

# Sistema Inteligente de Supervisão e Irrigação Usando Kafka, Microcontroladores e Computação em Nuvem

Kenner Azevedo<sup>1</sup> e Paulo Diniz<sup>2</sup>

## I. RESUMO

As mudanças climáticas afetam diretamente a produção agrícola no Brasil, de acordo com trabalho de Viola e Mendes [1], a variação climática é responsável por aproximadamente 30% da produtividade dos alimentos. Ou seja, quando o clima é imprevisível, a produção diminui, afetando principalmente a agricultura familiar, com menos recursos e capacidades tecnológicas.

Este projeto visa criar um sistema de supervisão de baixo custo para agricultura de precisão, que coleta e processa dados primários e secundários sobre fatores pedoclimáticos e ambientais. O sistema combina sensores e atuadores controlados por um microcontrolador e utiliza dados meteorológicos da plataforma OpenWeather. Ao integrar informações específicas do campo com previsões meteorológicas, o sistema possibilita práticas de irrigação mais precisas.

### A. Arquitetura Geral do Sistema

A arquitetura do projeto, ilustrada na Figura 3, é composta por componentes interligados que trabalham para garantir o monitoramento e controle eficientes do sistema agrícola.

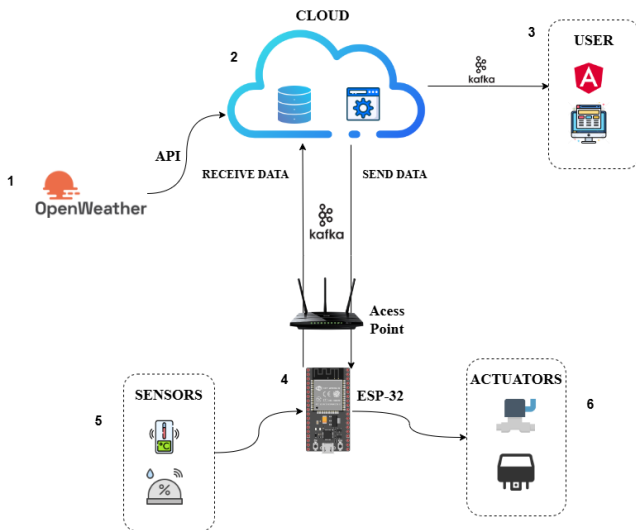


Fig. 1. Arquitetura

A seguir, está a descrição detalhada de cada componente da arquitetura:

- 1) **OpenWeather API:** fornece dados meteorológicos que são integrados ao sistema. Esses dados são enviados para o produtor na infraestrutura cloud, que os insere no Kafka como mensagens. Em seguida, o consumer

processa essas mensagens e realiza a inserção dos dados no banco de dados, garantindo uma comunicação eficiente e assíncrona entre os serviços.;

### 2) Cloud:

- a) Está conectada à API OpenWeather, de onde pode receber dados meteorológicos para complementar os dados dos sensores locais;
- b) Envia dados para a interface do usuário (UI) usando Kafka;
- c) Recebe os dados enviados pela ESP32;
- d) Processa os dados e os armazena em um banco de dados;
- e) Pode enviar comandos de volta à ESP32 para atuar nos dispositivos;

- 3) **User:** recebe dados da nuvem e pode monitorar ou controlar o sistema em tempo real;

### 4) ESP32:

- a) Recebe os dados dos sensores;
- b) Envia esses dados para a nuvem usando o protocolo MQTT ;
- c) Recebe comandos da nuvem para controlar os atuadores (como bombas d'água ou motores);

- 5) **Sensors:** os sensores conectados à ESP-32 coletam dados, como temperatura e umidade;

- 6) **Actuators:** recebem comandos da ESP-32 para realizar ações, como ligar ou desligar dispositivos.

### B. Arquitetura Front-Backend

A arquitetura apresentada no seu projeto utiliza um Reverse Proxy como elemento central para gerenciar e encaminhar as requisições entre os usuários e os serviços internos.

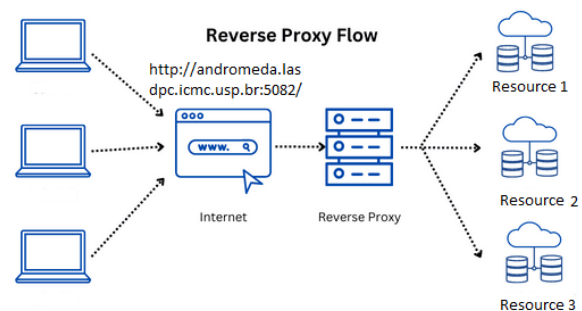


Fig. 2. Arquitetura

O frontend, acessível publicamente por meio da URL <http://andromeda.lasdpic.icmc.usp.br:5082/>, serve como a interface principal para interação dos usuários. Esse frontend, desenvolvido em React, está hospedado de maneira a ser acessível via internet, enquanto o backend e outros recursos permanecem isolados, acessíveis apenas pelo reverse proxy. O Reverse Proxy é essencial na comunicação entre o frontend e os serviços internos, recebendo requisições dos usuários e encaminhando-as para os serviços apropriados, como o backend ou bancos de dados. Ele protege o backend contra acessos diretos, melhora a segurança ao ocultar detalhes da infraestrutura interna e permite balanceamento de carga e gerenciamento de conexões HTTPS. Além disso, os recursos internos permanecem protegidos e acessíveis apenas pelo backend, garantindo flexibilidade e escalabilidade ao sistema, com centralização da gestão e maior eficiência da arquitetura.

## II. TRABALHOS RELACIONADOS

### A. *An IoT-Enabled Real-Time Smart Agriculture Monitoring and Intrusion Detection System [2]*

O estudo apresenta um sistema de monitoramento e segurança agrícola baseado em Internet das Coisas (IoT), projetado para otimizar a produtividade e reduzir o impacto ambiental na agricultura. O sistema utiliza um microcontrolador ESP32 para integrar sensores de umidade do solo, temperatura, umidade e nível de água, que coletam dados em tempo real e os enviam para a nuvem via módulo Wi-Fi ou módulo GSM. O microcontrolador também pode acionar bombas d'água via relé para ligar ou desligar a irrigação. Adicionalmente, um sensor PIR detecta movimentos suspeitos de animais ou seres humanos e alerta o proprietário. O sistema é caracterizado por seu baixo custo e eficiência energética, proporcionando uma solução prática e eficaz para enfrentar os desafios da agricultura moderna. O artigo conclui que o modelo proposto melhora a gestão agrícola e a segurança dos cultivos, com planos futuros para otimizar ainda mais a vida útil da bateria e a eficiência geral do sistema.

### B. *Data-Driven Precision Agriculture Advanced Irrigation System for Sustainable Smart Farming [3]*

Este artigo descreve um sistema de irrigação proposto para a Agricultura de Precisão no contexto da agricultura inteligente. O sistema integra sensores (umidade e temperatura do solo, umidade e temperatura do ar, nível de água e sensores de luz). Além disso, ele incorpora dados meteorológicos de uma plataforma de previsão, que inclui variáveis como velocidade e direção do vento e condições climáticas. Ao coletar e analisar dados específicos do campo junto com as previsões meteorológicas, o sistema permite a implementação de práticas de irrigação precisas e sustentáveis.

### C. *Web Scraping based Smart irrigation system with telegram alerts for farmers [4]*

Este artigo propõe um sistema que atualiza parâmetros em tempo real, como temperatura, umidade do solo e umidade

do ar, para qualquer região da Índia, utilizando técnicas de raspagem na web. Com isso, os agricultores podem acompanhar suas lavouras e protegê-las de chuvas fortes, além de controlar remotamente os motores de irrigação por meio do aplicativo. Adicionalmente, o aplicativo Telegram também fornece previsões do tempo para qualquer parte da Índia, ajudando os agricultores a minimizar perdas financeiras durante períodos críticos.

### D. *IoT based Smart Irrigation System using ESP32 [5]*

O estudo descreve o desenvolvimento de um sistema de irrigação inteligente baseado em IoT para a agricultura na Índia, visando otimizar o uso da água e melhorar a produtividade das culturas. A proposta envolve o uso de sensores de umidade, temperatura e umidade do solo conectados a um microcontrolador ESP32, que transmite dados para a nuvem via Wi-Fi. O sistema permite o monitoramento contínuo via app e SMS e o controle automático da irrigação com base nas condições climáticas e nos parâmetros do solo, ajustando a quantidade de água conforme necessário. O estudo compara a irrigação tradicional com a inteligente, demonstrando que a irrigação inteligente melhora o crescimento das plantas e reduz o desperdício de água. Além disso, o sistema é acessível por meio de um aplicativo móvel, que permite aos agricultores configurar e monitorar o processo de irrigação em tempo real, proporcionando uma gestão mais eficiente e sustentável da água.

### E. *Microcontroller Based Environmental Pollution Monitoring System through IoT Implementation [6]*

Nesse estudo, foi proposto e implementado um sistema automatizado para o monitoramento e notificação da poluição do ar e sonora, baseado em um microcontrolador (Arduino Uno). Este sistema visa mitigar os efeitos negativos da poluição e auxiliar na adoção de medidas corretivas, fornecendo informações digitais sobre os horários de maior concentração de poluentes ao longo do dia. O sistema detecta gases prejudiciais e níveis elevados de ruído utilizando sensores integrados ao microcontrolador. Os dados coletados pelos sensores são processados pelo microcontrolador e enviados para um servidor online, com o suporte da plataforma analítica de IoT (Internet das Coisas).

### F. *Internet of Things based Smart Environmental Monitoring using the Raspberry-Pi Computer [7]*

Este artigo desenvolve um dispositivo de monitoramento ambiental econômico e padronizado, utilizando um Raspberry Pi (R-Pi). O sistema, projetado com a linguagem de programação Python, pode ser controlado e acessado remotamente por meio de uma plataforma de Internet das Coisas. Ele capta dados ambientais através de sensores e os transmite diretamente para a internet, permitindo acesso a essas informações a qualquer hora e de qualquer lugar. Os resultados experimentais mostraram que o sistema mede com precisão a temperatura, a umidade, o nível de luz e as concentrações do poluente atmosférico monóxido de carbono. Além disso, o sistema inclui um sensor sísmico para a detecção de terremotos.

### G. Advanced Sensorless Weather Station Implementation Using ESP32 [8]

Este artigo descreve um sistema IoT que coleta dados climáticos de uma API de previsão para um local configurável e os exibe em uma tela TFT sensível ao toque. O sistema utiliza um microcontrolador ESP32, aproveitando seu Wi-Fi integrado para comunicação de dados. Adicionalmente, foi implementado um servidor web no ESP32 usando SPIFFS (Serial Peripheral Interface Flash File System), permitindo o monitoramento e controle do modo de consumo de energia em uma rede local, assim como a conexão segura com a Internet via laptop, celular ou tablet. Todas essas funcionalidades foram implementadas com o suporte multitarefa do FreeRTOS, otimizando o uso do microcontrolador dual-core ESP32.

### H. A Smart Home Agriculture System Based on Internet of Things [9]

Este artigo propõe um sistema de IoT voltado para a agricultura doméstica, destinado a criar pequenas fazendas em ambientes internos. O sistema utiliza dispositivos ESP32 e sensores conectados para monitorar condições do solo e do ar. Os dados coletados são enviados a um servidor web por meio de uma conexão Wi-Fi. Esse servidor armazena as informações recebidas e gera relatórios em tempo real sobre o ambiente e o estado das plantas. Com base nesses dados e nas necessidades específicas das plantas, o servidor pode enviar notificações diretamente para o celular do usuário.

### I. Design and Implementation of Intelligent Systems Using ESP-32 Based Fuzzy Logic on Smart Agriculture System for Mustard Plants [10]

Este artigo aborda a resolução de um problema por meio da criação e implementação de sistemas inteligentes utilizando lógica fuzzy com o ESP32 em sistemas de agricultura inteligente para plantas de mostarda. A pesquisa inclui a medição de vários parâmetros: umidade do solo com um sensor específico, temperatura do ar com um sensor DHT22 e altura da planta com um sensor infravermelho. Os dados coletados pelos sensores são processados pelo microcontrolador com lógica fuzzy, a fim de determinar a duração da irrigação conforme a condição das plantas. As leituras dos sensores são armazenadas em um banco de dados e apresentadas em um LCD e em um site para o monitoramento em tempo real do sistema.

### J. Agriculture 4.0 and climate change in Brazil [1]

Este artigo introduz o debate sobre transições sociotécnicas de baixo carbono no agronegócio brasileiro, partindo da premissa de que uma parcela crescente do setor adota tecnologias digitais em seus modelos de negócios, configurando um novo paradigma produtivo, a agricultura 4.0. O artigo utiliza a teoria de "Transições para a Sustentabilidade" para examinar a relação entre agricultura 4.0 e mudanças climáticas no Brasil. Para tanto, o artigo classifica o agronegócio em três subsectores: agricultura familiar, agronegócio conservador e agronegócio sustentável. O artigo demonstra desafios de

ordem técnica, institucional e político-econômica para cada um desses subsectores avançar em direção a uma economia de baixo carbono, sendo o agronegócio sustentável o que tem tido melhor desempenho até o momento.

## III. METODOLOGIA

A estrutura do projeto é organizada nos seguintes componentes, que serão descritos a seguir:

- MongoDB
- Kafka
- Broker MQTT - Mosquitto
- ProducerOpenWeather
- ConsumerOpenWeather
- Middleware - Back-End
- Front-end
- DataEsp - Recebimento de dados do mcu
- Command ESP - Envio de comando para o mcu

As instâncias de Back-End e Front-End serão responsáveis por integrar o código para desenvolver uma aplicação que possibilite a visualização, gerenciamento e administração de todo o projeto. O Back-End interagirá com os dados armazenados no banco de dados MongoDB e com a infraestrutura do ESP-32 [11].

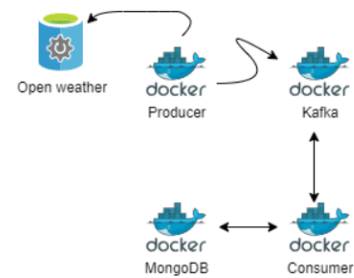


Fig. 3. Arquitetura core dos containers

### A. Ambiente Operacional

A solução será implantada em um ambiente de nuvem, com a infraestrutura inicialmente configurada na máquina virtual cgrupo3 da USP, utilizando um contêiner Docker para o Kafka [12] [13]. Essa abordagem facilita a comunicação entre o produtor e o consumidor através de tópicos e grupos, promovendo uma gestão eficiente dos dados.

1) *Integração com a OpenWeather API:* A aplicação será conectada à OpenWeather API [14], que fornece dados meteorológicos. Esses dados serão enviados para a nuvem, permitindo que o sistema tome decisões informadas com base nas condições climáticas.

2) *Funcionalidade da Nuvem: Recepção de Dados:* A nuvem receberá dados da OpenWeather API para complementar as informações obtidas pelos sensores locais.

**Envio de Dados para a Interface do Usuário:** Utilizando Kafka, a nuvem enviará dados processados para a interface do usuário (UI), permitindo que os usuários visualizem as informações em tempo real.

**Comunicação com o ESP-32:** A nuvem receberá os dados enviados pela ESP-32 e processará essas informações para armazenamento em um banco de dados.

**Comandos para o ESP-32:** A nuvem poderá enviar comandos de volta ao ESP-32, permitindo o controle de atuadores e outros dispositivos conectados.

3) *Interação do Usuário:* Os usuários receberão dados da nuvem e poderão monitorar ou controlar o sistema em tempo real, garantindo uma experiência interativa e responsiva por meio de uma aplicação Front-End/Back-End escrita em JS, HTML, CSS e frameworks necessários.

4) *Funcionamento do ESP-32:* :

O ESP-32 receberá dados dos sensores conectados, que coletarão informações como temperatura e umidade. Esses dados serão enviados para a nuvem usando o protocolo MQTT, assegurando uma comunicação eficiente.

Além disso, o ESP-32 receberá comandos da nuvem para controlar atuadores, como bombas d'água ou motores, em resposta às condições climáticas ou às solicitações dos usuários.

5) *Sensores e Atuadores:* :

Os sensores conectados ao ESP-32 desempenham um papel crucial na coleta de dados ambientais. O DHT11, utilizado no sistema, é um sensor simples e acessível usado para medir **temperatura** (0°C a 50°C) e **umidade relativa** (20% a 90%). Com fácil integração a microcontroladores e baixo consumo de energia, ele é ideal para projetos de monitoramento ambiental e IoT. No projeto, o DHT11 coleta dados em tempo real, como temperatura e umidade, que são enviados ao sistema para análise e uso em aplicações específicas. Os atuadores, por sua vez, receberão comandos do ESP-32 para executar ações, como ligar ou desligar dispositivos, garantindo a automação do sistema.

6) *Front-End:* O front-end do projeto foi desenvolvido utilizando o template "Mantis" disponibilizado pela CodedThemes (<https://codedthemes.gitbook.io/mantis>). Esse template é baseado em React e oferece uma estrutura moderna e responsiva, com diversos componentes pré-construídos que aceleram o desenvolvimento da interface. Ele foi escolhido por sua flexibilidade, design intuitivo e suporte a integração com APIs, o que facilita a comunicação com o back-end.

Todos os códigos da aplicação estão disponíveis em : [github/gcloudpos03](https://github.com/gcloudpos03)

## B. Testes Iniciais

Os testes iniciais da infraestrutura incluirão a validação da comunicação entre os contêineres, a verificação do funcionamento do Kafka e a integração com os serviços de nuvem. A abordagem utilizará o protocolo MQTT [15] para estabelecer comunicação via Wi-Fi entre o ESP-32 e a infraestrutura em nuvem, permitindo o envio e recebimento de dados. Por exemplo, implementaremos um sistema que possibilita a irrigação de plantas com base em métricas analisadas a partir dos dados do Open Weather.

Todas as especificações do ESP-32 estarão na pasta "Firmware", que conterá os códigos em C++. Essas etapas

garantirão que a infraestrutura esteja robusta e funcional, pronta para suportar as operações do projeto.

## IV. RESULTADOS

Nesta seção, avaliamos os resultados obtidos com o desenvolvimento da aplicação, que inclui funcionalidades de visualização de dados meteorológicos e controle de dispositivos ESP.

### A. Visualização dos Gráficos Meteorológicos

Na Figura 4, apresentamos a interface de visualização de dados meteorológicos em formato de gráficos de linha. Temos tanto a representação da umidade quanto temperatura, além de um campo específico para mostrar a velocidade do vento. Essa visualização permite ao usuário monitorar rapidamente as condições atmosféricas, auxiliando no controle e análise das variáveis meteorológicas relevantes.



Fig. 4. Interface de visualização dos gráficos meteorológicos.

### B. Comando de Dispositivos MCU's

A Figura 5 apresenta a interface de comando para dispositivos ESP, onde o usuário pode desligar ou ligar um atuador referente a uma ESP, além de inspecionar a temperatura de umidade referente a essa ESP. Essa funcionalidade permite ao usuário controlar remotamente dispositivos conectados ao sistema, sendo útil em contextos onde seja necessário ajustar o comportamento dos dispositivos com base em condições de ambiente ou requisitos operacionais.

Módulos Integrados					
ID	DISPOSITIVO	TEMPERATURA	UMIDADE	ATUADOR	DETALHES
6742476d8329e61ba9c1c18d	esp3	28 °C	55 %	Desligar Atuador	Ver Detalhes
6742476d8329e61ba9c1c18c	esp1	27 °C	55 %	Desligar Atuador	Ver Detalhes

Fig. 5. Interface de comando dos dispositivos ESP.

### C. Visualização dos Dados da MCU's

Na Figura 6, apresentamos a interface de visualização dos dados temperatura e umidade coletados da ESP cadastrados no sistema. Cada ESP tem seu próprio gráfico exibindo seus dados. Essa funcionalidade permite ao usuário monitorar o status e os detalhes dos dispositivos ativos no sistema, facilitando a manutenção e controle de cada dispositivo de forma individual.

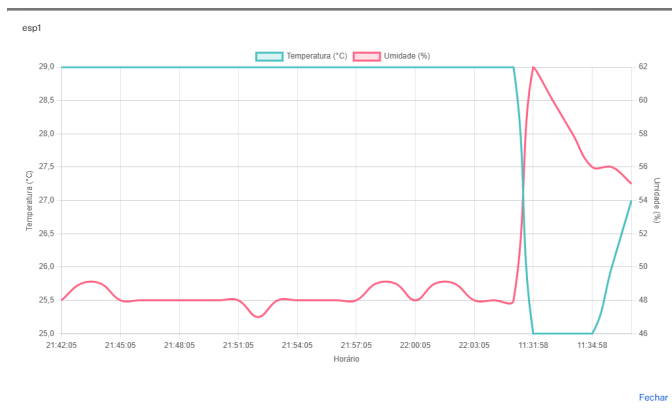


Fig. 6. Interface de visualização dos dispositivos ESP cadastrados.

#### D. Circuito

A Figura 7, apresentamos o circuito desenvolvido usando o microcontrolador ESP32, o sensor DHT11 e um Led, deixando claro a simplicidade de construção e integração além de evidenciar o baixo custo do projeto. Esse circuito pode facilmente ser integrado com outros atuadores de forma a complementar sua função e expandir as possibilidades de análises no front end.

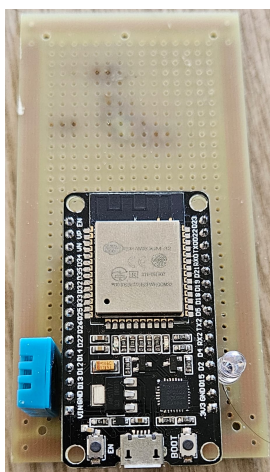


Fig. 7. Circuito do dispositivo ESP32 utilizado no sistema.

#### E. Análise dos Resultados

A aplicação desenvolvida apresenta uma interface gráfica funcional e interativa que facilita o monitoramento de variáveis meteorológicas e o controle de dispositivos ESP. Os gráficos de barras permitem uma análise rápida dos dados meteorológicos, enquanto as interfaces de comando e visualização dos dispositivos ESP proporcionam um controle eficiente e uma visão clara dos dispositivos em operação. Esses resultados demonstram a eficácia da aplicação em fornecer um sistema de monitoramento e controle centralizado, sendo uma ferramenta útil em aplicações de IoT e automação ambiental.

#### V. CONCLUSÃO

Destacamos aqui as principais contribuições do sistema desenvolvido para a agricultura de precisão, especialmente no contexto de uma gestão eficiente e sustentável da irrigação. Ao fornecer uma solução tecnológica baseada na integração de IoT e nuvem, o sistema permite aos agricultores monitorar e controlar a irrigação com precisão, otimizando o uso de água. Essa abordagem não só controla o desperdício hídrico, mas também minimiza os impactos das mudanças climáticas ao ajustar o consumo de acordo com as necessidades reais das culturas, promovendo práticas agrícolas mais responsáveis e alinhadas com a sustentabilidade ambiental.

A arquitetura implementada, que combina dispositivos ESP32, a plataforma de mensagens em tempo real Kafka e a nuvem, foi projetada com foco em flexibilidade e escalabilidade. Além disso, o uso da nuvem para armazenamento e processamento permite a centralização e a análise eficiente dos dados de campo, oferecendo uma visão abrangente e detalhada dos parâmetros críticos para a irrigação. Essa abordagem também facilita a incorporação futura de novos sensores e módulos, tornando o sistema capaz de acompanhar as evoluções tecnológicas sem a necessidade de mudanças significativas na infraestrutura.

Adicionalmente, o impacto do sistema para a agricultura familiar no Brasil merece destaque. A solução proposta, com custos acessíveis e baseada em uma tecnologia de fácil adoção, oferece uma oportunidade real para pequenos agricultores melhorarem suas práticas de cultivo, sem a necessidade de um alto investimento em tecnologias complexas. Dessa forma, a agricultura familiar pode se beneficiar de um sistema de irrigação mais eficiente, que contribui para aumentar a produtividade e reduzir os custos operacionais.

Confira o projeto no GitHub: [gcloudpos03.eldo](https://github.com/gcloudpos03.eldo)

#### REFERENCES

- [1] Eduardo Viola and Vinícius Mendes. Agriculture 4.0 and climate change in brazil. *Ambiente Sociedade*, 25:e02462, 2022.
- [2] Shafi Shaik, MD. Adhil, M. Jennifer, Mohan Krishna, B Kalyan, and Humera Begum. An iot-enabled real-time smart agriculture monitoring and intrusion detection system. In *2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, pages 1–5, 2024.
- [3] Akram Ghilan, EL Youssef, Mostafa MERRAS, and Nabil El Akkad. Data-driven precision agriculture advanced irrigation system for sustainable smart farming. In *2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, pages 01–06. IEEE, 2024.
- [4] Nagarjuna Telagam, U Somanaidu, B Naresh, M Arun Kumar, and K Nehru. Web scraping based smart irrigation system with telegram alerts for farmers. In *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–6. IEEE, 2021.
- [5] JE Tamil Malar and M Vaishnavi. Iot based smart irrigation system using esp32. In *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 1751–1755. IEEE, 2022.
- [6] Nadia Nowshin and Md Shajedul Hasan. Microcontroller based environmental pollution monitoring system through iot implementation. In *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pages 493–498. IEEE, 2021.

- [7] Mohannad Ibrahim, Abdelghafor Elgamri, Sharief Babiker, and Ahmed Mohamed. Internet of things based smart environmental monitoring using the raspberry-pi computer. In *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, pages 159–164. IEEE, 2015.
- [8] Jugurtha Chettouh and Samia Mezzah. Advanced sensorless weather station implementation using esp32. In *Advances in Computing Systems and Applications: Proceedings of the 4th Conference on Computing Systems and Applications*, pages 165–174. Springer, 2021.
- [9] Vahid Khalilpour Akram and Moharram Challenger. A smart home agriculture system based on internet of things. In *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–4. IEEE, 2021.
- [10] Anisa Lutfiani Salsabila, I Gede Puja Astawa, Anang Budikarso, and Budi Aswoyo. Design and implementation of intelligent systems using esp-32 based fuzzy logic on smart agriculture system for mustard plants. In *2023 International Electronics Symposium (IES)*, pages 224–229. IEEE, 2023.
- [11] Darko Hercog, Tone Lerher, Mitja Truntič, and Oto Težak. Design and implementation of esp32-based iot devices. *Sensors*, 23(15), 2023.
- [12] Neha Narkhede, Gwen Shapira, and Todd Palino. *Kafka: The Definitive Guide Real-Time Data and Stream Processing at Scale*. O’Reilly Media, Inc., 1st edition, 2017.
- [13] Jay Kreps. *Kafka : a distributed messaging system for log processing*. 2011.
- [14] OpenWeather. Api documentation, 2024. Acessado em: 29 de setembro de 2024.
- [15] G.C. Hillar. *MQTT Essentials - A Lightweight IoT Protocol*. Packt Publishing, 2017.