

Sistema IoT multipropósito para monitoramento de laboratórios e experimentos

Alan Franco Rosa^{✉, *}, David Vaz Teixeira^{✉, †} and Nilton Alves Júnior^{✉, ‡}

Laboratório de Instrumentação, Informação e IoT (Lab3I)

Centro Brasileiro de Pesquisas Físicas (CBPF),

Rua Dr. Xavier Sigaud 150,

CEP: 22290-180, Urca, RJ, Brasil

Submetido em 14/12/2021

Aceito em 22/12/2021

Resumo: Utilizando os dispositivos ESP32 e Raspberry Pi, além dos programas Mosquitto e Node-RED, foi possível elaborar um sistema baseado em *IoT* capaz de fazer a aquisição de dados, obtidos através de sensores específicos, de variáveis ambientais (temperatura, umidade, pressão e luminosidade) e de energia elétrica (tensão, corrente, potência ativa, fator de potência e frequência). Além disso, o sistema ainda permite que seja feita a atuação remota para modificação do estado de funcionamento do equipamento.

Palavras chave: Instrumentação, IoT, ESP32, MQTT, Node-RED.

Abstract: Using the ESP32 and Raspberry Pi devices, in addition to the Mosquitto and Node-RED programs, it was possible to develop a system based on *IoT* capable of acquiring data, obtained through specific sensors, of environmental variables (temperature, humidity, pressure, and luminosity) and electrical energy (voltage, current, active power, power factor, and frequency). In addition, the system also allows for remote actuation to change the operating status of the equipment.

Keywords: Instrumentation, IoT, ESP32, MQTT, Node-RED.

Internet para trocar **Dados** utilizando **Padrões** que garantem a **interoperabilidade** e permitem, de forma completa, a **Automação** dos processos.

1. INTRODUÇÃO

A Internet das Coisas (*Internet of Things - IoT*) pode ser, de forma simplificada, considerada como uma rede composta de equipamentos que trocam informações entre si através da internet (Network of Things). Cada “coisa” é identificada de forma única (ID) através de seu sistema embarcado, e é capaz de se comunicar utilizando a rede. De acordo com RAYES[1], a definição mais completa do termo IoT seria:

IoT é uma tecnologia que envolve **Padrões e Processos** que possibilitam que **Coisas** possam se conectar através da

No ambiente de pesquisa, principalmente nas áreas experimentais, a coleta de dados é uma etapa de extrema importância. Nesse sentido, o monitoramento adequado do ambiente e do próprio experimento, se apresenta como uma forma de aumentar a confiabilidade e a qualidade dos dados obtidos.

Conhecer a natureza dos eventos, que podem afetar o experimento e a operação de um equipamento, é essencial para definir qual atitude deve ser tomada para a eliminação da adversidade. Fenômenos dependentes da temperatura costumam ser lentos e necessitam de uma análise contínua e de longos períodos de medição. Eventos espúrios e intermitentes podem demandar registros contínuos, a fim de serem detectados. Episódios periódicos podem ainda requerer métodos mais sofisticados de análise. Utilizando os dados coletados e a partir de estudos utilizando modelos físicos ou matemáticos, que permitam realizar simulações computacionais, é possível entender melhor o fenômeno causador do problema; além de permitir a descoberta das condições em que certos problemas podem se manifestar.

*Endereço Eletrônico: alanfr@cbpf.br

†Endereço Eletrônico: davidvaz@cbpf.br

‡Endereço Eletrônico: naj@cbpf.br

Durante a operação de equipamentos ligados a rede elétrica, por exemplo, geralmente os problemas associados ao fornecimento de energia elétrica são identificados quando o dispositivo deixa de funcionar como deveria. Se esses problemas não forem prematuramente eliminados, poderão ocorrer prejuízos experimentais (ruídos diversos, redução da vida útil, ocorrência de falhas, erros sistemáticos de medidas ocasionando baixa precisão e acurácia), comprometendo a capacidade produtiva das atividades de P&D e dos instrumentos.

Outro aspecto importante é o de trazer para o ambiente de pesquisa as aplicações e recomendações que estão em voga na economia e na indústria. Com o Plano Nacional de Internet das Coisas[2] o governo pretende fomentar o desenvolvimento dessa tecnologia no Brasil e cabe ao Ministério de Ciência, Tecnologia e Inovações decidir quais são as áreas prioritárias que irão receber mais incentivos. A recente aprovação da PL 6549/2019[3], o estudo feito pelo BNDES[4] e as crescentes parcerias que são firmadas em nível de governo[5] reforçam essa ideia.

Sendo assim, o objetivo deste projeto é desenvolver um sistema multipropósito de instrumentação, utilizando IoT, para ser utilizado em laboratórios e experimentos. Com isso espera-se que o sistema possa ser utilizado para a medição de grandezas físicas, das condições ambientais de operação e atuação.

2. REVISÃO DA LITERATURA

Antes de se iniciar o desenvolvimento do sistema, foi feita uma análise do que tem sido publicado referente ao tema de monitoramento e medição utilizando o *Message Queuing Telemetry Transport* (MQTT) e aplicações.

O artigo publicado na revista *IEEE Transactions on Industrial Informatics*[6], menciona a crescente preocupação com a qualidade de energia e a penetração da tecnologia *IoT*.

O artigo define que “medidor inteligente” se refere a um sistema capaz de medir a eletricidade utilizada e/ou gerada e com a capacidade de controlar remotamente o suprimento e o corte quando necessário.

O que se espera de um medidor de energia é que ele possua as seguintes qualidades:

- Capacidade de monitoramento em tempo real ou próximo ao tempo real;
- Possibilidade de leitura dos dados de forma remota e/ou local;
- Controle remoto;
- Capacidade de integração com outros sistemas;
- Capacidade de captura de dados sobre a condição do sistema e a qualidade da energia(incluindo a tensão);
- Interoperabilidade com outros sistemas.

O projeto foi desenvolvido seguindo as premissas delineadas pelo que o artigo define como “estado da arte” em medição inteligente. Foram utilizadas ferramentas não proprietárias e com facilidade de integração com as mais diversas tecnologias existentes.

Em um artigo apresentado no *Second International Conference on Green Computing and Internet of Things (ICG-CIoT)*[7], o autor apresenta uma plataforma para monitoramento utilizando o protocolo MQTT, os *hardwares* Raspberry Pi e o ESP32. A arquitetura utilizada pelo autor é interessante, pois pode ser facilmente adaptada às necessidades do projeto.

Além disso, ele utiliza outras ferramentas que são de interesse, como os programas Mosquitto[8] e Node-RED[9]. O *software* Mosquitto é um *message broker* leve para implementação do protocolo MQTT e o Node-RED, é uma ferramenta para o desenvolvimento de programas baseados em diagramas de fluxo.

Na publicação do *IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS)*[10] fica claro que o tema é de interesse e foi impulsionado pelos impactos causados devido a pandemia causada pelo Covi-19.

Neste trabalho o autor cita as dificuldades de estar no laboratório e desenvolve um aparato que permite o monitoramento do laboratório, através de câmeras, utilizando a filosofia IoT, permitindo a condução dos experimentos de forma remota.

No artigo publicado na revista *Nature*[11] mostra um mercado que vem ganhando destaque, o de *smart labs* ou *IoLT* e aborda sobre a existência de empresas especializadas em instrumentação IoT para laboratórios, como a *Tetra-Science*[12].

Estes tipos de iniciativas abrem um leque de possibilidades. Com o monitoramento dos equipamentos é possível ampliar a gama de dados coletados, permitindo um melhor controle dos ativos. Também é possível identificar os usuários que estão fazendo uso, o horário e o tempo de utilização.

Outra possibilidade que surge com o monitoramento é o de identificar padrões que possam indicar a necessidade de manutenção (manutenção preditiva)[13]. Com isso, é possível programar os períodos de parada de forma a minimizar as ocorrências de falhas durante às atividades de pesquisa.

3. METODOLOGIA

Com base na revisão bibliográfica, que foi feita para verificar a viabilidade e a relevância do projeto, foi elaborado um planejamento para a execução:

- Etapa 1 : Escolha do material;
- Etapa 2 : Desenvolvimento do software necessário;
- Etapa 3 : Implementação da arquitetura de comunicação;
- Etapa 4 : Integração e testes.

Para o desenvolvimento do sistema foi levado em consideração algumas premissas, como:

- Forma de aquisição dos dados de interesse;
- Facilidade na aquisição dos componentes;
- Dispensabilidade da aquisição de licenças de *software*;
- Facilidade para modularização;
- Facilidade para adição de novos sensores;
- Robustez para aplicação em diversos ambientes.

3.1. Escolha do material

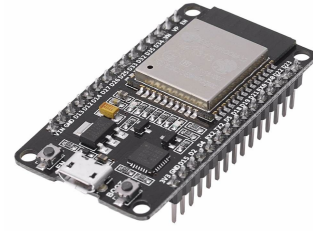
Seguindo as premissas e com base na literatura, foi feita a escolha dos componentes:

- Módulo Devkit v1 com o microcontrolador ESP32(Figura 1a);
- Raspberry Pi 4B - 4GB RAM(Figura 1b);
- Módulo PZEM-004T(Figura 2a), para medição de energia;
- Transformador de corrente não invasivo PZCT-02 (Figura 2b);
- Sensor de pressão e temperatura BMP280 (Figura 3a);
- Sensor de umidade e temperatura DHT11 (Figura 3b)
- Sensor de luminosidade MAX44009, instalado no módulo GY49 (Figura 3d);
- Sensor de presença por micro-ondas RCWL-9196, instalado no módulo RCWL-0516(Figura 3c).

O ESP32[14] faz parte de uma série de microcontroladores de baixo consumo de energia com Wi-Fi e Bluetooth integrados. O *hardware* é criado e desenvolvido pela empresa Espressif Systems e é fabricado pela TSMC usando o processo litografia de 40 nm(Figura 1a). O dispositivo ainda conta com uma boa capacidade de processamento, devido ao seu processador, que possui 240MHz, dois núcleos e aceita instruções de 32 bits.

O Raspberry Pi 4B é um computador voltado para o ensino básico de computação em escolas, mas ganhou fama em aplicações de *IoT*, devido ao seu tamanho e capacidade de processamento. Ele possui diversas portas *GPIO* que permitem a comunicação com sensores e outros dispositivos, além de possuir interfaces para comunicação USB 3.0, Bluetooth, WiFi e Ethernet. Além disso, ele conta com uma arquitetura ARM v8 64 bit, através do seu processador Cortex-A72 de quatro núcleos e 1.5GHz.

O módulo PZEM-004T (Figura 2a) é voltado para monitoramento elétrico. Ele pode ser utilizado para medição das grandezas de energia de tensão e corrente, além de possuir uma interface aberta, documentada, de fácil comunicação e



(a) Módulo ESP32 Devkit v1.



(b) Raspberry Pi 4B com 4GB de RAM.

Figura 1: Módulos de processamento.

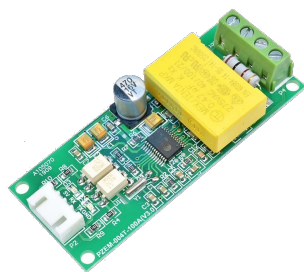
integração. Além de medir a tensão, corrente e a frequência da rede, ele também calcula o fator de potência e a potência ativa (Tabela I).

PZEM-004T	
Tensão de Alimentação [V]	3,3 a 5,5
Tensão [V]	80 a 260 $\pm$ 0,1 @0,5%
Corrente [A]	0 a 100 $\pm$ 0,001 @0,5%
Potência ativa [W]	0 a $23 \times 10^3 \pm$ 0,1 @0,5%
Fator de potência [Φ]	0 a 1 $\pm$ 0,01 @1%
Frequência [Hz]	45 a 65 $\pm$ 0,1 @0,5%
Energia ativa [kWh]	0 a 9999,00 $\pm$ 0,001 @0,5%
Interface	RS485(Modbus)

Tabela I: Dados do sensor PZEM-004T.

O sensor Pzct é um mini transformador de corrente no formato de alicate, é pequeno, fácil de transportar e possui relação de 1000:1 (entrada de até 100A e saída de até 100mA). Ainda possui boas propriedades mecânicas, resistência ao ambiente e alta capacidade de isolamento de tensão (Figura 2b).

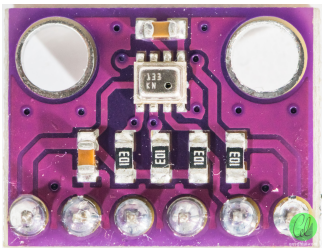
O BMP280 (Figura 3a) é um sensor que vem em um módulo e permite a medição de pressão e temperatura, com interfaces para comunicação I2C ou SPI. Possui tensão de operação de 3,3V e pode ser utilizado com o ESP32 (Tabela II).



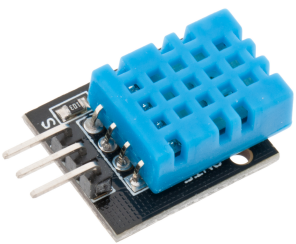
(a) Módulo PZEM-004T.



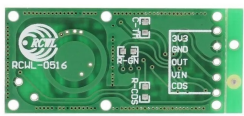
(b) Transformador de corrente.



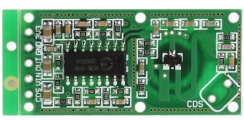
(a) Módulo com o sensor BMP280.



(b) Módulo DHT11.



(c) Módulo RCWL-0516.



(d) Módulo GY49.

Figura 2: Equipamentos para medição elétrica

BMP280	
Tensão de alimentação[V]	3,3
Pressão [hPa]	300 a 1100 ± 0,12
Temperatura [°C]	-40 a 85 ± 1
Interfaces	I2C e SPI

Tabela II: Dados do módulo BMP280.

O sensor DHT11[15] (Figura 3b) é um sensor de umidade e temperatura que utiliza um termistor para medir a temperatura e um sensor capacitivo para medir a umidade do ambiente. Ele possui um controlador de 8 bits que converte os sinais de temperatura e umidade e envia através do pino *data* (Tabela III).

O sensor MAX44009[16], embarcado no módulo GY49 (Figura 3d), possui saída I2C, resolução de 22 bits e opera na faixa dos 0,045 lux até 188 mil lux. A resposta dele é otimizada para ser compatível com a do olho humano e opera na faixa de 1,7V a 3,6V(Tabela IV).

O módulo sensor RCWL-0516[17] (Figura 3c) é um detector de movimento baseado em micro-ondas, ótimo para humanos, pois ignora os movimentos de outros materiais e pode detectar o movimento através de alguns materiais não metálicos (Tabela V).

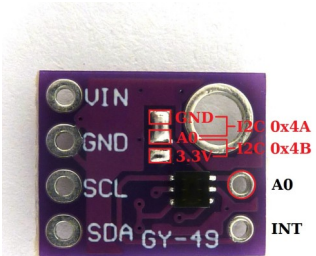


Figura 3: Sensores utilizados.

3.2. Desenvolvimento do firmware

Uma das ideias do sistema é que ele seja flexível, permitindo a implementação para diversos tipos de configurações. Sendo assim, foram desenvolvidas duas classes em C++ para facilitar os projetos.

A primeira classe é para a comunicação do módulo sensor PZEM-004T (Figura 2a). Foi encontrada uma biblioteca[18] que permite a comunicação com o ESP32 somente por *Hardware Serial*. A ideia foi adaptar a biblioteca encontrada para utilização com o ESP32 através de *Software Serial*,

DHT11	
Tensão de alimentação[V]	3 a 5,5
Umidade [%]	20 a 90 $\pm$ 5 @5%RH
Temperatura [$^{\circ}$ C]	0 a 50 $\pm$ 0,2
Histerese [s]	2
Interfaces	Serial

Tabela III: Dados do módulo DHT11.

GY49	
Tensão de alimentação[V]	1,7 a 3,6
Luminosidade [Lux]	45×10^{-3} a 188×10^3
Interfaces	I2C

Tabela IV: Dados do módulo GY49.

RCWL-0516	
Tensão de alimentação[V]	4 a 28
Raio de detecção [m]	5 a 9
Frequência [GHz]	3,2
Saída [V]	0 ou 3,3
Output Timing [s]	2

Tabela V: Dados do módulo RCWL-0516.

Após o desenvolvimento do *firmware* que irá embarcado no ESP32 e conexão com os sensores, os dados serão coletados e enviados para um servidor MQTT-Broker (Raspberry Pi). Todos os sistemas que precisarem consumir as informações obtidas pelos sensores irão se conectar ao servidor e fazer a requisição(Figura 4).

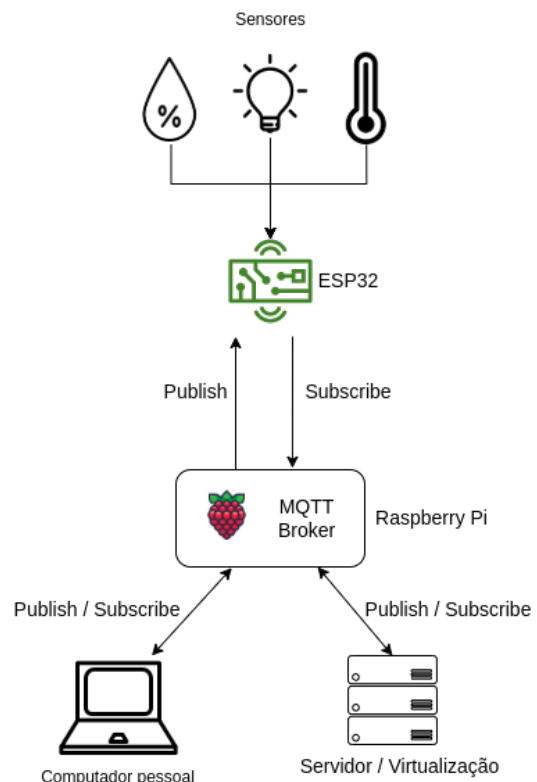


Figura 4: Visão sistêmica da solução.

utilizando o protocolo Modbus (RS-485), já que ele é compatível e possui documentação disponível[19]. Isso possibilitou colocar vários módulos em um único barramento, utilizando apenas dois pinos para comunicação.

Com isso o procedimento para conexão dos sensores passou a envolver duas etapas:

- Etapa 1: Conexão de cada sensor e associação do endereço que ele terá no barramento;
- Etapa 2: Conexão de todos os sensores e requisição dos dados.

Para evitar a necessidade de ter que reprogramar o projeto, toda vez que for utilizar um sensor novo, foi desenvolvida uma classe para “empacotamento” das funções utilizadas pelos sensores, tornando mais simples o desenvolvimento de novos projetos.

3.3. Implementação da arquitetura de comunicação

O protocolo MQTT possui uma filosofia de comunicação baseada em **Editores** (*Publishers*) e **Assinantes** (*Subscribers*). Nesta topologia (Figura 4) todos os **sistemas de medição** (sensores + ESP32) irão se comportar como Editores. Esses editores podem publicar suas informações no servidor (MQTT-Broker), em estruturas definidas por tópicos. O sistema de medição só precisa saber como se conectar a rede e qual o endereço do MQTT-Broker. A partir dessas informações, a implementação do código foi feita para que, ao ser ligado o sistema automaticamente se conecte e envie as informações no intervalo pré-estabelecido.

O MQTT-Broker contém o servidor responsável pela coleta, integração e distribuição dos dados MQTT(Mosquitto[8]) e a ferramenta Node-RED[9], que será utilizada para facilitar a forma como esses dados serão consumidos e transmitidos.

3.4. Integração e testes

Nesta etapa cada sensor é conectado ao ESP32 e são feitos testes de unidade do software, para certificação de que os valores obtidos estejam de acordo com o esperado, para cada sensor utilizado. Após a validação dos resultados obtidos de forma independente, os sensores são conectados um a um, com a consequente verificação, até que todos estejam ligados ao ESP32. A sequência de conexão utilizada foi: DHT11, em

seguida o GY49, BMP280 e RCWL-0516.

4. RESULTADOS

4.1. Sensores de ambiente

Os modelos implementados atualmente são: dallas, dht11, dht21, dht22, gy49, rcwl0516 e bmp280. Com isso, é possível tornar o desenvolvimento futuro mais simples a partir dos métodos que foram implementados. Os métodos genéricos irão se comportar de acordo com os parâmetros utilizados na inicialização do objeto.

Uma das dificuldades que existem no desenvolvimento de projetos IoT é que muitas vezes é necessário trabalhar com vários sensores. Cada sensor possui características que podem ser mais adequadas para o projeto que outros sensores, mas geralmente eles possuem em comum a grandeza que será medida. Por exemplo, os sensores Dallas DS18B20, DHT11, DHT21, DHT22 e BMP280 são capazes de medir a temperatura. Logo, ao invés de precisar modificar o código, caso ocorra a substituição de um desses sensores por outro, basta mudar a forma de inicialização do objeto, pois os métodos genéricos irão se encarregar de trazer o resultado

para o sensor específico.

Após a etapa de programação e configuração do sistema de medição, foi inicializada a fase de configuração da estrutura MQTT e Node-RED, que foi implementada em um Raspberry Pi 4B, com 4GB de RAM e 64GB de espaço de armazenamento, utilizando um cartão de memória microSD.

Após a instalação e configuração o acesso ao sistema pode ser feito através do endereço **http://IP:1880**.

Os tópicos foram definidos através do Node-RED como **lab3i/medida_do_sensor**, conforme a listagem a seguir.

- lab3i/pressure;
- lab3i/temperature;
- lab3i/altitude;
- lab3i/humidity;
- lab3i/lux;
- lab3i/state.

A tela de configuração no sistema é exibida na Figura 5.

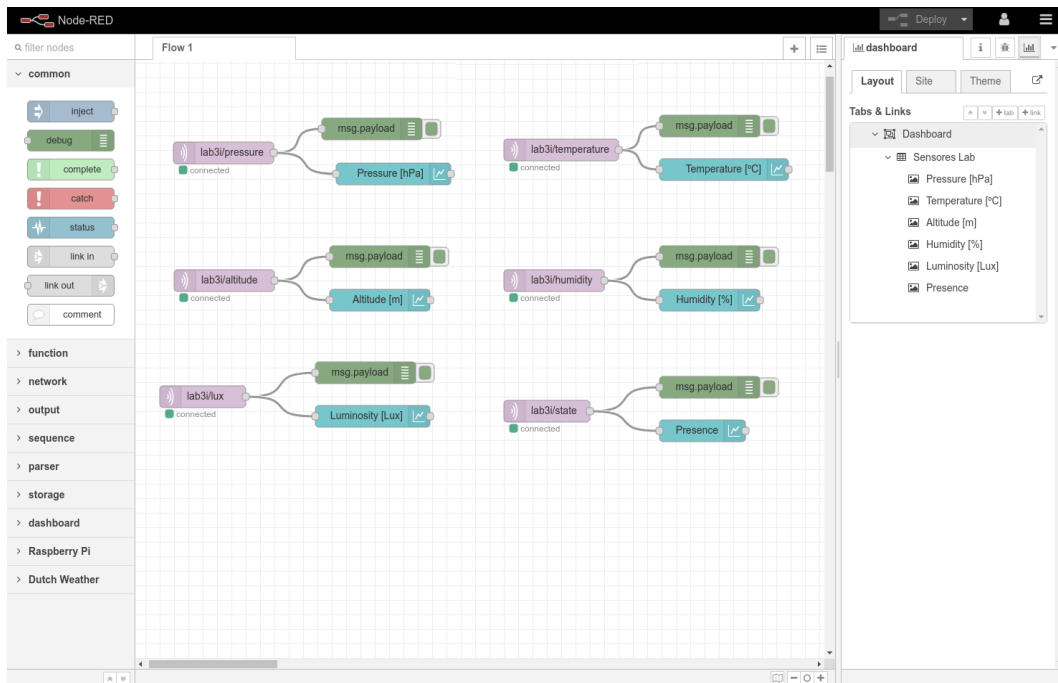


Figura 5: Ambiente de desenvolvimento do Node-RED.

Com o servidor MQTT-Broker devidamente configurado, foi feita a montagem do sistema utilizando um *protoboard* (Figura 6). Como pode ser observado na montagem, os sen-

sores utilizados nos testes foram o RCWL-0516 (presença), GY49 (luminosidade), BMP280 (pressão, temperatura e altitude) e DHT11 (umidade e temperatura).

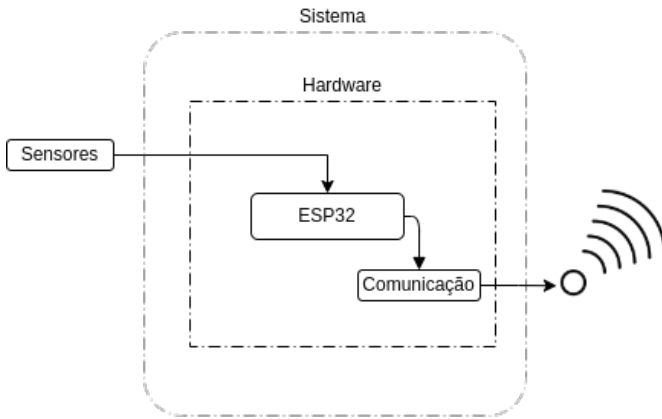


Figura 6: Diagrama da montagem dos sensores.

Após alguns ajustes, a integração do sistema foi feita de forma adequada e foi possível exibir os resultados no *dashboard* do Node-RED.

O painel apresenta de forma simples os dados coletados pelos sensores, sendo de fácil acesso através de qualquer dispositivo que possua um navegador.

Os dados coletados são apresentados em detalhe nos gráficos contendo os dados de pressão (Figura 7), temperatura (Figura 8), umidade relativa (Figura 9), luminosidade (Figura 10) e do sensor de presença (Figura 11).

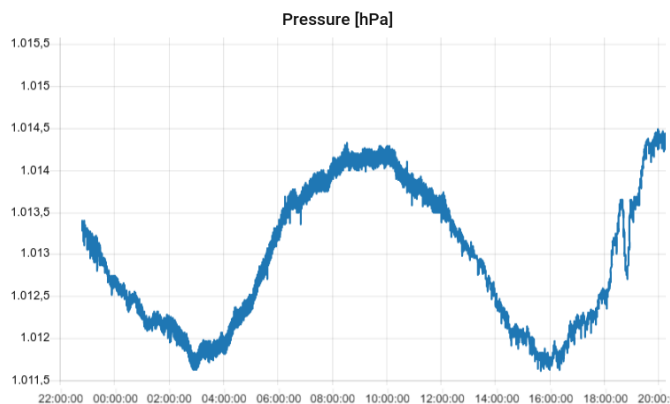


Figura 7: Dados de pressão atmosférica (BMP280).

As variações da umidade (Figura 9) foram testadas com a utilização de um umidificador de ar em uma sala fechada. Com o umidificador ligado e o ar condicionado desligado, foi possível notar o aumento no valor da umidade relativa, que passou a cair quando o aparelho foi desligado e o ar condicionado ligado no ambiente.

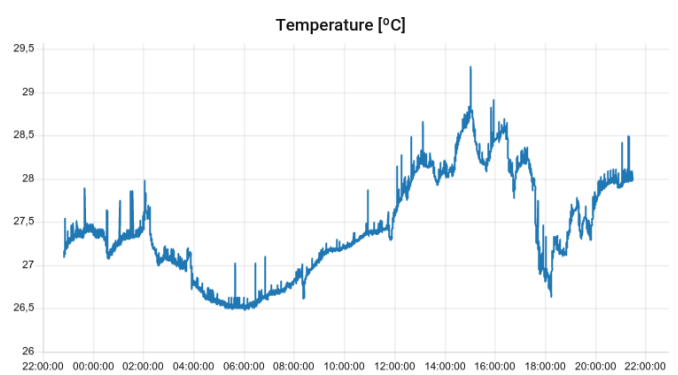


Figura 8: Dados de temperatura (BMP280).

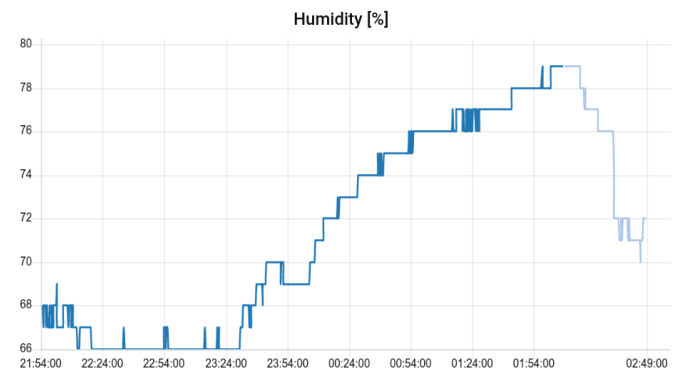


Figura 9: Dados para a umidade relativa (DHT11).

O sensor de luminosidade GY49 apresentou a resposta a luz ambiente conforme pode ser vista na Figura 10. As variações de luminosidade estão de acordo com o esperado.

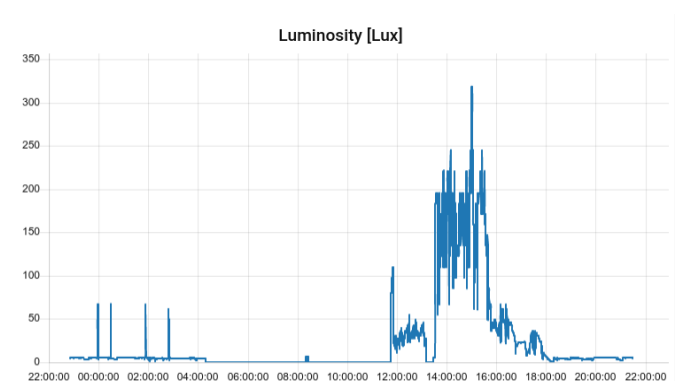


Figura 10: Dados para a luminosidade (GY49).

Após algumas adaptações, que serão discutidas na próxima seção, o sensor de presença passou a responder de forma adequada aos estímulos causados pela movimentação no ambiente (Figura 11).

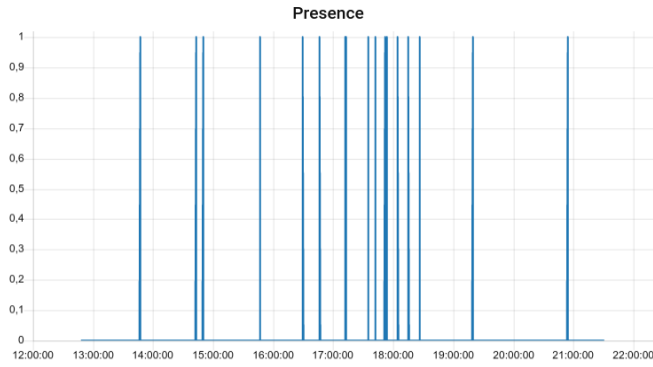


Figura 11: Dados para o sensor de presença (RCWL-0516).

4.2. Medidores de energia

A mesma ideia de implementação foi utilizada na elaboração da classe **PzemModbus**. A inicialização do objeto associado ao sensor pode ser feita das seguintes formas:

A montagem do sistema foi feita em bancada conforme pode ser visto na Figura 12.

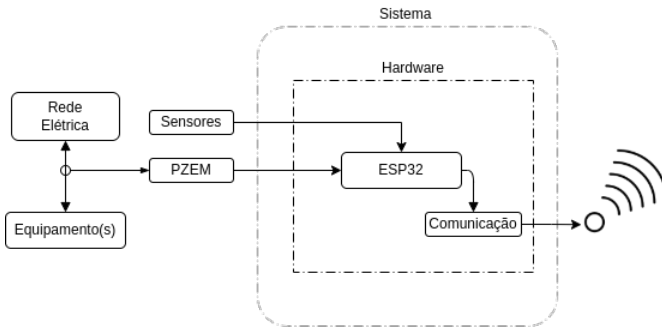


Figura 12: Diagrama do sistema de medição elétrica.

Após a programação o sistema foi conectado e integrado ao ambiente MQTT, de forma semelhante ao que foi feito com os sensores de ambiente. Dentro do ambiente do Node-RED foram criados dois fluxos para testes e visualização dos dados. O primeiro fluxo envolveu a criação do tópico de publicações para um único sensor, para coleta dos dados de tensão, corrente, frequência, potência ativa e fator de potência e do tópico para acionamento do led, que serve para simular o acionamento de um relé.

Após a configuração o painel de visualização do Node-RED exibe os dados coletados de tensão (Figura 13) e corrente (Figura 14).

O segundo fluxo foi criado para visualização das informações obtidas pelos vários sensores para coleta dos dados

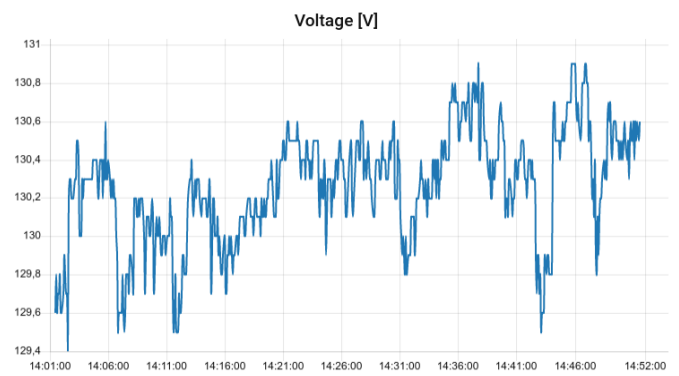


Figura 13: Variação da tensão no ponto de alimentação da carga ao longo do tempo.

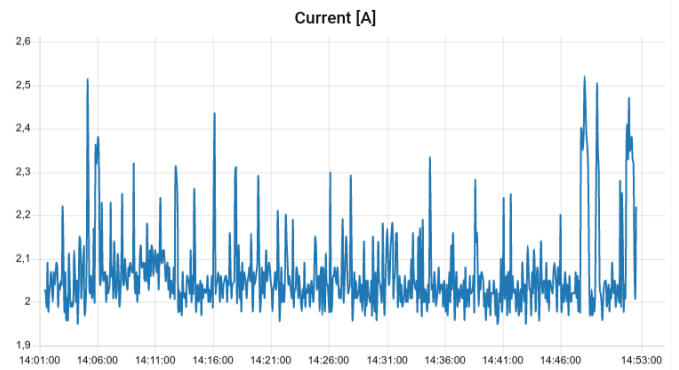


Figura 14: Variação da corrente no ponto de alimentação da carga ao longo do tempo.

de tensão e corrente para os quatro sensores, além do tópico para acionamento do led.

Os dados de tensão para os medidores PZEM-1 e PZEM-2 são exibidos na Figura 15, e para os medidores PZEM-3 e PZEM-4 são vistos na Figura 16.

Neste caso, devido às limitações do ambiente de testes, todos os sensores foram conectados ao mesmo barramento de alimentação da carga. O ponto de conexão é a entrada de um no-break que alimenta um computador, um monitor e um Raspberry Pi. Conforme o esperado, é possível notar que todos os sensores respondem de forma equivalente a variação da tensão.

5. DISCUSSÃO

Durante o desenvolvimento do projeto foram encontradas algumas dificuldades na utilização dos sensores. Após verificar a ocorrência de instabilidade no acesso ao sensor GY49, pois toda hora o sistema deixava de reconhecer, foi visto que

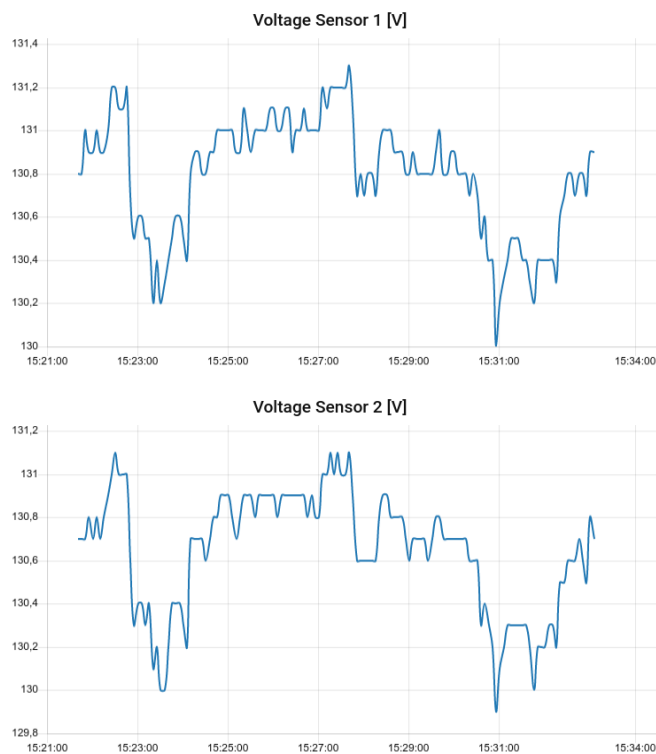


Figura 15: Tensão medida pelos sensores 1 e 2.

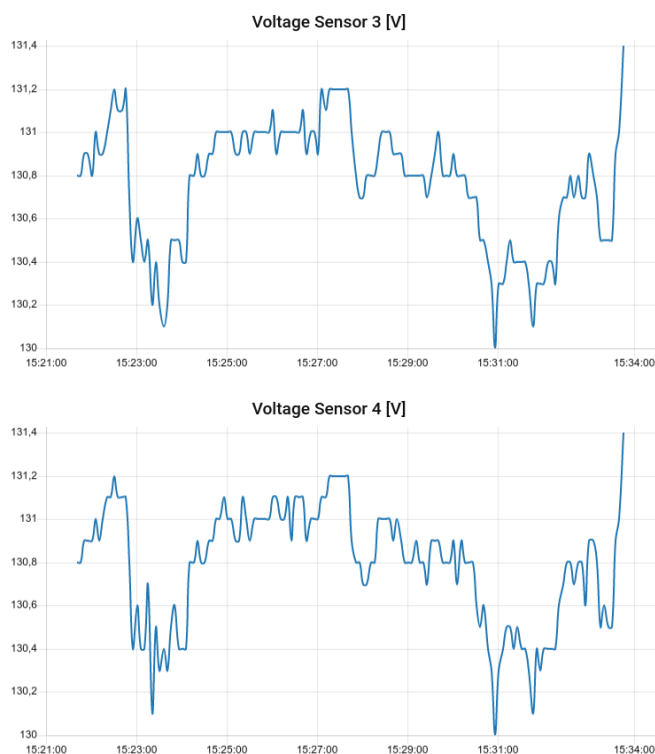


Figura 16: Tensão medida pelos sensores 3 e 4.

é necessário manter um dos pinos extras conectado (Figura 3d), para que o endereço dele não se altere no barramento I2C. A opção feita foi por conectar o pino A0 ao GND, fixando o endereço como 0x4A.

Um outro problema crítico identificado foi a incompatibilidade do sensor RCWL-0516 com o WIFI. Inicialmente o sensor se apresentava extremamente instável, chaveando constantemente entre detecção e não detecção de presença (Figura 17).

Após testes e consulta da documentação foi identificado que este sensor de micro-ondas trabalha na faixa dos 3,2GHz e a ativação da antena WIFI do ESP32 causa interferência[17]. Na verdade, é possível que qualquer antena WIFI cause interferência, então deve-se ter atenção com a localização e posicionamento deste sensor no ambiente.

A alternativa adotada foi programar a parte de utilização desse sensor de modo que a comunicação WIFI seja interrompida e a antena desligada antes da medição e toda a comunicação seja restabelecida após. Com isso o problema foi solucionado, conforme pode ser visto na Figura 18.

Outro problema é que ao se desligar a antena WIFI durante a medição, o sistema perde a capacidade de responder os comandos enviados via MQTT (acionamento remoto de uma led, por exemplo). Para que o sistema se mantenha responsivo é necessário que a conexão não seja perdida, mantendo a função de *callback* ativa.

Logo, no momento, deve-se optar pela utilização do acionamento remoto ou pela utilização do sensor RCWL-0516, pois existe conflito entre eles.

O problema envolvendo o detector de presença baseado

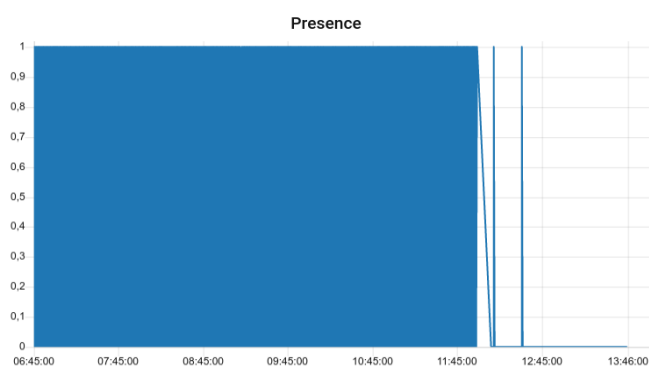


Figura 17: Resposta do RCWL0516 com o WIFI **ligado** durante a medição.

em micro-ondas é interessante e merece maiores investigações no futuro, utilizando equipamentos adequados encontrados em um laboratório de eletrônica, pois, devido a pandemia causada pelo Covid-19, todos os testes foram feitos na residência do autor.

Para a medição de energia foi possível aprender mais sobre o protocolo de comunicação Modbus, pois foi necessário entender como são feitas as requisições e quais são os tipos que o sensor suporta. Isso possibilitou o autor a desenvolver uma classe para comunicação adequado com vários sensores, utilizando um único dispositivo ESP32. Com isso, será possível o monitoramento de vários pontos de alimentação, assim como a medição de potências bifásicas e trifásicas.

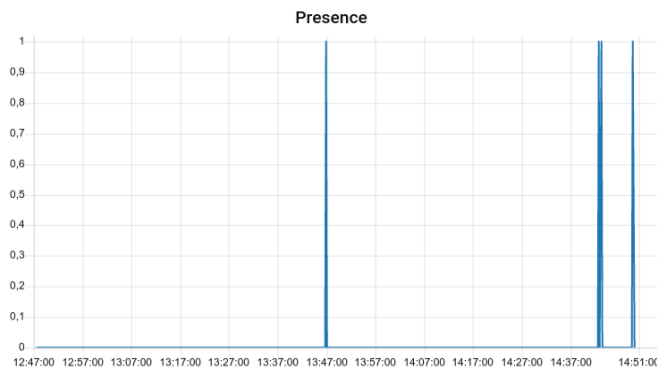


Figura 18: Resposta do RCWL0516 com o WIFI **desligado** durante a medição.

O desenvolvimento desse trabalho permitiu identificar algumas anomalias associadas a medição, assim como o desenvolvimento de uma solução completa baseada em IoT, reunindo todo o conhecimento adquirido durante o curso.

A implementação do protótipo, objeto de estudo desse trabalho, permitiu concluir que o sistema atendeu às necessidades do projeto proposto. Essa ferramenta será de grande utilidade para o desenvolvimento do sistema para monitoramento de laboratórios e experimentos, dentro e fora do CBPF.

- [1] OLSON, N. *Internet of things-from hype to reality: the road to digitization*. [S.l.]: UNIV SHEFFIELD DEPT INFORMATION STUDIES UNIV SHEFFIELD, WESTERN BANK ..., 2019.
- [2] DECRETO No 9.854, DE 25 DE JUNHO DE 2019. 2019. <https://www.in.gov.br/en/web/dou/-/decreto-n-9854-de-25-de-junho-de-2019-173021041>. [Online; Novembro-2021].
- [3] SENADO aprova isenções para incentivar Internet das Coisas; texto vai à sanção. 2020. <https://www12.senado.leg.br/noticias/materias/2020/11/19/senado-aprova-isencoes-para-incentivar-internet-das-coisas-texto-vai-a-sancao>. [Online; Novembro-2021].
- [4] MCTIC, B. e. Internet das coisas: um plano de ação para o brasil, relatório final do estudo-produto 9a, 2018. Retrieved May, v. 7, 2021.
- [5] ACCELERATING the Impact of IoT Technologies: SMEs & Industrial IoT. 2020. <https://www.weforum.org/projects/accelerating-the-impact-of-iot-technologies>. [Online; Outubro-2021].
- [6] ALAHAKOON, D.; YU, X. Smart electricity meter data intelligence for future energy systems: A survey. *IEEE Transactions on Industrial Informatics*, IEEE, v. 12, n. 1, p. 425–436, 2015.
- [7] Kodali, R. K.; Sahu, S. Mqtt based smart metering. In: *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*. [S.l.: s.n.], 2018. p. 399–402.
- [8] ECLIPSE Mosquitto - An open source MQTT broker. 2020. <https://mosquitto.org/>. [Online; Novembro-2021].
- [9] NODE-RED - Low-code programming for event-driven applications. 2021. <https://nodered.org/>. [Online; Novembro-2021].
- [10] Tran, T. V. et al. An application of iot for conduct of laboratory experiment from home. In: *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRON-ICS)*. [S.l.: s.n.], 2020. p. 1–4.
- [11] PERKEL, J. M. The internet of things comes to the lab. *Nature*, Nature Publishing Group, v. 542, n. 7639, p. 125–126, 2017.
- [12] TETRASCIENCE is the world's leading Life Sciences R&D Data Cloud. 2020. <https://www.tetrascience.com/>. [Online; Novembro-2021].
- [13] POGHOSYAN, G. et al. Extracting usage patterns of home iot devices. In: *IEEE. 2017 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.], 2017. p. 1318–1324.
- [14] ESP32 Series Datasheet. 2020. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Online; Novembro-2021].
- [15] DHT11 Datasheet. 2020. <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>. [Online; Novembro-2021].
- [16] MAX44009 Datasheet. 2020. <https://datasheets.maximintegrated.com/en/ds/MAX44009.pdf>. [Online; Agosto-2021].
- [17] RCWL-0516 GitHub informations. 2020. <https://github.com/jdesbonnet/RCWL-0516>. [Online; Outubro-2021].
- [18] PZEM-004T v3.0. 2021. <https://github.com/mandulaj/PZEM-004T-v30>. [Online; Novembro-2021].
- [19] PZEM-004T Datasheet. 2020. <https://innovators.guru.com/wp-content/uploads/2019/06/PZEM-004T-V3.0-Dat>

[asheet-User-Manual.pdf](#). [Online; Dezembro-2021].