UniAcademia
Centro Universitário

MONITORAMENTO DE UTILIDADES INDUSTRIAIS: UMA ABORDAGEM COM IOT E LORAWAN

SOUSA, José Edcarlos de [1]
PICCININI, Marco Aurélio[2]
ABRITTA, Camila do Carmo Almeida[3]
MAGRI, Luiz Paulo[4]

Linha de pesquisa: Automação

RESUMO

Este trabalho propõe o desenvolvimento de dispositivos IoT (*Internet of Things*) para setor industrial, com foco no monitoramento de utilidades industriais fazendo do uso de radiofrequência como meio de transmissão de dados, busca demonstrar a aplicação da comunicação de longa distância com o transceptor de rádio LoRa (*Long Range*). A coleta de dados utilizando essa tecnologia pode atingir até 3 km, dependendo da área de aplicação, como regiões remotas com poucas construções ou zonas abertas. De forma prática, será validada a confiabilidade dos dados com o menor consumo energético contínuo possível. Este trabalho tem como objetivo reduzir desperdícios e monitorar desvios, com potencial para aplicação em diversos consumos industriais, como água, vapor, lenha, cavaco, ar comprimido, entre outros, culminando na implementação real em processos industriais.

Palavras-chave: IoT. LoRa. Monitoramento. Utilidades. Industria.

1 INTRODUÇÃO

Muito se fala sobre sustentabilidade e o consumo consciente de recursos naturais, com o objetivo de reduzir o desperdício, sendo essa uma prática fundamental para maximização dos lucros e a redução dos impactos ambientais.

A indústria possui grande representação no consumo de recursos como de energia elétrica, sendo responsável por 32,3% do consumo do Brasil em 2020

[1] Graduando em Engenharia Elétrica pelo Centro Universitário Academia - UniAcademia
[2] Professor do curso de Engenharia Elétrica do Centro Universitário Academia - UniAcademia.
[3] Professora do curso de Engenharia Elétrica do Centro Universitário Academia - UniAcademia.
[4] Professor do curso de Engenharia Elétrica do Centro Universitário Academia - UniAcademia.

(EPE,2022), já o consumo de água representou 9,1% em 2022 enquanto o setor agrícola foi responsável por 50,5% (ANA, 2022).

Essas fontes de energia são de extrema importância para o desenvolvimento do país e devem ser mensuradas e controladas para garantir o uso responsável e eficiente, reduzindo as perdas e desvios de consumo. É fundamental que esses consumos sejam monitorados, uma vez que estudos apontam que o desperdício de energia elétrica por exemplo, gerou um prejuízo de 71 milhões de reais por dia de 2015 a 2017 (ABESCO, 2017).

A obtenção de dados de consumo é essencial para o desenvolvimento de modelos matemáticos que auxiliam em análises de desvios e desenvolvimento de perfil de consumo. Isso abre oportunidade para otimização de processos e reaproveitamento dos recursos, buscando maior eficiência nos sistemas industriais.

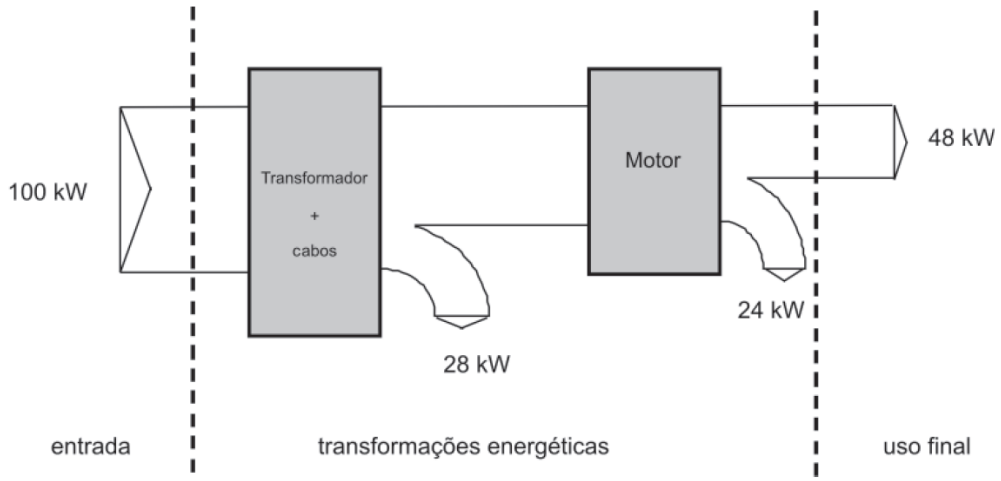
Todavia há processos em que se torna difícil obtenção de dados, seja por ausência de instrumentos, dificuldade de acesso, necessidade de coleta padronizada e regular, ou mesmo por riscos à saúde e a vida de colaboradores, tais recursos são, muitas vezes, custosos e impactam diretamente a continuidade do processo produtivo.

1.1 OBJETIVO

Este trabalho tem como objetivo o desenvolvimento da coleta de informações de consumo utilidades nos processos industriais, visando monitoramento e controle de desvios.

Como exemplo, no contexto do consumo de energia elétrica o diagrama de Sankey é uma ferramenta gráfica de representar o fluxo de potência em um equipamento, como exemplo, na Figura 1 o transformador com potência de 28kW e em um motor 24kW (Cesar, 2007). Diante desses valores, surge a seguinte questão: “E se as perdas aumentarem no transformador ou no motor, como identificar essa mudança ao longo do tempo?”.

FIGURA 1: Diagrama de Sankey



Fonte: Eficiência energética: teoria e prática, 2007.

1.2 MOTIVAÇÃO

Em diversas situações, o levantamento e registro dos consumos de eletricidade, água, lenha, cavaco, gás natural e outros recursos são realizados manualmente por operadores, o que os torna suscetíveis a erros humanos, falta de padronização, frequência inadequada de coleta e a variabilidade dos consumos ao longo do tempo, tornam o monitoramento desses dados complicada ou imprecisa, resultando em representações que não condizem com a realidade.

Para a transferência dos dados será utilizado a tecnologia de transmissão de via rádio frequência de longa distância, com o uso de rádio LoRa (*Long Range*), traduzindo para o português como “Longa Distância”, será associado ao uso do microcontrolador (MCU), formando o conjunto responsável pela coleta de dados e envio das informações.

A união das tecnologias é conhecida como IoT (*Internet of Things*) ou em português, “internet das coisas”, cujo princípio está baseado na conexão de todos os dispositivos à internet (Coelho, 2017).

2 REFERENCIAL TEÓRICO

Com o avanço da Indústria 4.0, o uso de dispositivos IoT que são dispositivos portados de microcontroladores com memória, processador e interfaces analógicas e

digitais tem se popularizado. São responsáveis por fazer a coleta dados de sensores e atuadores em tempo real, realizando tarefas com padronização e confiabilidade. Integrados à comunicação por rádio frequência, como o LoRa, esses dispositivos são ideais para ambientes industriais e áreas remotas.

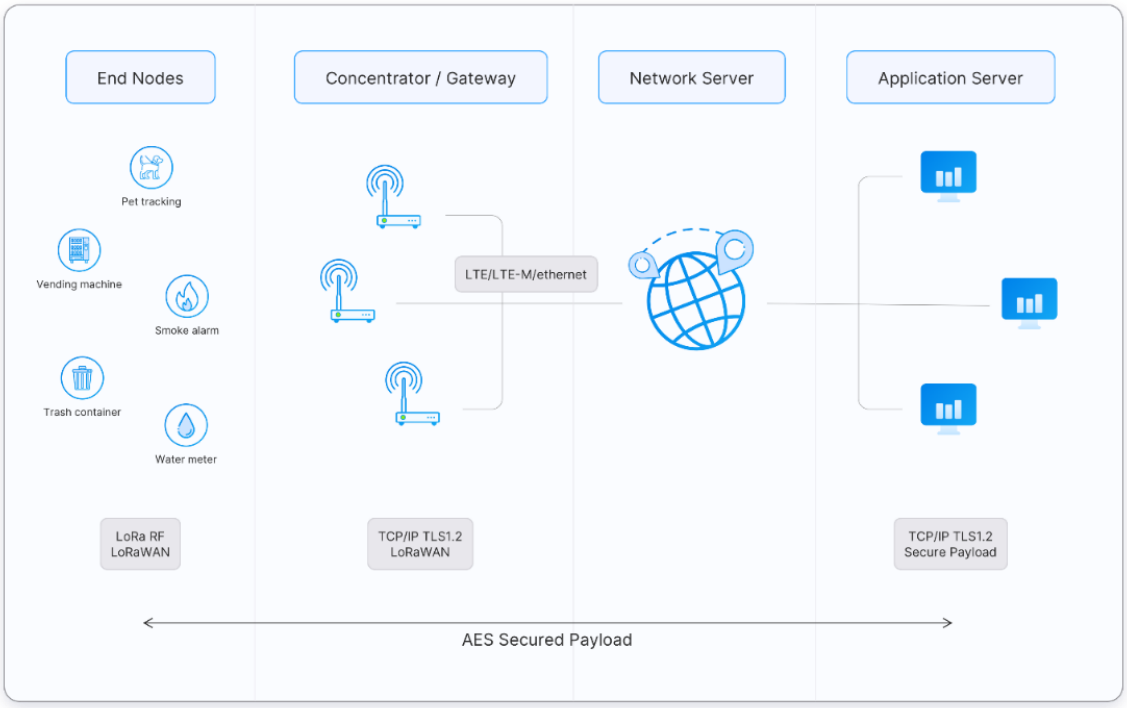
A comunicação eficiente é essencial para disponibilizar os dados de forma padronizada e cronológica, permitindo o monitoramento simultâneo de variáveis e a aplicação de modelos matemáticos, como regressão e aprendizado de máquina (The Things Network, 2024).

A instalação em ambientes industriais exige dispositivos robustos, resistentes à umidade, vapores, corrosão e interferências eletromagnéticas no qual é um dos atributos do LoRa.

2.1 PROTOCOLO LORAWAN DE COMUNICAÇÃO

O LoRaWAN (*Long Range Wide Area Network*) é um protocolo de comunicação desenvolvido e mantido pela LoRa Alliance onde um conjunto de aproximadamente 500 empresas em todo o mundo ajudam a manter a organização, possui a combinação de diversas tecnologias que juntas formam a arquitetura LoRaWAN. O nome deriva da combinação de duas tecnologias, sendo elas complementares a LoRa representando a camada física e o WAN a camada que proporciona gerenciamento da comunicação, segurança e o roteamento dos dados conforme a Figura 2 apresenta a topologia da rede LoRaWAN. (Bertoleti, 2023).

FIGURA 2: Topologia LoraWAN

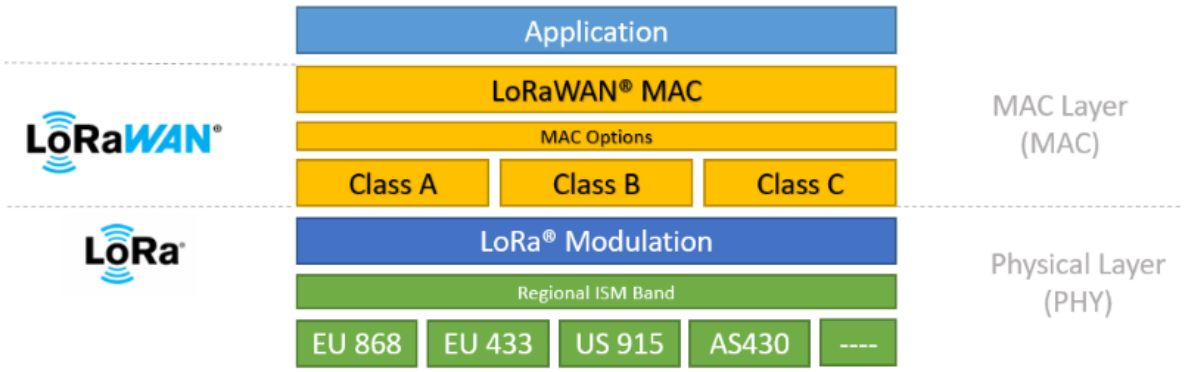


Fonte: The Things Network, 2024.

A arquitetura LoRaWAN utiliza a topologia em estrela, composta por *End Devices*, *Gateways* e *Network Server*. Os *End Devices* são sistemas embarcados com sensores ou atuadores que realizam tarefas específicas em campo, como coletar dados de sensores ou atuadores. *Gateways* atuam como intermediários entre os dispositivos e o servidor de rede, com capacidade de comunicação *half-duplex*, já o *Network Server* gerencia o tráfego de mensagens, garantindo integridade, segurança e eliminando duplicações por meio de identificadores únicos em sequência (Bertoleti, 2023).

Essa estrutura permite comunicações de longa distância com baixo consumo de energia, ideal para aplicações em IoT, sendo ainda resistente a interferências eletromagnéticas e compatível com fontes de energia como baterias e painéis solares (Figura 3).

FIGURA 3: Camada LoRaWAN e Rádio LoRa



Fonte: Moko Smart, 2019.

O rádio emissor de rádio frequência faz uso da tecnologia *Chirp Spread Spectrum* (CSS), essa técnica é semelhante a forma de comunicação existente na natureza usada por golfinhos para se comunicar e por morcegos para se localizar. (The Things Network, 2024). Utilizando a modulação de banda larga emite sinais de frequência modulada linear, os chamados *Chirp*, pode ser ascendente aumentando a frequência no tempo de forma linear ou descendente diminuindo a frequência, as vezes descrito na literatura como *UpChirps* e *DownChirps* (Inpixon, 2020).

2.1.1 Fundamentos da tecnologia LoRa

O conceito de *Chirp* traduzindo para o português gorjear, foi desenvolvido pelo professor Huttmann em aplicação de radares militares em 1940, onde mais tarde o conceito foi melhorado e surgiu a ideia de fazer uso do spectrum para na transferência de dados por Sidney Darlington em 1947, em seguida o projeto se deu continuidade pela empresa Canon. (The Things Network, 2024).

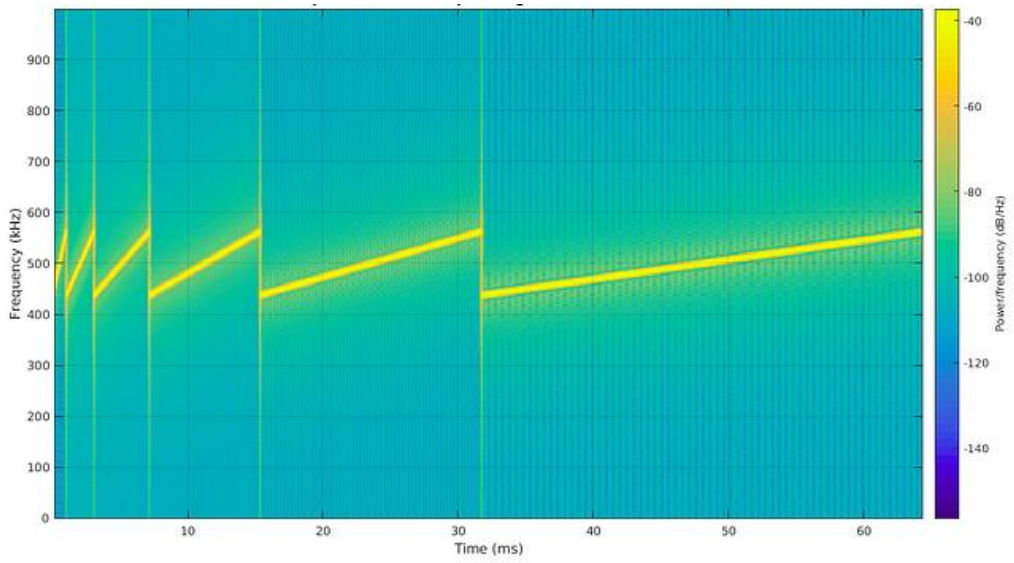
LoRa é baseada na modulação com o uso de *Chirp* em transmissão de dados sem fio, usa uma técnica chamada *Chirp Spread Spectrum*, patenteado pela empresa Semtech em 2012 e lançado em sua versão 1.0 em 2015. A tecnologia possui inúmeros benefícios, como baixo consumo de energia e alcance á longas distancias e imunidade a ruídos e interferências, porém deve ser levado em consideração que a

tecnologia é destinada a transmissão de pequenos pacotes de dados, por esse motivo possui um alcance superior comparado com outras tecnologias. (Semitech, 2025).

O sucesso do LoRa está no uso da modulação *Chirp Spread Spectrum* (CSS), que opera em três larguras de banda: 125 kHz, 250 kHz e 500 kHz. O *Spreading Factor* (SF) determina a velocidade do *chirp*, quanto maior o SF, mais lento ele é, aumentando o tempo no ar e a probabilidade de transferência de dados. O LoRa utiliza seis valores de SF, variando de SF7 a SF12.

A modulação do sinal de *Chirp* é semelhante a modulação por chaveamento de frequência (FSK) porém é um sinal mais resistente a interferências eletromagnéticas, ruídos e efeito *double* chegando a possuir um alcance 3 vezes maior do que o FSK. (Frugal prototype, 2016) (Figura 4).

FIGURA 4: Spreading Factor



Fonte: The Things Networks, 2024.

Cada *chirp* de transmissão LoRa é conhecido como um símbolo da senoide, que possui uma frequência variável, a alteração do valor da variável m na Equação 1 altera também o tempo que o símbolo será transmitido, que também está associado no aumento da sensibilidade do receptor, permitindo assim que o sinal seja detectado a maiores distâncias ou em canais com maiores ruídos, o formato do símbolo pode ser definido pela Equação 1 (Frugal Prototype, 2016).

$$Si(t) = A.\cos\left(\left(\omega 1 + \frac{1}{2}\, mt\right)\right)\ \forall t\ \in [0, \frac{1}{BW}] \quad \omega c = \ \omega 1 + \frac{BW}{4\pi}$$

(Equação 1)

sendo as variáveis:

$Si(t)$ - Símbolo de modulação LoRa dentro da largura de banda definida.

A – Amplitude do sinal.

$\omega 1$– Frequência de início da modulação.

m- Taxa de variação da frequência.

ωC– Frequência da portadora.

O SF é indiretamente relacionado a taxa de envio de bits por segundo quando maior o SF menor a taxa de transmissão, sendo assim manter um valor baixo de SF é o ideal quando se necessita aumentar a taxa de transferência, dois parâmetros fundamentais são a taxa de bits (Rb) e a taxa de símbolos (Rs) de definida pelas equações.

$$Rb = SF \cdot \frac{BW}{2^{SF}}$$

(Equação 2)

Rb – Taxa de bits.

SF – Fator de Espelhamento.

BW – Largura de Banda.

2^{SF} – Número de *chirps* por símbolo.

$$Rs = \frac{BW}{2^{SF}}$$

(Equação 3)

Rs – Taxa de símbolos.

BW – Largura de Banda.

2^{SF} – Número de *chirps* por símbolo.

$$T = \frac{2^{SF}}{BW}$$

(Equação 4)

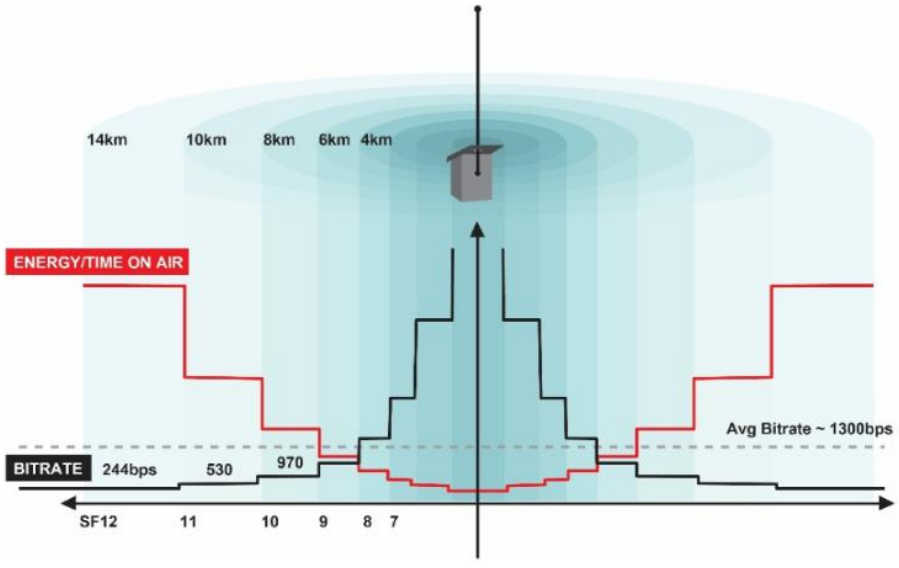
T – Duração do Chirp

Aumentar o SF no rádio LoRa amplia o alcance, mas reduz a taxa de transmissão. Já aumentar a largura de banda (BW) diminui a sensibilidade do rádio. O ideal é manter um equilíbrio entre SF e BW para otimizar o alcance, a taxa de transferência e o consumo de energia (Frugal Prototype, 2016).

Para reduzir o consumo energético, utiliza-se o *Adaptive Data Rate* (ADR), um recurso do protocolo LoRaWAN ajusta automaticamente o *Data Rate* (DR) e o tempo das mensagens no ar (*Airtime*), limitado a 400 ms (Bertoleti, 2023, p. 61). Quanto maior o DR, menor o SF, sendo que existem sete níveis de SF (SF7 a SF12) e DR correspondentes (DR0 a DR5). O ADR envia mensagens *downlink* para o dispositivo final, orientando a alteração do SF, otimizando o consumo de energia e o tempo de transmissão.

O ADR também avalia o link *budget* cuja é a diferença entre a potência do sinal e a sensibilidade do receptor, para realizar os ajustes de forma eficiente no DR e SF (Bertoleti, 2023) (Figura 5).

FIGURA 5: Taxa de transmissão, distância do end device ao gateway e tempo de mensagem no ar



Fonte: Ebyteiot, 2025.

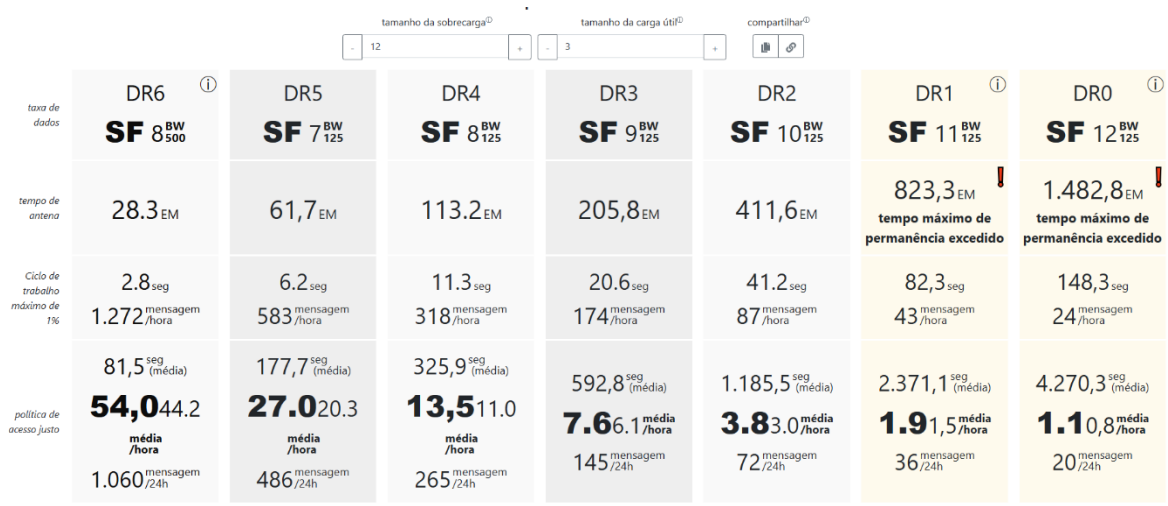
UniAcademia
Centro Universitário

2.1.2 Aspectos Regulatórios

LoRa opera em uma faixa de frequências não licenciadas, isso significa que qualquer pessoa pode fazer uso das frequências sem pagar taxas pelo direito de transmissão, alguns parâmetros são definidos pela Agência Nacional de telecomunicações (ANATEL), onde é definido para o Brasil o padrão Australiano AU915-928, com frequências de 915MHz e 928MHz com largura de banda de 125kHz e limite de mensagem no ar de 400ms, os dispositivos que fazem uso dessas frequências possuem requisitos que devem ser obedecidos (ANATEL, 2017).

Na Figura 6 é apresentado uma ferramenta chamada "*Airtime Calculator for LoRaWAN*" para realizar o cálculo do tempo que a mensagem ficará no ar, por exemplo com SP 11 e 12 o limite de tempo foi excedido (LoRa Alliance, 2025).

FIGURA 6: Ferramenta para calcular a carga de dados, SF e DR



Fonte: Avbentem, 2025.

2.1.4 Classes LoRaWAN

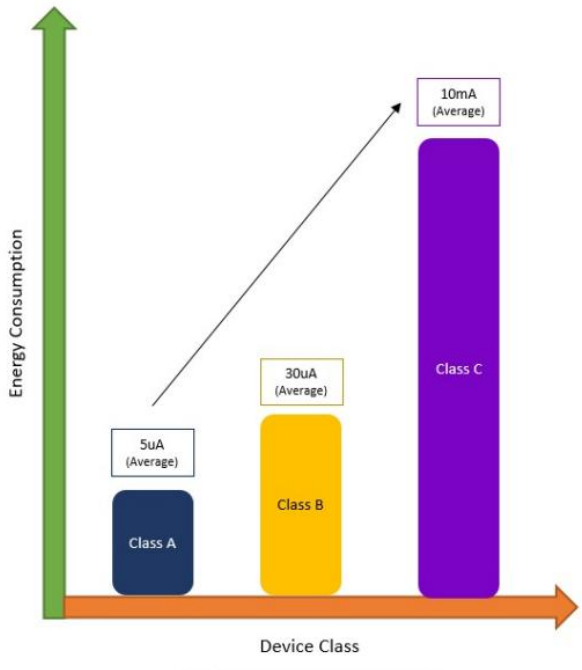
LoRaWAN opera com 3 classes diferentes, onde define o modo de operação dos dispositivos, a definição desses modos está diretamente ligada a taxa de transmissão e o consumo de energia pois dependendo do modo o rádio passa a ficar mais ou menos tempo ligados.

As classes são definidas por classe A, classe B ou classe C, sendo obrigatório a classe A caso não seja definida outra classe, outro ponto importante é que a definição da classe está definida também pelo modelo do transceptor (dispositivo que

executa papel de enviar e receber dados) onde deve ser verificado no Datasheet traduzindo para o português folha de dados do dispositivo (The Things Network, 2024).

A definição da classe de operação deve ser bem definida, levando em consideração a aplicação, se será necessário o modo de hibernação, quantidade de mensagens, necessidade de mensagens *downlink*, *uplinks*, capacidade do dispositivo de atender as classes determinadas e principalmente o consumo de energia e disponibilidade da fonte de energia, visando a optimização e maior eficiência do sistema, a depender da classe o consumo pode ser 333 vezes maior, conforme ilustrado na Figura 7.

FIGURA 7: Comparativo do consumo de energia de end devices na classe A, B e C



Fonte: Semtech, 2025.

No presente projeto, a comunicação entre os módulos será realizada em modo LoRa ponto a ponto, o que elimina a necessidade de *gateways*, servidores externos ou conexão com a internet.

Transceptores da série SX127X como por exemplo SX1262, SX1272 e SX1276 foram os primeiros a serem lançados possuindo um espectro espalhado de longo alcance e com imunidade a interferências e ruídos, consumindo o mínimo de corrente.

Por serem da mesma série possuem muitas semelhanças trabalhando em frequências de 860-1000MHz os modelos SX1272/73 e 137-1020MHz os modelos SX1276/77/78 e 79, com capacidade de +20dBm a 100mW operando na classe A, B ou C. (The Things Network, 2024).

2.1.5 Operação de rede e Modelos de Rádios

A comercialização do LoRaWAN no Brasil é feita pela a empresa americana American Tower, sendo responsável por manter a estrutura física de *gateways* LoRaWAN na rede pública nacional, a comercialização do direito de uso da rede pública é feita por empresas distribuidoras sendo essas empresas intermediarias responsáveis por fazer a entrega do serviço e gerenciamento das conectividades aos consumidores finais, alguns exemplos de empresas que comercializam esses direitos são Algar telecon, TCTEC telecom, NLT, KORE entre outras.

A área de cobertura atende grande parte do território povoado brasileiro, aproximadamente 51,1% do território, sendo assim há uma grande redução no custo e aquisição de infraestrutura para a implementação do sistema nessas áreas, onde na rede pública fica por responsabilidade da empresa distribuidora.

LoRa também possui a rede privada, onde quem faz o uso da rede deverá prover toda infraestrutura como *gateway*, torres, acesso à internet, aplicações e gerenciamento dos dispositivos (Bertoleti, 2023).

2.1.6 Onde a conectividade LoRa é indicada

Entre as aplicações mais compatíveis com a arquitetura LoRaWAN, incluem-se sistemas de envio de leituras de sensores ambientais, como temperatura, umidade, iluminância e consumo de utilidades água, gás e energia elétrica. A tecnologia também se mostra eficaz na transmissão de sinais discretos (on/off), além de alertas e alarmes de segurança, como notificações de violação de acesso, detecção de incêndio ou presença de gases inflamáveis. Outro campo de aplicação relevante é o controle remoto de periféricos, como relés, motores e sistemas de iluminação, além do envio de status operacionais periódicos de máquinas e equipamentos (Actility, 2023).

O uso do rádio LoRa ou LoRaWAN não é recomendado para sistemas que exigem transmissão contínua, grande volume de dados por pacote ou alta frequência de comunicação. Situações como envio de áudio ou vídeo, sincronização de arquivos entre dispositivos e servidores, transmissão de dados em tempo real que excedem os limites operacionais da tecnologia e em sistemas de segurança onde é requerido transmissão rápida onde o atraso coloque em risco máquinas e pessoas. Nestes casos, torna-se necessário adotar soluções com maior largura de banda e menor latência (Actility, 2023).

2.2 COMPARATIVO ENTRE PROTOCOLOS DE COMUNICAÇÃO NA INDÚSTRIA

Os protocolos de comunicação IoT é uma parte essencial quando se trata de definir o protocolo correto para o projeto, pois é ele que irá fazer a comunicação entre os dispositivos e garantir que funcione dentro dos requisitos esperados, sejam aplicações em casas inteligentes, ambientes fabris, hospitais ou áreas remotas o protocolo de comunicação é que determina a interação dos dispositivos (Calvi Wireless, 2025).

Durante a fase de dimensionamento é importante garantir e conhecer as camadas físicas e de enlace de dados a fim de dimensionar corretamente esses pontos que são de extrema importância, os protocolos de comunicação sem fio mais utilizados na camada física são, Bluetooth de baixa energia (BLE), WIFI, ZigBee, LoRaWAN, SigFox, a Figura 8 faz uma breve comparação entre protocolos. (Calvi Wireless, 2025).

FIGURA 8: Comparação protocolos de comunicação

Attributes	Wi-Fi HaLow	BLE	ZigBee	SigFox	LoRaWAN
Data Rate [1]	150 kbps	125 kbps	10 kbps	100 bps	250 bps
Range	1 km	100 m	14.5 km	10 km	15 km
Battery life [10]	Years	Years	Years	Years	Years
Network topology [10]	Star	P2P	Mesh	Star	Star
Standard	IEEE 802.11ah	Bluetooth SIG	IEEE 802.15.4	Proprietary	Proprietary
Subscription-based [10]	No	No	No	Yes	Yes

[1] Values for highest range.

Fonte: The Challenge of Long-Distance Over-the-Air Wireless Links in the Ocean, 2025.

2.3 MICROCONTROLADORES ESPECIFICAÇÕES TÉCNICAS BÁSICAS

Na automação industrial, os microcontroladores tornaram-se elementos centrais nos sistemas de controle distribuído, devido à sua capacidade de execução local de tarefas, economia de energia e custo reduzido. Esses dispositivos integram, em um único chip, o processador, memórias, interfaces de entrada e saída, e periféricos de comunicação, sendo amplamente aplicados em sistemas embarcados, instrumentação e controle de processos industriais (Próspero, 2020).

No contexto industrial, destaca-se também a importância das interfaces de comunicação, que viabilizam a integração dos microcontroladores com sistemas maiores. Protocolos como UART, SPI e I2C permitem a conexão direta com sensores, módulos de comunicação e outros dispositivos embarcados. Além disso, protocolos robustos como RS-232 e RS-485 são amplamente empregados para garantir a comunicação confiável entre equipamentos em ambientes industriais, mesmo em grandes distâncias e sob interferência eletromagnética (Próspero, 2020).

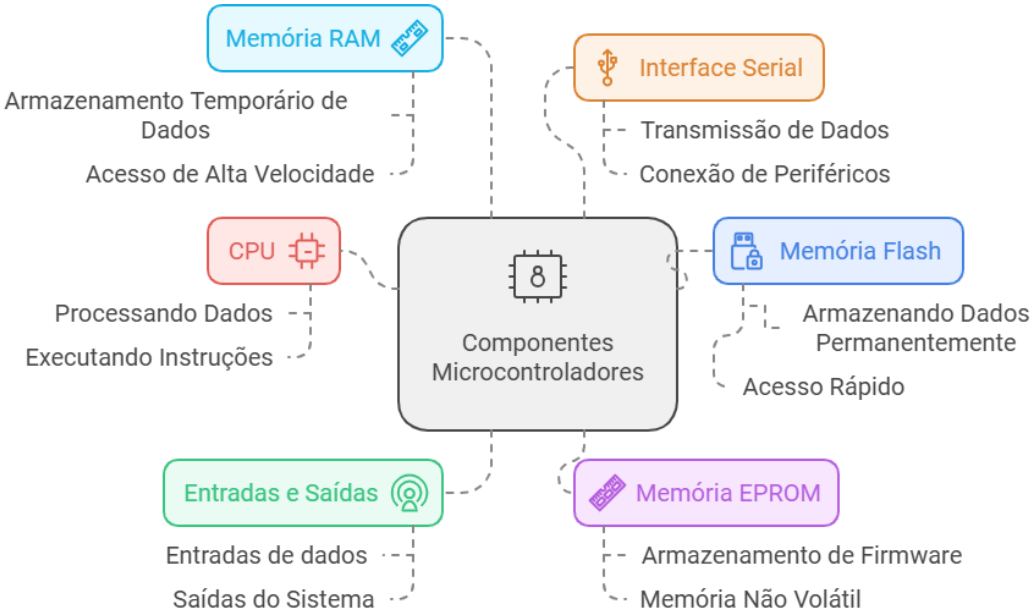
Um microcontrolador (MCU) é a base para os sistemas IoT e é considerado um pequeno computador, porém reduzido a um único chip sendo projetado para executar tarefas específicas sem necessidade de sistemas operacionais complexos (Figura 9).

Um sistema de Internet das Coisas (IoT) é composto por um conjunto de dispositivos eletrônicos interconectados com microcontroladores, capazes de coletar, processar e transmitir dados entre si ou para servidores remotos. A arquitetura básica desses sistemas contempla elementos de sensoriamento, processamento, comunicação e alimentação de energia, integrados de forma eficiente e escalável.

Do ponto de vista técnico, um sistema IoT típico é composto por:

- **Módulo de sensoriamento:** engloba sensores capazes de monitorar variáveis do ambiente, como temperatura, pressão, umidade, corrente elétrica ou vazão. Esses sensores podem ter saídas elétricas analógicas ou digitais, sendo conectados a um microcontrolador por meio de interfaces como GPIO, I2C, SPI ou com uso de placas adicionais especificas.

FIGURA 9: Morfologia de um Microcontrolador



Fonte: Autor, 2025.

- **Unidade de processamento:** geralmente constituída por um microcontrolador ou microprocessador embarcado é o coração do IoT, responsável por interpretar os sinais provenientes dos sensores, executar lógicas de controle local e preparar os dados para envio. Dispositivos como o ESP32 são amplamente utilizados por oferecerem conectividade integrada, múltiplos canais de entrada/saída e consumo energético reduzido (Heltec, 2022).
- **Módulo de comunicação:** permite a transmissão dos dados coletados. As tecnologias variam conforme a aplicação, podendo incluir Wi-Fi, Bluetooth Low Energy (BLE), ZigBee, LoRa, NB-IoT ou redes móveis 4G/5G. A escolha do protocolo depende de fatores como alcance, taxa de dados, consumo energético, confiabilidade e as características do projeto (Rodrigues, 2021).
- **Interface de alimentação:** normalmente composta por fontes de tensão estabilizadas, baterias recarregáveis ou sistemas de coleta de energia (harvesting). Projetos de IoT exigem alta eficiência energética, sendo comum o uso de modos de operação em *sleep* para reduzir o consumo durante períodos de inatividade (Heltec, 2022).
- **Plataforma de supervisão e análise de dados:** embora não faça parte diretamente do hardware embarcado, os sistemas IoT são usualmente

integrados a servidores locais ou em nuvem, utilizando protocolos como MQTT ou HTTP para envio contínuo de dados. A visualização e análise desses dados podem ser feitas por ferramentas como Node-RED, Grafana ou plataformas web personalizadas (ANATEL, 2020).

3 DESAFIOS NA IMPLEMENTAÇÃO INDUSTRIAL

A integração de soluções baseadas em Internet das Coisas (IoT) no ambiente industrial tem se mostrado essencial para o avanço da automação, redução do consumo de recursos, eficiência operacional e da coleta de dados em tempo real. No entanto, a implementação dessa tecnologia é um cenário complexo em muitas plantas fabris, redes de utilidades e sistemas descentralizados, apresentando uma série de obstáculos técnicos e estruturais. Esses desafios vão desde barreiras de infraestrutura e segurança até problemas de escalabilidade e compatibilidade entre sistemas.

A adoção de dispositivos conectados, sensores inteligentes e redes distribuídas exige que os projetos industriais contemplem soluções robustas para mitigar riscos e garantir a confiabilidade dos sistemas e das informações. há quatro principais desafios para a implementação de IoT na indústria, são eles a segurança da informação, cobertura de rede, escalabilidade e a interoperabilidade, mas tais necessidades podem ser tratadas.

3.1 SOLUÇÃO PARA OS DESAFIOS

Os quatro pontos de observações representam os principais desafios práticos enfrentados por engenheiros, gestores e técnicos ao incorporar a IoT em ambientes industriais. Antecipar e planejar soluções para essas dificuldades é essencial para garantir a eficiência, a segurança e a longevidade dos projetos de automação conectada.

Para superar os desafios mencionados, é de suma importância que os projetos de IoT industrial incorporem práticas consolidadas e validadas desde a fase de planejamento. Se tratando de segurança, recomenda-se o uso de criptografia, autenticação multifator e atualizações regulares, implementação em sistemas

isolados forma estratégica e contínua, buscando aumentar a robustez, a eficiência e a vida útil das soluções IoT implementadas na indústria (Emnify, 2024).

3.2 SEGURANÇA DE DADOS

Com o avanço das tecnologias IoT na indústria, cresce também a preocupação com a segurança cibernética aplicada a dispositivos embarcados, especialmente microcontroladores utilizados em sensores, atuadores e gateways de rede. Os microcontroladores possuem restrições de hardware, energia que exigem medidas de proteção específicas. A negligência na segurança pode levar a invasões, manipulação de dados, interrupções operacionais e até danos físicos a equipamentos ou pessoas.

Existem princípios fundamentais que devem ser considerados no desenvolvimento e na implantação de soluções IoT seguras, como a troca de senhas padrão de fábrica, desabilitar interfaces de depuração e comunicação que não estiverem em uso, utilização de algoritmos de criptografia e monitoramento contínuo da segurança dos dispositivos de preferência multicamada e com isolamento entre as camadas (Embarcados, s.d.).

3.3 RECURSOS DE SEGURANÇA NO ESP32

MCUs ESP32 já incluem de fábrica padrões de segurança visando garantir a integridade dos dados onde são implementadas proteções de inicialização, proteções de criptografia da memória flash, armazenamento seguro de chaves, geradores de números aleatórios verdadeiros e proteções contra leitura de memória (Espressif, 2024).

A Espressif na qual é a fabricantes dos controladores faz a implementação de recursos de segurança colocam o ESP32 em conformidade com práticas modernas de proteção embarcada, sendo especialmente úteis em aplicações críticas como automação industrial, telemetria remota, controle de infraestrutura e dispositivos médicos. Sua utilização fortalece a confiabilidade, a privacidade e a resiliência dos sistemas baseados em IoT.

UniAcademia
Centro Universitário

4 MODBUS TCP, NODERED, INFLUXDB E GRAFANA

4.1 PROTOCOLO MODBUS

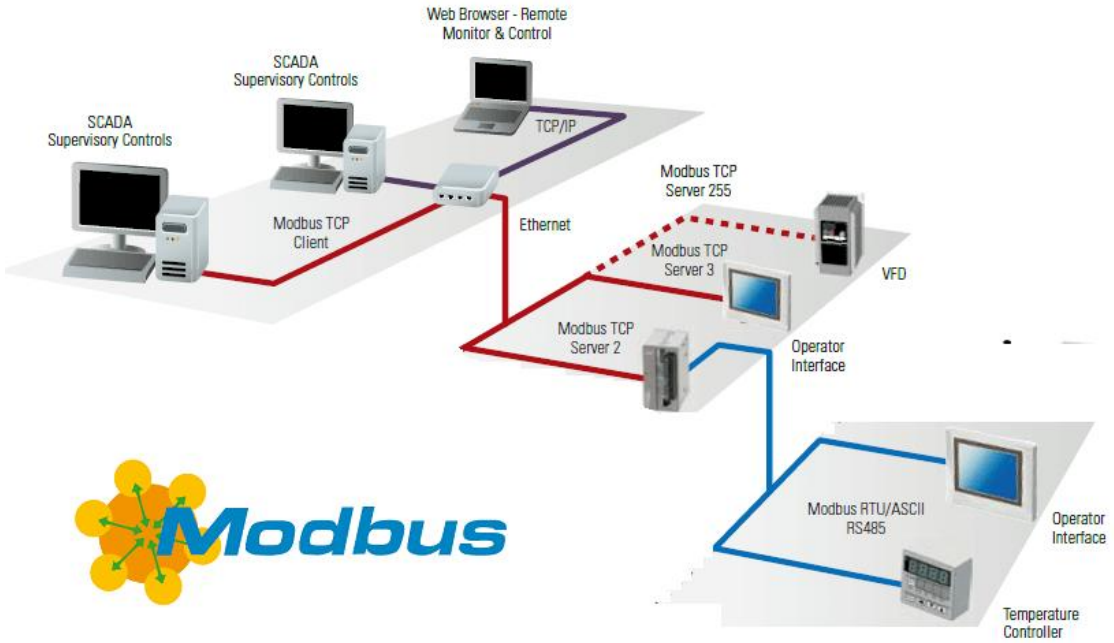
O *Modbus* é um protocolo de comunicação serial amplamente utilizado na automação industrial para a troca de informações entre máquinas. Desenvolvido originalmente pela Modicon em 1979, seu principal objetivo é permitir a comunicação entre controladores lógicos programáveis (CLPs) denominado com mestre e dispositivos como sensores, atuadores e sistemas supervisórios denominado também como escravos.

A estrutura do protocolo baseia-se em um modelo mestre-escravo, no qual um único dispositivo (por exemplo, um CLP ou computador) inicia todas as transações, enquanto os escravos apenas respondem às solicitações. Essa arquitetura simples torna o *Modbus* ideal para redes industriais onde é necessária uma comunicação confiável e determinística entre múltiplos dispositivos (National instruments, 2024).

O *Modbus* (Figura 10) pode operar sobre diferentes meios físicos e formatos, sendo os principais:

- ***Modbus* RTU:** versão serial que utiliza comunicação binária e é geralmente implementada sobre RS-485;
- ***Modbus* ASCII:** usa codificação ASCII é implementado sobre RS-232 ou RS-485;
- ***Modbus* TCP/IP:** adapta o protocolo para redes Ethernet, encapsulando as mensagens Modbus em pacotes TCP/IP.

FIGURA 10: Protocolo *Modbus*, RTU, ASCII e TCP



Fonte: Envisia, 2025.

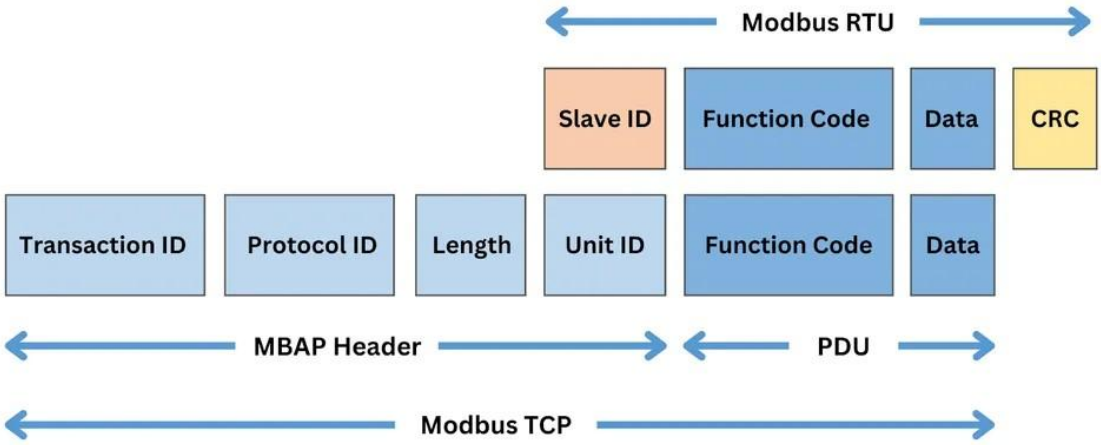
A estrutura de mensagem do protocolo inclui um identificador de dispositivo (endereço do escravo), um código de função (por exemplo, leitura ou escrita de registradores), dados e um campo de verificação de erro. A simplicidade dessa estrutura contribui para sua ampla adoção, inclusive em sistemas abertos e de baixo custo.

A robustez, a padronização e a fácil implementação fazem do Modbus um dos protocolos mais populares por ser simples pois em aplicações, como por exemplo RTU basta dois cabos para realizar a comunicação com diversos escravos (National instruments, 2024).

Neste projeto será aplicado apenas o *Modbus* TCP que é uma implementação do protocolo Modbus tradicional projetada para operar em redes Ethernet ou redes Wireless, utilizando o protocolo TCP/IP como meio de transporte para comunicação entre dispositivos industriais. O *Modbus* TCP preserva a simplicidade e a estrutura do protocolo Modbus RTU, mas adaptando-o à arquitetura moderna das redes de computadores (Wevolver, 2023).

A estrutura de dados do *Modbus* TCP consiste em duas partes como pode ser observado na Figura 11:

FIGURA 11: Estrutura de dados *Modbus* TCP



Fonte: Wevolver, 2025.

- **MBAP Header (Modbus Application Protocol Header):** É a estrutura do cabeçalho de 7 bytes que inclui:
  - *Transaction Identifier* (2 bytes): identifica a transação Modbus;
  - *Protocol Identifier* (2 bytes): valor sempre 0 para Modbus;
  - *Length* (2 bytes): número de bytes restantes na mensagem;
  - *Unit Identifier* (1 byte): identifica o destino no servidor (útil em gateways).
- **PDU (Protocol Data Unit):** composta por:
  - *Function Code* (1 byte): define a ação (ex: leitura de registradores);
  - *Data*: conteúdo específico da requisição ou resposta.

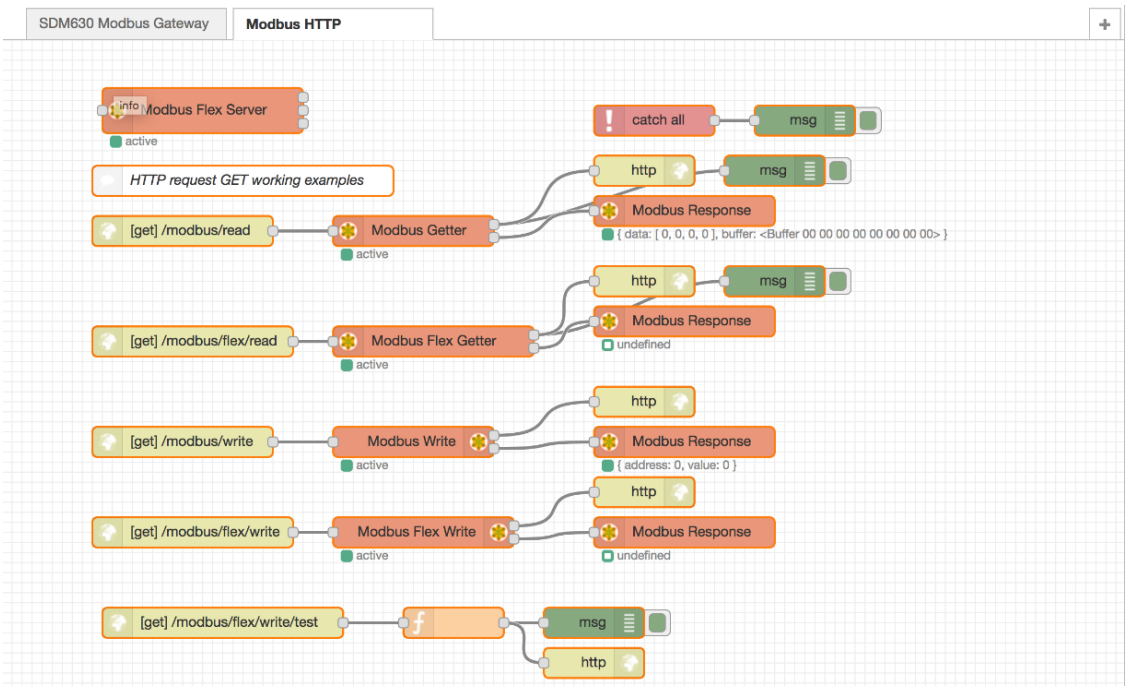
Uma característica valiosa do *Modbus* TCP é sua simplicidade e a alta interoperabilidade, sendo suportado por uma ampla gama de fabricantes e dispositivos. Por ser baseado em protocolos padronizados e usar *Ethernet* ou rede *Wireless* como meio físico, ele pode ser facilmente implementado em microcontroladores modernos como ESP32, sistemas embarcados com interface TCP/IP, e softwares industriais, utilizando bibliotecas abertas onde possui versões aprimoradas para microcontroladores amplamente disponíveis (Wevolver, 2023).

As funções do protocolo *Modbus* TCP são responsáveis para realização das operações especificas, como ler ou gravas dados em saídas e registradores e podem ser consultadas na documentação do protocolo.

4.2 CONFIGURAÇÃO DO NODE-RED

O nome *Node-RED* (Figura 12) é dado por possuir a ideia de trabalhar com "nós" por isso nome *Node* e o *Red* vem da ideia de evento de desenvolvimento rápido ou no inglês "*Rapid Event Developer*", é uma plataforma de desenvolvimento baseada em fluxo que permite a integração visual entre dispositivos, sendo amplamente utilizada em aplicações de Internet das Coisas (IoT), automação industrial, automação residencial e supervisão remota. Foi Desenvolvido a princípio pela IBM e disponibilizado como projeto de código aberto sob licença Apache 2.0 (Node-red, 2024).

FIGURA 12: Exemplo de Fluxo *Node-RED*



Fonte: Flows

Cada "nó" do *Node-RED* representa uma função modular, podendo ser configurado individualmente com parâmetros específicos. A biblioteca oficial da plataforma e a comunidade ativa fornecem centenas de nós prontos para uso, incluindo suporte nativo para protocolos industriais como *Modbus*, MQTT, HTTP,

LoRaWAN, TCP/UDP, além de integração com bancos de dados como InfluxDB e MySQL, e com plataformas como Grafana.

A plataforma oferece suporte a execução assíncrona, persistência de variáveis, tratamento de erros e controle de fluxo, o que a torna robusta o suficiente para uso em sistemas produtivos. Sua interface web facilita o desenvolvimento colaborativo e o versionamento de projetos.

A aplicação *Node-RED* cria um servidor onde pode ser implementado em computadores em redes locais ou em servidores online além de possuir em suas configurações o desenvolvimento de perfis de acesso aos fluxos, é ágil de baixa curva de aprendizado (*Node-RED*, 2024).

4.3 BANCOS DE DADOS INFLUXDB

Garantir que os dados serão armazenados para posteriormente serem analisados é primordial e o banco de dados InfluxDB é perfeito pois é um banco de dados projetado especificamente para armazenar, consultar e gerenciar grandes volumes de dados de séries temporais que são dados que variam com o tempo, como medições de sensores, registros de sistemas industriais, logs de eventos e métricas de desempenho. Por essa razão, é amplamente utilizado em soluções de Internet das Coisas (IoT), monitoramento industrial, automação predial e aplicações de energia inteligente (Influxdata, 2024).

No contexto industrial, o InfluxDB é particularmente valioso para armazenar medições de sensores conectados via ESP32, LoRa, Modbus ou outras interfaces. Isso permite o registro histórico contínuo de variáveis como temperatura, corrente elétrica, pressão, nível de reservatórios ou status de máquinas. Tais dados podem então ser acessados por aplicações de supervisão, inteligência artificial ou manutenção preditiva.

Dessa forma, o InfluxDB consolida-se como uma ferramenta central em sistemas modernos de automação conectada e monitoramento contínuo, atendendo com precisão às demandas da Indústria 4.0 (Influxdata, 2024).

4.4 GRAFANA

O Grafana é uma plataforma de análise de dados, visualização e monitoramento amplamente usada, tanto em sua versão *open source* (OSS) ou seja versão de *software* de código aberto ideal para quem busca versões gratuitas, porém não deixando de ser robusta, quanto em versões comerciais (*Enterprise* e *Cloud*) onde são versões empresariais onde possui integração com fontes de dados corporativos, controle de acesso e suporte técnico 24x7 (Grafana labs, 2024).

Alguns dos pontos fortes do grafana são, dashboards personalizados onde é permitido combinar gráficos, tabelas, métricas e logs em painéis configuráveis com visualização dinâmica, possui também alertas integrados onde torna-se possível a criação de alertas de visualização com envio de e-mails facilitando o monitoramento em tempo real (Grafana labs, 2024).

5 DESENVOLVIMENTO PRÁTICO

É importante antes da fase de execução do projeto, avaliar as necessidades onde o dispositivo irá trabalhar, visando atender os requisitos necessários, portanto a definição do MCU, quantidade de dados coletados, periodicidade, fonte de alimentação, intemperes, condicionamento de sinais necessários para a coleta de dados e tecnologias envolvidas.

5.1 DEFINIÇÃO DO MICROCONTROLADOR

A definição do microcontrolador é essencial para o sucesso do projeto, devendo considerar critérios como quantidade de entradas e saídas (I/O), presença de ADC, DAC, PWM e tipos de saídas digitais. Também é fundamental avaliar a capacidade de memória (Flash, RAM, EEPROM) de acordo com a complexidade do programa e volume de dados. Um bom dimensionamento garante eficiência, viabilidade e facilita futuras atualizações (Monk, 2017).

Quando se trabalha com microcontroladores e comunicação — independentemente do protocolo utilizado torna-se necessário o uso de interfaces apropriadas. Avaliar a compatibilidade dessas interfaces com os sistemas do projeto é um fator decisivo para o sucesso da aplicação (Monk, 2017).

Para este projeto foi selecionado um microcontrolador com as características necessárias para a transmissão de dados com uso de rádio LoRa SX1262 já integrado, visando tornar o projeto mais compacto atendendo todas as necessidades industriais.

O microcontrolador fabricado pela Heltec está na versão atual V3, onde diversas melhorias foram implementadas, reduzindo potenciais problemas durante a fase de desenvolvimento, a placa também possui diversas formas de se realizar comunicação, seja por WiFi 802.11 b/g/n, Bluetooth entre outras, o MCU possui as seguintes características fornecidas pelo (Heltec, 2025) (Tabela 1).

TABELA 1: Dados técnicos ESP32 LoRa SX1262 Heltec V3

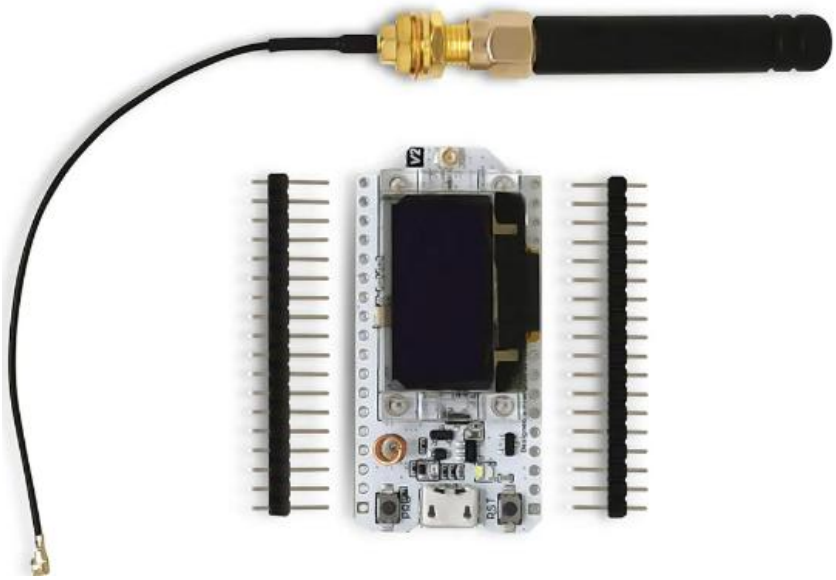
Parâmetro	Descrição
Processador (Master Chip)	ESP32 S3FN8 (Xtensa® 32 bit LX7, dual core, 240 MHz)
Módulo LoRa	SX1262
Conversor USB–Serial	CP2102
Faixa de Frequência LoRa	470–510 MHz; 863–928 MHz
Potência máxima de TX (LoRa)	21 ± 1 dBm
Sensibilidade de recepção (LoRa)	–134 dBm @ SF12 BW=125 kHz
Wi Fi	802.11 b/g/n – até 150 Mbps
Bluetooth	Bluetooth 5 (LE)
Memória interna	384 KB ROM;512 KB SRAM;16 KB RTC SRAM
Memória Flash	8 MB SiP Flash
Interfaces analógicas e periféricos	7 × ADC1, 2 × ADC2; 7 touch; 3 × UART; 2 × I²C; 2 × SPI; etc
Display embutido	OLED 0,96″ – 128×64 pixels
USB	USB C, proteção ESD e curto-circuito
Antena	2,4 GHz (Wi Fi/BLE) conector IPEX para LoRa
Temperatura de operação	–20 °C a +70 °C
Dimensões	50,2 × 25,5 × 10,2 mm

Fonte: Heltec, 2025.

O Esp32 LoRa SX1276 Heltec se mostra um microcontrolador ideal, tanto pelas características técnicas como também por questões de tamanho e configurações, reduzindo a necessidade de compra de módulos extras, o dispositivo possui antena LoRa integrada para o protocolo LoRaWAN ou a realização de comunicação ponto a ponto com rádio LoRa.

Apesar da grande vantagem de já vir projetado com o rádio LoRa integrado assim como o display OLED também se torna uma desvantagem, dependendo do projeto pois o pinout que é a configuração de pinos de fábrica, já possui configurações definidas como é o caso dos pinos do rádio LoRa e com o display, na Figura 13 abaixo os pinos do rádio LoRa está definido na legenda.

FIGURA 13: Esp32 SX1276



Fonte: Wjcomponentes, 2025.

5.2 TOPOLOGIA DE REDE

A Interoperabilidade no ambiente fabril é um desafio para implementação de monitoramento, já que há uma grande diversidade de formas de comunicação entre máquinas, realizar a comunicação de equipamentos onde se possui diversos fabricantes, dispositivos com tecnologias diferentes, também desenvolvidos em épocas diferentes é uma tarefa que os microcontroladores podem realizar com excelência.

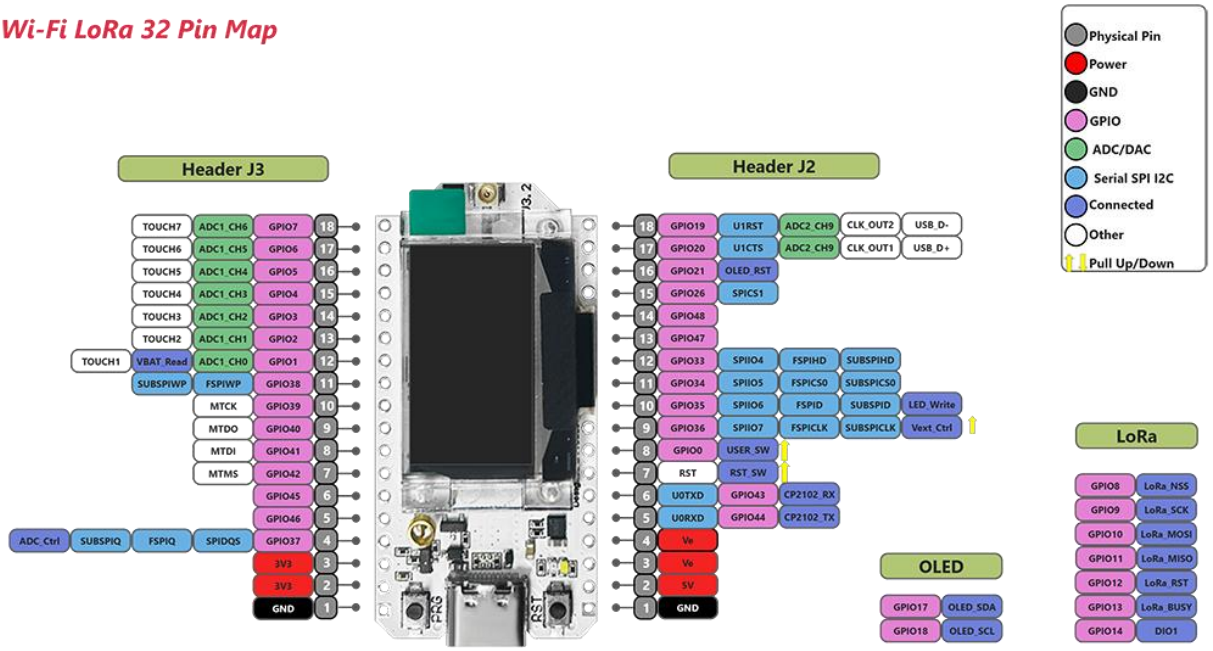
Há diversos módulos desenvolvidos para atender a comunicação e sensoriamento dos microcontroladores, tornando possível trabalhar com protocolos menos usados ou até com protocolos mais sofisticados.

Para simplificar e tornar mais prática a apresentação e a utilização dos microcontroladores equipados com rádio transmissor e receptor LoRa, foi utilizado o estado das próprias entradas digitais do microcontrolador para realizar o monitoramento de um sistema Chiller de resfriamento industrial, disponibilizando os

dados de funcionamento do equipamento em tempo real e o mais importante, o histórico de funcionamento, revezamento dos compressores e ações da operação como exemplo desligamento do equipamento nos períodos de não uso, caso seja necessário pode ser projetado o uso de outros protocolos de comunicação para somar com o LoRa, basta ter conhecimento dos *pinouts* do MCU.

É possível como exemplo realizar a comunicação com PLCs por meio de rede Ethernet, utilizando o módulo ENC28J60 (Figura 14).

FIGURA 14: Pinout Esp32 SX1276 Heltec



Fonte: Heltec, 2025.

Através do protocolo Modbus TCP, essa comunicação pode ser feita por meio de par trançado e conector RJ45 ou caso de necessidade com o módulo MAX485 pode ser utilizado o Modbus RTU com o protocolo RS-485 atendendo até 256 dispositivos.

Além disso, caso seja desejável, com o uso do módulo XY10 é possível realizar a leitura de dados provenientes de transdutores de pressão, temperatura, nível, entre outros sensores que operam com sinal de 4 a 20 mA, um padrão amplamente utilizado na indústria, tanto em sistemas de controle quanto em sistemas de monitoramento, sendo aplicável às mais diversas finalidades.

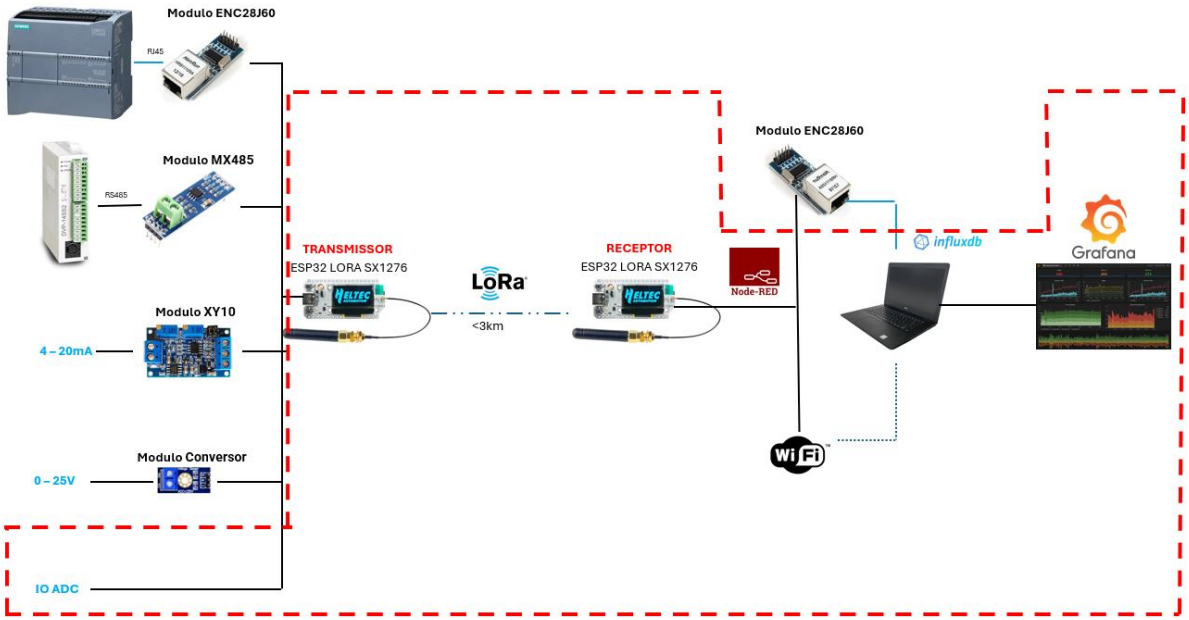
Outro sinal muito utilizado é o de tensão, geralmente na faixa de 0 a 10 V em corrente contínua. Essa faixa também é compatível com uma grande variedade de sensores e transdutores que utilizam esse intervalo para representar a variável medida.

Visando utilizar o mínimo de recursos possível e mesmo assim fazer a obtenção de dados valiosos será feito o uso apenas de entradas digitais do MCU onde o sinal de 3,5VDC (volts em corrente contínua) realiza a mudança do estado da entrada digital, sendo ela LOW (baixo) quando a tensão aplicada na entrada digital é 0 e HIGH (Alto) quando a tensão aplicada na entrada digital é de 3,5VDC.

Na topologia de rede destacada, são utilizados diversos dispositivos com funções específicas. O ADC lê entradas digitais (ligado/desligado). O MCU transmissor LoRa gerencia as leituras, configura o rádio LoRa, prepara o payload e exibe o status em um display OLED. O rádio LoRa realiza a transmissão de dados com alcance de até 3,5 km e baixa interferência.

O MCU receptor LoRa recebe, decodifica e armazena temporariamente os dados, exibe as informações em um display, retransmite via rede wireless para o Node-RED e registra dados via Modbus (funções 0x04 e 0x01). O Node-RED recebe os dados via Modbus TCP, processa e armazena no banco de dados InfluxDB, indexando por data e hora. Por fim o Grafana interpreta esses dados, apresentando gráficos em tempo real e permitindo análises e geração de relatórios (Figura 15).

FIGURA 15: Topologia de rede



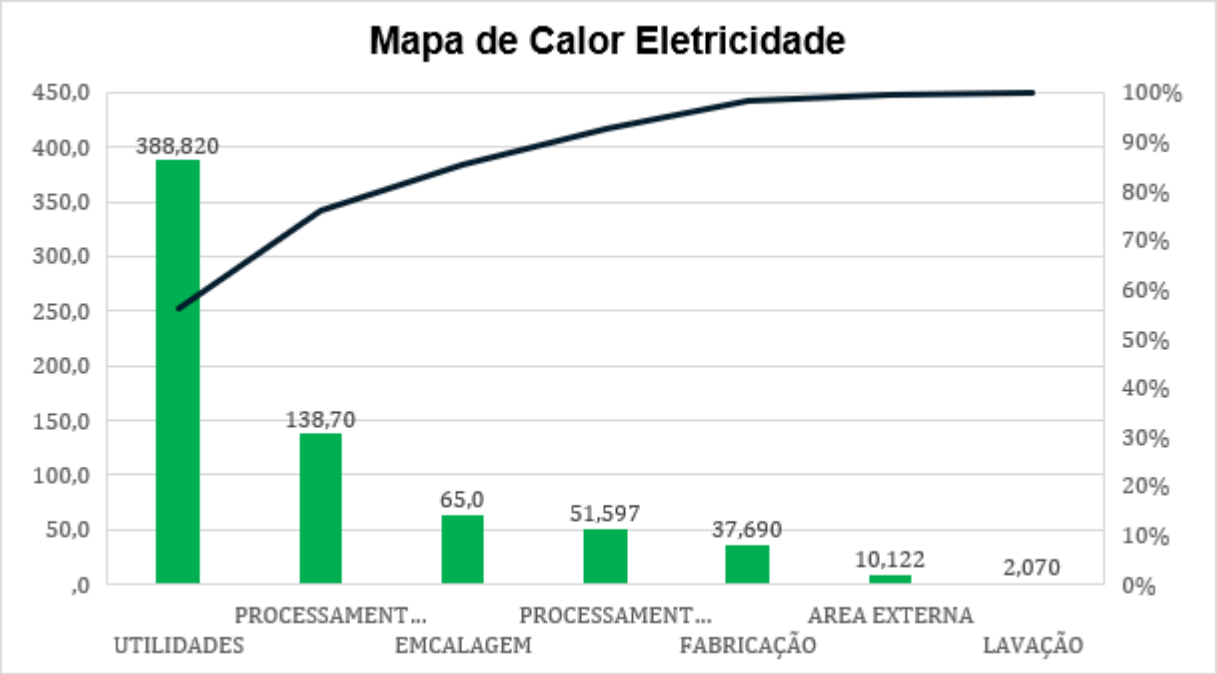
Fonte: Autor, 2025.

A integração de todas as interfaces é feita entre os diversos dispositivos onde o resultado é a leitura dos dados do equipamento onde é possível analisar os desvios, fazer modelagens matemáticas, cálculos de potência de consumo de energia, onde o equipamento monitorado é responsável por 30% do consumo de energia da unidade fabril, monitoramento e o controle do sistema torna-se essencial para a redução do consumo de energia elétrica, além de ser um equipamento com alto custo de manutenção.

5.3 DEFINIÇÃO DO PONTO A SER ANALISADO

A identificação do ponto a ser analisado é fundamental, sempre priorizando os maiores ofensores, ou seja, os processos com maior potencial de redução de custos e maior impacto no consumo de utilidades industriais. Por esse motivo, definiu-se que o equipamento com maior consumo energético e impacto econômico para a companhia seria o Chiller de água gelada, responsável pelo processo de resfriamento (Figura 16).

FIGURA 16: Mapa de calor, setores maiores ofensores de energia elétrica



Fonte: Autor, 2025.

O equipamento possui consumo variável, com estágios de carga que variam entre 25%, 50%, 75% e 100%, conforme a demanda térmica. Entretanto, mesmo no menor estágio de operação, que é 25% — quando apenas um dos dois compressores está em funcionamento — o consumo ainda é de 134,9 kW/h.

Por meio do monitoramento da utilização do equipamento, é possível analisar o tempo em que permanece ligado sem estar em produção, identificando assim oportunidades de economia de energia elétrica.

O equipamento conta também com um sistema de bombas com redundância onde as bombas devem ser operadas em revezamento, visando assim o controle do desgaste das bombas além de possuir sempre os equipamentos em condições de uso, o monitoramento auxilia na detecção de erro de operação, seja por falha na verificação das bombas, queda de energia e não religamento do equipamento, ou manobras feitas sem a permissão do setor de manutenção, por exemplo operar equipamento que está aguardando manutenção (Tabela 2).

TABELA 2: Dados Chiller WCC800

Categoria	Especificações Técnicas
Modelo	WCC800
Cap. Resfriamento	778 kW (a 35 °C ambiente, 12 °C entrada / 6 °C saída)
Temp. Fluido (min/máx)	–12 °C a +15 °C
Temp. Ambiente	–18 °C a +50 °C
Compressores	2 × CSH9563-210Y-40D (25–100% de carga variável)
Evaporador	Tipo casco e tubo; aço/cobre; capacidade: 500 L
Vazão de projeto	31,4 L/s (mín: 16 L/s | máx: 50 L/s)
Pres. máx. evaporador	10 bar
Perda carga evapor.	0,43 bar
Volume mín. do sistema	2000 L
Condensador	Tubos de cobre, aletas de alumínio, 10 ventiladores
Ventiladores	2,8 kW cada; fluxo total: 224.000 m³/h; 30 Pa máx
Refrigerante	R134a (2 circuitos); 125 kg por circuito
Óleo do compressor	Tipo BSE170; 30 L por circuito
Alimentação elétrica	380V–460V / 3 fases / 50–60 Hz
Corrente nominal	438 A
Corrente máx. (full load)	782 A
Corrente de partida	1200 A (com transição fechada) / 1800 A (sem transição)
Disjuntor principal	800 A
Conexão elétrica	Pino M18
Limite corrente ajust.	25–100%
Dimensões	6,058 m × 2,438 m × 2,591 m
Peso	Seco: 9050 kg / Operacional: 9650 kg
Conexões hidráulicas	4" Camlock ou 6" Flange

Fonte: Manual Chiller WCC800, 2025.

Com o monitoramento por meio do Grafana é possível evidenciar o tempo de funcionamento do compressor 01, compressor 02, bomba de água 01 à 04 e o sistema de ventilação dos condensadores, cada sinal dos componentes desse sistema possuem uma entrada digital no microcontrolador que ao ser acionada, realiza a mudança no estado da entrada, alternando entre ligado e desligado e no influxDB o Node-RED realiza o registro no banco de dados e atualizado a cada 5 segundos e exibindo os dados no Grafana (Figura 17).

FIGURA 17: Foto instalações Chiller de água gelada WCC800

Fonte: Autor, 2025.

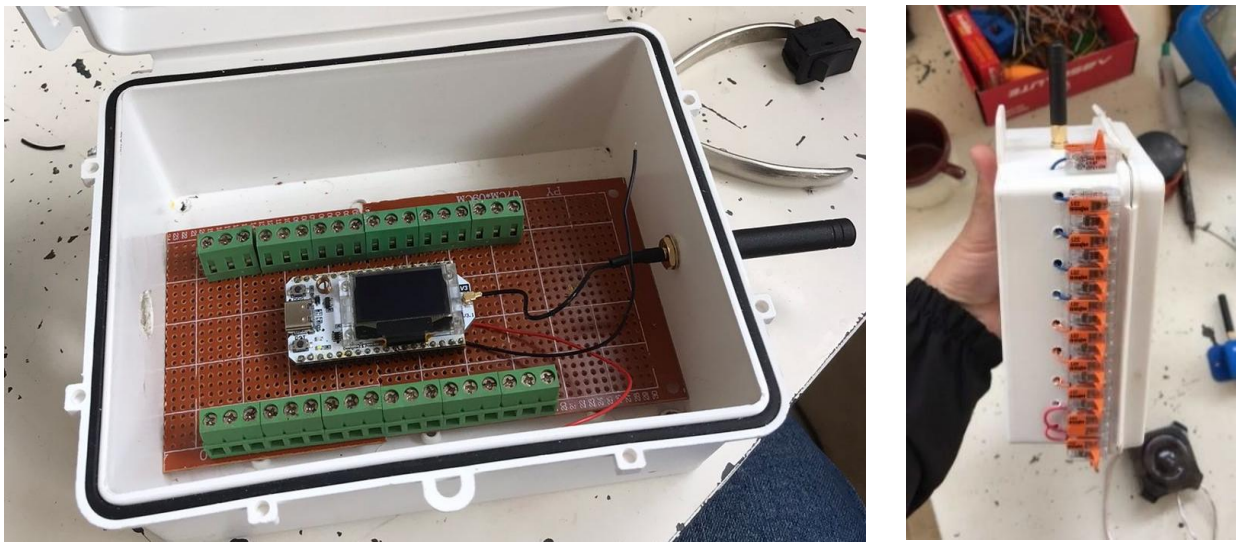
5.4 MONTAGEM DO CIRCUITO

A montagem dos MCUs requiriu que ambos possuíssem fontes para alimentação externa, apesar de consumirem pouca energia, como o recurso está disponível nos locais onde foram instalados, visando também a melhor transmissão dos dados sem perda de pacotes ou conflitos.

Conforme já falado, para que a implementação seja simples, porém com grande potencial, foi apenas feita a instalação monitorando os sinais digitais, sendo assim o transmissor instalado no painel do Chiller, alimentado pela fonte de alimentação do equipamento, para que seja simples a ligação dos cabos de sinais o MCU necessariamente foi instalado em uma placa perfurada onde os pinos ligados do lado externo da caixa de proteção.

Cada entrada digital é responsável por receber o sinal de 3,5VDC onde esse sinal é emitido com o fechamento geralmente de contatos auxiliares, relés de interface, ou podem ser utilizados qualquer sinal, porém deve ser atentar ao condicionamento do sinal, para que assim ele possa ser lido corretamente pelo MCU e retransmitido para o receptor (Figura 18).

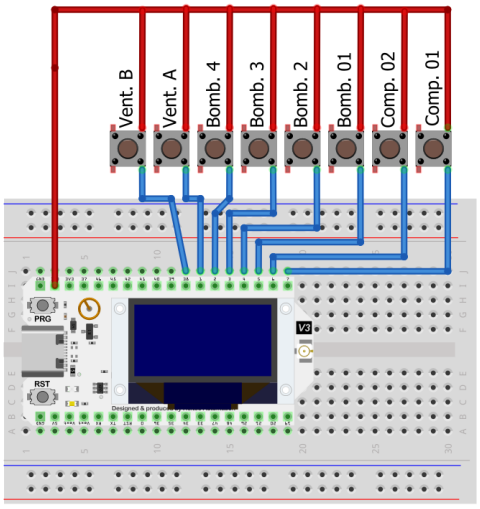
FIGURA 18: Caixa e proteção e ligação dos pinos com área externa



Fonte: Autor, 2025.

Um ponto importante é que entradas digitais podem gerar pulsos fantasmas, os quais precisam ser mitigados por meio de filtros digitais. No transmissor, essa mitigação é realizada através de um algoritmo que avalia a consistência do sinal, garantindo que a entrada digital represente um sinal verdadeiro. Para isso, o sinal é comparado por um curto período, sendo validado apenas se permanecer estável durante esse intervalo (Figura 19).

FIGURA 19: Esquema de ligação, botões simulando entradas



Fonte: Software Fritzing, 2025.

5.5 CIRCUITO DO RECEPTOR E CAIXA DE PROTEÇÃO

Já para o circuito de Recepção de dados, onde exerce a função de gateway não é necessário realizar nenhuma ligação, além da fonte de alimentação do dispositivo que já é feita via fonte 5VDC. Deve ser previsto proteção contra poeira ou umidade visando a durabilidade o equipamento, para o caso a fixação foi realizada por meio de imãs de neodímio em painéis elétricos (Figura 20).

FIGURA 20: Caixa de proteção, Esp32 modo Recepção

Fonte: Autor, 2025.

6 COLETA DAS INFORMAÇÕES

Na realização dos testes a potência de transmissão configurada no código fonte do transmissor é de 14dBm e a potência recebida no receptor foi de -90dBm, onde o transmissor estava instalado dentro do painel do equipamento, protegido de intemperes, realizando os cálculos para determinar a distância por meio da Equação 5 da perda no caminho (Haykin, 2008).

$$L_r = L_0 + 10 * n * log\left(\frac{r}{r_0}\right)$$

(Equação 5)

L_r = Perda média do caminho em dB na distância r

L_0 = perda em uma distância de referência r_0

r = Distância entre transmissor e receptor

r_0 = Distância de referência (tipicamente 1 m)

n = Coeficiente de perdas no meio

Determinação de L_r:

$$L_r = P_t - P_r$$

(Equação 6)

f = Frequência 915MHz

n = 2 – Espaço livre

n = 2,7 - 3.5 – Urbano

n = 3-5 – Ambiente interno com obstáculos

n = 1,5 – 2.0 – Ambientes rurais

Aplicação da Equação (6):

$$L_r = 14 - (-90)$$

$$L_r = 104\ dB$$

Isolando a distância r:

$$r = 10^{\left(\frac{L_r - L_0}{10 - n}\right)}$$

Há uma pequena perda de sinal por existir relevo no terreno, porém mesmo com o desnível a distância calculada é de 135,3 metros sem a realização da otimização dos parâmetros do rádio LoRa.

$$r = 10^{(2,133)} = 135,3\ m$$

O rádio LoRa mesmo sendo usado ponta a ponta possui grande eficiência na transmissão dos dados, com grande resistência a interferências eletromagnéticas

muito presente em ambientes fabris, o uso do SF aumenta a sensibilidade do rádio garantindo a entrega dos pacotes de dados.

Ainda pode ser feito testes com valores de SF menores, porém na ocasião a unidade fabril possui uma área relativamente pequena (Figura 21).

FIGURA 21: Exibição dos dados no Grafana

Fonte: Autor, 2025.

Usando um *Notebook* como servidor de dados em rede local é possível acompanhar o estado do equipamento, sinalizando qual compressor está ligado, se a operação está trabalhando com revezamento das bombas, revezamento dos compressores caso esteja em baixa demanda de frio, tempo de cada compressor ligado, se a ventilação está funcionando corretamente ou se o equipamento ficou ligado durante um tempo em que não há produção, onde é a ocorrência que mais impacta no custo de energia elétrica (Figura 22).

FIGURA 22: Distância entre transmissor e receptor real 115,12m.

Fonte: Google Maps, 2025.

7 CUSTOS

A Tabela 3 apresenta todos os componentes necessários para a implementação, juntamente com os custos associados ao sistema de monitoramento. Os valores cotados são baseados em preços de varejo, mas os custos podem ser reduzidos por meio de negociações em compras por atacado ou diretamente com fornecedores especializados ou por meio de importações.

Além de ser uma solução econômica, o sistema oferece fácil instalação, baixo consumo de energia, expansão simplificada e manutenção acessível, tornando-o uma opção altamente eficiente para o monitoramento proposto.

TABELA 3: Custos do projeto

CUSTOS DOS COMPONENTES				
ITEM	**DESCRIÇÃO DO ITEM**	**QUANT.**	**UNITÁRIO**	**SUBTOTAL**
1	MICROCONTROLADOR ESP32 LORA SX1276 HELTEC OLED V3	2	R$ 280,00	R$ 560,00
2	CAIXA HERMÉTICA MULTIUSO BABY BRANCA 14,5 X 10,5 X 7CM	1	R$ 12,90	R$ 12,90
3	CABO USB TIPO C	2	R$ 9,00	R$ 18,00
4	CARREGADOR TIPO C 5V PARA CELULAR	2	R$ 19,99	R$ 39,98
5	PLACA FENOLITE PERFURADA 20 X 10	2	R$ 18,00	R$ 36,00
6	CONECTOR BORNE 3 VIAS 5,08MM	12	R$ 2,61	R$ 31,32
TOTAL				**R$ 698,20**

Fonte: Valores baseados em Wjcomponentes, Unicaserv, 2025.

8 CONCLUSÃO

O desenvolvimento e aplicação do sistema de monitoramento permitiu não apenas a coleta em tempo real dos dados operacionais do equipamento, mas também a geração de informações confiáveis que foram fundamentais para a análise de desempenho e eficiência. Através do acompanhamento contínuo, foi possível identificar que o equipamento permaneceu em funcionamento por mais de 8 horas fora do horário regular de produção. Esse dado, por si só, revela um potencial significativo de desperdício de energia e, consequentemente, de aumento nos custos de utilidades da unidade.

A evidência levantada durante o monitoramento destaca a importância do controle inteligente de sistemas industriais. A simples permanência do equipamento ligado sem necessidade representa um impacto direto no consumo energético chegando nas 8 horas de funcionamento a 1.079,2 kW, em reais R$ 762,99 considerando o consumo mínimo do equipamento e a tarifa de energia atual, que poderia ser evitado com medidas de gestão mais rígidas.

O uso de tecnologias como IoT, comunicação LoRa e sistemas embarcados, como o ESP32, mostrou-se eficiente para realizar esse tipo de monitoramento de forma remota, segura e com baixo custo de implementação.

Além disso, o armazenamento e a visualização das informações coletadas permitem não apenas a tomada de decisão imediata, mas também a construção de um histórico que pode ser utilizado para auditorias, relatórios e ações de melhoria contínua.

Portanto, conclui-se que o projeto atingiu seus objetivos ao demonstrar, na prática, como o monitoramento pode ser utilizado como ferramenta estratégica para a redução de desperdícios, aumento da eficiência energética e otimização da operação de equipamentos. Com base nos dados registrados, é de boa pratica a implementação de rotinas automatizadas de desligamento e a criação de procedimentos internos que garantam o funcionamento dos equipamentos somente durante os períodos necessários.

Por fim, destaca-se que a metodologia utilizada neste trabalho pode ser expandida para outros ativos da planta industrial, possibilitando um controle mais amplo dos recursos e contribuindo significativamente para a sustentabilidade financeira e ambiental do processo produtivo.

Prospectando oportunidades futuras, a realização do monitoramento por meio de outros protocolos, como o Modbus RTU, possibilita que grande parte da unidade fabril seja monitorada com o uso de IoT. Além do monitoramento, destaca-se a viabilidade da implementação de sistemas de controle automático, como o intertravamento de sistemas, que impede o funcionamento dos equipamentos em situações inadequadas, de fato há muitas oportunidades para a implementação do sistema com ganhos significativos.

ABSTRACT
This work proposes the development of IoT (Internet of Things) devices for the industrial sector, focusing on monitoring industrial utilities using radio frequency as a means of data transmission. It seeks to demonstrate the application of long-distance communication with the LoRa radio transceiver. Data collection using this technology can reach up to 3 km, depending on the application area, such as remote regions with few buildings or open areas. In a practical way, the reliability of the data will be validated with the lowest possible continuous energy consumption. The approach aims to reduce waste and monitor deviations, with potential for application in various industrial consumptions, such as water, steam, firewood, wood chips, compressed air, among others, culminating in actual implementation in industrial processes.

Keywords: IoT. LoRa. Monitoring. Utilities. Industry.

UniAcademia
Centro Universitário

REFERÊNCIAS

ACSiP TECHNOLOGY CORP. **LoRa SiP Module S76G**. LoRa Alliance. Disponível em: https://lora-alliance.org/marketplace/acsip-technology-corp/lora-sip-module-s76g/. Acesso em: 24 abr. 2025.

AGÊNCIA BRASIL – EBC. **Relatório Conjuntura dos Recursos Hídricos no Brasil**: Informe Anual 2023. Brasília: Agência Nacional de Águas e Saneamento Básico (ANA); Agência Brasil – EBC, fev. 2024. Disponível em: https://agenciagov.ebc.com.br/noticias/202402/relatorio-conjuntura-dos-recursos-hidricos-no-brasil-atualiza-informacoes-sobre-aguas-do-pais. Acesso em: 01 jun. 2024.

BERTOLETI, Pedro. **Conectividade LoRaWAN**: fundamentos e prática. São Paulo: NCB, 2023.

BRAGA, Newton C. **Módulo Sub-G CMWX1ZZABZ da Murata (COMD013)**. 2017. Disponível em: https://www.newtoncbraga.com.br/novos-componentes/17716-modulo-sub-g-cmwx1zzabz-da-murata-comd013.html. Acesso em: 24 abr. 2025.

BRASIL. Agência Nacional de Telecomunicações (ANATEL). Ato nº 1.444, de 4 de agosto de 2017. Aprova o Regulamento para Certificação de Produtos de Telecomunicações. **Diário Oficial da União**, Brasília, DF, 7 ago. 2017. Disponível em: https://informacoes.anatel.gov.br/legislacao/atos-de-certificacao-de-produtos/2017/1139-ato-14448. Acesso em: 20 abr. 2025.

CAVLI WIRELESS. **Communication Protocols in IoT**: The Nerdiest of Things, 2025. Disponível em: https://www.cavliwireless.com/blog/nerdiest-of-things/communication-protocols-in-iot. Acesso em: 20 mar. 2025.

CESAR, Milton (Coord.); HADDAD, Jamil (Coord.). **Eficiência energética**: teoria e prática. Itajubá, 2007. 20 p.

CHENGDU EBYTE ELECTRONIC TECHNOLOGY CO., LTD. **E220-900T22D User Manual EN v1.0. 2020**. Disponível em: https://cdn.awsli.com.br/945/945993/arquivos/E220-900T22D_UserManual_EN_v1.0.pdf. Acesso em: 24 abr. 2025.

CICLOVIVO. **Brasil desperdiça R$71 milhões em energia por dia**. Disponível em: https://ciclovivo.com.br/planeta/energia/brasil-desperdica-r71-milhoes-em-energia-por-dia/. Acesso em: 12 jun. 2024.

COELHO, Darlene Figueiredo Borges. **Edifícios inteligentes**: uma visão das tecnologias aplicadas. São Paulo: Blucher, 2017.

EMBARCADOS. **Oito princípios para segurança em IoT**. Disponível em: https://embarcados.com.br/oito-principios-seguranca-em-iot/. Acesso em: 19 maio 2025.

EMPRESA DE PESQUISA ENERGÉTICA – EPE. **Relatório Síntese do Balanço Energético Nacional – BEN 2024**: ano base 2023. Rio de Janeiro: EPE, 2024. Disponível em: https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-819/topico-715/BEN_S%C3%ADntese_2024_PT.pdf. Acesso em: 01 jun. 2024.

ESPRESSIF. **Security Features** – ESP – IDF Programming Guide. Disponível em: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/security/security.html. Acesso em: 22 maio 2025.

FERMARC ROBÓTICA. **LoRa 915 MHz RF Módulo E220-900T22D (Sem Antena)**. Disponível em: https://www.fermarc.com/lora-915-mhz-rf-modulo-e220-900t22d-sem-antena?srsltid=AfmBOopMppijRS9b3MT0N75VlJggH74sTHvtAPdWmnEae5zq9fDxTwd1. Acesso em: 18 jun. 2025.

FRITZING. **Software de montagem**. Disponível em: https://fritzing.org/. Acesso em: 8 out. 2024.

FRUGAL PROTOTYPE. **AN1200.22 – LoRa™ Spread Spectrum** – A Technical Overview. [S. l.], ago. 2016. Disponível em: https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf. Acesso em: 12 mar. 2025.

GRAFANA LABS. **Grafana Documentation**. [S. l.], 2024. Disponível em: https://grafana.com/docs/grafana/latest/. Acesso em: 13 jun. 2025.

HAYKIN, Simon; MOHER, Michael. **Introdução aos sistemas de comunicação**. Tradução de Gustavo Guimarães Parma. 2. ed. Porto Alegre: Bookman, 2008.

HELTEC AUTOMATION. **WiFi LoRa 32** (V3). Disponível em: https://heltec.org/project/wifi-lora-32-v3/. Acesso em: 05 jun. 2025.

IMPRENSA EMNIFY. **Os quatro maiores desafios do mercado de IoT e como resolvê-los**. Indústria 4.0, 12 mar. 2024. Disponível em: https://www.industria40.ind.br/noticias/24802-os-quatro-maiores-desafios-mercado-iot-resolve-los. Acesso em: 15 maio 2025.

INFLUXDATA. **InfluxDB Core Concepts**. 2024. Disponível em: https://docs.influxdata.com/influxdb3/core/. Acesso em: 28 mar. 2025.

INPIXON. **Chirp Spread Spectrum RTLS, Location Tracking, & Positioning**. Disponível em: https://www.inpixon.com/technology/standards/chirp-spread-spectrum. Acesso em: 09 set. 2024.

IOT LABS. **LoRaONE** – Quadrar Tecnologia. Disponível em: https://iot-labs.io/portfolio/loraone-quadrar-tecnologia/. Acesso em: 18 jun. 2025.

LUZ, Sérgio Luiz da; ADAMSHUK, Rodrigo**. Automação e instrumentação industrial com Arduino**: teoria e projetos. 1. ed. Curitiba: Editora InterSaberes, 2021.

MAGRANI, Eduardo. **A internet das coisas**. 1. ed. Rio de Janeiro: FGV, 2018. 22 p.

MOKO SMART. **LoRa Technology**. Disponível em: https://www.mokosmart.com/lora-technology/. Acesso em: 18 jun. 2025.

MONK, Simon. **Programação com Arduino**: começando com sketches. 2. ed. Porto Alegre: Bookman, 2017. (Série Tekne).

NI – NATIONAL INSTRUMENTS. **The Modbus Protocol in Depth**. Disponível em: https://www.ni.com/en/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/the-modbus-protocol-in-depth.html. Acesso em: 05 abr. 2025.

NODE-RED. **About Node-RED**. 2024. Disponível em: https://nodered.org/about/. Acesso em: 13 dez. 2024.

PRÓSPERO, Israel. **Microcontroladores**: o que são e aplicações. São Carlos: EESC JR., 18 ago. 2020. Disponível em: https://eescjr.com.br/blog/microcontroladores-o-que-sao-e-aplicacoes/. Acesso em: 13 maio 2025.

SCHWAB, Klaus. **Quarta revolução industrial**. Tradução de Daniel Moreira. São Paulo: Edipro, 2016.

SEMTECH. **A Brief History of LoRa**: Three Inventors Share Their Personal Story at The Things Conference. Blog Semtech. Disponível em: https://blog.semtech.com/a-brief-history-of-lora-three-inventors-share-their-personal-story-at-the-things-conference. Acesso em: 18 mar. 2025.

THE THINGS NETWORK. **LoraWAN Architecture**. Disponível em: https://www.thethingsnetwork.org/docs/lorawan/architecture/. Acesso em: 29 set. 2024.

THE THINGS NETWORK. **Arquitetura Típica da rede Lora**. Disponível em: https://www.thethingsnetwork.org/docs/lorawan/architecture/. Acesso em: 15 jun. 2024.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO. **Grupo 12. Resumo de LoRa. 2022**. Disponível em: https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2022_2/grupo_12/resumo-de-lora/. Acesso em: 18 jun. 2025.

WEVOLVER. **Modbus TCP Explained**. 2023. Disponível em: https://www.wevolver.com/article/modbus-tcp. Acesso em: 12 fev. 2025.

UniAcademia
Centro Universitário

ANEXO A – CÓDIGO FONTE LORA RECEPTOR

```
#include "LoRaWan_APP.h"
#include "Arduino.h"
#include "HT_SSD1306Wire.h" // Lib para exibir dados no display OLED
#include <Wire.h>
#include "WiFi.h"
#include <ModbusIP_ESP8266.h> // Lib para comunicacao ModbusTCP
#define RF_FREQUENCY                                915000000 // Frequência LoRa em Hz
#define TX_OUTPUT_POWER                             14        // Potência de transmissão em dBm
#define LORA_BANDWIDTH                              0         // Largura de banda: 0 = 125kHz
#define LORA_SPREADING_FACTOR                       7         // Fator de espalhamento SF7
#define LORA_CODINGRATE                             1         // Taxa de codificação 4/5
#define LORA_PREAMBLE_LENGTH                        8         // Tamanho do preâmbulo
#define LORA_SYMBOL_TIMEOUT                         0         // Timeout de símbolo
#define LORA_FIX_LENGTH_PAYLOAD_ON                  false     // Payload de tamanho fixo desabilitado
#define LORA_IQ_INVERSION_ON                        false     // Inversão de IQ desabilitada
#define RX_TIMEOUT_VALUE                            1000      // Timeout de recepção
#define BUFFER_SIZE                                 30        // Tamanho máximo do pacote
// Buffers para envio e recepção de pacotes
char txpacket[BUFFER_SIZE];
char rxpacket[BUFFER_SIZE];
static RadioEvents_t RadioEvents;
ModbusIP mb; // Instância do Modbus
int16_t txNumber;
int16_t rssi, rxSize;
const int HREG01Temp   = 100; const int HCoilComp01  = 1; const int HCoilComp02  = 2;
const int HCoilB1      = 3; const int HCoilB2      = 4;   const int HCoilB3      = 5;
const int HCoilB4      = 6; const int HCoilV1      = 7;  const int HCoilV2      = 8;
// Variáveis auxiliares
long ts; // Timestamp para controle de tempo
const char* ssid     = "SSID";
const char* password = "Key";
float temperaturaAgua;
int comp1, comp2;
int bomba1, bomba2, bomba3, bomba4;
int vent1, vent2;
// Estado do LoRa
bool lora_idle = true;
// Instância do display OLED
static SSD1306Wire display(0x3c, 500000, SDA_OLED, SCL_OLED, GEOMETRY_128_64, RST_OLED);
IPAddress ipFixo(192, 168, 0, 6);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);
IPAddress dns(8, 8, 8, 8);
void setup() {
    Serial.begin(115200);
    Mcu.begin(HELTEC_BOARD, SLOW_CLK_TPYE);
    VextON();   // Liga o display OLED
    delay(100);
    WiFi.begin(ssid, password);
    if (!WiFi.config(ipFixo, gateway, subnet, dns)) {
        Serial.println("Falha ao configurar IP fixo!");
    }
    mb.addHreg(HREG01Temp); mb.addCoil(HCoilComp01);mb.addCoil(HCoilComp02);
    mb.addCoil(HCoilB1);mb.addCoil(HCoilB2);mb.addCoil(HCoilB3);mb.addCoil(HCoilB4);
    mb.addCoil(HCoilV1);mb.addCoil(HCoilV2);
    ts = millis(); // Inicia contador de tempo
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("WiFi conectado");
    Serial.println(WiFi.localIP());
    mb.server(); // Inicia servidor Modbus TCP
```

```
    // Configurações LoRa
    txNumber = 0;
    rssi = 0;
    RadioEvents.RxDone = OnRxDone;
    Radio.Init(&RadioEvents);
    Radio.SetChannel(RF_FREQUENCY);
    Radio.SetRxConfig(MODEM_LORA, LORA_BANDWIDTH, LORA_SPREADING_FACTOR, LORA_CODINGRATE, 0, LORA_PREAMBLE_LENGTH,
    LORA_SYMBOL_TIMEOUT, LORA_FIX_LENGTH_PAYLOAD_ON, 0, true, 0, 0, LORA_IQ_INVERSION_ON, true);
    // Inicialização do display OLED
    display.init();
    display.setFont(ArialMT_Plain_10);
    display.setTextAlignment(TEXT_ALIGN_CENTER);
}
void VextON(void){
  pinMode(Vext, OUTPUT);
  digitalWrite(Vext, LOW);
}
void VextOFF(void) {
  pinMode(Vext, OUTPUT);
  digitalWrite(Vext, HIGH);
}
void loop() {
  if (lora_idle) {
    lora_idle = false;
    Radio.Rx(0); }
  Radio.IrqProcess();
  // Processa tarefas Modbus
  mb.task();
  if (WiFi.status() == WL_CONNECTED) {
    display.drawString(64, 35,  "WiFi Conectado!");}

  // Atualiza dados a cada 2 segundos
  if (millis() > ts + 2000) {
    ts = millis();
    parseDados(rxpacket);
    Serial.print("Temperatura: "); Serial.println(temperaturaAgua);
    Serial.print("Compressor1: "); Serial.println(comp1);Serial.print("Compressor2: "); Serial.println(comp2);
    Serial.print("Bomba1: "); Serial.println(bomba1);Serial.print("Bomba2: "); Serial.println(bomba2);
    Serial.print("Bomba3: "); Serial.println(bomba3);Serial.print("Bomba4: "); Serial.println(bomba4);
    Serial.print("Ventilacao1: "); Serial.println(vent1);Serial.print("Ventilacao2: "); Serial.println(vent2);
    // Atualiza registradores Modbus
    mb.Hreg(HREG01Temp, (int)(temperaturaAgua * 100));mb.Coil(HCoilComp01, comp1);mb.Coil(HCoilComp02, comp2);
    mb.Coil(HCoilB1, bomba1);mb.Coil(HCoilB2, bomba2);mb.Coil(HCoilB3, bomba3);mb.Coil(HCoilB4, bomba4);
    mb.Coil(HCoilV1, vent1);mb.Coil(HCoilV2, vent2);
    delay(10);
  }
}
// Função chamada quando um pacote é recebido via LoRa
void OnRxDone(uint8_t *payload, uint16_t size, int16_t rssiParam, int8_t snr) {
  rssi = rssiParam;
  rxSize = size;
  memcpy(rxpacket, payload, size);
  rxpacket[size] = '\0';
  Radio.Sleep();
  Serial.printf("\r\nPacote recebido \"%s\" com RSSI %d, tamanho %d\r\n", rxpacket, rssi, rxSize);
  lora_idle = true;
  // Exibe dados no OLED
  display.drawString(64, 5,  "Temperatura: " + String(rxpacket[0]));
  display.drawString(64, 15, "IO: " + String(rxpacket));
  display.drawString(64, 25, "Tamanho: " + String(rxSize) + " Bits");
  display.drawString(64, 45, "Potência: " + String(rssi) + "dBm");
  display.display();
  display.clear();
}
// Faz parsing do pacote recebido para extrair os dados
void parseDados(const char* dados) {
  sscanf(dados, "%f,%d,%d,%d,%d,%d,%d,%d,%d,%d",
         &temperaturaAgua,&comp1, &comp2,&bomba1, &bomba2, &bomba3, &bomba4,&vent1, &vent2);
}
```

UniAcademia
Centro Universitário

ANEXO B – CODIGO FONTE LORA TRANSMISSOR

```
#include "LoRaWan_APP.h"
#include "Arduino.h"
#include "HT_SSD1306Wire.h" // Biblioteca para display OLED
#include <Wire.h>
#include "WiFi.h"
// Definições para transmissão LoRa
#define RF_FREQUENCY 915000000 // Frequência LoRa em Hz
#define TX_OUTPUT_POWER 5 // Potência de saída em dBm
#define LORA_BANDWIDTH 0 // Largura de banda: 0 = 125kHz
#define LORA_SPREADING_FACTOR 7 // Fator de espalhamento SF7
#define LORA_CODINGRATE 1 // Taxa de codificação 4/5
#define LORA_PREAMBLE_LENGTH 8 // Comprimento do preâmbulo
#define LORA_SYMBOL_TIMEOUT 0 // Tempo limite de símbolo
#define LORA_FIX_LENGTH_PAYLOAD_ON false
#define LORA_IQ_INVERSION_ON false
#define RX_TIMEOUT_VALUE 1000 // Tempo limite de recepção
#define BUFFER_SIZE 30 // Tamanho do buffer de dados
char txpacket[BUFFER_SIZE];
char rxpacket[BUFFER_SIZE];
// Parâmetros do filtro digital
#define NUM_AMOSTRAS 50 // Número de amostras para filtro digital
#define TEMPO_AMOSTRA 15 // Tempo entre amostras em ms
// Entradas para monitoramento de equipamentos
float Temperatura = A6; // Simulação de leitura de temperatura
const int Compressor01 = 7;const int Compressor02 = 6;const int Bomba01 = 5;
const int Bomba02 = 4;const int Bomba03 = 3;const int Bomba04 = 2;
const int Ventilacao01 = 1;const int Ventilacao02 = 38;
const int ledPin = LED_BUILTIN; // Pino do LED onboard
bool lora_idle = true;
// Eventos do rádio LoRa
static RadioEvents_t RadioEvents;
void OnTxDone(void);
void OnTxTimeout(void);
// Configuração do display OLED
static SSD1306Wire display(0x3c, 500000, SDA_OLED, SCL_OLED, GEOMETRY_128_64, RST_OLED);
// Variável de controle de tempo
unsigned long lastSwitchTime = 0;
void setup() {
    Serial.begin(115200); // Inicializa comunicação serial
    VextON(); // Liga alimentação externa
    delay(100);
    Mcu.begin(HELTEC_BOARD, SLOW_CLK_TPYE); // Inicializa placa Heltec

    randomSeed(analogRead(0));
    pinMode(ledPin, OUTPUT); // Define LED como saída

    // Define os pinos de entrada
    pinMode(Compressor01, INPUT);pinMode(Compressor02, INPUT);pinMode(Bomba01, INPUT);
    pinMode(Bomba02, INPUT);pinMode(Bomba03, INPUT);pinMode(Bomba04, INPUT);
    pinMode(Ventilacao01, INPUT);pinMode(Ventilacao02, INPUT);
   // Configuração de eventos do rádio
    RadioEvents.TxDone = OnTxDone;
    RadioEvents.TxTimeout = OnTxTimeout;
    Radio.Init(&RadioEvents);
    Radio.SetChannel(RF_FREQUENCY);
    Radio.SetTxConfig(
        MODEM_LORA,TX_OUTPUT_POWER,0, LORA_BANDWIDTH, LORA_SPREADING_FACTOR,
        LORA_CODINGRATE,LORA_PREAMBLE_LENGTH,LORA_FIX_LENGTH_PAYLOAD_ON,true, 0, 0,
        LORA_IQ_INVERSION_ON,3000);
    // Inicializa display
    display.init();
```

```
    display.setFont(ArialMT_Plain_10);}
// Liga alimentação externa
void VextON(void) {
    pinMode(Vext, OUTPUT);
    digitalWrite(Vext, LOW);}
// Desliga alimentação externa
void VextOFF(void) {
    pinMode(Vext, OUTPUT);
    digitalWrite(Vext, HIGH);
}
void loop() {
    unsigned long currentTime = millis();
    // Simula valor de temperatura
    float temperaturaAgua = -1.5;
    // Filtra valores digitais dos equipamentos
    bool compressor1 = filtrarValorDigital(Compressor01); bool compressor2 = filtrarValorDigital(Compressor02);
    bool ventilacao1 = filtrarValorDigital(Ventilacao01);bool ventilacao2 = filtrarValorDigital(Ventilacao02);
    bool bomba1 = filtrarValorDigital(Bomba01);bool bomba2 = filtrarValorDigital(Bomba02);
    bool bomba3 = filtrarValorDigital(Bomba03);bool bomba4 = filtrarValorDigital(Bomba04);
    // Imprime os valores no monitor serial
    Serial.println("Temperatura: " + String(temperaturaAgua));
    Serial.println("Compressor 01: " + String(compressor1));
    Serial.println("Compressor 02: " + String(compressor2));
    Serial.println("Bomba 1: " + String(bomba1));
    Serial.println("Bomba 2: " + String(bomba2));
    Serial.println("Bomba 3: " + String(bomba3));
    Serial.println("Bomba 4: " + String(bomba4));
    Serial.println("Ventilação A: " + String(ventilacao1));
    Serial.println("Ventilação B: " + String(ventilacao2));
    delay(1000); // Aguarda 1 segundo
    // Envia dados via LoRa
    if (lora_idle == true) {
        sprintf(txpacket, "%.2f,%d,%d,%d,%d,%d,%d,%d,%d",
            temperaturaAgua,compressor1,compressor2,bomba1,bomba2,
            bomba3,bomba4,ventilacao1,ventilacao2);
        Serial.printf("\r\nEnviando pacote \"%s\" , tamanho %d\r\n", txpacket, strlen(txpacket));
        Radio.Send((uint8_t *)txpacket, strlen(txpacket));

        // Mostra os dados no display OLED
        display.drawString(0, 15, "IO: " + String(txpacket));
        display.display();
        delay(500);
        display.clear();

        lora_idle = false;
    }

    Radio.IrqProcess(); // Processa interrupções do rádio
}
void OnTxDone(void) {
    Serial.println("Transmissão concluída.");
    lora_idle = true;
}
void OnTxTimeout(void) {
    Radio.Sleep();
    Serial.println("Tempo excedido para transmissão.");
    lora_idle = true;
}
// Função para aplicar filtro digital em entradas
bool filtrarValorDigital(int pino) {
    for (int i = 0; i < NUM_AMOSTRAS; i++) {
        if (digitalRead(pino) != 1) {
            return false; // Se alguma leitura for 0, retorna false
        }
        delay(TEMPO_AMOSTRA);
    }
    return true; // Retorna true se todas forem 1
}
```