um aprofundamento teórico dos conceitos a serem abordados para sustentar sua importância, viabilidade e praticidade. Seguindo isso, nessa seção é realizada uma breve revisão bibliográfica acerca dos conceitos e sistemas mais importantes desse projeto, envolvendo esclarecimento relacionado ao que é a Internet das Coisas (*IoT*) e suas derivações, além de uma breve explicação do funcionamento e especificidades dos sistemas e sensores utilizados. Também, ao final, é realizada uma pequena análise comentando de alguns trabalhos semelhantes disponíveis na literatura.

3.1 Internet das Coisas

Para o desenvolvimento do presente trabalho é imprescindível revisitar e aprofundar os conhecimentos acerca do conceito de *Internet of Things (IoT)* e como esse paradigma transforma o desenvolvimento de novas tecnologias e permite que trabalhos como este sejam desenvolvidos. Esse conceito propõe que coisas físicas como objetos, eletrodomésticos, pequenas máquinas ou aparelhos estejam conectadas entre si e troquem informações pela internet, preferencialmente através de uma rede sem fio como *Wi-Fi*, rede celular, *Bluetooth*, entre outras, trazendo uma capacidade computacional a essas “coisas” do cotidiano.

As tecnologias que surgem a partir do conceito de *IoT* têm aumentado consideravelmente nos últimos anos devido à demanda crescente de conectar diversos dispositivos, equipamentos e ferramentas à internet a fim de monitorar ou automatizar algum processo ou tarefa. Tal demanda de digitalização pode ser encontrada em indústrias, transporte, áreas de serviços gerais, hospitais, ou até mesmo dentro das residências e impacta diretamente a gestão e comportamento dentro desses ambientes. Além disso, é beneficiada pelo esforço de uma grande comunidade colaborativa que se formou para desenvolver e compartilhar conhecimentos sobre novas tecnologias, incluindo desde pesquisadores e pessoas que estudam isso por hobby, surgindo daí diversas iniciativas de softwares e hardwares *OpenSource* (SINGH; KAPOOR, 2017).

Sendo um conceito que visa a conexão de todas as “coisas”, o *IoT* acaba exigindo que as tecnologias cumpram alguns requisitos importantes como a confiabilidade na comunicação remota e na transferência de dados via rede sem fios, terem um baixo custo agregado, possuir um pequeno tamanho físico, serem eficientes energeticamente e possuírem uma considerável capacidade de processamento (MAIER *et al*, 2017).

No mercado atual há uma enorme variedade de placas e equipamentos orientados para *IoT* que são desenvolvidos pelas mais diversas empresas, sejam elas *OpenSource* ou não, dentre elas podemos citar o *Arduino*, *Raspberry*, *Cypress*, *Xbee*, *WhizFi*, *Espressif* (desenvolvedora da ESP32, placa microcontroladora que será utilizado no desenvolvimento do presente trabalho), entre diversas outras.

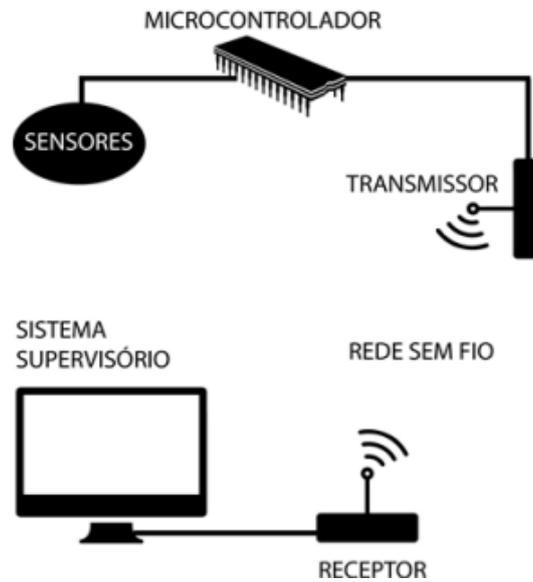
3.1.1 Telemetria

Telemetria, que vem do grego *teles* (longe) e *metron* (medida), é uma técnica de obtenção de dados que, através de tecnologias de medições somada a conexão a uma rede de comunicação possibilita medição remota, seja em tempo real ou não, de diversos tipos de grandezas (PIOVESAN, 2008).

Há muito tempo a telemetria tem sido uma importante ferramenta utilizada para monitoramento de diversas grandezas, ambientes, máquinas e status, porém, ganhou especial importância e avanços juntamente com o provento da Internet das Coisas, aumentando consideravelmente as áreas de aplicação, possibilitando, por exemplo, o desenvolvimento de sistemas *machine to machine (M2M)* graças a melhora da conexão com a internet, aumento da capacidade de processamento e do poder de processamento e armazenagem de dados. A utilização desses sistemas remotos surgiu da necessidade de se realizar medições em locais pouco acessíveis ao homem de maneira facilitada (GOMES, 2016).

A Figura 1 mostra um esquema básico de um sistema de telemetria *IoT*, onde há um microcontrolador responsável pela aquisição de dados e enviando-os para um sistema supervisor, que pode ser também um servidor *Web*, através de um protocolo de comunicação sem fio.

Figura 1 - Esquema de funcionamento básico de um sistema de telemetria



Fonte: (GOMES, 2016)

3.1.2 Sistemas embarcados

Um importante conceito a ser esclarecido para pleno entendimento deste trabalho é o de Sistemas Embarcados. Segundo a literatura, um sistema é dito embarcado quando este é projetado para executar uma determinada tarefa e para interagir com o ambiente através de sensores e atuadores de maneira contínua (CHASE, 2007).

A nomenclatura “embarcado” surge do fato de que, normalmente, esses sistemas estão integrados dentro de um outro sistema ou aplicação, não possuem uma fonte de energia fixa dedicada para ela e, ainda assim, terem uma unidade de processamento própria. Pelas características citadas, é de se esperar que um sistema embarcado possua tamanho e peso reduzido, tenha um baixo consumo de energia, tenha robustez perante o ambiente, tenha uma boa capacidade de processamento e comunicação (CUNHA, 2007).

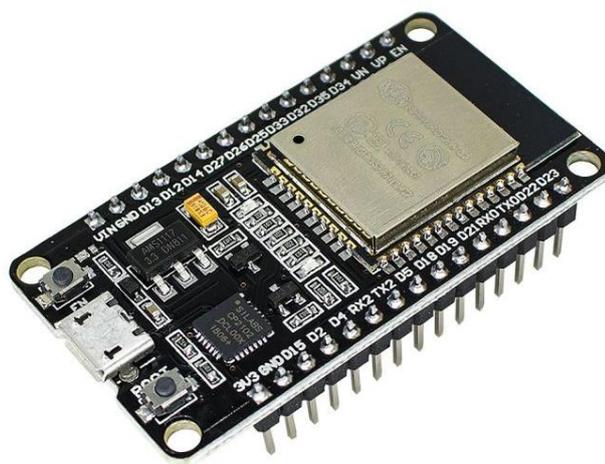
A ESP32, sistema a ser discutido no próximo tópico desta seção, é um exemplo de sistema embarcado, pois é um microcontrolador que já possui diversos módulos, sensores e transmissores embutidos em uma mesma placa.

3.1.3 ESP32

Desenvolvido pela companhia chinesa *Espressif Systems*, a ESP32 é um SOC (*System On Chip*) que possui um microcontrolador Tensilica Xtensa LX6 integrado, com dois núcleos de processamento, 520 kB de *SRAM*, além de módulos *Wi-Fi* (IEEE 802.11) e *Bluetooth 4.2*. Possui 36 pinos digitais de *Input* e *Output* e 4MB de memória *flash* (expansível com auxílio de módulos *SD Card*). Esse dispositivo é sucessor do ESP8266 e possui uma série de variações de arquiteturas que podem incluir módulos de expansão de memória, *displays*, baterias, *ethernet* e outros periféricos acoplados (MAIER, 2017). Sua arquitetura mais básica é a mesma apresentada na Figura 2.

É uma robusta, e ainda assim pouco custosa e de fácil configuração, opção de microcontrolador que pode ser usado em prototipagem de aplicações de Internet das Coisas baseado em *Cloud*, automações, sensoriamento, comunicação M2M, e outros campos. Pode ter códigos facilmente gravados em seu núcleo através do ambiente de desenvolvimento da plataforma Open Source Arduino, que utiliza a linguagem de programação *Wiring*, que possui muita similaridade com o *framework* C++. Por fim, além dos benefícios apontados anteriormente, o ESP32 é um sistema de baixo potência elétrica, o que o torna uma ótima opção para projetos que visam eficiência de recursos (BABIUCH *et al*, 2019).

Figura 2 - Módulo ESP32



Fonte: (SARAVATI, 2023)

3.2 Métodos de medidas (sensores)

Nesta seção será discutido o funcionamento dos sensores responsáveis pela medição das grandezas elétricas, temperatura, umidade e de luminosidade a serem estudadas no decorrer do projeto proposto pelo presente trabalho.

3.2.1. SCT-013

O sensor SCT-013 é um Transformador de Corrente (TC) de relação de 100A para 5mA. Esse dispositivo, mostrado na Figura 3, tem por objetivo reduzir a corrente de um sistema a um nível adequado para que seja medido por instrumentos que possuam esse fim de modo a manter a relação fixa e proporcional entre o valor de corrente de um circuito com as saídas do sensor (PONCHET, 2016).

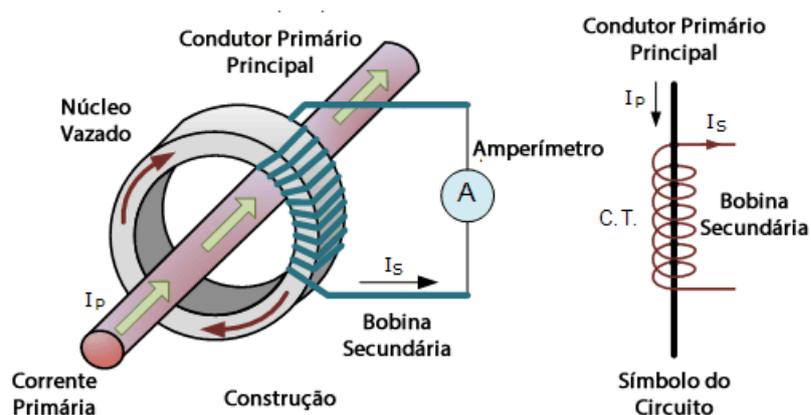
Figura 3 - Sensor de corrente SCT-013



Fonte: (ARDUROBOTICA, 2023)

A Figura 4 ilustra o funcionamento de um transformador de corrente, onde podemos observar a espira (N) primária e secundária em um núcleo de alta permeabilidade magnética:

Figura 4 - Funcionamento de um transformador de corrente



Fonte: (AUGUSTO, 2018)

Dessa forma, a relação de corrente entre o primário e secundário é:

$$\frac{I_p}{I_s} = \frac{N_p}{N_s} \quad (1)$$

Através do valor de corrente obtido por esse sensor, será possível calcular a potência ativa e consumo energético do circuito elétrico monitorado.

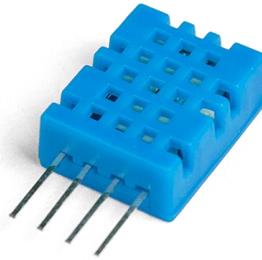
3.2.2 DHT11

O DHT11, apresentado na Figura 5, é um sensor integrado capaz de mensurar a temperatura e a umidade relativa do ar do ambiente em que estiver inserido (SRIVASTAVA *et al*, 2016).

A umidade relativa do ar está relacionada com a quantidade de vapor de água presente no ar em relação a quantidade máxima de vapor que o ar suportaria na temperatura atual. Para detectar esses valores, o sensor possui um substrato que possui dois eletrodos em sua superfície e tem a capacidade de reter umidade. A variação da resistência entre esses dois eletrodos é proporcional a umidade relativa do ar (FERESU, 2022).

Para monitorar temperatura, o sensor utiliza de um termistor NTC capaz de variar a resistência elétrica do dispositivo conforme a temperatura do ambiente monitorado aumenta ou diminui (FERESU, 2022).

Figura 5 - Sensor de temperatura DHT11

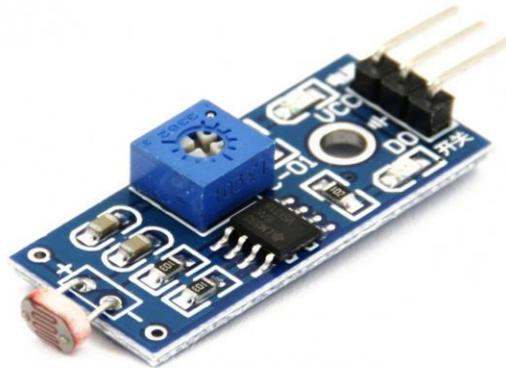


Fonte: (ELETROGATE, 2023)

3.2.3 Módulo LDR

O módulo LDR é um pequeno circuito eletrônico que possui um sensor semicondutor LDR (Resistor dependente de luz) acoplado com uma alta resistência. Apresentado na Figura 6, esse sensor varia sua resistência elétrica (variando entre $10\text{k}\Omega$ e $1\text{M}\Omega$) conforme há incidência de luz no dispositivo. Luminosidade e resistência são inversamente proporcionais, logo, quando há pouca luz, a resistência é alta e vice-versa (SILVA, 2020).

Figura 6 - Módulo LDR



Fonte: (HUINFINITO, 2023)

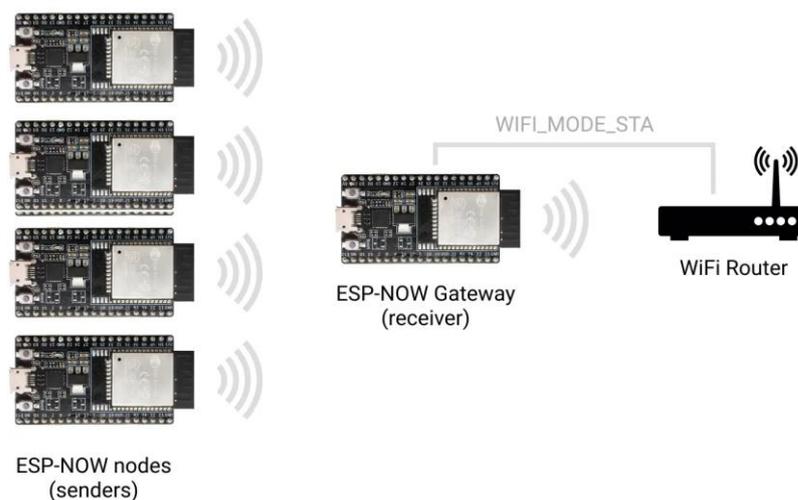
3.3 Protocolos de comunicação

Acompanhando os avanços tecnológicos já citados anteriormente, as comunicações móveis e seus respectivos protocolos também apresentaram novos e diversos desenvolvimentos dedicados a diversas aplicações, dentre esses protocolos alguns recentes que ganharam destaque são o *LTE (4G)*, *5G* e *LoRaWan*. Especialmente para este trabalho, iremos aprofundar nos protocolos *ESP Now* e *Wi-Fi*, que são de suma importância para o funcionamento do sistema aqui proposto.

3.3.1 ESP-NOW

O *ESP-NOW* é um protocolo *peer-to-peer* utilizado para realizar comunicações entre placas *ESP32* sem a necessidade de um roteador ou de outro protocolo de rede. Como visto na Figura 7, nesse protocolo um dispositivo *ESP32* exerce o papel de *gateway/Master* enquanto os demais são os *endpoints/slaves* do sistema. Neste modelo, apenas o master necessitará realizar uma conexão com a internet via *Wi-Fi* para postar os dados na rede.

Figura 7 - Topologia do protocolo ESP-NOW



Fonte: (CARRASCO, 2021)

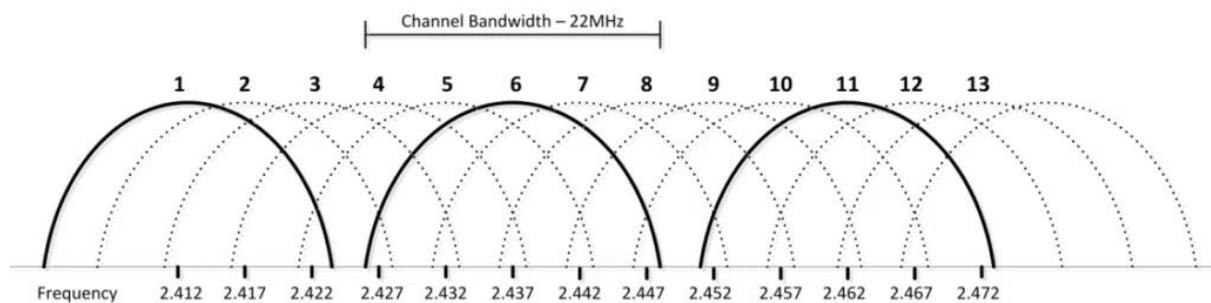
A rede de comunicação *ESP-NOW* pode comunicar com todos os dispositivos através de *broadcast*, *unicast* ou *multicast* com uma taxa de dados de 1Mbps. Além disso, por se tratar de uma comunicação *peer-to-peer* de baixo consumo energético, o *ESP-NOW* é um protocolo mais eficiente energeticamente do que o *Wi-Fi* e de fácil implantação (HOANG et al, 2019).

3.3.2 Wi-Fi (IEEE 802.11 b/g/n)

O protocolo *Wi-Fi*, também conhecido como padrão IEEE 802.11, é um protocolo de comunicação que atende dispositivos de uma rede local sem fio (WLAN). O termo se origina das palavras *Wireless Field*, ou “área sem fio” em português. Para a criação de uma rede *Wi-Fi* é necessária a utilização de um dispositivo roteador que criará a rede sem fio.

Sua primeira versão, desenvolvida em 1997, utiliza a modulação DSSS (*Direct Sequence Spread Spectrum*) em canais de 22 MHz de largura na banda de 2,4 GHz. Nessa banda existem 3 canais sem sobreposição, de forma que, se houver mais de 3 dispositivos conectados à rede, haverá interferência entre eles conforme mostra a Figura 8. Devido a essa razão, o padrão utiliza os canais 1, 6 e 11 (EMBARCADOS, 2021).

Figura 8 - Espectro de canais da banda de 2,4 GHz



Fonte: (EMBARCADOS, 2021)

Desde o surgimento, diversas novas versões deste protocolo têm sido desenvolvidas. Os dispositivos ESP32 utiliza especificações as versões 802.11b, g e n, sendo a versão “n” a mais recente e possuindo as taxas de dados até 54Mbps e portabilidade com multiplexação OFDM (*Orthogonal Frequency-division multiplexing*) utilizando MIMO (*Multiple Input - Multiple Output*).

3.4 Interface com o usuário

Uma essencial parte deste e de outros projetos de tecnologia que envolva fornecimento de informações e insights ao homem é a interface por meio da qual esses dados serão exibidos. Uma visualização coesa e facilitada dos dados é de suma importância para que os operadores de um sistema possam agilidade e precisão na operação e monitoramento dos dados de grandezas que estão sendo acompanhadas pelo sistema de sensores.

Durante o desenvolvimento do projeto foram elencadas duas ferramentas de fácil acesso e manuseio que estão disponíveis a todos os alunos da Unicamp através das ferramentas disponibilizadas pelo *Google for Students*, que são o *Google Spreadsheets* e o *Looker Studio*.

3.4.1 Google Spreadsheets

O Google oferece a possibilidade de integrar funcionalidades do “Planilhas” com outras aplicações via API, de maneira a tornar possível que um ou mais dispositivos ESP32 sejam capazes de gravar dados coletados por sensores e atuadores diretamente em uma tabela em nuvem de maneira muito simplificada, mas ainda assim com robustez e confiabilidade. Toda essa integração pode ser feita via *HTTP* e arquivos *JSON* através do *Google Scripts* (IOTDESIGN PRO, 2022).

3.4.2 Looker Studio

Anteriormente conhecido como *Google Data Studio*, este é um serviço de visualização de dados do pacote de serviços em nuvem do Google que foi desenvolvido em 2016 com o objetivo de ser uma ferramenta de fácil manuseio para criação de dashboards dinâmicas voltadas para a análise de dados (SNIPES, 2018).

Esta ferramenta possui fácil integração com *MySQL* e *Google Spreadsheets*, o que torna uma ferramenta muito viável para a criação das dashboards gráficas dos dados a serem coletados no projeto a ser discutido neste trabalho, conforme será apresentado com detalhes nas seções seguintes.

3.5 Trabalhos semelhantes

Na literatura é possível encontrar trabalhos sobre sistemas de telemetria para monitoramento de ambientes controlados desde o início dos anos 2000. Apesar disso, foram nos últimos 10 anos que se desenvolveram muitos projetos baseados em *IoT*, com diversos tipos de tecnologias de comunicação, formas de coleta de dados e aplicações. Dessa forma, a seguir serão descritos alguns trabalhos anteriores dentro desse campo de estudo, com comentários em pontos pertinentes relacionados ao presente trabalho.

Litjens (2009) desenvolveu um projeto que visava modernizar estufas agrícolas através de sensores e automação ligadas por um módulo sem-fio Xbee, dispositivo que opera pelo protocolo de rede ZigBee. Com esse sistema, foi possível realocar sensores livremente por toda a extensão da estufa, provando a robustez da área de cobertura sem fio criado pelo módulo Xbee, além de ter obtido sucesso na detecção de alterações nas grandezas e objetos monitorados remotamente.

Barbosa e Martins (2014) apresentaram em seu trabalho a proposta de um sistema de telemetria para monitoramento em uma casa-de-vegetação. Neste foi desenvolvido um circuito de hardware coletar dados de um termohigrômetro via porta serial RS-232 e enviar esses dados sem interferências e com raio de até 120 metros para um outro circuito receptor via rádio frequência em 433MHz. Os dados são visualizados diretamente em um software instalado em um computador que esteja dentro desse raio de cobertura.

O trabalho de Melo e Vieira (2016) consistiu em um projeto de sistema de telemetria capaz de coletar dados, publicá-los em um banco de dados em nuvem e tratá-los através de uma interface de *Data Analytics*. Esse sistema foi desenvolvido com o intuito de se aplicar na indústria de manufatura de couro e enfatiza o conceito de *IoT* como uma fonte de tecnologias inovadoras. Neste trabalho é utilizado um processador ARM compatível com sistema operacional Linux para desenvolver uma arquitetura capaz de comunicar com a internet via Rede *Wi-Fi* ou ZigBee.

O sistema criado por Zanuzzo (2017) teve foco em monitoramento de vazão de água visando encontrar oportunidades de economia em um ambiente residencial através de um aplicativo móvel desenvolvido para o sistema operacional Android. O sensor de medição é conectado ao mini-computador *Raspberry Pi 3 Model B* que armazena os dados em um banco de dados em *SQLite* e os envia ao aplicativo móvel através do protocolo Bluetooth.

Pereira (2021) traz uma proposta de sistema de telemetria aplicado à metrologia. Com o auxílio de uma plataforma Web e o desenvolvimento de um aplicativo Android, foi possível monitorar remotamente um termômetro digital utilizado em um laboratório de sua universidade. Esse sistema permitiu que os pesquisadores do laboratório não precisassem mais realizar medições manuais da temperatura do ambiente, uma vez que as mesmas poderiam ser encontradas armazenadas em nuvem.

Lafratta *et al* (2021) propuseram monitorar remotamente um processo de compostagem através de aplicações de Internet das Coisas. Com o auxílio de sensores e atuadores conectados a um controlador ESP8266, foi possível monitorar a umidade e temperatura durante um ciclo de 21 dias de uma composteira em tempo real através da plataforma *ThingSpeak*. O sistema permitiu verificar com clareza o comportamento das fases de um processo de compostagem e auxiliar na otimização do desempenho do mesmo.

Cabral (2023) criou uma arquitetura de sistema para monitoramento remoto de dados de temperatura e umidade do ar que fosse independente de uma rede local para se conectar à Internet, dessa forma, o pesquisador utilizou as redes celulares GSM e GPRS (2G) para enviar os dados até um servidor Web. Para o controle desse sistema e compilação dos dados, foi utilizada uma placa Arduino UNO. Além disso, o usuário do sistema possui a possibilidade de receber alertas de alterações ou ultrapassagem de limites das grandezas monitoradas através de SMS.

Com a análise das referências mencionadas é possível notar que os autores focam em monitorar grandezas específicas utilizando formas de coleta de dados e protocolos de comunicação que - na maioria das vezes - são semelhantes. Dessa forma, o presente trabalho visa agregar vários sensores em um mesmo sistema, trazendo uma visão ainda mais completa acerca do ambiente monitorado, com destaque para o monitoramento de grandezas elétricas, que não foi abordado em nenhum dos trabalhos citados.

4 MATERIAIS E MÉTODOS

Nos projetos citados na seção anterior observa-se algumas das principais evoluções de cada sistema ao longo do tempo, algumas sendo amplamente utilizadas nos dias de hoje. Dentro do contexto atual e do cenário das tecnologias, uma que se destaca é a conectividade em ambientes controlados que, em muitas vezes, possuem diversas restrições devido ao cenário de riscos ou nível de adversidade local.

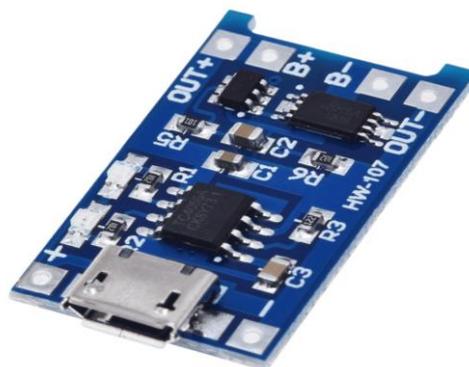
Dessa forma, nesta seção será descrito o desenvolvimento de um projeto de telemetria que é capaz de monitorar dados de corrente elétrica, temperatura, umidade e luminosidade, com possibilidade de expansão para monitorar outras grandezas, em tempo real via uma dashboard online. Para esse desenvolvimento será utilizada a plataforma embarcada do ESP32 e seu protocolo de comunicação próprio *ESP-NOW*, já abordado no presente trabalho.

4.1 Estrutura do sistema

A estrutura do sistema desenvolvido consiste em duas partes principais, sendo a primeira responsável pelo sistema de alimentação da placa ESP 32 e a segunda são os circuitos independentes dedicados para o funcionamento correto de cada sensor instalado.

A primeira parte é responsável pela alimentação do sistema. Para simplificar foi utilizado o módulo TP4056, mostrado na figura 9. Ele é utilizado para regular a corrente e tensão que é enviada para o módulo central do ESP-32 ao mesmo tempo que é um sistema que possibilita também a operação através de sistemas de baterias, tornando a solução mais versátil e implementável em qualquer lugar, mesmo que não haja uma fonte de alimentação constante.

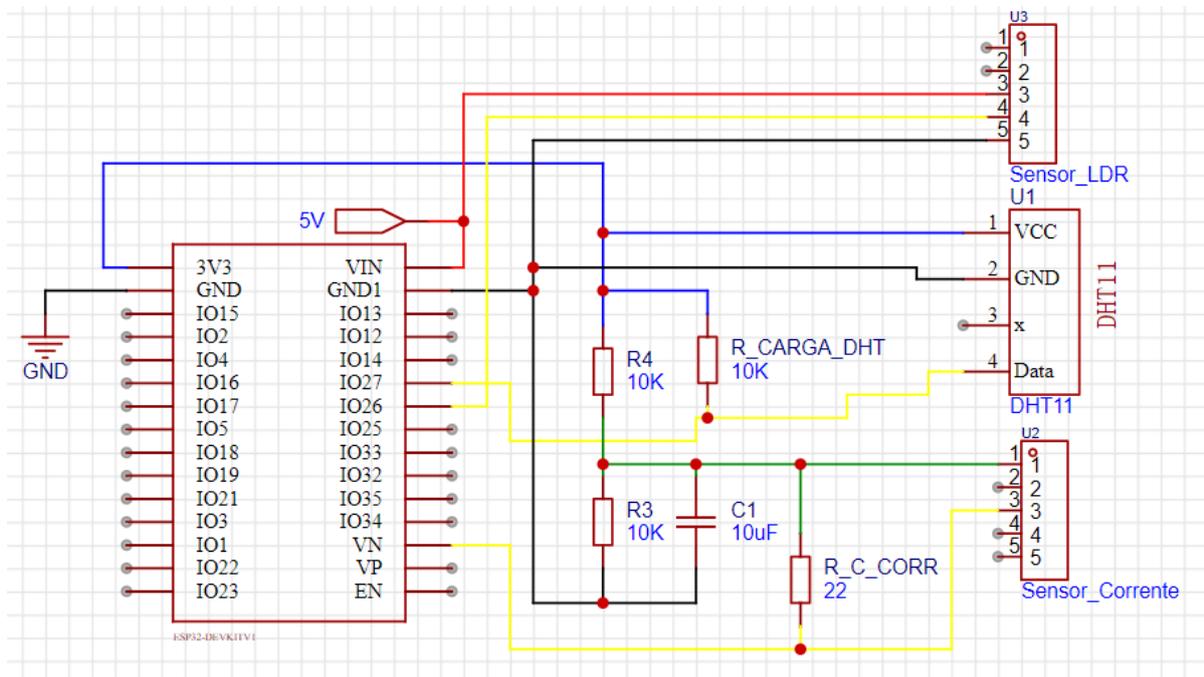
Figura 9 - Módulo TP4056



Fonte: (ELETROGATE, 2021)

A segunda parte é responsável pela captura de informações através dos sensores, no qual estão anexados às portas analógicas do ESP32 para a leitura pelo dispositivo. Observamos na Figura 10 o esquema geral de funcionamento da segunda parte do sistema.

Figura 10 - Circuito desenvolvido para o protótipo



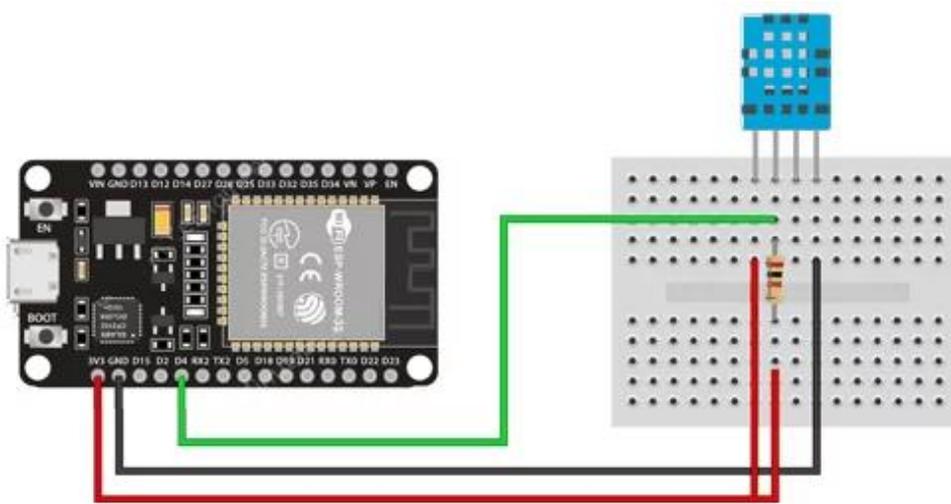
Fonte: Autoria própria

Para cada um dos sensores associados ao módulo é necessária a construção de um circuito para adaptar para os parâmetros lidos pelas portas analógicas da placa. Nessa implementação, foram utilizados circuitos eletrônicos para cada um dos sensores de temperatura, umidade e corrente, enquanto o sensor de luminosidade, por se tratar de um módulo integrado, já possui o circuito próprio construído para implementação direto nas portas analógicas do ESP32.

4.1.1 Sensor de Temperatura e umidade

Em se tratando do sensor de temperatura e umidade, o circuito necessário para operação é apenas um resistor de carga que conecte a entrada 5 Volts ou o VCC ao sinal de saída do sensor. Em seguida, o sinal de dado é conectado saindo direto no dispositivo medição de temperatura e umidade foi conectado na porta Analógica 27 do ESP32, conforme a Figura 11.

Figura 11 - Ligação circuito sensor DHT11

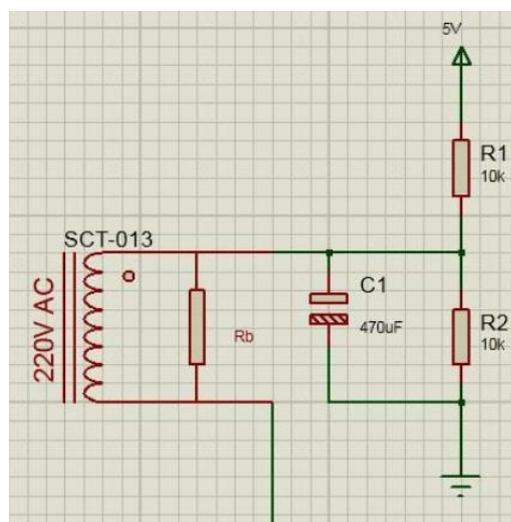


Fonte: Autoria própria

4.1.2 Sensor de Corrente

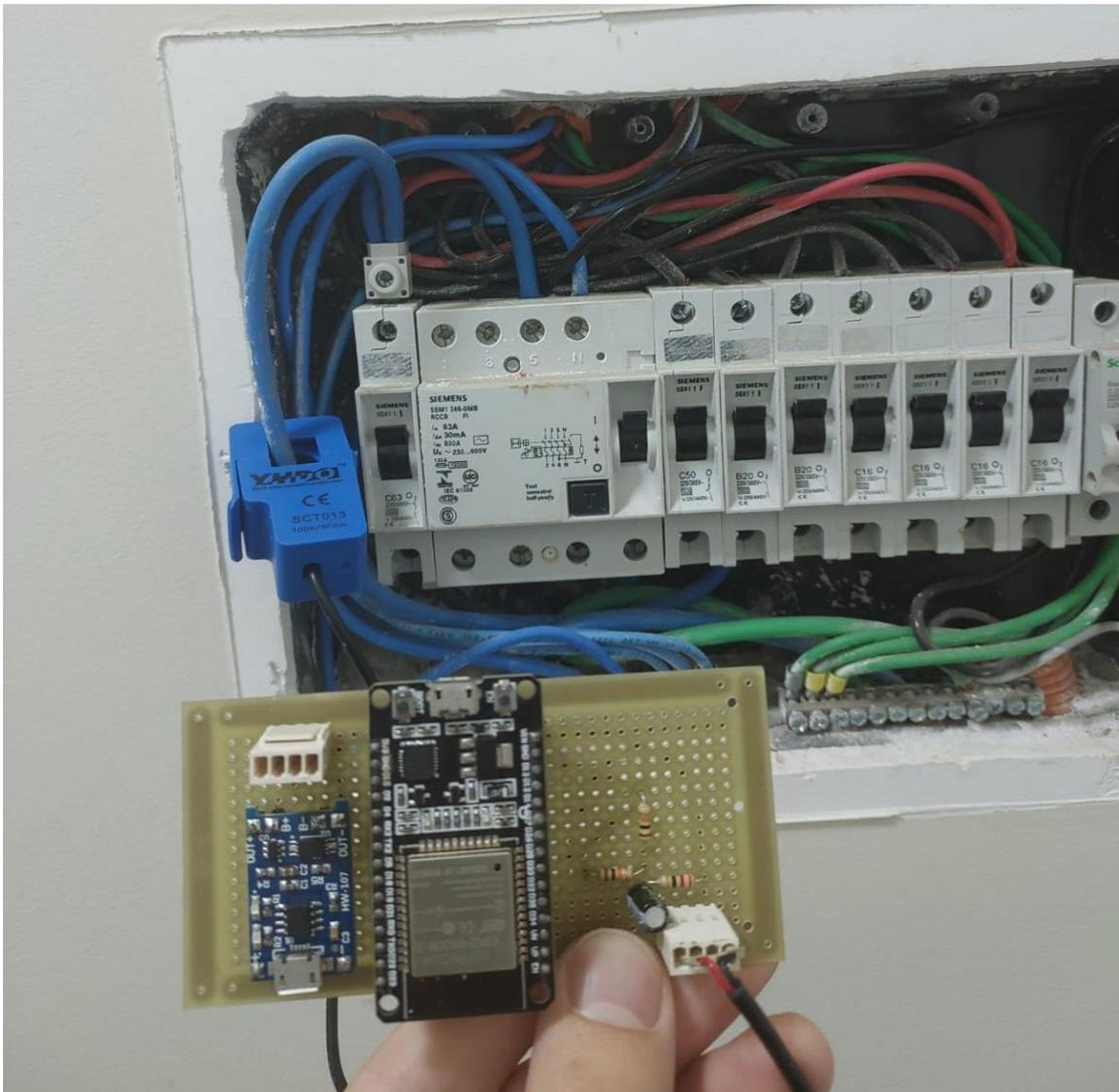
Para o sensor de corrente, observa-se que a composição da estrutura é extremamente enxuta e pode ser utilizada para ampliação em futuros trabalhos aplicados com a mesma plataforma de controle ESP32. A Figura 12 ilustra as ligações que foram necessárias para conexão deste sensor na placa desenvolvida e a Figura 13 apresenta a instalação do sensor em um quadro de distribuição residencial para primeira teste de calibração e validação do sensor.

Figura 12 - Ligação circuito Sensor SCT-013



Fonte: Autoria própria

Figura 13 - Primeiro teste Sensor SCT-013

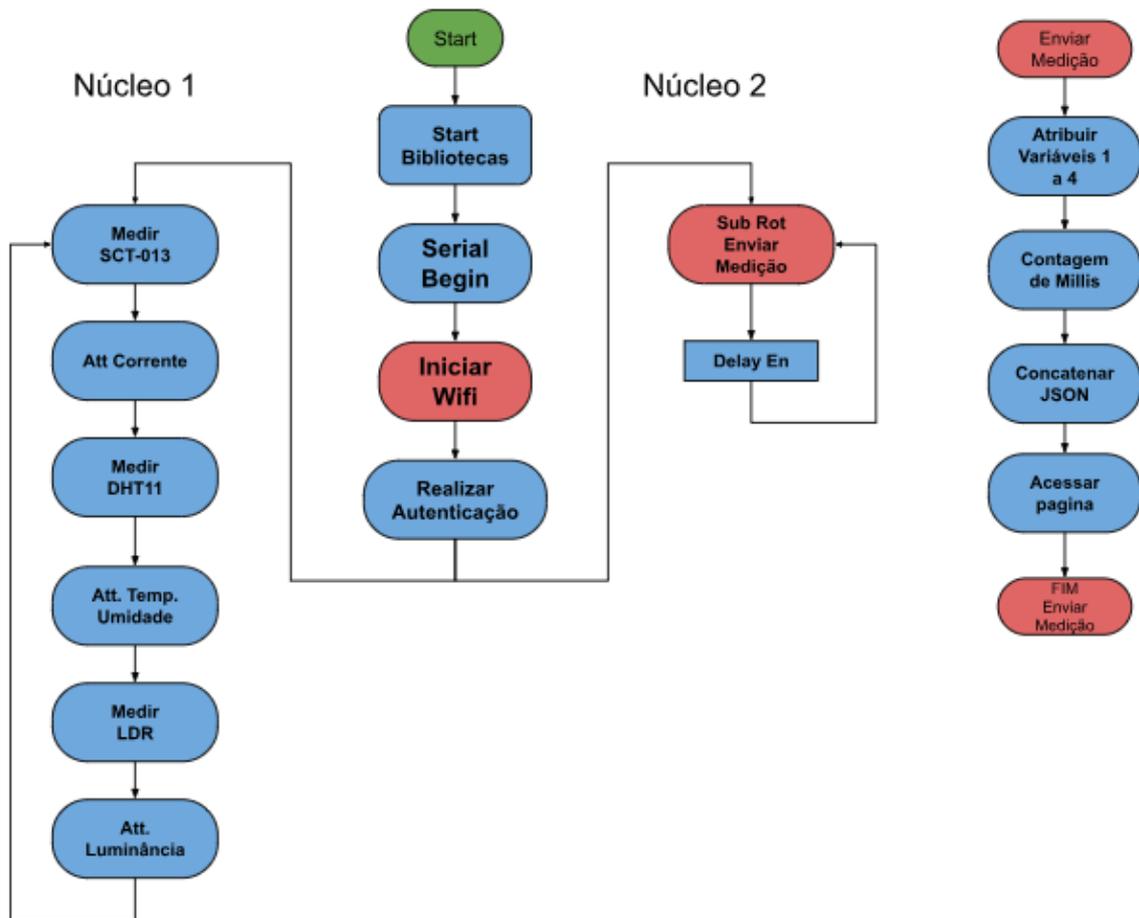


Fonte: Autoria própria

4.1.3 Programação do sistema

Para a programação do microcontrolador ESP32 foi utilizado o ambiente de programação da plataforma Arduino e a linguagem orientada a objetos C++. Na figura 14 é apresentado o fluxograma utilizado para o desenvolvimento da lógica de programação aplicada no sistema do ESP 32 para o funcionamento do sistema de monitoramento.

Figura 14 - Diagrama de blocos da programação do ESP32



Fonte: Autoria própria

O código que implementa a lógica desenvolvida na Figura 14 está no Anexo A. Nele contém quatro arquivos que são responsáveis por toda implementação desse programa, isso se dá pelo fato de que o sistema irá operar em dois núcleos de processamento diferentes, denominados “*multitask*”.

O primeiro arquivo, chamado de *Multitask.c*, é responsável pelo início do programa e realiza a chamada dos outros 3 arquivos da pasta, sendo eles *Task1.c*, *Task2.c* e *Z_Wifi.c* que são responsáveis pela definição do que cada núcleo realizará, com exceção o *Z_Wifi* que trata da inicialização da conexão via *Wi-Fi* do ESP32 e do pacote de ações que o *Task2.C* utiliza para a confecção e envio do arquivo *JSON* que publicará os registros na base de dados.

Outra característica desse código é a utilização de bibliotecas para cada um dos sensores, as quais deixam o código mais enxuto e facilitam o desenvolvimento do protótipo. Em se tratando das duas principais bibliotecas utilizadas, as bibliotecas DHT.h e Emon1.h possuem métodos definidos para utilização dos sensores de temperatura, umidade e corrente, respectivamente.

No sistema *multitask*, temos a divisão em sub-rotinas que são implementadas em cada um dos núcleos da placa, esse modelo de arquitetura faz com que, além de estar medindo as leituras dos sensores, também possa enviar dados para a rede ao mesmo tempo, evitando as variações na conectividade local. Essa característica de *multitask*, é um grande diferencial de arquitetura de software, pois permite fazer medições ao mesmo tempo que informações estão sendo enviadas sem que os núcleos fiquem sobrecarregados.

O primeiro núcleo é responsável por todo ciclo de aferição e atribuição nas variáveis globais do sistema, já o segundo é responsável pela conectividade e pela confecção do *JSON* de envio dos valores coletados. Portanto, para cada interação do segundo núcleo será gerado um registro para o sistema do Script do Google, que será explicado posteriormente.

4.1.4 Conexão com a internet

Uma das questões que se mostrou desafiadora durante o desenvolvimento do projeto foi a forma como fornecer acesso à internet no local de instalação do protótipo. Isso ocorreu, pois, embora haja uma rede Wi-Fi destinada a projetos de *IoT* na Unicamp, o dispositivo ESP32 instalado possui características particulares que poderiam impedir ou dificultar sua conexão a essa rede.

Desta forma foi instalado no local o dispositivo *RuralMax*, desenvolvido pela Neger Telecom e apresentado na Figura 15. Este é um dispositivo com a capacidade de prover internet móvel gerando uma rede Wi-Fi através do sinal 4G LTE recebido pelo equipamento. Inicialmente desenvolvido para áreas rurais, esse dispositivo possui características idênticas ao de um roteador, assim, ao ligá-lo usando um chip de qualquer operadora que possua tecnologia 4G, é gerada uma rede própria dedicada para prover conectividade através de *Wi-Fi* para o ESP32. No caso específico deste projeto, foi utilizado um chip de conectividade da Vivo, pois é a operadora que dispõe da melhor qualidade de sinal no local de instalação do sistema.

Figura 15 - Rural Max



Fonte: (NEGER, 2023)

4.1.5 Armazenamento dos dados

Para o armazenamento dos dados coletados pelo sistema foi utilizado o Google Planilhas através de uma integração com o ESP32 para que os dados fossem gravados em uma tabela comum. Para que essa integração ocorresse, foi necessária a configuração do ESP32, que inclui a definição de parâmetros de rede *Wi-Fi* para permitir que o dispositivo se conecte à internet. Em seguida, a planilha do Google foi preparada, criando uma planilha e a compartilhando com a conta de serviço do Google que será usada para autenticar a *API* de acesso à planilha.

A integração utiliza a *API* do Google Planilhas para autenticar o ESP32 e permitir a comunicação com a tabela de dados. O ESP32 é programado para enviar os dados dos sensores em intervalos regulares de cerca de 3 minutos, visando totalizar, aproximadamente, 500 registros ou linhas por dia. Uma vez configurado, os dados dos sensores são automaticamente registrados na planilha, permitindo o monitoramento em tempo real e a análise posterior. Essa integração é altamente flexível e é aplicável em uma variedade de cenários onde a coleta de dados é essencial para tomadas de decisões informadas. Toda essa configuração é feita através de um script em Javascript rodando através do *Google Scripts* que vincula a planilha desejada com os parâmetros que se desejam salvar.

Abaixo é apresentado o script desenvolvido para o projeto proposto neste trabalho.

```
function doGet(e) {
  Logger.log( JSON.stringify(e) ); // visualiza os parametros
  var result = 'Ok'; // assume sucesso
  if (e.parameter == 'undefined') {
    result = 'No Parameters';
  }
  else {
    var sheet_id = '1zqr_AQCpU5keABmUwe2aMzcKNNvBd7kMqxDJE-EzfIo'; // ID
da planilha
    var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet(); // get
Active sheet
    var newRow = sheet.getLastRow() + 1;
    var rowData = [];
    d = new Date();
    rowData[0] = d; // Data na coluna 1
    rowData[1] = d.toLocaleTimeString(); // Hora na coluna 2

    for (var param in e.parameter) {
      Logger.log('In for loop, param=' + param);
      var value = parseFloat(stripQuotes(e.parameter[param]));
      Logger.log(param + ':' + e.parameter[param]);
      switch (param) {
        case 'value1': //Temperatura
          rowData[2] = value;
          result = 'Written on column A';
          break;
        case 'value2': //Corrente
          rowData[3] = value;
          result += ' Written on column B';
          break;
        case 'value3': //Umidade
          rowData[4] = value;
          result += ' Written on column C';
          break;
        case 'value4': //Luminosidade
          rowData[5] = value;
```

```

        result += ' Written on column D';
        break;
    case 'value5': //Contagem milllisegundos
        rowData[6] = value;
        result += ' Written on column E';
        break;
    default:
        result = "unsupported parameter";
    }
}
Logger.log(JSON.stringify(rowData));
// Escreve uma nova linha
var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
newRange.setValues([rowData]);
}
// Retorna o resultado da operação
return ContentService.createTextOutput(result);
}
function stripQuotes( value ) {
    return value.replace(/^["]|["]$/g, "");
}
}

```

Dessa forma, a base de dados criada no Google planilhas, será utilizada para salvar em cada linha os registros de leitura de cada uma das grandezas monitoradas, bem como a data e hora desses respectivos registros. O Diagrama ER da Figura 16 apresenta a estrutura a ser seguida na base de dado proposta:

Figura 16 - Diagrama ER da base de dados desenvolvida

Registro de leitura	
Data	char
Hora	int
Temperatura	float
Corrente	float
Umidade	float
Luminosidade	float
Register Count	int

Fonte: Autoria própria

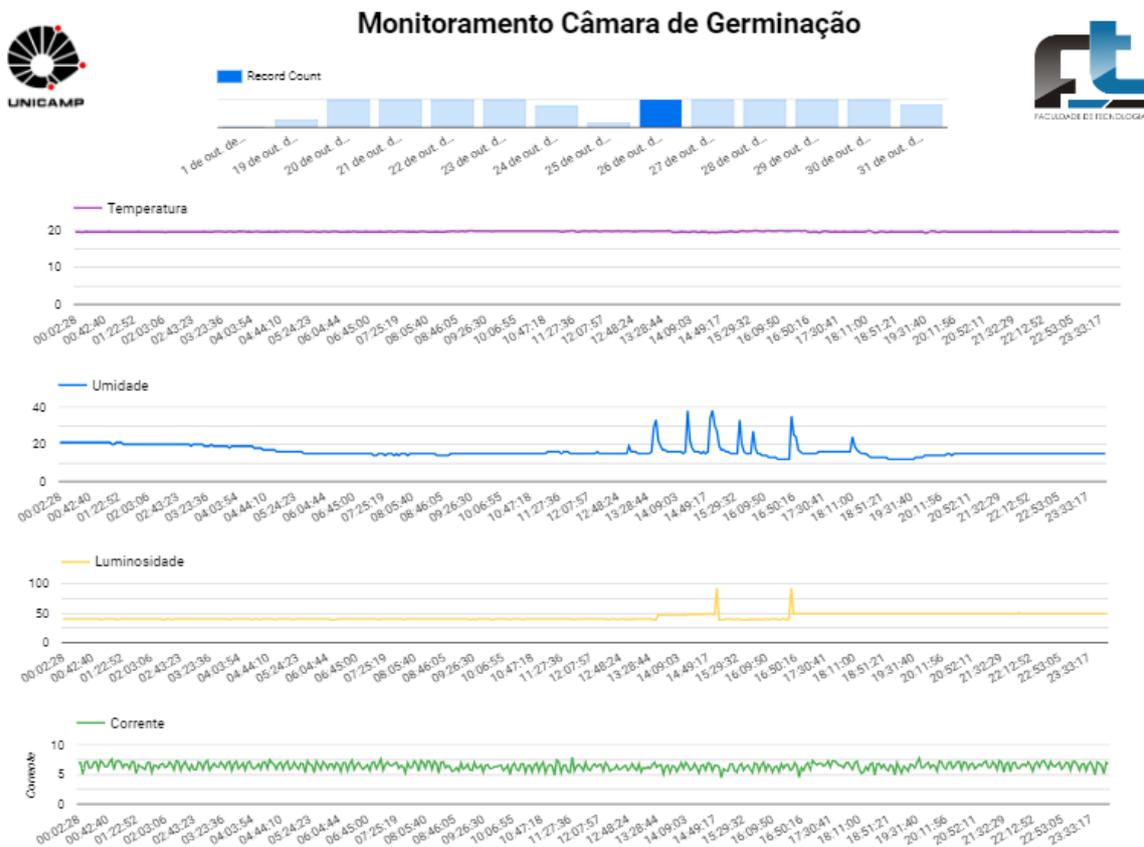
4.1.6 Desenvolvimento da Dashboard

O *Looker Studio* foi a ferramenta utilizada para a exibição dos dados coletados pelo sistema. A ferramenta faz parte do pacote de serviços online do Google e por isso possui fácil interação com o Google Planilhas, manuseio simplificado que permite criação de gráficos e visualizações comparativas através do método *drag and drop*, sendo esses os principais motivos da escolha dessa ferramenta para o projeto.

A primeira versão da *dashboard* criada conta com a presença de 5 gráficos, sendo o primeiro deles um contador de registros por datas, que também serve para selecionar a data dos dados que serão visualizados, enquanto os demais apresentam os dados de temperatura, umidade, luminosidade e corrente elétrica, respectivamente. Nessa versão será possível acompanhar a variação das grandezas, que estão no eixo y do gráfico, de acordo com o horário de cada dia apresentado no eixo x. Como novos dados são registrados a cada 3 minutos, espera-se que ao final de cada dia seja possível plotar um gráfico com cerca de 500 registros de cada grandeza.

A ferramenta oferece a possibilidade de personalizar todo o layout, fato que permitiu que fosse inserido os logotipos da Unicamp e da Faculdade de Tecnologia na *dashboard* desenvolvida, conforme apresenta a Figura 17.

Figura 17 - Dashboard desenvolvida



Fonte: Autoria própria

4.2 Prova de conceito

Para validar a aplicabilidade e funcionalidade do sistema de telemetria desenvolvido em um ambiente controlado, foi disponibilizado o acesso a uma estufa de controle de cultivo de microorganismos do LEMA - Laboratório de Estudos em Microbiologia Aplicada da Faculdade de Tecnologia. Para que os organismos cultivados tenham desenvolvimento satisfatório, é necessário que a estufa mantenha constantemente temperatura igual à 20°C, mantenha iluminação ativa periodicamente e tenha um nível de umidade do ar satisfatório, além de que idealmente não possa ocorrer falhas na alimentação elétrica da estufa.

Dessa forma, o ambiente citado se mostra extremamente adequado para testar e validar o sistema desenvolvido em todas as suas frentes de monitoramento, além de ter a possibilidade de apresentar dados reais em uma aplicação extremamente importante para as pesquisas de microbiologia e demais áreas relacionadas.

4.2.1 O ambiente a ser monitorado

O Laboratório de Estudos em Microbiologia Aplicada da Faculdade de Tecnologia (FT/Unicamp) utiliza as estufas, com a apresentada na Figura 18, para recriar condições controladas em uma simulação do meio ambiente, com objetivo de cultivar os microrganismos nas condições ideais para vida dos mesmos. Devido a isso, o fotoperíodo é controlado, para basicamente substituir a luz solar e ter o ciclo de luz de um dia. Ao mesmo tempo, a temperatura controlada para que os cultivos de organismos fiquem em uma temperatura ideal estabelecida por Normas ABNT. Junto a isso, a umidade também permite que os organismos fiquem em meio de cultivo ideais.

Figura 18 - Estufa de germinação



Fonte: Autoria própria

4.2.2 Montagem do sistema no ambiente

Para a montagem do protótipo na primeira estufa monitorada, apresentada na Figura 19, foi necessário fazer algumas adaptações para o cenário desejado utilizando alguns cabos maiores para que os sensores ficassem dentro das estufas. Foi utilizado o cabo de 8 vias (CAT 4) para expandir o alcance dos sensores de modo que o equipamento principal, o ESP32, fique na parte externa da estufa e os sensores fiquem na parte interna, por exceção do sensor de corrente que estará externo junto aos cabos de alimentação elétrica. Após um primeiro período de testes, o mesmo sistema de montagem foi usado para uma segunda estufa, apresentada na Figura 20.

Figura 19 - Protótipo instalado na 1ª estufa



Fonte: Autoria própria

Figura 20 - Protótipo instalado na 2ª estufa



Fonte: Autoria própria

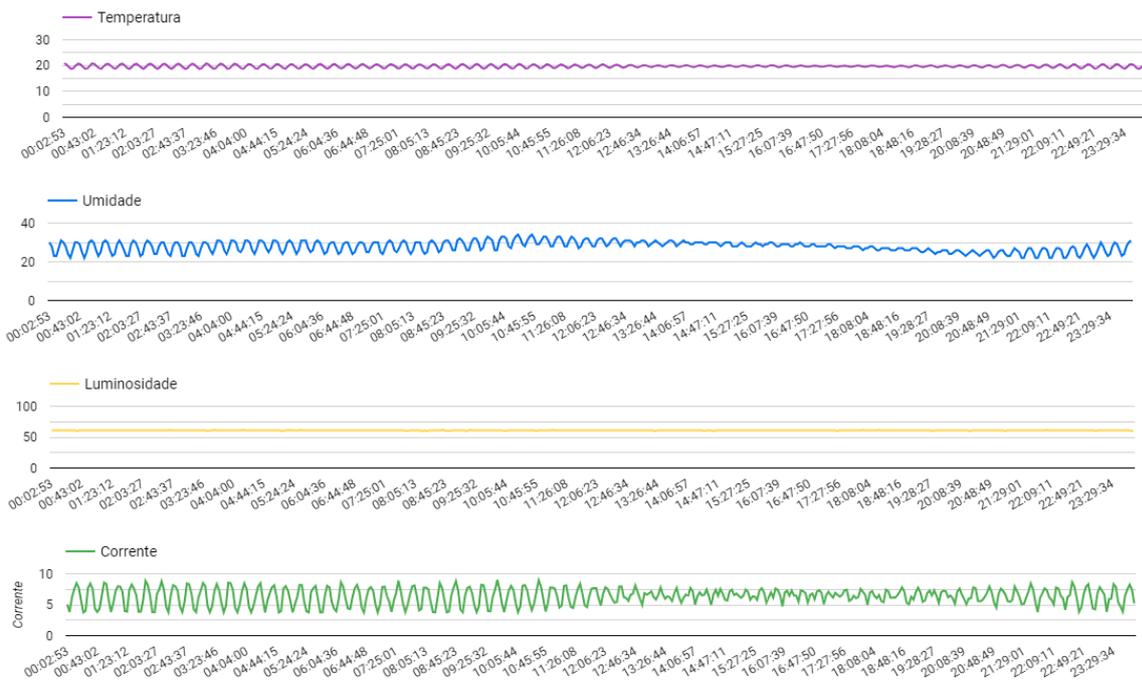
5 RESULTADOS E DISCUSSÕES

Realizado o desenvolvimento do protótipo e sua ativação no ambiente determinado, iniciou-se um primeiro período de testes que almejava coletar os dados da estufa de germinação durante 12 dias para avaliar o comportamento e integridade do protótipo em campo e também ter uma primeira impressão quanto ao funcionamento da câmara.

Neste primeiro teste, notou-se que o sistema teve pleno funcionamento na maior parte do tempo, com algumas pequenas interrupções devido à desconexão com a internet e por algumas reinicializações automáticas não previstas no ESP32, que logo foram normalizadas após pequena intervenção *in loco*.

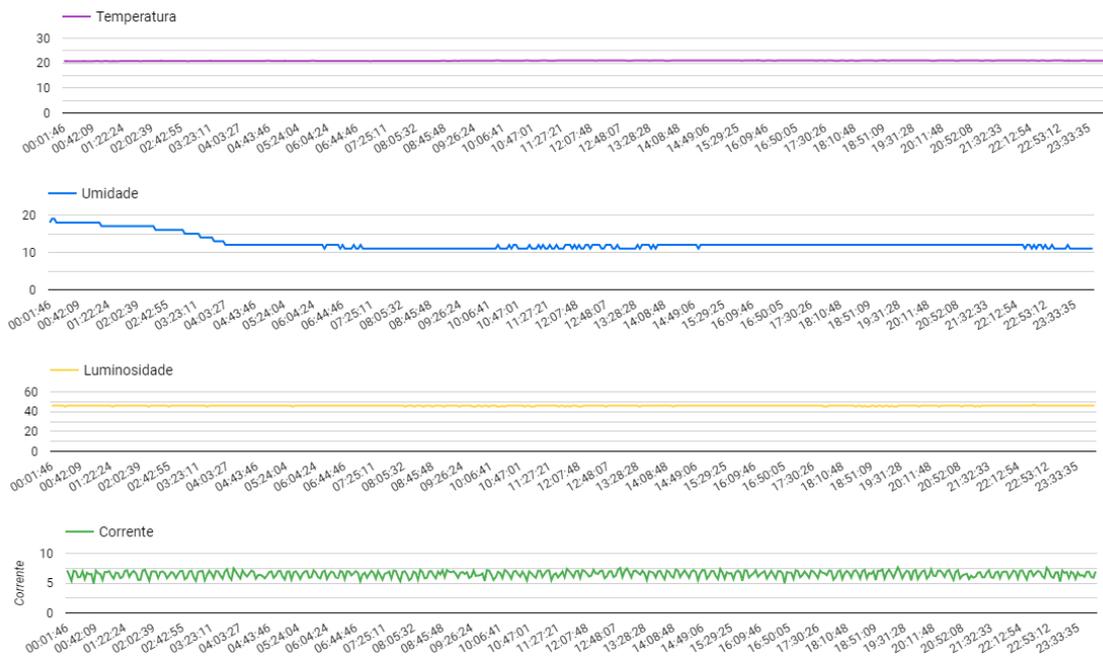
O teste permitiu verificar que durante os dias úteis, com pesquisadores e estudantes constantemente trabalhando no laboratório, há diversas variações na temperatura, umidade e corrente dentro da estufa devido a várias aberturas de sua porta durante o dia de trabalho. Já durante os finais de semana as grandezas monitoradas se mantiveram mais constantes, com exceção do final de semana dos dias 21 e 22 de outubro de 2023, quando todas as grandezas apresentaram um comportamento cíclico, com duração de 20 minutos em cada ciclo, conforme pode ser verificado na Figura 21. Este comportamento pode vir a ser um problema de mau funcionamento da estufa, que pode comprometer o ciclo de vida dos organismos que ali são criados devido às constantes variações. Logo nos primeiros dias já pode-se verificar também um comportamento não desejado relacionado ao fotoperíodo que a estufa deveria obedecer; como pode se observar nas Figuras 21 e 22, a luminosidade dentro desta estufa manteve-se a mesma todo o tempo em níveis acima de 50% de iluminância, indicando que havia uma fonte luminosa acesa o tempo todo. Pode-se notar também que o nível de umidade relativa do ar nesta estufa esteve sempre abaixo de 20%.

Figura 21 - Dashboard da 1ª estufa em 21 de outubro de 2023



Fonte: Autoria própria

Figura 22 - Dashboard da 1ª estufa em 28 de outubro de 2023



Fonte: Autoria própria

Finalizada a coleta de cerca de 12 dias de dados na primeira câmara de germinação, o dispositivo foi instalado em uma outra câmara semelhante para verificar o seu comportamento e compará-lo com os dados obtidos anteriormente. Nesse segundo local de testes foram observadas diferenças no funcionamento da estufa através dos dados obtidos, diagnosticando que, assim como na primeira estufa, o sistema de fotoperíodo do equipamento não estava atuando corretamente uma vez que a luz se mantinha desligada todo o tempo, podendo prejudicar a pesquisa realizada no laboratório. A equipe do laboratório foi alertada que este comportamento não estava correspondendo com o esperado e logo constatou que o sistema de fotoperíodo não estava ativado na configuração da câmara. O sistema de fotoperíodo foi então ativado adequadamente e logo observou-se através da dashboard a mudança de comportamento do ciclo de luz na câmara, que agora passou a operar em ciclos de aproximadamente 8 horas de luz e 16 horas desligadas. Nesta segunda estufa, percebeu-se que a umidade relativa do ar se manteve sempre entre 50% e 55%.

A Figura 23 apresenta os dados obtidos após a correção dos ciclos da segunda estufa, mostrando que o sistema do fotoperíodo passou a funcionar normalmente.

Figura 23 - Dashboard da 2ª estufa em 04 de novembro de 2023



Fonte: Autoria própria

Quanto ao monitoramento de temperatura, ambas as estufas mantiveram níveis dentro do esperado, ou seja, aproximadamente 20°C durante todo o período monitorado. As únicas exceções foram os dias em que a primeira estufa apresentou o comportamento cíclico citado anteriormente e os momentos em que havia abertura da porta das estufas, onde era possível observar um pequeno aumento momentâneo de temperatura seguido de uma breve normalização nos minutos seguintes.

Já a corrente manteve comportamento muito semelhante em ambas as estufas. Na primeira estufa obteve uma média de 6.56A consumido e na segunda obteve 6.52A. Apesar disso, apresentaram uma pequena variação na sua curva de consumo conforme mostra a Figura 24, onde a primeira estufa tem um comportamento mais constante enquanto a segunda estufa apresenta alguns ciclos de aumento e diminuição na corrente do circuito.

Figura 24 - Comparação de corrente da estufa 1 (cima) e estufa 2 (baixo)



Fonte: Autoria própria

Em suma, a fase inicial de testes do protótipo revelou que, embora tenha havido um funcionamento geral satisfatório na maioria do tempo, pequenas interrupções, desconexões com a internet e reinicializações não previstas no ESP32 foram detectadas. A observação dos dados coletados durante o período de testes indicou variações significativas na temperatura, umidade e corrente dentro da estufa, especialmente durante os dias úteis, associados às aberturas frequentes da porta. Pequenas variações na corrente foram notadas durante os finais de semana, onde ocorreram ciclos de aumento e diminuição das grandezas monitoradas. Adicionalmente, um comportamento não desejado relacionado ao fotoperíodo foi detectado, com a luz interna da estufa permanecendo constantemente acesa. Ao transferir o dispositivo para uma segunda estufa, os problemas com o fotoperíodo se repetiram, exigindo uma verificação manual da programação da estufa para garantir o ciclo adequado de luz, sendo

corrigido após essa intervenção. Apesar de todas as observações citadas, as estufas mantiveram os níveis de temperatura satisfatórios durante todo o período dos testes.

6 CONCLUSÃO

No decorrer deste Trabalho de Conclusão de Curso, foram explorados conceitos ligados à Internet das Coisas aplicada a sistemas de monitoramento via internet. O estudo aqui descrito visou desenvolver uma solução de telemetria para o monitoramento de ambientes controlados, de modo a aperfeiçoar a utilização de recursos disponíveis e a diminuição de possíveis perdas. O estudo, em termos específicos, desenvolveu um protótipo capaz de monitorar temperatura, umidade e luminosidade e corrente elétrica em uma estufa de cultivo de microrganismos, e em termos mais amplos, posiciona-se na interface entre as Engenharias de Telecomunicações e Ambiental.

Descobertas de outros estudos dessa área do conhecimento, apresentados anteriormente, foram fundamentais para embasar as considerações do presente estudo, destacando os diferenciais, melhorias e limites de tais sistemas.

Os resultados obtidos no presente trabalho indicam que o protótipo desenvolvido é uma ferramenta funcional para o monitoramento em tempo real de um ambiente controlado, com precisão nos dados e versatilidade para as mais diversas aplicações. Esse sistema ajudou a equipe técnica do laboratório a identificar falhas e algumas variações antes não percebidas no funcionamento de suas estufas, o que demonstra que o objetivo do trabalho foi atingido de maneira satisfatória. No mais, o sistema desenvolvido mostra-se escalável e facilmente adaptável para outros tipos de tecnologias de comunicação e também outros tipos de sensores.

As implicações deste trabalho são significativas para aplicações que exijam um monitoramento detalhado, como por exemplo em indústrias, com o monitoramento dos insumos energéticos e dos status de máquinas em ambientes industriais, para monitorar os gastos hídricos na agricultura e pecuária, para obter economia de energia em ambientes residenciais, ou até mesmo em centros de pesquisas que exijam uma alta precisão no controle ambiental. Este estudo também ajuda a abrir caminho para futuras pesquisas sobre o monitoramento de quase qualquer grandeza dentro de um ambiente, e que explorem aspectos como o uso de baterias, monitoramento de consumo de água, desenvolvimento de ferramenta de alarmes, visualização via aplicativo móvel e utilização de rede de telefonia celular, entre outros.

Em última análise, este Trabalho de Conclusão de Curso representa um esforço dedicado e uma contribuição para a aplicação do conceito de Internet das Coisas em sistemas de monitoramento. Ele destaca a importância de se ter uma comunidade unida em prol do desenvolvimento tecnológico colaborativo, de baixo custo e eficiente energeticamente, além de mostrar como isso pode afetar positivamente na qualidade de produção e pesquisa. A pesquisa e a dedicação investidas neste estudo proporcionaram uma compreensão mais profunda sobre sistemas de telemetria e contribuem para o crescimento contínuo deste campo. Espera-se que este trabalho inspire futuros pesquisadores a expandir os limites do conhecimento em Internet das Coisas.

BIBLIOGRAFIA

ARDUROBOTICA. **Sensor De Corrente Não Invasivo 100a Sct-013**. 2023. Disponível em:< <https://www.ardurobotica.com.br/MLB-864840980-sensor-de-corrente-no-invasivo-100a-sct-013-para-arduino- JM>>. Acesso em 16 de novembro de 2023.

AUGUSTO, T.R., **Correção de Fator de Potência de Baixo Custo Utilizando Banco Automático de Capacitores Controlado pelo Microcontrolador 8051**. Trabalho de conclusão de curso de Engenharia da Computação. Centro Universitário de Brasília. Brasília, DF, Brasil 2018. Disponível em:<<https://repositorio.uniceub.br/jspui/bitstream/123456789/3137/2/20565213.pdf>>. Acesso em 16 de novembro de 2023.

BABIUCH, M., FOLTYNEK, P., SMUTNY, P., **Using the ESP32 Microcontroller for Data Processing**. In 2019 20th International Carpathian Control Conference (ICCC), Krakow-Wieliczka, Poland. 2019. Disponível em:<<https://ieeexplore.ieee.org/document/8765944>>. Acesso em 15 de setembro de 2023.

BARBOSA, R. Z., MARTINS, J. E. M., **Desenvolvimento de um sistema de telemetria para monitoramento térmico em casas de vegetação**. Brazilian Journal of Biosystems Engineering vol. 8: 25-33. 2014. Disponível em:<<http://seer.tupa.unesp.br/index.php/BIOENG/article/view/175>>. Acesso em 25 de setembro de 2023.

CABRAL, S., **Um sistema de telemetria com tecnologias GSM/GPRS para a área ambiental**. Trabalho de Conclusão de Curso. Curso de Ciência da Computação, Universidade Estadual Paulista. 2023. Disponível em:<<https://repositorio.unesp.br/items/de2e53b2-a86e-4a7f-85d2-a2ef8155a3fb>>. Acesso em 25 de setembro de 2023.

CARRASCO, D., **ESP32: WiFi and ESP-NOW simultaneously**. ElectroSoftCloud. 2019. Disponível em: < <https://www.electrosoftcloud.com/en/esp32-wifi-and-esp-now-simultaneously/>>. Acesso em 22 de setembro de 2023.

CHASE, O., **Sistemas embarcados**. Revista Sociedade Brasileira de Automática Jovem, 2007. Disponível em: <<http://www.lyfreitas.com.br/ant/pdf/Embarcados.pdf>>. Acesso em 11 de setembro de 2023.

CUNHA, A. F. **O que são sistemas embarcados?**. Revista Saber Eletrônica, n. 414, 2007. Disponível em:<https://files.comunidades.net/mutcom/ARTIGO_SIST_EMB.pdf>. Acesso em 11 de setembro de 2023.

DATTA, P., SHARMAN, B. **A survey on IoT architectures, protocols, security and smart city-based applications**. In: 8th International Conference on Computing, Communication and Networking Technologies, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8203943>>. Acesso em 28 de Maio de 2023.

ELETROGATE. **Módulo Carregador de Baterias de Lítio TP4056 – MICRO USB**. 2021. Disponível em:< <https://www.eletrogate.com/modulo-carregador-de-baterias-de-litio-tp4056-micro-usb>>. Acesso em 16 de novembro de 2023.

ELETROGATE. **Sensor de Umidade e Temperatura DHT11**. 2023. Disponível em: <<https://www.eletrogate.com/sensor-de-umidade-e-temperatura-dht11>>. Acesso em 16 de novembro de 2023.

EMBARCADOS. **A evolução do protocolo Wi-Fi (IEEE 802.11)**. 2021. Disponível em: <<https://embarcados.com.br/a-evolucao-do-protocolo-wi-fi-ieee-802-11/>>. Acesso em 22 de setembro de 2023.

FERESU, Z., **DHT11 Based Temperature and Humidity Measuring System**. J Electr Eng Electron Technol Vol: 11. University of Zimbabwe, Zimbabwe. 2022. Disponível em: <https://www.scitechnol.com/peer-review/dht11-based-temperature-andhumidity-measuring-system-OqwE.php?article_id=20039#:~:text=The%20DHT11%20is%20a%20basic,is%20fairly%20imple%20to%20use.>. Acesso em 20 de setembro de 2023.

GOMES, P. H. O., **Concepção de sistema de telemetria para aeronaves remotamente controladas**. Monografia apresentada ao Curso de Engenharia de Controle e Automação. Universidade Federal de Ouro Preto. Ouro Preto, MG, Brasil. 2016. Disponível em: <https://www.monografias.ufop.br/bitstream/35400000/284/1/MONOGRAFIA_ModelamentoSistemaTelemetria.pdf>. Acesso em 13 de setembro de 2023.

HOANG, T. N., VAN, S., NGUYEN B. D., **ESP-NOW Based Decentralized Low Cost Voice Communication Systems For Buildings**. In International Symposium on Electrical and Electronics Engineering (ISEE), Ho Chi Minh City, Vietnam. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8921062>>. Acesso em 7 de setembro de 2023.

HUINFINITO. **Módulo Sensor de Luz - LDR**. 2023. Disponível em: <<https://www.huinfinito.com.br/arduino/970-modulo-sensor-de-luz-ldr.html>>. Acesso em 16 de novembro de 2023.

IOTDESIGN PRO. **ESP32 Data Logging to Google Sheets with Google Scripts**. 2022. Disponível em: <<https://iotdesignpro.com/articles/esp32-data-logging-to-google-sheets-with-google-scripts>>. Acesso em 20 de setembro de 2023.

LAFRATTA, J. M., LOPES, D. N., CASTRO, M. A. S., **Desenvolvimento de sistema de monitoramento remoto de processo de compostagem baseado no conceito de Internet das Coisas**. In 31º Congresso da Associação Brasileira de Engenharia Sanitária e Ambiental, Curitiba, PR, Brasil. 2021. Disponível em: <https://drive.google.com/file/d/1q3LA4LJ_EX14qPHyl_kYIGRUZhWezBsC/view?usp=drive_link>. Acesso em 25 de setembro de 2023.

LITJENS, O. J., **Automação de estufas agrícolas utilizando sensoriamento remoto e o protocolo Zigbee**. Trabalho de Conclusão de Curso. Curso de Engenharia Elétrica, Universidade de São Paulo. 2009. Disponível em: <http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-22042010-103032/publico/Litjens_Otto_Jacob.pdf>. Acesso em 25 de setembro de 2023.

LUCON, O., GOLDEMBERG, J. **Crise financeira, energia e sustentabilidade no Brasil**. São Paulo-SP. Scientific Electronic Library Online, 2009. Disponível em:

<<https://www.scielo.br/j/ea/a/3t4kHdSrn7rKbGSLcRkKDTd/?lang=pt#>>. Acesso em 21 de maio de 2023.

MAIER, A., SHARP, A., VAGAPOV, Y., **Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things**. In *2017 Internet Technologies and Applications (ITA)*, Wrexham, UK, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8101926>>. Acesso em 12 de setembro de 2023.

MARTINS, M.P. **Inovação Tecnológica e Eficiência Energética**. Dissertação (MBA em Engenharia Elétrica). Rio de Janeiro-RJ: Universidade Federal do Rio de Janeiro, 1999. Disponível em: <<https://q.eletronbras.com/pt/EstudantesePesquisadores/biblioteca/Inova%C3%A7%C3%A3o%20Tecnol%C3%B3gica%20e%20Efici%C3%Aancia%20Energ%C3%A9tica.pdf>>. Acesso em 21 de maio de 2023.

MELO, A. B., VIERA, J., **Desenvolvimento de um Sistema de Telemetria utilizando conceitos de IOT**. In XXXIV Simpósio Brasileiro de Telecomunicações (SBrT), Santarém, PA, Brasil. 2016. Disponível em: <<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwipiuW32viBAxV0r5UCHVELC7UQFnoECA4QAQ&url=https%3A%2F%2Fbiblioteca.sbrt.org.br%2Farticlefile%2F1288.pdf&usg=AOvVaw0kKIMNk7vSywrnF-HyZyJN&opi=89978449>>. Acesso em 25 de setembro de 2023.

NEGER. **RuralMax Pro – Plataforma avançada de conectividade rural**. 2023. Disponível em: <<https://www.neger.net.br/produtos/internet-rural/kit-essencial-ruralmax-pro-plataforma-avancada-de-conectividade-rural>>. Acesso em 14 de novembro de 2023.

PEREIRA, G. M., **Princípios de telemetria aplicados ao monitoramento de hardware via interface web**. Trabalho de Conclusão de Curso. Curso de Engenharia Elétrica, Universidade do Sul de Santa Catarina. 2022. Disponível em: <<https://repositorio.animaeducacao.com.br/handle/ANIMA/27276>>. Acesso em 25 de setembro de 2023.

PIOVESAN, F. C., **Telemetria aplicada na mecanização agrícola utilizando o datalogger CR 1000**. Trabalho de conclusão de curso de Ciência da Computação. Universidade Federal de Santa Maria. Santa Maria, RS, Brasil. 2008. Disponível em: <<https://repositorio.ufsm.br/handle/1/25413>>. Acesso em 13 de setembro de 2023.

PONCHET, P. B., **Avaliação do transformador de corrente SCT-013 aplicado em medidor eletrônico de potência**. Trabalho de Conclusão de Curso de Engenharia. Universidade Federal de Campina Grande. Campina Grande, PB, Brasil. 2016. Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/18687>>. Acesso em: 20 de setembro de 2023.

SARAVATI. **Placa ESP32 WiFi / Bluetooth DEVKit V1**. 2023. Disponível em: <<https://www.saravati.com.br/placa-esp32-wifi-bluetooth-devkit-v1-30-pinos.html>>. Acesso em 16 de novembro de 2023.

SHEN, G., LIU, B. **The visions, technologies, applications and security issues of Internet of Things**. In: International conference on E-Business and E-Government, Shanghai, 2011.

Disponível em: <<https://ieeexplore.ieee.org/document/5881892>>. Acesso em 05 de junho de 2023.

SINGH, K. J., KAPOOR, D.S., **A survey of IoT platforms: Create Your Own Internet of Things**. in IEEE Consumer Electronics Magazine, vol. 6, no. 2, pp. 57-68, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/7879392/citations#citations>>. Acesso em 12 de setembro de 2023.

SILVA, A. S. B., BERNARDO, A. C. F. *et al* **Qualidade total: Os princípios de Deming em uma multinacional**. In: XXXVI Encontro Nacional de Engenharia de Produção, 2016. Disponível em: <https://abepro.org.br/biblioteca/TN_STP_227_328_29587.pdf>. Acesso em 23 de Maio de 2023.

SILVA, I. L., **Desenvolvimento de uma Estufa Automatizada Baseada em IOT para Uso Residencial**. Trabalho de Conclusão de Curso. Curso de Engenharia Elétrica, Universidade Regional do Noroeste do Estado do Rio Grande do Sul. 2020. Disponível em: <https://bibliodigital.unijui.edu.br:8443/xmlui/handle/123456789/436/browse?order=ASC&pp=20&sort_by=1&etal=-1&offset=61&type=title&starts_with=C>. Acesso em 16 de setembro de 2023.

SNIPES, G., **Google Data Studio, Product Review**. Journal of Librarianship and Scholarly Communication. 2018. Disponível em: <<https://www.iastatedigitalpress.com/jlsc/article/id/12800/>>. Acesso em 23 de setembro de 2023.

SOURCEFORGE. **Open Source vs. Closed Source (Proprietary) Software: A Never-Ending Debate**. 2022. Disponível em: <<https://sourceforge.net/articles/open-source-vs-closed-source-proprietary-software-a-never-ending-debate/>>. Acesso em 01 de junho de 2023.

SRIVASTAVA, D., KESARWANI, A., DUBEY, S., **Measurement of Temperature and Humidity by using Arduino Tool and DHT11**. In International Research Journal of Engineering and Technology (IRJET). 2018. Disponível em: <<https://www.irjet.net/archives/V5/i12/IRJET-V5I12167.pdf>>. Acesso em 20 de setembro de 2023.

VERMA, P. K., VERMA, R., PRAKASH, A., AGRAWAL, A. *et al*. **Machine-to-Machine (M2M) communications: A survey**. Journal of Network and Computer Applications, v. 66, p. 83-105, 2016. Disponível em: <<https://doi.org/10.1016/j.jnca.2016.02.016>>. Acesso em 01 de junho de 2023.

ZANUZZO, L. T., **Sistema de monitoramento do consumo de água controlado por um aplicativo Android**. Trabalho de Conclusão de Curso. Curso de Engenharia da Computação, Universidade de São Paulo. 2017. Disponível em: <http://www.tcc.sc.usp.br/tce/disponiveis/97/970010/tce-13042018-164107/publico/Zanuzzo_Lucas_tcc.pdf>. Acesso em 25 de setembro de 2023.

ANEXO A - Código de programação do ESP32

- Código multitask:

```
#include <HTTPClient.h>
#include <WiFi.h>
#include "string.h"
#include "EmonLib.h" // Inclui a biblioteca EmonLib
#include <DHT.h>

#define DHTPIN 27 //Pino onde o DHT está ligado]
#define LDRPIN 26 // Pino Leitura LDR

const char* ssid = "Trunder"; // Nome da rede Wi-Fi
const char* password = "12345678"; // Senha da rede

char *server = "script.google.com"; // Server URL

char *GScriptId = "AKfycbxEySxkypAITF3SqUzzvmcaWAzJH_XF-
zMoQYmYNVRA7o_E3Q6eP99XaA4mqntJox4D";

const int httpsPort = 443;

TaskHandle_t Task1;
TaskHandle_t Task2;

WiFiClientSecure client;

DHT dht(DHTPIN, DHT11);

double Irms1;

float leitura = 39;

EnergyMonitor emon1;

void setup(){
  pinMode(LDRPIN, INPUT);

  Serial.begin(115200);

  dht.begin();

  emon1.current(leitura, 90.91);

  connect_wifi();

  xTaskCreatePinnedToCore(
    Task1code, /* Task function. */
    "Task1", /* name of task. */
```

```

    10000,    /* Stack size of task */
    NULL,    /* parameter of the task */
    1,       /* priority of the task */
    &Task1,  /* Task handle to keep track of created task */
    1);     /* pin task to core 0 */

/
xTaskCreatePinnedToCore(
    Task2code, /* Task function. */
    "Task2",   /* name of task. */
    10000,    /* Stack size of task */
    NULL,     /* parameter of the task */
    1,       /* priority of the task */
    &Task2,   /* Task handle to keep track of created task */
    0);     /* pin task to core 1 */

}

void loop() {
    disableCore0WDT();
}

```

- Código Task1:

```

double correnteFinal;
double tempoEnvio;
double tempFinal;
float temperature, humidity;
int luminosidade;

void Task1code( void * pvParameters ) {

    for (;;) {

        Irms1 = emon1.calcIrms(4096);

        readSensor();

        correnteFinal = Irms1;

        tempFinal = temperature;

// bloco de verificação dos sensores
        Serial.println("-----");
        Serial.print("Corrente Final em Watts ");
        Serial.println(correnteFinal);
        Serial.print("Temperatura Final em (°C)");
        Serial.println(temperature);
        Serial.print("Humidade Final em (%)");
        Serial.println(humidity);
        Serial.print("Luminosidade em (%)");

```

```

    Serial.println(luminosidade);

    delay(2000);

}

}

void readSensor(){

    temperature = readDHTTemperature();
    humidity = readDHTHumidity();
    luminosidade = readLDR();

    /*int lumi = analogRead(35);
    Serial.print("Luminosidade ");
    Serial.println();

    int valor = analogRead(35); // Lê o valor analógico do pino GPIO35
    Serial.print("Valor Analógico: ");
    Serial.println(valor);

    delay(1000);*/

}

//-----
float readDHTTemperature() {
    // Leitura em graus Celsius
    float t = dht.readTemperature();
    if (isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return -1;
    }
    else {
        //Serial.println(t);
        return t;
    }
}

float readDHTHumidity() {
    float h = dht.readHumidity();
    if (isnan(h)) {
        Serial.println("Failed to read from DHT sensor!");
        return -1;
    }
    else {
        //Serial.println(h);
        return h;
    }
}

float readLDR() {

```

```

int valor = analogRead(35); // Lê o valor analógico do pino GPI035
int lumi;

if (isnan(valor)) {
  Serial.println("Failed to read from LDR sensor!");
}

else {

  lumi = map(valor, 0, 4095, 100, 0);

  return lumi;
}
}

```

- Código Task 2:

```

void Task2code( void *pvParameters ){

  for(;;){
    enviarMedicao();
    delay(2000);
  }
}

```

- Código Z_Function:

```

void connect_wifi()
{

  Serial.print("Connecting to wifi: ");
  Serial.println(ssid);
  Serial.flush();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {

    delay(500);
    Serial.print(WiFi.status());
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void enviarMedicao()

```

```

{
  HTTPClient http;

  tempoEnvio = millis() - tempoEnvio;
  String url = String("https://script.google.com") + "/macros/s/" + GScriptId + "/exec?" +
"value1=" + tempFinal + "&value2=" + String(correnteFinal, 2) + "&value3=" + humidity +
"&value4=" + luminosidade + "&value5=" + tempoEnvio;
  tempoEnvio = millis();

  Serial.print("Making a request");
  http.begin(url.c_str());
  http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
  int httpCode = http.GET();
  String payload;
  if (httpCode > 0) {
    payload = http.getString();

    Serial.println(httpCode);
    Serial.println(payload);
    // testdrawstyles(payload);
    //if (httpCode == 200 or httpCode == 201) tempPing.Saida(0);
  }
  else {
    Serial.println("Error on HTTP request");
  }
  http.end();
}

```