

Controlo da Rega

Nome Gabriel Noira (a31240)

Nome David Soares Vieira (a56550)

Resumo

O presente projeto tem como objetivo o desenvolvimento de um sistema de rega automática inteligente, baseado em tecnologias de Internet das Coisas (IoT), com vista à monitorização remota e controlo eficiente da rega de plantas. O sistema utiliza um microcontrolador ESP32, sensores ambientais e uma bomba de água, sendo controlado remotamente através de uma interface desenvolvida em Node-RED.

A solução implementada permite a leitura periódica de dados de temperatura, humidade do ar e humidade do solo, os quais são enviados via protocolo MQTT para o Node-RED, onde são processados e visualizados num dashboard interativo. Com base nos valores de humidade do solo, é tomada uma decisão automática sobre a ativação ou não da bomba de rega. O utilizador dispõe ainda de um botão para ativar manualmente o sistema, independentemente da decisão automática.

Para garantir a comunicação entre dispositivos, foi utilizado um broker MQTT público. Os dados recolhidos são armazenados numa base de dados InfluxDB, permitindo a visualização de histórico em forma de gráfico. A implementação contempla ainda a receção de comandos de controlo da rega e a atualização em tempo real do estado do sistema.

O sistema simula a leitura de dados através do ambiente de simulação Wokwi, permitindo testes funcionais sem a necessidade de hardware físico. A estrutura modular do projeto facilita futuras melhorias, como integração com previsões meteorológicas via API externa (IPMA), deteção de depósito de água vazio e mecanismos de encriptação para segurança dos dados.

Em suma, este projeto demonstra a aplicabilidade da IoT na agricultura de precisão, promovendo o uso eficiente da água e a automação de tarefas com base em dados em tempo real.

Palavras-chave

Internet das Coisas, Rega Automática, ESP32, MQTT, Monitorização Ambiental

1. Introdução

A crescente necessidade de otimização de recursos na agricultura tem impulsionado a adoção de tecnologias emergentes capazes de melhorar a eficiência e reduzir o desperdício, nomeadamente no que diz respeito à utilização da água. A Internet das Coisas (IoT) destaca-se como uma das abordagens mais promissoras, permitindo a recolha, análise e atuação sobre dados ambientais em tempo real.

Neste contexto, o presente projeto propõe o desenvolvimento de um sistema de rega automática, concebido para monitorizar e gerir remotamente as condições de uma planta, com recurso a sensores de temperatura, humidade do ar e humidade do solo. A automação do processo de rega visa garantir que a planta receba água apenas quando necessário, com base em dados precisos, promovendo assim a sustentabilidade e o uso racional da água.

O sistema é implementado com um microcontrolador ESP32, sensores ambientais e uma bomba de água simulada. A comunicação entre os dispositivos é realizada através do protocolo MQTT, utilizando um broker público, enquanto a lógica de decisão, visualização e controlo é gerida através do ambiente de programação visual Node-RED.

A interface desenvolvida permite ao utilizador acompanhar os dados recolhidos, acionar manualmente o sistema de rega e acompanhar o estado da bomba. Além disso, o sistema foi testado no ambiente de simulação Wokwi, o que permitiu validar a lógica de funcionamento sem a necessidade de hardware físico.

Este relatório descreve a conceção, implementação e resultados obtidos com a solução proposta, destacando a sua aplicabilidade prática e a possibilidade de evolução para um sistema mais robusto e completo, com integração de previsões meteorológicas, deteção do nível de água e armazenamento histórico de dados.

2. Materiais e Métodos

2.1 Materiais Utilizados

A implementação do sistema de rega automática foi realizada com recurso a componentes de hardware e software que permitiram a simulação e controlo do sistema em tempo real. A tabela seguinte resume os principais materiais utilizados:

Componente	Descrição
ESP32	Microcontrolador com Wi-Fi integrado para controlo e comunicação
Sensor DHT22	Sensor digital utilizado para medir temperatura e humidade do ar
Potenciómetro	Utilizado para simular um sensor de humidade do solo (entrada analógica)
LED	Representa a bomba de rega, ligado a um pino digital do ESP32
Broker MQTT	Broker público (broker.emqx.io) para troca de mensagens entre dispositivos
Node-RED	Plataforma de programação visual usada para lógica, controlo e dashboard
Wokwi	Simulador online de microcontroladores, usado para testes do sistema

2.2 Arquitetura do Sistema

O sistema segue uma arquitetura distribuída baseada em IoT, onde o ESP32 atua como nó de aquisição e execução, enviando os dados ambientais através do protocolo MQTT para o Node-RED. O Node-RED processa os dados recebidos, toma decisões com base nas leituras e envia comandos de controlo de rega para o ESP32. O dashboard do Node-RED permite ainda a intervenção manual e a visualização em tempo real das variáveis monitorizadas.

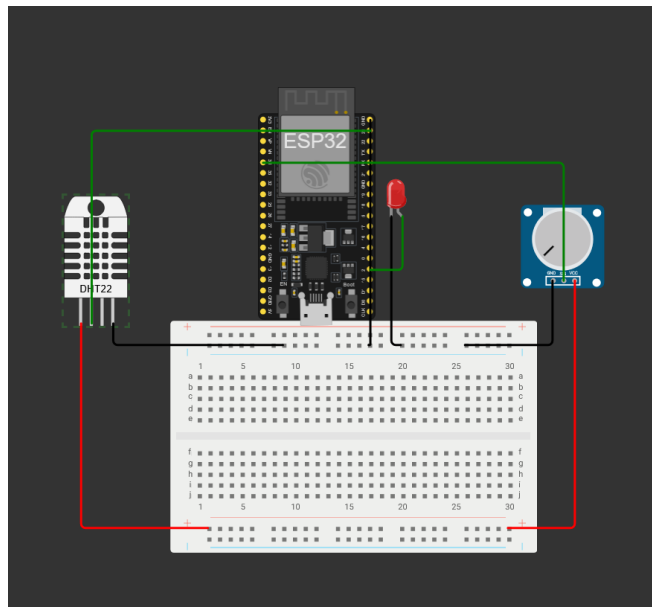
2.3 Estrutura dos Tópicos MQTT

Os seguintes tópicos foram utilizados para a comunicação entre o ESP32 e o Node-RED:

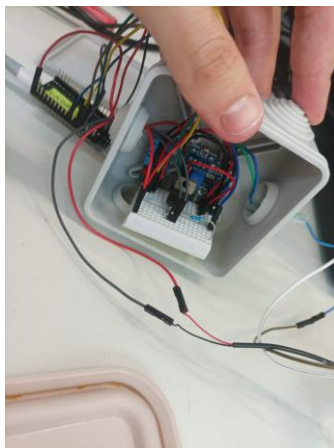
Tópico MQTT	Função
IPB/IoT/PW2/TurnoD/Group2/Temperatura	Publicação da temperatura ambiente
IPB/IoT/PW2/TurnoD/Group2/HumidadeAr	Publicação da humidade do ar
IPB/IoT/PW2/TurnoD/Group2/HumidadeSolo	Publicação da humidade do solo
IPB/IoT/PW2/TurnoD/Group2/ComandoRega	Subscrição do ESP para comando de rega
IPB/IoT/PW2/TurnoD/Group2/EstadoRega	Publicação do estado atual da bomba

2.4 Ambiente de Simulação e Ambiente Físico

Para a simulação e testes do sistema, foi utilizado o Wokwi, um ambiente online que permite emular microcontroladores como o ESP32 e uma variedade de sensores. Neste projeto, o sensor de temperatura e humidade foi simulado com o modelo DHT22, enquanto o sensor de humidade do solo foi simulado através de um potenciómetro. A bomba de água foi representada por um LED digital.



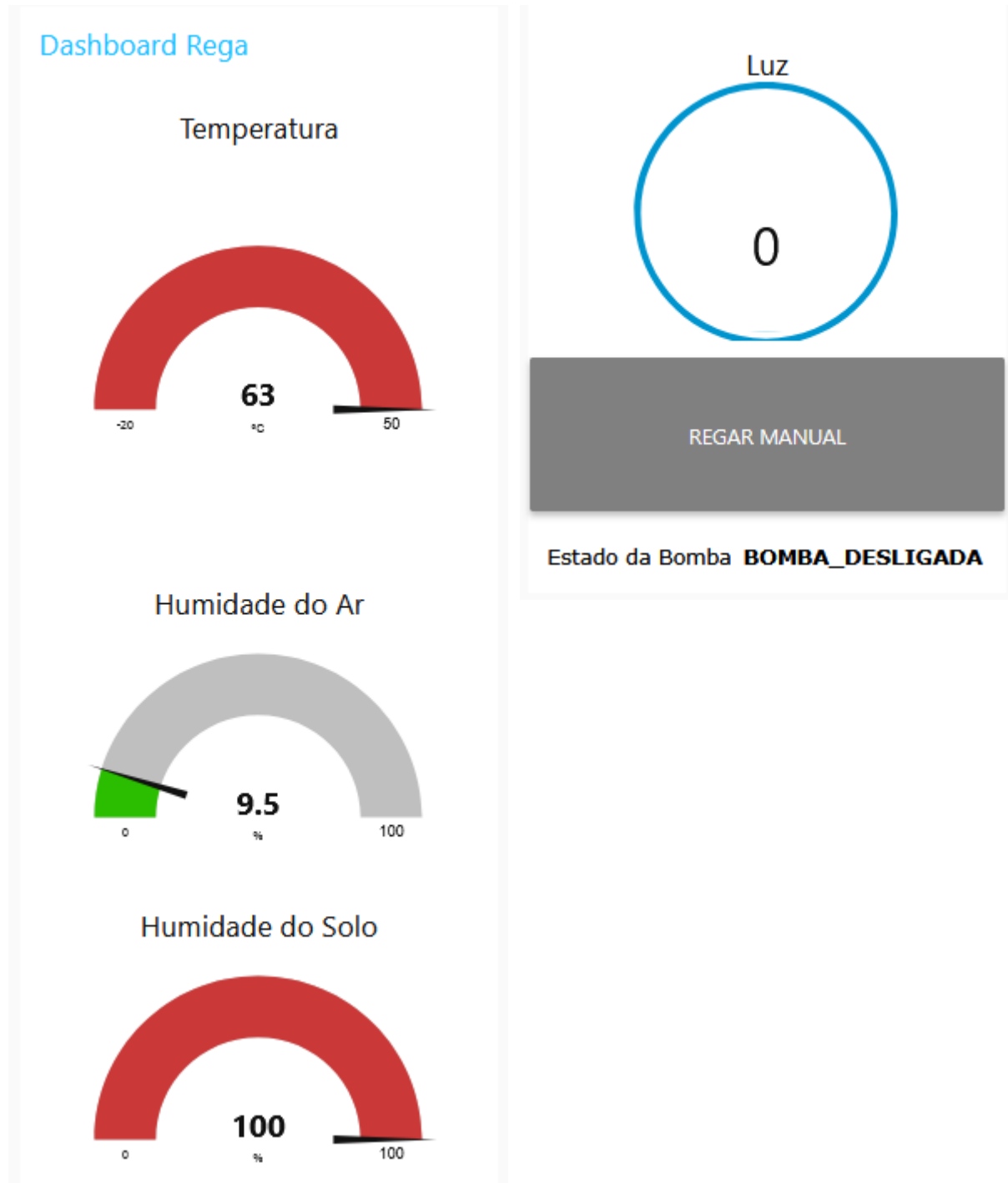
Complementarmente à simulação, foi também realizada uma montagem física do sistema com os mesmos componentes: uma placa ESP32 real, um sensor DHT22 físico, um sensor de humidade de solo capacitivo e uma bomba de água. Esta abordagem permitiu validar a funcionalidade do código em ambiente real, verificar a precisão dos sensores físicos e testar a atuação da bomba mediante os comandos recebidos por MQTT. A transição entre o ambiente de simulação e o ambiente físico foi realizada sem alterações significativas no código, o que evidencia a portabilidade e robustez da arquitetura implementada.



2.5 Dashboard Node-RED

No Node-RED foi construído um dashboard com os seguintes elementos:

- Medidores (gauges) de temperatura, humidade do ar e do solo;
- Indicador de estado da bomba;
- Botão de controlo manual da rega;
- Lógica de decisão automática com base no nível de humidade do solo.



3. Implementação

3.1 Código no ESP32

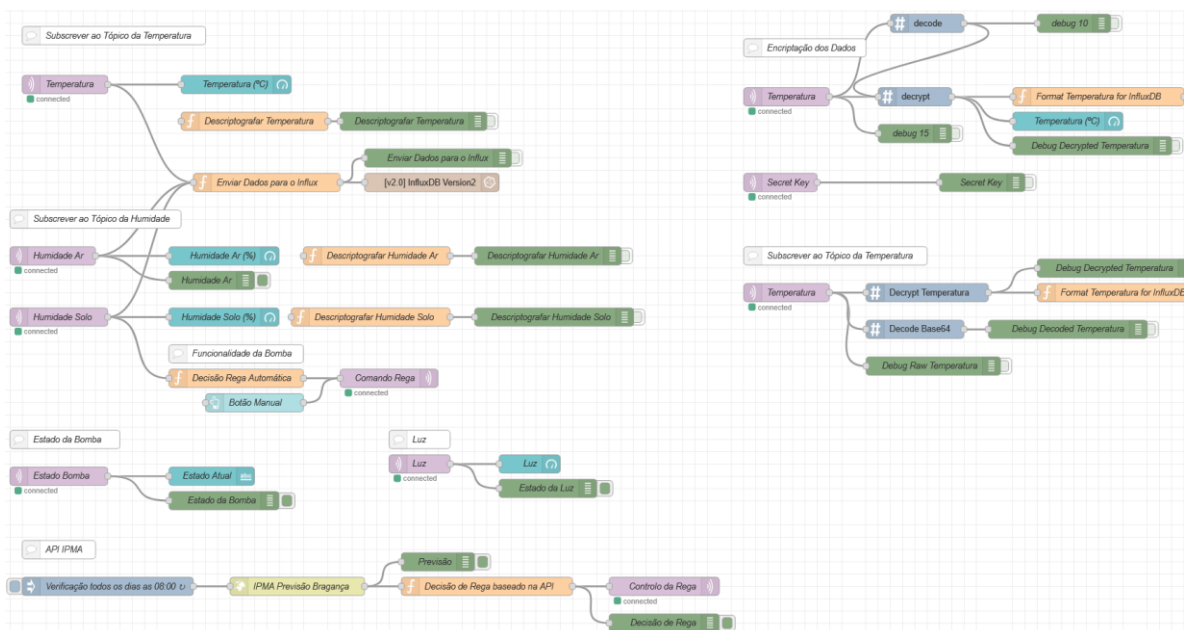
O firmware do ESP32 foi desenvolvido em C++ utilizando a plataforma Arduino. O código é responsável por recolher dados dos sensores, enviá-los ao Node-RED via MQTT e executar comandos recebidos para controlar a bomba de rega. A leitura da temperatura e humidade do ar é feita através do sensor DHT22, utilizando a biblioteca DHT.h. A humidade do solo é simulada com um potenciômetro ligado a um pino analógico do ESP32. Os valores lidos são enviados a cada 5 segundos para o broker MQTT nos tópicos definidos.

Além disso, o ESP32 subscreve-se ao tópico de controlo de rega, onde recebe comandos com os valores REGAR ou NAO_REGAR. Quando o comando REGAR é recebido, o pino associado à bomba (simulado com um LED) é ativado. O estado da bomba é, de seguida, publicado num tópico próprio para ser visualizado no dashboard. A gestão de conectividade Wi-Fi e MQTT inclui reconexão automática em caso de falha, assegurando a robustez do sistema.

3.2 Flow no Node-RED

O Node-RED foi utilizado como plataforma central para:

- Receber os dados de temperatura, humidade do ar e humidade do solo via MQTT
- Apresentar os dados num dashboard com gauges interativos
- Tomar decisões automáticas com base na humidade do solo
- Permitir o controlo manual da bomba
- Mostrar o estado atual da bomba em tempo real



4. Resultados

A solução foi testada no ambiente de simulação Wokwi, o que permitiu validar todas as funcionalidades previstas, desde a recolha de dados até à execução da lógica de rega automática. A comunicação entre o ESP32 e o Node-RED, via broker MQTT público, funcionou corretamente, com publicação periódica das leituras dos sensores e receção dos comandos de controlo.

4.1 Comportamento do Sistema

Durante os testes, verificou-se que o sistema respondia adequadamente às condições simuladas. Quando o valor da humidade do solo era inferior a 30%, o Node-RED enviava automaticamente o comando REGAR, que resultava na ativação do LED (simulando a bomba). Quando o valor subia acima do limiar, o comando NAO_REGAR era enviado e o LED desligava.

Além disso, o botão de rega manual permitiu ao utilizador acionar a bomba independentemente da decisão automática, demonstrando o correto funcionamento da lógica de prioridade.

4.2 Visualização no Dashboard

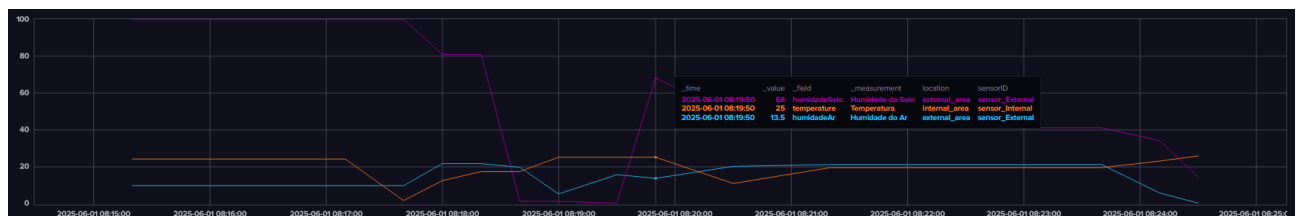
O dashboard do Node-RED apresentou corretamente os valores atuais dos sensores através de medidores (gauges) para:

- Temperatura (°C)
- Humidade do ar (%)
- Humidade do solo (%)

Foi ainda disponibilizado um campo de texto para o estado atual da bomba, bem como o botão de controlo manual.

4.3 Exemplos de Dados Recolhidos

Os dados ambientais recolhidos pelo ESP32, nomeadamente a temperatura (°C), humidade relativa do ar (%) e humidade do solo (%), foram enviados ciclicamente para o Node-RED através de mensagens MQTT. Após receção, os valores foram armazenados numa base de dados InfluxDB, especialmente adequada para dados temporais, pela sua eficiência no armazenamento e consulta de séries temporais.



4.4 Dificuldades

Durante o desenvolvimento do projeto, surgiram diversas dificuldades relacionadas com a implementação de funcionalidades mais avançadas, integração de componentes e estabilidade da comunicação entre dispositivos. Estas dificuldades serviram de base para reflexão sobre melhorias e possíveis evoluções futuras do sistema.

Uma das principais limitações encontradas relacionou-se com a encriptação dos dados. Foi implementado um mecanismo de encriptação no ESP32, onde os dados dos sensores são codificados antes de serem enviados via MQTT. No entanto, a desencriptação no Node-RED não foi bem-sucedida, devido a dificuldades de compatibilidade e gestão de chaves. Este problema impediu a utilização eficaz da segurança de ponta a ponta, sendo necessário, no futuro, estabilizar a comunicação com bibliotecas de encriptação AES compatíveis entre Arduino e JavaScript (Node.js).

Outra limitação observada foi a integração com a API do IPMA para obter previsões meteorológicas. Apesar de a API ter sido consultada com sucesso, a lógica implementada apenas verifica a previsão uma vez por dia, às 8:00 da manhã. Esta abordagem, embora funcional, é estática e poderia ser melhorada com uma lógica mais dinâmica e reativa, que condicionasse a rega automática ao longo do dia, dependendo da evolução da previsão de precipitação.

A visualização dos dados históricos através do InfluxDB também apresentou desafios. Embora funcional, a interface nativa é limitada em termos gráficos e interatividade. Seria vantajoso integrar uma plataforma como o Grafana, que permitiria uma visualização mais moderna e intuitiva dos dados, com possibilidade de análise comparativa, alertas e personalização.

Por fim, identificou-se como ausência relevante a verificação do nível de água no depósito. A implementação de um sensor de nível (por exemplo, ultrassónico ou com boia) permitiria prevenir falhas no processo de rega e notificar o utilizador da necessidade de reposição de água, aumentando a robustez do sistema.

Apesar destes constrangimentos, o projeto cumpriu os objetivos essenciais e estabelece uma base sólida para futuras melhorias.

5. Conclusões

O desenvolvimento deste sistema de rega automática permitiu demonstrar a aplicabilidade prática da Internet das Coisas (IoT) em contextos de monitorização ambiental e automação agrícola. A solução concebida proporciona a recolha contínua de dados sobre temperatura, humidade do ar e humidade do solo, possibilitando a tomada de decisões automáticas quanto à ativação de rega, promovendo o uso eficiente de recursos como a água.

A implementação baseada em ESP32, aliada à comunicação MQTT e à lógica de controlo desenvolvida em Node-RED, evidenciou a viabilidade de construir sistemas distribuídos, modulares e escaláveis. A simulação no ambiente Wokwi facilitou a validação funcional do sistema, mesmo na ausência de hardware físico, permitindo a reprodução de diferentes cenários e a observação do comportamento do sistema em tempo real.

Entre as principais particularidades do projeto destaca-se a separação clara entre os componentes de aquisição de dados, lógica de decisão e visualização, permitindo uma arquitetura facilmente extensível. A introdução de um botão de rega manual garantiu flexibilidade adicional ao utilizador, respeitando a lógica de priorização manual sobre a decisão automática.

Do ponto de vista prático, esta solução pode ser adaptada a diferentes contextos agrícolas ou domésticos onde exista a necessidade de automatizar a rega de plantas, promovendo a sustentabilidade e a redução do desperdício.

Em suma, o projeto atinge os seus objetivos ao conjugar tecnologias de IoT numa aplicação realista e funcional, servindo como base para aplicações mais robustas no domínio da agricultura inteligente.

Referências Bibliográficas

[1] Arduino Project Hub. (n.d.). DHT22 Temperature and Humidity Sensor with Arduino. Disponível em: <https://create.arduino.cc/projecthub> (Acessado em maio de 2025)

[2] EMQ Technologies Co., Ltd. (2024). EMQX - MQTT Broker. Disponível em: <https://www.emqx.io> (Acessado em maio de 2025)

[3] Node-RED. (n.d.). Node-RED User Guide. Disponível em: <https://nodered.org/docs> (Acessado em maio de 2025)

[4] Wokwi. (n.d.). ESP32 Simulator Online. Disponível em: <https://wokwi.com> (Acessado em maio de 2025)