

## Protótipo de Monitoramento de Nível de Cisternas em Rede LoRa, utilizando controladores ESP32

Victor Czarnobay (IFPB, Campus João Pessoa), José J. P. Gonçalves (IFPB, Campus João Pessoa), Luciana Pereira Oliveira (IFPB, Campus João Pessoa).

**E-mails:** victor.czarnobay@academico.ifpb.edu.br, pires.jefferson@academico.ifpb.edu.br, luciana.oliveira@ifpb.edu.br.

**Área de conhecimento (Tabela CNPq):** 1.03.04.02-9 Arquitetura de Sistemas de Computação.

**Palavras-chave:** esp32, lora, mqtt, sistemas distribuídos.

### 1. Introdução

Com o avanço tecnológico e a popularização de sistemas computacionais embarcados, expandiram-se as aplicações de monitoramento e controle automático em áreas como agricultura, meteorologia, aquicultura e pecuária. Quando integrados a redes de comunicação sem fio, esses sistemas ganham ainda mais relevância, especialmente em regiões remotas com infraestrutura limitada e recursos tecnológicos ou econômicos restritos.

Entre essas tecnologias, a rede de comunicação sem fio LoRa se destaca por oferecer longo alcance, baixo custo e consumo energético reduzido, tornando-se uma opção viável para aplicações em áreas extensas e isoladas (LoRa Alliance, 2015).

Embora diversos estudos tenham explorado o uso de redes LoRa em aplicações como agricultura (Verma e Bodade, 2025) e aquicultura (Soto-Ortiz, Tolentino-Cruz e Porras-Rosas, 2024), observa-se uma lacuna significativa no monitoramento sistemático de cisternas de água, que são utilizadas em regiões semiáridas como o interior do Nordeste brasileiro. Nessas localidades, as cisternas desempenham um papel vital no armazenamento e no acesso à água para o consumo humano, animal e agrícola. No entanto, a ausência de monitoramento em tempo real dificulta o planejamento eficiente do abastecimento, podendo levar ao esgotamento não previsto dos reservatórios ou à distribuição ineficaz de recursos hídricos.

Este trabalho propõe o desenvolvimento de um protótipo de sistema de monitoramento remoto e de baixo custo para cisternas, utilizando controladores ESP32 em rede LoRa, integrados a um servidor responsável por armazenar os dados, apresentar informações em um *dashboard* e permitir a configuração remota dos parâmetros do sistema. Diferente das abordagens anteriores, este sistema foca especificamente na gestão hídrica descentralizada em comunidades do semiárido, contribuindo para o uso racional da água e o planejamento preventivo do abastecimento.

### 2. Materiais e métodos

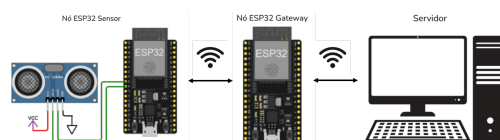
Os controladores ESP32 utilizados são da marca Heltec (v2), com capacidade de comunicação LoRa e Wi-Fi (Heltec, 2023). Foram desenvolvidos dois modelos com base nesses controladores, de acordo com seus pontos de utilização no projeto, que definimos como nós “ESP32 Sensor” e “ESP32 Gateway”:

- Nó “ESP32 Sensor”: módulo do ESP32 equipado com um sensor ultrassônico modelo HC-SR04 a ser instalado em uma cisterna. Possui rotinas de software desenvolvidas para medir o nível de água, converter os dados para mensagens padronizadas e transmitir dados via rede LoRa.

- Nó “ESP32 Gateway”: módulo do ESP32 que funciona como um centralizador de informações, a ser instalado em um ponto médio e estratégico entre os nós “ESP32 Sensores”. Possui rotinas de software desenvolvidas para adquirir dados via rede LoRa e transmiti-los via rede MQTT.

Artefatos de software também foram confeccionados e aplicados a um computador *desktop* com ambiente Linux, denominado “Servidor”, com funções de adquirir, monitorar e gravar os dados enviados pelos nós e configurar e gerenciar as aplicações.

Figura 1 - Arquitetura geral do sistema.



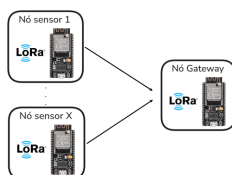
Fonte: autor (2025)

#### 2.1 Redes LoRa e MQTT

Uma rede LoRa foi implementada para troca de mensagens entre os nós “ESP32 Sensor” e “ESP32 Gateway”. Os nós “ESP32 sensor” são responsáveis por adquirir os sinais dos sensores de nível instalados nas cisternas, tratá-los

e periodicamente enviá-los para o nó “ESP32 Gateway; este último, então, adquire as informações de cada nó sensor, converte para uma estrutura de dados padrão e periodicamente retransmite para o Servidor.

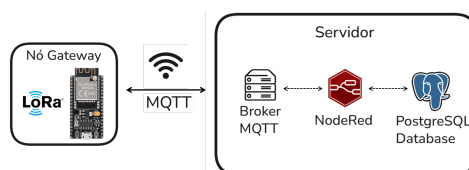
Figura 2 - Arquitetura da rede LoRa.



Fonte: autor (2025).

A comunicação entre o nó “ESP32 Gateway” e o Servidor é realizada a partir do protocolo MQTT, em que o primeiro subscrive as informações vindas dos nós sensores em um tópico específico do MQTT broker situado no Servidor.

Figura 3 - Arquitetura de rede MQTT.



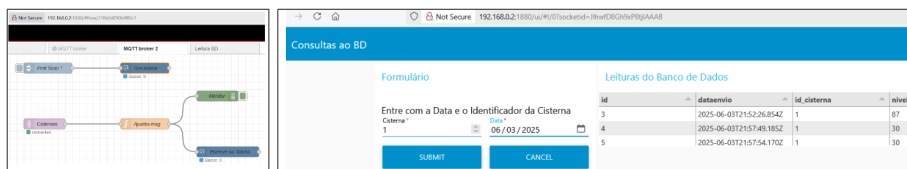
Fonte: autor (2025).

No servidor foi desenvolvido um cluster de containers em um ambiente Linux Ubuntu, composto por três principais serviços:

- Broker MQTT: cria um link de comunicação com o Nó Gateway a partir de uma rede MQTT. O Nó Gateway funciona como um *publisher*, escrevendo periodicamente as informações obtidas dos nós sensores em um Tópico; o Broker MQTT tem a função de *subscriber*, gerenciando o recebimento dos pacotes inseridos no tópicos e disponibilizando as informações para os demais serviços;
- NodeRed: recebe os dados do Broker MQTT, trata as informações em um formato adequado e faz operações de leitura e escrita em um banco de dados. Disponibiliza *dashboards* para configurações e apresentação dos dados gravados no banco de dados.
- PostgreSQL Database: recebe os dados obtidos do NodeRed e os grava na base de dados.

O serviço NodeRed disponibiliza uma interface acessível via web para programação e criação de *dashboards* para a aplicação, sendo possível utilizar ferramentas de codificação “Node.js” ou gráficas (*low-code*). Nela foram desenvolvidas as rotinas para configuração do Broker MQTT, tratamento de dados e integração com o banco de dados PostgreSQL, além de *dashboards*, criados para facilitar a visualização dos dados e possibilitar a configuração amigável dos parâmetros do sistema.

Figura 4 - Exemplos de rotina de programação em blocos e *dashboard* de consultas ao banco de dados.



Fonte: autor (2025).

### 3. Resultados e discussão

Os primeiros testes envolvendo o nó “ESP32 sensor” com sensores de nível e comunicação LoRa estão descritos no trabalho predecessor de Gonçalves (2023). O presente trabalho concentrou-se no desenvolvimento das rotinas do nó “ESP32 Gateway”, na criação dos artefatos de software do “Servidor”, integração entre os nós do sistema e testes em bancada com os componentes.

Em laboratório foi montada uma estrutura similar ao apresentado na Figura 1 deste trabalho, com um nó “ESP32 Sensor” conectado a um computador para *debug* e envio de mensagens; o nó “ESP32 Gateway” foi ligado a um segundo computador e o “Servidor” aplicado a um terceiro computador.

Para simular o funcionamento do sistema uma mensagem é enviada no primeiro computador para o nó “ESP32

Sensor”, emulando os resultados de medição de um sensor de nível, em uma estrutura pré-definida, conforme a imagem da figura 5.

Figura 5 - comando enviado ao nó “ESP32 sensor”.

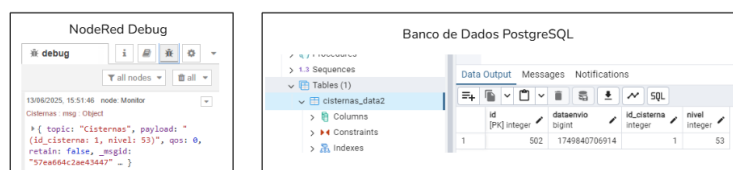
```
PS C:\Program Files\mosquitto> .\mosquitto_pub.exe -i mosq_pub1 -t "Cisternas" -m "(id_cisterna: 1, nivel: 53)" -h 192.168.0.2 -p 1883
```

Fonte: autor (2025)

A mensagem transmitida pelo ESP32 é enviada ao tópico “Cisternas”, com o identificador da cisterna “id\_cisterna: 1” e o valor simulado de nível “nivel: 53”.

A mensagem é recebida pelo “Servidor”, que monitora o tópico “Cisternas”. Na Figura 6 são expostos os dados recebidos na ferramenta de Debug do NodeRed e no banco de dados PostgreSQL a partir da ferramenta PgAdmin.

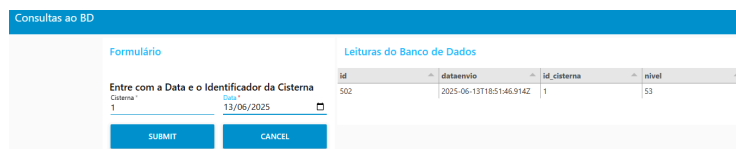
Figura 6 - mensagem observada no NodeRed Debug e gravada no banco de dados PostgreSQL



Fonte: autor (2025)

E no *dashboard* de visualização de dados do NodeRed é possível recuperar as informações gravadas no banco de dados:

Figura 7 - *Dashboard* com os dados da mensagem de teste.



Fonte: autor (2025)

#### 4. Considerações finais

A partir do trabalho realizado foi possível testar e validar os protótipos criados e as rotinas de software desenvolvidas em laboratório. Foi possível estabelecer e testar as comunicações LoRa e MQTT entre os dispositivos ESP32 e entre o ESP32 e o Servidor.

Como próximos passos para o avanço do projeto estão previstos testes de campo com os ESP32 e sensor ultrassônico instalado em uma cisterna real e testes de comunicação LoRa entre dois nós ESP32 para validar a distância máxima permissível.

A infraestrutura de containers criada no Servidor se mostrou adequada e útil para o monitoramento, análise e persistência de dados, além da configuração de parâmetros dos nós ESP32. A próxima evolução prevista será encapsular o projeto em um Dockerfile e instalar o sistema em um ambiente em nuvem.

#### Referências

GONÇALVES, J. J. P. **Proposta de Sistema de Gerenciamento de Cisternas Utilizando Sensoriamento e Internet das Coisas**. 2023. Monografia (Bacharelado em Computação) - Centro de Ciências Exatas e Sociais Aplicadas, UEPB, Patos-PB. Disponível em <https://dSPACE.bc.uepb.edu.br/jspui/handle/123456789/29718>. Acesso em 13 jun 2025.

HELTEC Automation. **Wi-Fi LoRa 32 (v2)**. 2023. Disponível em <https://heltec.org/project/wifi-lora-32v2/>. Acesso em: 10 jun 2025.

LORA ALLIANCE. **What is LoraWAN: a technical overview of LoRa and LoraWAN**. 2015. Disponível em <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>. Acesso em 10 jun 2025.

SOTO-ORTIZ, S.I.; TOLENTINO-CRUZ, M.G.; PORRAS-ROSAS, U. IoT monitoring and control prototype with Zabbix and LoRa communication, for Tilapia Biofloc systems. **2024 IEEE International Conference on Engineering Veracruz (ICEV)**, Boca del Rio, Veracruz, Mexico, 2025. DOI: 10.1109/ICEV63254.2024.10766033.

VERMA, A.; BODADE, R. Design and Implementation of a Low-Cost IoT-Based Smart Agricultural System Using ESP32 and LoRa Technology. **2025 International Conference on Computational, Communication and Information Technology (ICCCIT)**, Indore, India. p. 697-702, 2025. DOI: 10.1109/ICCCIT62592.2025.10928022.