

IrrigaFlow: An IoT Architecture Integrated with Fuzzy Logic for Sustainable Agricultural Irrigation Optimization

Francisco Fábio de Oliveira
ffabioliveira81@gmail.com
Universidade Federal do Rio Grande
do Norte (UFRN)
Natal, RN, Brasil

Diego Pereira
diego.pereira@ifrn.edu.br
Instituto Federal do Rio Grande do
Norte (IFRN)
Parnamirim, RN, Brasil

Douglas D. J de Macedo
douglas.macedo@ufsc.br
Universidade Federal de Santa
Catarina (UFSC)
Florianópolis, SC, Brasil

Geraldo Pereira Rocha Filho
geraldo.rocha@uesb.edu.br
Universidade Estadual do Sudoeste da
Bahia (UESB)
Vitória Da Conquista, BA, Brasil

Roger Immich
roger@imd.ufrn.br
Universidade Federal do Rio Grande
do Norte (UFRN)
Natal, RN, Brasil

Abstract

Context: The growing demand for food security and the scarcity of natural resources pose global challenges, particularly in agriculture, which accounts for a significant portion of water consumption. In Brazil, agriculture holds substantial economic importance, but its intensive water use calls for more sustainable practices. **Problem:** Managing irrigation adaptively and efficiently is challenging due to the complexity of multiple environmental variables. This reduces water-use efficiency and negatively impacts agricultural productivity. **Solution:** IrrigaFlow is a modular architecture that automates irrigation using IoT, fuzzy logic, and distributed processing. It consists of three layers, namely IoT Module, Network Edge, and Cloud, enabling real-time monitoring and adjustments based on local environmental data. This approach optimizes water usage and improves responsiveness to climatic conditions. **Information Systems Theory:** Grounded in the Sociotechnical Systems Theory, this proposal balances advanced technology with human and organizational contexts, promoting efficiency and sustainability by dynamically adapting to environmental and operational conditions. **Methodology:** A qualitative interpretative approach was employed, combining case studies and simulations. Environmental data collection, fuzzy logic processing, and MQTT-based communication were validated to ensure the system's effectiveness before practical implementation. **Results:** The system demonstrated efficiency in irrigation management by adjusting water volume and timing based on environmental variables, showcasing its potential to optimize water use in agriculture. **Contributions and Impact on Information Systems:** From an academic perspective, this work lays a foundation for research on distributed technologies applied to agriculture. For the industry, it offers a replicable model that enhances water efficiency and sustainability.

CCS Concepts

• **Applied computing** → **Forecasting; Decision analysis;** •
Hardware → **Sensors and actuators.**

Keywords

IrrigaFlow, IoT, fuzzy logic, irrigation management, agricultural automation, sustainability

1 Introdução

A crescente demanda por segurança alimentar, combinada com a escassez de recursos naturais, representa um dos maiores desafios globais. Isso é especialmente evidente no setor agrícola, que consome grandes quantidades de água, energia e fertilizantes. Esse cenário resulta em impactos ambientais consideráveis, como aquecimento global e poluição, afetando diretamente a gestão de água em áreas agrícolas e aumentando a frequência de extremos hidrológicos, como secas e inundações [30]. Diante desse contexto, os Objetivos de Desenvolvimento Sustentáveis (ODS), especificamente o ODS 2 e o ODS 6, enfatizam a importância de práticas agrícolas sustentáveis e de uma gestão responsável dos recursos hídricos para assegurar um desenvolvimento global equilibrado [27].

No Brasil, a agricultura desempenha um papel fundamental no desenvolvimento econômico, correspondendo a uma parte significativa do Produto Interno Bruto (PIB) nacional [8]. Entretanto, o manejo eficiente da água na irrigação é desafiador, pois o consumo hídrico agrícola representa entre 70% e 80% do uso total em regiões intensamente irrigadas, destacando a necessidade de otimizar a gestão desses recursos [6]. A adoção de práticas que promovam um uso racional da água contribui para aumentar a produtividade agrícola, ao mesmo tempo em que reduz os custos e minimiza os impactos ambientais.

Para alcançar uma gestão mais eficaz da irrigação, é essencial empregar soluções tecnológicas que auxiliem na tomada de decisões e na alocação adequada dos recursos hídricos. O uso de sistemas automatizados baseados em tecnologias de Internet das Coisas (IoT) e computação em nuvem tem se mostrado uma abordagem viável, permitindo um controle adaptativo e ajustado às diferentes condições de cultivos e clima [21]. Esses sistemas podem integrar diversos parâmetros, oferecendo uma estratégia de gerenciamento que equilibra a demanda das culturas e a disponibilidade de água.

Neste contexto, é proposto o **IrrigaFlow**, uma arquitetura projetada com suporte a algoritmos de decisão baseados em lógica fuzzy. Essa abordagem permite que os sistemas de irrigação sejam mais responsivos às variações ambientais e às necessidades de diferentes tipos de cultura, otimizando o uso da água e promovendo práticas agrícolas sustentáveis. A integração de camadas de processamento

de dados, desde a coleta até a análise em nuvem, facilita a coordenação das informações e o controle eficiente da irrigação, adaptando as operações de acordo com as mudanças nas condições de cultivo. Essa estrutura busca oferecer um gerenciamento mais eficiente e adaptável da irrigação, contribuindo para um uso mais racional dos recursos hídricos.

O restante do trabalho é dividido da seguinte forma. A Seção 2 aborda a fundamentação teórica essencial para o entendimento do tema; a Seção 3 discute os trabalhos relacionados, destacando suas principais contribuições e limitações; a Seção 4 detalha o desenvolvimento e a estrutura da metodologia aplicada; e, por fim, a Seção 5 apresenta as conclusões do estudo e propostas para trabalhos futuros.

2 Fundamentação Teórica

Esta seção apresenta os fundamentos teóricos, abrangendo Dispositivos IoT e suas aplicações, borda da rede e nuvem, lógica *fuzzy* e a plataforma Node-RED, que é uma ferramenta de programação visual baseada em fluxos, amplamente utilizada para o desenvolvimento de sistemas IoT.

2.1 Princípios da Irrigação Agrícola

A irrigação é fundamental para garantir a produtividade agrícola, especialmente em regiões com precipitação irregular ou insuficiente. Os principais métodos de irrigação incluem:

- **Irrigação por superfície:** A água é distribuída diretamente sobre o solo, infiltrando-se até as raízes das plantas. Exemplos incluem irrigação por sulcos e inundação. Embora simples e econômico, esse método pode apresentar perdas significativas por evaporação e percolação profunda [25].
- **Irrigação por aspersão:** A água é pulverizada sobre as plantas em forma de chuva artificial, utilizando aspersores fixos ou móveis. Esse método melhora a uniformidade da aplicação de água, mas pode ser influenciado por fatores climáticos, como vento e temperatura [25].
- **Irrigação localizada (gotejamento):** A água é aplicada diretamente na zona radicular das plantas por meio de emissores, como gotejadores. Esse método minimiza desperdícios e é altamente eficiente em termos de uso da água [25].
- **Irrigação por pivô central:** Utilizada principalmente em grandes áreas, esse sistema distribui água de maneira uniforme sobre as plantações, movimentando um conjunto de aspersores ao redor de um ponto central [25].

A escolha do método de irrigação depende de fatores como tipo de solo, topografia, disponibilidade de água e tipo de cultura. Uma seleção adequada contribui para a eficiência no uso da água e para a sustentabilidade agrícola.

2.2 Automação da Irrigação

A automação na irrigação permite o controle preciso da aplicação de água, utilizando tecnologias que monitoram variáveis ambientais e ajustam os sistemas de irrigação em tempo real. Isso resulta em uma gestão mais eficiente dos recursos hídricos e na melhoria da produtividade agrícola.

2.3 Dispositivos IoT e suas Aplicações

A IoT permite a comunicação entre objetos físicos e a Internet, possibilitando a coleta, o processamento e o compartilhamento de dados para aprimorar processos em setores como saúde e cidades inteligentes [11]. Essa interconexão ocorre por meio de sensores, atuadores e dispositivos embarcados que captam informações do ambiente e as transmitem para sistemas de controle e análise, criando um ecossistema interligado [5].

Os dispositivos IoT utilizam sensores para coletar variáveis como temperatura, umidade e luminosidade, e atuadores que podem executar ações automatizadas, como ajuste de temperatura ou controle de válvulas. Para que essa comunicação ocorra de maneira eficiente e escalável, os dispositivos são conectados a redes maiores por meio de *gateways*, que atuam como intermediários, gerenciando o fluxo de informações entre os sensores locais e servidores na nuvem.

A transmissão de dados entre dispositivos IoT pode ser realizada por diversos protocolos, sendo dois dos mais empregados o MQTT (*Message Queuing Telemetry Transport* – Transporte de Telemetria com Fila de Mensagens) e o CoAP (*Constrained Application Protocol* – Protocolo para Aplicações com Restrições). O MQTT é um protocolo de comunicação baseado no modelo *publish/subscribe* (publicação/assinatura), projetado para ambientes com restrições de largura de banda e alta latência. Ele é amplamente utilizado em aplicações IoT devido à sua eficiência na transmissão de mensagens com baixo consumo de energia e recursos computacionais [13].

Já o CoAP, foi desenvolvido especificamente para dispositivos IoT com recursos limitados, oferecendo um modelo de comunicação semelhante ao HTTP (Hypertext Transfer Protocol – Protocolo de Transferência de Hipertexto), mas adaptado para redes de baixa largura de banda. Diferente do MQTT, que usa um modelo assíncrono de publicação e assinatura, o CoAP funciona em um esquema *request/response* (requisição/resposta) sendo mais adequado para sistemas que precisam de comunicação direta ponto a ponto [22].

Para reduzir a latência e distribuir a carga de processamento, muitos dispositivos IoT processam dados localmente em camadas de borda antes de enviá-los para a nuvem, permitindo que decisões sejam tomadas rapidamente sem depender exclusivamente de uma conexão externa [23]. A IoT pode ser organizada em três camadas, conforme descrito abaixo.

- **Camada de percepção:** Responsável pela captura de dados do ambiente por meio de sensores e atuadores, que coletam informações sobre temperatura, umidade, luminosidade e outros fatores ambientais.
- **Camada de Rede:** Garante a transporta dados entre dispositivos IoT, *gateway* e servidores, empregando uma variedade de protocolos de comunicação, incluindo MQTT, CoAP e outros, conforme a necessidade do sistema. Além disso, essa camada pode utilizar diferentes tecnologias de transmissão, como redes cabeadas ou sem fio. Nessa camada, também podem ser implementados nós de *fog computing* para processamento descentralizado de dados.
- **Camada de Aplicação:** Interage com o usuário e armazena os dados coletados, frequentemente utilizando serviços baseados em nuvem para análise e visualização em tempo real.

Esse modelo de camadas organiza os componentes da IoT de forma estruturada. Isto permite que diferentes tecnologias sejam

combinadas de acordo com as necessidades específicas de cada aplicação.

2.4 Funcionamento da borda da rede e da nuvem

A computação em nuvem, sozinha, pode não atender a requisitos de latência para aplicativos IoT devido ao tempo necessário para processar grandes volumes de dados. A computação em borda descentraliza serviços da nuvem, aproximando-os dos usuários para reduzir a latência [11, 18, 21]. A borda atua nos pontos entre a origem dos dados e o *data center* em nuvem, permitindo processamento local e a troca de dados com a nuvem [4, 5].

A computação em nuvem surgiu como uma abstração de redes para centralizar armazenamento e o processamento de dados, utilizando a Internet. Entre os benefícios da computação em nuvem, destaca-se a redução de custos operacionais, pois as empresas deixam de investir em infraestrutura própria de servidores, manutenção física, consumo de energia e gerenciamento de hardware local. Além disso, o modelo de pagamento flexível permite escalabilidade conforme a demanda [28].

A computação em nuvem pode ser classificada em quatro tipos principais:

- **Nuvem Pública:** Gerenciada por provedores terceirizados, acessível por múltiplos usuários, como AWS, Microsoft Azure e Google Cloud.
- **Nuvem Privada:** Infraestrutura exclusiva de uma única organização, proporcionando maior controle e segurança.
- **Nuvem de Comunidade:** Compartilhamento entre várias organizações com necessidades semelhantes, como universidades ou entidades governamentais.
- **Nuvem Híbrida:** Combina nuvens públicas e privadas, permitindo flexibilidade na alocação de recursos e maior controle sobre dados críticos.

2.5 Fundamentos da Lógica Fuzzy

A lógica *fuzzy* ampliou a teoria dos conjuntos clássicos, permitindo graus intermediários de pertinência e oferecendo uma base matemática para lidar com a imprecisão e incerteza do raciocínio humano [20]. Criada para superar as limitações da lógica binária, essa abordagem reflete melhor a complexidade de situações do mundo real.

A lógica *fuzzy* se diferencia da lógica clássica ao permitir valores de verdade graduados entre 0 e 1, representando níveis de pertinência e permitindo que elementos pertençam parcialmente a diferentes conjuntos [7]. Por exemplo, uma pessoa de 50 anos pode ter um grau de pertinência de 0,4 na categoria “jovem” e 0,6 em “idoso” [26]. Essa flexibilidade é útil para lidar com conceitos vagos, como “frio” ou “quente”, que não se limitam a classificações absolutas.

A lógica *fuzzy* tem se mostrado eficaz em diversas áreas, como controle de sistemas complexos e tomada de decisões. Existem exemplos de aplicação deste para melhorar a qualidade da transmissão de vídeos [16], inclusive de vídeos transmitidos por aeronaves não tripuladas [15]. Outro exemplo é o metrô de Tóquio, que obteve maior eficiência em comparação aos métodos tradicionais [17].

Além disso, a lógica *fuzzy* é usada em aplicações que exigem flexibilidade e tratamento de incertezas, como diagnósticos médicos e sistemas de controle de processos industriais [20].

Nos conjuntos *fuzzy*, a pertinência de um elemento é representada por valores entre 0 e 1, permitindo uma transição gradual entre categorias [29]. Essa abordagem captura melhor as nuances da percepção humana, possibilitando respostas que vão além do simples “sim” ou “não”, adaptando-se a cenários onde a subjetividade é relevante [26].

Um sistema de inferência *fuzzy* utiliza regras do tipo “Se... então...” para processar entradas e gerar saídas que refletem a realidade com mais precisão. Essa abordagem é particularmente útil em situações contínuas e complexas, onde a indefinição é comum e a modelagem tradicional não é suficiente [10].

2.6 Plataforma de Programação Visual Node-RED

O Node-RED é uma ferramenta projetada para simplificar o desenvolvimento de aplicativos de IoT por meio de uma abordagem de programação visual. Ele permite que desenvolvedores criem fluxos de trabalho conectando blocos de código pré-definidos, chamados ‘nós’, para executar tarefas específicas de forma eficiente [19].

Conforme apresentado na Figura 1, a plataforma oferece um editor baseado em navegador que facilita a construção e conexão de fluxos, utilizando uma ampla gama de nós disponíveis na paleta.

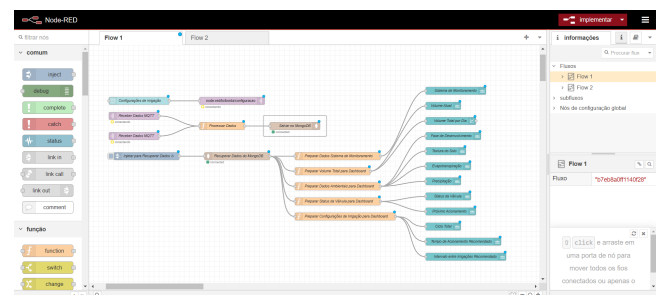


Figura 1: Interface da Plataforma Node-RED

O Node-RED é popular por suas características de simplificação, eficiência e praticidade. O editor de fluxo permite a criação rápida e intuitiva de aplicativos, eliminando a necessidade de codificação complexa e garantindo componentes consistentes e funcionais [12].

3 Trabalhos relacionados

Diversos estudos exploram essas abordagens para aumentar a precisão e a eficiência dos sistemas de irrigação, considerando variáveis climáticas e ambientais. Nesta seção, são apresentados trabalhos que aplicam essas tecnologias em diferentes contextos agrícolas.

3.1 Sistema de irrigação inteligente com IoT e inferência fuzzy de Sugeno

O estudo propõe um sistema de irrigação inteligente baseado em IoT e energia solar, aplicando a lógica fuzzy de Sugeno para determinar o momento, duração e quantidade de água a ser aplicada, considerando variáveis climáticas e necessidades das plantas [2]. O

sistema é autônomo, dividido em três seções: aquisição de dados, análise e decisão, e monitoramento, utilizando sensores para captar condições locais e previsões de precipitação. A inferência fuzzy de Takagi-Sugeno analisa a umidade do solo e as previsões climáticas para ajustar a irrigação, enquanto o aplicativo Blynk monitora variáveis e a resposta do sistema. Os autores destacam que o modelo ajuda a economizar água e aumentar a produtividade agrícola, com irrigação realizada às 5h30 e 15h30, horários de maior queda de umidade.

3.2 Sistema de irrigação IoT baseado em lógica fuzzy

O sistema de irrigação proposto é baseado em IoT e lógica fuzzy visa controlar a bomba de água usando quatro parâmetros ambientais: temperatura do ar, umidade do ar, umidade do solo e intensidade de luz [24]. O projeto de hardware utiliza NodeMCU ESP8266, com sensores para captar as variáveis e um atuador composto por um conversor de energia, relé, fonte de 12V e bomba de água de 12V DC.

Os dados dos sensores são enviados via MQTT para o banco de dados, onde o sistema difuso analisa e define a saída. O modelo difuso possui 54 regras que determinam o estado da bomba ('On' ou 'Off'), utilizando funções de pertinência para interpretar as características dos dados ambientais.

3.3 Controle de umidade com irrigação automática usando Node-RED e MQTT

O estudo propõe um sistema de monitoramento e controle para irrigação automática de cactos, utilizando o NodeMCU ESP8266 como microcontrolador e sensores de temperatura e umidade do solo [9]. Os dados são processados com lógica fuzzy e publicados no broker MQTT para controle e monitoramento.

O sistema inclui três componentes principais: publisher, broker e subscriber. A lógica fuzzy integrada ao NodeMCU define a duração e a intensidade da irrigação por gotejamento. O painel de monitoramento é exibido no dashboard da plataforma Node-RED.

A modelagem fuzzy utiliza uma função de transferência para controlar a temperatura e umidade, com entradas de temperatura e umidade do solo e saída determinando o tempo de operação da bomba de água. Regras difusas foram criadas para cada combinação de entrada e saída.

Os resultados mostraram a eficácia do sistema, com erros calculados por equações específicas e dados enviados via MQTT, exibidos em gráficos na plataforma Node-RED. O sistema atende de forma eficaz às necessidades dos cactos, que exigem condições específicas de umidade e temperatura.

3.4 Controle climático em IoT para cultivo de cogumelos com lógica fuzzy

O estudo apresenta um sistema de controle climático baseado em IoT para cultivo de cogumelos-ostra, utilizando lógica fuzzy para melhorar a estabilidade e reduzir o uso de água, resultando em maior rendimento [3]. Dividido em ambiente de cultivo, controle climático e interface web, o sistema inclui IoT Control Box, Access

Point, bomba de água, ventilador, sensores e prateleiras, permitindo controle remoto e exibição de dados.

A lógica fuzzy é executada no NodeMCU, convertendo valores de temperatura e umidade em dados processados. A base de regras segue uma matriz de 25 regras de inferência Mamdani, testada com dados reais.

Sensores DHT22 monitoram temperatura e umidade, controlados via relés, com leituras exibidas em um display LCD. O sistema possui regulador de voltagem para garantir energia estável e mantém condições ideais no cultivo, protegendo os cogumelos e aumentando a eficiência do uso de água.

A solução automatiza a monitorização e regulação climática, permitindo gestão remota e maior produtividade no cultivo de cogumelos-ostra.

3.5 Sistema de irrigação inteligente com lógica fuzzy

O estudo apresenta um sistema de irrigação inteligente usando lógica fuzzy para melhorar a confiabilidade e eficácia do controle, considerando cinco parâmetros de entrada: temperatura, umidade, luminosidade, irradiação solar e velocidade do vento [14]. A saída do sistema é a taxa de bombeamento de água, variando de 0 a 400m³/s.

Os dados brutos são convertidos em conjuntos difusos por meio de fuzzificação, utilizando funções de adesão gaussianas para maior robustez. As entradas são descritas por regiões linguísticas como 'BAIXO', 'MÉDIO' e 'ALTO' para temperatura e outras descrições semelhantes para os demais sensores. As regras são modeladas com base nas condições ambientais, e a defuzzificação, usando a metodologia Sugeno, gera saídas nítidas.

O sistema foi testado com dados simulados no Matlab, comparando os métodos Mamdani e Sugeno. Os resultados indicaram que o método Sugeno é mais eficaz e confiável para otimizar a irrigação, adaptando-se automaticamente às variações climáticas e melhorando a eficiência no uso da água.

3.6 IrrigaFlow

Os estudos discutidos anteriormente exploram diversas abordagens para a automação da irrigação agrícola, utilizando tecnologias como IoT e lógica fuzzy para adaptar o consumo de água conforme as condições climáticas e as necessidades das plantas. Esses trabalhos, como os de [2] e [14], trouxeram avanços importantes, mas cada um apresenta limitações em termos de escopo e adaptabilidade. Por exemplo, alguns sistemas são projetados para contextos específicos, como o cultivo de cactos ou cogumelos, enquanto outros se concentram em monitorar e controlar apenas um conjunto limitado de variáveis ambientais.

O **IrrigaFlow** propõe uma solução mais abrangente e flexível, estruturada em uma arquitetura de três camadas – Módulo IoT, Borda da Rede e Nuvem – que permite monitoramento em tempo real e execução automatizada de ações baseadas em dados ambientais mais complexos, como evapotranspiração, precipitação, fase de desenvolvimento da cultura e textura do solo. Diferentemente dos estudos apresentados, a lógica fuzzy no **IrrigaFlow** é aplicada na camada de Borda, possibilitando o processamento local das informações e permitindo que o sistema se adapte rapidamente a variações ambientais, sem depender de uma conexão constante com

a nuvem. Essa descentralização melhora a resiliência do sistema, um ponto que grande parte dos estudos revisados não abordam em profundidade.

O Módulo IoT, projetado para simular a operação de sensores e atuadores em tempo real, é um dos principais elementos do **IrrigaFlow**. Essa simulação permite que testes sejam conduzidos de forma prática e acessível, criando uma base sólida para ajustes e refinamentos antes da implementação em campo. Essa abordagem é especialmente valiosa em contextos acadêmicos e em projetos que enfrentam limitações de acesso a equipamentos físicos, proporcionando um ambiente de experimentação que reduz custos e riscos.

A Borda da Rede, que integra o simulador climático, é outro diferencial do **IrrigaFlow**. Esse simulador permite a análise de variáveis ambientais importantes, como temperatura, umidade, precipitação, velocidade do vento e radiação solar, em um ambiente controlado. A evapotranspiração é calculada com base nesses dados, aplicando a fórmula de Penman-Monteith, um modelo amplamente reconhecido para estimar a demanda hídrica das culturas. Embora esse método tenha sido formalizado há décadas pela FAO (Food and Agriculture Organization of the United Nations – Organização das Nações Unidas para a Alimentação e Agricultura) em seu relatório técnico FAO-56 [1], ele continua sendo a principal referência para cálculos de evapotranspiração, sendo amplamente adotado na literatura científica e em aplicações práticas.

Essa capacidade de modelar um conjunto completo de variáveis ambientais e integrá-las ao sistema de lógica *fuzzy* torna o IrrigaFlow mais abrangente e adaptável às necessidades específicas de diferentes cenários agrícolas. Embora trabalhos como os de [2] e [14] já tenham sido testados em campo, a presença do simulador climático no IrrigaFlow facilita a experimentação e o aprimoramento do sistema antes da aplicação prática, tornando-o uma plataforma acessível e adaptável para outros pesquisadores que desejam explorar soluções semelhantes.

Portanto, o **IrrigaFlow** se posiciona como uma solução que não só complementa os estudos existentes, mas também oferece uma plataforma modular e flexível. Essa estrutura permite uma experimentação robusta e uma adaptação eficiente em diferentes cenários, promovendo avanços na pesquisa e desenvolvimento de sistemas de irrigação inteligentes e sustentáveis.

4 Desenvolvimento e Estrutura da Solução Proposta

A arquitetura de sistemas do **IrrigaFlow** foi desenvolvida seguindo uma abordagem de pesquisa qualitativa com uma posição interpretativa, explorando e descrevendo os fenômenos relacionados à automação da irrigação. O desenvolvimento baseou-se em uma combinação de estudo de caso e simulação como métodos de pesquisa, com coleta de dados a partir de simulações controladas e análise por meio de observação e testes práticos. Esse rigor metodológico permite monitorar e gerenciar os sistemas de irrigação em tempo real, oferecendo uma ampla gama de funcionalidades para otimizar o uso da água e adaptar as operações às necessidades específicas de cada cultura.

Nesta seção, apresenta-se uma visão geral da arquitetura proposta. Inicialmente, são detalhados os requisitos da solução, seguidos pela descrição da arquitetura desenvolvida. Em seguida, são fornecidos detalhes sobre o desenvolvimento e a implementação dos componentes. Por fim, são discutidos as considerações gerais sobre o desenvolvimento da arquitetura proposta.

4.1 Requisitos

O IrrigaFlow deve monitorar continuamente dados ambientais, como evapotranspiração, precipitação, fase de desenvolvimento da cultura e textura do solo. Essas informações garantem que a irrigação seja ajustada dinamicamente com base nas condições atuais, otimizando o uso da água.

O sistema deve processar os dados ambientais em tempo real, utilizando um modelo de lógica *fuzzy* para calcular automaticamente o tempo de irrigação e o volume de água necessários. Isso permite uma adaptação precisa às necessidades específicas da cultura e do ambiente.

A execução da irrigação deve ser totalmente automatizada, incluindo o controle de abertura e fechamento de válvulas, além da aplicação do volume de água conforme as instruções geradas pela análise de dados. A comunicação eficiente entre as camadas do sistema assegura a coordenação entre controle de irrigação, processamento de dados e interface de usuário.

4.2 Arquitetura do Sistema

A arquitetura do **IrrigaFlow**, Figura 2, foi projetada para oferecer uma solução de irrigação personalizada e eficiente, integrando dispositivos IoT e lógica *fuzzy* para uma gestão otimizada da água. A estrutura é dividida em três camadas: Módulo IoT, Borda da Rede e Nuvem, cada uma desempenhando um papel no funcionamento do sistema.

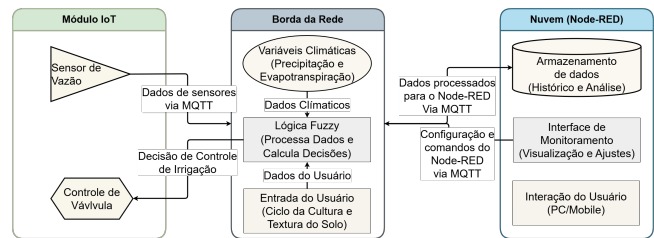


Figura 2: Diagrama da Arquitetura do Sistema IrrigaFlow

A primeira camada, **Módulo IoT**, é responsável por monitorar o fluxo e controlar a operação das válvulas, utilizando sensores e atuadores distribuídos pelo campo. As medições de vazão geram dados sobre o volume de água utilizado, os quais são enviados para a Borda da Rede. O controle das válvulas é ajustado automaticamente com base nas instruções da lógica *fuzzy* permitindo modificações na irrigação conforme necessário.

A segunda camada, a **Borda da Rede**, realiza o processamento local dos dados recebidos do Módulo IoT e aplica a lógica *fuzzy* para determinar as ações de irrigação. A Borda da Rede integra dados do campo e informações fornecidas pelo usuário, como a fase de desenvolvimento da cultura e a textura do solo, além de incorporar dados climáticos, como evapotranspiração e precipitação.

O uso de lógica *fuzzy* permite que a Borda da Rede tome decisões adaptativas com base nas condições variáveis do ambiente, ajustando de forma automática o tempo e o volume de irrigação. Essa capacidade de processamento local reduz a necessidade de envio de grandes volumes de dados para a Nuvem, diminuindo a latência e aumentando a eficiência do sistema. Além disso, a Borda atua como um ponto intermediário que garante a comunicação entre o Módulo IoT e a Nuvem, mantendo a operação contínua mesmo em condições de conectividade limitada.

Por fim, a **Nuvem** armazena dados históricos e fornece uma interface de monitoramento e controle remoto via Node-RED. Essa camada permite ajustes nos parâmetros de irrigação e exibe relatórios detalhados, oferecendo uma visão integrada do sistema. A Nuvem complementa o processamento da Borda, funcionando como uma plataforma para análise contínua e suporte a decisões estratégicas, permitindo identificar padrões de uso de água e oportunidades de otimização na gestão hídrica.

Essa arquitetura integrada assegura uma operação eficiente, combinando o processamento rápido na Borda com as capacidades de armazenamento e análise da Nuvem, resultando em uma solução flexível e eficaz para diferentes cenários de cultivo.

4.3 Desenvolvimento e Implementação

Esta seção apresenta os componentes técnicos e metodológicos aplicados no desenvolvimento do sistema de irrigação automatizado baseado em lógica *fuzzy*, abordando o ambiente de simulação, a implementação dos módulos e integração entre as camadas.

Para analisar a arquitetura antes da aplicação prática, foi desenvolvido um ambiente de simulação utilizando *scripts* em Python. Esse ambiente simula o funcionamento do Módulo IoT, da Borda da Rede e da Nuvem em um contexto controlado.

A seguir, descrevem-se as implementações dos principais componentes que compõem o sistema, detalhando suas funcionalidades e a forma como cada módulo contribui para o processo de automação da irrigação.

4.3.1 Módulo IoT. Desenvolvido em Python, é um simulador projetado para realizar medições virtuais de vazão de água e controlar as válvulas, transmitindo dados para a Borda da Rede por meio do protocolo MQTT. A Figura 3 apresenta um diagrama que organiza e ilustra os componentes principais desse módulo e sua interação, destacando a estrutura planejada para monitorar e automatizar o fluxo de água. Cada componente desempenha uma função específica no controle e monitoramento do processo de irrigação.

O **Controlador de Lógica de Irrigação** gerencia o fluxo de água ao receber comandos MQTT para iniciar ou finalizar a irrigação. Quando acionado, o sistema começa a modelagem, medindo o volume de água a uma vazão simulada de 1 litro a cada 15 segundos, permitindo o cálculo do volume acumulado durante o processo. A medição é encerrada ao receber o comando de fechamento, momento em que o sistema registra o volume total de água utilizado. O controle da irrigação também envia atualizações à Borda da Rede para informar o status do processo.

Para calcular o volume de água, o **Monitor de Dados de Vazão** usa a vazão configurada e o tempo de abertura do controlador. Ao iniciar a medição, registra o momento inicial e calcula o volume

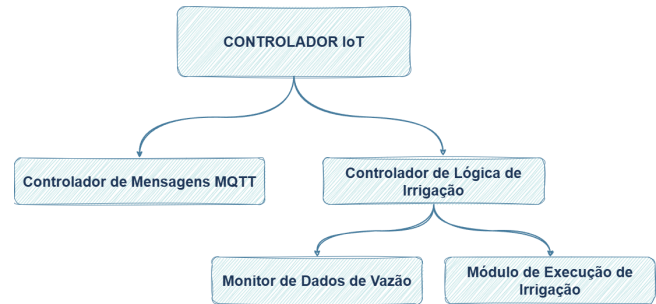


Figura 3: Diagrama do Módulo IoT

acumulado de acordo com o tempo decorrido, permitindo uma simulação precisa da quantidade de água usada. Além disso, o monitor fornece o volume atual a qualquer momento e reinicia a contagem ao término de cada sessão.

Encerrando o fluxo de água, o **Módulo de Execução de Irrigação** controla a abertura e o fechamento do fluxo. Ao receber um comando para iniciar, verifica se o fluxo está inativo e, se necessário, altera o estado para “aberto”, confirmando a ação. Caso contrário, exibe uma notificação. Da mesma forma, ao receber um comando de encerramento, altera o estado para “fechado” ou exibe uma mensagem informativa caso o fluxo já esteja fechado.

A comunicação entre o Módulo IoT e a Borda da Rede ocorre por meio do **Controlador de Mensagens MQTT**, que permite a troca bidirecional de mensagens. Esse controlador conecta-se ao broker MQTT e gerencia eventos de conexão e recepção de mensagens. Uma vez conectado, publica uma mensagem de confirmação de status online ou exibe um código de erro em caso de falha. Com capacidade para inscrever-se em tópicos específicos, ele também processa mensagens recebidas com comandos externos, respondendo a solicitações como início ou fim da irrigação e garantindo que o Módulo IoT permaneça conectado de forma eficiente com a Borda da Rede para um controle remoto eficaz do sistema.

4.3.2 Borda da Rede. Esta camada processa as informações coletadas pelo Módulo IoT e aplica a lógica *fuzzy* para tomar decisões de irrigação, como ilustrado na Figura 4. O sistema *fuzzy* recebe variáveis de entrada, como a fase de desenvolvimento da cultura e condições climáticas, determina o tempo e o intervalo de irrigação. A borda é programada em Python, utilizando bibliotecas como *scikit-fuzzy*.

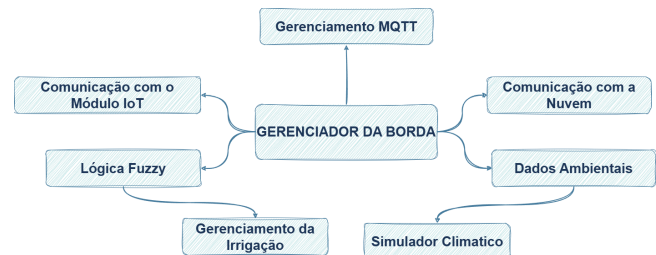


Figura 4: Diagrama da Borda da Rede

O **Gerenciador da Borda** é o componente central da camada da Borda da Rede, responsável pela coordenação das atividades de irrigação e pela comunicação com outros módulos do sistema. Esse gerenciador monitora o ciclo de irrigação, processa dados ambientais, aplica lógica *fuzzy* para determinar o tempo e o intervalo de irrigação e envia atualizações para a Nuvem e para o Módulo IoT.

Inicialmente, o gerenciador estabelece uma conexão com o broker MQTT, permitindo a troca de mensagens. Além disso, configura interfaces de comunicação tanto com a Nuvem quanto com o Módulo IoT, de forma a receber dados e enviar comandos. Nesse estágio, o sistema aguarda a definição de parâmetros essenciais, como o ciclo total e a textura do solo, fundamentais para o controle da irrigação.

Durante o ciclo de irrigação, o gerenciador coleta dados ambientais que são analisados por um sistema de lógica *fuzzy*. Essa análise resulta no cálculo do tempo ideal de irrigação e no intervalo entre as sessões, garantindo uma adaptação contínua às condições ambientais e ao progresso do ciclo. Qualquer mudança detectada leva a ajustes automáticos no plano de irrigação.

No que se refere ao controle e envio de dados, o gerenciador transmite para a Nuvem informações sobre o status e os parâmetros de irrigação, incluindo o volume de água utilizado e a condição atual do solo. Isso permite que o usuário acompanhe o processo em tempo real. Além disso, o gerenciador comunica-se com o Módulo IoT, enviando dados ambientais e acionando o controlador de fluxo conforme necessário.

A operação do gerenciador é contínua, monitorando e ajustando o fluxo de irrigação até o encerramento do ciclo da cultura. Caso necessário, o processo pode ser interrompido manualmente, com o sistema exibindo mensagens de status para indicar qualquer mudança no ciclo.

O **Gerenciamento de Borda** é, portanto, a base para a coordenação inteligente e automatizada da irrigação, garantindo que o sistema opere de maneira eficiente.

O **Módulo de Lógica Fuzzy** é responsável por processar dados ambientais e informações sobre o desenvolvimento da cultura para calcular, por meio de um sistema de inferência *fuzzy*, o tempo e o intervalo ideais para irrigação. Esse sistema permite que o controle de irrigação se ajuste dinamicamente, considerando as condições específicas do solo e de clima.

Primeiramente, o módulo configura variáveis de entrada como a fase de desenvolvimento da planta, a textura do solo, a evapotranspiração e a precipitação. Cada uma dessas variáveis é associada a termos *fuzzy* (por exemplo, “alta”, “média”, “baixa”) que representam categorias amplas e permitem que o sistema lide com incertezas e variações. As variáveis de saída, como o tempo de acionamento do sistema de irrigação e o intervalo entre as sessões de irrigação, também são configuradas com termos *fuzzy* como “curto” e “longo”.

Com as variáveis devidamente configuradas, o módulo estabelece um conjunto de regras que orientam a resposta do sistema diante de diferentes combinações de condições ambientais e do estado da planta. Por exemplo, se a evapotranspiração for classificada como “alta” e a precipitação como “muito baixa”, o sistema determina um tempo de acionamento “longo”. Essas regras são formuladas com base em cenários típicos de irrigação para garantir que as necessidades de água sejam supridas de forma eficiente.

Após a configuração das variáveis e das regras, o sistema *fuzzy* é criado e pronto para o processamento. Durante a execução, os valores atuais de fase de desenvolvimento da cultura, textura do solo, evapotranspiração e precipitação são inseridos no sistema. Este então processa os dados e calcula o tempo ideal de irrigação e o intervalo apropriado, retornando valores específicos. O módulo permite que o sistema de irrigação seja ajustado em tempo real, assegurando a aplicação da água e a adaptação às condições atuais da cultura e do ambiente.

O **Gerenciamento da Irrigação** é o módulo responsável por coordenar o acionamento e desligamento do fluxo de irrigação com base nos parâmetros calculados e nas condições do sistema. Ele assegura que as operações de irrigação ocorram no momento certo e por um período apropriado, seguindo as diretrizes estabelecidas pela lógica *fuzzy* e os dados ambientais.

Quando a irrigação deve começar, o módulo envia um comando para abrir o fluxo de água e inicia uma contagem de tempo para controlar a duração da irrigação, definida em minutos. Durante esse período, o status do sistema é atualizado e transmitido para a nuvem, informando que a irrigação está em andamento e permitindo um acompanhamento em tempo real.

Ao final do tempo de acionamento, o sistema desliga o fluxo de água e registra o momento da próxima irrigação. O intervalo é calculado e uma mensagem de status é enviada para a nuvem, informando quando a próxima irrigação ocorrerá. Em seguida, o sistema entra em um estado de espera até o início do próximo ciclo, garantindo um planejamento contínuo e automatizado.

O módulo monitora constantemente o estado do fluxo para verificar se o tempo atual coincide com os horários agendados para iniciar ou finalizar a irrigação. Essa supervisão contínua assegura precisão das operações e mantém um histórico claro das atividades realizadas.

O **Gerenciador da Irrigação** desempenha um papel essencial para garantir que o processo de irrigação seja automatizado, eficiente e sincronizado com os dados fornecidos pelos outros do sistema. Ele viabiliza um controle detalhado e uma comunicação contínua com a camada de nuvem, assegurando uma irrigação precisa e adaptativa às necessidades da cultura.

O **Módulo de Dados Ambientais** tem como função fornecer as informações climáticas e do solo necessárias para a gestão da irrigação. Utiliza um simulador climático, ele cria um histórico de dados e disponibiliza informações em tempo real sobre o ambiente, ajustando o ciclo de irrigação conforme as condições ambientais atuais.

Na inicialização, o módulo recebe do usuário, via nuvem, a definição do ciclo de crescimento da cultura em dias e a textura do solo em porcentagem. Com base nessas informações, o módulo inicia a coleta contínua de dados climáticos fornecidos por um simulador, que envia informações da evapotranspiração e precipitação, ao longo do ciclo. Esse fluxo contínuo de dados permite que o sistema se ajuste automaticamente às mudanças ambientais, recalculando as condições conforme necessário. Dessa forma, a abordagem garante que as necessidades da cultura sejam atendidas de forma adaptada às variações climáticas em tempo real.

Durante cada ciclo de irrigação, o módulo realiza a coleta dos dados ambientais mais recentes, incluindo a evapotranspiração e a precipitação no momento da consulta. Além disso, o módulo inclui

funções que formatam os dados de tempo em minutos e horas, tornando mais compreensível as informações de tempo de irrigação e intervalo para a integração com outras partes do sistema. Essa formatação facilita a comunicação e a operação entre diferentes módulos do sistema de irrigação.

O **Módulo de Dados Ambientais** fornece, portanto, informações essenciais para o controle adaptativo de irrigação, permitindo uma resposta eficiente às variações do ambiente e assegurando que o sistema funcione em harmonia com as condições climáticas simuladas e observada ao longo do ciclo da cultura.

O **Simulador Climático** cria um ambiente virtual que reproduz condições climáticas, permitindo a previsão e o ajuste das estratégias de irrigação com base em variáveis como precipitação, temperatura, umidade, velocidade do vento e radiação solar. Para gerar esses dados, o simulador utiliza algoritmos baseados em distribuições estatísticas e cálculos matemáticos que imitam as condições ambientais.

Os dados de temperatura, umidade, vento e radiação solar são gerados de maneira aleatória, porém controlada, utilizando distribuições estatísticas. Por exemplo, a temperatura é simulada com uma distribuição normal que tem média de 28°C e desvio padrão de 3°C, representando variações típicas do clima. A umidade, a velocidade do vento e a radiação solar são criadas com variações uniformes dentro de intervalos predefinidos, garantindo a variabilidade dos dados ao longo do tempo simulado.

A simulação de precipitação é realizada de forma horária e leva em conta a umidade relativa e o mês do ano. Durante os meses mais chuvosos de março a julho, a probabilidade de chuva aumenta em função da umidade, e a quantidade de precipitação por hora é modelada por uma distribuição de Poisson com média de 0,5mm. Nos meses secos, a chance de chuva diminui, e, quando ocorre, a quantidade é menor, refletindo as condições climáticas típicas.

A Evapotranspiração é calculada com base nas condições ambientais previamente simuladas, como temperatura, umidade, vento e radiação solar. Para isso, é aplicada a fórmula de Penman-Monteith, amplamente reconhecida e utilizada na ciência ambiental para estimativas dessa variável.

Dessa forma, o simulador produz um conjunto de dados climáticos estruturados, possibilitando a utilização dessas informações para prever e ajustar o processo de irrigação conforme as condições simuladas e as variáveis ambientais em constante mudança.

O **Gerenciamento MQTT** na camada de Borda da Rede é responsável por coordenar a comunicação entre o computador de borda, o Módulo IoT e a Nuvem. Esse componente gerencia a troca de mensagens em tempo real, assegurando uma comunicação contínua entre as partes envolvidas no processo de monitoramento e controle da irrigação.

Na etapa de conexão e configuração inicial, o módulo se conecta ao broker MQTT e define eventos para o tratamento de mensagens recebidas e o gerenciamento da conexão. Uma vez estabelecida a conexão, o módulo se inscreve em tópicos específicos, como o tópico do Módulo IoT e o tópico de configuração da Nuvem, permitindo o recebimento de dados e comandos em tempo real.

O módulo também permite o envio de mensagens para tópicos MQTT conforme necessário, como atualizações de status e dados de controle. As mensagens recebidas são processadas para identificar seu conteúdo e orientar as ações subsequentes. Por exemplo, o

módulo pode receber informações do Módulo IoT, como o volume total de água utilizado e o status do fluxo de água, ou dados de configuração enviados pela Nuvem, como o ciclo total da cultura e a textura do solo.

Quando uma mensagem é recebida, o sistema interpreta e realiza ações baseadas nas informações contidas. Se a mensagem confirmar a conexão do Módulo IoT, o sistema notifica a Nuvem sobre o status atualizado. Para mensagens de configuração, o módulo ajusta parâmetros essenciais, como o ciclo total da cultura e a textura do solo, garantindo que o sistema de irrigação seja configurado de acordo com as necessidades atuais.

O acompanhamento da conexão com o Módulo IoT é outra função importante. Caso ocorra uma desconexão, o sistema inicia tentativas periódicas de restabelecimento, enviando mensagens de status para a Nuvem a fim de informar o progresso e a situação da conexão.

Esse módulo centraliza e facilita a comunicação entre o Computador de Borda, o Módulo IoT e a Nuvem, permitindo o monitoramento contínuo e uma resposta ágil a quaisquer mudanças no sistema de irrigação.

O **Componente de Comunicação com o Módulo IoT** na camada de Borda da Rede gerencia a troca de informações e comandos com o Módulo IoT, garantindo um controle eficiente do sistema de irrigação.

Uma das principais funcionalidades é o envio de comandos de controle. O módulo permite que comandos sejam enviados para ligar e desligar o controlador de fluxo de água. Quando a irrigação precisa ser ativada, o comando “liga_valvula” é enviado, e, da mesma forma, quando a irrigação deve ser interrompida, o comando “desligar_valvula” é transmitido, acompanhado de uma mensagem de status para a Nuvem informando o desligamento e o volume total de água utilizado.

Além disso, o módulo é capaz de atualizar dados ambientais ao enviar informações para o Módulo IoT, como a fase de desenvolvimento da cultura, textura do solo, evapotranspiração e precipitação. Esses dados são compilados e transmitidos para que o Módulo IoT possa processá-los e ajustar a operação conforme necessário.

O componente também lida com o processamento de mensagens de volume, recebendo dados que contêm o volume total de água utilizado durante as sessões de irrigação. Essas mensagens são decodificadas e o volume é atualizado internamente para permitir o monitoramento e registro do status do sistema.

Para garantir a robustez e a confiabilidade da comunicação, o módulo possui mecanismos de gestão de erros. Ele inclui tratamentos de exceções que capturam e relatam falhas na comunicação, facilitando a identificação e correção de problemas e assegurando a continuidade do sistema.

O **Componente de Comunicação com o Módulo IoT** assegura a troca de comandos e dados entre a camada da Borda e o Módulo IoT, possibilitando uma operação automatizada e uma integração fluida com a Nuvem, para monitoramento e controle em tempo real.

O **Componente de Comunicação com a Nuvem** na camada de Borda da Rede facilita a troca de dados com sistemas baseados em nuvem, como o Node-RED, desempenhando um papel na centralização e visualização das informações do sistema de irrigação.

Uma das principais funções desse componente é o envio de dados ambientais e operacionais. Ele reúne informações como o ciclo de irrigação, tempo recomendado para acionamento, intervalo entre

as sessões de irrigação, fase de desenvolvimento da cultura, textura do solo, níveis de evapotranspiração e precipitação. Esses dados são formatados e transmitidos ao Node-RED para que possam ser exibidos e analisados, permitindo uma visão abrangente do sistema.

Além disso, o componente envia mensagens de status, relatando eventos importantes como a ativação ou desativação do controlador de fluxo de água e, quando aplicável, o volume total de água utilizado. Esse monitoramento em tempo real permite que os operadores acompanhem as condições e ações do sistema à medida que ocorrem.

Para assegurar a confiabilidade da comunicação, o componente possui um sistema de tratamento de exceções. Ele captura e reporta falhas na transmissão de dados, facilitando a identificação de problemas e garantindo que a conexão com a nuvem seja estável.

O **Componente de Comunicação com a Nuvem** é fundamental para a integração da camada da Borda com plataformas de visualização e análise de dados, proporcionando um monitoramento detalhado do sistema de irrigação e suportando decisões baseadas em dados em tempo real.

4.3.3 Nuvem. A automatização e controle na nuvem para o sistema de irrigação são realizados por meio de um fluxo de trabalho definido no Node-RED, que permite a coleta, processamento, e visualização dos dados e gerenciamento remoto.

O processo se inicia com a recepção de dados enviados pelo sistema de borda via protocolo MQTT, conforme ilustrado na Figura 5. Esses dados incluem tanto informações de status quanto dados específicos do ciclo de irrigação, recebidos em dois tópicos distintos: um para mensagens gerais de status e outro para dados detalhados do ciclo de irrigação. No node-RED, os nós de recepção MQTT captam essas informações e as encaminham para processamento.

Após a recepção, os dados são processados por uma função que os categoriza conforme o tipo de mensagem, distinguindo entre dados de status e do ciclo de irrigação, conforme a Figura 5. Essa organização é fundamental para preparar as informações para armazenamento no banco de dados e posterior exibição no *dashboard*. Os dados processados são então armazenados em uma coleção específica no MongoDB, garantindo que todas as informações sejam persistidas para consultas futuras e análise históricas. O MongoDB é escolhido devido à sua flexibilidade e capacidade de lidar com grandes volumes de dados.



Figura 5: Nós de recepção, processamento e armazenamento de dados no Node-RED

Para a atualização da interface do *dashboard*, um nó de injeção no Node-RED é configurado para recuperar periodicamente os dados armazenados no MongoDB. Esses dados são processados para extrair informações relevantes sobre o sistema de irrigação, como volume total de água utilizado, fase de desenvolvimento das culturas, textura do solo, evapotranspiração e precipitação. Em seguida, os dados são formatados e enviados aos nós de interface do usuário, que atualizam os componentes do *dashboard*, como

mostrado na Figura 6. Isso permite que os operadores monitorem o sistema em tempo real e tomem decisões informadas.

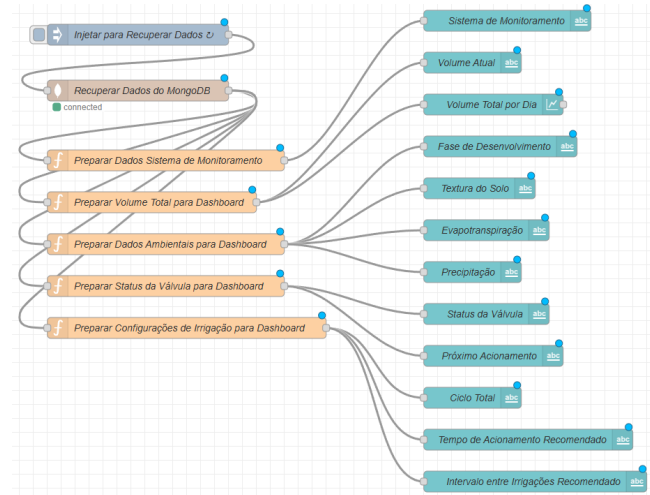


Figura 6: Nós de atualização do dashboard no Node-RED

A interface do *dashboard* foi projetada para apresentar as informações de forma clara e organizada, exibindo dados detalhados sobre o sistema de irrigação. Isto facilita o gerenciamento eficiente e visualização em tempo real das operações.

Além disso, o Node-RED inclui uma interface que permite aos usuários ajustar parâmetros do sistema de irrigação, como o ciclo total de irrigação e a textura do solo. Na interface, os usuários podem inserir esses valores em campos específicos e confirmar as alterações por meio de um botão de atualização. As configurações inseridas são enviadas de volta ao sistema de borda via MQTT, possibilitando ajustes dinâmicos das operações de irrigação conforme as necessidades.

Dessa forma, a estrutura modular do **IrrigaFlow** proporciona uma solução eficiente para a automação da irrigação, integrando tecnologias IoT, lógica fuzzy e uma arquitetura distribuída confiável. O desenvolvimento detalhado e a integração entre as camadas de IoT, Borda da Rede e Nuvem garantem a otimização do uso de recursos hídricos, oferecendo controle e monitoramento em tempo real. Essa abordagem possibilita uma adaptação dinâmica às condições ambientais e às necessidades específicas das culturas, resultando em um sistema de irrigação inteligente e sustentável.

5 Conclusão

A necessidade de práticas agrícolas eficientes e sustentáveis tem impulsionado o desenvolvimento de tecnologias para otimizar o uso de recursos hídricos. O **IrrigaFlow**, com sua arquitetura modular baseada em IoT, lógica fuzzy e processamento em camadas (IoT, Borda da Rede e Nuvem), foi projetado com o objetivo de oferecer uma solução de controle automatizado de irrigação, integrando coleta de dados ambientais, análise em tempo real e controle adaptativo. A implementação do sistema em ambiente simulado mostrou-se funcional para coordenar e ajustar a irrigação conforme as variáveis climáticas e as condições das culturas, destacando-se

como uma abordagem promissora para o gerenciamento eficiente da água na agricultura.

O desenvolvimento do **IrrigaFlow** trouxe várias contribuições importantes para o campo da gestão de irrigação inteligente. A estrutura em três camadas – Módulo IoT, Borda da Rede e Nuvem – permite a coleta, análise e monitoramento de dados em tempo real, com processamento descentralizado que aumenta a resiliência do sistema. A aplicação de lógica *fuzzy* para a tomada de decisões adaptativas com base em múltiplas variáveis ambientais proporciona uma maior precisão no controle da irrigação, ajustando automaticamente o volume e o tempo de irrigação às necessidades específicas da cultura e do clima.

O Módulo IoT e o simulador climático, oferece um ambiente de testes prático para validação e ajuste da lógica de controle, permitindo que outros pesquisadores e desenvolvedores realizem experimentos sem a necessidade imediata de recursos físicos em campo. Além disso, a interface de controle e monitoramento via Node-RED fornece uma visualização intuitiva e abrangente das operações do sistema, possibilitando ajustes de parâmetros e acompanhamento remoto em tempo real.

O aprimoramento do **IrrigaFlow** inclui sua implementação em campo para validação em ambientes reais, a integração com machine learning para prever necessidades hídricas com base em dados históricos e a expansão da escalabilidade para suportar múltiplos módulos IoT. Além disso, a análise do impacto ambiental, considerando a redução do consumo de água e a eficiência energética, é essencial para consolidá-lo como uma solução sustentável e inovadora na agricultura.

Referências

- [1] R. G. Allen, L. S. Pereira, D. Raes, and M. Smith. 1998. *Crop Evapotranspiration: Guidelines for Computing Crop Water Requirements*. FAO Irrigation and Drainage Paper, Vol. 56. FAO – Food and Agriculture Organization of the United Nations, Rome, Italy. <http://www.fao.org/3/X0490E/X0490E00.htm>
- [2] Eeshan Amiy, Prabhakar Kumar Upadhyay, and Rishabh Raj. 2022. An IOT based Smart Irrigation System Using Sugeno's Fuzzy Inference & Solar Power. In *Proceedings of the 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCTST 2021)*. Institute of Electrical and Electronics Engineers Inc., India, 125–129. <https://doi.org/10.1109/ICRTCTST54752.2022.9781942> Conference name: 4th International Conference on Recent Trends in Computer Science and Technology, ICRTCTST 2021; Conference date: 11 February 2022 through 12 February 2022; Conference code: 179571.
- [3] M.A.M. Ariffin, M.I. Ramli, Z. Zainol, M.N.M. Amin, M. Ismail, R. Adnan, N.D. Ahmad, N. Husain, and N. Jamil. 2021. Enhanced iot-based climate control for oyster mushroom cultivation using fuzzy logic approach and nodemcu micro-controller. *Pertanika Journal of Science and Technology* 29, 4 (2021), 2863–2885. <https://doi.org/10.47836/PJST.29.4.34> cited By 0.
- [4] Sandeep Bhowmik. 2017. *Cloud Computing*. Cambridge University Press, Cambridge, United Kingdom.
- [5] Luiz Bittencourt, Roger Immich, Rizos Sakellariou, Nelson Fonseca, Edmundo Madeira, Marília Curado, Leandro Villas, Luiz DaSilva, Craig Lee, and Omer Rana. 2018. The Internet of Things, Fog and Cloud continuum: Integration and challenges. *Internet of Things* 3-4 (2018), 134 – 155. <https://doi.org/10.1016/j.iot.2018.09.005>
- [6] Dirceu Brasil Vieira and Dirceu D'Alkmin Telles. 2001. Panorama da irrigação no Brasil: evolução, tendências, novas legislações. *Ingeniería del agua* 8, 2 (2001), 207–217.
- [7] Jenny Carter, Francisco Chiclana, Arjab Singh Khuman, and Tianhua Chen (Eds.). 2021. *Fuzzy Logic: Recent Applications and Developments*. Springer Nature Switzerland AG, Cham, Switzerland. <https://doi.org/10.1007/978-3-030-66474-9>
- [8] Confederação da Agricultura e Pecuária do Brasil – CNA. 2022. PIB DO AGRO-NEGÓCIO CRESCEU ABAIXO DAS PROJEÇÕES. <https://www.cepea.esalq.usp.br/br/pib-do-agronegocio-brasileiro.aspx>
- [9] Rachma Dianty, Rina Mardiaty, Edi Mulyana, and Dedi Supriadi. 2021. Design of Humidity Control with Automatic Drip Irrigation System Based on Fuzzy Logic Using Node-RED and MQTT on Cactus Plants. In *Proceedings of the 7th International Conference on Wireless and Telematics (ICWT 2021)*. Institute of Electrical and Electronics Engineers Inc. (IEEE), Indonesia, 1–6. <https://doi.org/10.1109/ICWT52862.2021.9678449>
- [10] Vladimir Dimitrov and Victor Korotkich. 2002. *Fuzzy logic: a framework for the new millennium* (1 ed.). Vol. 81. Springer Science & Business Media, Berlin, Germany.
- [11] Pedro F. do Prado, Maycon L. M. Peixoto, Marcelo C. Araújo, Eduardo S. Gama, Diogo M. Gonçalves, Matheus V. S. Silva, Roger Immich, Edmundo R. M. Madeira, and Luiz F. Bittencourt. 2021. *Mobile Edge Computing for Content Distribution and Mobility Support in Smart Cities*. Springer International Publishing, Cham, 473–500. https://doi.org/10.1007/978-3-030-69893-5_19
- [12] Taiji Hagino. 2021. *Practical Node-RED Programming: Learn powerful visual programming techniques and best practices for the web and IoT*. Packt Publishing Ltd, Birmingham, UK.
- [13] Rolando Herrero. 2023. *Practical Internet of Things Networking: Understanding IoT Layered Architecture*. Springer Nature Switzerland AG, Cham, Switzerland. <https://doi.org/10.1007/978-3-031-28443-4> Disponível em formato físico e eBook.
- [14] Fuseini S Ibrahim, Dominic Konditi, and Stephen Musyoki. 2018. Smart irrigation system using a fuzzy logic method. *International Journal of Engineering Research and Technology* 11, 9 (2018), 1417–1436.
- [15] R. Immich, E. Cerqueira, and M. Curado. 2014. Towards the enhancement of UAV video transmission with motion intensity awareness. In *2014 IFIP Wireless Days (WD)*. 1–7. <https://doi.org/10.1109/WD.2014.7020820>
- [16] R. Immich, E. Cerqueira, and M. Curado. 2016. Towards a QoE-driven mechanism for improved H.265 video delivery. In *Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. 1–8. <https://doi.org/10.1109/MedHocNet.2016.7528427>
- [17] Rahul Kar, Dac-Nhuong Le, Gunjan Mukherjee, Biswadip Basu Mallik, and Ashok Kumar Shaw (Eds.). 2023. *Fuzzy Logic Applications in Computer Science and Mathematics*. Scrivener Publishing LLC, John Wiley & Sons, Inc., Hoboken, NJ, USA.
- [18] Mohammed Laroui, Boubakr Nour, Hassine Mounsla, Moussa A Cherif, Hossam Afifi, and Mohsen Guizani. 2021. Edge and fog computing for IoT: A survey on current research activities & future directions. *Computer Communications* 180 (2021), 210–231.
- [19] Rodger Lea. 2023. Node-RED: Lecture 1 – A brief introduction to Node-RED. <http://noderedguide.com/nr-lecture-1/>.
- [20] James K Peckol. 2021. *Introduction to Fuzzy Logic* (1 ed.). John Wiley & Sons, Hoboken, NJ, USA.
- [21] Flavia Pisani, Fabiola de Oliveira, Eduardo S Gama, Roger Immich, Luiz F Bittencourt, and Edson Borin. 2020. Fog Computing on Constrained Devices: Paving the Way for the Future IoT. *Advances in Edge Computing: Massive Parallel Processing and Applications* 35 (2020), 22. <https://doi.org/10.3233/APC200003>
- [22] Kolla Bhanu Prakash (Ed.). 2021. *Internet of Things: From the Foundations to the Latest Frontiers in Research*. De Gruyter, Berlin/Boston. <https://www.degruyter.com>
- [23] Alexandru Radovici and Ioana Culic. 2022. *Getting Started with Secure Embedded Systems: Developing IoT Systems for micro:bit and Raspberry Pi Pico Using Rust and Tock*. Apress, Wylidori, Bucharest, Romania. <https://doi.org/10.1007/978-1-4842-7789-8> Printed on acid-free paper.
- [24] I.N. Rudy Hendrawan, L. Putu Yulyantari, G.A. Pradipta, and P. Bayu Starriawan. 2019. Fuzzy Based Internet of Things Irrigation System. In *Proceedings of the 2019 1st International Conference on Cybernetics and Intelligent System (ICORIS 2019)*. Institute of Electrical and Electronics Engineers (IEEE), Denpasar, Bali, Indonesia, 146–150. <https://doi.org/10.1109/ICORIS.2019.8874900> cited By 8.
- [25] Roberto Testezlaf. 2017. *Irrigação: Métodos, Sistemas e Aplicações*. Unicamp/FEAGRI, Campinas, SP. https://www.researchgate.net/publication/318197795_Irigacao_metodos_sistemas_e_aplicacoes E-book, distribuição gratuita.
- [26] The MathWorks, Inc. 2022. *Fuzzy Logic Toolbox User's Guide* (r2022a ed.). The MathWorks, Inc., Natick, MA, USA. <https://www.mathworks.com/help/fuzzy/> Acesso em: 2022-09-30.
- [27] United Nations. 2023. Sustainable Development Goals. <https://sdgs.un.org/goals> Acesso em: 3 set. 2024.
- [28] Sandhya Arora Urmila Shrawankar, Latesh Malik. 2021. *Cloud Computing Technologies for Smart Agriculture and Healthcare (Chapman & Hall/CRC Cloud Computing for Society 5.0)* (1 ed.). Chapman and Hall/CRC, Boca Raton, FL, USA. <http://gen.lib.rus.ec/book/index.php?md5=77A6966650CF33FAA9CC9CFF2C57F359>
- [29] Hsien-Chung Wu. 2023. *Mathematical Foundation of Fuzzy Sets*. Wiley, Hoboken, NJ, USA.
- [30] Zhenci Xu, Xiuzhi Chen, Jianguo Liu, Yu Zhang, Sophia Chau, Nishan Bhattarai, Ye Wang, Yingjie Li, Thomas Connor, and Yunkai Li. 2020. Impacts of irrigated agriculture on food–energy–water–CO₂ nexus across metacoupled systems. *Nature communications* 11, 1 (2020), 5837.