

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

**JOÃO EDUARDO SACZUK MACHADO PEREIRA
LUCAS SILVA BENETI**

SOLUÇÃO DE MONITORAMENTO REMOTO DE ESTUFA

CURITIBA

2023

**JOÃO EDUARDO SACZUK MACHADO PEREIRA
LUCAS SILVA BENETI**

SOLUÇÃO DE MONITORAMENTO REMOTO DE ESTUFA

Remote Greenhouse Monitoring Solution

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. André Eugênio Lazzaretti

CURITIBA

2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**JOÃO EDUARDO SACZUK MACHADO PEREIRA
LUCAS SILVA BENETI**

SOLUÇÃO DE MONITORAMENTO REMOTO DE ESTUFA

Trabalho de conclusão de curso de graduação
apresentado como requisito para obtenção do título
de Bacharel em Engenharia Eletrônica da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Data de aprovação: 16/Junho/2023

André Eugênio Lazzaretti
Doutor
Universidade Tecnológica Federal do Paraná

Bruno Sens Chang
Doutor
Universidade Tecnológica Federal do Paraná

Guilherme de Santi Peron
Doutor
Universidade Tecnológica Federal do Paraná

CURITIBA

2023

AGRADECIMENTOS

Gostaríamos de agradecer às pessoas que contribuíram direta ou indiretamente para o desenvolvimento do projeto.

Ao nosso orientador e professor André Eugênio Lazzaretti pelo apoio e auxílio durante o projeto.

Aos nossos familiares e amigos, que nos convenceram a não desistir.

Aos professores da UTFPR que nos guiaram ao longo de nossa jornada acadêmica.

A todas as outras pessoas que participaram de alguma forma da nossa jornada.

RESUMO

Com o crescimento da urbanização, o cultivo interno de plantas em ambientes urbanos tem se tornado mais relevante. No entanto, o mercado falha em oferecer ferramentas para que cultivadores de pequena escala inseridos nesses ambientes possam trabalhar de maneira inteligente. O presente trabalho descreve o desenvolvimento de um dispositivo de monitoramento de estufas que se conecta à rede Wi-Fi e envia dados para um servidor remoto, os quais são apresentados ao usuário por meio de um aplicativo móvel. A solução busca oferecer a cultivadores localizados em ambientes urbanos um meio eficiente e prático de monitorar o ambiente de estufas internas, proporcionando a possibilidade de detecção precoce de problemas e controle mais preciso das condições ideais para o cultivo.

Palavras-chave: telemetria; sistemas embarcados; aplicativo móvel; servidor remoto; cultivo em estufa.

ABSTRACT

With urbanization growth, indoor plant cultivation in urban environments has become more relevant. However, the market fails to provide tools for small-scale growers operating in these environments. The present work describes the development of a greenhouse monitoring device that connects to a Wi-Fi network and sends data to a remote server, which is presented to the user through a mobile application. The solution aims to provide growers located in urban environments with an efficient and practical means of monitoring the indoor greenhouse environment, offering the possibility of early detection of issues and more precise control of the ideal conditions for cultivation.

Keywords: telemetry; embedded systems; mobile applications; remote server; greenhouse cultivation.

LISTA DE ILUSTRAÇÕES

Figura 1 - Pinout da placa de desenvolvimento ESP-32 DEVKIT V1	21
Figura 2 - Módulo do sensor BH 1750	21
Figura 3 - Módulo do sensor SHT41	22
Figura 4 - Módulo do sensor CCS811	23
Figura 5 - Taxas de transmissão e alcance para Wi-Fi, 4G/5G e Bluetooth	25
Figura 6 - Exemplo de dados no formato JSON e XML	28
Figura 7 - Dispositivo Pulse One	30
Figura 8 - Netatmo <i>Smart Weather Station</i>	30
Figura 9 - Dispositivo HT.w da empresa SensorPush	31
Figura 10 - Diagrama da solução	33
Figura 11 - Dispositivo ainda em estado de protótipo	34
Figura 12 - Diagrama esquemático do projeto da PCI dedicada	36
Figura 13 - PCI com os componentes e trilhas traçadas (esq. top layer, dir. <i>bottom layer</i>)	36
Figura 14 - PCI resultante do projeto	37
Figura 15 - Fluxograma da rotina que gerencia o estado AP	39
Figura 16 - Fluxograma da rotina que gerencia o estado WIFI	40
Figura 17 - Exemplos de envio de dados do dispositivo	42
Figura 18 - Exemplo de obtenção dos dados de um dispositivo	43
Figura 19 - Tela inicial do aplicativo	44
Figura 20 - Tela de configuração de dispositivo	45
Figura 21 - Tela de detalhes do dispositivo	46
Figura 22 - Sensores de umidade e temperatura testados	47
Figura 23 - Montagem do teste de umidade	48
Gráfico 1 - Gráfico do teste de umidade dos sensores	49
Figura 24 - Montagem dos sensores juntos para o teste	50
Gráfico 2 - Gráfico do teste de temperatura dos sensores	50
Figura 25 - Exemplo de resposta do servidor à requisição 1.1	52
Figura 26 - Esquema de ligação para o teste de consumo de energia	53
Gráfico 3 - Gráfico da corrente observada durante o teste de consumo de energia	53

Figura 27 - Estufa montada para demonstração da solução 56

LISTA DE QUADROS

Quadro 1 - Valores de umidade, temperatura e dióxido de carbono	17
Quadro 2 - Especificações técnicas da placa de desenvolvimento ESP-32 DEVKIT V1	20
Quadro 3 - Diferentes modos de taxa de transferência permitidas pelo protocolo	24
Quadro 4 - Etapas de funcionamento do dispositivo e quantidade de corrente relacionada	54
Quadro 5 - Custo de prototipação	57

LISTA DE TABELAS

Tabela 1 - Autonomia em relação a capacidade de baterias comuns	55
--	-----------

LISTA DE ABREVIATURAS E SIGLAS

VPD	Diferencial de Pressão de Vapor
DP	Ponto de Orvalho (<i>Dew Point</i>)
Wi-Fi	<i>Wireless Fidelity</i>
SoC	<i>System-on-a-chip</i>
GPIO	<i>General Purpose Input-Output</i>
I2C	<i>Inter-Integrated Service</i>
VOC	<i>Volatile Organic Compounds</i>
eCO ₂	<i>Equivalent Carbon Dioxide</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
SDA	<i>Serial Data</i>
SCL	<i>Serial Clock</i>
CI	Circuito integrado
SSID	<i>Service Set Identifier</i>
bps	Bits por segundo
WPA	<i>Wi-Fi Protected Access</i>
AP	<i>Access Point</i>
LAN	<i>Local Area Network</i>
SSL	<i>Secure Sockets Layer</i>
TLS	<i>Transport Layer Security</i>
ROM	<i>Read Only Memory</i>
JSON	<i>JavaScript Object Notation</i>
XML	<i>Extensible Markup Language</i>
PCI	Placa de Circuito Impresso
NoSQL	<i>No Structured Query Language</i>
IoT	<i>Internet of Things</i>
RAM	<i>Random Access Memory</i>
CPU	<i>Central Processing Unit</i>
RTC	<i>Real Time Clock</i>
ULP	<i>Ultra Low Power</i>
NVS	<i>Non-volatile Storage</i>

LISTA DE SÍMBOLOS

CO ₂	Dióxido de Carbono
H ₂	Hidrogênio
NH ₃	Amônia Anidra
°C	Graus Celsius
ppm	Partes por milhão
V	Volts
A	Ampére
mAh	Miliampere hora
uA	Micro-ampere hora

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivação	14
1.2	Objetivo Geral	15
1.3	Objetivos Específicos	15
2	FUNDAMENTOS TEÓRICOS	17
2.1	Conceitos e métricas importantes para o cultivo de plantas	17
2.1.1	Umidade, temperatura e luminosidade	18
2.1.2	VPD e <i>Dew Point</i>	19
2.2	Hardware utilizado	20
2.2.1	ESP-32	20
2.2.2	Módulo BH1750 (Sensor de luminosidade)	21
2.2.3	Módulo SHT41 (Sensor de umidade e temperatura)	22
2.2.4	Módulo CCS811 (Sensor de dióxido de carbono)	22
2.3	Métodos de comunicação utilizados	23
2.3.1	I2C	24
2.3.2	Wi-Fi e Access Point (AP)	25
2.3.3	HTTP/HTTPS	26
2.4	Software	27
2.4.1	Firmware	27
2.4.2	Servidor	28
2.4.3	Aplicativos móveis	28
2.5	Soluções concorrentes	29
3	DESENVOLVIMENTO	32
3.1	Diagrama da solução	32
3.2	Hardware	33
3.2.1	Projeto da PCI da <i>baseboard</i> final	35
3.3	Firmware	37
3.3.1	Estado AP - <i>Access Point</i>	39
3.3.2	Estado Wi-Fi	40
3.4	Servidor	41
3.5	Aplicativo Móvel	44
4	RESULTADOS	47

4.1	Testes dos módulos de sensores de temperatura e umidade	47
4.2	Testes de envio dos dados por HTTP e HTTPS	51
4.3	Testes do consumo de energia do dispositivo	52
4.4	Fluxo de uso completo da solução final	55
4.4.1	Demonstração da solução final	56
4.4.2	Custo final do protótipo	57
5	CONCLUSÃO	58
5.1	Trabalhos futuros	58
	REFERÊNCIAS	60

1 INTRODUÇÃO

Com o aumento da urbanização e a falta de espaço para o cultivo tradicional de plantas, o cultivo em ambientes internos tem se tornado uma alternativa promissora. Além disso, a demanda por alimentos frescos e a consciência ambiental têm impulsionado a busca por métodos eficientes de produção agrícola em ambientes controlados. Nesse contexto, o monitoramento preciso das condições de cultivo, como luz, umidade, temperatura, dióxido de carbono (CO₂) e valores de Ponto de Orvalho (*Dew Point*) e Diferencial de Pressão de Vapor (VPD), torna-se essencial para garantir o crescimento saudável das plantas.

O cultivo de plantas em ambientes internos apresenta uma série de desafios. A disponibilidade limitada de luz natural, a regulação adequada da umidade e da temperatura, bem como o controle dos níveis de CO₂, são alguns dos fatores críticos para o sucesso do cultivo indoor.

Plantas podem ser divididas em três grupos: plantas de dias longos, dias curtos ou neutras (Aldrich e Bartok, 1994). Pelo simples fato da luz natural não estar necessariamente disponível em um ambiente de cultivo indoor, se faz necessária uma fonte de iluminação artificial cujo espectro esteja próximo do fornecido pela luz solar. A utilização de uma fonte de luz artificial torna possível o aumento do fotoperíodo ao qual a planta será submetida, sendo assim possível simular dias mais longos ou mais curtos. Torna-se possível controlar precisamente o fotoperíodo disponibilizado para as plantas, otimizando assim a fotossíntese realizada.

1.1 Motivação

Para o cultivo otimizado de plantas em ambientes internos, é desejável ter controle total das variáveis do ambiente nas quais a cultura está inserida. É muito importante manter os níveis de umidade, temperatura, luminosidade e dióxido de carbono dentro de uma faixa adequada, sempre levando em consideração o estágio de crescimento no qual a planta se encontra.

Para auxiliar no monitoramento destas variáveis, além dos valores de umidade, temperatura, dióxido de carbono e luminosidade, outras duas métricas podem ser muito úteis para que o cultivador tenha um controle mais preciso: o VPD, que é a diferença entre a pressão de vapor atual e a pressão de vapor máxima que

o ar pode conter a uma determinada temperatura, e o ponto de orvalho (*dew point*), que é o ponto onde o ar se torna saturado e ocorre a condensação. Ambas são métricas fundamentais para que a eficiência da transpiração das plantas possa ser avaliada.

1.2 Objetivo Geral

O objetivo deste projeto é desenvolver um dispositivo digital voltado para o uso em estufas internas e capaz de medir as grandezas anteriormente mencionadas. Os dados serão periodicamente obtidos através de módulos de sensores com os quais o dispositivo é equipado; estes serão processados e enviados para um servidor remoto, que por sua vez irá armazená-los em um banco de dados. O usuário do dispositivo terá acesso a esses valores via aplicativo móvel.

Não é possível encontrar no mercado brasileiro soluções semelhantes voltadas ao segmento de mercado de cultivadores de pequena escala. As soluções disponíveis são mais voltadas para a produção em larga escala. No mercado americano e europeu é possível encontrar algumas opções, como o Pulse One¹, o Sensor Push² e o Netatmo³. Das soluções citadas, apenas o Pulse One se aproxima mais do proposto pelo presente trabalho.

Um ponto importante é que o Pulse One tem um custo de \$199, sem considerar envio e impostos regionais. Para o público alvo do projeto, esse custo pode ser considerado alto. Almeja-se desenvolver uma solução com custos reduzidos.

1.3 Objetivos Específicos

- Projetar dispositivo físico a ser colocado em estufa;
- Selecionar os módulos de sensores a serem utilizados pelo dispositivo;
- Implementar rotina de coleta das variáveis de ambiente no dispositivo;
- Desenvolver um software de servidor a ser executado em uma máquina remota para coleta e armazenamento dos dados obtidos pelo dispositivo;
- Desenvolver um aplicativo móvel dedicado para uso com o dispositivo, permitindo a configuração e gerenciamento eficaz das suas funcionalidades;

¹ Disponível em: <https://pulsegrow.com/products/pulse-one?variant=48139205901>. Acesso em 10 mai. 2023

² Disponível em: <https://www.sensorpush.com/products/p/ht-w>. Acesso em 10 mai. 2023

³ Disponível em: <https://www.netatmo.com/pt-pt/smart-weather-station>. Acesso em 10 mai. 2023

- Enviar dados do dispositivo ao servidor utilizando rede Wi-Fi determinada pelo usuário;
- Realizar testes de comportamento e medições da solução, avaliando o desempenho da solução em termos de estabilidade e eficiência energética.

2 FUNDAMENTOS TEÓRICOS

Plantas possuem três diferentes fases da vida: plântula, período vegetativo e floração (ou frutificação, dependendo da cultura em questão). Para cada fase de desenvolvimento, são necessários níveis específicos de temperatura, umidade, luminosidade e dióxido de carbono.

Para que a planta alcance sua máxima taxa de crescimento em cada fase da sua vida, é necessário que o cultivador tenha medições frequentes dessas grandezas. Dessa forma, a planta poderá transpirar melhor, se alimentar mais e, conseqüentemente, ter um melhor desenvolvimento.

Para auxiliar o cultivador, métricas foram criadas (ou adaptadas de outras áreas) para que se tenha com maior precisão, uma visão de como está o ambiente para a cultura.

2.1 Conceitos e métricas importantes para o cultivo de plantas

Durante o cultivo - tanto interno quanto externo - é importante que se tenha o monitoramento das variáveis do ambiente. Esses valores são de grande importância pois afetam diretamente a fotossíntese da planta, ditando seu crescimento e consumo de nutrientes.

Em cada período da planta é importante que o ambiente de cultivo esteja dentro de uma faixa de valores. Tal faixa varia de acordo com fase de crescimento e espécie da planta. No quadro 1 é possível ver alguns valores gerais para o cultivo de plantas (Nelson e Bartok, 1994; Went, 1953).

Quadro 1 - Valores base de umidade, temperatura e dióxido de carbono

Estágio da planta	Temperatura [°C]	Umidade [%]	CO ₂ [ppm]
Germinação e início do vegetativo	24 - 27	90 - 100	-
Vegetativo	23 - 30	50 - 60	350 - 1200
Flora/Frutificação	23 - 28	40 - 50	400 - 1500

Fonte: Nelson e Bartok (1994); Went, (1953)

É importante lembrar que, embora os valores citados possam ser seguidos para qualquer cultivo, é desejável que sejam encontrados os parâmetros ideais para os quais a planta cultivada melhor se adapta.

Como os parâmetros do ambiente são de extrema importância em todo tipo de cultivo, seja interno ou externo, se faz necessária uma forma eficiente de visualização e monitoramento de dados.

2.1.1 Umidade, temperatura e luminosidade

As medições de temperatura, umidade, dióxido de carbono (CO_2) e luminosidade desempenham um papel fundamental no cultivo de plantas. Esses parâmetros influenciam diretamente o seu crescimento, desenvolvimento e saúde. Monitorá-los adequadamente é essencial para que se obtenha bons resultados.

A temperatura afeta diretamente as taxas metabólicas das plantas e é um fator crítico no cultivo. Cada espécie de planta tem uma faixa de temperatura ideal para o seu crescimento ótimo. Temperaturas muito baixas podem retardar o crescimento, enquanto temperaturas muito altas podem causar estresse térmico e danificar os tecidos das plantas.

A umidade é outro fator vital, pois influencia a absorção de água pelas raízes e a transpiração das plantas. A umidade adequada no ar e no solo é crucial para a absorção eficiente de nutrientes pelas raízes e a transpiração equilibrada das folhas. Umidade excessiva pode levar ao desenvolvimento de doenças fúngicas e problemas de murchamento, enquanto a falta de umidade pode causar estresse hídrico e afetar negativamente o crescimento das plantas.

O dióxido de carbono (CO_2) é essencial para a fotossíntese das plantas. Através da fotossíntese, as plantas convertem dióxido de carbono, água e luz em energia e nutrientes. O monitoramento do CO_2 permite otimizar a concentração desse gás no ambiente de cultivo, garantindo que as plantas tenham acesso suficiente para um crescimento saudável. Níveis inadequados de CO_2 podem limitar a taxa de crescimento e afetar negativamente a produção de biomassa das plantas.

A luminosidade desempenha um papel crucial na fotossíntese, influenciando diretamente a taxa de crescimento. Plantas cultivadas em ambientes internos dependem da iluminação artificial para compensar a falta de luz solar natural.

O monitoramento preciso das variáveis citadas em ambientes de cultivo indoor é de extrema importância para criar condições ideais para o crescimento das plantas. Isso permite que cultivadores possuam uma melhor ideia do estado atual do

ambiente de cultivo: torna-se possível isolar as causas de eventuais problemas e ajustar os parâmetros em momentos de crise.

2.1.2 VPD e Dew Point

Vapor saturado é o estado em que uma substância está na forma de vapor ao mesmo tempo em que está em equilíbrio com sua fase líquida, tendo absorvido a quantidade máxima de calor possível para a temperatura e pressão específicas. Nesse estado, o vapor contém a concentração constante de moléculas de substância, com a taxa de evaporação igual à taxa de condensação, mantendo um equilíbrio térmico.

No contexto de cultivo interno de plantas, o VPD (*Vapor Pressure Deficit*) refere-se à diferença entre a pressão de vapor real do ar e a pressão de vapor saturado que o ar pode conter a uma determinada temperatura. É um parâmetro relevante para a avaliação da disponibilidade de água no ambiente em relação à capacidade das plantas de transpirem e absorverem água através das raízes.

Um VPD adequado é essencial para garantir um equilíbrio hídrico adequado nas plantas (Zhang, 2017; Shamshiri et al., 2018). Quando o VPD é muito alto, o ar está seco demais, o que pode levar à transpiração excessiva e à perda de água pelas folhas. Por outro lado, quando o VPD é muito baixo, o ar está úmido demais, reduzindo a eficiência da transpiração das plantas e sua capacidade de absorver água.

Ao monitorar e controlar o VPD no cultivo interno de plantas, um agricultor pode ajustar a umidade relativa do ambiente, bem como a ventilação e a irrigação, para fornecer condições ideais de crescimento. Manter um VPD adequado ajuda a maximizar a absorção de água pelas raízes, a eficiência da transpiração e o desenvolvimento saudável das plantas, resultando em um crescimento mais robusto e uma melhor produção.

O Ponto de Orvalho (*Dew Point*) é a temperatura na qual o vapor presente no ar poderá condensar (Daly et al, 2015) dada a atual umidade do ambiente. Caso a condensação do vapor ocorra, a planta não conseguirá absorver a água corretamente, fragilizando o cultivo e aumentando os riscos de aparição de fungos indesejados e pragas que se beneficiam de um ambiente com altos níveis de umidade.

2.2 Hardware utilizado

À seguir serão descritos os elementos de hardware utilizados para o projeto: o microcontrolador e os módulos de sensores.

2.2.1 ESP-32

O ESP-32 é um SoC (*System-on-a-chip*) versátil e comumente utilizado em projetos de eletrônica. Possui quantidade razoável de pinos GPIO, comunicação I2C e, dependendo de sua versão, módulo e antena de Wi-Fi.

Para o projeto, foi selecionada a placa de desenvolvimento ESP-32 DEVKIT V1, que conta com um chip ESP32-WROOM-32 (Espressif, 2023). Um quadro com as especificações técnicas pode ser vista no quadro 2.

Quadro 2 - Especificações técnicas da placa de desenvolvimento ESP-32 DEVKIT V1

Categories	Items	Specifications
Certification	RF certification	See certificates for ESP32-WROOM-32
	Wi-Fi certification	Wi-Fi Alliance
	Bluetooth certification	BQB
	Green certification	RoHS/REACH
Test	Reliability	HTOL/HTSL/uHAST/TCT/ESD
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps) A-MPDU and A-MSDU aggregation and 0.4 μ s guard interval support
	Center frequency range of operating channel	2412 ~ 2484 MHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and Bluetooth LE specification
	Radio	NZIF receiver with -97 dBm sensitivity Class-1, class-2 and class-3 transmitter
	Audio	AFH CVSD and SBC
Hardware	Module interfaces	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC, Two-Wire Automotive Interface (TWA [®]), compatible with ISO11898-1 (CAN Specification 2.0)
	Integrated crystal	40 MHz crystal
	Integrated SPI flash	4 MB
	Operating voltage/Power supply	3.0 V ~ 3.6 V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
	Recommended operating ambient temperature range	-40 °C ~ +85 °C
	Package size	18 mm x 25.5 mm x 3.10 mm
Moisture sensitivity level (MSL)	Level 3	

Fonte: Espressif, 2023

O *pinout* da placa pode ser visto na Figura 1. Um fator crucial para a escolha da placa foi a disponibilidade de pinos de comunicação I2C: os módulos mais robustos encontrados do mercado utilizam essa forma de comunicação.

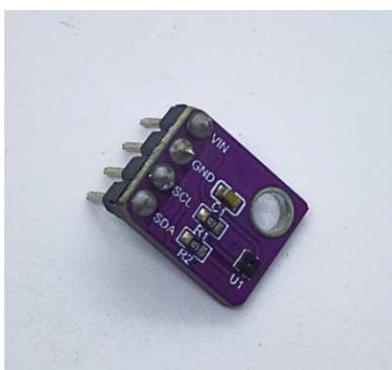
2.2.3 Módulo SHT41 (Sensor de umidade e temperatura)

O SHT41 é um sensor digital de umidade relativa e temperatura com diferentes possibilidades de precisão (Sensirion, 2023). O módulo físico pode ser visto na figura 3. O método de comunicação do sensor é I2C, o que permite manter a operação em baixo consumo de energia.

Em seu encapsulamento, o sensor conta também com um aquecedor interno. Este pode ser utilizado para descondensar as gotículas de água presentes na região da membrana do sensor, aumentando a precisão das medições.

Como o caso de uso do projeto não leva em consideração situações extremas (como acúmulo de gelo ou condensação próxima ao módulo), o uso do aquecedor interno não foi necessário. De qualquer forma, a presença dele foi um fator relevante para a escolha do módulo, que se mostra desejável para melhoramentos futuros.

Figura 3 - Módulo do sensor SHT41

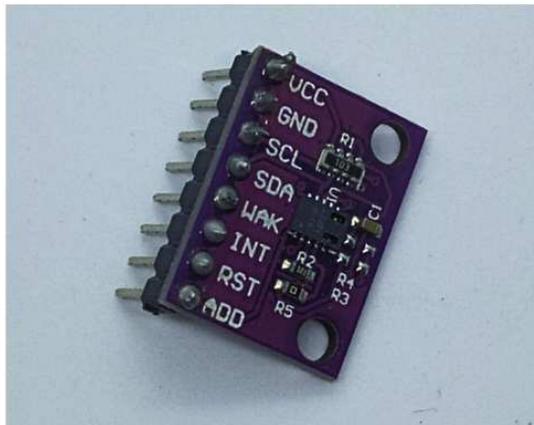


Fonte: Autoria própria (2023)

2.2.4 Módulo CCS811 (Dióxido de carbono)

O CCS811 é um sensor digital de gás *ultra-low power* (ScioSense, 2021) que integra um sensor de óxido metálico capaz de detectar uma variedade de Compostos Orgânicos Voláteis (de sigla em inglês VOCs) para monitoramento de qualidade do ar. Também conta com uma unidade microcontroladora que inclui um conversor analógico-digital e interface I2C.

O sensor converte internamente valores de VOCs para valores de CO₂ equivalente (eCO₂). Uma imagem do módulo pode ser vista na figura 4.

Figura 4 - Módulo do sensor CCS811

Fonte: Autoria própria (2023)

Este sensor possui requisitos de funcionamento diferentes dos sensores mais comumente encontrados no mercado. Um exemplo são os sensores MQ-135⁵ e MQ-5⁶. Ambos os sensores necessitam de alimentação 5V e os fabricantes recomendam que estes sejam calibrados em ambientes controlados. O sensor MQ-5 necessita de um ambiente com concentração de 1000 ppm de H₂, e o MQ-135 necessita de uma concentração de 100 ppm de NH₃.

Segundo o fabricante do sensor CCS811, apesar da calibração ser recomendada, o sensor já é entregue calibrado, pronto para uso. Além disso, é capaz de operar com apenas 3,3V e uma corrente de 30 mA, consumindo apenas 60 mW em seu modo de operação normal.

2.3 Métodos de comunicação utilizados

No decorrer do projeto, diversos métodos de comunicação foram empregados, tais como I2C, Wi-Fi e HTTPS. Cada um desses métodos foi escolhido com base em sua aplicabilidade e relevância para diferentes partes específicas da solução.

⁵ Disponível em: <https://datasheetspdf.com/pdf-file/605077/Hanwei/MQ135/1>. Acesso em: 9 jul. 2023

⁶ Disponível em: https://www.mouser.com/datasheet/2/744/Seeed_101020056-1217478.pdf. Acesso em 9 jul. 2023

2.3.1 I2C

O protocolo I2C (*Inter-integrated Circuit*) foi desenvolvido em 1982 pela empresa Philips Semiconductors, atualmente conhecida como NXP Semiconductors. Sua primeira especificação foi lançada no mesmo ano (NXP SEMICONDUCTORS, 2021). Esse protocolo é projetado para permitir o controle de vários periféricos por meio de um único controlador. Para estabelecer a comunicação, basta conectar os periféricos a duas portas do controlador, uma para o clock serial (SCL) e outra para o envio e recebimento de dados seriais (SDA) do microcontrolador. Essa característica permite que vários periféricos sejam conectados às mesmas duas portas. O I2C é considerado *half-duplex*, pois permite a comunicação bidirecional entre o controlador e os periféricos, podendo ser iniciada por qualquer uma das partes.

No contexto do protocolo I2C, é importante ter um meio de diferenciar os diversos periféricos conectados ao controlador por meio das mesmas duas portas. Para isso, a fabricante do circuito integrado (CI) com interface I2C atribui um endereço de 7 bits a cada periférico, permitindo que o controlador envie comandos específicos, como comandos de leitura, para um periférico em particular. Além disso, o protocolo também prevê a existência de um "endereço geral", que o controlador pode utilizar para se comunicar com todos os periféricos que possuem essa interface.

Atualmente, o protocolo I2C oferece diferentes taxas de comunicação (quadro 3).

Quadro 3 - Diferentes modos de taxa de transferência permitidas pelo protocolo

Modos de velocidade de transmissão	
Standard mode	até 100 kbps
Fast-mode	até 400 kbps
Fast-mode Plus	1 Mbps
High-speed mode	3,4 Mbps

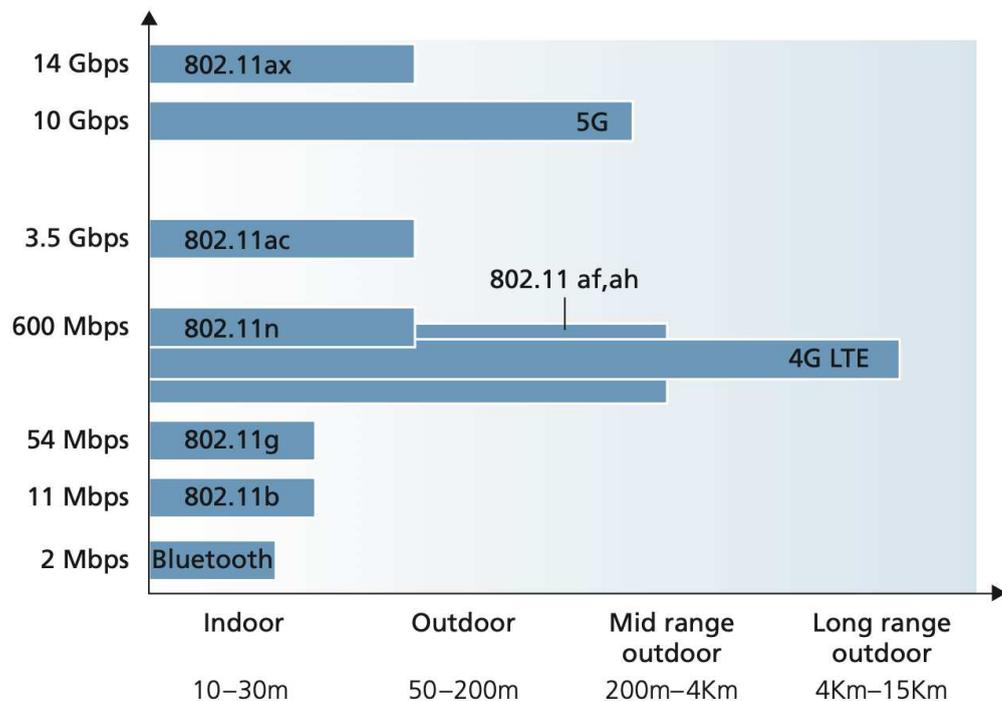
Fonte: NXP SEMICONDUCTORS (2021)

2.3.2 Wi-Fi e Access Point (AP)

A rede sem fio 802.11, conhecida como Wi-Fi, utiliza ondas de rádio para transmitir dados de forma wireless, eliminando a necessidade de conexões físicas por cabos. Para conectar a uma rede Wi-Fi, é preciso conhecer o SSID (*Service Set Identifier*), que é o nome da rede. Em muitos casos, as redes Wi-Fi são protegidas por senhas escolhidas pelo proprietário para garantir a segurança da conexão.

O protocolo 802.11 possui várias implementações, como o 802.11a, 802.11b, 802.11g, entre outros (Kurose, 2020). Cada implementação é projetada com diferentes alcances e taxas de transmissão, conforme ilustrado na figura 5.

Figura 5 - Taxas de transmissão e alcance para Wi-Fi, 4G/5G e Bluetooth



Fonte: Kurose, 2020

Devido à natureza da transmissão de dados por meio de ondas de rádio na rede Wi-Fi, há uma vulnerabilidade em relação à interceptação de dados. Para garantir a segurança, são adotados protocolos de segurança como o WPA (*Wi-Fi Protected Access*) e o WPA2 (Kurose, 2020). Esses protocolos têm a função de criptografar os dados transmitidos e permitir que sejam descriptografados somente por partes autorizadas da conexão.

Um Access Point (ponto de acesso) é um dispositivo de rede que atua como um hub central para conectar dispositivos sem fio a uma rede local (LAN) ou a Internet. O Access Point funciona como um ponto de conexão central para dispositivos, como laptops, smartphones e tablets, permitindo que eles se conectem à rede sem fio e acessem recursos e serviços disponíveis na rede.

Para estabelecer a conexão com uma rede Wi-Fi, é necessária a presença de um dispositivo atuando como *Access Point* (AP). O AP pode ser conectado a um roteador para acesso à rede ou operar como um servidor local exclusivo para armazenamento e compartilhamento de dados.

2.3.3 HTTP/HTTPS

A fim de garantir uma comunicação consistente com os serviços online, é essencial adotar um padrão para as mensagens, solicitações (requests) e respostas. O protocolo HTTP (*HyperText Transfer Protocol*) [RFC 1945; RFC 7230; RFC 7540] (Kurose, 2020) estabelece as diretrizes para o envio e recebimento de informações nesses serviços.

No protocolo HTTP, as mensagens enviadas pelo programa do cliente requerem informações adicionais no cabeçalho da requisição. Essas informações incluem a versão do protocolo (por exemplo, HTTP/1.0, HTTP/1.1 ou HTTP/2.0), o método da requisição (GET, POST, PUT ou DELETE), o formato do conteúdo da requisição (application/json, text/plain, text/html quando aplicável) e as credenciais de acesso ao servidor (quando aplicável). Esses elementos garantem uma comunicação efetiva entre o cliente e o servidor.

Após o programa do cliente enviar a mensagem, é responsabilidade do programa do servidor validar os dados contidos no cabeçalho e determinar a ação a ser tomada com base nesses dados. Assim que a resposta é formada, uma mensagem é enviada do servidor para o cliente, contendo o resultado da requisição. Essa resposta incluirá, no mínimo, um código de status que segue um padrão estabelecido pela comunidade. Esse código de status pode ser interpretado de várias maneiras, mas fornecerá ao remetente uma indicação sobre o sucesso da requisição.

O protocolo HTTPS (*HyperText Transfer Protocol Secure*) é uma implementação do HTTP com uma camada adicional de segurança, utilizando a criptografia SSL/TLS (*Secure Sockets Layer/Transport Layer Security*). Essa

criptografia garante que toda a comunicação entre o cliente e o servidor seja protegida e criptografada, impedindo o acesso não autorizado aos dados transmitidos.

A segurança no protocolo HTTPS é estabelecida por meio da verificação de um certificado emitido por uma Autoridade de Certificação. Quando uma conexão HTTPS é estabelecida, o servidor envia seu certificado contendo informações sobre o proprietário e a validade do certificado. O cliente verifica e valida esse certificado. Após a validação, a conexão segura é estabelecida e todos os dados transmitidos são criptografados, garantindo a segurança da comunicação.

2.4 Software

Software é uma sequência de instruções que, quando interpretadas em uma ordem específica, tem a capacidade de controlar dispositivos eletrônicos ou computadores. Ele representa a maneira de controlar, gerenciar e processar dados em dispositivos físicos, servidores e aplicações em geral.

2.4.1 Firmware

O firmware é um tipo de software responsável por controlar dispositivos eletrônicos, como microcontroladores, roteadores e celulares. Ele desempenha diversas funções, incluindo o controle de entrada e saída de dados, conexão com redes externas, gerenciamento de energia e modos de operação do dispositivo. Assim, o firmware exerce um papel fundamental no funcionamento e no desempenho desses dispositivos.

O firmware é comumente desenvolvido em linguagens de programação que oferecem acesso direto aos recursos do hardware e possuem poucas camadas de abstração. Isso ocorre devido à necessidade de lidar com elementos físicos do dispositivo, como memória e registradores.

Uma das características distintivas de firmwares em relação programas convencionais é a dificuldade de atualização ou modificação dos mesmos, uma vez que geralmente são armazenados em memórias que suportam apenas operações de leitura (ROM - *Read Only Memory*).

2.4.2 Servidor

Um servidor é um programa encarregado de processar as solicitações dos clientes, que são programas responsáveis por fazer requisições ao servidor, seguindo as regras de negócio estabelecidas pelo desenvolvedor. Em síntese, o servidor recebe os dados, efetua as alterações requeridas, armazena as informações pertinentes e responde às requisições dos clientes.

No processo de transferência de dados entre servidor e cliente, seja este um dispositivo microcontrolado ou um aplicativo móvel, é essencial estabelecer um formato padrão para a transmissão dos dados. Duas opções comumente utilizadas são JSON (*Javascript Object Notation*) e XML (*eXtensible Markup Language*). A figura 6 apresenta exemplos de ambos os formatos.

Figura 6 - Exemplo de dados no formato JSON e XML

```
// JSON
{
  "serial": "00000",
  "temperatura": 23.2,
  "umidade": 56.4,
}
// XML
<dadosensor>
  <serial>00000</serial>
  <temperatura>23.2</temperatura>
  <umidade>56.4</umidade>
</dadosensor>
```

Fonte: Autoria própria (2023)

Em sistemas modernos, o formato JSON é amplamente adotado devido à sua simplicidade e facilidade de interpretação por humanos. Essa característica representa a principal vantagem desse formato.

O formato XML é mais adequado em situações em que o sistema é mais antigo e precisa garantir compatibilidade com outros sistemas de maneira mais abrangente.

2.4.3 Aplicativos móveis

Aplicativos móveis são softwares desenvolvidos para dispositivos móveis, como *smartphones* e *tablets*, e são executados em sistemas operacionais

específicos, como Android (Google) e iOS (Apple). Para serem utilizados em um dispositivo, os aplicativos precisam ser desenvolvidos na plataforma nativa do respectivo sistema operacional.

O desenvolvimento de aplicações nativas multi-plataforma (ou seja, disponíveis em mais de um sistema operacional) pode ser complexo. Em teoria, parte do software (ou todo ele, em alguns casos) deve ser reescrito para cada uma das plataformas suportadas.

Algumas ferramentas resolvem esse problema; o código é escrito apenas uma vez e compilado para todas as plataformas desejadas. Uma dessas ferramentas chama-se React Native; este é um framework de desenvolvimento de aplicativos móveis que permite a criação de aplicativos multiplataforma, ou seja, que podem ser executados tanto em dispositivos Android quanto iOS. Ele utiliza a biblioteca React, escrita na linguagem Javascript, para construir interfaces de usuário nativas, proporcionando uma experiência de aplicativo nativo com alto desempenho e acesso a recursos do dispositivo. Isso é possível por meio da tradução automática de componentes e funcionalidades em código nativo, proporcionando eficiência no desenvolvimento e a possibilidade de compartilhar código entre as plataformas.

2.5 Soluções concorrentes

Levando em consideração o caso de uso proposto (cultivo em pequena escala), existem concorrentes apenas nos mercados americano e europeu. São três soluções que se aproximam do projeto proposto em termos de usabilidade e funcionalidade.

O Pulse One, desenvolvido pela empresa americana Pulse, é um dispositivo projetado especificamente para o monitoramento do ambiente de cultivo (Figura 7). Seu modelo mais simples realiza medições de temperatura, umidade e luminosidade. O dispositivo possui aplicativos disponíveis para plataformas iOS e Android, por meio das quais os usuários podem acessar dados como umidade, temperatura, luminosidade, VPD (Diferença de Pressão de Vapor) e ponto de orvalho. O Pulse One pode ser alimentado tanto por bateria quanto por tomada. Seu valor de venda é de \$199 dólares, equivalente a aproximadamente R\$1000 (cotação do dólar R\$ 4,99 em 26/05/2023), sem considerar taxas de importação e frete.

O Smart Weather Station, desenvolvido pela empresa francesa Netatmo, é um dispositivo projetado para o monitoramento do ambiente interno de residências, apresentando um design moderno. A solução consiste em dois dispositivos: um dispositivo sem fio (localizado à direita na figura 8), destinado ao uso externo, como em jardins, e um dispositivo de conexão à tomada (localizado à esquerda na figura 8), para uso em ambientes internos. O dispositivo externo coleta dados de umidade e temperatura, enquanto o dispositivo interno também registra informações sobre a concentração de dióxido de carbono no ambiente.

Figura 7 - Dispositivo Pulse One



Fonte: Página do Pulse

Figura 8 - Netatmo Smart Weather Station



Fonte: Página da Netatmo

A descrição do produto não esclarece o método de comunicação entre os dois módulos, uma vez que o módulo externo não se conecta à *internet*. No entanto, o módulo interno aparenta possuir conectividade com a internet para fornecer os dados coletados. A interação com o usuário é realizada por meio de um aplicativo móvel. O preço da solução é de \$179,00, equivalente a aproximadamente R\$895,00

(cotação do dólar R\$ 4,99 em 26/05/2023), sem considerar taxas de importação ou frete.

O módulo HT.w da empresa americana Sensor Push é uma opção mais econômica, com um preço de cerca de \$69,00, equivalente a aproximadamente R\$345,00 (cotação do dólar R\$ 4,99 em 26/05/2023). No entanto, este é um dispositivo mais simples que mede apenas umidade e temperatura, e não possui conexão com a *internet*. O sensor realiza medições a cada minuto e armazena os dados localmente. Quando o usuário se aproxima do dispositivo e configura o aplicativo móvel correspondente, ele recebe os dados mais recentes coletados. O HT.w funciona com bateria, que tem uma vida útil de até 2 anos em condições normais de operação. A figura 9 ilustra o dispositivo.

Figura 9 - Dispositivo HT.w da empresa SensorPush



Fonte: Página do HT.w⁷

Entre as soluções avaliadas, o Pulse One é o único que se assemelha à proposta do projeto, pois realiza medições e disponibiliza as mesmas grandezas e métricas relevantes para o cultivo interno de plantas. No entanto, devido ao seu alto custo e considerando as taxas de importação e frete, esse dispositivo não é uma opção viável para o mercado brasileiro, com seu valor podendo chegar a aproximadamente R\$2000,00 para o consumidor final.

⁷ Disponível em: <https://www.sensorpush.com/products/p/ht-w>. Acesso em: 10 mai. 2023

3 DESENVOLVIMENTO

A solução desenvolvida consiste em três componentes principais: o dispositivo, o servidor e o aplicativo móvel. O dispositivo é composto por um microcontrolador, módulos de sensores e o firmware responsável pelo controle do hardware. O servidor é uma aplicação em software executada de forma remota, sendo responsável por receber os dados enviados pelo dispositivo, armazená-los em um banco de dados e disponibilizá-los ao aplicativo móvel. Por sua vez, o aplicativo móvel oferece uma interface ao usuário, facilitando a configuração de novos dispositivos e o acesso aos dados de todos os dispositivos sob controle do usuário.

3.1 Diagrama da solução

O dispositivo microcontrolador coleta dados dos sensores, como temperatura, umidade, luminosidade, e dióxido de carbono, e envia esses dados periodicamente para o servidor.

O servidor recebe e processa os dados enviados pelo dispositivo. Ele disponibiliza rotas ou *endpoints* para que o dispositivo microcontrolador possa enviar os dados coletados. Essas rotas são definidas por meio de APIs (Interface de Programação de Aplicações) que permitem a comunicação entre o dispositivo e o servidor.

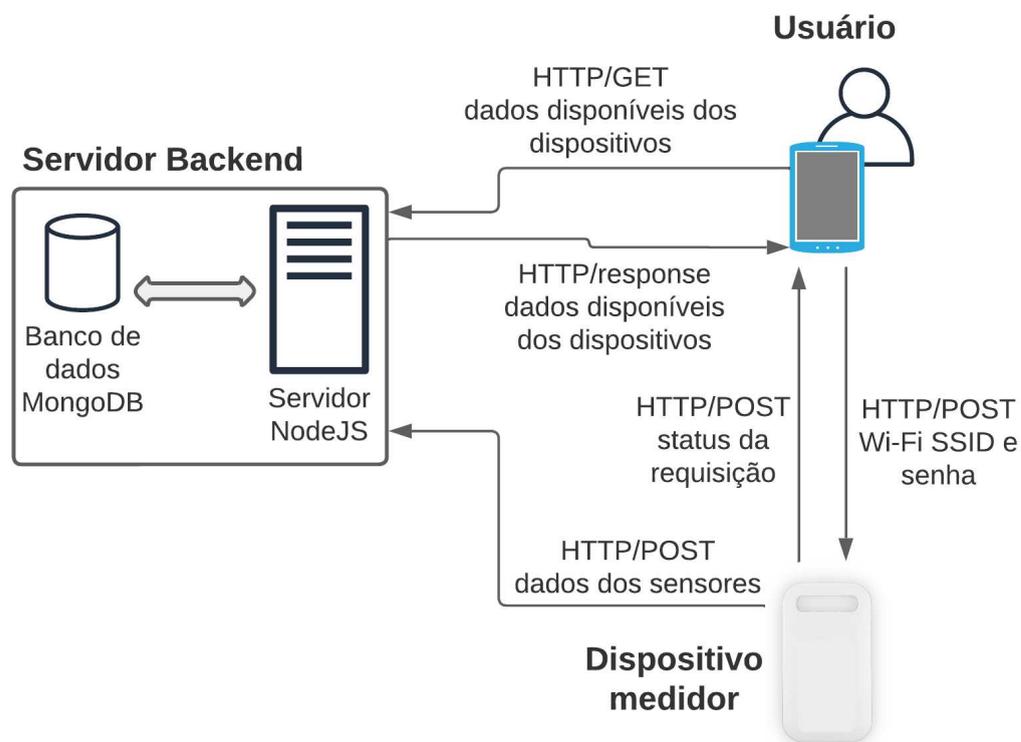
Após receber os dados, o servidor armazena essas informações em um banco de dados não relacional utilizando uma estrutura de dados temporais (*timeseries data*), devido ao alto volume de registros enviados pelo dispositivo. Esse tipo de banco de dados é especialmente adequado para armazenar medições de sensores coletados ao longo do tempo, permitindo consultas eficientes e rápidas para análises posteriores. Bancos de dados não relacionais, como o MongoDB e o Firebase, são amplamente utilizados nesse tipo de aplicação (Mota et al., 2022; Iivari e Koivusaari, 2016), pois oferecem ferramentas e algoritmos projetados para lidar com grandes volumes de dados. Dessa forma, eles são altamente eficientes em termos de busca e organização desses documentos.

Além de disponibilizar rotas para que o dispositivo envie dados dos sensores, o servidor também controla rotas para disponibilizar dados armazenados

no banco de dados para o aplicativo móvel. Este pode fazer solicitações ao servidor para obter os dados dos sensores específicos e recebe as informações organizadas de forma apropriada para a visualização em gráficos. Isso permite que o usuário tenha uma representação visual clara e compreensível dos dados coletados pelos sensores em tempo real ou em períodos específicos.

Uma representação gráfica da arquitetura do sistema da solução pode ser vista na figura 10.

Figura 10 - Diagrama da solução



Fonte: Autoria própria (2023)

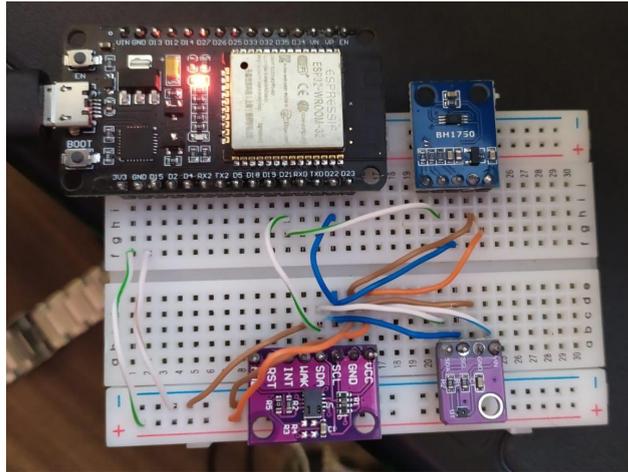
A arquitetura escolhida possibilita uma integração eficiente e flexível das diferentes partes do sistema. A coleta, armazenamento e visualização dos dados dos sensores é feita de maneira organizada e acessível. Os dados são salvos de maneira clara e estruturada e as funcionalidades da solução podem ser facilmente estendidas.

3.2 Hardware

O dispositivo físico consiste de um microprocessador ESP32-DevkitV1, um sensor de luminosidade BH1750, um sensor de umidade e temperatura SHT41 e um

sensor de dióxido de carbono CCS811. Todos estes são conectados a uma placa de circuito impresso feita especialmente para o projeto. A placa é alimentada diretamente com um carregador 5V de entrada micro-USB disponibilizado pela placa ESP32-Devkit V1. Na figura 11 é possível ver o protótipo do hardware montado em uma *protoboard*; facilitando o desenvolvimento e os testes do firmware da solução.

Figura 11 - Dispositivo ainda em estado de protótipo



Fonte: Autoria própria (2023)

No projeto da placa *baseboard*, a adoção de módulos com interface I2C proporcionou uma série de benefícios práticos. Houve uma significativa redução no número de conectores e portas utilizados na placa de desenvolvimento. Mesmo com a incorporação de três módulos de sensores distintos, todos eles foram conectados aos mesmos dois pinos SDA e SCL da placa. Essa estratégia permitiu simplificar o design da placa, otimizando o aproveitamento dos recursos disponíveis e facilitando a integração dos módulos no sistema.

Outro aspecto relevante para a solução, com impacto direto no consumo de energia, é o uso do recurso de hibernação denominado *deep-sleep*. Esse modo permite que a placa entre em um estado de economia de energia, desativando a maioria dos recursos, enquanto alguns permanecem operacionais. Essa estratégia resulta em uma considerável redução no consumo de energia, pois periféricos essenciais que consomem corrente, como a CPU e a interface Wi-Fi durante a transmissão ou recebimento de dados, são desligados. No projeto em questão, as leituras dos sensores são realizadas em intervalos de 5 minutos. Portanto, o modo *deep-sleep* é empregado para evitar que o dispositivo fique totalmente ativo durante períodos nos quais nenhum dos módulos ou interfaces está em uso.

3.2.1 Projeto da PCI da *baseboard* final

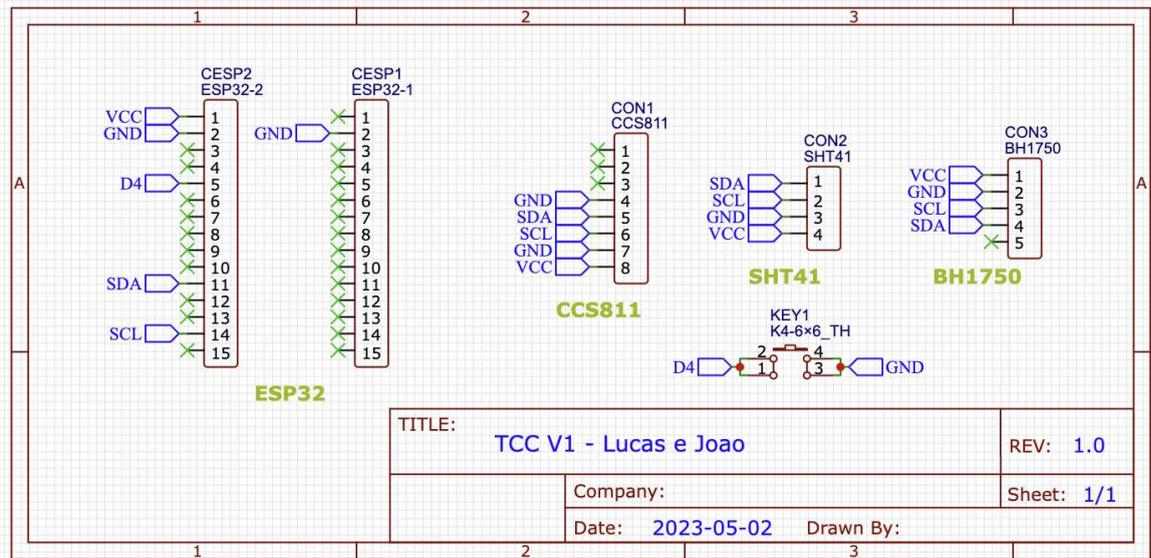
Com o objetivo de aumentar a robustez da solução, foi elaborada uma Placa de Circuito Impresso (PCI) para acomodar de maneira mais adequada os módulos de sensores e o microcontrolador. No contexto de projetos protótipos, como o presente, é comum contar com o auxílio de empresas especializadas que oferecem suporte no desenvolvimento e na geração dos arquivos necessários para a fabricação das PCIs. Nesse sentido, optou-se por utilizar as ferramentas disponibilizadas pela empresa chinesa JLCPCB⁸, que abrangem desde o esquema da placa até as opções de exportação e impressão.

O diagrama esquemático foi elaborado utilizando o editor online EasyEDA Designer, na versão gratuita *standard*. Essa plataforma dispõe de uma biblioteca de componentes fornecida pela empresa JLCPCB, fabricante da placa de circuito impresso. Com essa biblioteca, torna-se viável o desenvolvimento dos esquemáticos e a posterior geração do arquivo necessário para envio à empresa responsável pela fabricação da placa. A figura 12 apresenta o esquemático com os componentes que serão utilizados na placa, evidenciando as conexões necessárias para cada módulo. É de suma importância identificar cada módulo e pino, pois isso facilita consideravelmente a etapa seguinte, que consiste na organização dos componentes no esquemático da placa a ser impressa.

Após a elaboração do esquemático da placa, o próximo passo consiste em converter esse esquemático em um arquivo de placa de circuito impresso (PCI). É necessário organizar os componentes de forma a minimizar o número e o comprimento das trilhas. O resultado final da organização dos componentes e das trilhas de conexões pode ser visualizado na figura 13, onde a posição de cada módulo é indicada por meio de seu desenho e nome do sensor.

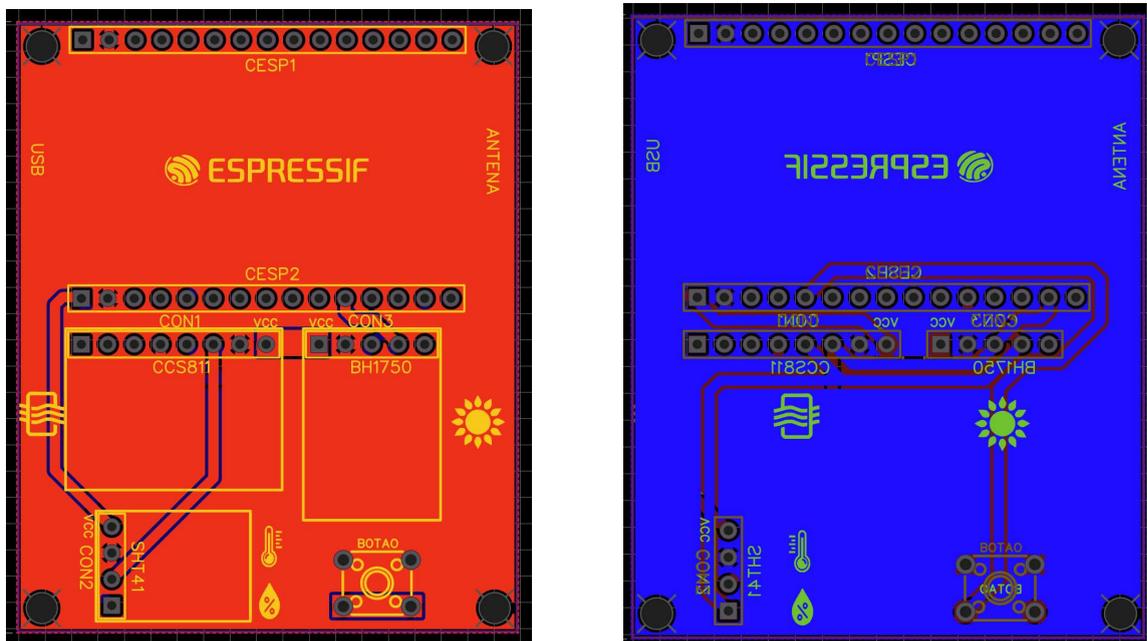
⁸ Disponível em: <https://jlcpcb.com/>. Acesso em: 12 mai. 2023

Figura 12 - Diagrama esquemático do projeto da PCI dedicada



Fonte: Autoria própria (2023)

Figura 13 - PCI com os componentes e trilhas traçadas, (esq. top layer, dir. bottom layer)



Fonte: Autoria própria (2023)

Tanto a posição dos componentes quanto as trilhas (largura, espessura e comprimento) devem seguir alguns padrões básicos do projeto de PCI (Bogatin, 2021; Lima, 2010).

O resultado final da placa base pode ser visto na figura 14, apenas com os conectores para os módulos e para a ESP-32.

Figura 14 - PCI resultante do projeto



Fonte: Autoria própria (2023)

No projeto do dispositivo, os módulos são encaixados em pinos fêmea, que são soldados na placa ao invés de serem embarcados diretamente nela. Nesse sentido, foram utilizados os parâmetros padrões fornecidos pelo fabricante, incluindo a espessura das trilhas. É importante destacar que a fabricante realiza testes na placa após a conclusão da fabricação, a fim de verificar a presença de possíveis curtos-circuitos, sem nenhum custo adicional.

A presença de um módulo Wi-Fi na placa simplificou severamente o projeto. Sua ausência acarretaria na necessidade de mais um módulo externo a ser incluído no dispositivo físico, o que aumentaria a complexidade de roteamento entre os módulos e o risco de interferência entre os mesmos.

3.3 Firmware

Para a implementação do código, foi adotado o ambiente de desenvolvimento integrado (IDE) do Arduino, utilizando bibliotecas convencionais disponíveis da API Arduino. O código foi desenvolvido em linguagem C++ e faz uso de bibliotecas de código aberto, tanto mantidas pela comunidade quanto por fabricantes de módulos de sensores. Um exemplo é a biblioteca da empresa americana Adafruit⁹, utilizada para estabelecer a comunicação com os referidos módulos.

⁹ Disponível em: <https://www.adafruit.com/>. Acesso em: 12 mai. 2023

Conforme mencionado anteriormente, a funcionalidade de *deep-sleep* desempenhou um papel essencial na solução proposta, permitindo que a placa entre em um estado de espera até que um temporizador a reative para retomar suas operações planejadas. Durante esse modo, a CPU principal, a maior parte da memória RAM, os pinos GPIO não controlados pelo RTC e a interface Wi-Fi são desativados, permanecendo ativos apenas o controlador RTC, o co-processador ULP e a memória RTC.

As formas possíveis de despertar a placa neste modo são: via temporizador, via touchpad, via entradas externas ext0 e ext1, e via co-processador ULP. Para o presente projeto, apenas o temporizador é utilizado como fonte de despertar, sendo necessário chamar a função correspondente, especificando a quantidade de tempo em microssegundos que a placa deve permanecer em modo *deep-sleep*.

Considerando a relevância do modo *deep-sleep* para o adequado funcionamento do projeto, tornou-se necessário encontrar uma solução que permitisse armazenar variáveis na placa sem que as mesmas fossem apagadas durante reinicializações. Para atender a essa necessidade, foi utilizada uma funcionalidade da placa denominada *non-volatile storage* (NVS), que funciona como memória e que preserva as informações mesmo durante reinicializações e o modo *deep-sleep*. Quando a placa volta a ser energizada normalmente, é possível acessar os endereços de memória NVS que contêm dados relevantes, como as credenciais da rede Wi-Fi que o usuário deseja que o dispositivo se conecte e o estado atual da placa.

Com o objetivo de assegurar o funcionamento adequado do dispositivo, foram estabelecidos dois estados de operação: o estado AP (Access Point) e o estado Wi-Fi (quando as credenciais já estão armazenadas na memória do ESP-32). Após a inicialização da placa, o código segue um caminho determinado conforme o estado em que se encontra. Os fluxos correspondentes a cada estado serão abordados detalhadamente nas próximas subseções. O código fonte referente ao firmware pode ser encontrado no seguinte endereço <<https://github.com/LucasBeneti/flowering-firmware>>.

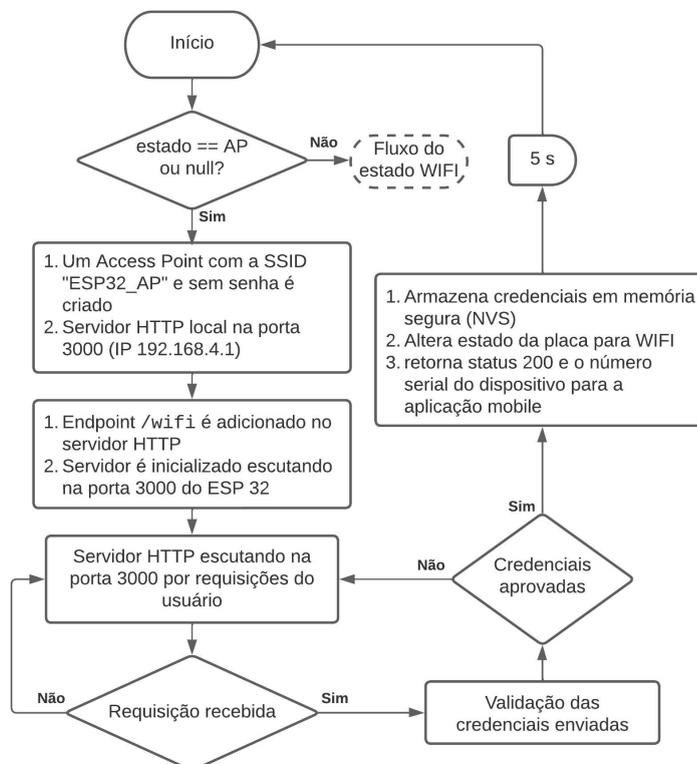
3.3.1 Estado AP - Access Point

O estado AP (*Access Point*) é o estado inicial do dispositivo, uma vez que não há credenciais válidas armazenadas em memória para que o dispositivo se conecte a uma rede Wi-Fi e envie as medições ao servidor.

Nesse estado, o dispositivo atua como um ponto de acesso, disponibilizando sua própria rede Wi-Fi. Ao mesmo tempo em que o ponto de acesso está ativo, o dispositivo também funciona como servidor HTTP, esperando por requisições do usuário conectado.

Uma vez conectado ao Access Point e com acesso ao aplicativo móvel, o usuário pode fazer a configuração inicial do dispositivo. As configurações são enviadas para o dispositivo via requisição HTTP e devem conter o SSID e senha da rede Wi-Fi a serem utilizados pelo dispositivo após a etapa de configuração. Uma explicação detalhada do funcionamento do estado AP (e transição para o estado Wi-Fi) pode ser vista na Figura 15.

Figura 15 - Fluxograma da rotina que gerencia o estado AP



Fonte: Autoria própria (2023)

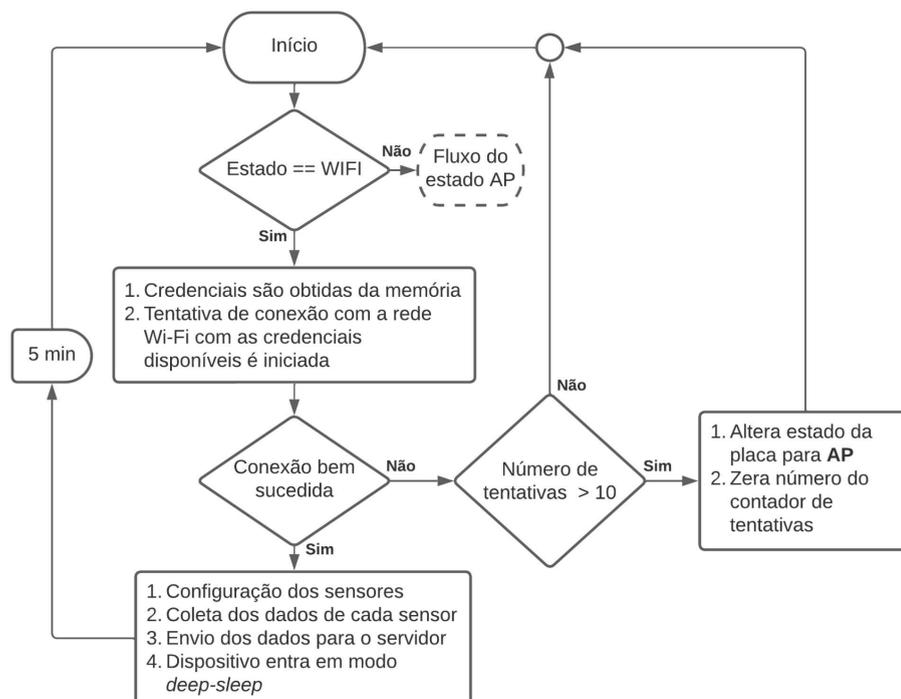
O dispositivo, após receber as credenciais de rede Wi-Fi do usuário, armazena os dados recebidos em memória NVS. Em seguida, o dispositivo é reiniciado e começa a operar em modo WIFI.

3.3.2 Estado WIFI

No estado Wi-Fi, o dispositivo tem as credenciais necessárias para se conectar à rede local selecionada, permitindo o envio dos dados coletados pelos sensores para o servidor na nuvem. Uma vez em modo de operação Wi-Fi, não é necessária nenhuma interação adicional do usuário com o dispositivo (assumindo que não há problemas com a rede de conexão).

Ao inicializar, a placa realiza tentativas de conexão à rede Wi-Fi utilizando as credenciais armazenadas. Caso não seja possível estabelecer uma conexão, a placa entra em modo deep-sleep por 30 segundos. Ao reiniciar, repete-se o processo de conexão por mais 10 vezes, totalizando cerca de 5 minutos de tentativas de reconexão à rede Wi-Fi. O fluxo detalhado desse funcionamento pode ser observado na figura 16.

Figura 16 - Fluxograma da rotina que gerencia o estado WIFI



Fonte: Autoria própria (2023)

No caso de todas as tentativas de conexão com a rede Wi-Fi falharem, a placa altera seu estado de Wi-Fi para AP, apaga as credenciais armazenadas na memória e realiza um reset. Com o estado configurado como AP, ao ligar, a placa seguirá o fluxo descrito anteriormente, relacionado ao modo AP (figura 15).

Caso a conexão com a rede salva seja estabelecida com êxito, as configurações dos módulos dos sensores são iniciadas, seguidas pela obtenção das medições, que são organizadas em uma estrutura de dados com pares de chave e valor. Essa estrutura é serializada e enviada ao servidor remoto por meio de uma chamada específica. Após o envio dos dados, o dispositivo retorna ao modo *deep-sleep* por um período de 5 minutos.

3.4 Servidor

Dentro de uma arquitetura como a da solução proposta, que envolve um dispositivo de *hardware* cuja finalidade principal é coletar dados e cuja única interface com o usuário é um aplicativo móvel, é necessário que seja implementada uma aplicação remota (servidor) capaz de se comunicar tanto com os dispositivos quanto com o aplicativo móvel (clientes). Nesse contexto, sua responsabilidade se resume a receber os dados coletados pelos dispositivos, armazená-los em um banco de dados e fornecer os dados coletados aos usuários do aplicativo. Também é papel do servidor a implementação de qualquer rotina ou procedimento lógico necessário para o correto funcionamento da solução.

Em termos de funcionalidade, o servidor expõe uma API (*Application Programming Interface*) que define os contratos utilizados pelo dispositivo e pelo aplicativo móvel para comunicação. Esta API consiste de rotas HTTP/HTTPS que devem ser acessadas pelos clientes para armazenamento e leitura dos dados, os chamados endpoints. Cada endpoint estabelece um formato de regras para a troca de informação com um cliente.

Na figura 17 é possível ver um exemplo de chamada POST para o *endpoint* de criação de um registro dos dados dos sensores de um dispositivo cujo número de série é “sn425671”.

Por meio desses *endpoints*, o servidor deve receber os dados coletados pelo dispositivo de *hardware*, armazená-los no banco de dados e disponibilizar esses dados para o aplicativo móvel, que é a única interface real com o usuário. As ações tomadas pelo servidor são disparadas à medida que as requisições são realizadas.

Para a implementação do servidor foram utilizados o ambiente de execução Node.js e a linguagem Typescript. Ambas as ferramentas foram selecionadas devido à simplicidade de configuração da infraestrutura envolvida, facilidade para construção de APIs web e facilidade de acesso e edição de informações armazenadas no banco de dados.

Figura 17 - Exemplos de envio de dados do dispositivo

The screenshot displays a REST client interface with the following details:

- Endpoint e parâmetros:** `r.com/measurements/sn425671`
- Método:** `POST`
- Status:** `201 Created` (19.3 s, 230 B)
- Dados sendo enviados (Request Body):**

```

1 {
2   "lux": 74.16666412,
3   "t": 22.36171722,
4   "h": 71.01037598,
5   "co2": 400
6 }

```
- Exemplo de resposta (Response Body):**

```

1 {
2   "message": "Data point created!",
3   "newData": {
4     "t": 23.46189117,
5     "h": 75.12649536,
6     "co2": 400,
7     "lux": 40,
8     "vpd": 0.516,
9     "dp": 18.49,
10    "metadata": {
11      "d_sn": "sn425671"
12    },
13    "timestamp": "2023-05-26T12:58:04.957Z",
14    "_id": "6470acdc408b697ac86422de",
15    "_v": 0
16  }
17 }

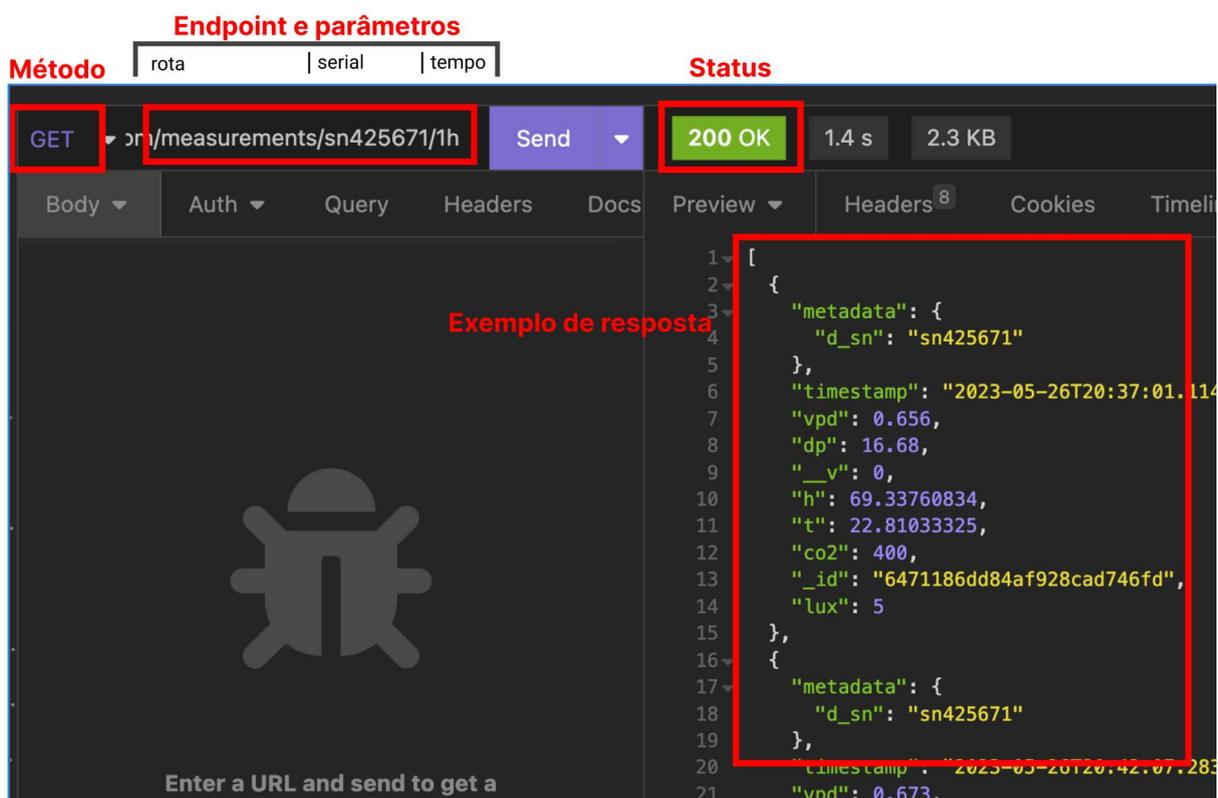
```

Fonte: Autoria própria (2023)

A partir da análise da figura 17, observa-se que na requisição são enviados dados no formato JSON. O servidor, ao receber essa requisição, calcula o VPD e o *Dew Point* a partir das medições de umidade e temperatura. Os dados são então inseridos no banco de dados para eventuais consultas feitas pelo aplicativo móvel.

A figura 18 apresenta um exemplo de requisição para obter os dados das últimas horas de um dispositivo específico. Considerando que o número serial do dispositivo seja o mesmo do exemplo anterior, é necessário realizar uma requisição GET para a rota `/measurements/sn425671/1h`. Nessa requisição é utilizada a rota "measurements" seguida pelo número de série do dispositivo "sn425671". O parâmetro "1h" indica que os dados das últimas horas de operação do dispositivo estão sendo requisitados.

Figura 18 - Exemplo de obtenção dos dados de um dispositivo



Fonte: Autoria própria (2023)

Os registros correspondentes ao dispositivo específico, gerados nas últimas horas e armazenados no banco de dados, serão retornados ao cliente como um *array* de documentos no formato JSON. Dessa forma, o cliente poderá receber, interpretar e exibir os dados ao usuário. Em caso de ocorrência de erros durante a interação entre cliente e servidor, como falta de informações ou tempo de conexão expirado, o servidor fornecerá uma resposta adequada, informando a causa do erro. Essa resposta permitirá que o cliente trate a situação na interface com o usuário, proporcionando uma melhor experiência.

Para esta solução, optou-se pelo uso do banco de dados MongoDB, uma opção NoSQL não relacional. A escolha foi motivada, principalmente, pela capacidade do MongoDB de armazenar dados como séries temporais sem demandar esforços adicionais por parte do desenvolvedor ou cliente. Isso viabiliza a identificação precisa do momento em que cada dado foi armazenado.

O código fonte do servidor pode ser encontrado no seguinte endereço <<https://github.com/flwr-flwr/flower-server/tree/main>>.

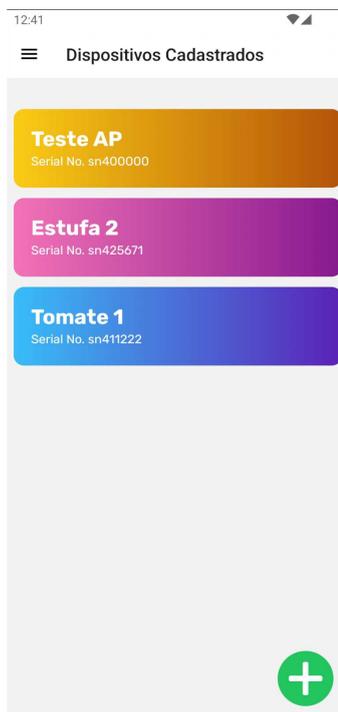
3.5 Aplicativo Móvel

A solução proposta conta com um aplicativo móvel que permite aos usuários visualizar os dados obtidos pelo seu dispositivo. Também suporta cenários onde mais de um dispositivo está sendo utilizado. Por meio dessa aplicação, os usuários podem receber alarmes e notificações em seus dispositivos móveis.

O software por trás do aplicativo móvel foi construído com base na biblioteca React Native, do ecossistema de bibliotecas Javascript. Para facilitar o desenvolvimento e aprimorar a experiência do desenvolvedor, a ferramenta Expo¹⁰ foi utilizado como plataforma de desenvolvimento. Esta fornece um conjunto de ferramentas e bibliotecas que simplificam o processo de criação, teste e implantação do aplicativo.

A tela inicial do aplicativo mostra a lista de dispositivos cadastrados pelo usuário (figura 19).

Figura 19 - Tela inicial do aplicativo



Fonte: Autoria própria (2023)

O usuário deve se conectar ao Access Point disponibilizado pelo dispositivo físico e abrir a tela de configuração de dispositivo (figura 20), onde as credenciais da

¹⁰ Disponível em: <https://expo.dev/>. Acesso em 8 mai. 2023

rede Wi-Fi selecionada devem ser inseridas. Ao enviar os dados, o dispositivo automaticamente é cadastrado e o usuário é redirecionado a tela inicial.

Figura 20 - Tela de configuração de dispositivo



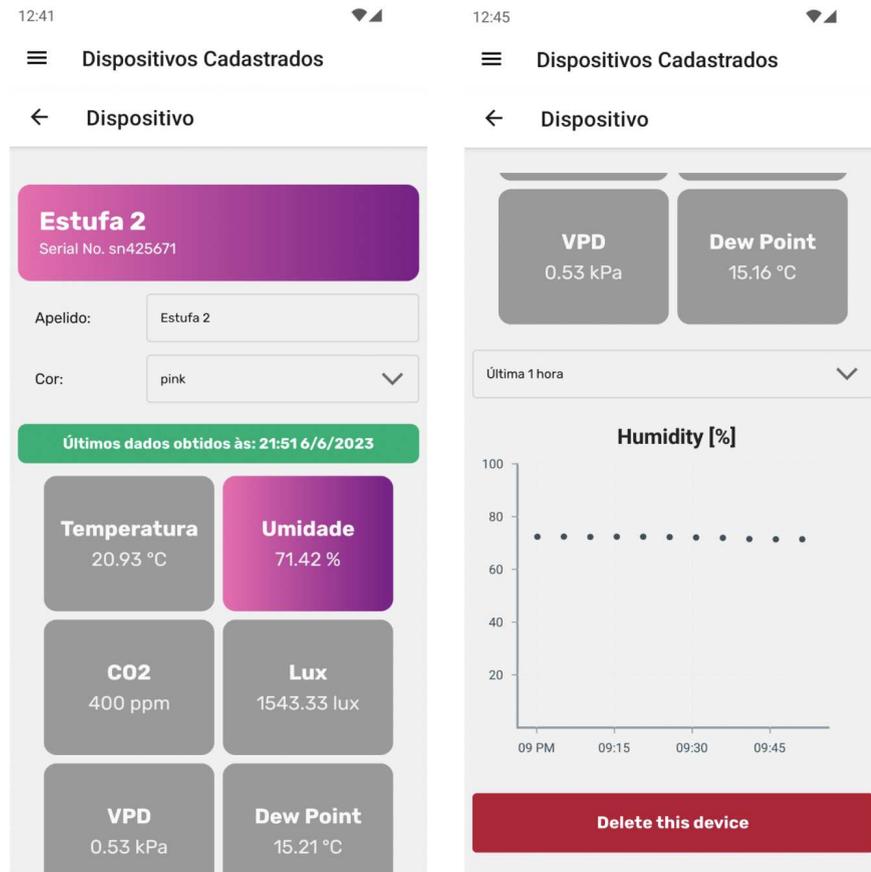
Fonte: Autoria própria (2023)

Os dados de um dispositivo cadastrado no aplicativo podem ser vistos em uma das telas do aplicativo (figura 21). Nessa tela, o usuário poderá visualizar a última medida disponível para aquele dispositivo, contando com gráficos de cada grandeza medida. Para uma melhor experiência, o usuário pode escolher um intervalo de tempo para as medições a serem apresentadas.

O volume de dados coletados por um dispositivo tende a ser grande. Por esse motivo, o aplicativo conta com uma lógica para a persistência local de dados (isto é, no próprio dispositivo que executa o aplicativo móvel) a fim de evitar requisições desnecessárias ao servidor remoto.

O código fonte do dispositivo móvel pode ser encontrado no seguinte endereço <<https://github.com/flwr-flwr/tcc-app/tree/melhorias>>.

Figura 21 - Tela de detalhes do dispositivo



Fonte: Autoria própria (2023)

4 RESULTADOS

Durante o processo de desenvolvimento do dispositivo e das decisões relacionadas ao *software*, foram realizados testes para alcançar a versão final.

4.1 Testes dos módulos de sensores de temperatura e umidade

Para seleção dos módulos de sensores utilizados no projeto, foram considerados critérios como interface I2C para comunicação, encapsulamento único para umidade e temperatura e custo acessível. Com base nesses requisitos, foram escolhidos os sensores AHT10, SHT41 e BME280, que podem ser visualizados na figura 22. Os mesmos foram submetidos a uma série de testes.

Figura 22 - Sensores de umidade e temperatura testados



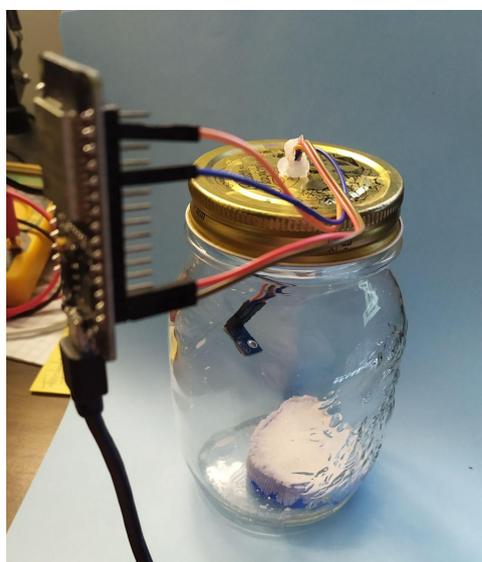
Fonte: Autoria própria (2023)

É desejável que os testes dos sensores sejam feitos em um ambiente cujos níveis de umidade e temperatura possam ser controlados. Tal ambiente não estava disponível no âmbito do projeto apresentado. Devido a esta limitação, para a seleção dos sensores, foi utilizado um método de calibração que utiliza uma solução composta por Cloreto de Sódio (NaCl, sal de cozinha). Esse método consiste em deixar uma porção da solução saturada de Cloreto de Sódio e água em um recipiente hermeticamente fechado. Passadas aproximadamente 24h, a umidade relativa presente dentro do recipiente será de aproximadamente 75% (Greenspan,

1977; Gonzalo et al, 2011), podendo variar alguns décimos dependendo da temperatura do ambiente naquele momento.

Na figura 23 é possível ver o sistema montado para este teste, realizado para cada um dos três módulos selecionados. O ambiente montado consistia de um recipiente hermético e um conjunto de 4 *jumpers* que atravessavam um furo em sua tampa, conectando o ESP-32 ao sensor a ser testado. O sensor é colocado no lado interno do recipiente junto com a solução saturada, enquanto o microcontrolador permanece do lado de fora.

Figura 23 - Montagem do teste de umidade



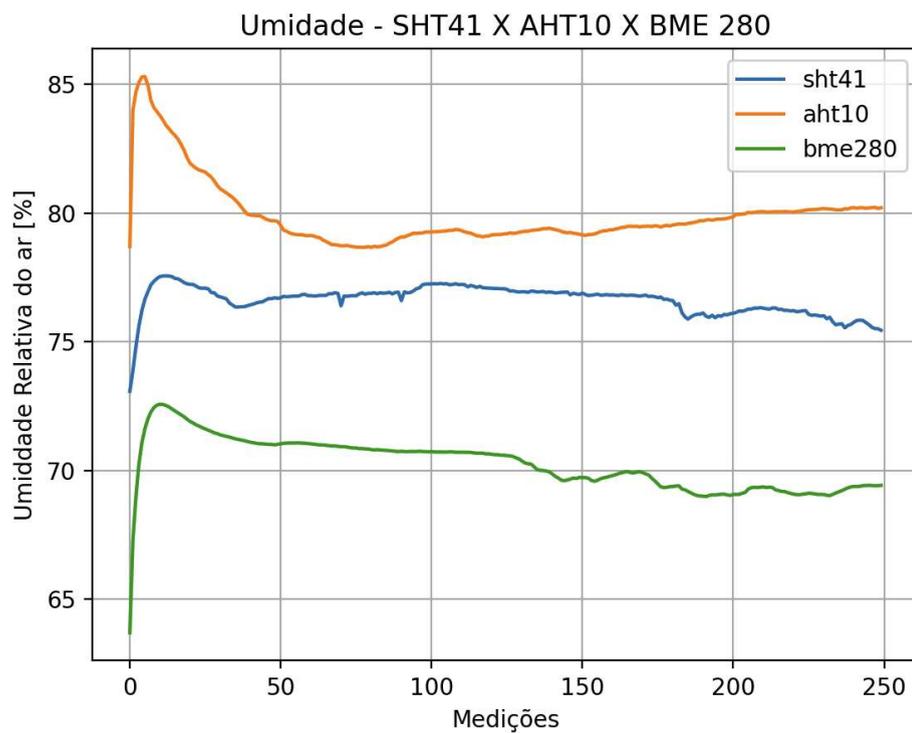
Fonte: A autoria própria (2023)

Para realizar o teste, foi desenvolvido um *firmware* para a placa e um servidor simples. No *firmware*, foram implementadas as configurações, coleta de dados e envio para o servidor. Após o envio, o dispositivo entrava em modo de baixo consumo de energia (*deep-sleep*) por dois minutos e, em seguida, reiniciava o processo. O servidor, por sua vez, salvava os dados obtidos em um arquivo local.

Cada sensor foi testado separadamente. O recipiente ficou fechado por pelo menos 24 horas antes de cada teste. Após essas 24 horas, o respectivo sensor foi inserido no recipiente com a solução e suas medidas de umidade foram coletadas por aproximadamente 8 horas. Tais testes resultaram em 250 pontos de medição para cada módulo.

As medições de umidade (que podem ser vistas no gráfico 1) apresentaram certa discrepância. Analisando os resultados presentes no gráfico, foi possível averiguar que o módulo de sensor SHT41 se aproximou muito do valor de umidade de 75% esperado para o experimento. Os outros dois sensores apresentaram valores de umidade diferentes do esperado: o AHT10 se estabilizou perto de 80% de umidade relativa do ar e o BME280 se estabilizou perto de 70% de umidade relativa do ar.

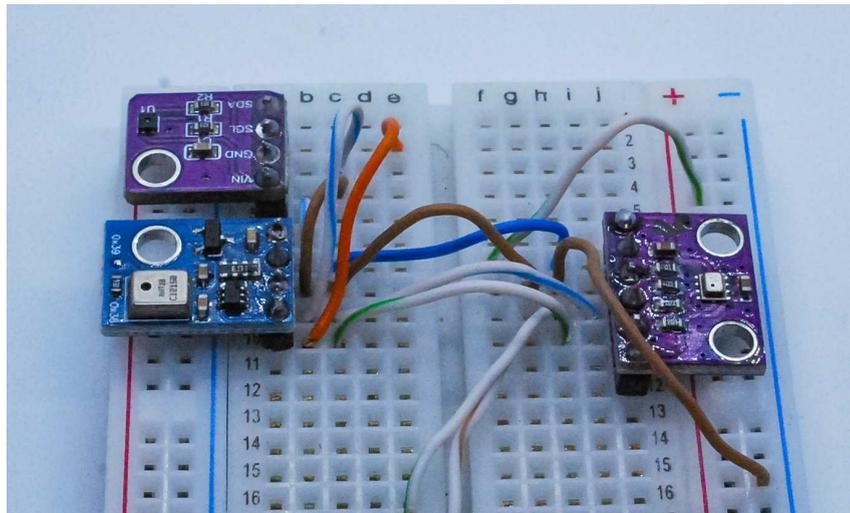
Gráfico 1 - Gráfico do teste de umidade dos sensores



Fonte: Autoria própria (2023)

Devido ao tempo necessário para estabilizar o ambiente de teste de umidade, os testes para os três sensores ocorreram ao longo de aproximadamente 6 dias. Deste modo, não seria interessante comparar as medidas de temperatura dos diferentes sensores devido à oscilação da temperatura ambiente durante esses dias. Foi realizado, portanto, um novo teste: os três dispositivos foram conectados juntos, como pode ser visto na figura 24. Desta forma, as medidas de temperatura dos três sensores foram obtidas simultaneamente.

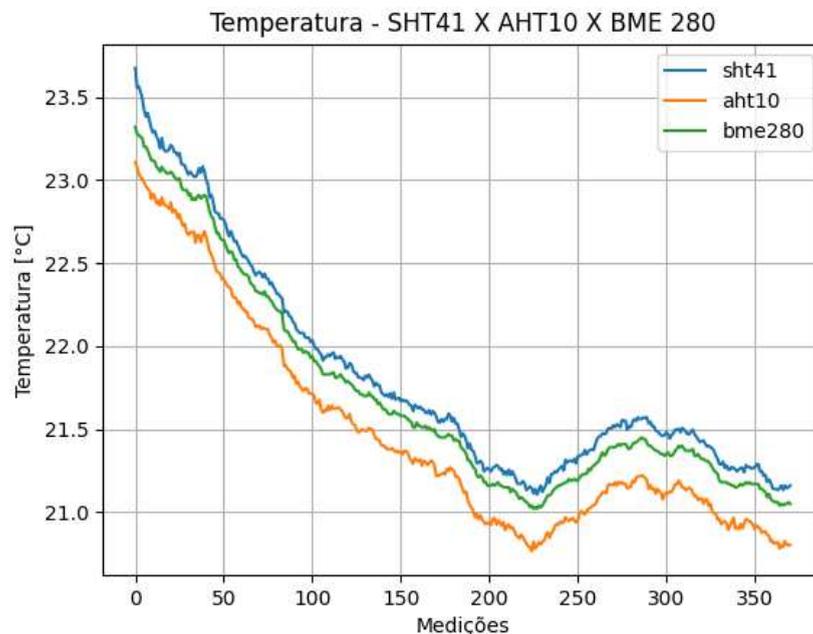
Figura 24 - Montagem dos sensores juntos para o teste



Fonte: Autoria própria (2023)

No que diz respeito às medições de temperatura (que podem ser vistas no Gráfico 2), não foram observadas diferenças significativas que pudessem afetar o bem-estar das plantas.

Gráfico 2 - Gráfico do teste de temperatura dos sensores



Fonte: Autoria própria (2023)

Devido a falta de controle do ambiente, outros aspectos dos sensores foram considerados, como sua construção e funcionalidades. Dentre os sensores, o AHT10 possui o menor tamanho físico, sendo essa sua única funcionalidade de interesse para o projeto. O SHT41 tem um encapsulamento mais reduzido e possui

um aquecedor interno. Isso permite que o sensor opere em condições mais extremas do que os outros sensores. Já o BME280 conta com um barômetro no mesmo encapsulamento, podendo ser utilizado para estimar a altitude ou apenas obter a pressão atmosférica do ambiente; dados estes irrelevantes para o produto apresentado. Outra desvantagem é o seu custo mais elevado.

Após análise, constatou-se que o SHT41 seria o sensor mais adequado para a solução proposta. Durante os testes, esse sensor apresentou os valores mais próximos do esperado de umidade. Além disso, o SHT41 possui um aquecedor interno que pode ser utilizado para a tomada de medidas mais assertivas em condições menos favoráveis do ambiente.

4.2 Testes de envios dos dados por HTTP e HTTPS

Durante a etapa de desenvolvimento da solução, o servidor foi inicialmente executado em máquinas locais, utilizando a própria rede local. O endereço para as requisições tinha a forma `http://192.168.0.1:8080`, como de praxe para endereços da rede local. Após etapas iniciais de prototipação e testes, visando facilitar o restante do desenvolvimento e tornar a solução mais robusta e preparada para cenários de uso reais, o servidor foi hospedado em um serviço de nuvem. Isso possibilita a realização de requisições a partir de qualquer cliente conectado à internet.

Para que ocorra o envio dos dados pelo dispositivo, é indispensável que o servidor esteja em execução e acessível para receber requisições. No dispositivo, são realizadas requisições HTTP utilizando o método POST, e as informações relevantes ao tipo de dado são incluídas no cabeçalho da requisição. O corpo da requisição contém um JSON com os dados de temperatura, umidade, CO₂ e luminosidade do ambiente em que o dispositivo está instalado. O servidor é capaz de receber requisições tanto em HTTP quanto em HTTPS. No entanto, em conformidade com os padrões estabelecidos pela empresa que gerencia o serviço de hospedagem, o comportamento do servidor varia de acordo com o modo pelo qual a requisição é efetuada.

No caso de uma requisição HTTPS, o serviço trata a requisição, realiza a validação e responde de acordo com as regras de negócio do servidor correspondente ao endpoint. No caso de uma requisição HTTP, existem algumas diferenças. Após receber a requisição, o servidor retorna uma resposta semelhante à mostrada na figura 24, com código de status 307, indicando que a requisição deve

ser refeita para outro endereço. Este endereço pode ser encontrado no campo "Location" da resposta.

Figura 25 - Exemplo de resposta do servidor a requisição HTTP 1.1

```
< HTTP/1.1 307 Temporary Redirect
< Date: Sat, 22 Apr 2023 03:00:00 GMT
< Content-Length: 0
< Connection: keep-alive
< Location: https://flwr-server.onrender.com/measurements/sn425671
< CF-Ray: 7bba9d2f2da1a585-GRU
< CF-Cache-Status: DYNAMIC
< Server: cloudflare
< alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400
```

Fonte: A autoria própria (2023)

Essa rotina de redirecionamento não é realizada automaticamente para as bibliotecas nativas do ESP-32. Por este motivo, optou-se por utilizar diretamente o protocolo HTTP/2 ou HTTPS.

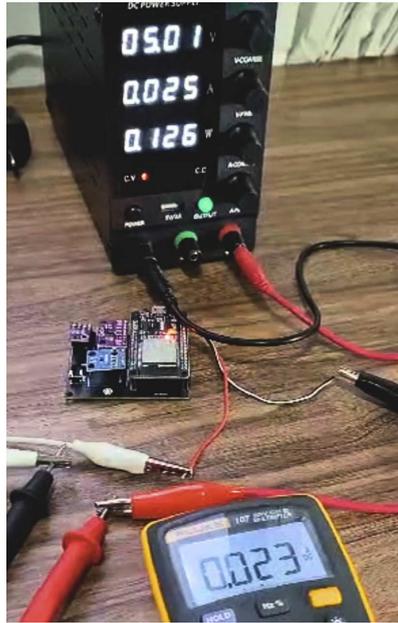
4.3 Teste do consumo de energia

O teste de consumo de energia foi realizado com o intuito de investigar a possibilidade de utilização de baterias como fonte de energia do dispositivo. Em tese, isso facilitaria o uso da solução, já que esse poderia ser posicionado em qualquer lugar da estufa.

Para o teste realizado, a placa é inicializada diretamente no modo de funcionamento de Wi-Fi. Isso significa que o fluxo de funcionamento do dispositivo consiste em se conectar a uma rede Wi-Fi, realizar as medições do ambiente e enviar os dados para o servidor. Após o envio, a placa é colocada em modo *deep-sleep* por 300 segundos.

Na figura 25 é possível ver a forma com que o dispositivo foi alimentado e como as medições de corrente foram aferidas. O dispositivo foi alimentado com 5V e a corrente foi observada durante todo o ciclo de operação.

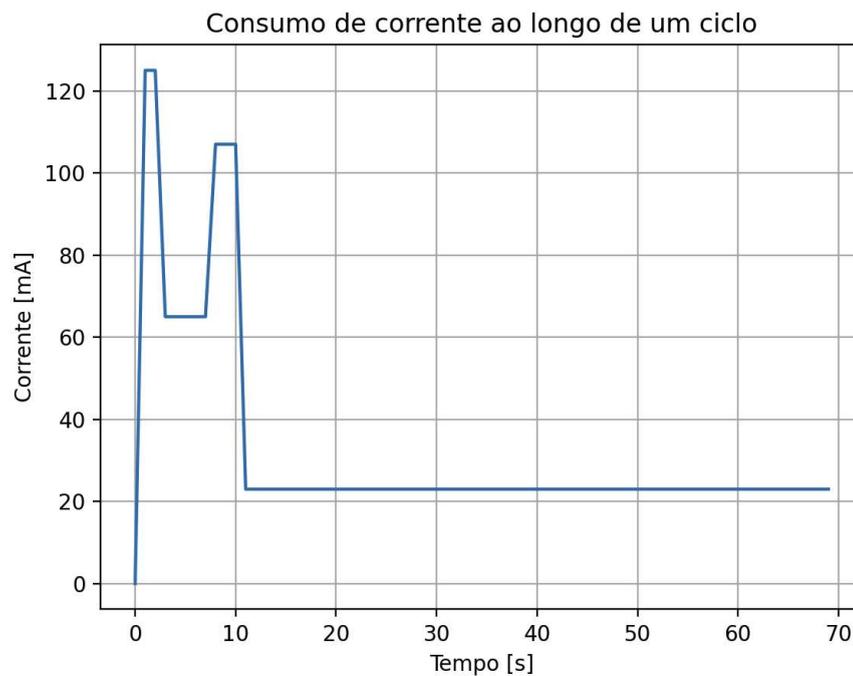
Figura 26 - Esquema de ligação para o teste de consumo de energia



Fonte: Autoria própria (2023)

A partir do momento que o dispositivo é energizado, a corrente oscila consideravelmente, se estabilizando em pelo menos quatro fases diferentes. Estas fases duram poucos segundos e suas durações e consumo de corrente podem ser vistas no Gráfico 3.

Gráfico 3 - Gráfico da corrente observada durante o teste de consumo de energia



Fonte: Autoria própria (2023)

O quadro 4 descreve estas etapas em detalhes, contando também com o tempo de duração de cada uma delas e a corrente média durante aquele determinado período de tempo.

Quadro 4 - Etapas de funcionamento do dispositivo e quantidade de corrente relacionada

Etapa de funcionamento	Duração (s)	Corrente (mA)
Inicialização e conexão à rede Wi-Fi	2	125
Configuração dos sensores e obtenção de dados	5	65
Envio dos dados ao servidor	3	107
Modo <i>deep-sleep</i>	300	23

Fonte: Autoria própria (2023)

Como esperado, as etapas que mais utilizam corrente estão relacionadas à configuração, conexão e envio de dados via rede Wi-Fi.

Para entender melhor o consumo de corrente, seu valor em miliampere-hora foi utilizado. Este foi comparado aos valores encontrados em baterias disponíveis no mercado, a fim de dimensionar uma bateria que forneça uma autonomia adequada ao dispositivo.

O procedimento para estabelecer o referido valor em miliampere-hora foi baseado em uma média ponderada que contemplou o tempo de cada fase e sua respectiva taxa média de corrente consumida (equação 1).

$$I = \frac{(T1 \cdot I1 + T2 \cdot I2 + T3 \cdot I3 + T4 \cdot I4)}{T1 + T2 + T3 + T4} \quad (1)$$

Onde I é a corrente média, T1, T2, T3 e T4 são os tempos de cada etapa e I1, I2, I3 e I4 são as respectivas correntes.

Substituindo os respectivos valores na equação 1, obtém-se:

$$I = \frac{(2 \cdot 125 + 5 \cdot 65 + 3 \cdot 107 + 300 \cdot 23)}{2 + 5 + 3 + 300} = 25,14 \text{ mA} \quad (2)$$

A equação 3 determina o tempo de autonomia da operação de um dispositivo equipado com uma bateria imaginária de 5000 mAh:

$$\text{Autonomia} = \frac{5000 \text{ mAh}}{25,14 \text{ mA}} = 198,89 \text{ h} \approx 8 \text{ dias e } 6 \text{ h} \quad (3)$$

Para uma bateria de 5000 mAh, a autonomia do dispositivo é de aproximadamente 8 dias. Na tabela 1 é possível ver a autonomia para alguns valores de capacidade de baterias mais comuns no mercado.

Tabela 1 - Autonomia em relação a capacidade de baterias comuns

Capacidade da bateria (mAh)	Tempo de autonomia aproximado (h)	Tempo (dias)
2200	87,48	~3,5
3000	119,29	~5
8800	349,93	~14,5

Fonte: Autoria própria (2023)

No contexto de dispositivos relacionados à Internet das Coisas (IoT), a simulação de autonomia não é de grande relevância, uma vez que esses dispositivos são projetados para terem uma autonomia de meses, no mínimo.

Embora as etapas ativas da placa não apresentem problemas significativos em relação ao consumo de energia, identificou-se que a etapa de modo *deep-sleep* consumiu uma corrente próxima de 23 mA, ao invés da faixa dos microamperes (μA) esperada. Isso se dá, possivelmente, pelo LED interno da placa de desenvolvimento, que não foi desativado mesmo não estando em uso. O mesmo acontece com os módulos dos sensores, que continuam energizados durante todo o período de *deep-sleep*.

Com base nos valores identificados e considerando as baterias mais comumente disponíveis no mercado, concluiu-se que o dispositivo não se beneficiaria significativamente ao ser alimentado por bateria. Além disso, a utilização desse tipo de bateria requer a implementação de circuitos de proteção contra possíveis picos de corrente e, em alguns casos, um circuito conversor de tensão que garanta uma alimentação estável.

4.4 Fluxo de uso completo da solução final

O fluxo de uso da solução completa envolve todas as partes anteriormente descritas.

O usuário adquire o dispositivo físico e o coloca em uma estufa interna, que deve estar dentro do alcance de uma rede Wi-Fi com acesso à internet. O usuário deve realizar o download do aplicativo móvel em seu celular.

Após ligar o dispositivo, o mesmo passa a atuar como Access Point, criando um novo ponto de acesso visível para outros dispositivos. O usuário deve então alterar a rede Wi-Fi do seu celular para o *Access Point* disponibilizado pelo

dispositivo. Uma vez conectado ao *Access Point*, o usuário acessa a tela de configuração do dispositivo dentro do aplicativo móvel. Nessa tela, devem ser fornecidas as credenciais da rede Wi-Fi (SSID e senha) à qual o dispositivo físico deve se conectar após a configuração inicial.

Após a inserção das credenciais, o usuário é redirecionado imediatamente para a tela principal do aplicativo, onde o dispositivo é cadastrado automaticamente na lista de dispositivos disponíveis. O dispositivo físico deixa de atuar como *Access Point* e passa então a operar no modo Wi-Fi, conectando-se à rede especificada durante a configuração.

A partir desse momento, o usuário pode utilizar o produto normalmente. Ele poderá conectar seu celular novamente à internet e utilizar todas as funcionalidades disponíveis no aplicativo móvel.

4.4.1 Demonstração da solução final

O teste da solução final consiste em demonstrar o funcionamento da solução completa, com cada parte integrada. Uma estufa foi montada para tal demonstração e dentro dela estão alguns elementos para melhor simular o ambiente externo, como exaustor, ventilador para a ventilação e painel de LED para iluminação, como pode ser visto na figura 26. O objetivo do teste é mostrar a solução completamente integrada sendo utilizada em um sistema de cultivo interno.

Figura 27 - Estufa montada para demonstração da solução



Fonte: Autoria própria (2023)

Uma demonstração em vídeo está disponível (Beneti e Pereira, 2023) e explica com mais detalhes o funcionamento da solução completa.

4.4.2 Custo final do protótipo

O custo do protótipo foi de aproximadamente R\$ 243,16, como pode ser visto no quadro 5. A placa de circuito impresso, por mais que tenha sido fabricada e vinda da China, não sofreu taxa o alfandeg ria. Alguns componentes foram comprados da China e alguns do Brasil, todos os componentes est o listados no quadro 5 com seus respectivos custos.

Alguns pontos dificultam a estimativa do custo para a produ o do dispositivo em larga escala. Fatores como exig ncias regulamentares, fretes agregados para produtos comprados em grande quantidade, tributos e custos de m o-de-obra.

Quadro 5 - Custos da prototipa o

Pe�a	Pa�s da compra	Custo (R\$)
M�dulo de sensor CCS811	Brasil	89,90
ESP32-DevKitV1	Brasil	64,90
Push-button	Brasil	2,05
M�dulo de sensor BH1750	Brasil	15,48
M�dulo de sensor SHT41	China	14,61
Placa de circuito impresso	China	56,22
Total		243,16

Fonte: Autoria pr pria (2023)

Mesmo assim, estima-se que o valor do produto final seja consideravelmente mais barato quando comparado aos concorrentes. Al m disso, o dispositivo apresenta funcionalidades que n o podem ser encontradas nos dispositivos concorrentes, como   o caso de medi es de CO₂.

5 CONCLUSÃO

Foi desenvolvido um dispositivo de monitoramento de estufas que se conecta à Wi-Fi e envia dados para um servidor remoto, os quais são apresentados ao usuário por meio de um aplicativo móvel. A solução busca oferecer aos cultivadores um meio eficiente e prático de monitorar o ambiente das estufas, facilitando a detecção precoce de problemas e controle mais preciso das condições ideais para o cultivo.

Os sensores utilizados foram selecionados cuidadosamente, garantindo a funcionalidade e confiabilidade do dispositivo. A integração com a rede Wi-Fi permitiu a transmissão contínua dos dados coletados para o servidor remoto, enquanto o aplicativo móvel foi desenvolvido com uma interface intuitiva e amigável, proporcionando aos usuários acesso fácil e rápido aos dados da estufa .

Foram realizados diversos testes para avaliar o desempenho do dispositivo de monitoramento. O dispositivo foi mantido ligado durante períodos muito longos de tempo sem problemas em seu funcionamento. Durante os testes, foram coletados dados em uma estufa própria. Os resultados demonstraram que o dispositivo foi capaz de monitorar com relativa precisão variáveis críticas para a manutenção e sucesso da cultura. Isso contribui para um cultivo mais eficiente e sustentável em espaços limitados, minimizando perdas e otimizando a produção agrícola nas áreas urbanas.

A solução proposta apresenta vantagens para os usuários em termos de usabilidade. O dispositivo físico possui um tamanho pequeno, oferecendo uma configuração inicial simples por meio de um aplicativo móvel dedicado. A interface do usuário é intuitiva e suporta múltiplos dispositivos.

No entanto, existem algumas limitações a serem consideradas. A solução depende de uma rede Wi-Fi funcional, o que pode restringir seu uso em áreas com pouca cobertura. Seu consumo de energia é elevado devido a limitações da implementação, mas estas podem ser contornadas com melhorias futuras.

5.1 Trabalhos futuros

Fica evidente o potencial do dispositivo da solução para melhorar as possibilidades de cultivo em áreas urbanas, aumentando a eficiência e reduzindo as perdas.

Embora os resultados sejam promissores, reconhece-se que ainda existem desafios a serem enfrentados.

Em seu atual estado, o dispositivo físico é ligado diretamente à tomada. Seria interessante que o mesmo fosse alimentado por uma bateria recarregável, possibilitando que este pudesse ser posicionado em qualquer local da estufa sem grandes empecilhos. Um dos fatores que impedem o uso de bateria é o grande consumo de energia por parte do dispositivo. Tal fator pode ser mitigado com ajustes ao firmware e hardware da solução: os sensores conectados à placa podem ser desligados quando a mesma entra em modo de deep-sleep; os LEDs indicadores do dispositivo podem ser ajustados a fim de reduzir o consumo de energia.

Também existem falhas relacionadas à questão de usabilidade do dispositivo: no momento, não há maneira de fazer com que o dispositivo volte a funcionar como *Access Point* arbitrariamente. O dispositivo também não possui nenhum armazenamento interno de medições, o que possibilitaria seu funcionamento contínuo mesmo em casos onde não há conexão com a *internet*: medições poderiam ser armazenadas e enviadas ao servidor posteriormente.

O aplicativo móvel também pode ser. Uma possível funcionalidade a ser adicionada é a de relatórios da estufa por período de tempo, o que daria ao usuário uma visão mais geral de como seu cultivo tem evoluído ao longo do tempo. Outra funcionalidade é a de alarmes: o usuário determinaria “valores limite” para determinadas variáveis do ambiente, recebendo notificações quando esses mesmos valores fossem atingidos. Isso possibilitaria uma reação mais imediata a possíveis problemas decorrentes dessas variações.

REFERÊNCIAS

ALDRICH, R. A., BARTOK, J. Jr. W., **Greenhouse operation and management**. Ithaca, N.Y: Northeast Regional Agricultural Engineering Service, 3rd ed., 1994.

BENETI, L. S., PEREIRA, J. E. S. M. **TCC - Solução de monitoramento remoto de estufas**. Lucas Silva Beneti, 06 jun. 2023. Disponível em: <https://www.youtube.com/watch?v=b5i-KybwVOU>. Acesso em: 6 jun. 2023.

BOGATIN, E., **Bogatin's practical guide to prototype breadboard and PCB design**. Artech House, 2021.

DALY, C., SMITH, J. I., OLSON, K. V. **Mapping atmospheric moisture climatologies across the conterminous United States**. PLOS ONE, 2015. DOI: 10.1371/journal.pone.0141140.

Espressif, **ESP32-WROOM-32 Datasheet**, 2023. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datash_eet_en.pdf. Acesso em: 10 mar. 2023.

GONZALO, Q., MIGUEL, A., JOAQUIM, B., et al. **Use of salt solutions for assuring constant relative humidity conditions in contained environments**. Fundação para a Ciência e a Tecnologia. 2011.

GREENSPAN, L. **Humidity fixed points of binary saturated aqueous solutions**. J Res Natl Bur Stand A Phys Chem. 1977. DOI: 10.6028/jres.081A.011.

IIVARI, A., KOIVUSAARI, J., **A RESTful sensor data back-end for the internet of things**. INFOCOMP 2016: The Sixth International Conference on Advanced Communications and Computation (pp. 51-55), 2016. Curran Associates Inc.. International Conference on Advanced Communications and Computation Vol. 6 Disponível em: https://www.thinkmind.org/index.php?view=article&articleid=infocomp_2016_4_20_6_0051. Acesso em 3 de abr. 2023.

KUROSE, J. F., ROSS, K. W., **Computer networking: a top-down approach**, 8th, Pearson, 2020.

LIMA, C. B., **Projeto de placas de circuito impresso - básico**. 2010. Disponível em:
https://professorpetry.com.br/Bases_Dados/Apostilas_Tutoriais/Projeto_PCI_Charles.pdf. Acesso em: 12 abr. 2023.

MOTA, B. D., MATALOTO, B., COUTINHO, C., **Sustainable gardens for smart cities using low-power communications**, 2022 IEEE 28th International Conference on Engineering, Technology and Innovation (ICE/ITMC) & 31st International Association For Management of Technology (IAMOT) Joint Conference, pp.1-7, 2022.

NELSON, P. V., BAROK, J. Jr. W., **Greenhouse management**. Ithaca, N.Y: Northeast Regional Agricultural Engineering Service, 1996.

NXP SEMICONDUCTORS, **UM10204 - I2C-bus specification and user manual**. 7rev. 2021.

Rohm, **Digital 16bit serial output type ambient light sensor IC - BH1750FVI**. 2011. Disponível em:
<https://www.mouser.com/datasheet/2/348/bh1750fvi-e-186247.pdf>. Acesso em: 10 mar. 2023.

ScioSense, **CCS811: Ultra-low power digital gas sensor for monitoring indoor air quality**, rev 3, jul. 2021. Disponível em:
<https://www.sciosense.com/wp-content/uploads/documents/SC-001232-DS-3-CCS811B-Datasheet-Revision-2.pdf>. Acesso em 5 de abr. 2023.

Sensirion, **Datasheet - SHT4x: 4th Gen. High-accuracy, ultra-low-power, 16-bit, relative humidity and temperature Sensor**. 2023. Disponível em:
https://sensirion.com/media/documents/33FD6951/640B22DB/Datasheet_SHT4x.pdf. Acesso em: 5 abr. 2023.

SHAMSHIRI, R. R., JONES, J. W., THORP, K. R., et al. **Review of optimum temperature, humidity, and vapour pressure deficit for microclimate evaluation and control in greenhouse cultivation of tomato: a review**. International Agrophysics, 32(2), 287–302., 2018. DOI: 10.1515/intag-2017-0005.

WENT, F. W. **The effect of temperature on plant growth.** Annual Review of Plant Physiology, p. 347–362., 1953. DOI:10.1146/annurev.pp.04.060153.002023.

ZHANG, D., Du, Q., ZHANG, Z. et al. **Vapour pressure deficit control in relation to water transport and water productivity in greenhouse tomato production during summer.** Sci Rep 7, 2017, 43461. DOI: <https://doi.org/10.1038/srep43461>.