

Sistema Integrado de Monitoramento para Fazendas Hidropônicas: Uma Abordagem IoT para Otimização da Produção Agrícola

Eduardo Fraga de Freitas
Centro universitário SATC
Criciúma, Brazil
eduardofragadefreitas@gmail.com

Sofia Martins Silva
Centro universitário SATC
Criciúma, Brazil
sofia.martinslv@gmail.com

Vagner Rodrigues
Centro universitário SATC
Criciúma, Brazil
vagner.rodrigues@satc.edu.br

Ramon Venson
Centro universitário SATC
Criciúma, Brazil
ramon.venson@gmail.com

Rodrigo Cesar Nunes Maciel
Centro universitário SATC
Criciúma, Brazil
rodrigo.maciel@satc.edu.br

Abstract: Hydroponics is a soil-free cultivation technique, offering a sustainable and economical alternative. This work proposes the development of a prototype monitoring system for hydroponic farms, using sensors connected to a microcontroller to collect environmental and nutritional data, along with embedded software to capture and transmit the collected information, and a web platform for data visualization. The solution is declared functional and scalable, allowing for real-time and remote monitoring. The results indicate that the system has potential, but it was not possible to validate the solution's efficiency in a real production environment.

Keywords: Hydroponics; Internet of Things (IoT); Agricultural Automation; Embedded System; Controlled Cultivation;

Resumo: A hidroponia é uma técnica de cultivo que dispensa o uso do solo, sendo uma alternativa sustentável e econômica. Este trabalho propõe o desenvolvimento de um protótipo de um sistema de monitoramento para fazendas hidropônicas, utilizando sensores conectados a um microcontrolador para coletar dados ambientais e nutricionais, com um software embarcado para capturar e transmitir as informações coletadas e uma plataforma web para visualização dos dados. A solução demonstrou-se funcional e escalável, permitindo monitoramento em tempo real e remoto. Os resultados indicam que o sistema tem potencial, mas não foi possível realizar a validação da eficiência da solução em um ambiente de produção real.

Palavras-chaves: Hidroponia; Internet das Coisas (IoT); Automação Agrícola; Sistema Embarcado; Cultivo Controlado;

I. INTRODUÇÃO

Dispensando o uso do solo para o cultivo, a agricultura hidropônica tem se destacado como uma alternativa sustentável e rentável, especialmente por exigir pouco espaço para a produção, pelo baixo custo de manutenção e a eliminação do uso de agrotóxicos [1]. Nessa forma de cultivo as plantas recebem os nutrientes necessários para o crescimento por meio de um fluxo de solução nutritiva,

composta por água e concentrados contendo macro e micronutrientes [2].

A fim de evitar desequilíbrios no desenvolvimento do plantio, o monitoramento da solução nutritiva é essencial para o sucesso do cultivo, visto que ela estará em circulação no sistema e será submetida a alterações de acordo com o ambiente [3]. O monitoramento acompanha parâmetros fundamentais, como o pH, condutividade elétrica (EC), temperatura e umidade para garantir a qualidade e o desenvolvimento saudável dos alimentos produzidos [4].

Com o avanço da tecnologia, abriu-se uma série de alternativas para facilitar o cultivo hidropônico. De acordo com [5] a hidroponia enfrenta diversos desafios, principalmente tecnológicos, e sistemas hidropônicos devem ser mais desenvolvidos, simplificados e automatizados para torná-los ainda mais acessíveis. O uso de tecnologias de monitoramento por meio de sensores pode tornar possível acompanhar as variáveis essenciais do ambiente e da água para mais eficiência e melhores resultados.

Diante disso, este projeto tem como objetivo desenvolver um protótipo de um sistema de monitoramento para fazendas hidropônicas que permita o acompanhamento em tempo real de parâmetros críticos e possibilite a tomada de decisões proativas para otimizar a produção.

Para alcançar esse objetivo, este trabalho propõe-se a:

- Coletar os dados ambientais e nutricionais com sensores e microcontroladores;
- Desenvolver um software embarcado responsável pela captura e transmissão das informações coletadas;
- Implementar uma plataforma web para visualização e análise dos dados monitorados.

II. FUNDAMENTAÇÃO TEÓRICA

Este capítulo busca estabelecer o arcabouço teórico fundamental para o desenvolvimento do protótipo do sistema de monitoramento proposto. Inicialmente, exploraremos os princípios da hidroponia, detalhando suas vantagens, técnicas de cultivo e a importância do manejo da solução nutritiva. Em seguida, introduziremos o conceito da

Internet das Coisas (IoT), delineando sua arquitetura e protocolos de comunicação. Finalmente, faremos uma revisão das aplicações da IoT na hidroponia, ilustrando como a tecnologia pode otimizar o monitoramento e o controle dos cultivos, pavimentando o caminho para a apresentação do protótipo do sistema desenvolvido neste trabalho.

A. Hidroponia

A hidroponia é uma técnica de cultivo agrícola que não faz o uso do solo, utilizando o meio aquoso para oferecer os nutrientes necessários para o desenvolvimento das plantas. Ela se popularizou na década de 80 sendo utilizada, principalmente, para a produção e comercialização de hortaliças [3] e vem sendo reconhecida pela facilidade de manutenção, pela qualidade dos alimentos produzidos e por ser uma abordagem sustentável de produção de alimentos. Segundo [1] (tradução nossa) “Através do sistema de cultivo hidropônico podemos garantir economia mínima de água de até 70-80% em comparação ao sistema de cultivo baseado no solo.”

Nessa técnica, ao invés de extrair do solo os nutrientes necessários para o crescimento das plantas, as raízes são mergulhadas em uma mistura de água e nutrientes, chamada de solução nutritiva, que é oferecida de forma precisa conforme as necessidades do alimento cultivado [5], minimizando o desperdício.

Na hidroponia, o controle e manutenção dos principais fatores ambientais, incluindo a concentração de nutrientes, pH, condutividade elétrica e temperatura é fundamental para o crescimento saudável das plantas [1]. Por ser desenvolvida em um meio em que as condições ambientais são controladas, como em estufas, os sistemas hidropônicos podem ser uma solução em lugares com condições ambientais desfavoráveis para o plantio tradicional.

Partindo do princípio da ausência de terra e da utilização de água para o cultivo, existem diferentes tipos de sistemas de hidroponia. Eles são categorizados em circulantes, em que a água circula no sistema, ou não-circulantes, que não dependem de sistemas de bombeamento de [5]. Cada método possui suas vantagens, e a escolha de qual deve ser utilizado depende do tipo de planta que será cultivada e do espaço disponível.

1) Solução Nutritiva

Independentemente do método utilizado, um dos aspectos mais importantes na hidroponia é o monitoramento dos nutrientes fornecidos para as plantas, uma vez que ela desempenha o papel que, na agricultura convencional, seria desempenhado pelo solo. No total, existem 17 nutrientes essenciais para o desenvolvimento saudável do plantio, sendo fornecidos através da solução nutritiva em contato com as raízes [1].

A formulação da solução geralmente é feita por meio da diluição de concentrados, que costumam ser divididos em duas partes: Parte A e Parte B ou, alternativamente, divididos em "macro", contendo os macronutrientes (C, H, O, N, P, K, Ca, Mg e S), e "micro", contendo os micronutrientes (B, Cl, Cu, Fe, Mn, Mo, Ni e Zn) [2] ; [4].

A solução nutritiva deve ser balanceada ao longo do seu

uso, a fim de evitar desequilíbrios na absorção dos nutrientes pelas raízes. A sua ausência ou excesso pode comprometer o crescimento saudável e o desenvolvimento adequado das plantas [2]. Nos sistemas hidropônicos, a concentração de nutrientes é monitorada através da condutividade elétrica (EC) que, apesar de não identificar qual nutriente está em falta ou excesso, é recomendada para a reposição de nutrientes [6].

Além da composição, é fundamental o monitoramento da temperatura, pH e da condutividade elétrica (EC) da solução nutritiva. A sua temperatura deve ser mantida entre 20 °C e 25 °C, dado que variações extremas podem comprometer a saúde das raízes e favorecer a proliferação de fungos e bactérias [3]. Já os valores ideais de pH e EC variam conforme com a planta, conforme apresentado na Fig. 1.

Planta	EC (mS/cm)	pH
Manjerição	1.0 - 1.6	5.5 - 6.0
Feijão	2.0 - 4.0	6.0 - 6.5
Banana	1.8 - 2.2	5.5 - 6.5
Brócolis	2.8 - 3.5	6.0 - 6.8
Repolho	2.5 - 3.0	6.5 - 7.0
Abobrinha	1.8 - 2.4	6.0
Pepino	1.7 - 2.0	5.0 - 5.5
Berinjela	2.5 - 3.5	6.0
Alho-poró	1.4 - 1.8	6.5 - 7.0
Alface	1.2 - 1.8	6.0 - 7.0
Acelga	1.5 - 2.0	6.5 - 7.0
Pimentões	0.8 - 1.8	5.5 - 6.0
Salsa	1.8 - 2.0	5.5 - 6.0
Espinafre	1.8 - 2.3	6.0 - 7.0
Morango	1.8 - 2.2	6.0
Sálvia	1.0 - 1.6	5.5 - 6.5
Tomate	2.0 - 4.0	6.0 - 6.5

Fig 1 pH e EC ideais por planta

B. Internet das Coisas (IoT)

A Internet das Coisas (IoT - Internet of Things) é uma tecnologia em rápido crescimento e disseminação. Seu objetivo principal é fazer com que dispositivos colem, troquem e processem dados automaticamente, com pouca ou nenhuma intervenção humana [7]. Esses dispositivos podem incluir sensores, dispositivos eletrônicos, eletrodomésticos, veículos, máquinas industriais, entre outros.

O termo surgiu inicialmente para se referir a objetos conectados, capazes de se comunicar entre si e ser identificados utilizando identificação por radiofrequência, ou RFID (do inglês "Radio-Frequency IDentification"), sendo amplamente adotado para identificação em setores como o varejo, logística e produção farmacêutica. Com o tempo, IoT passou a ser associado a outras tecnologias, como sensores, atuadores, dispositivos móveis e GPS [8].

Um dos aspectos mais importantes para o funcionamento dos dispositivos IoT é a comunicação. De acordo com [9] a arquitetura mais comum possui quatro camadas. Cada camada desempenha funções que garantem a comunicação dos dispositivos, sendo elas:

- **Camada de aplicação:** Responsável pela interação com o usuário final;
- **Camada de processamento:** Processa, armazena e

analisa os dados recebidos;

- **Camada de rede:** Transfere os dados coletados da camada de percepção para a camada de processamento;
- **Camada de percepção:** Responsável pelos sensores e outros dispositivos físicos para coleta de dados.

C. Aplicações da IoT na Hidroponia

De acordo com Sousa et al., (2024), uma das principais limitações para o desenvolvimento da hidroponia doméstica, atualmente, é a tecnológica. A manutenção do pH, condutividade elétrica, concentração dos nutrientes precisa ser cuidadosamente feita por alguém com alto conhecimento técnico [1].

Com o avanço da tecnologia, sistemas de monitoramento tornaram-se aliados na prática da hidroponia. Por meio de sensores eletrônicos é possível acompanhar em tempo real as variáveis cruciais do cultivo. Essa coleta de dados possibilita ajustes precisos na solução nutritiva, assegurando a qualidade do plantio, o crescimento saudável das plantas, otimizando a gestão do tempo de manejo e reduzindo perdas.

Em [10], utilizaram um microcontrolador ESP32 com sensores de umidade e temperatura para desenvolver um sistema que é capaz de monitorar as variáveis dos sensores e controlar automaticamente válvulas que abrem/fecham para deixar a água circular pelas plantas. O sistema também foi integrado com um chatbot no Telegram para a visualização dos dados e controle das válvulas.

Em outro projeto, [11] desenvolveram um sistema chamado iPONICS utilizando Wireless Sensor Networks (WSNs), ou Redes de Sensores Sem Fio, que tem como proposta facilitar o monitoramento remoto de fazendas hidropônicas com um baixo custo. Além de monitorar as principais variáveis da solução nutritiva e do ambiente e controlar automaticamente as bombas de fluxo da água, também foi desenvolvido alarmes de segurança para detectar entradas não autorizadas.

No iPONICS, a frequência da coleta dos dados difere para cada parâmetro de acordo com sua taxa de variação, como mostra o Qd. 1.

Comparativamente, o sistema proposto neste trabalho difere dos projetos por não incorporar atuadores para controle automático, focando exclusivamente no monitoramento. No entanto, um diferencial significativo deste projeto reside na sua integração flexível, que oferece múltiplos pontos de acesso aos dados: tanto via HTTP, por meio do servidor ou aplicação web, quanto via protocolo MQTT. Além disso, oferece tanto a possibilidade de monitoramento em tempo real quanto a visualização histórica dos dados. Esses diferenciais garantem que os usuários possam acessar as informações cruciais de forma adaptável às suas necessidades e cenários

D. Comunicação de Dados em Sistemas IoT

A comunicação de dados é um elemento essencial em sistemas de Internet das Coisas (IoT), pois permite a troca estruturada e confiável de informações entre sensores, microcontroladores e servidores. Essa comunicação depende

de protocolos de rede e de aplicação, que atuam em diferentes camadas da pilha de comunicação e definem as regras para transmissão, encaminhamento e interpretação dos dados [7].

QUADRO 1
REQUISITOS DE COLETA DE DADOS E ALARMES

Requisito	Período de amostragem	Alarme/aviso
Monitorar qualidade da água (pH, temperatura, condutividade elétrica, oxigenação)	diariamente	Aviso dependendo da planta
Monitorar temperatura do ambiente	8 segundos, registra de hora em hora	Alerta (possível fogo) Aviso dependendo do tipo de cultivo
Monitorar umidade do ambiente	de hora em hora	Aviso dependendo do tipo de cultivo
Detectar entrada não autorizada	orientado a evento (interrupção)	Alerta
Colocar nós para dormir para conservar energia	N/A	N/A
Fornecer interface gráfica de usuário simples (painel) para vários sites	N/A	N/A

Neste projeto, foi utilizada a tecnologia Wi-Fi (IEEE 802.11) como meio de comunicação sem fio, devido à sua boa taxa de transmissão, compatibilidade com redes IP e facilidade de integração com a infraestrutura de rede existente [12]; [13]). Embora demande fornecimento contínuo de energia, sua adoção é adequada em contextos controlados, como laboratórios ou estufas, onde essas condições estão disponíveis.

Na camada de aplicação, foram adotados dois protocolos complementares: HTTP e MQTT. O Hypertext Transfer Protocol (HTTP) é amplamente utilizado em aplicações web, baseado em um modelo de requisição/resposta em que o cliente acessa recursos via URIs [14]. Ele é empregado neste trabalho na interface web para visualização dos dados, aproveitando sua simplicidade e ampla compatibilidade com navegadores e APIs REST.

O Message Queuing Telemetry Transport (MQTT), por sua vez, é um protocolo leve e eficiente, desenvolvido para comunicação máquina a máquina (M2M). Utiliza o modelo

de publicação/assinatura, onde dispositivos publicam dados em um broker, que os entrega aos assinantes dos respectivos tópicos. Essa abordagem reduz o tráfego de rede e o consumo energético, sendo ideal para o envio contínuo de dados de sensores. Cada mensagem transmitida é associada a um identificador chamado tópico, que organiza semanticamente a informação [14].

A integração entre Wi-Fi, HTTP e MQTT garante ao sistema desenvolvido conectividade local eficiente, comunicação em tempo real e compatibilidade com plataformas web, atendendo aos requisitos de desempenho e usabilidade da solução proposta.

III. PROCEDIMENTO EXPERIMENTAL

Com base nos conceitos de hidroponia e nas tecnologias de Internet das Coisas (IoT), incluindo os protocolos de comunicação previamente discutidos, desenvolveu-se este trabalho, que consistiu na criação de um protótipo de sistema integrado de monitoramento para fazendas hidropônicas. A seguir, são apresentados os materiais utilizados e os métodos adotados na construção do sistema, abrangendo tanto os componentes de hardware quanto às soluções de software.

A. Materiais

Os materiais contemplados são divididos em materiais físicos e recursos computacionais que viabilizam a criação do circuito eletrônico, as aplicações de software e suas interações.

1) Materiais Físicos

Para a montagem do circuito eletrônico, foram utilizados os seguintes componentes:

1.1 Sistema-em-Chip (SoC) ESP32

O ESP32 é o microcontrolador principal do sistema, responsável pelo processamento dos dados dos sensores, comunicação via Wi-Fi e controle do sistema embarcado. Sua compatibilidade com interfaces como GPIO (General Purpose Input/Output), I2C (Inter-Integrated Circuit) e ADC (Analog-to-Digital Converter) viabiliza a integração dos sensores utilizados. Neste projeto, foi adotado o módulo ESP-WROOM-32, que se destaca por integrar o chip ESP32 com conectividade Wi-Fi e Bluetooth, memória flash e outros componentes essenciais, otimizando o desenvolvimento de aplicações IoT compactas e eficientes.

1.2 Sensor de Umidade e Temperatura Ambiente DHT22

O DHT22 mede a umidade relativa do ar e a temperatura ambiente próximas à plantação. Esses dados são fundamentais para monitorar as condições ambientais que afetam o crescimento das plantas e para ajustes no sistema de controle.

1.3 Sensor de Temperatura da Água DS18B20

O sensor DS18B20 mede a temperatura da solução nutritiva no sistema hidropônico. Esse parâmetro é crucial, pois impacta a absorção de nutrientes pelas plantas e seu desenvolvimento saudável.

1.4 Sensor de Condutividade Elétrica (EC) da Água

9SSI7

O sensor 9SSI7 mede a condutividade elétrica da água, relacionada à concentração de sais minerais na solução nutritiva. Controlar essa condutividade é essencial para garantir a disponibilidade adequada de nutrientes às plantas. O sensor fornece saída de 0 a 2,3 VDC em escala ppm, requerendo ajuste de tensão para compatibilidade com o ESP32.

1.5 Sensor de pH da Água PH-4502C

O sensor de pH monitora a acidez ou alcalinidade da solução nutritiva, parâmetro crucial para a absorção adequada dos nutrientes pelas plantas. Manter o pH na faixa ideal assegura um ambiente favorável ao crescimento e previne problemas nas culturas.

1.6 Real Time Clock (RTC) DS3231

O módulo RTC DS3231 fornece marcação precisa do tempo, permitindo registrar com exatidão os horários das leituras e eventos. Isso facilita o monitoramento temporal e a análise histórica dos dados da fazenda hidropônica.

1.7 Raspberry Pi Model 4

O Raspberry Pi Model 4 é a máquina hospedeira da infraestrutura do projeto, oferecendo poder computacional suficiente para rodar todos os serviços e conectores USB para periféricos, como HDs externos.

1.8 Outros

Foram usados jumpers, protoboards e sondas para montar o circuito, com auxílio de um multímetro para verificar as condições elétricas durante a montagem. Um HD externo de 1 TB foi utilizado para armazenar os dados dos bancos de dados.

2) Materiais de Desenvolvimento Embarcado

O desenvolvimento do firmware embarcado para o módulo ESP-WROOM-32 foi realizado utilizando a extensão PlatformIO, integrada ao ambiente de desenvolvimento Visual Studio Code (VS Code). Essa combinação oferece um ambiente robusto, multiplataforma e altamente configurável para desenvolvimento em sistemas embarcados.

O PlatformIO permite a gestão automatizada de bibliotecas, ambientes de compilação e dependências, além de suportar múltiplos frameworks e placas de desenvolvimento. Para este projeto, foi adotado o framework Arduino, amplamente utilizado no ecossistema ESP32, por sua simplicidade, vasta comunidade e compatibilidade com bibliotecas existentes. O desenvolvimento foi realizado na linguagem C++, padrão no framework Arduino.

As bibliotecas utilizadas foram selecionadas para atender às necessidades específicas de comunicação, aquisição de dados e processamento dos sensores.

3) Materiais de Desenvolvimento Web

O desenvolvimento web do sistema foi dividido em três componentes principais, alinhados às necessidades específicas da arquitetura da solução: um servidor web embarcado no microcontrolador, uma aplicação web full-stack e um serviço back-end auxiliar para persistência

dos dados provenientes do broker MQTT.

O servidor web embarcado no ESP32 foi implementado utilizando a biblioteca nativa WebServer, disponível no framework Arduino para ESP32. Este servidor tem a finalidade de disponibilizar uma interface de acesso local via protocolo HTTP, permitindo que usuários realizem consultas rápidas, comandos de controle e configuração diretamente no dispositivo. As páginas web são embutidas diretamente no firmware, por meio de strings contendo código HTML e eventuais trechos de JavaScript e CSS, dispensando o uso de sistemas de arquivos embarcados como SPIFFS ou LittleFS. A biblioteca WebServer permite o roteamento de requisições, gerenciamento de respostas e controle de métodos HTTP básicos (GET, POST, etc.).

A aplicação web principal foi desenvolvida com uma abordagem full-stack utilizando a linguagem TypeScript. Para comunicação com as APIs back-end, foi utilizada a biblioteca Axios, que facilita a realização de requisições HTTP e o tratamento de respostas assíncronas.

O Back-end foi desenvolvido com NestJS, framework Node.js fortemente baseado em TypeScript, que oferece uma arquitetura robusta e escalável. Ele é responsável por implementar as regras de negócio, expor APIs REST, realizar autenticação, controle de acesso e integração com serviços externos.

Além disso, foi desenvolvido um serviço auxiliar em Python, responsável pela captura dos dados publicados via MQTT e sua persistência no banco de dados NoSQL em uma coleção de dados unificada. Este serviço utiliza:

- **Paho MQTT:** Biblioteca cliente MQTT, utilizada para a assinatura dos tópicos e recepção dos dados transmitidos pelos dispositivos IoT;
- **PyMongo:** Driver oficial do MongoDB para Python, que permite a inserção, consulta e manipulação dos documentos no banco de dados.

Esse serviço desempenha um papel fundamental na ponte entre a comunicação assíncrona do protocolo MQTT e o armazenamento dos dados, garantindo que as informações estejam disponíveis para processamento e análise posterior.

Cada um desses componentes foi selecionado considerando critérios de escalabilidade, robustez, modularidade e aderência às melhores práticas de desenvolvimento de sistemas web modernos e aplicações IoT.

4) Materiais de Infraestrutura

A infraestrutura da solução foi provisionada utilizando um Raspberry Pi 4 Model B como máquina hospedeira, suportando toda a stack necessária para operação do sistema. O ambiente conta com um disco rígido externo de 1TB.

A orquestração dos serviços é realizada com Docker Compose, permitindo automação do deploy, gerenciamento de dependências, configuração de redes internas e mapeamento de volumes persistentes.

1.1 Bancos de Dados

O banco de dados adotado para persistência dos dados sensoriais é o MongoDB, utilizando uma imagem da versão

4.4.18 otimizada para a arquitetura ARM (arm64v8/mongo), compatível com o hardware do Raspberry Pi.

A escolha do MongoDB se deve à sua natureza NoSQL orientada a documentos, que oferece flexibilidade na modelagem dos dados, boa performance em leituras e escrita, além de alinhamento natural ao formato dos dados provenientes dos sensores IoT (em JSON).

1.2 Broker MQTT

A comunicação assíncrona entre os dispositivos IoT e os serviços back-end é realizada via protocolo MQTT, utilizando como broker uma instância do mochi-mqtt, desenvolvido em Go. O mochi-mqtt foi escolhido por ser leve, eficiente e adequado a ambientes embarcados. O container do broker expõe as portas:

- **1883** - Porta padrão MQTT TCP;
- **1880** - Interface HTTP para administração e monitoramento (se habilitada via configuração).

A configuração do broker é gerenciada por meio de um arquivo de template (mqtt-broker-config.template.yml) que define parâmetros operacionais como tópicos, autenticação (se aplicável) e políticas de retenção de mensagens.

1.3 Orquestração

O provisionamento e gerenciamento dos serviços são realizados via Docker Compose, que define e interliga os containers do Qd. 3.

O serviço mqtt-to-mongodb-service é construído a partir do diretório “../hydroponics-mqtt-to-mongodb-service”, definido no Compose como “build.context”, permitindo integração direta com o pipeline de desenvolvimento.

B. Métodos

Os métodos adotados neste trabalho descrevem as etapas de execução do projeto. A Fig. 12 apresenta o diagrama do fluxo de execução do trabalho, que ilustra como os dados percorrem os diferentes componentes do sistema, desde sua aquisição até sua disponibilização para os usuários finais.

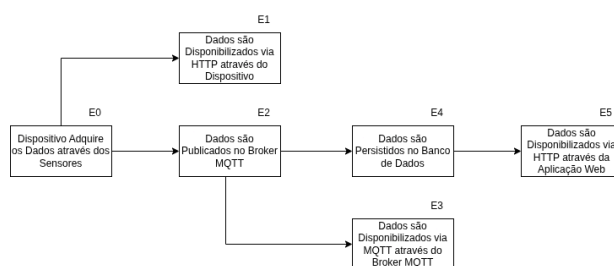


Fig 2 Diagrama do fluxo de execução do trabalho

O fluxo é composto pelas seguintes etapas:

- **E0** - Aquisição dos dados: O dispositivo IoT realiza a leitura dos dados provenientes dos sensores (temperatura, umidade, condutividade, entre outros). Essa etapa é executada de forma contínua pelo firmware embarcado no ESP32.
- **E1** - Disponibilização via HTTP no dispositivo: Como uma alternativa direta de acesso aos dados, o dispositivo permite que os dados sensoriais sejam consultados diretamente via requisições HTTP, por meio do servidor web embarcado no próprio ESP32.

- **E2** - Publicação no Broker MQTT: Os dados adquiridos são publicados no broker MQTT, operando no modelo publish/subscribe, que permite a comunicação assíncrona entre dispositivos e serviços interessados nos dados.
- **E3** - Disponibilização via MQTT: Qualquer cliente MQTT pode se inscrever nos tópicos definidos para receber, em tempo real, os dados disponibilizados pelo dispositivo. Essa abordagem favorece integrações dinâmicas e comunicação eficiente entre sistemas.
- **E4** - Persistência no banco de dados: Paralelamente, um serviço back-end, implementado em Python, consome as mensagens publicadas no broker MQTT e realiza a persistência dos dados no banco de dados MongoDB, assegurando a armazenagem histórica e a possibilidade de análises posteriores.
- **E5** - Disponibilização via aplicação web: Os dados persistidos podem ser consultados por meio da aplicação web, que disponibiliza uma interface HTTP para os usuários. Essa interface permite tanto o consumo dos dados de forma estruturada quanto a visualização dos dados históricos e em tempo real.

O sistema, portanto, oferece múltiplos meios de acesso aos dados, tanto diretamente via HTTP (E1 e E5) quanto via protocolo MQTT (E3). Essa diversidade de formas de disponibilização busca garantir maior flexibilidade, permitindo atender diferentes cenários e necessidades, seja para consumo humano (através da interface web) ou para integrações com outros sistemas e dispositivos (via MQTT ou API HTTP).

Cada uma dessas abordagens possui vantagens e limitações. A comunicação via MQTT é altamente eficiente e adequada para transmissão de dados em tempo real, enquanto o acesso via HTTP é mais convencional e acessível para usuários finais e sistemas que não implementam clientes MQTT.

IV. DESENVOLVIMENTO

Este capítulo busca descrever cada etapa de desenvolvimento do sistema proposto. Inicialmente, será abordado o desenvolvimento do circuito eletrônico, detalhando sua estrutura, leitura e normalização dos sinais dos sensores, conversão e publicação dos dados e gestão de falhas. Por fim, entraremos em detalhe quanto ao desenvolvimento da aplicação web, fazendo uma revisão dos procedimentos utilizados pelo front-end e back-end.

A. Desenvolvimento do Circuito Eletrônico

O circuito eletrônico foi desenvolvido considerando uma alimentação de 5V através da porta micro USB do ESP32, ele é representado através da Fig. 13. Os materiais utilizados foram aqueles citados na Seção 3.1.1. O circuito contemplou sensores analógicos, digitais e seguindo o padrão 1-Wire. Por isso, as principais saídas e entradas utilizadas foram as GPIOs e ADCs. Considerando-se a tensão de saída do módulo PH-4502C de 0 a 5V e a limitação das GPIOs do ESP32 de manter a tensão de entrada em níveis de 0V e 3.3V (não devendo exceder $3.3V + 0.3V$), inseriu-se um trimpot de 10 k Ω entre a saída analógica do sensor e a entrada ADC do ESP32, conforme a referência R1 na Fig.

12.

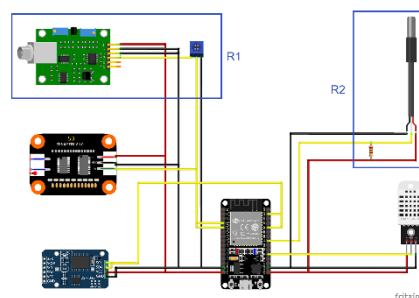


Fig 3 Desenho do circuito eletrônico

Essa solução possibilita o ajuste do ponto de offset, compensando variações da tensão de referência e garantindo maior precisão de leitura e adequação da tensão de saída para a escala de 0V a 3.3V recomendada para o ESP32.

Também, inseriu-se um resistor de série de 220 Ω na linha de dados do sensor de temperatura da água DS18B20, atuando como limitador de corrente durante picos transitórios e protegendo o pino GPIO do ESP32 contra sobrecorrentes, conforme a referência R2 na Fig. 13

1) Desenvolvimento do Firmware Embarcado

O desenvolvimento do firmware embarcado visou implementar as funcionalidades de aquisição de dados, processamento e disponibilização local via HTTP e transmissão das informações provenientes dos sensores para o sistema remoto, por meio do protocolo MQTT.

O firmware foi desenvolvido na linguagem C++, utilizando VSCode, conforme citado na Seção 3.1.2. As principais etapas desse desenvolvimento serão exploradas nas próximas seções.

1.1 Leitura dos Sinais Analógicos e Digitais

Os sensores de temperatura (DS18B20), umidade e temperatura ambiente (DHT22) possuem interfaces digitais, cujos dados são lidos diretamente por protocolos específicos (1-Wire e barramento digital simples, respectivamente).

Já os sensores de pH (PH-4502C) e condutividade elétrica (9SSI7) possuem saídas analógicas, cuja leitura é realizada pelas entradas ADC do ESP32, que convertem a tensão de entrada, na faixa de 0 a 3.3V, em valores digitais de 0 a 4095, graças à resolução de 12 bits do conversor analógico-digital.

1.2 Conversão dos Dados Para Unidades Físicas

Após a aquisição, o firmware realiza a conversão dos sinais analógicos em suas respectivas unidades físicas, quando aplicável, por meio de interpolação linear baseada nas especificações técnicas de cada sensor. Por exemplo, no sensor de pH, a faixa de leitura de 0 a 4095 bits corresponde ao intervalo de 14 a 0 unidades de pH, enquanto no sensor de condutividade elétrica (EC), a mesma faixa representa valores de 0 a 1000 ppm.

1.3 Conectividade com a Rede Wi-Fi

Para garantir a disponibilização dos dados para consumo, um módulo de conexão Wi-Fi foi desenvolvido. Sua responsabilidade é estabelecer a conexão do circuito

com a Internet, com métodos de retry que garantem que mesmo diante de falhas do sistema ou perda de energia, o sistema físico volte a conectar-se com a rede sempre que perder a conexão. Sem este módulo a disponibilização dos dados é inviabilizada. Também, é através dele que o módulo Web Server é instanciado e permite que clientes HTTP possam se comunicar diretamente com o circuito, conforme proposto na etapa E1 da Seção 3.2.

Publicação dos Dados Via MQTT

O firmware, por sua vez, realiza a publicação periódica dos dados no broker MQTT configurado, utilizando tópicos hierárquicos que seguem a estrutura “hydroponics-esp32-controller/measurements/water/temperature”. Por padrão, a publicação ocorre em intervalos de 5 minutos, mas é configurável no firmware.



Fig 4 Estrutura do dado publicado em um tópico MQTT

Após a publicação, os dados são disponibilizados para consumo conforme proposto na etapa E3 da Seção 3.2, dentro do padrão estabelecido na Fig. 14.

1.4 Gestão de Falhas e Integridade dos Dados

Foram implementadas rotinas de verificação da integridade dos dados, gestão de desconexões de rede e reinicialização automática em caso de falhas críticas, visando aumentar a robustez do sistema no ambiente agrícola.

B. Desenvolvimento da Aplicação Web

A aplicação web desenvolvida neste trabalho tem como objetivo fornecer uma interface gráfica intuitiva e responsiva para visualização, análise e acompanhamento dos dados coletados pelo sistema de monitoramento da fazenda hidropônica. Ela é responsável por garantir a disponibilidade dos dados conforme proposto na etapa E5 da Seção 3.2. Sua arquitetura foi concebida seguindo o princípio da separação de responsabilidades, adotando uma estrutura composta por

front-end e back-end de maneira desacoplada. O front-end foi desenvolvido utilizando o framework React, explorando uma arquitetura baseada em componentes e no paradigma de Single Page Application (SPA). Para a construção da interface, foram utilizados os componentes da biblioteca Shaden UI, fundamentada nos primitivos do Radix UI, combinados com a biblioteca de estilização utilitária TailwindCSS. Essa combinação permite a construção de interfaces acessíveis, escaláveis e com alto grau de consistência visual.

As Figs. 15 e 16 mostram a Seção de monitoramento das condições da água no front-end da aplicação web desenvolvida.

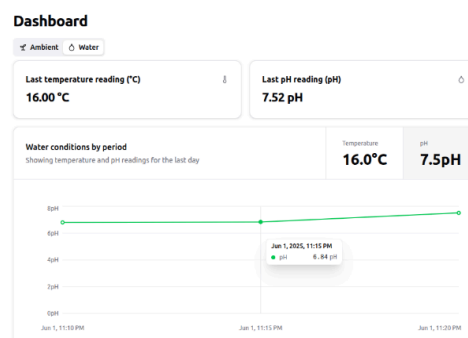


Fig 5 Front-end da aplicação web mostrando as condições da água

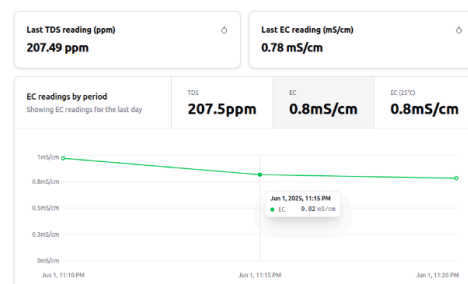


Fig 6 Front-end da aplicação web mostrando as medições de EC

Do qual, encontram-se as seguintes funcionalidades:

- Dashboard em tempo real, onde são exibidos os valores atuais dos sensores, incluindo temperatura ambiente, umidade, temperatura da água, pH e condutividade elétrica (EC);
- Consulta histórica de medições por período de um dia;
- Representação visual dos dados por meio de gráficos e indicadores, utilizando a biblioteca Recharts para geração dos gráficos de linha.

A comunicação com o back-end é realizada por meio de requisições HTTP, utilizando a biblioteca Axios para a realização de chamadas assíncronas, em conjunto com React Query, com suas configurações padrões, para o gerenciamento eficiente de estado remoto, incluindo cache, sincronização e controle de atualizações automáticas dos dados.

O back-end foi implementado utilizando o framework NestJS, que combina conceitos de desenvolvimento modular, orientação a objetos e injeção de dependências. Dentre as suas principais funcionalidades, destacam-se:

- Disponibilização de uma API RESTful, que permite ao front-end consumir os dados provenientes dos sensores;
- Definição de endpoints estruturados que seguem o padrão dos tópicos MQTT utilizados na comunicação embarcada. Por exemplo, o endpoint responsável pela consulta da temperatura da água é acessível por meio da URL “http://localhost:3333/measurements/water/temperature”. Após a consulta, o servidor deverá retornar uma resposta conforme a Fig. 17;

- Implementação de filtros temporais por meio de parâmetros de query, como “from” e “to” (no formato Date), permitindo ao usuário obter dados referentes a períodos específicos;
- Suporte à paginação dos resultados, mediante os parâmetros “page” (número da página) e “pageSize” (quantidade de registros por página), assegurando desempenho adequado na manipulação de grandes volumes de dados;
- Realização de testes unitários e de integração utilizando o framework Vitest, promovendo a confiabilidade e robustez do sistema.



Fig 7 Retorno da consulta da temperatura da água na API RESTful

Os endpoints foram organizados de forma coerente com a hierarquia dos tópicos MQTT, adaptando-os para o contexto HTTP. O Qd. 2 lista os principais caminhos implementados.

QUADRO 2
PRINCIPAIS ENDPOINTS DA API RESTFUL

Endpoint	Descrição
/measurements/ambient/temperature	Consulta da temperatura ambiente
/measurements/ambient/humidity	Consulta da umidade relativa do ambiente
/measurements/water/temperature	Consulta da temperatura da água
/measurements/water/ec	Consulta da condutividade elétrica (EC)
/measurements/water/ph	Consulta do pH da solução nutritiva

C. Configuração da Infraestrutura e Integração dos Sistemas

A gestão e orquestração dos serviços foram realizadas utilizando Docker, tecnologia que permite a criação de ambientes isolados e portáteis, facilitando a replicação da

infraestrutura em diferentes ambientes, como desenvolvimento, testes e produção. Cada componente foi encapsulado em containers independentes, conforme demonstrado no Qd. 3.

QUADRO 3
RELAÇÃO DE CONTAINERS E SUAS IMAGENS DOCKER.

Serviço	Descrição	Imagem Docker
broker-mqtt	Broker MQTT (mochi-mqtt)	mochimqtt/server
mongodb	Banco de dados noSQL	arm64v8/mongo:4.4.18
mqtt-to-mongo-odb-service	Serviço de persistência de dados	build.context (imagem própria)

O Docker Compose é o agente que gerencia as redes internas para isolamento dos serviços, garantindo que as comunicações ocorram de forma segura e eficiente, além de assegurar reinicialização automática dos containers (restart: unless-stopped) em caso de falhas ou reinicialização do host.

Além dos serviços mencionados, a aplicação web (front-end em React e back-end em NestJS) também é hospedada no mesmo ambiente, rodando independentemente. Dessa forma, todo o ecossistema opera localmente, sem dependência de serviços externos, garantindo autonomia e resiliência da solução.

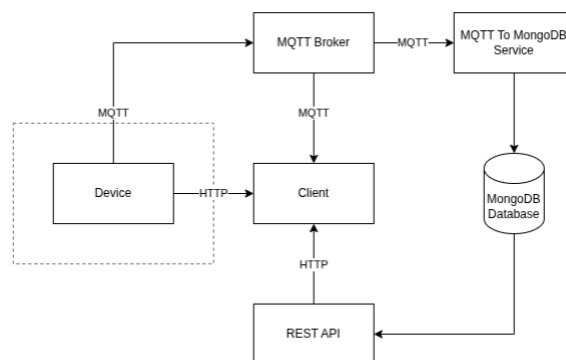


Fig 8 Diagrama da arquitetura de integração dos sistemas

Por fim, a Fig. 18 traz a arquitetura de integração dos sistemas, onde é possível observar a comunicação entre os dispositivos IoT, o broker MQTT, o serviço de persistência de dados, o banco de dados e a aplicação web. Esta arquitetura foi projetada visando garantir a interoperabilidade entre os componentes, bem como a escalabilidade da solução, mesmo sendo executada em um ambiente de recursos limitados, como o Raspberry Pi 4.

V. RESULTADOS E DISCUSSÕES

O sistema desenvolvido com base na Seção anterior atendeu aos requisitos propostos para o monitoramento dos parâmetros físico-químicos da solução nutritiva em cultivo

hidropônico, possibilitando a coleta, visualização e acompanhamento remoto das variáveis ambientais e da água.

Para os testes experimentais, foi utilizado um método de hidroponia não-circulante, em que as plantas ficam suspensas em um reservatório com solução nutritiva e suas raízes ficam submersas, absorvendo os nutrientes. Dessa forma, foi possível testar a eficácia e funcionalidades do sistema através do monitoramento da solução nutritiva do recipiente.

Foi possível utilizar o sistema como apoio na regulação inicial da quantidade da solução nutritiva e no controle dos níveis de pH e condutividade elétrica (EC). Os testes foram feitos durante o período de cinco dias, em que parâmetros foram monitorados através dos sensores mergulhados na solução nutritiva e consultados na plataforma web, de forma que eles permanecessem dentro das faixas recomendadas para o cultivo.



Fig 9 Recipiente com muda, solução nutritiva e sondas

A Fig. 19 apresenta o recipiente contendo a solução nutritiva que está sendo monitorada. Para os testes foi utilizada uma muda de alface americana que, de acordo com Kannan et al. (2022), deve ser cultivada em um EC de 1.2 a 1.8 mS/cm e um pH de 6.0 a 7.0.

A Fig. 20 apresenta a maquete desenvolvida, com o recipiente e o circuito devidamente integrados. Assim, viabilizando o fluxo proposto na Seção 3.2.

Após a implantação do sistema, os dados são imediatamente disponibilizados através dos seguintes meios:

- O web server embarcado no ESP32. Que permite a visualização dos dados on-site, caso haja indisponibilidade na aplicação web;
- O front-end da aplicação web, conforme exemplificado nas Figs. 15 e 16 da Seção 4.2.

Apesar dos resultados funcionais, foram identificadas limitações associadas à calibração dos sensores utilizados. O sensor de condutividade elétrica (EC) apresentou uma variação média de aproximadamente 0,3 a 0,5 na leitura dos seus valores, o que impacta diretamente na precisão do monitoramento da concentração de sais dissolvidos na solução. No caso do sensor de pH, embora a oscilação observada tenha sido de menor magnitude, também foi verificada a necessidade de aprimoramento no processo de calibração, a fim de garantir maior acurácia nas medições.

Tais imprecisões não inviabilizaram o funcionamento geral do sistema, mas evidenciam a importância de procedimentos rigorosos de calibração, bem como a necessidade de considerar sensores de maior precisão para aplicações em ambientes produtivos.

Além disso, destaca-se que o período de apenas cinco

dias e a limitação de um único pé de alface para a realização dos testes restringem a abrangência dos resultados e impede conclusões mais amplas sobre o desempenho do sistema em diferentes condições ou em cultivos de maior escala.

Ainda assim, o sistema demonstrou-se eficaz como ferramenta de apoio para acompanhamento dos parâmetros essenciais, contribuindo para ajustes manuais na reposição da solução nutritiva e no equilíbrio das condições da água.

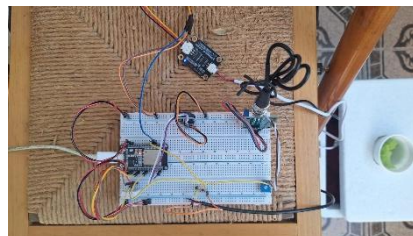


Fig 10 Maquete com circuito e recipientes integrados

VI. CONSIDERAÇÕES FINAIS

O desenvolvimento do protótipo do sistema foi concluído com sucesso em termos de implementação e integração de hardware e software. No entanto, a validação da eficiência em um ambiente de produção real não foi possível devido ao tempo restrito, o que impediu a coleta de dados quantitativos e qualitativos suficientes para aferir métricas de performance operacional, ganhos de otimização de processos, redução de custos ou aumento da produtividade agrícola. Tais análises são cruciais para comprovar os benefícios econômicos e operacionais da abordagem IoT no setor hidropônico.

Apesar dessa limitação, os resultados obtidos em testes laboratoriais demonstraram que a arquitetura proposta é funcional, escalável e possui potencial para atender às demandas do monitoramento agrícola. Por isso, recomenda-se que estudos futuros considerem uma implantação prolongada da solução em contextos reais de produção. Isso permitiria a avaliação de indicadores como confiabilidade, disponibilidade, impacto na gestão dos cultivos e retorno sobre investimento (ROI).

Adicionalmente, o projeto contribui para a comunidade de hardware/software livre ao utilizar tecnologias de código aberto, como ESP32, PlatformIO, Arduino, Docker, MongoDB, mochi-mqtt, Paho MQTT, PyMongo, React e NestJS. Este trabalho serve como estudo de caso e base para inovações futuras, beneficiando-se da robustez e flexibilidade dessas ferramentas. A disponibilização da documentação e do código-fonte [\[15\]\[16\]\[17\]\[18\]\[19\]](#) pode fomentar o aprendizado e a replicação por outros desenvolvedores e pesquisadores, promovendo a disseminação do conhecimento e o avanço coletivo em soluções de monitoramento IoT para agricultura hidropônica.

REFERÊNCIAS

- [1] KANNAN, M. et al., “Hydroponic farming – A state of art for the future agriculture,” *Materials Today: Proceedings*, v. 68, p. 2163–2166, set. 2022.
- [2] JONES, J. Benton, *Hydroponics: A Practical Guide for the Soilless Grower*, [S.l.]: CRC Press, 2004.
- [3] CARRIJO, Osmar Alves; MAKISHIMA, Nozomu, “Princípios de hidroponia,” *Embrapa Hortaliças - Circular Técnica*, nov. 2000, disponível em: <http://www.infoteca.cnptia.embrapa.br/handle/doc/769981>, acesso em: 7 mar. 2025.
- [4] EGÍDIO, Neto Bezerra; LEVY, Barreto Paes, “AS TÉCNICAS DE HIDROPONIA,” *Anais da Academia Pernambucana de Ciência Agronômica*, v. 8, n. 9, p. 107–137, 2011, disponível em: <https://www.journals.ufrpe.br/index.php/apca/article/view/152>, acesso em: 4 maio 2025.
- [5] SOUSA, Rui de et al., “Challenges and Solutions for Sustainable Food Systems: The Potential of Home Hydroponics,” *Sustainability*, v. 16, p. 817, 17 jan. 2024.
- [6] NETA, Helena Maria de Moraes, “Cultivo de alface em sistemas hidropônicos e concentrações de solução nutritiva,” *Universidade Federal Rural do Semi-Árido*, 24 fev. 2021, disponível em: <https://repositorio.ufersa.edu.br/handle/prefix/7363>, acesso em: 20 maio 2025.
- [7] BIBI, Nawab et al., “A Survey of Application Layer Protocols of Internet of Things,” dez. 2021, v. 21, p. 301–311.
- [8] SHAMMAR, Elham Ali; ZAHARY, Ammar Thabit, “The Internet of Things (IoT): A survey of techniques, operating systems, and trends,” *Library Hi Tech*, v. 38, n. 1, p. 5–66, 6 abr. 2020.
- [9] THILAKARATHNE, Navod et al., “Internet of Things (IoT) Security: Status, Challenges and Countermeasures,” *Int. J. Advanced Networking and Applications*, v. 14, n. 03, p. 5444–5454, 16 nov. 2022.
- [10] REDDY, Pravalika; SOHEL, Mohammed; NAIK, Santosh Madeva, “An IoT Based Smart Hydroponics - Plant Automation BOT,” 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems, ICETIS 2024, p. 1203–1207, 2024.
- [11] TATAS, K. et al., “IPONICS: IoT Monitoring and Control for Hydroponics,” 10th International Conference on Modern Circuits and Systems Technologies, MOCAS 2021, 5 jul. 2021.
- [12] SURESH, P. et al., “A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment,” 2014 International Conference on Science Engineering and Management Research, ICSEMR 2014, 17 fev. 2014.
- [13] MARCOS AMOROSO, Marcelo et al., “Wireless Network Technologies for Smart Homes: A Technical and Economic Analysis,” *IEEE Latin America Transactions*, v. 19, n. 5, p. 717–725, 1 maio 2021.
- [14] NAIK, Nitin, “Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP,” 2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings, 26 out. 2017.
- [15] FREITAS, Eduardo, “hydroponics-esp32-controller,” GitHub, disponível em: <https://github.com/duftrss/hydroponics-esp32-controller>, acesso em: 9 out. 2025.
- [16] FREITAS, Eduardo, “hydroponics-api,” GitHub, disponível em: <https://github.com/duftrss/hydroponics-api>, acesso em: 9 out. 2025.
- [17] FREITAS, Eduardo, “hydroponics-app,” GitHub, disponível em: <https://github.com/duftrss/hydroponics-app>, acesso em: 9 out. 2025.
- [18] FREITAS, Eduardo, “hydroponics-mqtt-to-mongodb-service,” GitHub, disponível em: <https://github.com/duftrss/hydroponics-mqtt-to-mongodb-service>, acesso em: 9 out. 2025.
- [19] FREITAS, Eduardo, “hydroponics-infrastructure,” GitHub, disponível em: <https://github.com/duftrss/hydroponics-infrastructure>, acesso em: 9 out. 2025.