

Sistema Embarcado de Monitoramento Industrial de Baixo Custo Utilizando Kit BitDogLab, MQTT e Node-RED

Leandro Ariel A. do Nascimento
Residência Embarcatech
Instituto Federal do Maranhão
leandroariel@acad.ifma.edu.br

Joanne Bruna da S. Santos
Residência Embarcatech
Instituto Federal do Maranhão
brunajoanne@acad.ifma.edu.br

Abellard Christley Mainviel
Residência Embarcatech
Instituto Federal do Maranhão
mainvielc@acad.ifma.edu.br

Tatiana Izidio de Oliveira
Residência Embarcatech
Instituto Federal do Maranhão
tatiana.i@acad.ifma.edu.br

Danúbia Soares Pires
Residência Embarcatech
Instituto Federal do Maranhão
danubiapires@ifma.edu.br

Almir Souza e S. Neto
Residência Embarcatech
Instituto Federal do Maranhão
almir.neto@ifma.edu.br

Orlando Donato Rocha Filho
Residência Embarcatech
Instituto Federal do Maranhão
orlando.rocha@ifma.edu.br

Washington Luis Santos Silva
Residência Embarcatech
Instituto Federal do Maranhão
prof.washington.silva@acad.ifma.edu.br

Claudio Leão Torres
Residência Embarcatech
Instituto Federal do Maranhão
claudioleao@ifma.edu.br

Jorge Renato dos Santos Silva
Residência Embarcatech
Instituto Federal do Maranhão
jorger@ifma.edu.br

Abstract - This study details the development of a low-cost integrated industrial monitoring system, primarily aimed at meeting the needs of small and medium-sized industrial enterprises. The proposed solution utilizes the BitDogLab kit with the Raspberry Pi Pico W, the MQTT communication protocol, and the Node-RED platform. This system enables the implementation of predictive maintenance strategies through continuous observation of critical variables such as temperature, vibration, and noise levels. The integration of accessible educational hardware with the Node-RED platform—for creating supervisory interfaces and dashboards—and the MQTT protocol for data exchange, demonstrates a practical approach to process automation. This drives the modernization and efficiency of industrial operations in the Industry 4.0 era.

Index Terms— Embedded System, Industrial Monitoring, Predictive Maintenance, SCADA, Internet of Things, MQTT, Node-RED.

I. INTRODUÇÃO

A Indústria 4.0 representa uma profunda transformação digital no setor industrial, impulsionada por tecnologias avançadas como Internet das Coisas (IoT), Inteligência Artificial (IA), Big Data e automação [1]. Este paradigma acentua a criticidade de sistemas de monitoramento em tempo real para otimizar a eficiência operacional e a confiabilidade dos processos industriais. Nesse contexto, a manutenção emerge como um pilar fundamental para assegurar a continuidade operacional e a máxima disponibilidade dos ativos na planta industrial [2]. O monitoramento em tempo real possibilita uma avaliação

constante do desempenho dos equipamentos, sendo crucial para a prevenção de falhas, a mitigação de interrupções não planejadas e o subsídio à tomada de decisões e estratégias de manutenção preditiva, visando a redução de perdas produtivas [3], [4].

A manutenção preditiva, conforme preconizada pela ABNT NBR 5462, fundamenta-se na aplicação sistemática de técnicas de análise de dados para diagnosticar o estado funcional dos equipamentos e antever a iminência de falhas, empregando para tanto meios de supervisão centralizados ou por amostragem [5]. Seu objetivo é, portanto, a antecipação de falhas e o prolongamento da vida útil de componentes e instalações, frequentemente, por meio de sistemas supervisórios que proveem funcionalidades de inspeção, supervisão e controle.

Contudo, a problemática central que se observa reside no fato de que os sistemas de supervisão e controle de processos industriais disponíveis comercialmente, embora robustos, frequentemente implicam custos de aquisição, implementação e manutenção significativos [6]. Tais custos podem se configurar como uma barreira considerável, especialmente para pequenas e médias empresas (PMEs), dificultando a adoção de tecnologias habilitadoras da Indústria 4.0 e, por conseguinte, a implementação de estratégias eficazes de manutenção preditiva. Esta lacuna tecnológica e econômica evidencia a necessidade de soluções alternativas de baixo custo que democratizem o acesso a sistemas avançados de monitoramento.

Nesse sentido, a literatura científica e técnica tem explorado diversas abordagens para o monitoramento industrial e manutenção preditiva, muitas vezes com foco em reduzir custos ou aplicar novas tecnologias. Corrêa e Almeida [7] investigaram o uso de redes neurais artificiais no monitoramento de evaporadores industriais, evidenciando a capacidade de modelos baseados em dados históricos para detectar anomalias em processos químicos. De forma mais ampla, Gouvêa et al. [8] apresentaram o Sistema de Monitoramento e Diagnóstico Online (SMDO), uma plataforma robusta para o diagnóstico de falhas em plantas industriais de grande porte, integrando dados de diversos níveis da automação. A aplicação direta da IoT no monitoramento de equipamentos elétricos foi demonstrada por Fabrício et al. [9], que desenvolveram um sistema para coleta e processamento de dados de corrente elétrica em tempo real, com envio para uma plataforma IoT para detecção de desvios. Similarmente, Oliveira et al. [10] propuseram um sistema IoT para manutenção preditiva focado na análise de assinatura de corrente. Em outras frentes, Azevedo [11] implementou um sistema de monitoramento de disponibilidade para sistemas de água quente industrial utilizando Controladores Lógicos Programáveis (CLPs) e comunicação OPC, enquanto Reis [12] focou no desenvolvimento de um sistema de monitoramento de eficiência energética para refrigeração industrial. Farias [13] explorou a tecnologia LoRa®, uma rede de baixa potência e longo alcance (LPWAN), como alternativa para a coleta de parâmetros em processos industriais, visando baixo custo e consumo energético.

Embora essas contribuições sejam significativas e demonstrem a viabilidade de diferentes tecnologias para o monitoramento industrial, muitas das soluções ainda podem apresentar complexidades de implementação ou custos que, embora menores que os sistemas comerciais tradicionais, podem não ser ideais para PMEs com orçamentos extremamente limitados ou para fins educacionais e de prototipagem rápida. Adicionalmente, a sinergia entre plataformas de desenvolvimento de baixíssimo custo, como microcontroladores da família Raspberry Pi Pico, com ferramentas de desenvolvimento visual e protocolos abertos para IoT industrial ainda é um campo com oportunidades de exploração. Diante deste panorama, o presente trabalho se distingue ao propor uma solução de monitoramento industrial que prioriza a acessibilidade e a facilidade de implementação. Neste cenário, o emprego de sistemas embarcados, aliado a ferramentas de desenvolvimento visual como o Node-RED [14] e protocolos de comunicação leves como o MQTT (Message Queuing Telemetry Transport) [6], [15], emerge como uma alternativa promissora para a criação de soluções acessíveis de supervisão e aquisição de dados, fomentando o desenvolvimento de tecnologias de automação e manutenção preditiva. Dentro deste escopo, o kit didático BitDogLab [16], equipado com a Raspberry Pi Pico W, apresenta-se como uma plataforma viável e de baixo custo para a prototipagem e desenvolvimento de sistemas de aquisição de dados em contextos industriais simulados ou de pequena escala [17], [18]. Este trabalho

propõe, assim, o desenvolvimento de um sistema embarcado de monitoramento industrial de baixo custo utilizando o kit BitDogLab para aquisição de dados, comunicação via protocolo MQTT, e interfaces de supervisão desenvolvidas em Node-RED para visualização e análise de dados em tempo real.

II. REFERENCIAL TEÓRICO

A. Sistemas Embarcados

Sistemas embarcados são sistemas computacionais dedicados a funções específicas, muitas vezes com requisitos de processamento em tempo real, integrados como parte de um sistema maior. Diferentemente dos computadores de propósito geral, eles são projetados e otimizados para tarefas particulares, operando sob restrições de custo, tamanho, consumo de energia e desempenho [19].

A presença de sistemas embarcados no cotidiano é onipresente e fundamental, moldando a forma de interação com a tecnologia e impulsionando a inovação em diversos setores. Eles são a espinha dorsal de inúmeros dispositivos utilizados diariamente, desde smartphones e wearables até eletrodomésticos inteligentes, veículos autônomos, equipamentos médicos e infraestruturas industriais [20], [21]. Esses sistemas estão presentes na automação industrial, dispositivos médicos, eletrodomésticos, veículos e na Internet das Coisas (IoT) [22]. A arquitetura de um sistema embarcado inclui um microcontrolador ou microprocessador, memória, interfaces de entrada/saída (I/O), sensores e atuadores específicos para a aplicação [23].

O software embarcado, frequentemente desenvolvido em linguagens como C ou C++, é crucial para gerenciar o hardware de forma eficiente e atender às restrições de tempo e energia [3]. A conectividade sem fio, como Wi-Fi e Bluetooth, é uma tendência crescente – especialmente em aplicações IoT – permitindo que sistemas embarcados se comuniquem com outros dispositivos e servidores em nuvem para monitoramento remoto, controle e coleta de dados [4], [24].

A constante evolução tecnológica e a crescente demanda por soluções inteligentes e conectadas impõem novos desafios e impulsionam tendências na área de sistemas embarcados. Um dos principais focos é a segurança e a confiabilidade. Com a interconexão crescente desses sistemas, a proteção contra ataques cibernéticos e a garantia de operação contínua tornam-se primordiais, especialmente em aplicações críticas como veículos autônomos e dispositivos médicos [25], [26]. A complexidade inerente a esses sistemas – que combinam hardware e software altamente especializados – exige metodologias de design robustas e rigorosos processos de validação e verificação para assegurar sua integridade e desempenho [27].

Outro ponto relevante é a evolução das plataformas de desenvolvimento para sistemas embarcados. Ferramentas e ambientes de programação que facilitam a prototipagem rápida e a depuração de código são cruciais para acelerar o ciclo de desenvolvimento e reduzir custos [30]. A popularização de plataformas de hardware de baixo custo e código aberto, como o Raspberry Pi e o Arduino, tem democratizado o acesso à tecnologia embarcada, permitindo que um leque mais amplo

de desenvolvedores e entusiastas experimente e crie soluções inovadoras [31].

A Indústria 4.0 e a IoT continuam sendo grandes impulsionadoras da evolução dos sistemas embarcados. Nesses cenários, a capacidade de coletar e transmitir dados de forma eficiente, bem como a interoperabilidade entre diferentes dispositivos e plataformas, são aspectos fundamentais. Isso exige que os sistemas embarcados sejam projetados com flexibilidade e escalabilidade em mente, aptos a se adaptar a novas tecnologias e padrões de comunicação [32], [33].

Diante desse cenário de constante evolução, a área de sistemas embarcados enfrenta a necessidade premente de novos projetos que foquem na inovação, explorando novas arquiteturas, algoritmos e materiais e, crucialmente, que possibilitem o desenvolvimento de tecnologias de baixo custo. Isso é vital para a democratização do acesso à tecnologia e para a expansão da Indústria 4.0, tornando-a viável para um número maior de empresas e aplicações [34], [35].

B. Kit BitDogLab

O Kit BitDogLab é uma plataforma de desenvolvimento educacional criada no âmbito do Projeto Escola 4.0 da Universidade Estadual de Campinas (UNICAMP), com foco no ensino de eletrônica, sistemas embarcados, IoT e ciência de dados. Essa plataforma tem sido utilizado em iniciativas como o Programa Embarcatech, coordenado pela Softex [17], [36]. Entre suas principais vantagens destacam-se o caráter didático, a modularidade, o baixo custo e a facilidade de integração com aplicações de IA e IoT, além de ampla documentação e suporte comunitário [18].

Dentre as limitações observadas, encontram-se a precisão dos sensores para aplicações industriais críticas e os recursos computacionais do microcontrolador RP2040. O kit é compatível com as placas Raspberry Pi Pico e Pico W (RP2040) e conta com uma placa de expansão dotada de conectores JST, que facilitam a conexão de sensores e atuadores. Entre os sensores integrados ou acopláveis estão microfone, sensor de temperatura/umidade, luz e vibração; os atuadores incluem LEDs, buzzer e botão. A programação pode ser realizada em C/C++ (utilizando o Pico SDK) ou MicroPython, com suporte a redes Wi-Fi no modelo Pico W [16].

O microcontrolador RP2040 presente na Raspberry Pi Pico W é equipado com um processador Dual-core ARM Cortex-M0+ até 133 MHz, 264 KB de RAM interna e diversas interfaces de comunicação, como I2C, SPI, UART, ADC e PWM [3], [4].

C. Protocolo MQTT

O MQTT (Message Queuing Telemetry Transport) é um protocolo leve de mensagens, baseado no modelo publicar/assinar (*publisher/subscriber*), sendo ideal para comunicação em redes com largura de banda restrita ou para dispositivos com recursos computacionais limitados, como os frequentemente utilizados em aplicações IoT [3], [4], [15]. Sua concepção minimalista o

diferencia de outros protocolos, tornando-o particularmente adequado para cenários que exigem eficiência energética e baixo consumo de dados[37].

A arquitetura do MQTT representa uma solução apropriada para microcontroladores como o Raspberry Pi Pico W, devido à simplicidade de implementação, à ampla disponibilidade de bibliotecas escritas em C otimizadas para sistemas embarcados e ao baixo uso de recursos. Sua eficiência operacional possibilita uma comunicação robusta, mesmo em redes com alta latência ou baixa confiabilidade – características comuns em ambientes IoT [38].

A comunicação ocorre por meio de um servidor central denominado *broker*, responsável pelo gerenciamento e distribuição das mensagens entre os clientes. Os clientes publicam mensagens em tópicos específicos, enquanto outros clientes, previamente inscritos nesses tópicos, recebem as mensagens correspondentes [3]. Essa abordagem de desacoplamento entre publicadores e assinantes aumenta a escalabilidade e flexibilidade do sistema [39].

O protocolo MQTT oferece três níveis de Qualidade de Serviço (*Quality of Service – QoS*), que garantem diferentes graus de confiabilidade na entrega das mensagens [3], [24]: QoS 0 ("No máximo uma vez"), QoS 1 ("pelo menos uma vez") e QoS 2 ("Exatamente uma vez"). Essa flexibilidade permite a adaptação do protocolo a diversas aplicações, desde a simples leitura de sensores até comandos críticos para atuadores [40].

Neste projeto, o *broker* utilizado é o Mosquitto, uma solução *open-source*, gratuita, leve e de fácil instalação e configuração. Trata-se de uma alternativa adequada tanto para ambientes de desenvolvimento e testes quanto para implantações de pequeno e médio porte. Desenvolvido em C/C++, o Mosquitto oferece suporte completo às especificações MQTT 3.1.1 e 5.0, além de funcionalidades como autenticação e criptografia via TLS/SSL, garantindo a segurança das comunicações [41], [42]. A comunidade ativa em torno do Mosquitto assegura atualizações constantes e suporte técnico abrangente.

D. Node-RED

O Node-RED é uma ferramenta de programação baseada em fluxo, originalmente desenvolvida pela IBM, que permite a integração de dispositivos de hardware, APIs e serviços online de forma intuitiva [3], [14]. Sua interface gráfica, baseada em operações de arrastar e soltar (*drag-and-drop*), facilitam o desenvolvimento de aplicações complexas, inclusive por usuários sem formação aprofundada em programação tradicional [43].

Essa ferramenta é amplamente adotada em projetos IoT para implementação de lógicas de automação, processamento de dados e construção de painéis de visualização (*dashboards*) personalizados, possibilitando a supervisão e o controle em tempo real de sistemas [3], [44], [45]. O Node-RED pode operar como cliente MQTT, publicando e subscrevendo mensagens, além de servir como interface de desenvolvimento de lógicas de controle e visualização de dados [3].

Sua ampla biblioteca de nós pré-configurados acelera significativamente o desenvolvimento de aplicações, ao simplificar a integração entre diferentes tecnologias [47]. Devido ao seu caráter de código aberto e sua versatilidade, o

Node-RED é frequentemente escolhido para prototipagem rápida e implementação de sistemas complexos, aplicando-se desde soluções residenciais até ambientes industriais [48], [49].

E. Análise Preditiva em Ambientes de Baixo Código com Node-RED

A implementação de estratégias de manutenção preditiva em ambientes industriais é frequentemente limitada pela complexidade da instrumentação e pela exigência de conhecimento especializado em ciência de dados. Nesse contexto, plataformas de baixo código (*low-code*) voltadas à IoT, como o Node-RED, destacam-se como ferramentas relevantes para a prototipagem e implementação ágil de soluções de monitoramento [50].

Nesse cenário, o módulo *node-red-contrib-industrial-machine-forecast* ganha destaque, pois reflete a democratização da análise preditiva ao encapsular um modelo de suavização exponencial para previsão de séries temporais e detecção de anomalias em dados industriais, uma vez que não apenas fornece uma previsão do próximo valor da série, como também calcula, de forma dinâmica, as bandas de confiança superior e inferior, estabelecendo limiares operacionais para identificação de comportamentos atípicos.

A detecção de anomalias constitui um subcampo importante da *Machine Learning*, com o objetivo de identificar padrões que se desviam do comportamento esperado. Técnicas de aprendizado de máquina, como redes neurais e máquinas de vetores de suporte, podem ser aplicadas aos dados históricos para desenvolver modelos preditivos mais precisos, capazes de prever a vida útil remanescente dos componentes (RUL - *Remaining Useful Life*) [61]. Nesse módulo, a anomalia é detectada comparando o valor atual com a previsão, considerando a variância dos erros preditivos. Isso envolve um aprendizado do comportamento esperado dos dados e a posterior identificação de desvios significativos [51].

A aplicação desse módulo em estratégias de manutenção preditiva representa uma incorporação direta de técnicas de IA e *Machine Learning* utilizadas na Indústria 4.0. Ao proporcionar a previsão de falhas e a detecção de anomalias em tempo real, essa abordagem facilita a transição de uma manutenção reativa – isto é, intervenções apenas após falhas – para uma manutenção proativa, na qual ações corretivas são antecipadas, otimizando custos operacionais e reduzindo o tempo de inatividade dos equipamentos [51]. Esse avanço é viabilizado pelo monitoramento contínuo de grandezas físicas como temperatura, pressão, vibração e ruído, que são fundamentais para a gestão eficiente de processos industriais [53].

Para projetos de sistemas embarcados focados no monitoramento industrial, a escolha e a integração dos sensores são etapas críticas. Sensores de baixo custo e alta precisão são essenciais para viabilizar soluções economicamente competitivas, sem comprometer a qualidade dos dados [56], [57]. A utilização de redes de sensores sem fio (WSNs) ganham destaque nesse cenário, oferecendo flexibilidade na instalação e reduzindo a complexidade da fiação, especialmente em ambientes de difícil acesso ou perigosos [58].

Além disso, a capacidade de processamento embarcado, mesmo em microcontroladores de recursos limitados como o RP2040, permite a filtragem e pré-análise dos dados na "borda" (*edge computing*), reduzindo a quantidade de dados transmitidos e otimizando o uso da largura de banda da rede [35], [59].

Essa abordagem de *edge computing* é crucial para a tomada de decisões em tempo real, diminuindo a latência e a dependência da nuvem para tarefas mais imediatas [60]. Plataformas como o Node-RED, quando integradas com sensores e o protocolo MQTT, permitem a coleta, processamento e visualização contínua de dados. Os dados podem ser exibidos em dashboards customizados, com gráficos de tendência em tempo real, históricos (armazenados em arquivos ou em bancos de dados simples conectados ao Node-RED) e a configuração de alertas [3], [54].

Embora não substituam sistemas SCADA completos em complexidade, soluções baseadas em Node-RED podem oferecer funcionalidades de supervisão e alerta eficazes, servindo como uma ponte acessível para a implementação de sistemas de monitoramento avançados em pequenas e médias indústrias [62].

III. METODOLOGIA

Neste trabalho é proposto um sistema de monitoramento industrial de baixo custo. A arquitetura do sistema consiste no kit BitDogLab, que incorpora a Raspberry Pi Pico W como unidade central de aquisição de dados provenientes de sensores de temperatura, vibração e ruído. Utilizando linguagem de programação em C, a Raspberry Pi Pico W coleta os dados dos sensores e os publica em tópicos MQTT específicos por meio de sua conectividade Wi-Fi. O *broker* Mosquitto [3] é configurado em um servidor local. Na Figura 1, observa-se a um esquema da arquitetura do sistema de monitoramento industrial proposto.



Figura 1. Arquitetura do Sistema de Monitoramento Industrial com MQTT e Node-Red. Fonte: elaborado pelo autor.

A. Sensor de Ruído

Para a coleta dos dados de ruído, utilizou-se o microfone de eletreto acoplado a placa BitDogLab. O microfone capta as ondas sonoras e gera um sinal elétrico proporcional, que é convertido em sinal digital pelo conversor analógico-digital (ADC) da placa.

Para o processamento desses dados, foi necessário transformar o sinal digital do domínio do tempo para o domínio da frequência, por meio da Transformada Rápida de Fourier (FFT), para identificar o nível de ruído presente em diferentes frequências. A aplicação da FFT foi necessária para aplicar a correção de frequência baseada na Ponderação A, exigida pela norma IEC 61672. Dessa forma, é possível filtrar as frequências que o ouvido humano consegue perceber (20 a 20kHz) e ignorar frequências fora dessa faixa.

Os dados de ruído foram coletados a uma taxa de amostragem de 40 kHz, essa taxa foi baseada no Teorema de Nyquist, que afirma que a frequência de amostragem deve ser o dobro da maior frequência que se deseja analisar (20kHz). A calibração do microfone foi realizada por meio do aplicativo NIOSH, com uma precisão de ± 1 dB(A).

O Node-RED, executando no mesmo servidor do *broker* Mosquitto ou em outro dispositivo da rede, é responsável por subscrever aos tópicos MQTT para receber os dados dos sensores. Os fluxos desenvolvidos no Node-RED tem como objetivo o processamento dos dados, bem como a construção de dashboards interativos e mecanismos de previsão. Esses dashboards exibem os dados em tempo real, apresentam gráficos de tendência, para acompanhamento e permitem a configuração de alertas visuais ou notificações por e-mail.

Por meio do módulo *node-red-contrib-industrial-machine-forecast*, o nó *forecast* foi configurado com parâmetros definidos, tal qual o fator de suavização *alpha*, (que determina o peso atribuído à observação mais recente na previsão do próximo valor da série temporal). Também foi ajustado o limiar de detecção de anomalias por meio do parâmetro de número de desvios padrão (*standard deviations*), o qual define a distância que uma observação deve estar em relação a previsão para ser considerada anômala. Além disso, ativou-se o modo adaptativo, permitindo que o valor de *alpha* se ajuste automaticamente com base na volatilidade dos dados de entrada, tornando o modelo mais sensível a mudanças abruptas e menos reativo quando os dados permanecessem estáveis. O nó recebe um valor numérico por meio de *msg.payload* e, como saída, gera um objeto que inclui a observação atual (*value*) e um campo de status, o qual indica se o valor está dentro da faixa normal ou se constitui uma anomalia, podendo ser classificada como um pico ou uma queda repentina.

O Node-RED pode ser configurado para armazenar dados históricos em arquivos locais ou em bancos de dados simples, conferindo uma facilidade à implementação de estratégias de manutenção preditiva [3]. A validação do sistema é obtida através da simulação de condições operacionais típicas de equipamentos industriais, com o monitoramento em tempo real das variáveis e registro nos dashboards do Node-RED, bem como da funcionalidade dos alertas previamente configurados.

IV. RESULTADOS E DISCUSSÃO

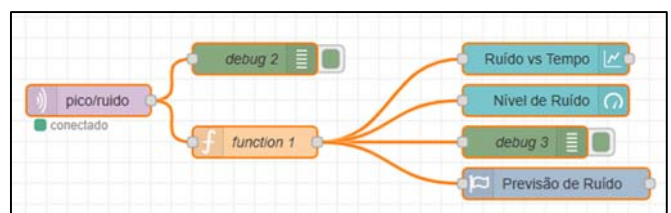
Inicialmente, o fluxo de dados foi realizado utilizando o sensor de ruído acoplado à Raspberry Pi Pico W. A visualização e processamento no Node-RED, operou de forma coesa e em tempo real através do broker MQTT Mosquitto.

O primeiro estágio de validação consistiu na criação de um fluxo de monitoramento de ruído no Node-RED, conforme ilustrado na Figura 2. Nesse fluxo, o nó de entrada MQTT (*pico_w/sensor/ruído_db*) recebe os dados publicados pelo microcontrolador e os direciona para componentes de visualização do painel de controle (*dashboard*).

O resultado dessa implementação é um dashboard funcional, apresentado na Figura 3, que fornece ao operador duas visualizações cruciais da variável monitorada: (a) um gráfico de tendência temporal, que permite a análise do comportamento do ruído ao longo do tempo, e (b) um medidor (“*gauge*”) para a visualização instantânea do nível de ruído em decibéis(dB). A análise do gráfico temporal revela a capacidade do sistema em capturar dinâmicas de processo, como a transição de um estado de baixo ruído para um regime operacional com maior variabilidade em torno das 15:50h, informação esta fundamental para o diagnóstico de condições operacionais.

Avançando para a implementação da capacidade preditiva, o fluxo foi estendido a um nó de função (“coletar”), o qual foi inserido para agregar as leituras sequenciais em um array de dados históricos. Esse array é então fornecido ao nó *forecast*, que encapsula o modelo de análise preditiva. Esta arquitetura valida a premissa central do trabalho: a viabilidade de construir um sistema de monitoramento com capacidades de manutenção preditiva em uma plataforma de baixo custo. A abstração da complexidade estatística pelo nó *forecast* permite que a inteligência preditiva seja integrada no *edge* com esforço de desenvolvimento mínimo. O sistema demonstrou ser não apenas uma ferramenta de visualização passiva, mas uma plataforma ativa para a detecção precoce de anomalias, alinhando-se diretamente aos objetivos da Indústria 4.0 de criar sistemas ciberfísicos autônomos e acessíveis, especialmente para PMEs.

Figura 2. Fluxo de monitoramento básico no Node-RED. Fonte: elaborado pelo autor.



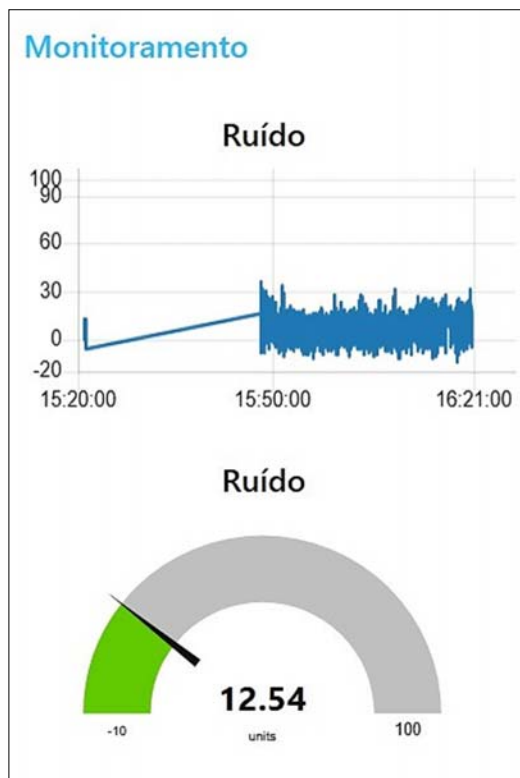


Figura 3. Dashboard de monitoramento de ruído gerado no Node-RED. Fonte: elaborado pelo autor.

Para contextualizar a performance observada, é relevante notar que a arquitetura proposta se baseia em tecnologias com características de desempenho bem documentadas. O protocolo MQTT, quando operando em uma rede Wi-Fi local estável, tipicamente exibe uma alta taxa de confiabilidade na entrega de mensagens, frequentemente superior a 99,9%, com uma latência (tempo entre publicação e recebimento pelo broker) que geralmente se situa entre 20 a 100 milissegundos. A capacidade de processamento do RP2040, com seu processador dual-core de 133 MHz, é suficiente para coletar dados de sensores a taxas de amostragem de centenas de amostras por segundo e realizar o encapsulamento e publicação via MQTT sem se tornar um gargalo para as variáveis monitoradas (temperatura, ruído, vibração) [63]. No que tange à análise preditiva, o modelo de suavização exponencial utilizado pelo nó `node-red-contrib-industrial-machine-forecast` é computacionalmente leve.

A eficácia de tais modelos em detectar anomalias depende criticamente da natureza do sinal e da parametrização. Estudos de caso em monitoramento de condição frequentemente reportam que, para sinais com comportamento relativamente estável, a detecção de desvios significativos (acima de 3 desvios-padrão) pode atingir taxas de detecção superiores a 90%, embora um ajuste fino seja necessário para balancear a sensibilidade e a ocorrência de falsos positivos.

A escolha do kit BitDogLab com Raspberry Pi Pico W, em detrimento de outras plataformas de baixo custo como o ESP32, representa uma decisão estratégica. Enquanto o ESP32 oferece conectividade Wi-Fi e Bluetooth nativa a um custo similar, o RP2040 se destaca por sua arquitetura dual-core e, principalmente, pelo seu subsistema de I/O Programável (PIO), que permite a criação de interfaces de hardware customizadas sem sobrecarregar a CPU. A Tabela expande a comparação

com outras tecnologias relevantes.

Tabela - Análise Comparativa Aprofundada de Soluções de Monitoramento

Característica	Sistema Proposto (BitDogLab/Pico W)	Solução baseada em ESP32	Solução baseada em LoRaWAN
Microcontrolador	RP2040 Dual-Core M0+	Tensilica Xtensa LX6/LX7	Variado (usualmente com MCU de baixo consumo)
Custo de Hardware	Muito Baixo	Muito Baixo	Baixo a Médio
Latência Típica	Baixa (20-100 ms via Wi-Fi local)	Baixa (similar ao Pico W via Wi-Fi)	Alta (500 ms a vários segundos)
Alcance de comunicação	~20-50m (interno, com obstáculos) Até 100m (externo, linha de visada)	~20-50m (interno, com obstáculos) Até 100m (externo, linha de visada)	1-5 km (urbano, com obstáculos) Até 15 km (rural, linha de visada)
Consumo de energia	Ativo (Wi-Fi): ~90-150 mA Repouso Leve: ~40 mA Dormant Mode: ~1.2 mA	Ativo (Wi-Fi): ~120-240 mA Deep Sleep: ~10-150 µA (microampères)	Transmissão (Tx): ~40-120 mA (curta duração) Deep Sleep: ~1-10 µA (microampères)
Complexidade	Baixa (Node-RED abstrai complexidade)	Média (ecossistema Arduino/ESP-IDF)	Média a Alta (requer gateway e servidor de rede)

Figura 4. Análise comparativa aprofundada de soluções de monitoramento. Fonte: elaborado pelo autor.

A tabela evidencia que, enquanto soluções com LoRaWAN são superiores para monitoramento de áreas extensas e sem infraestrutura de energia, a abordagem deste trabalho é otimizada para cenários de baixo custo, baixa latência e prototipagem rápida dentro de uma planta com cobertura Wi-Fi existente.

CONCLUSÃO

O sistema de monitoramento industrial de baixo custo proposto neste trabalho foi implementado, demonstrando a viabilidade e a eficácia da integração entre o kit BitDogLab com Raspberry Pi Pico W, o protocolo MQTT e a plataforma Node-RED. A solução desenvolvida valida a hipótese de que é possível democratizar o acesso a ferramentas da Indústria 4.0, oferecendo a pequenas e médias empresas uma alternativa acessível para a implementação de estratégias de manutenção preditiva.

Os resultados confirmaram que a arquitetura proposta permite não apenas o monitoramento em tempo real de variáveis críticas, como o ruído, mas também a incorporação de inteligência preditiva diretamente no edge. A utilização do módulo de previsão no Node-RED abstrai a complexidade dos modelos estatísticos, permitindo a criação de um sistema capaz de antecipar anomalias com mínima sobrecarga de programação. A interface de supervisão provou ser intuitiva e informativa, fornecendo dados essenciais para o diagnóstico de condições operacionais e a tomada de decisão.

Conclui-se, portanto, que este trabalho apresenta uma contribuição significativa ao oferecer um caminho prático para a modernização industrial. A solução não substitui a complexidade de sistemas SCADA comerciais, mas atua como uma ponte tecnológica crucial, promovendo a eficiência operacional e a confiabilidade dos processos. Contudo, para direcionar o desenvolvimento futuro, é fundamental reconhecer as limitações técnicas da abordagem. O microcontrolador RP2040, por não possuir uma Unidade de Ponto Flutuante (FPU) por hardware, apresenta desempenho reduzido em operações matemáticas complexas. Os sensores MEMS utilizados, embora eficazes para prototipagem, são suscetíveis a ruído e possuem precisão inferior à dos sensores industriais. Adicionalmente, a comunicação via Wi-Fi pode ser vulnerável à interferência eletromagnética comum em

ambientes de fábrica. Nesse contexto, a flexibilidade e a escalabilidade do sistema abrem portas para futuras expansões, incluindo a integração de sensores de padrão industrial (ex: acelerômetros IEPE) é um passo crucial para aumentar a fidelidade dos dados, a aplicação de algoritmos de aprendizado de máquina mais avançados via TinyML para análise *on-device*, e a validação do sistema em um ambiente de produção real, consolidando ainda mais seu valor no cenário da manutenção preditiva.

AGRADECIMENTOS

Os autores agradecem à Softex, através da Residência Embarcatech, e ao IFMA pelo apoio financeiro.

REFERÊNCIAS

- [1] I. A. Cabral, “Protótipo de computação vestível para monitoramento de locomoção de idoso utilizando node-red com esp32 e raspberry pi,” Monografia, Univ. Federal do Amazonas, Manaus, 2022.
- [2] S. A. Boyer, *SCADA: Supervisory Control and Data Acquisition*, 4th ed., ISA, 2010.
- [3] R. K. Mobley, *An Introduction to Predictive Maintenance*, 2nd ed., Burlington: Elsevier, 2002.
- [4] Centro de Gestão e Estudos Estratégicos, “Indústria 4.0: Segmentos ou nichos com maior potencial para o desenvolvimento tecnológico nacional,” CGEE, Tech. Rep., Brasília, 2022.
- [5] Associação Brasileira de Normas Técnicas (ABNT), *NBR 5462: Confiabilidade e Manutenibilidade*, Rio de Janeiro, 1994.
- [6] V. Lampkin et al., *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry*, IBM Redbooks, 2012.
- [7] C. V. Corrêa and G. M. de Almeida, “Monitoramento de um evaporador industrial via redes neurais artificiais,” in *ERMAC 2010*, MG, Nov. 2010, pp. 277–278.
- [8] M. R. Gouvêa et al., “Sistema de monitoramento e diagnóstico de falhas em plantas industriais,” in *51º Seminário de Laminação*, PR, 2014, pp. 332–342.
- [9] M. Fabricio, F. Behrens, and D. Bianchini, “Monitoring of industrial electrical equipment using IoT,” *IEEE Lat. Am. Trans.*, vol. 18, no. 8, pp. 1425–1432, Aug. 2020.
- [10] J. F. M. Oliveira et al., “Sistema de monitoramento de máquina industrial para manutenção preditiva utilizando IoT,” in *XVII SEGET*, 2017.
- [11] C. L. Azevedo, “Implantação de sistema de monitoramento de disponibilidade em sistema de água quente industrial,” UTFPR, Curitiba, PR, Monografia, 2014.
- [12] A. R. Reis, “Desenvolvimento de um sistema de monitoramento de eficiência energética para refrigeração industrial,” TCC, UFAM, Manaus, AM, 2023.
- [13] D. Farias, “Sistema de monitoramento de parâmetros de processos industriais utilizando a tecnologia LoRa®,” TCC, UNISINOS, RS, 2018.
- [14] OpenJS Foundation, “Node-RED: About,” 2020. [Online]. Available: <https://nodered.org/about/>
- [15] U. Hunkeler et al., “MQTT-S—a publish/subscribe protocol for wireless sensor networks,” in *COMSWARE*, 2008, pp. 791–798.
- [16] BitDogLab, “BitDogLab hardware,” GitHub, 2023. [Online]. Available: <https://github.com/BitDogLab/BitDogLab>
- [17] T. Sousa, “Datasheet BitDogLab,” GitHub, 2023. [Online]. Available: <https://github.com/ThiagoSousa81/Datasheet-BitDogLab>
- [18] R. Giusti, “BitDogLab: Uma Jornada Educativa com Eletrônica, Embarcados e IA,” *Embarcados*, 2023. [Online]. Available: <https://embarcados.com.br/bitdoglab-uma-jornada-educativa-com-eletronica-embarcados-e-ia/>
- [19] W. Wolf, *Computers as Components*, 2nd ed., Morgan Kaufmann, 2008.
- [20] E. A. Lee and S. A. Seshia, *Embedded Systems Design*, Prentice Hall, 2006.
- [21] P. Marwedel, *Embedded System Design*, 2nd ed., Springer, 2011.
- [22] M. Barr and A. Massa, *Programming Embedded Systems*, 2nd ed., O’Reilly, 2006.
- [23] J. Ganssle, *The Art of Designing Embedded Systems*, 2nd ed., Newnes, 2008.
- [24] D. Hanes et al., *IoT Fundamentals*, Cisco Press, 2017.
- [25] P. Koopman and S. Chakravarty, “A case for safety and security of automotive embedded systems,” *Computer*, vol. 47, no. 11, pp. 31–37, 2014.
- [26] P. Ostin and D. Hedberg, “Security challenges in embedded systems,” *IJECE*, vol. 9, no. 5, pp. 3861–3869, 2019.
- [27] J. G. Albert, *The Firmware Handbook*, Elsevier, 2010.
- [28] T. Chen et al., “Deep learning for embedded systems: a survey,” *IEEE Access*, vol. 7, pp. 70656–70671, 2019.
- [29] Xilinx, “AI on Embedded Systems: Accelerating Vision-Based AI at the Edge,” Xilinx White Paper WP510, 2020. [Online]. Available: https://www.xilinx.com/content/dam/xilinx/support/documentation/white_papers/wp510-ai-embedded-systems.pdf. Acessado em: 5 de junho de 2025.
- [30] T. Noergaard, *Embedded Systems Architecture*, Elsevier, 2013.
- [31] S. Monk, *Programming the Raspberry Pi: Getting Started with Python*, McGraw-Hill, 2016.
- [32] A. Gilchrist, *Industry 4.0: The Industrial Internet of Things*, Apress, 2016.
- [33] H. Boyes et al., “The internet of things: challenges and opportunities,” *Computer Fraud & Security*, vol. 2018, no. 2, pp. 11–17, 2018.
- [34] B. C. Sahoo and S. S. Dash, “A survey on low-cost IoT devices and their applications,” *Int. J. Adv. Res. Comput. Eng. Technol.* (IJARCET), vol. 6, no. 1, pp. 50–255, 2017.
- [35] J. Chen and D. Wang, “Review of low-cost embedded systems for IoT applications,” *J. Commun. Inf. Netw.*, vol. 5, no. 1, pp. 55–67, 2020.
- [36] Softex, “IFRN entrega placas eletrônicas BitDogLab durante o EmbarcaTech Hands-On,” 2023. [Online]. Available: <https://softex.br/>
- [37] G. Thangavelu et al., “A comparison of IoT communication protocols: MQTT, CoAP, and HTTP,” *J. Comput. Sci. Eng.*, vol. 10, no. 1, pp. 1–6, 2019.
- [38] D. Kim et al., “MQTT-based sensor data publishing and subscription for IoT applications,” *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 10, 2017.
- [39] Q. Feng, C. Liu, and L. Liu, “Research on MQTT protocol and its application in smart home system,” *J. Phys.: Conf. Ser.*, vol. 1087, no. 4, 2018.
- [40] C. Douligieris and T. Douligieris, “A survey of MQTT for

IoT applications,” *J. Commun. Netw.*, vol. 20, no. 6, pp. 555–569, 2018.

[41] Eclipse Foundation, “Mosquitto MQTT Broker.”

[Online]. Available: <https://mosquitto.org/>

[42] S. Biswas et al., “Performance analysis of MQTT protocol for IoT applications,” in *Proc. IEEE ECCE*, 2019, pp. 1–6.

[43] N. Yoshida et al., “Developing IoT applications with Node-RED: A visual programming approach,” in *Proc. IEEE FiCloud*, 2017, pp. 133–138.

[44] D. Clerissi et al., “Towards an approach for developing and testing Node-RED IoT systems,” in *Proc. ACM Workshop on Ensemble-Based Software Eng.*, 2018.

[45] K. Ferencz and J. Domokos, “Using Node-RED platform in an industrial environment,” in *Jubileumi Kandó Konferencia*, 2019, pp. 1–5.

[46] P. Choubey et al., “IoT based smart home automation using Node-RED,” *Int. J. Eng. Res. Technol. (IJERT)*, vol. 9, no. 05, 2020.

[47] V. Ramesh et al., “IoT dashboard and control system using Node-RED,” in *Proc. IEEE ICCCI*, 2020, pp. 1–6.

[48] A. Patel et al., “Node-RED: A visual programming tool for IoT data processing,” *Int. J. Eng. Res. Technol. (IJERT)*, vol. 8, no. 05, 2019.

[49] R. Sharma et al., “Low-cost IoT-based smart agriculture system using Node-RED and Raspberry Pi,” *J. Ambient Intell. Humaniz. Comput.*, pp. 1–13, 2021.

[50] H. Glaess et al., “Rapid prototyping of a digital twin for an industrial application using Node-RED,” in *Proc. IEEE INDIN*, 2021, pp. 1–6.

[51] V. Chandola et al., “Anomaly Detection: A Survey,” *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[52] L. C. M. S. Portela, “Desenvolvimento de plataforma para automação residencial com dispositivos inteligentes utilizando protocolo MQTT e Node-RED,” *TCC (Engenharia Elétrica)*, 2022.

[53] M. B. Prieto et al., “Data-driven predictive maintenance applications in industrial SCADA systems,” *IEEE Access*, vol. 8, pp. 186638–186648, 2020.

[54] J. Lee, H.-A. Kao, and S.-C. Yang, “Industrial internet of things: Towards a new era of manufacturing,” *Mach. Learn. Cybern.*, vol. 2, no. 1, pp. 5–7, 2013.

[55] A. Alvarez-Lopez et al., “Low-cost wireless sensor network for industrial monitoring applications,” *Sensors*, vol. 19, no. 17, p. 3723, 2019.

[56] H. Xiao et al., “A comprehensive review of low-cost MEMS sensors for structural health monitoring,” *Sensors*, vol. 21, no. 10, p. 3551, 2021.

[60] S. Mahapatra et al., “Edge computing for industrial IoT: A survey,” *J. Ind. Inf. Integr.*, vol. 23, p. 100236, 2021.

[61] M. Ge et al., “A review on prognostics and health management of rolling element bearings,” *J. Intell. Manuf.*, vol. 28, pp. 1115–1132, 2017.

[62] S. K. Giri et al., “Low-cost IoT-based smart energy monitoring and management system for industrial applications,” *Mater. Today: Proc.*, vol. 28, pp. 301–306, 2020.

[63] Bosch Sensortec. (2018, January 8). BMP280: Digital Pressure Sensor - Data sheet (BST-BMP280-DS001-19, Rev. 1.19).