UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

CAIO PESSOA MASCARIN
N° USP: 10350091

DESENVOLVIMENTO DE SENSOR PARA ALERTA DE CRITICIDADE DE MÁQUINAS INDUSTRIAIS EM TEMPO REAL

Orientador: Daniel Capaldo Amaral

São Carlos
2022

CAIO PESSOA MASCARIN

DESENVOLVIMENTO DE SENSOR PARA ALERTA DE CRITICIDADE DE MÁQUINAS INDUSTRIAIS EM TEMPO REAL

Monografia apresentada ao Curso de Engenharia de Produção, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro de Produção.

Orientador: Daniel Capaldo Amaral

São Carlos
2022

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

MM395d

Mascarin, Caio Pessoa
Desenvolvimento de sensor para alerta de criticidade de máquinas industriais em tempo real / Caio Pessoa Mascarin; orientador Daniel Capaldo Amaral. São Carlos, 2022.

Monografia (Graduação em Engenharia de Produção) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2022.

1. sensor. 2. vibracao. 3. temperatura. 4. hardware. 5. industria. 6. alerta. 7. preditiva. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE APROVAÇÃO

Candidato: Caio Pessoa Mascarin
Título do TCC: Desenvolvimento de sensor para alerta de criticidade de máquinas industriais em tempo real
Data de defesa: 05/12/2022

Comissão Julgadora	**Resultado**
Professor Associado Daniel Capaldo Amaral (orientador)	Aprovado
Instituição: EESC - SEP	
Professor Associado Eraldo Jannone da Silva	Aprovado
Instituição: EESC - SEP	
Pesquisadora Silvia Ronsom	Aprovado
Instituição: EESC - SEP	

Presidente da Banca: **Professor Associado Daniel Capaldo Amaral**

DEDICATÓRIA

Dedico este trabalho primeiramente ao meu avô, que despertou em mim a vontade de cursar Engenharia. Dedico também aos meus pais e minha namorada, que me apoiaram durante toda essa trajetória.

AGRADECIMENTOS

Agradeço, primeiramente, à minha família, que investiu tempo e dinheiro para que eu pudesse chegar até aqui, sempre dispostos a me apoiar.

À minha namorada, que além do incentivo durante toda a graduação, me auxiliou nas dúvidas durante a escrita deste trabalho.

Ao professor Doutor Daniel Capaldo Amaral, pois além da importância para o curso de Engenharia de Produção, acompanhou este trabalho sendo meu orientador.

Aos meus professores, tanto do ensino fundamental e médio, quanto da graduação, que forneceram a base necessária para que eu pudesse construir este trabalho.

Aos meus companheiros de equipe da extracurricular EESC USP BAJA, que confiaram em mim e possibilitaram que eu pudesse aplicar na prática muito dos conhecimentos adquiridos durante a graduação.

Aos meus colegas de trabalho, que me apoiaram e permitiram boa parte dos testes práticos deste protótipo.

EPÍGRAFE

“A tecnologia move o mundo.”

Steve Jobs (2007)

RESUMO

PESSOA MASCARIN, Caio, **Desenvolvimento De Sensor Para Alerta de Criticidade De Máquinas Industriais Em Tempo Real**, 2022. [--] f. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2022.

Para manter altos índices de disponibilidade dos ativos industriais, assim como definido pela norma NBR 5462 (1994), as indústrias necessitam reduzir o número de falhas e de paradas não planejadas. Para que o setor de manutenção atue de forma eficaz, é importante contar com formas de aumentar a confiabilidade dos dados de operação das máquinas, tanto em estado normal, quanto em estado de criticidade. Este trabalho tem o objetivo de desenvolver um sensor de vibração e temperatura para ativos industriais, pois a partir dessas duas grandezas, é possível definir alertas de funcionamento fora do padrão esperado. O principal foco está na escolha do *hardware* fundamental, visando soluções para alta viabilidade de implementação do projeto, tanto pela facilidade de instalação quanto pela disponibilidade dos componentes principais no mercado brasileiro. Para permitir uma fácil instalação, o sensor é alimentado por bateria, o que evita a necessidade de instalação de tomadas convencionais próximo das máquinas. Além disso, conta com um receptor com capacidade de se conectar com rede celular, ou seja, dispensa a necessidade de rede Wi-Fi dentro da indústria. Portanto, o sensor busca apresentar um funcionamento independente da infraestrutura pré-instalada na planta industrial que será utilizado.

Palavras-chave: Sensor, Vibração, Temperatura, Hardware, Indústria, Alerta, Preditiva.

ABSTRACT

PESSOA MASCARIN, Caio, **Development Of A Sensor For Real-Time Industrial Machine Criticality Alert**, 2022. [--] Monography (Final Paper) – São Carlos School of Engineering, University of São Paulo, São Carlos, 2022.

To maintain high levels of availability of industrial assets, as defined by NBR 5462 (1994), industries need to reduce the number of failures and unplanned stops. For the maintenance sector to act effectively, it is important to have ways to increase the reliability of machine operation data, both in normal and critical states. This work aims to develop a vibration and temperature sensor for industrial assets, because from these two inputs, it is possible to define alerts of operation outside the expected standard. The main focus is on the choice of the fundamental hardware, aiming at solutions for high project implementation viability, both for ease of installation and for the availability of the main components in the Brazilian market. To allow easy installation, the sensor is battery powered, which avoids the need for conventional outlets near the machines. In addition, it has a receiver which is capable of connecting to a cellular network, i.e., it eliminates the need for a Wi-Fi network inside the industry. Therefore, the sensor seeks to work independently of the infrastructure pre-installed in the industrial plant where it will be used.

Keywords: Sensor, Vibration, Temperature, Hardware, Industry, Warning, Predictive.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT - Associação Brasileira de Normas Técnicas

NBR - Norma Brasileira

RMS - Root Mean Square

DSR - Design Science Research

SPI - Serial Peripheral Interface

UART - Universal Asynchronous Receiver-Transmitter

MAC - Media Access Control

SUMÁRIO

1 INTRODUÇÃO

O rápido crescimento tecnológico constatado nos últimos anos, englobando áreas de eletrônica e telecomunicações, ganha espaço em todos os setores da sociedade. Dentro do setor industrial, essas novas tecnologias impactam não só os meios de produção, mas também as formas de planejar e realizar a manutenção nas indústrias. As soluções de automação são um dos caminhos promissores e parte importante da transformação digital que está acontecendo nas empresas.

Diversos trabalhos buscam formas de implementar esses avanços diretamente ao setor da manutenção. Segundo o projeto de González (2014), é possível ver um detalhamento completo da implementação de um sensor de vibração, explicando conceitos importantes na área de manutenção e predição de falhas.

Além do cenário da manutenção preditiva, os sensores podem ser eficazes também para o cálculo de disponibilidade das máquinas industriais, por meio da aquisição de vibração e temperatura. Esse indicador, que é definido pela NBR 5462 (1984), está relacionado ao tempo de operação de cada máquina. Com base nele, podemos saber quais são os ativos industriais mais críticos para o processo, levando em conta os tempos de parada medidos pelo sensor. Dessa forma, o impacto do monitoramento passa a valer não só para a manutenção preditiva, mas também para um melhor planejamento de manutenções preventivas e corretivas.

A aplicação de um sensor no meio industrial pode ser utilizada para definir alertas de mau funcionamento das máquinas. Isso pode ser feito por meio da configuração de limites de operação ideal dos ativos monitorados, tanto no aspecto da temperatura, quanto da vibração. No entanto, uma das dificuldades enfrentadas pela indústria na utilização desses sensores é o custo agregado à implementação, tanto em relação ao *hardware*, como em relação à plataforma de análise de dados e geração de alertas.

Como proposta de aplicação para o cenário acima, este trabalho foca no desenvolvimento de um sensor com capacidade de medir vibração e temperatura, alimentado por bateria, que envia as coletas para a nuvem em tempo real por meio de um receptor com conexão à rede celular. Esta combinação facilita a implementação dentro das indústrias, visto que não é necessário instalar tomadas nas regiões das máquinas, além de evitar a necessidade de rede Wi-Fi para envio dos dados coletados. Todo esse conjunto leva em consideração o problema de

aplicação em função dos altos custos, de forma que, esse trabalho visa alcançar uma solução também escalável do ponto de vista de custo.

Dessa forma, com base nos dados monitorados de vibração e temperatura, é possível acompanhar em tempo real quando o ativo não está operando dentro dos padrões de funcionamento pré estabelecidos. No caso da detecção de uma anormalidade, o usuário receberá um email para que possa tomar uma iniciativa de manutenção antes mesmo da quebra e parada por completo do equipamento monitorado.

Levando em consideração o foco no baixo custo, esse sistema pode ser utilizado principalmente por pequenas e médias empresas que tem a finalidade de implementar o uso da tecnologia no ramo da manutenção preditiva, com o objetivo principal de melhorar o planejamento da produção e da manutenção e, por consequência, aumentar a produtividade e a confiabilidade da indústria.

1.1 OBJETIVO GERAL

Após identificar lacunas, proposições e hipóteses por meio da revisão bibliográfica sistemática, desenvolver um sensor capaz de medir vibração e temperatura de ativos industriais e enviar esses dados para um servidor na internet, sem que haja necessidade de conexão com rede Wi-Fi ou tomada para alimentação do sensor na região do ativo monitorado. Com base nesses dados, notificar o usuário final quando um equipamento monitorado estiver operando fora do padrão de funcionamento pré estabelecido. Esse conjunto deve levar em consideração a facilidade de implementação do projeto, tanto do ponto de vista de custo, quanto do ponto de vista de instalação.

1.2 ESTRUTURA DO TRABALHO

Este relatório está dividido na seguinte estrutura:

- Capítulo 1: contém a introdução e contextualização do problema estudado, os objetivos e a estrutura do trabalho;
- Capítulo 2: apresenta o referencial teórico dos conceitos utilizados ao longo do trabalho;
- Capítulo 3: apresenta a metodologia aplicada no desenvolvimento do projeto;
- Capítulo 4: apresenta a aplicação prática e os resultados obtidos;
- Capítulo 5: contém as conclusões do trabalho;

- Referências: aponta as referências utilizadas nas citações;
- Apêndices: contém conteúdo adicional de elaboração própria.

2 REVISÃO BIBLIOGRÁFICA

2.1 CONCEITOS DE MANUTENÇÃO

O termo “manutenção” se originou na nomenclatura militar, e, neste contexto, carregava o significado de “manter os efetivos e provisões constantes”, entretanto, até então, sua aplicação era limitada às unidades de combate, e só se tornou parte da terminologia industrial a partir de meados de 1950, nos Estados Unidos, após a Segunda Guerra Mundial (MONCHY, 1989).

Considerada, até pouco tempo atrás, como apenas um fator de custos e gastos, o conceito de manutenção, hoje, tem sofrido mudanças, graças à sua grande relevância quando o assunto é a parada e a quebra das máquinas durante o período produtivo, causadas por erros de gerenciamento e técnica, também conhecidas como “Down Time” (WYREBSKI, 1997). Por isso, atualmente, “manutenção” se refere à combinação de todas as ações, técnicas e administrativas, destinadas a manter ou recolocar um item em um estado no qual possa desempenhar a função requerida (ABNT/ NBR 5462, 1994).

2.1.1 Falha

"Falha" é o termo utilizado para descrever um evento que impossibilita que algum componente de um equipamento desempenhe, adequadamente, as suas funções, diferentemente do termo "desvio", que representa alguma alteração nas características de determinado item, sem, necessariamente, comprometer a atividade do equipamento. (ABNT/ NBR 5462, 1994).

Diante disso, com a evolução da falha, tem-se a quebra, que pode ser definida como um estado, e normalmente, representa o estopim de um processo gradativo de perda de funcionalidade, até que o equipamento já não tenha a capacidade de realizar suas atividades, como representada no gráfico abaixo. A vantagem de um sensor que atua na manutenção preditiva é que ele poderia ser capaz de identificar o problema entre a falha potencial e a falha funcional, ou seja, antes do equipamento apresentar a quebra de fato (Figura 1). (ABNT/ NBR 5462, 1994; CYRINO, 2017)

Figura 1 - Evolução da Falha

Fonte: Cyrino (2017)

2.1.2 Tipos de manutenção

Assim como já abordado na revisão sistemática e definido pela norma NBR 5462 (1994), existem alguns tipos de manutenção, sendo as principais: Corretiva, Preventiva e Preditiva. A norma traz a definição detalhada de cada uma delas.

A manutenção corretiva é aquela que vem em sequência de uma falha, ou seja, ela vem para resolver um problema que já ocorreu. Por se tratar de uma ação reativa, normalmente ela está associada a um menor nível de planejamento, o que pode acarretar um custo e um tempo de parada de produção maior.

A manutenção preventiva é aquela que acontece antes de uma falha evidente acontecer e está associada a uma rotina de execução, seja por intervalos de tempo preestabelecidos pelo fabricante de um determinado componente ou pelo próprio programa de manutenção da indústria.

Por fim, tem-se a manutenção preditiva, que vem ganhando cada vez mais espaço graças a implementação de tecnologia no monitoramento dos ativos. Com esta técnica, a manutenção ocorre antes da falha ocorrer de fato, pois algum mecanismo apontou uma não conformidade inicial dentro do sistema de produção.

2.2 MEDIÇÃO DE ACELERAÇÃO

Os acelerômetros podem ser definidos como dispositivos de medição inercial, que exigem fontes externas de alimentação e trabalham dentro de um espectro de frequências que variam de modelo para modelo. Sua função principal consiste em converter movimentos mecânicos em sinais de tensão (SCHEFFER; GIRDHAR, 2004).

No setor industrial, é extremamente útil para mensurar a aceleração do movimento de vibração, que condicionam sinais através de integradores e diferenciadores, possibilitando o uso de acelerômetros para a avaliação da aceleração, velocidade e deslocamento, que se configuram como os três parâmetros principais de uma vibração (DEFENDI, 2020).

A leitura feita por um acelerômetro, ou seja, os movimentos mecânicos, pode se tratar de uma vibração aleatória, já que seu comportamento futuro não pode ser completamente determinado. Para quantificar essa leitura é utilizado o cálculo do valor RMS (*Root Mean Square*), que representa a energia total contida no evento de vibração medido (CRANDALL, 1958).

Segundo Thompson (1965), o cálculo do valor RMS, que traduzido é raiz do valor quadrático médio, para uma coleção de N elementos pode ser definida como mostra a equação abaixo (Equação X).

$$RMS = \sqrt{\frac{1}{N}\sum_{i} x_i^2} \quad (1)$$

2.3 FOLGA MECÂNICA DE MÁQUINAS

Segundo Galli (2017), as folgas mecânicas indesejadas podem surgir em função de encaixes impróprios e falta de rigidez entre os componentes de uma máquina. Este comportamento pode ser iniciado desde a fabricação do conjunto, quando ocorre um erro dimensional de projeto, como também após um processo de desgaste e deformação de alguns componentes críticos. Essa condição pode ser constatada por meio da análise de vibrações da máquina, visto que ocorre um aumento da aceleração medida.

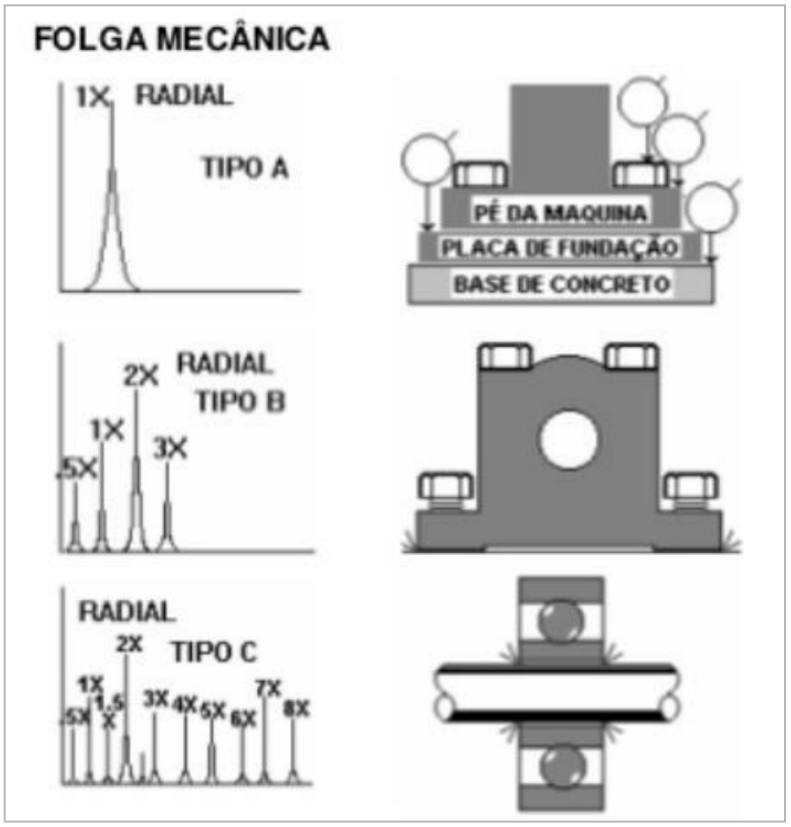
Galli (2017) também define os tipos de folgas mecânicas:

- **Tipo A:** caracterizada por falta de rigidez estrutural, seja nos pés, base ou fundação da máquina, além da deterioração do apoio ao solo, folga de parafusos que sustentam a base e distorções da armação ou base da máquina (pé frouxo por exemplo);
- **Tipo B:** caracterizada por falha de fixação, geralmente causado por parafuso solto, trincas no pé ou em uma das bases (usualmente chamado de pé manco);

- **Tipo C:** causada por ajuste impróprio entre componentes para forças dinâmicas do rotor. Geralmente, ocorre por ajuste impróprio entre o anel externo do rolamento e caixa do mancal ou anel interno e eixo, também por folga excessiva em buchas ou rotor solto com folga em relação ao eixo.

Na figura abaixo (Figura 2), é possível observar os principais espectros envolvidos com cada um dos tipos de folga.

Figura 2 - Espectros dos tipos de folgas mecânicas



Fonte: Galli (2017)

3 METODOLOGIA

O trabalho utilizou como guia a teoria sobre Design Science Research (DSR). Segundo Hevner et al. (2004), o objetivo desta metodologia é desenvolver soluções baseadas em tecnologias para resolver problemas empresariais relevantes. Estas soluções estão associadas ao desenvolvimento de um artefato.

Segundo Peffers et al. (2007), a DSR está segmentada em seis principais etapas:

1. Identificação: identificar problemas, lacunas e motivações envolvidas com o projeto;
2. Definição: com bases nos problemas levantados, definir objetivos e soluções;
3. Desenvolvimento: utilizando os objetivos como guia, desenvolver o artefato;
4. Demonstração: com o artefato desenvolvido, encontrar um contexto de aplicação e utilizar o artefato para resolver o problema identificado inicialmente;
5. Avaliação: após a aplicação do artefato, verificar se o problema de fato foi resolvido;
6. Comunicação: difundir os resultados obtidos por meio de publicações acadêmicas e profissionais.

Levando em consideração o escopo e o tempo de construção do trabalho, esta metodologia foi aplicada com algumas alterações, de forma que as etapas foram simplificadas em:

1. Estudo do problema e definição de objetivos;
2. Desenvolvimento do artefato;
3. Demonstração e avaliação.

Por fim, a etapa de comunicação, como a metodologia propõe, se trata deste próprio trabalho, como forma de divulgação dos métodos aplicados e dos resultados obtidos.

4 RESULTADOS

Como descrito na metodologia e proposto pelo DSR, os resultados obtidos estão em função do levantamento de hipóteses e de lacunas a serem constatadas na etapa final de demonstração, após a definição de soluções e a criação do artefato, que neste caso, se trata de um produto mecatrônico.

4.1 ESTUDO DO PROBLEMA E DEFINIÇÃO DE OBJETIVOS

Optou-se pelo método da Revisão Bibliográfica Sistemática (RBS) como primeira etapa na identificação das lacunas e proposições para o sensor de

criticidade de máquinas. O intuito desta etapa foi o de identificar, nas bases de dados científicas, as propostas já publicadas no meio acadêmico para a medição de vibração e temperatura de máquinas em tempo real. Em seguida, foi realizada a definição dos objetivos e soluções com base nos dados levantados pela RBS.

4.1.1 - Revisão sistemática da literatura

A revisão sistemática utilizou a plataforma de pesquisas Scopus com os assuntos e as palavras-chaves exibidas na tabela abaixo (Tabela 1).

Tabela 1 - Domínios da pesquisa do Scopus

Domínio	Assunto do artigo	Descritores e comando de busca
1	Manutenção	Maintenance AND availability AND (machine OR asset OR industrial OR industry)
2	Vibração e temperatura	Vibration AND temperature
3	Prototipagem	Prototype OR proposal OR sensor OR device
4	Online	Online OR (real AND time) OR iot

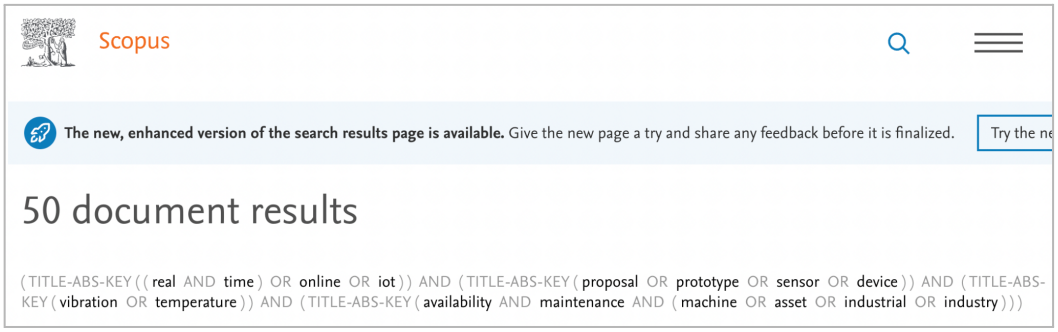
Fonte: Elaboração própria.

O intuito de utilizar cada um destes quatro domínios foi:

1. Limitar a busca no escopo da manutenção, agregando termos como disponibilidade e aplicação em máquinas industriais;
2. Criar um vínculo dos artigos com os conceitos de vibração e temperatura, que são fundamentais para este trabalho;
3. Direcionar a busca para a criação e o desenvolvimento de soluções de *hardware*;
4. Focar em soluções que operam de forma online, ou seja, apresentam dados sendo coletados em tempo real.

Ao realizar a união destes quatro domínios acima através do operador AND, o resultado obtido foi de 50 artigos científicos relacionados, mostrados abaixo (Figura 3).

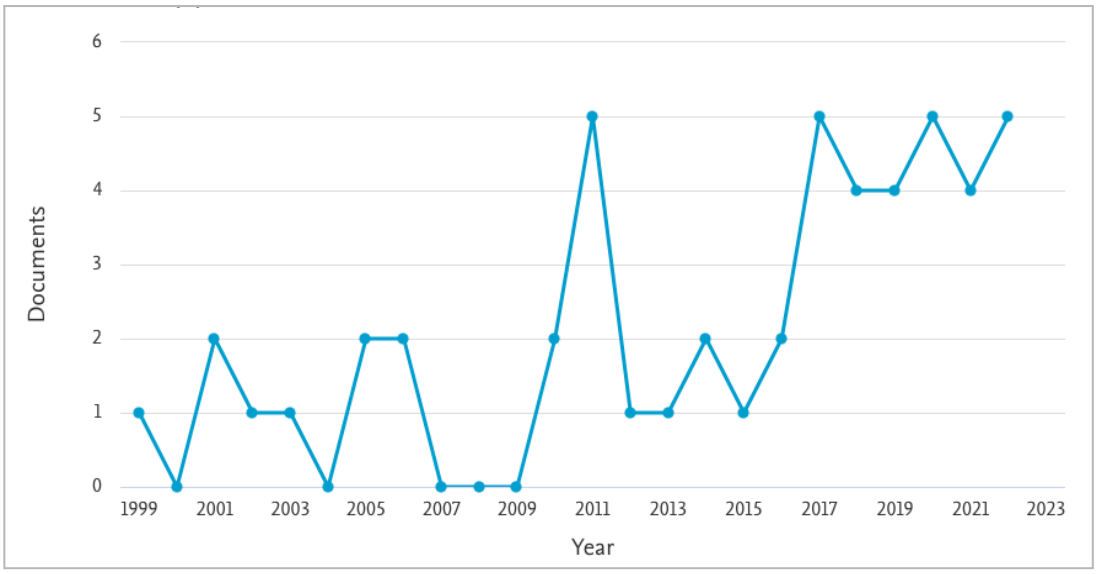
Figura 3 - Resultado da busca no Scopus



Fonte: Elaboração própria.

Utilizando a ferramenta de análise de resultado do Scopus, é possível destacar a evolução deste tema ao decorrer dos anos (Figura 4). Fica claro o crescimento dessas publicações em função da crescente demanda por novas tecnologias.

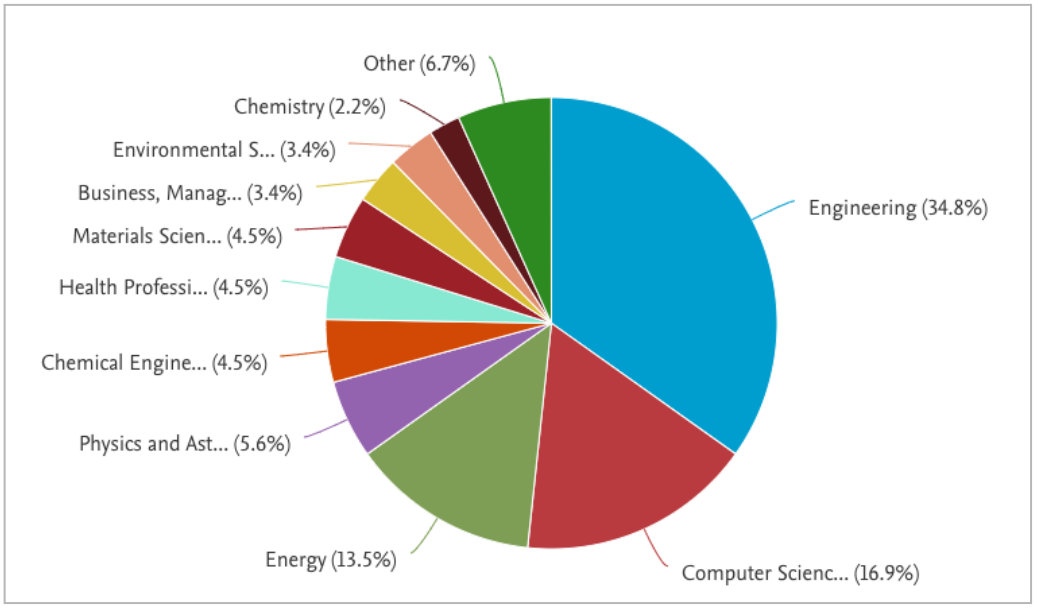
Figura 4 - Evolução do tema ao decorrer dos anos



Fonte: Elaboração própria.

Além da evolução do tema ao decorrer dos anos, foi possível também agrupar em subáreas de atuação. Na figura abaixo (Figura 5) é possível observar as duas principais: engenharia e ciência computacional. Além disso, algumas publicações estão focadas em determinados ramos da indústria, tais como química e energética.

Figura 5 - Subáreas dos resultados obtidos



Fonte: Elaboração própria.

Analisando os artigos obtidos para verificar a relação com o escopo deste trabalho, temos alguns aspectos importantes. Na prática, existem três estratégias mais utilizadas para gerenciar equipamentos industriais: estratégia preventiva, estratégia corretiva e estratégia preditiva. Assim como já definido pela NBR 5462 (1994) no tópico de revisão bibliográfica deste trabalho, em resumo, a preventiva tem o objetivo de estimar o tempo de duração da máquina, fazendo uso de dados estatísticos de falha dos componentes. Entretanto, nesta estratégia, as previsões de vida útil podem ser demasiadamente conservadoras, acarretando um gasto adicional por trás do descarte de componentes que ainda poderiam funcionar corretamente por um período maior do que o estimado (Hashemian, 2006).

Na abordagem corretiva, tem-se o funcionamento do equipamento até a falha, suspendendo, assim, o custo gerado por etapas de manutenção ao longo da sua vida útil, entretanto, quando um componente deixa de funcionar, pode haver prejuízo do mecanismo como um todo, acarretando a necessidade de substituição do equipamento por inteiro, que além de altos custos, pode gerar prejuízos pelo longo período de inatividade e, consequentemente, queda de lucros de uma determinada empresa.

Sendo assim, para minimizar o tempo de não-funcionamento de ativos industriais, tem-se a manutenção preditiva, que utiliza dados coletados com o intuito de avaliar as condições de funcionamento das máquinas, antes que estas

apresentem falhas, minimizando riscos, custos e ampliando a disponibilidade industrial destes equipamentos. Com isso, esta estratégia proporciona uma nova forma de planejar a manutenção, ainda que com limitações por depender da frequência da coleta dos dados, análise e interpretação de informações coletadas, qualidade de sinal, dentre outras variáveis. (Hashemian, 2011).

Dessa forma, os artigos resultantes desta busca no Scopus que foram levados em consideração são aqueles que incluem um propósito de desenvolvimento e melhoria em cima desta última abordagem, ou seja, a manutenção preditiva.

4.1.2 - Definição de objetivos e solução

Analisando os trabalhos publicados envolvendo a implementação prática da manutenção preditiva e de acordo com Siddhartha et al. (2020), no cenário de competitividade atual, é importante que o chão de fábrica seja operado com a maior eficiência possível visando o menor tempo de inatividade, menor índice de falhas, e, consequentemente, menos paradas na produção por problemas identificados nos ativos, possibilitando a alta produtividade e lucratividade das empresas.

Amruthnath e Gupta (2018) desenvolveram um modelo para detecção de falhas, seguindo o modelo da manutenção baseada em condições (manutenção preditiva), mas neste caso, só pode ser implementado em máquinas rotativas, uma vez que utiliza como atributo principal de análise, a vibração do equipamento, e não leva em consideração fatores tais como pressão ou temperatura.

Por outro lado, Uhlmann et al. (2017) apresentam uma abordagem para monitoramento de condições que se baseiam em componentes eletrônicos simplificados, utilizados com o objetivo de criar uma rede de sensores econômica, e escalável, capaz de monitorar sistemas de produção e seus equipamentos, que por sua vez, são susceptíveis a desgastes e falhas. Isso foi possível, no trabalho citado, graças ao processamento descentralizado de dados dos sensores, reunidos na nuvem, e analisados em conjunto.

Siddhartha et al. (2020), já citado anteriormente, apresenta um projeto de sistema, utilizando módulo para monitoramento de condições de máquinas usando parâmetros como vibração, corrente e temperatura, onde os dados de máquina selecionados são exibidos em tempo real. O usuário tem a opção de configurar os valores de limite e um alerta é gerado, utilizando estes valores como referência.

Estes trabalhos, de forma geral, apresentam algumas limitações tanto na escolha dos componentes utilizados, tais como os microcontroladores e os módulos de sensoriamento, como também nas tecnologias e comunicações empregadas. Esses aspectos são importantes por dois principais motivos: a escolha de componentes de alta oferta e de baixo custo facilita a escalabilidade do projeto ao reduzir o custo final de construção do sensor e o tempo de aquisição dos componentes; e a definição da comunicação que será utilizada aumenta a viabilidade de instalação nas plantas industriais, pois ter uma alternativa à conexão Wi-Fi facilita a instalação dos sensores em locais que não contam com tal infraestrutura.

Com base nestes princípios, foi elaborada a tabela abaixo (Tabela 2), contendo as proposições e hipóteses levantadas, que serviram como referência para o desenvolvimento do artefato.

Tabela 2 - Proposições e hipóteses levantadas

Proposições	**Hipóteses**
Apresentar componentes com alta oferta e baixo custo no mercado	Deve aumentar a escalabilidade do projeto, tendo em vista a maior facilidade de aquisição dos componentes utilizados
Possibilitar conexão com rede celular, além da rede Wi-Fi	Deve aumentar a viabilidade de implementação do projeto por meio da facilidade de conexão com a Internet.

Fonte: Elaboração própria.

4.2 DESENVOLVIMENTO DO ARTEFATO

Em função das proposições levantadas acima, deu-se início ao desenvolvimento do artefato, que como comentado anteriormente, trata-se de um produto mecatrônico. Este desenvolvimento foi dividido em três principais etapas: escolha do hardware, desenvolvimento do firmware e preparação da plataforma de dados.

4.2.1 Escolha do *hardware*

A escolha do hardware seguiu as diretrizes definidas pelo conjunto de proposições e hipóteses da etapa anterior. Portanto, a prioridade principal era a de desenvolver um sensor com um foco na viabilidade de implementação na fábrica,

tanto do ponto de vista de custo, quanto do ponto de vista de disponibilidade dos componentes no mercado. A escolha abrangeu a seleção de um microcontrolador, um modem e um acelerômetro.

4.2.1.1 Microcontrolador e Modem

Ao pesquisar os principais microcontroladores presentes nos módulos de desenvolvimento comercializados no mercado, alguns fabricantes se destacam: Microchip, Espressif e ST. Com base nesses fabricantes, foram separados alguns modelos para comparação na tabela abaixo (Tabela 3), levando em consideração o custo unitário, o consumo energético e a conectividade integrada ao módulo.

Tabela 3 - Comparação entre principais microcontroladores

Microcontrolador	Fabricante	Custo	Corrente nominal	Conectividade
ATMEGA328	Microchip	R$ 48,00	19mA	-
ESP32	Espressif	R$ 46,00	20mA (até 240mA)	Wi-Fi, BT e BLE
STM32F103	ST	RS 39,00	13mA	-

Fonte: Elaboração própria

Como é possível observar no tabela acima, dentre os principais modelos encontrados, o ESP32 se destaca pelos diferentes tipos de conectividade sem fio de forma embarcada. Tanto a Microchip quanto a ST também suportam esse tipo de conexão, mas seria necessário adicionar um módulo adicional ou partir para outro modelo de placa de desenvolvimento, que seria menos disponível no mercado.

Segundo Pereira (2020), o ESP32 pertence a uma série de microcontroladores de baixo custo e consumo energético, que tem sua indicação para projetos de IoT devido principalmente à capacidade de conectividade e à quantidade de memória RAM, o que viabiliza operações de processamento de dados, tais como as coletas de um acelerômetro.

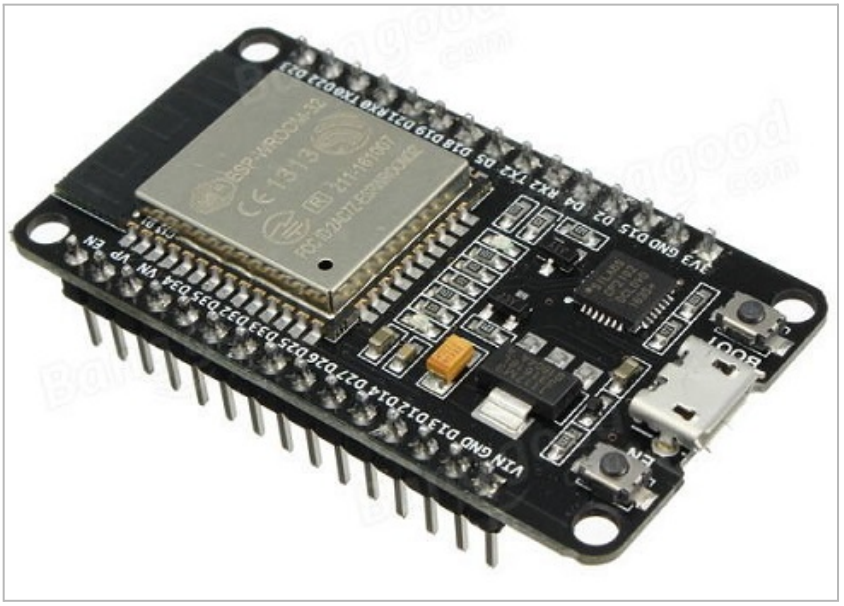
Além disso, a fabricante Espressif, disponibiliza um protocolo proprietário de comunicação que funciona por meio do protocolo Wi-Fi, ou seja, também opera na faixa de 2,4GHz, chamado de ESP NOW. Ele foi desenvolvido para proporcionar uma rápida comunicação entre microcontroladores da própria fabricante, de modo

que, mesmo operando na frequência do Wi-Fi, ele pode alcançar distâncias maiores e latências menores.

O fato do protocolo ESP NOW proporcionar tempos de resposta e conexão reduzidos faz com que o sensor precise ficar menos tempo conectado, o que diminui o consumo energético. A Tabela 3 mostrou que o ESP32 apresenta um consumo de até 240mA, e isso se dá justamente no momento de transmissão de dados, ou seja, quanto menor o tempo de transmissão, maior a autonomia da bateria do sensor.

Por se tratar de um protocolo proprietário, ele restringe sua comunicação apenas aos microcontroladores fabricados pela Espressif. Dessa forma, para que seu uso seja possível, tanto o sensor quanto o receptor devem ser compatíveis. No caso do sensor, a placa de desenvolvimento utilizada especificamente é a ESP32 DOIT DevKit (Figura 6).

Figura 6 - Módulo ESP32 DOIT DevKit



Fonte: ESP32-DevKitC (2022).

Já no caso do receptor, ao levar em conta a facilidade de implementação de hardware e disponibilidade de componentes no mercado brasileiro, a escolha do módulo TTGO TCALL 1.4 (Figura 7) se torna interessante, já que ela apresenta de forma embarcada o modem Simcom SIM800, que tem capacidade de conexão com a rede de celular.

Figura 7 - Módulo TTGO TCALL 1.4

Fonte: LILYGO TTGO T-Call V1.4 ESP32 (2022)

O SIM800 é um módulo para comunicação através dos protocolos GSM e GPRS que permite que o microcontrolador acesse a rede de telefonia móvel 2G e possa utilizar da transmissão de dados, o que se torna um grande benefício em locais sem conexão com uma rede Wi-Fi. Outras vantagens do SIM800 são sua boa captação de sinal, baixo custo e baixo consumo. A única exigência para o seu funcionamento inclui a utilização de um chip SIM com acesso a um plano de celular. (MENEZES, 2020).

O Chip utilizado neste projeto será o comum encontrado em diversos estabelecimentos, tais como mercados, com o acréscimo de crédito na linha para permitir a conexão com os dados móveis.

4.2.1.2 Acelerômetro

Assim como a escolha do microcontrolador e do modem foi com base na viabilidade de implementação, tanto em relação ao custo e à facilidade de implementação, quanto em relação à disponibilidade no mercado, a escolha do acelerômetro não foi diferente. Na Tabela 4 abaixo estão os principais modelos encontrados, levando em conta o custo, protocolo de comunicação e capacidade de medir temperatura de forma integrada.

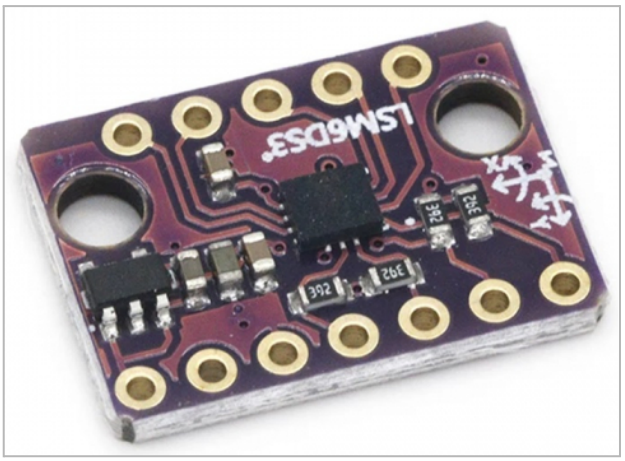
Tabela 4 - Comparação entre principais acelerômetros

Acelerômetro	Custo	Protocolo	Temperatura
LSM6DS3	R$ 22,00	SPI, I2C	Sim
MPU6050	R$ 14,00	I2C	Não
ADXL345	RS 27,00	SPI, I2C	Não

Fonte: Elaboração própria

Consultando a tabela acima, é possível concluir que, embora o MPU6050 apresente um custo menor, o modelo LSM6DS3 (Figura 8) é mais vantajoso para a aplicação neste projeto, já que facilita a implementação da leitura de temperatura e permite a comunicação por meio do protocolo SPI, que apresenta uma maior simplicidade de configuração comparado ao I2C.

Figura 8 - Acelerômetro LSM6DS3



Fonte: Adafruit (2022).

4.2.2 Desenvolvimento do *firmware*

Com todos os componentes críticos definidos, o projeto passa a necessitar da programação destes dispositivos para que operem da forma esperada. Com isso, temos o desenvolvimento do *firmware* segmentado em alguns pilares:

- Comunicação entre microcontrolador e acelerômetro: ocorre no sensor, em que o ESP32 coleta os dados do acelerômetro pelo protocolo SPI e realiza um processamento para calcular os valores RMS;

- Comunicação entre sensor e receptor: ocorre em ambos, em que há necessidade de implementar a transmissão das coletas do sensor para o receptor por meio do protocolo ESP NOW;
- Comunicação entre microcontrolador e modem: ocorre no receptor, em que o ESP32 envia os dados recebidos por ESP NOW para o servidor por meio do modem SIM800.

Toda programação do *firmware* foi feita dentro do *software* Microsoft Visual Code com o auxílio da extensão PlataformIO, utilizando a linguagem de programação C++ e o framework Arduino 3.5.0.

4.2.2.1 Comunicação entre microcontrolador e acelerômetro

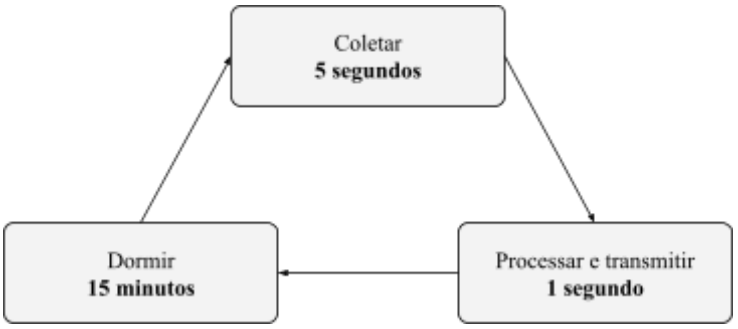
Para desenvolver a comunicação entre a ESP32 e o LSM6DS3, foi utilizada a biblioteca da Adafruit, dentro das extensões do PlatformIO. A partir disso, a configuração do protocolo SPI e dos registradores do acelerômetro ficaram mais simples. Esse código foi inspirado na própria documentação do criador da biblioteca, que disponibilizou um projeto dentro do Github (Adafruit, 2022).

Após a definição da estrutura básica de código do sensor, foi possível obter os dados de temperatura e vibração medidos pelo LSM6DS3. Para preparar esses dados para envio, foi implementada a função do cálculo de valores RMS, tanto para vibração, quanto para temperatura. Essa função recebe como parâmetro uma lista dos dados coletados e com base nos valores e na quantidade de dados, calcula a média final daquela variável.

Para complementar o funcionamento e garantir uma maior duração de bateria do sensor, foi implementada uma lógica de repouso do sensor logo após transmitir a coleta. Esse repouso é definido pela Espressif como *deep sleep* e reduz a corrente do microcontrolador para 10μA.

Na prática, o diagrama de funcionamento do sensor pode ser visto abaixo (Figura 9).

Figura 9 - Diagrama de funcionamento do sensor



Fonte: Elaboração própria

4.2.2.2 Comunicação entre sensor e receptor

Para que o sensor envie as coletas para a internet, é necessário passar os dados para o receptor. Como já abordado acima, esta transmissão foi realizada utilizando o protocolo ESP NOW, já que ele apresenta uma velocidade de conexão e transmissão de dados coerente com a necessidade do projeto, o que implica em um tempo menor de transmissão e, consequentemente, menor gasto de bateria do sensor.

Assim como a biblioteca da Adafruit, o ESP NOW também apresenta sua documentação no Github (ARDUINO, 2022), com exemplos de aplicações realizadas pela própria Espressif. Esse protocolo se baseia na função de *callback*, que é automaticamente chamada quando um dado é recebido ou enviado.

Esta etapa do projeto é uma das mais importantes para o funcionamento do conjunto sensor-receptor, pois algumas regras de operação precisam ser levadas em consideração. O primeiro aspecto é que o sensor não sabe qual é seu receptor, ou seja, não existe uma configuração inicial que vincula o conjunto. Dessa forma, o sensor precisa se adaptar aos receptores disponíveis no ambiente no momento em que for transmitir algum dado. O segundo aspecto é que a comunicação precisa ter um filtro que diferencia quem são os sensores e quem são os receptores dentro da rede, pois seria um problema caso um sensor enviasse uma coleta para outro sensor.

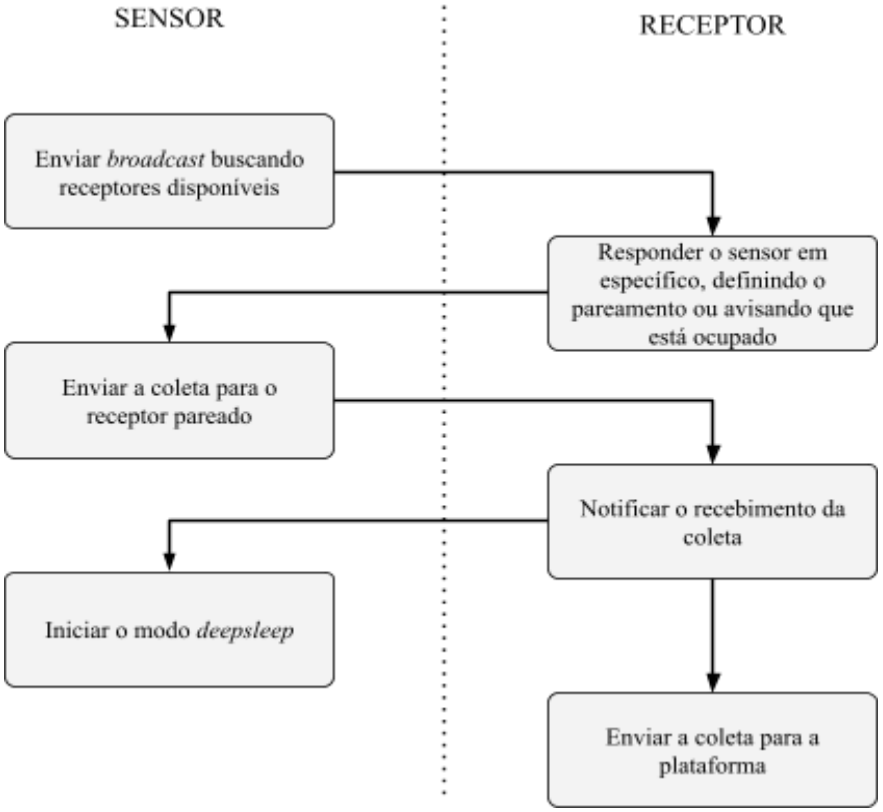
Para solucionar esses requisitos e melhorar o funcionamento e segurança da comunicação, foram implementadas as seguintes lógicas:

1. O sensor procura pelos receptores por meio de uma mensagem *broadcast*, que na prática é uma mensagem sem um destino definido, ou seja, são enviados para todas os ESP32 presentes no local;

2. Antes do sensor enviar uma coleta, ele envia uma mensagem de pareamento que apenas os receptores estão programados para responder, de forma que a comunicação entre dois sensores fica bloqueada;
3. Quando um receptor está ocupado, ou seja, já está enviando uma outra coleta, ele avisa o sensor. Caso todos os receptores estejam ocupados, o sensor entra no modo *deepsleep* com um fator de dessincronia, que na prática, significa esperar pelos 15 minutos padrões mais 10 segundos de espera, para que com o tempo, o sensor encontre seu intervalo de operação com algum receptor. Dessa forma, após a instalação dos sensores, cada sensor vai automaticamente se ajustando para encontrar o melhor intervalo de comunicação com os receptores.

Utilizando as especificações e conhecimentos destes componentes, foi desenvolvido o diagrama de conexão e transmissão de dados entre sensor e receptor para ilustrar as regras de comunicação (Figura 10). O Apêndice A apresenta as principais definições da comunicação dentro do *firmware*.

Figura 10 - Diagrama de funcionamento da conexão sensor-receptor



Fonte: Elaboração própria

Como é possível observar, o sensor não envia diretamente o dado coletado para o receptor, de forma que antes uma mensagem é enviada para verificar a disponibilidade de comunicação. Isso previne a perda de dados, já que o receptor poderia estar se comunicando com outro sensor no exato momento.

4.2.2.3 Comunicação entre microcontrolador e modem

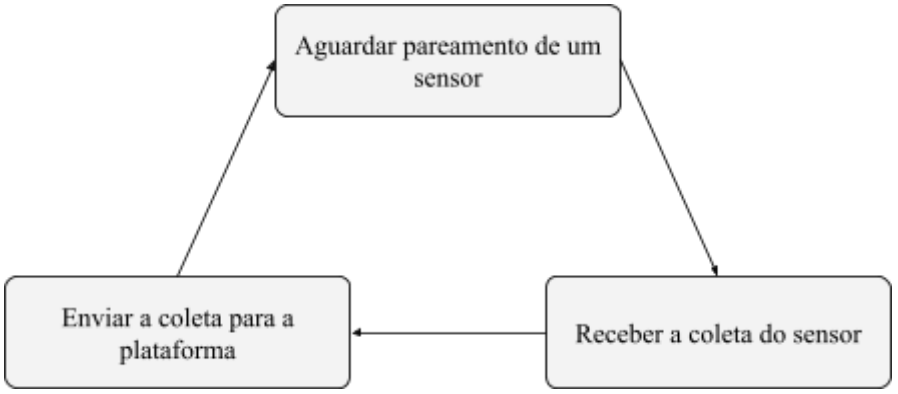
Para desenvolver a comunicação entre ESP32 e SIM800, por meio do protocolo UART, foi utilizada a biblioteca TinyGSM, inclusa dentro das extensões do PlatformIO. Assim como apresentado sobre a biblioteca da Adafruit e o acelerômetro, essa biblioteca do modem já define todas as funções principais para seu funcionamento. Dessa forma, é necessário chamar apenas algumas dessas funções para inicializar o modem.

Vale ressaltar que, quando se trabalha com operadores de celular, é importante definir a autenticação da linha, de forma que cada operadora apresenta a sua. No caso deste projeto, foi utilizado um chip da Vivo, sendo necessário realizar algumas configurações específicas de autenticação desta operadora.

A partir da configuração básica do modem utilizando a TinyGSM, foi necessário implementar a comunicação com o servidor do ThingSpeak, que é a plataforma utilizada para receber os dados e será abordada em detalhes abaixo.

Para resumir o funcionamento padrão do receptor, foi elaborado um diagrama, que pode ser visualizado abaixo (Figura 11).

Figura 11 - Diagrama de funcionamento do receptor



Fonte: Elaboração própria

O funcionamento do receptor é mais simples e é completamente dependente do sensor. Isto ocorre porque a função do receptor é única e exclusivamente enviar os dados coletados pelo sensor, de forma que, quando um receptor está ligado fora do alcance de sensores, ele fica totalmente ocioso aguardando por uma comunicação.

Pelo fato do receptor depender da conexão com rede celular, que tem velocidade variável em função do ambiente, não é possível prever a duração de cada etapa do seu funcionamento. Sabe-se, porém, que o receptor fica ocupado enquanto está comunicando com um sensor e enviando sua coleta para a plataforma e, portanto, a capacidade de sensores que ele consegue lidar varia de acordo com a velocidade da rede móvel, em que, quanto mais rápido o envio da coleta para a plataforma, mais tempo livre para receber coletas e, consequentemente, maior a frequência de monitoramento.

4.2.3 Preparação da plataforma de dados

A plataforma é o componente do sistema proposto que tem como objetivo receber os dados coletados pelos sensores. Ela pertence à empresa MathWorks, criadora do software Matlab, de forma que algumas funcionalidades presentes nele estão inclusas gratuitamente no ThingSpeak, tais como a visualização de gráficos em tempo real e a possibilidade de executar cálculos através de gatilhos programados.

A descrição da plataforma está dividida em duas partes: canal do sensor e função gatilho.

4.2.3.1 Canal do sensor

O canal de recebimento é a componente da plataforma que permite a visualização em tempo real dos dados coletados. Ele corresponde a um link de internet para enviar as coletas realizadas. Cada sensor necessita de um canal exclusivo, já que os parâmetros de configuração limite de vibração e temperatura variam para cada máquina monitorada.

O receptor precisa identificar o canal de cada sensor, o que é feito por meio de uma definição no seu *firmware*. A lista que define a relação dos sensores contém o número de identificação do canal e o endereço MAC do sensor. O MAC (Media

Access Control) é um endereço físico e único utilizado para garantir a identidade de um dispositivo dentro de uma rede, tal como o Wi-Fi e o ESP NOW.

Após a configuração inicial do canal de um sensor, a interface básica já passa a exibir os dados coletados em tempo real, como a figura abaixo mostra (Figura 12).

Figura 12 - Exemplo da interface de um canal do ThingSpeak

Fonte: Elaboração própria.

No caso deste canal, os valores exibidos já estão com a temperatura em Celsius e a aceleração em G. Para que isto seja alterado, é necessário a configuração tanto do *firmware* quanto da plataforma.

4.2.3.2 Função gatilho de alerta

Com o canal de dados criado, a plataforma passa a receber os dados. O próximo passo é o desenvolvimento do script que contém as instruções necessárias quando um novo dado é coletado. Este código é responsável por definir os limites, ou *thresholds*, de aceleração e temperatura, que são estabelecidos dentro da plataforma no momento da instalação do sensor. A vantagem deste dado pertencer à plataforma é que não é necessário realizar nenhuma alteração de *firmware* para alterar o comportamento dos alertas gerados. Na figura abaixo (Figura 13), é possível observar a estrutura básica da função de alarme, que apresenta comentários para explicar linha por linha.

Figura 13 - Código da função gatilho do ThingSpeak

```
% Thresholds/limites de temperatura e aceleração
temp_threshold = 50
accel_threshold = 3

% Numero de identificação e autenticação do canal do ThingSpeak
channel_id = 1874092;
channel_read_key = '84SLNJH6R8E495QS';

% Chave de API do ThingSpeak usada para autenticação dos alertas de email
api_key = 'TAK1+Aa5Cg5iAQhbLF3';

% Definição padrão dos alertas do ThingSpeak
alert_url="https://api.thingspeak.com/alerts/send";
options = weboptions("HeaderFields", ["ThingSpeak-Alerts-API-Key", api_key ]);

% Informações do email de alerta
email_subject = sprintf("Alerta de criticidade de máquina %d", channel_id);
email_body = sprintf("OK");
should_alert = 0

% Leitura dos dados do canal do sensor (temperatura e aceleração)
temp_data = thingSpeakRead(channel_id,'NumDays',90,'Fields',1,ReadKey=channel_read_key
accel_x_data = thingSpeakRead(channel_id,'NumDays',90,'Fields',2,ReadKey=channel_read_
accel_y_data = thingSpeakRead(channel_id,'NumDays',90,'Fields',3,ReadKey=channel_read_
accel_z_data = thingSpeakRead(channel_id,'NumDays',90,'Fields',4,ReadKey=channel_read_

% Verificar se o canal foi lido corretamente
if isempty(temp_data)
    fprintf("Sem dados no canal dentro do periodo lido\n");
else
    last_temp = temp_data(end);
    last_accel_x = accel_x_data(end);
    last_accel_y = accel_x_data(end);
    last_accel_z = accel_x_data(end);

    if (last_temp >= temp_threshold) || (last_accel_x >= accel_threshold) || (last_acc
        email_body = 'Temperatura e/ou aceleração fora do limite estabelecido';
        should_alert = 1;
    end
end

if (should_alert == 1)
    try
        webwrite(alert_url , "body", email_body, "subject", email_subject, options);
        fprintf("Alerta enviado");
    catch someException
        fprintf("Failed to send alert: %s\n", someException.message);
    end
else
    fprintf("Maquina OK\n");
end
```

Fonte: Elaboração própria.

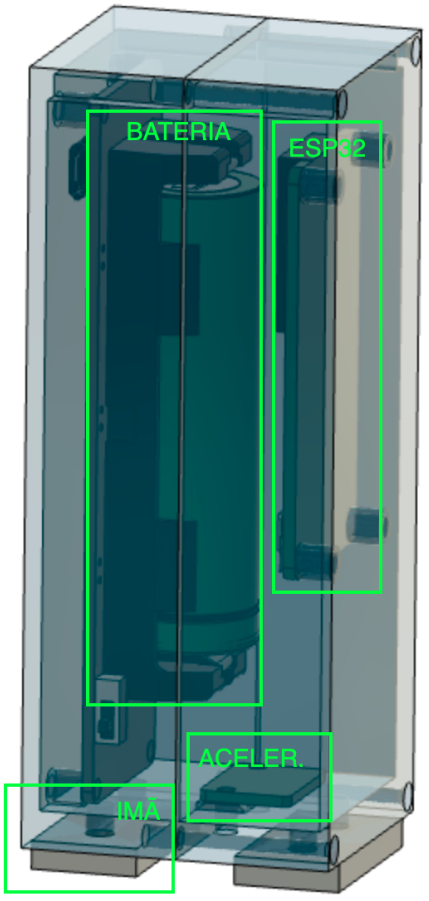
De forma simplificada, este script apresenta inicialmente a definição dos *thresholds* de temperatura (*temp_threshold* em Celsius) e aceleração (*accel_threshold* em G), seguidos das configurações do canal e do serviço de email. Após isso, os dados de temperatura (*temp_data*) e aceleração (*accel_x_data*, *accel_y_data* e *accel_z_data*) do sensor são obtidos e comparados com os *thresholds* definidos: se algum deles superar o threshold, um alerta é gerado automaticamente no email.

4.2.4 Desenho dos invólucros

Para permitir a fixação e proteção do sensor e do receptor, foi necessário desenhar seus respectivos invólucros. O *software* utilizado para o desenho foi o Autodesk Fusion 360, em conjunto com o *software* Prusa Slicer, que permite imprimir o protótipo em uma impressora da fabricante Prusa.

Para o desenho do sensor, foi levado em consideração a necessidade de colocar uma bateria de Li-Ion 18650, que são facilmente encontrados no mercado e garantem uma alimentação contínua para o sensor, até que a carga se esgote. Além disso, o sensor precisa ser fixado nas máquinas industriais, que normalmente são de metal, de forma que foi adicionado um ímã para tornar a instalação mais simples. O desenho final do sensor pode ser visto na figura abaixo (Figura 14).

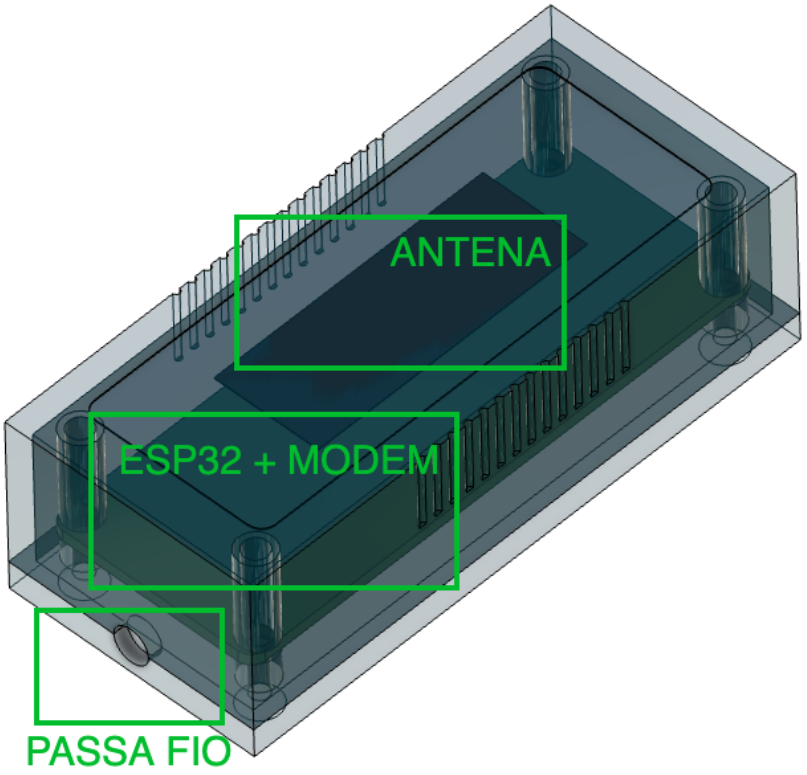
Figura 14 - Desenho do invólucro do sensor



Fonte: Elaboração própria

Para o desenho do receptor, foi levado em consideração a necessidade da abertura para passagem do cabo de alimentação, que vem de uma fonte genérica de 5V e de um espaço interno para a colagem da antena de rede celular. Com isso, o desenho final do receptor pode ser visto na figura abaixo (Figura 15).

Figura 15 - Desenho do invólucro do receptor



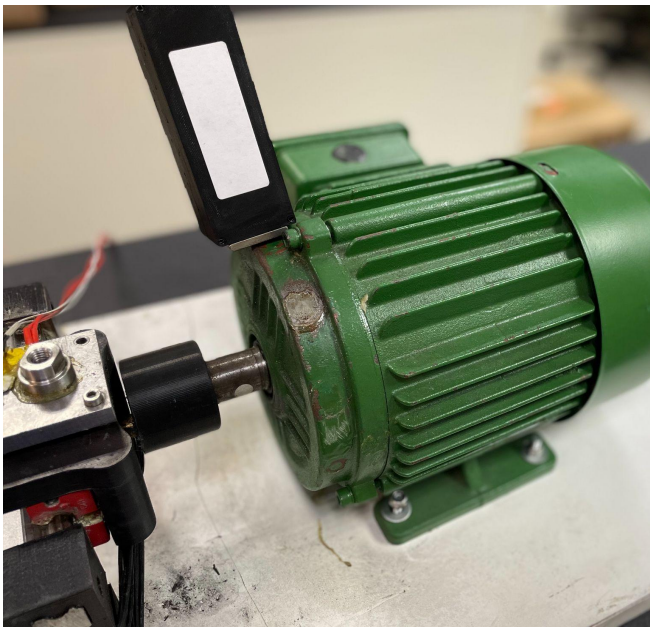
Fonte: Elaboração própria

Os detalhes dos desenhos estão exibidos no Apêndice B.

4.3 DEMONSTRAÇÃO E AVALIAÇÃO

Para testar na prática a aplicação do projeto, todos os aspectos do sensor e receptor foram avaliados em uma utilização real do conjunto. Estabeleceu-se um experimento de coleta de simulação de uma uma folga mecânica de tipo B em um motor para testar a função gatilho e o recebimento do email de alerta. As figuras abaixo mostram a fixação do sensor no motor (Figura 16) e a conexão do receptor à tomada, que neste caso, ficou na tomada convencional mais próxima do motor, em torno de 10 metros de distância do sensor. (Figura 17).

Figura 16 - Fixação do sensor no motor



Fonte: Elaboração própria

Figura 17 - Conexão do receptor na tomada

Fonte: Elaboração própria

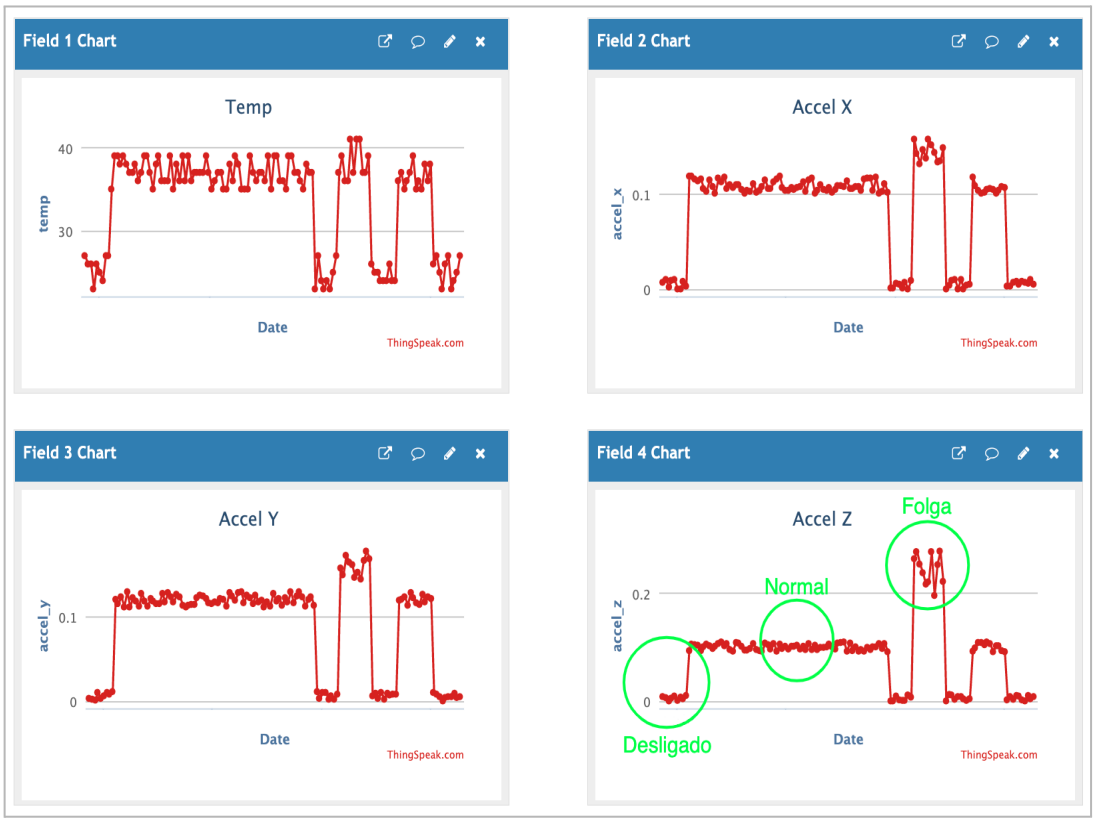
A princípio, as coletas foram analisadas com o motor funcionando de forma ideal, ficando claro alguns aspectos. O primeiro deles é que por meio do gráfico é possível observar os pontos de funcionamento do motor, ou seja, quando ele está ligado e quando está desligado. Dessa forma, seria possível utilizar a função gatilho para avisar quando o motor está sendo ligado ou até calcular o tempo de utilização do motor. Porém, como o foco é testar um alerta, a função gatilho foi configurada para a identificação de folga mecânica. Vale ressaltar que os ajustes de limite de

temperatura e de aceleração foram definidos com base nos dados do próprio funcionamento normal do motor.

O segundo aspecto para destacar é que cerca de 5% das coletas foram perdidas e essa perda se deu pela falha de transmissão entre sensor e receptor ou entre receptor e servidor. Ambas as conexões não apresentam uma lógica de reenvio em caso de falha, o que faz com que o dado seja perdido. Além disso, quando um segundo sensor de teste foi ligado no ambiente, o primeiro sensor teve que reajustar sua janela de operação, o que fez com que o primeiro perdesse algumas coletas. Uma possível melhoria para este aspecto seria implementar um armazenamento de coletas na memória do sensor quando o envio falhasse. Como 95% das coletas foram realizadas com sucesso, foi possível continuar com os testes e simular de fato uma aplicação de folga mecânica da fixação do motor.

Para realizar a simulação de folga mecânica de tipo B, os parafusos de fixação da base do motor foram afrouxados, o que fez com ele vibrasse acima da média. Os dados de vibração e temperatura coletados pelo sensor estão representados pelos gráficos contidos na Figura 18, retirados do ThingSpeak.

Figura 18 - Coletas de vibração e temperatura realizadas pelo sensor



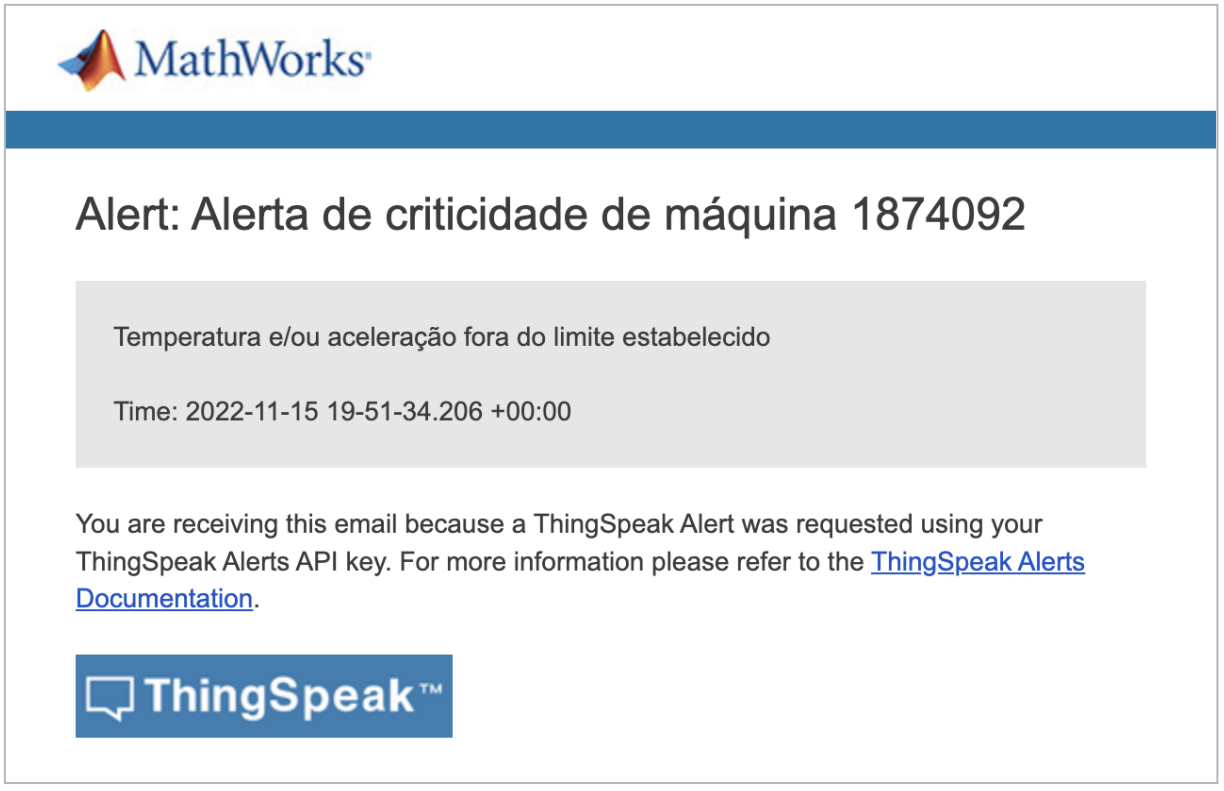
Fonte: Elaboração própria

A figura acima exibe três diferentes estados do motor, que são mais evidentes no último gráfico (*Accel Z*):

1. Valores baixos: motor desligado;
2. Valores médios: motor em funcionamento normal;
3. Valores altos: motor em funcionamento com folga mecânica tipo B.

Dessa forma, como o *threshold* foi configurado para uma valor acima do normal e abaixo do valor coletado de vibração, assim que a coleta foi recebida pelo ThingSpeak, um alerta foi gerado no email, conforme Figura 19.

Figura 19 - Email de alerta gerado



Fonte: Elaboração própria

5 DISCUSSÃO DOS RESULTADOS

Ao retomar as proposições e hipóteses levantadas no início do projeto (Tabela 2), devem ser destacadas algumas conclusões obtidas após a etapa de demonstração. Em relação à escolha de componentes com alta oferta e baixo custo no mercado, a pesquisa demonstrou a viabilidade de construção do artefato conforme a hipótese proposta, dado que foi possível gerar um produto funcional (artefato) e que o experimento de simulação demonstrou a capacidade de cumprir a missão.

Quanto aos custos, obteve-se um Custo de Material do protótipo do sensor em torno de 70 reais e do receptor em torno de 200 reais. Ao comparar com a solução de Siddhartha et al. (2020), que apresenta também uma solução de hardware voltada para alertas com base em temperatura e vibração, apenas um dos componentes utilizados (Raspberry PI 4) já ultrapassa todo custo do projeto, pois no Brasil, em preços atuais foi cotado em torno de 800 reais.

Além disso, como o projeto apresenta um conjunto sensor-receptor no qual o receptor é capaz de lidar com alguns sensores, para monitorar mais uma máquina em uma mesma região, por exemplo, é necessário apenas a utilização de um segundo sensor, aproveitando o receptor já instalado. Com isso, o custo de escalabilidade da solução dentro de uma indústria fica ainda menor, levando em conta que o sensor tem um custo de material reduzido comparado ao receptor.

Já em relação a segunda hipótese levantada, em que a possibilidade de conectar na rede celular facilitaria o processo de instalação e aumentaria a escalabilidade do projeto, foi possível constatar que de fato a rede celular torna a solução *Plug and Play*, visto que foi necessário apenas conectar o receptor na tomada para que o sensor enviasse as coletas para a plataforma. Não foi necessário configurar nenhum Wi-Fi no sistema, embora fosse possível.

Como mostrado na seção de Demonstração e Avaliação, cerca de 5% das coletas foram perdidas e isso se deu, majoritariamente, em função das instabilidades na rede celular durante o envio das coletas. Considerando a lógica atual de envio, o conjunto sensor-receptor fica muito sensível às possíveis falhas de envio, já que não apresenta nenhuma nova tentativa em caso de perda de conexão. Dessa forma, para aumentar a confiabilidade do sensor, seria importante adicionar uma função que executasse essas tentativas.

Por fim, a pesquisa permitiu identificar aspectos observados durante a aplicação do produto que não estavam diretamente relacionados às proposições iniciais, mas que podem servir como possíveis melhorias neste artefato, tais como:

- Melhoria do mecanismo de coleta de dados, tanto da configuração dos parâmetros do acelerômetro, quanto das funções de processamento de dados, tal como a troca do função RMS pela função FFT, que permitiria a análise de espectro de vibração;
- Implementação de armazenamento de dados coletados no sensor para evitar a perda quando a transmissão falhar;
- Implementação de um algoritmo de verificação de integridade de dados, tanto no receptor quanto no servidor, para garantir que todos os dados enviados são de fato dados coletados pelo sensor;
- Criação de uma plataforma personalizada para o recebimento de dados do sensores, que possibilitaria enviar configurações diferentes para o sensor e aumentaria ainda mais a escalabilidade do projeto;
- Melhoria dos invólucros para permitir aplicações em ambientes expostos à chuva;
- Adição da compatibilidade com rede celular 3G e 4G, que aumentaria a área de cobertura das operadoras móveis.

6 CONCLUSÕES

A pesquisa indica que é possível criar um conjunto sensor-receptor capaz de alertar uma falha potencial em uma máquina fora dos padrões ideais de funcionamento, tanto pela vibração, quanto pela temperatura, com custo abaixo dos demonstrados na literatura.

Conclui-se que o objetivo inicial de construir um conjunto focado na viabilidade de implementação, que leva em conta a disponibilidade de componentes no mercado e o custo reduzido de hardware foi atingido, com a escolha de componentes e tecnologias que estão mais acessíveis ao consumidor. A possibilidade de conexão com rede celular de fato aumentou a facilidade de instalação do sistema.

A solução encontrada, embora simples, já conseguiu cumprir a missão em ambiente controlado. Foi possível também identificar uma lista de pontos de melhoria que podem torná-la ainda mais robusta. A expectativa de melhoria na infraestrutura de internet em muitos ambientes industriais pode impulsionar nos próximos anos estas soluções de baixo custo para problemas de manutenção, indicando que este campo poderá se expandir à médio prazo.

REFERÊNCIAS

ADAFRUIT. LSM6DS3. Disponível em: <https://www.adafruit.com/product/5543>. Acesso em: 20 nov. 2022.

ADAFRUIT. LSM6DS3 Github Project. Disponível em: <https://github.com/adafruit/Adafruit_LSM6DS>. Acesso em: 20 nov. 2022.

ARDUINO. ESP32 Github Project. Disponível em: <https://github.com/espressif/arduino-esp32>. Acesso em: 18 nov. 2022.

Associação Brasileira de Normas e Técnicas (ABNT), 1994.

CRANDALL, S.H. (ed.),1958, Random Vibration, New York: MIT Press/Wiley.

CYRINO, L. Falhas Evolução até Quebra. Manutenção em Foco, 22 Outubro 2017. Disponível em: <https://www.manutencaoemfoco.com.br/falhas-evolução-ate-quebra>. Acesso em: 10 de julho de 2022.

DEFENDI, V. Sistema de Manutenção Preditiva Aplicado a Compressores Radiais Industriais Utilizando Análise de Vibração. Bento Gonçalves, 2020.

DILLENBURG, M.R. Alternativas de Aplicação do Serviço GPRS da Rede Celular GSM em Telemetria Pela Internet.

ESP32-DevKitC. Development Boards - Espressif Systems. Disponível em: <https://www.espressif.com/en/products/devkits/esp32-devkitc/overview>. Acesso em: 20 nov. 2022.

FELISBERTO, L.A. Sistemas de Comunicação Sem Fio (Wireless). Angicos, 2018. Universidade Federal Rural do Semi-Árido.

GALLI, V. Manutenção Preditiva por Análise de Vibração Mecânica em Máquinas Rotativas: Estudo de caso, 2017.

GONZÁLEZ, R. D. Desenvolvimento de um Protótipo Analisador de Vibração de Baixo Custo Para Uso em Manutenção Preditiva. Florianópolis, 2014. Dissertação (Mestrado em Engenharia Mecânica) - Universidade Federal de Santa Catarina.

GUPTA, T., AMRUTHNATH, N. Fault Class Prediction in Unsupervised Learning Using Model-Based Clustering Approach. International Conference on Information and Computer Technologies. 2018.

HASHEMIAN, H.M. Aging Management through On-Line Condition Monitoring. In: Presented at the PLIM + PLEX 2006 Conference, Paris, France.

HASHEMIAN, H.M.Wireless Sensors for Predictive Maintenance of Rotating Equipment in Research Reactors. Annals of Nuclear Energy, Vol 38. 2011.

HEVNER, A. et al. Design Science Research in Information Systems. 2004.

LILYGO TTGO T-Call V1.4 ESP32 Wireless Module SIM Antenna SIM Card Module - Shenzhen Xin Yuan Electronic Technology Co., Ltd. Disponível em: <http://www.lilygo.cn/claprod_view.aspx?TypeId=62&Id=1403&Fld=t28:62:28>. Acesso em: 20 nov. 2022.

MENEZES, L.M.S. Desenvolvimento de Sistema de Monitoramento IoT Para Rios Urbanos de Fácil Implantação e Baixo Custo. São Carlos, 2020.

MONCHY, François. A Função Manutenção - Formação para a Gerência da Manutenção Industrial. São Paulo: Editora Durban Ltda., 1989.

PEFFERS, K. et al. A Design Science Research Methodology for Information Systems Research. 2007.

PEREIRA, M.R.S. A Aplicação do Microcontrolador ESP32 no Ensino: Medindo Posições em Função do Tempo Utilizando o Sensor VL53L0X Associado ao ESP32. Macapá, 2020.

SCHEFFER, Cornelius; GIRDHAR, Paresh. Practical machinery vibration analysis and predictive maintenance. Elsevier, 2004. ISBN 0-7506-6275-1.

SIDDHARTHA, B., et al. IoT Enabled Real-Time Availability and Condition Monitoring of CNC Machines. The IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2020.

THOMPSON, Sylvanus P. Calculus Made Easy. Macmillan International Higher Education. p. 185, 1965. ISBN 9781349004874.

UHLMANN E., et al. Smart Wireless Sensor Network and Configuration of Algorithms for Condition Monitoring Applications, Journal of Machine Engineering, Vol. 17, No. 2, 2017.

WYREBSKI, J. Manutenção Produtiva Total – Um Modelo Adaptado. Dissertação (Mestrado em Engenharia de Produção e Sistemas) - Centro Tecnológico, Universidade Federal de Santa Catarina. Florianópolis, 1997.

APÊNDICE A - COMUNICAÇÃO ENTRE SENSOR E RECEPTOR

```h
#pragma once

#include <Arduino.h>
#include <CRC32.h>
#include <WiFi.h>
#include <esp_now.h>

Caio Mascarin, 17 minutes ago | 1 author (Caio Mascarin)
struct EspnowMessage {
    int acknoledge = 0;
    int channelId = 0;
    char channelKey[20];
    float temp = 0;
    float accelX = 0;
    float accelY = 0;
    float accelZ = 0;
    int bootCount = 0;
    int checksum = 0;
};

enum ACKNOLEDGES {
    SEARCHING_RECEIVER = 100,
    SENDIND_SAMPLE,
    RECEIVER_FREE,
    RECEIVER_BUSY,
    RECEIVER_EOT,
};

extern uint8_t broadcastAddress[];
extern int espnowTimeoutMs;

bool beginEspnow(esp_now_recv_cb_t rx_cb);
int calculateChecksum(EspnowMessage message);
bool sendMessage(const uint8_t *macAddr, EspnowMessage message);
bool isValidMessage(EspnowMessage message);
bool waitResponse(int timeout);
void resetTimeout();
```

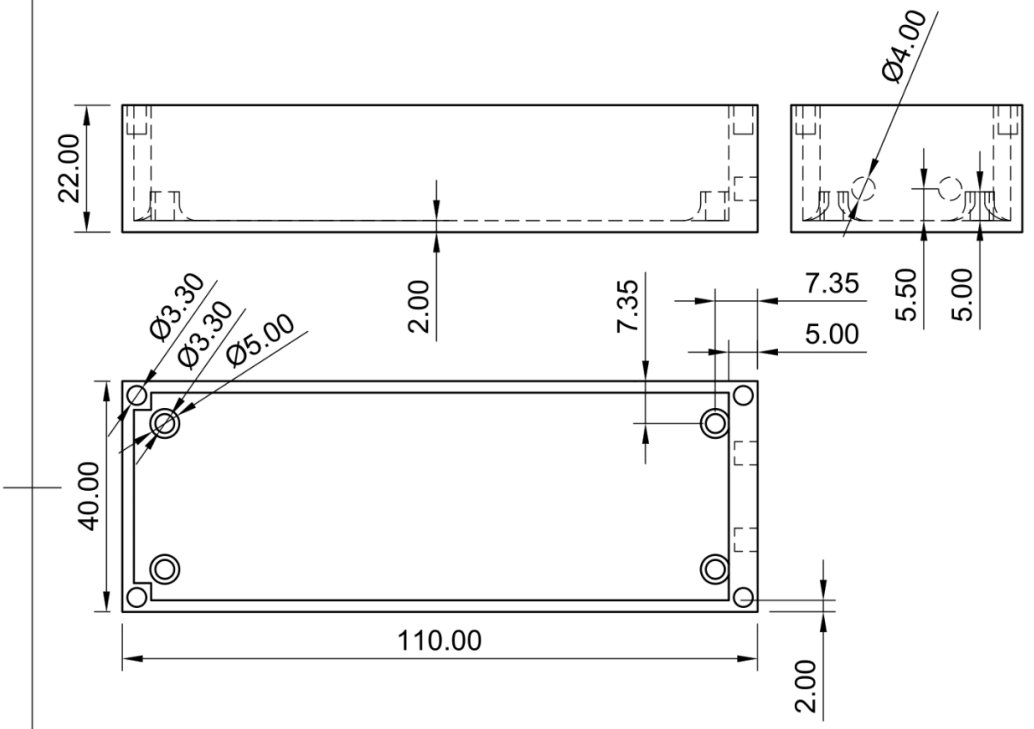
```cpp
#include "common.h"

uint8_t broadcastAddress[] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
int espnowTimeoutMs = 0;

bool beginEspnow(esp_now_recv_cb_t rx_cb) {
    WiFi.mode(WIFI_STA);
    if (esp_now_init() != ESP_OK) {
        Serial.println("Failed to init ESPNOW!");
        return false;
    }
    esp_now_register_recv_cb(rx_cb);
    return true;
}

int calculateChecksum(EspnowMessage message) {
    return CRC32::calculate(&message, sizeof(EspnowMessage));
}

bool sendMessage(const uint8_t *macAddr, EspnowMessage message) {
    esp_now_peer_info_t peerInfo;
    memcpy(peerInfo.peer_addr, macAddr, 6);
    peerInfo.channel = 0;
    peerInfo.encrypt = false;
    if (esp_now_add_peer(&peerInfo) != ESP_OK) {
        Serial.println("Failed to add peer");
        return false;
    }
    int checksum = calculateChecksum(message);
    message.checksum = checksum;
    if (esp_now_send(macAddr, (uint8_t *)&message, sizeof(message)) != ESP_OK) {
        Serial.println("Failed to send message");
        return false;
    }
    return true;
}       Caio Mascarin, 2 hours ago • feat: add espnow callback functions
```

```
bool isValidMessage(EspnowMessage message) {
    int receivedChecksum = message.checksum;
    message.checksum = 0;
    int calculatedChecksum = calculateChecksum(message);
    if (receivedChecksum != calculatedChecksum) {
        Serial.println("Invalid message!");
        return false;
    }
    return true;
}

bool waitResponse(int timeout) {
    espnowTimeoutMs = timeout + millis();
    while (millis() < espnowTimeoutMs) {
        // waiting for response
    }
    if (espnowTimeoutMs != 0) {
        return false;
    }
    return true;
}

void resetTimeout() {
    espnowTimeoutMs = 0;
}
```

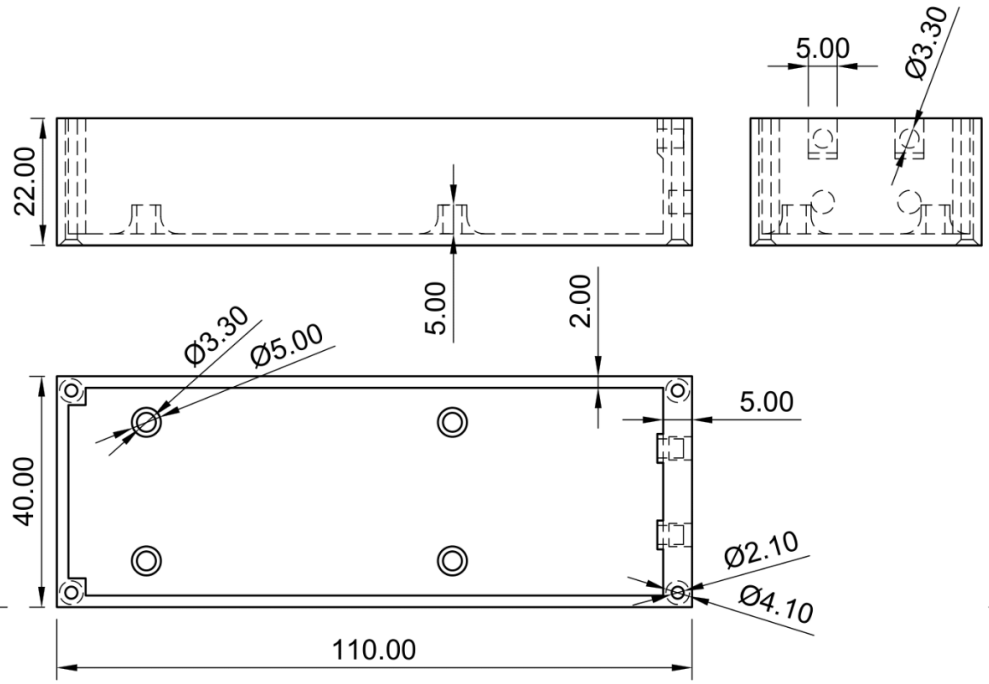
```
void receiveCallback(const uint8_t *macAddr, const uint8_t *incomingData, int dataLen) {
    EspnowMessage message;
    memcpy(&message, incomingData, sizeof(dataLen));
    if (!isValidMessage(message)) {
        return;
    }
    switch (message.acknoledge) {
        case RECEIVER_FREE:
            Serial.println("Receiver free");
            sensorState = SENDIND;
            resetTimeout();
            return;
        case RECEIVER_EOT:
            Serial.println("Receiver EOT");
            sensorState = IDLE;
            resetTimeout();
            return;
        case RECEIVER_BUSY:
            Serial.println("Receiver busy");
            return;
        default:
            return;
    }
}
```

```
void receiveCallback(const uint8_t *macAddr, const uint8_t *incomingData, int dataLen) {
    if (!shouldPair(macAddr)) {
        Serial.println("Can not pair to the sensor!");
        EspnowMessage response;
        response.acknoledge = RECEIVER_BUSY;
        sendMessage(macAddr, response);
        return;
    }
    EspnowMessage message;
    memcpy(&message, incomingData, sizeof(dataLen));
    if (!isValidMessage(message)) {
        return;      Caio Mascarin, 19 minutes ago • feat: add callback function
    }
    switch (message.acknoledge) {
        case SEARCHING_RECEIVER:
            Serial.println("Sensor is searching");
            receiverState = RECEIVING;
            return;
        case SENDIND_SAMPLE:
            Serial.println("Sensor is sendind");
            memcpy(&lastSample, &message, sizeof(EspnowMessage));
            receiverState = POSTING;
            resetTimeout();
            return;
        default:
            return;
    }
}
```

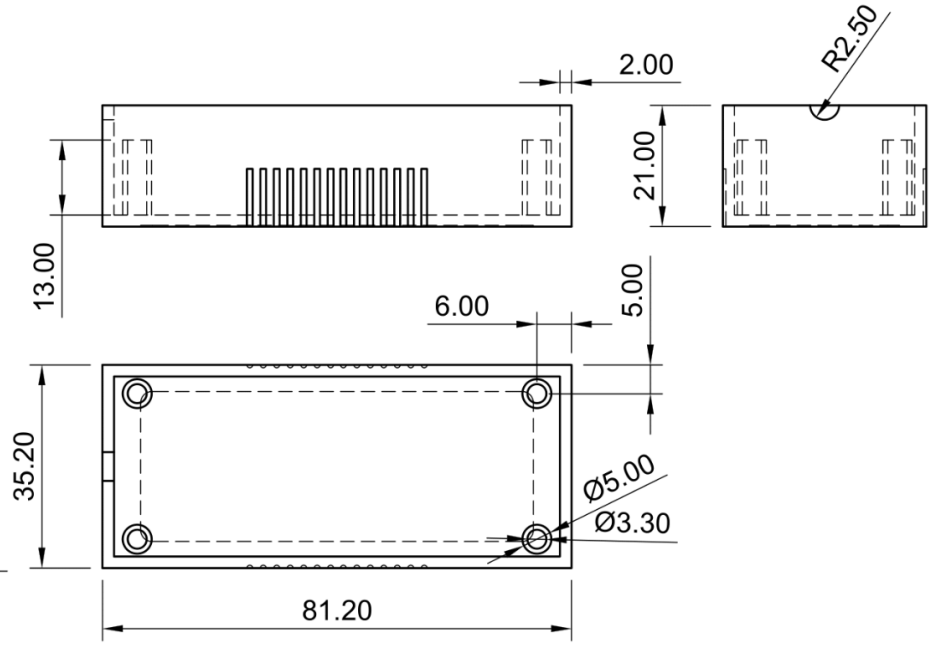
APÊNDICE B - DESENHO 2D DOS INVÓLUCROS



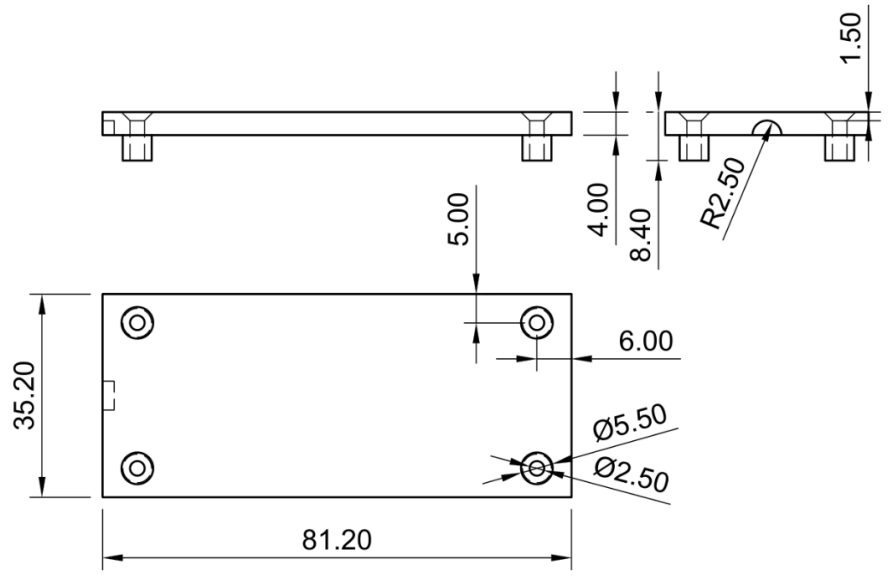
Dept.	Technical reference	Created by Caio Mascarin 28/11/2022	Approved by
		Document type	Document status
		Title: Case	DWG No.
			Rev. / Date of issue / Sheet: 1/1



Dept.	Technical reference	Created by Caio Mascarin 28/11/2022	Approved by	
		Document type	Document status	
		Title Tampa	DWG No.	
			Rev.	Date of issue / Sheet 1/1



Dept.	Technical reference	Created by Caio Mascarin 28/11/2022	Approved by	
		Document type	Document status	
		Title Case	DWG No.	
			Rev. / Date of issue	Sheet 1/1



Dept.	Technical reference	Created by Caio Mascarin 28/11/2022	Approved by	
		Document type	Document status	
		Title Case	DWG No.	
			Rev. / Date of issue	Sheet 1/1