

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE ENGENHARIA MECÂNICA
ENGENHARIA MECÂNICA**

BRUNO SILVA DE OLIVEIRA

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA DE
MONITORAMENTO REMOTO VIA IOT APLICADO A UM
COMPRESSOR PNEUMÁTICO**

TRABALHO DE CONCLUSÃO DE CURSO

**GUARAPUAVA
2022**

BRUNO SILVA DE OLIVEIRA

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA DE
MONITORAMENTO REMOTO VIA IOT APLICADO A UM
COMPRESSOR PNEUMÁTICO**

**Development and implementation of a remote IOT monitoring
system applied to a pneumatic compressor**

Trabalho de Conclusão de curso apresentado à
Coordenação de Engenharia Mecânica, como
requisito parcial à obtenção do Título de Bacharel
em Engenharia Mecânica, da Universidade
Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Arthur Hirata Bertachi

GUARAPUAVA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

BRUNO SILVA DE OLIVEIRA

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA DE
MONITORAMENTO REMOTO VIA IOT APLICADO A UM
COMPRESSOR PNEUMÁTICO**

Trabalho de conclusão de curso de graduação
apresentada como requisito para obtenção do título de
Bacharel em Engenharia Mecânica da Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 30/11/2022

Professor Dr. Arthur Hirata Bertachi

Membro 1

Universidade Tecnológica Federal do Paraná

Professor Dr. Hilson Henrique Daum

Membro 2

Universidade Tecnológica Federal do Paraná

Professor Dr. Aldo Przybysz

Membro 3

Universidade Tecnológica Federal do Paraná

GUARAPUAVA

2022

AGRADECIMENTOS

Agradeço a Deus por ter me permitido viver até aqui, não ter me deixado adoecer nem desistir, e sempre me levar de encontro a pessoas boas.

Aos meus pais, Nailor e Eronilda, por acreditarem em mim, por jamais medirem esforços para investir em meu conhecimento, por me proporcionarem tantas experiências e oportunidades, me permitindo evoluir tanto como estudante quanto pessoa.

A minha namorada Eloiza, que nunca me deixou duvidar das minhas próprias capacidades, teve paciência, me deu apoio emocional, e ficou ao meu lado nos momentos mais difíceis e de maior incerteza.

A minha sogra Valdineia, que sempre me apoiou, e incessantemente colocou meu nome em suas orações.

A todos os professores que fizeram parte da minha jornada acadêmica, em especial, professor Arthur, que desde o primeiro contato acreditou no desenvolvimento do presente trabalho, mesmo partindo do absoluto zero, orientou e norteou os estudos, para que o mesmo fosse possível.

Por último, e não menos importante, meus amigos e colegas de faculdade, que estiveram comigo em tantos momentos felizes.

“Se eu vi mais longe, foi por estar sobre os ombros de gigantes”

Isaac Newton

RESUMO

O presente trabalho apresenta o protótipo de um sistema de monitoramento contínuo de máquinas através de placas ESP32 e comunicação MQTT via IOT, com a finalidade de obter parâmetros da máquina em tempo real, focando sua utilização no âmbito da manutenção preditiva e preventiva. O conjunto de sensores, placas e equipamentos tem como objetivo principal aferir em tempo real parâmetros como: vibração mecânica, condições ambientais, temperatura de óleo e aquecimento do bloco de um compressor, por meio da interação entre máquina-ESP32-internet. As variáveis coletadas são apresentadas em tempo real, tanto em um computador como em um smartphone. A sistematização de monitoramento foi implementada em um sistema de ar comprimido, acionado por um motor elétrico de 2 cv. Os resultados mostraram a possibilidade, de que através de recursos de código aberto, é possível monitorar, alertar, e gerar histórico de funcionamento utilizando tecnologia IOT.

Palavras-chave: Indústria 4.0. Comunicação Wi-fi. Monitoramento contínuo. Manutenção preditiva. Manutenção preventiva. IOT.

ABSTRACT

This paper presents the prototype of a continuous machine monitoring system through ESP32 boards and MQTT communication via IOT, in order to obtain machine parameters in real time, focusing on its use in the scope of predictive and preventive maintenance. The main objective of the set of sensors, boards and equipment is to measure in real time parameters such as: mechanical vibration, environmental conditions, oil temperature and block heating of a compressor, through the interaction between machine-ESP32-internet. The collected variables are presented in real time, both on a computer and on a smartphone. The monitoring system was implemented in a compressed air system, driven by a 2 hp electric motor. The results showed the possibility, that through open-source resources, it is possible to monitor, alert, and generate operation history using IoT technology.

Keywords: Industry 4.0. Wi-Fi communication. Continuous monitoring. Predictive maintenance. Preventive maintenance. IOT.

LISTA DE FIGURAS

Figura 1 - Evolução das revoluções industriais.	15
Figura 2 - Evolução da manutenção industrial.	16
Figura 3 - Classificação de compressores.....	24
Figura 4 - Compressores alternativos de pistão.	24
Figura 5 - Pistão e diafragma.	25
Figura 6 - Aspectos construtivos de um transdutor de pressão piezoelétrico	27
Figura 7 - Sensor de temperatura DS18B20	28
Figura 8 - Sensor acelerômetro MPU6050	29
Figura 9 - O essencial da industrial IOT.	32
Figura 10 - Compressor Schulz Pro CSV10.....	34
Figura 11 - Características técnicas compressor Schulz linha PRO.	34
Figura 12 - Requisitos de Ambiente para instalação do compressor.	35
Figura 13 - Fluxograma de ferramentas.	38
Figura 14 - Distribuição de sensores.....	40
Figura 15 - Adaptação IDE parte 1.....	41
Figura 16 - Adaptação IDE parte 2.....	41
Figura 17 - Adaptação IDE parte 3.....	42
Figura 18 - Criação de Telegram BOT.	43
Figura 19 - Painéis Node-RED.....	44
Figura 20 - Arquitetura de nós Node-RED.	45
Figura 21 - Inicialização Node-RED	45
Figura 22 - WAMP server.....	47
Figura 23 - Circuito de montagem do sensor DS18b20 cápsula.	48
Figura 24 - Montagem de sensor e termômetro no cárter do compressor.	49
Figura 25 - Circuito de montagem sensor DS18b20 chip.....	49
Figura 26 - Circuito de montagem sensor DHT11.....	50
Figura 27 - Circuito de montagem sensor MPU6050.	51
Figura 28 - Instalação completa do sistema de monitoramento.	52
Figura 29 - Análise de temperaturas na compressão de 0 a 130 Psi.....	53
Figura 30 - Acoplamento do termômetro ao cárter do óleo	54
Figura 31 - Análise de temperatura do sensor DS18b20 vs Termômetro digital.	55
Figura 32 - Monitoramento em tempo real Node-RED dashboard, 5° partida.....	56

Figura 33 - Análise de partidas de 90 a 130 Psi.....	57
Figura 34 - Arquitetura de Nós para alerta de temperatura ambiente.	58
Figura 35 - Arquitetura de nó para temperatura do bloco elevada.	59
Figura 36 - Arquitetura de Nós para alerta de temperatura do óleo elevada.....	59
Figura 37 - Arquitetura de nós para alerta da alta umidade.	60
Figura 38 - Sistema de envio de alertas no Telegram.....	60
Figura 39 - Teste prático do sistema de alertas.	61
Figura 40 – Gráfico de variáveis do teste prático do sistema de alertas.	62
Figura 41 - Resultado do teste prático de alertas.....	62
Figura 42 - Armazenamento de dados no banco MySQL.	63
Figura 43 - Modelagem de relações em sistema star-schema.....	64
Figura 44 - Incidência de erro de valor no dashboard.	65
Figura 45 - Dado inconsistente no banco de dados.	65
Figura 46 - Dashboard normalizado com amplitude temporal total.	66
Figura 47 - Dashboard 8° partida com amplitude temporal de 5 minutos.....	67
Figura 48 - Configuração nó Mosca Broker.....	89
Figura 49 - Configuração nó MQTT.....	89
Figura 50 - Configuração nó Dashboard gauge	90
Figura 51 - Configuração nó Dashboard chart	90
Figura 52 - Configuração nó String	91
Figura 53 - Configuração nó Switch	91
Figura 54 - Configuração nó MySQL.....	92
Figura 55 - Configuração nó endereçamento do banco MySQL	92
Figura 56 - Configuração nó Telegram.....	92

LISTA DE TABELAS

Tabela 1 - Períodos de trocas de óleo lubrificante	36
Tabela 2 - Óleo librificante indicado para temperatura ambiente	36
Tabela 3 - Diagnóstico de falhas	37
Tabela 4 - Lista de materiais e orçamento	39
Tabela 5 - Características DS18b20	48
Tabela 6 - Características do sensorDHT11	50
Tabela 7 - Características MPU 6050	51
Tabela 8 - Teste de acionamentos consecutivos	57

LISTA DE ABREVIATURAS E SIGLAS

AI	<i>Artificial Intelligence</i>
CLP	<i>Controle Lógico Programável</i>
CPS	<i>Cyber Physical Systems</i>
IOT	<i>Internet of Things</i>
IIOT	<i>Industrial Internet of Things</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
API's	<i>Applications Programing Interfaces</i>

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	CONTEXTUALIZAÇÃO E CONSIDERAÇÕES INICIAIS	14
1.2	JUSTIFICATIVA.....	18
1.3	OBJETIVOS	19
1.3.1	OBJETIVO GERAL	19
1.3.2	OBJETIVO ESPECÍFICO.....	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	MANUTENÇÃO	20
2.1.1	Manutenção corretiva	20
2.1.2	Manutenção preventiva	21
2.1.3	Manutenção preditiva	22
2.2	COMPRESSORES	23
2.3	MANUTENÇÃO DE COMPRESSORES	25
2.4	SENSORES APLICÁVEIS A PLACAS EMBARCADAS	26
2.4.1	Sensor de pressão piezoelétrico.....	27
2.4.2	Sensores de temperatura.....	27
2.4.3	Sensor de vibrações acelerômetro.....	28
2.5	SISTEMAS EMBARCADOS	29
2.6	IIOT – INDUSTRIAL INTERNET OF THINGS.....	31
3	ASPECTOS METODOLÓGICOS.....	33
3.1	COMPRESSOR.....	33
3.1.1	Manutenção preventiva do compressor.....	35
3.2	MANUTENÇÃO CORRETIVA DO COMPRESSOR.....	36
4	MATERIAIS E MÉTODOS.....	38
4.1	PLANEJAMENTO DE PESQUISA	38
4.2	MATERIAIS	39
4.3	IDE ARDUINO	40
4.4	TELEGRAM.....	42
4.5	NODE RED.....	43
4.5.1	Plataforma Node-RED	43
4.5.2	Estrutura de nós	44
4.6	BANCO DE DADOS MYSQL	46
4.7	POWER BI.....	47

4.8	SENSOR DE TEMPERATURA DS18B20.....	47
4.9	SENSOR DE TEMPERATURA E UMIDADE DHT11	49
4.10	SENSOR ACELERÔMETRO MPU6050	50
5	RESULTADOS	52
5.1	ANÁLISE DE PARTIDA COM RESERVATÓRIO VAZIO	52
5.2	ANÁLISE DE PARTIDAS CONSTANTES.....	56
5.3	SISTEMA DE SEGURANÇA EM NODE-RED COM ALERTAS VIA TELEGRAM.....	58
5.4	RELATÓRIO EM POWER BI DE DASHBOARDS COM HISTÓRIO DE FUNCIONAMENTO.....	63
6	CONCLUSÃO	68
	REFERÊNCIAS.....	70
	APÊNDICE A – CÓDIGOS.....	74
	APÊNDICE B - CONFIGURAÇÕES	89

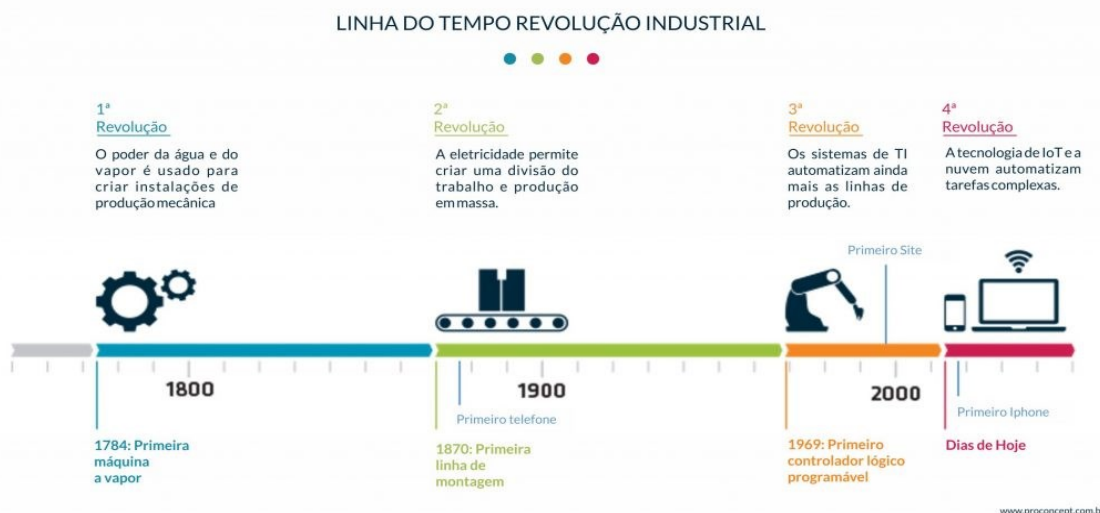
1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E CONSIDERAÇÕES INICIAIS

A indústria 1.0 marca o início da era industrial, a qual tem início por volta de 1800, onde é dado início a produção através de máquinas, atuando nas primeiras ferrovias, na indústria pesada e também na atuação do motor a vapor, que ocasionou a substituição de boa parte do trabalho braçal da época, ocasionando inquietação da população. Ao fim do século XVIII, o motor a vapor deixa de fazer parte apenas do meio de transporte passa a fazer parte da indústria, sendo pioneira do Reino Unido, fornecendo energia independentemente da função ou local de atuação. A indústria 2.0 tem início no final do século XIX, com chegada da energia elétrica, revolucionando as linhas de montagem, atuando na indústria automobilística, trazendo desta vez trabalhadores especializados, atuando em áreas específicas, e nasce a produção em série auxiliadas pelo início da aviação, a qual facilitava e agilizava o transporte de produtos manufaturados. Chega-se então à indústria 3.0 iniciando-se na década de 1970, a qual traz a digitalização dos processos através da eletrônica. Com a chegada da internet, dá-se acesso à informação de maneira global, gerando maior automação nas etapas de trabalho, onde ocorre ainda mais o processo de substituição do trabalho humano por máquinas atuando em série (WISSKIRCHEN et al., 2017).

A reformulação tecnológica que acontece atualmente, impulsionam avanços consideráveis nas próximas décadas, com uma proporção suficiente para ocasionar uma grande mudança de padrões da sociedade, uma vez que se desencadeiam mudanças ligadas a produção de novos produtos e novas prestações de serviços, atuando em diferentes formas organizacionais de trabalho, tanto do setor secundário quanto do setor primário. Transformações estas, que estão ligadas a uma nova era de automação, com desdobramento e aperfeiçoamento da conectividade e sistemas de comunicação, como em redes internas através por exemplo, da internet das coisas e o seu desenvolvimento no processo de sensoriamento, juntamente com o processamento da quantidade massiva de dados (*Big Data*) e uma maior capacidade robótica, onde os robôs deixam de fazer atividades repetitivas e passam a colaborar com capacidade cognitiva, intitulada como inteligência artificial, agregando ao crescimento da digitalização da economia (GIMENEZ, 2019).

Figura 1 - Evolução das revoluções industriais.



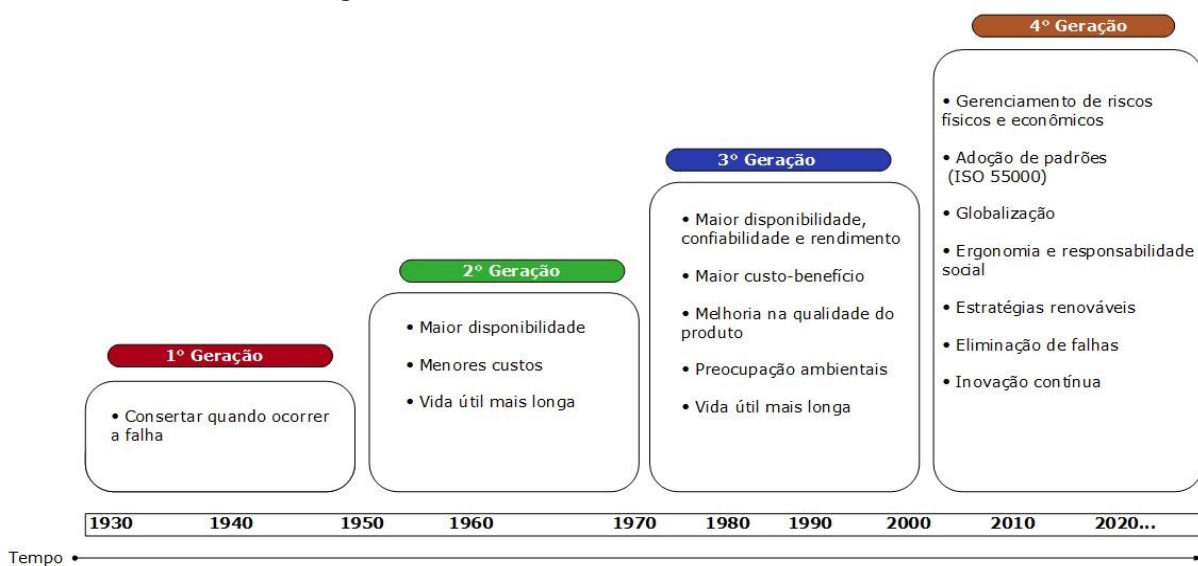
Fonte: (PROCONCEPT, 2019).

O termo manutenção industrial passou a ser comum por volta de 1930, dentro do âmbito militar, visando a manutenção das unidades de combate ligadas ao funcionamento e conservação deles. Até o final de 1950 o desenvolvimento da manutenção se restringia a manutenção de forma corretiva, ou seja, a substituição direta de elementos que apresentassem defeitos, esse período recebe o título de Fase 1 da evolução da manutenção. A partir de 1960 começam a surgir as primeiras linhas de produção, nasce então a preocupação de se evitar uma parada de máquina para não comprometer a produção e impactar diretamente nos custos do produto. Outro fator que colaborou para tal desenvolvimento foi a aviação, onde ocorria a impossibilidade de manutenção durante o voo. Busca-se então adiantar possíveis modos de falhas por meio de uma manutenção preventiva, esse período recebe o título de Fase 2 da evolução da manutenção. Em 1970 o desenvolvimento industrial atinge um patamar que induz a geração de mais de um turno de produção, logo a manutenção acaba sendo direcionada para o período noturno e de finais de semana, enfatizando a importância da plena disponibilidade dos equipamentos, busca-se então reduzir o tempo de manutenção, analisando-se padrões de eficiência das máquinas, classificando tal período como Fase 3 da evolução da manutenção (CARREIRA; SILVA; CANEIRA, 2010).

A fase 4 da manutenção industrial, revoluciona o funcionamento da indústria, onde a adição de elementos tecnológicos faz com que a fábrica transforme seu título de “fábrica convencional” para “fábrica inteligente”, principalmente pela aplicação das

ferramentas de CPS e IOT. Tal digitalização promove a combinação de troca de informações rápida, e faz com que os operadores de manutenção passem por processos de especialização, para que se adaptem ao ambiente digitalizado, e sejam capazes de interagir com robôs, bancos de dados e computadores, contribuindo assim para o processo de aplicação da indústria 4.0 na manufatura (GALLO; SANTOLAMAZZA, 2021).

Figura 2 - Evolução da manutenção industrial.



Fonte: Adaptado de (AMP, 2016).

Quanto aos principais tipos de manutenção, podemos citar a Manutenção Corretiva, a qual é dividida em duas, a Não-Planejada, que pode ocorrer na presença do modo de falha ou perda de eficiência, sem que haja tempo para a preparação dos serviços a serem executados, e a Manutenção Corretiva planejada, a qual é elaborada em torno de um acompanhamento preditivo, de forma detectiva ou até que se decida por prosseguir até a ocasião da falha. A Manutenção Preventiva, busca atuar reduzindo falhas ou perdas de desempenho, os quais são analisados em um período definido, com a finalidade de se ajustar a variação temporal para que não ocorra a troca em intervalos menores do que o necessário, ocasionando em uma manutenção desnecessária. A manutenção Preventiva trabalha de maneira sincronizada com a dificuldade de análise da possibilidade de falha, onde se podem diagnosticar erroneamente defeitos devido falha humana, ou provocar uma nova falha, devido montagem e desmontagem, contaminação de fluidos, falhas ocasionadas pela partida

e parada do sistema, dentre outras possibilidades. Na Manutenção Preditiva temos uma análise voltada para aspectos relacionados à mudança de parâmetros de condição ou desempenho, o qual é estabelecido através de uma sistemática. A Manutenção preditiva trabalha por meio do acompanhamento de diversos parâmetros sendo menos invasivo, evitando a parada do ciclo produtivo e elevando a longevidade de operação do equipamento o maior possível. Por fim, temos a Manutenção Detectiva, a qual atua em parâmetros imperceptíveis a equipe de manutenção, trazendo maior confiabilidade para o conjunto, onde cada vez mais se fazem presentes sistemas como computadores digitais e CLPs, para a coleta e controle de dados, atuando de maneira importante em plantas de processos contínuos (KARDEC; NASCIF, 2009).

Apesar de a manutenção preditiva não ser uma ferramenta recente, a sua aplicação é feita de maneira geral, e possui diferentes formas e graus de aplicações. Em um grau primário é feita a análise visual, através da experiência de um técnico, com determinada periodicidade. O próximo grau de manutenção preditiva consiste em se fazer o uso de instrumentos manuais de medição, e a análise passa a ser através da óptica dos parâmetros coletados, visando o momento mais apropriado para se reparar ou efetuar a troca de algum componente. Em um terceiro grau, temos o monitoramento contínuo em tempo real, o qual é feito através do sensoriamento, coletando dados e oscilações de parâmetros de maneira ininterrupta, enviando-os para um banco de dados, o qual é analisado pela equipe de engenharia, buscando planejar e encontrar o melhor momento para que a aplicação da manutenção seja executada, fazendo uso do status do equipamento, seu meio ambiente e seu histórico de manutenção (SAHBA et al., 2021).

O processo de industrialização ocorrido no século XIX, teve como característica principal a substituição do trabalho manual pelo uso na energia mecânica, através de ferramentas movidas pelo uso da energia elétrica, vapor e ar comprimido. Dentre os meios energéticos citados, o ar comprimido teve destaque, principalmente pelo fato de que as ferramentas pneumáticas apresentavam sistema mecânico simples, poucas peças possuíam mobilidade, eram confiáveis, robustas, de simples manutenção, com boa aplicabilidade e peso reduzido. Características estas, que ainda são levadas em conta nos dias de hoje, apesar de a produção de ar comprimido apresentar uma eficiência energética relativamente baixa, quando comparada a outros processos. A manutenção de sistemas de compressão de ar, deve ser planejada com a finalidade

de elevar os níveis de eficiência energética do sistema, e elevar a durabilidade do equipamento e de seus equipamentos auxiliares. O tipo de manutenção aplicada, está relacionado a parâmetros como o modelo do compressor, equipamentos auxiliares, grau de utilização, e qualidade do ar utilizado. A utilização de ar com boa qualidade, resulta em uma menor manutenção, tornando o sistema mais confiável e minimizando os desgastes da máquina, necessitando de menor tratamento de impurezas (NEHLER, 2018).

1.2 JUSTIFICATIVA

Segundo (COPCO, 2010), a manutenção do compressor, quando bem planejada, possibilita a antecipação de custos e eleva a vida da máquina e de seus equipamentos auxiliares. A utilização de eletrônica avançada, permite um diagnóstico do equipamento, gerando utilização dos componentes de maneira otimizada, onde a substituição ocorre apenas quando existe a necessidade. A possibilidade de recondicionamento de peças, acontece de maneira prematura, antes que ocorra um dano significativo, evitando assim uma sequência de danos, e tempo de máquina parada desnecessário. A manutenção do compressor na indústria, deve ser acompanhada através de junção de equipamentos eletrônicos e equipe de manutenção qualificada, por meio de inspeção diária, física e virtual, uma vez que olhos e ouvidos, podem diagnosticar falhas que sistemas eletrônicos não podem.

Como descrito anteriormente, a manutenção de sistemas de compressão de ar, é uma área presente na indústria, apesar de nem sempre receber a devida. Com a evolução da robótica, vemos cada dia mais a presença de máquinas e robôs dentro da indústria. Logo, com uma maior quantidade de equipamentos mecânicos e robóticos dentro indústria, cresce também o valor agregado no chão de fábrica, tornando ainda mais importante o controle e planejamento da manutenção para o bom funcionamento e sucesso da produção.

O presente projeto busca, auxiliar a manutenção proativa através do sensoriamento e monitoramento remoto em tempo real, fazendo uso da tecnologia da placa de prototipagem Arduino e da internet das coisas. Reduzindo custos de cabeamento, deslocamento, diagnóstico, prognóstico de elementos mecânicos, buscando planejar e organizar a manutenção assertiva, através de relatórios de

parâmetros da máquina, procurando por desvios no padrão de comportamento do compressor baseado nas informações coletadas.

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

A proposta deste trabalho é o desenvolvimento do protótipo de um sistema de aquisição de dados utilizando código aberto, aplicado diretamente a um compressor pneumático, para a aquisição das principais variáveis, através do monitoramento contínuo e remoto de um compressor de ar de duplo pistão, visando facilitar a atuação da manutenção proativa.

1.3.2 OBJETIVO ESPECÍFICO

A seguir, são apresentados os objetivos deste trabalho:

- Utilizar um sensor de temperatura para coletar valores e variações de temperatura do conjunto compressivo de dois estágios, de maneira numérica e gráfica;
- Utilizar um sensor de vibração para coletar valores e variações de vibrações do conjunto compressor das direções X, Y e Z, de maneira gráfica;
- Utilizar um sensor de temperatura de fluídos, para coletar valores e variações de temperatura do óleo do conjunto compressivo de dois estágios, de maneira numérica e gráfica;
- Configurar placas ESP32 para ler os valores coletados pelos sensores, e projetá-los em um computador/celular através da internet, de maneira contínua e remota, numérica e gráfica;
- Instalar o conjunto em um compressor real, além de apresentar e avaliar o seu desempenho e funcionamento prático; e
- Disponibilizar publicamente os dados coletados para possibilitar sua utilização por outros pesquisadores.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 MANUTENÇÃO

A manutenção é definida de diferentes formas, por diferentes instituições. Porém, todas buscam enfatizar a preocupação com a integridade e funcionamento pleno de máquinas e equipamentos, através de medidas e procedimentos realizados em ferramentas, peças, equipamentos, entre outros (ALMEIDA; PAULO; SAMUEL, 2016).

2.1.1 Manutenção corretiva

Segundo a Associação de Normas Técnicas Brasileira (ABNT) NBR 5462/1994, a manutenção corretiva pode ser estabelecida como “Manutenção efetuada após a ocorrência de uma pane destinada a recolocar um item em condições de executar uma função requerida”.

Ao se buscar reparar um equipamento que apresenta algum defeito ou desempenho não esperado, estamos fazendo uma manutenção corretiva, logo a manutenção corretiva nem sempre pode ser considerada com uma manutenção de emergência. A manutenção corretiva pode ocorrer através de duas hipóteses, sendo elas a baixa de desempenho ou a ocorrência da falha, podendo ser de maneira não planejada ou planejada. Na Manutenção Não Planejada, ou também chamada de Manutenção Não Programada, é feita a ação da manutenção a partir de um fato já ocorrido, o qual não há tempo hábil para a sua preparação, e traz como característica o alto custo financeiro, uma vez que a quebra pode acarretar em; uma parada de produção, decaimento da qualidade, comprometimento de outros componentes, fatores estes, que podem ser potencializados quando se trata de um sistema de processamento contínuo, além do alto valor indireto dos custos de manutenção. Na manutenção corretiva planejada, a característica principal é a qualidade das informações coletadas, a qual facilita a manutenção, reduzindo os custos, elevando a qualidade, além de tornar a manutenção mais ágil e segura, podendo-se controlar os fatores que atingem diretamente a produção (KARDEC; NASCIF, 2009).

Apesar de muitos estudos apontarem a manutenção corretiva como uma ação indesejada pela indústria, devido às análises do seu custo-benefício, efetuar a

aplicação da manutenção corretiva de forma planejada, ainda é de grande importância, uma vez que garante a continuidade da produção, por meio do cuidado com determinados aspectos, como custo de ação e tempo de inatividade da máquina. Para que se possa ter uma manutenção corretiva econômica, dois fatores de custos devem ser levados em conta, sendo o de custo de inatividade e custo de atividade referente a ação de reposição. Métodos de monitoramento também podem ser de grande valia para manutenção corretiva planejada, principalmente quando a sua intervenção é inevitável, e a ação tem um elevado tempo de conclusão, gerando um tempo considerável de inatividade. Neste âmbito, determinar os momentos de intervenções através do monitoramento de componentes, são imprescindíveis para se reduzir custos da manutenção corretiva (ÖZGÜR-ÜNLÜAKIN; TÜRKALI; AKSEZER, 2021).

2.1.2 Manutenção preventiva

A manutenção preventiva busca reduzir ou evitar a falha e a queda de desempenhos nos equipamentos, através de um planejamento preliminar, por meio de espaços temporais definidos. Em determinadas aplicações, como a da aviação, a manutenção preventiva se torna indispensável por estar relacionada diretamente com o fator segurança. No âmbito industrial, os dados dos equipamentos são peça chave para o planejamento da manutenção, juntamente com as condições ambientais e operacionais, as quais têm influência direta sobre a conservação dos equipamentos. O planejamento busca equilibrar as ações, para que não ocorra uma falha imprevisível em determinado período, e para que também não ocorra a troca prematura de peças e equipamentos. Quanto maior for a disponibilidade e facilidade do elemento mecânico a ser reparado, mais conveniente será a ação de manutenção, porém o ato da manutenção pode gerar outros fatores indesejáveis como por exemplo; falhas durante a partida e a parada, falha humana, falhas sobressalentes, e contaminações oriundas do contato com o sistema de lubrificação (KARDEC; NASCIF, 2009).

Através de estudos realizados pela marinha americana, constatou-se que grande parte das falhas de equipamentos, derivam de acontecimentos de falha aleatórios, e não relacionados à idade do maquinário. Demonstrando que assumir valores prevendo a falha esperada na manutenção, nem sempre é opção viável, uma vez que o tempo não pode atuar como um fator determinante na tomada de decisões.

Realizar aferições de maquinários regularmente, além de demandar tempo, e investimento, tornam os dados obtidos subjetivos, dependendo do tipo de informação que se busca coletar, se exige um operador com experiência e conhecimento para realizar medições assertivas. O monitoramento contínuo pode enxugar custos preventivos, como por exemplo, a troca filtros de pressão do ar utilizados na indústria farmacêutica, que são trocados preventivamente por tempo de uso, poderiam ser direcionados para a troca quando as leituras de ambos os lados do filtro não condissessem. Como no exemplo do filtro, diversos outros elementos trocados preventivamente, podem ser substituídos assim que apresentem indício de perda de função, trocados ainda em uma manutenção declarada preventiva, porém a partir de parâmetros indicativos, semelhantes aos utilizados na manutenção preditiva (IBM, 2021).

2.1.3 Manutenção preditiva

A manutenção preditiva, acontece através do monitoramento das condições reais do elemento ou sistema mecânico, e intervenção é efetuada através da manutenção corretiva planejada. A manutenção preditiva tem por objetivo prevenir falhas através da análise de parâmetros, prolongando o funcionamento do equipamento o máximo possível, efetuando aferimentos de condições com a máquina em plena produção. No momento em que o grau de degradação do componente se aproxima do limite, é tomada a decisão da intervenção para a correção, assim como na manutenção planejada, possibilitando com que haja a elaboração e organização de um plano de manutenção. Para o uso da manutenção preditiva, alguns parâmetros são necessários. O equipamento deve possibilitar alguma maneira de medição para se efetuar o monitoramento, o valor agregado do componente deve ser levado em consideração devido aos custos agregados processo, as falhas a serem evitadas, devem ser de causas que possibilitem monitoramento, seja implementado um sistema de análise e diagnóstico sistematizado (KARDEC; NASCIF, 2009).

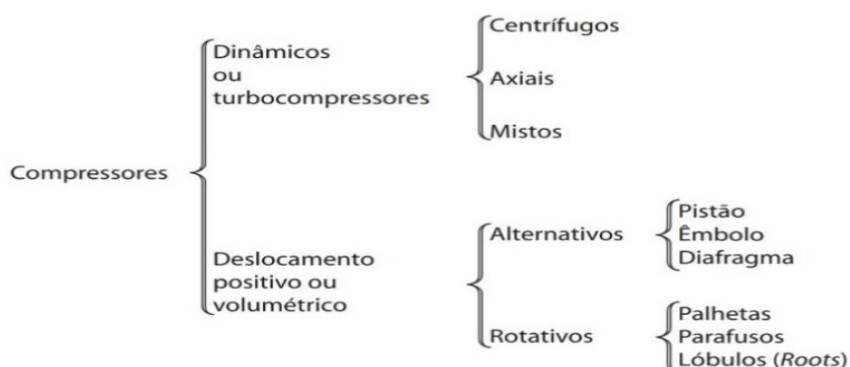
Dentro do conceito da indústria 4.0, a manutenção preditiva passa a acontecer por caminhos mais curtos do que os usuais. Tarefas que antes demandava da comunicação verbal entre o operador e o setor de gestão de manutenção, para que as informações fossem repassadas para o planejamento e controle da manutenção, dá lugar a interação direta entre máquina e o setor de responsável pelos reparos,

através da compilação de dados entre softwares, encurtando assim a comunicação dos registros de anomalias detectadas. Tal mudança impacta a setor de gestão da manutenção de maneira geral, facilitando criação e cancelamento de ordens de serviço, possibilitando a localização imediata da necessidade de manutenção dentro do espaço da indústria, e melhor administrando o estoque do almoxarifado de manutenção, geração de cadastros e detalhamento do histórico de máquinas. Outra mudança ocorre na função direta do reparador, onde ele deixa de colaborar apenas com a ação de reparo, e passa a propor mudanças e melhorias, participando de maneira ativa na mudança dos controles do processo produtivo (SACOMANO, 2018).

2.2 COMPRESSORES

A pneumática busca a geração de energia mecânica através da utilização de um compressor de ar para a obtenção do ar comprimido em determinados parâmetros de pressão e volume, a partir da eletricidade, gasolina, vapor, ou outra forma de produção de força motriz. Possibilitando assim, promover ações em sistemas pneumáticos, com uma variedade de utilizações, como por exemplo; limpeza, acionamento de atuadores, refrigeração, pintura, fluidização e componentes auxiliares de motores veiculares, podendo abranger uma grande gama do mercado, como; de refrigeração, ferramental pneumático, industrial, petroquímico, hospitalar e odontológico. Quanto à classificação de compressores, podemos dividi-los em dois princípios de funcionamento, que são os dinâmicos e os volumétricos (CREMASCO, MARCO, 2018).

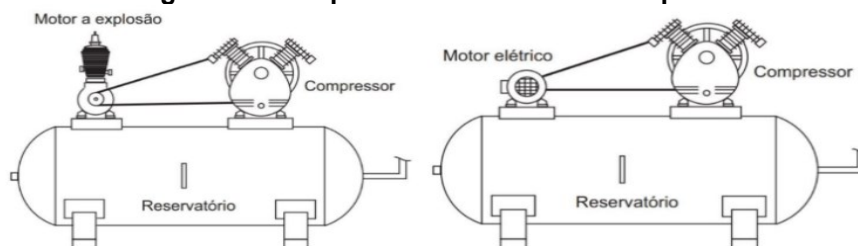
Figura 3 - Classificação de compressores.



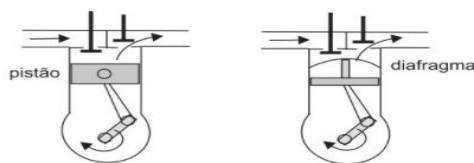
Fonte: (CREMASCO, MARCO, 2018).

Os compressores, como ilustrado na *Figura 3*, são divididos em dois principais grupos, dinâmicos e de deslocamento positivo. A classe dos compressores, está contida dentro das máquinas operatrizes de fluxo, podendo atuar em uma larga faixa de operação, atendendo diversas necessidades. A escolha de um modelo de compressor, começa pela definição do fluxo de ar e da pressão de trabalho exigida pela máquina, ferramenta ou conjunto de um sistema mecânico. Neste trabalho será abordada especificamente a família dos compressores de deslocamento positivo, o qual abrange os modelos de pistão, êmbolo e diafragma. O foco será no modelo alternativo de pistão, devido a sua fácil disponibilidade, além de estar presente em diversas aplicações dentro pequenas empresas de áreas variadas, modelo o qual, será utilizado para o desenvolvimento do protótipo, utilizando o monitoramento contínuo e em tempo real de parâmetros para auxílio da manutenção preditiva, porém os princípios discutidos aqui, podem em grande parte, ser aplicados tanto ao vasto modelo de compressores, quanto a outras máquinas mecânicas, térmicas e de fluxo.

Figura 4 - Compressores alternativos de pistão.



Fonte: (FILHO, 2014).

Figura 5 - Pistão e diafragma.

Fonte: Adaptado de (FIALHO, 2015).

O funcionamento desta classe de compressores, é baseada através no movimento alternativo, que pode ser gerado tanto por motores a explosão quanto elétricos, definidos através das limitações do meio no qual o compressor executará o seu trabalho. O seu funcionamento ocorre a partir de dois princípios básicos, de admissão e descarga, onde a admissão é proporcionada pelo deslocamento de meia volta do eixo com a abertura da válvula de admissão e fechamento da válvula de descarga. A restante da volta do eixo, faz com as válvulas de admissão se fechem e a de exaustão se abra, executando assim a compressão do ar, o qual é direcionado para o armazenamento no cilindro de gás. Existem formas, quantidades, e posicionamentos variados quanto disposição dos pistões e também quanto a sua arquitetura de funcionamento tanto da admissão e da descarga, podendo ser diretamente em contato com pistão, ou através da utilização de um sistema de diafragma, o qual busca criar uma barreira de isolamento entre os componentes mecânicos e o ar comprimido, evitando assim a possibilidade de que partículas do sistema de lubrificação possam estar contaminando o ar pressurizado. Tal sistema de isolamento é comumente utilizado em indústrias que possuam maior exigência de qualidade do ar comprimido, como utilização em áreas de produção farmacêutica e uso hospitalar (FIALHO, 2015).

2.3 MANUTENÇÃO DE COMPRESSORES

No livro publicado por (BLOCH; GEITNER, 2012), a manutenção de compressores, é vista como uma manutenção que pode ser dividida em 4 níveis, sendo elas; Manutenção reativa, onde se espera que a falha do compressor aconteça, e assim se busca reparar a falha. Porém tal método, acarreta a falta de planejamento, traz insegurança ao processo, e deve ser evitado. O segundo nível seria a Manutenção preventiva seletiva, onde se inclui cuidados com a lubrificação e válvula de admissão do sistema do compressivo, e reparos de maneira proativa. No terceiro

nível, adiciona-se o requisito da calibração junto às medidas do nível dois, em tal nível, já se notam reduções nas intervenções de medidas corretivas. O quarto e último nível, é a Manutenção preditiva com reparo proativo, o qual é feito através do monitoramento das operações, diagnosticando-se possíveis tendências de se ocorrer alguma forma de falha. Os reparos preditivos são prolongados de tal modo, que se caso não ocorra a manutenção, a falha irá ocorrer. Os níveis de manutenção abrangem tanto compressores de deslocamento dinâmico quanto positivo. As informações diretas de intervenções só podem ser mensuradas a partir da disposição dos valores do tempo de atividade de cada equipamento, sem esse parâmetro não é possível estabelecer qual nível de manutenção deve ser empregado ao compressor. Os níveis de monitoramento podem variar, sendo de maneira manual pontual ou contínuo, a segunda forma é encontrada principalmente em grandes sistemas compressivos, sendo o tipo mais econômico de manutenção, uma vez que o componente do sistema compressivo trabalha até que se apresente resquícios indicando uma possível falha.

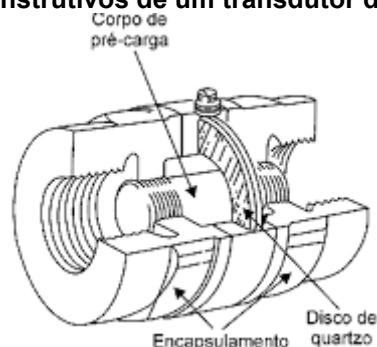
2.4 SENSORES APLICÁVEIS A PLACAS EMBARCADAS

O sensoriamento, se inicia desde a criação dos primeiros seres vivos, onde nativamente, organismos vivos já apresentavam sistemas de captação de sinais como; bioelétricos, biogenéticos, bióticos, biomecânicos, dentre outros, também exemplificados pelos sentidos humanos como; visão, olfato, audição e os demais. Dentro da sociedade o sensoriamento nasce a partir da criação das unidades de medidas, onde surge a necessidade da instrumentação e desenvolvimento de mecanismos para se obter medidas como por exemplo, a balança, que é um instrumento precursor dentro do desenvolvimento das civilizações. No âmbito industrial, a demanda de medição chega à tecnologia dos sensores energizados, buscando reconhecer e analisar variáveis físicas por meio de medições processadas através da eletricidade. Junto dos sensores, também possuímos os transdutores, os quais transformam um sinal em uma unidade na forma física, em outro tipo de sinal também na forma física. Tantos sensores quanto transdutores, trazem por característica a mesma finalidade, estar apresentando informações relacionadas a medidas de maneira geral (CAPELLI, 2013).

2.4.1 Sensor de pressão piezoelétrico

Os sensores de pressão piezoelétricos, funcionam através da utilização das características físicas do cristal, principalmente de quartzo e turmalina, que quando deformados de maneira elástica, alteram seu potencial elétrico, gerando uma variação de tensão, convertendo assim a aplicação de força em variação da eletricidade, onde as propriedades dos cristais garantem a resposta em um pequeno intervalo de tempo. A sua utilização é feita através de uma grande variedade de modelos e de tamanhos, comumente utilizados em áreas como a hidráulica e a pneumática. A calibração de transdutores piezoelétricos, é efetuada acima do seu campo de frequência especificado, executado de maneira estática, para que se atinja uma considerável precisão de medição (TOMAZINI; DANIEL; PEDRO, 2020).

Figura 6 - Aspectos construtivos de um transdutor de pressão piezoelétrico



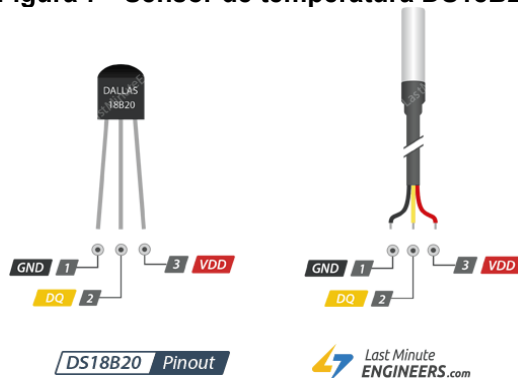
Fonte: (TOMAZINI; DANIEL; PEDRO, 2020).

2.4.2 Sensores de temperatura

A temperatura é uma unidade de medida física, a qual tem por finalidade média o grau de agitação das moléculas cinética das partículas atômicas de um sistema. A aferição de temperatura é costumeiramente realizada por meio de sensores que fazem uso do termo resistência como princípio de funcionamento, o qual utiliza a variação sincronizada do aumento da temperatura com o aumento da resistência elétrica. Dentre as possibilidades de sensores, o modelo DS18B20, se destaca dos demais sensores, por se tratar de um sensor digital, descartando-se a necessidade da utilização de conversão de sinais analógicos para digitais. Outro fator que colabora para sua utilização, está na sua precisão e faixa de atuação, indicado para ser utilizado em sistemas de medições de temperatura precisa e repetitiva.

(MARTINAZZO; ORLANDO; DA URI, 2016). O sensor mencionado, traz versões as quais permite realizar medidas de temperaturas de objetos sólidos, e outra versão que possibilita coletar medidas de temperatura de fluidos, como mostrado na Figura 7, ambas de maneira consistente. Fatores estes, que são objetivos a serem alcançados no presente trabalho.

Figura 7 - Sensor de temperatura DS18B20

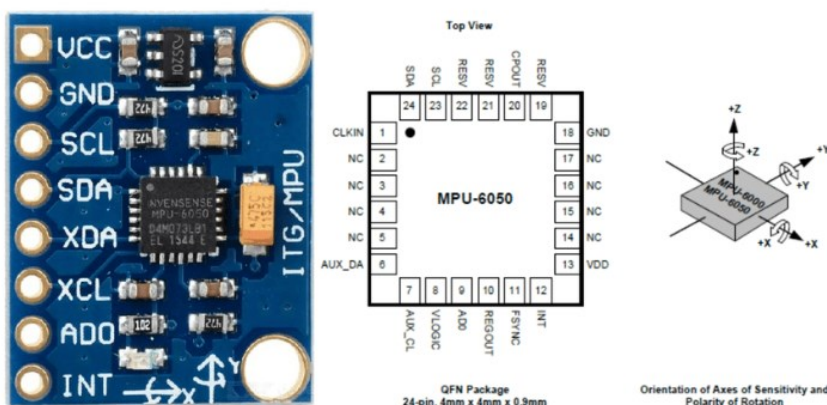


Fonte: (“Interfacing DS18B20 1-Wire Digital Temperature Sensor with Arduino”, 2018)

2.4.3 Sensor de vibrações acelerômetro

O sensor MPU6050 trata-se de um acelerômetro que efetua medidas de vibrações em 3 eixos distintos, se os eixos X, Y e Z. O sensor traz por características uma alta sensibilidade a vibrações, somadas a um baixo consumo de energia, qualidade a qual se torna primordial em projetos que fazem o uso de baterias, possibilitando também a utilização da fonte de tensão da placa embarcada. Outra vantagem do sensor é a sua faixa de alimentação, a qual permite a alimentação por 5V ou 3,3V, uma vez que o sensor demanda de baixas tensões para seu funcionamento. De maneira eletrônica, o sensor apresenta seis pinos para conexões, os quais são distribuídos sendo um deles para autoteste, três para as informações de cada eixo cartesiano, um para aterramento (GND), e por fim o último pino de alimentação (VCC).

Figura 8 - Sensor acelerômetro MPU6050



Fonte: (MITRA; GANIGA, 2019).

2.5 SISTEMAS EMBARCADOS

O sistema embarcado ESP32 pode ser descrito como um microcontrolador, de placa única com plataforma eletrônica e de código aberto, o qual pode ser usado por uma gama geral de pessoas, sem exigir capacidade elevada e específica em áreas de conhecimento como eletrônica, engenharia ou programação. As placas ESP32, normalmente possuem design compacto, e trazem a característica de fácil programação, podendo ser utilizados nos mais variados tipos de projetos, como robótica, inovação de dispositivos, sensoriamento, entre outros. A placa eletrônica, basicamente, tem seu funcionamento baseado na comunicação através da variação do pulso elétrico durante o percurso, com percurso denotado de um começo, um meio e um fim.

A programação da ESP32 é dividida em duas partes, sendo ela “setup()”, a qual será responsável por informar ao Arduino o que é preciso ele saber, e como deve funcionar, quais serão os pinos configurados como entradas e saídas, e como deve ser seu funcionamento. A outra parte, é a “loop ()”, que tem como objetivo, demonstrar ao ESP32 o que deve ser feito com os dados do “setup ()”, através da execução do código no formato de loop, que nada mais é do que a repetição do código de maneira contínua, ou até que alguma condição específica seja satisfeita (GERTZ; JUSTO, 2012).

Apesar da evolução da tecnologia, e o melhoramento contínuo de placas embarcadas, algumas limitações ainda são impostas ao seu uso, segundo (KONDAVEETI et al., 2021), suas principais limitações são; potência de processamento, espaço de armazenamento e memória com dimensão na casa de

kilobytes, trabalho esforçado ocasionado pela baixa capacidade de processamento e armazenamento, limitação de complexidade e de hardware. Apesar das limitações, o Arduino permite a interações com outros softwares comuns no mercado, característica que acaba facilitando e deixando interativo possíveis análises. Dentre os softwares que permitem interação, podemos citar com exemplo; Excel, LABVIEW e MATLAB, disponibilizando dados em tempo real, contribuindo e facilitando possíveis análises através da interação entre sistemas embarcados e ferramenta computacional.

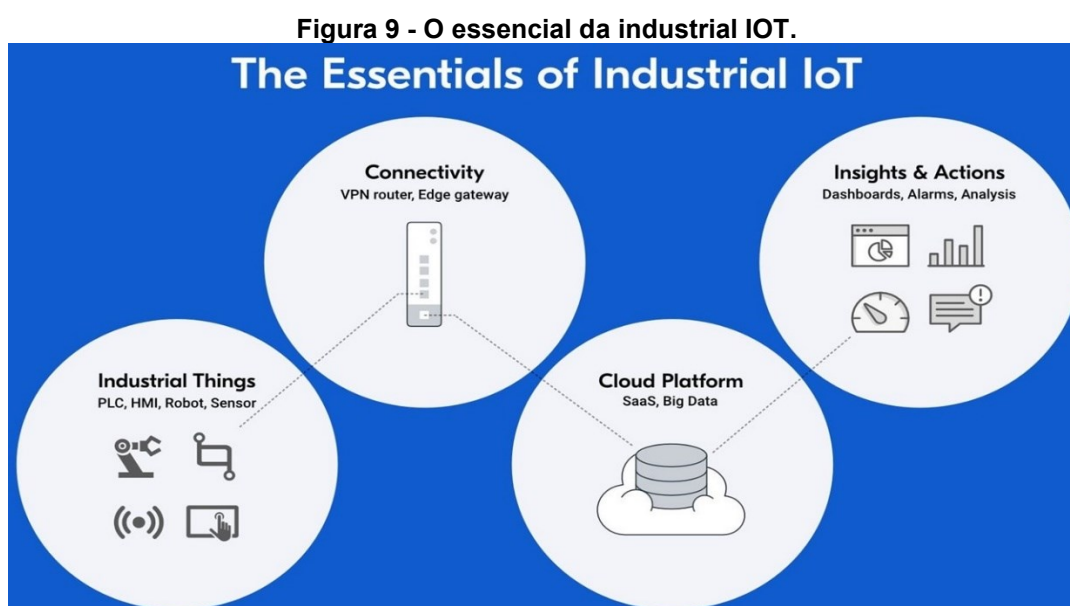
2.6 IIOT – INDUSTRIAL INTERNET OF THINGS

O termo IIOT (*Industrial Internet of Things*), o qual traduzido, significa internet das coisas industrial, faz referência a uma nova tecnologia que atua na indústria de maneira inteligente, a qual faz a comunicação através da internet de peças, ferramentas, equipamentos e também atua em setores de logísticas e controle de estoques, através de sensores e dispositivos. A tecnologia IIOT permite que dentro da indústria, quase tudo possa ser digitalizado, reduzindo o risco de erros humanos e agilizando processos, conseguindo observar com olhar mais detalhado imperfeições presentes na planta industrial e no processo produtivo, efetuando a tomada de decisões baseada através dos dados obtidos, auxiliando a indústria como um todo na resolução de problemas. Outro fator com que a IIOT colabora, é para o da manutenção inteligente, auxiliando a equipe de manutenção através do acesso rápido de informações e instruções, onde os dados são encaminhados para a nuvem, analisados através do uso de Softwares, e assim indicando uma possível ação, como a emissão de alertas sem a necessidade de comunicação física de alguém. Não só as máquinas podem ser cuidadas pelo sistema conectado, mas também os trabalhadores, através do monitoramento de atividades que possam estar colocando em risco a saúde e a integridade física de pessoas no chão de fábrica, e também da saúde geral dos trabalhadores, utilizando tecnologia de conexão através de roupas especiais e sensores inteligentes conectados junto ao corpo, podendo indicar um possível posicionamento perigoso dentro do espaço da manufatura produtiva (JAVAID et al., 2021).

O sistema IIOT veio para mudar o conceito e a forma com que a fabricação de produtos é feita, juntamente com a inteligência artificial, proporciona um enorme acesso de informações em nuvem, e uma rápida adaptação a mudança de novas condições de operação, servindo como uma das bases para a quarta revolução industrial, desenvolvendo a manufatura inteligente juntamente com tecnologias aliadas como; rede 5G, manufatura aditiva e tecnologia de fabricação avançada. Para que as tecnologias possam se comunicar, e que possam ser utilizadas de maneira segura pela indústria, alguns pontos devem ser levados em conta, como; segurança de conectividade dos equipamentos de maneira privativa, confiabilidade, e qualidade de troca de informações, campos estes, que ainda estão em desenvolvimento, apesar de já apresentarem algumas normas referentes a segurança, como por exemplo a

norma IEC 624443, desenvolvido pela Comissão Eletrônica Internacional, voltada para a segurança de automação e controle industrial (IEC) (LU; WITHERELL; JONES, 2020) .

A tecnologia IOT, causa mudanças não só na relação entre homem e máquina, mas também nas relações entre máquina e máquina. Quando comparados a um operador, a utilização do IOT se destaca em aspectos de velocidade, precisão e resistência, atuando através de elementos mecânicos, pneumáticos, hidráulicos, elétricos e computacionais, por meio de CLPs e da utilização de aquisição de dados SCADA trabalhando de maneira integrada, reduzindo gargalos de desperdício e aumentando a produção. Dentro do chão de fábrica, se torna comum a utilização de uma vasta quantidade de modelos de sensores com interação por internet das coisas. O sensor é utilizado com uma entrada de dados para um aplicativo IIOT, a localização dos aplicativos quanto o posicionamento dos sensores, são feitos de maneira estratégica, buscando fornecer a cabine de comando, informações atualizadas, sendo através de um computador, ou do próprio dispositivo móvel do supervisor de produção, ajudando a gerir o controle dos processos, permitindo assim, que tais informações possam estar chegando ao supervisor independentemente da localização do mesmo dentro do ambiente industrial, desde que haja conexão com a internet ou outro tipo de conexão (ELANGO VAN, 2020).



Fonte: (WEIJIS, 2021).

3 ASPECTOS METODOLÓGICOS

O presente trabalho consiste na aplicação de um sistema de monitoramento contínuo de máquinas, aplicado em um compressor modelo de entrada da marca Schulz, representado na Figura 12. O sistema desenvolvido para a aquisição dos dados composto por 4 placas embarcadas com 4 sensores, os quais ao todo coletam 7 parâmetros de comportamento da máquina. O método de comunicação predominante do fluxo de informações é o MQTT, o qual possibilita utilização de conceitos e aplicações de IOT dentro do estudo do protótipo. Com a finalidade de auxiliar com informações detalhadas a tomada de decisões dentro do âmbito da engenharia de manutenção.

3.1 COMPRESSOR

O compressor Schulz pro CSV10 utilizado para objeto de estudo é do modelo CSV 10/100, o qual trás características técnicas com deslocamento teórico de 10 pés³ por minuto, pressão máxima na casa de 140 lbf/pol², reservatório de armazenamento de ar comprimido com 100 litros de capacidade, tempo de enchimento de aproximadamente 5 minutos e 55 segundos. Possui sistema de compressão com duplo pistão, o qual utiliza correia 1-3VX, tocados por um motor elétrico monofásico de 2 HP. Sua saída de ar possui diâmetro compatível com mangueiras de ¼", seu reservatório de óleo é de 350 ml, totalizando ao todo um conjunto com massa de 54,5 Kg. Seu projeto e fabricação levou a utilização de normas como; EN 1012-1, ABNT NBR NM 60335-1:2010 e NR12 (WERLANG, [s.d.]).

Figura 10 - Compressor Schulz Pro CSV10.



Fonte: (WERLANG, [s.d.]);

Figura 11 - Características técnicas compressor Schulz linha PRO.

CARACTERÍSTICAS TÉCNICAS														SCHULZ												
CARACTERÍSTICAS TÉCNICAS TECHNICAL FEATURES														PRO												
Modelo Model	Desplazamiento Teórico Displacement Theoretical		Presión Máxima Max. Pressure		Reservorio Depósito (Tanque) Air tank		Q _T gpm	P Polea Pulley Inch				Motor Eléctrico Mótor Eléctrico Electric Motor	Dimensiones Dimensiones Inch			Conexión Descarga Connection Air discharge connection	Bote Lubr. Aceite Lubr. Oil Cap	Peso el motor Weight with motor	Pintura Painting	Dimensiones Reservorio Dimensiones Dimensiones (mm)						
	psic/min. psi/min.	cm ³	bar	psi	litros geom. Volume geom. Geom. Volume (l)	Temp. Echim. Temp. Líquido Filling up Time		50 Hz	60 Hz	Correa Aix	hp		KW	Torpedo (V)	A					C	L	D	E	F	G	
CSV 10 AD	10	283	140	9,7	-	-	-	-	-	97	-	3 Vx.335	2	1,5	127/220 M/N	310	435	320	1/4"	350	27,1	Banco (Preto/Brilho) Tanque/Base (Preto em pó)	-	-	-	-
CSV 10/50	10	283	140	9,7	46	3'05"	1900	120	-	95	-	3 Vx.335	2	1,5	127/220 M/N	820	830	410		350	54,5	Bomba (Negro Brilho) Tanque/Base (negro em pó)	-	-	-	-
CSV 10/100	10	283	140	9,7	93	5'12"	1900	120	-	95	-	3 Vx.335	2	1,5	127/220 M/N	810	800	450		350	64,8	Bomba (Negro Brilho) Tanque/Base (negro em pó)	450	300	Ø 15	276,5
CSV 10/110	10	283	140	9,7	106	5'35"	1900	120	-	95	-	1-3Vx	2	1,5	127/220*	810	1000	820		350	54,0	Pump (Black) Tank/Base (Black in powder)	800	281,0	Ø 15	-
CSV 10/200	10	283	140	9,7	183	6'30"	1900	120	-	95	-	1-3Vx	2	1,5	127/220*	870	1300	850		350	78,8	Pump (Black) Tank/Base (Black in powder)	800	320,2	Ø 15	288,4
CSV 10/220	10	283	140	9,7	220	7'00"	1900	120	-	95	-	1-3Vx	2	1,5	127/220*	870	1300	850		350	78,8	Pump (Black) Tank/Base (Black in powder)	800	320,2	Ø 15	288,4

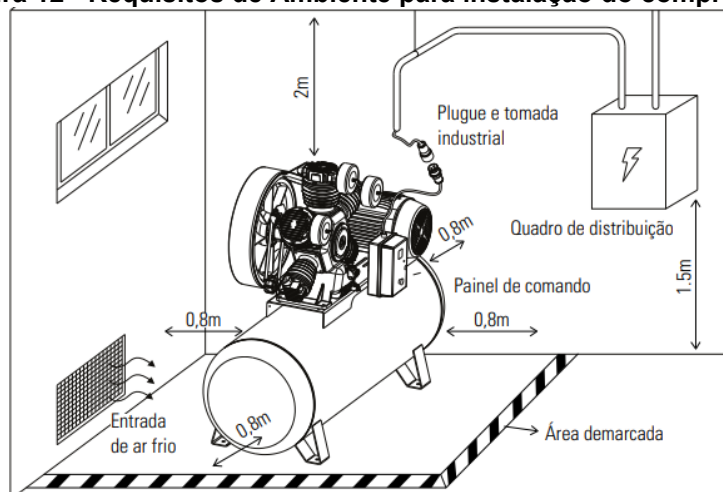
Fonte: (WERLANG, [s.d.]).

O compressor em questão, pode ser classificado como um equipamento de hobby e de uso doméstico, atendendo principalmente atividades de pintura com ar comprimido, com pistolas de baixa produção, ligado a funções básicas como; inflar, encher, pulverizar e desodorizar.

Sua instalação, segundo o fabricante, deve ser em área coberta, com ventilação, sem a presença de gases tóxicos e poeira, atendendo requisitos de temperatura ambiente de funcionamento entre a mínima de 5°C e máxima de 40°C. Tem-se a necessidade de adquirir equipamentos que amortecem suas vibrações de acordo com o seu peso e sua vibração, com sua inclinação não devendo superar o ângulo de 15° e afastamento de 80 cm de qualquer parede. Compressores com mais

de um 1.5 HP necessitam a utilização de uma chave de partida entra o pressostato e ligação do motor. (WERLANG) Para o funcionamento do protótipo, todas as indicações do fabricante foram seguidas.

Figura 12 - Requisitos de Ambiente para instalação do compressor.



Autor: (WERLANG, [s.d.]).

3.1.1 Manutenção preventiva do compressor

Para a manutenibilidade do equipamento, o fabricante indica de forma diária, deve-se realizar a drenagem do condensado de água acumulado no interior do vaso de pressão, verificar se ocorrem ruídos fora do comum. Indica-se também a verificação do nível de óleo presente no reservatório, quando preciso for, completá-lo até o nível. Semanalmente deve-se verificar o funcionamento da válvula de segurança e prezar pela limpeza do equipamento, o qual deve ser feito com o auxílio de detergente neutro. Mensalmente é importante que se efetue a verificação do pressostato, alinhamento entre polia e volante, tensionamento da correia de transmissão e sua condição e revisão de instalações elétricas por profissional habilitado.

As 3 primeiras trocas de óleo, devem ser feitas do período correto, com antecipação do período sempre que o equipamento trabalhar fora dos parâmetros ideais de funcionamento. A utilização específica do tipo de óleo é feita com parâmetros da temperatura ambiente, indicados nas tabelas.

Tabela 1 - Períodos de trocas de óleo lubrificante.

1ª Troca	2ª Troca	3ª Troca
8 horas de serviço	40 horas de serviço seguintes a primeira troca	A cada 300 horas ou 3 meses para linha AUDAZ, a cada 400 horas ou 3 meses para a MSQ 60 + Fort e demais produtos a cada 200 horas ou 2 meses, o que ocorrer primeiro.

Fonte: (WERLANG, [s.d.]).

Tabela 2 - Óleo lubrificante indicado para temperatura ambiente.

Temperatura ambiente	Tipo de óleo	Requisitos para especificação do óleo
Abaixo de 0°C	SAE 10W ou ISO 32	Óleo mineral de base naftênica ou mista. Aditivação: anti-oxidantes (descarbonizantes)
De 0°C até 10°C	SAE 20W ou ISO 68	e anti-espumantes (air release). Não usar óleo multiviscoso. Não usar óleo sintético ou semi-sintético. Consulte o SAC Schulz antes
De 10°C até 40°C	MS LUB SCHULZ (SAE 30W o ISO VG 100)	de usar qualquer outra aditivação diferente da indicada acima.

Fonte: (WERLANG, [s.d.]).

3.2 MANUTENÇÃO CORRETIVA DO COMPRESSOR

A manutenção corretiva, indica-se, ser feita apenas em estabelecimentos credenciados, para garantir a segurança e a confiabilidade do produto. Porém, o fabricante indica alguns possíveis diagnósticos de falha, os quais o quais podem ser precocemente identificados através da aplicação do sistema de aquisição de dados por sensoriamento.

Tabela 3 - Diagnóstico de falhas.

Defeito eventual	Causa provável	Solução	Método de verificação
Pressão do reservatório eleva-se rapidamente ou partidas muito frequentes	Excesso de água no reservatório	Drene o reservatório através do purgador	Acompanhamento de parâmetros e indicadores gerais
Superaquecimento do bloco	Temperatura ambiente elevada	Utilize óleo adequado, verifique o nível	Monitoramento do sensor de temperatura acoplado ao bloco
	Sentido de rotação incorreto	Redimensione o compressor se necessário	
	Demanda do sistema acima da capacidade do compressor	Substitua-o	
Motor elétrico/compressor funcionam interruptamente	Elemento filtrante obstruído	Proceda limpeza	Acompanhamento de parâmetros e indicadores gerais
	Carbonização da placa de válvula		
	Demanda do sistema acima da capacidade do compressor	Redimensione o compressor	

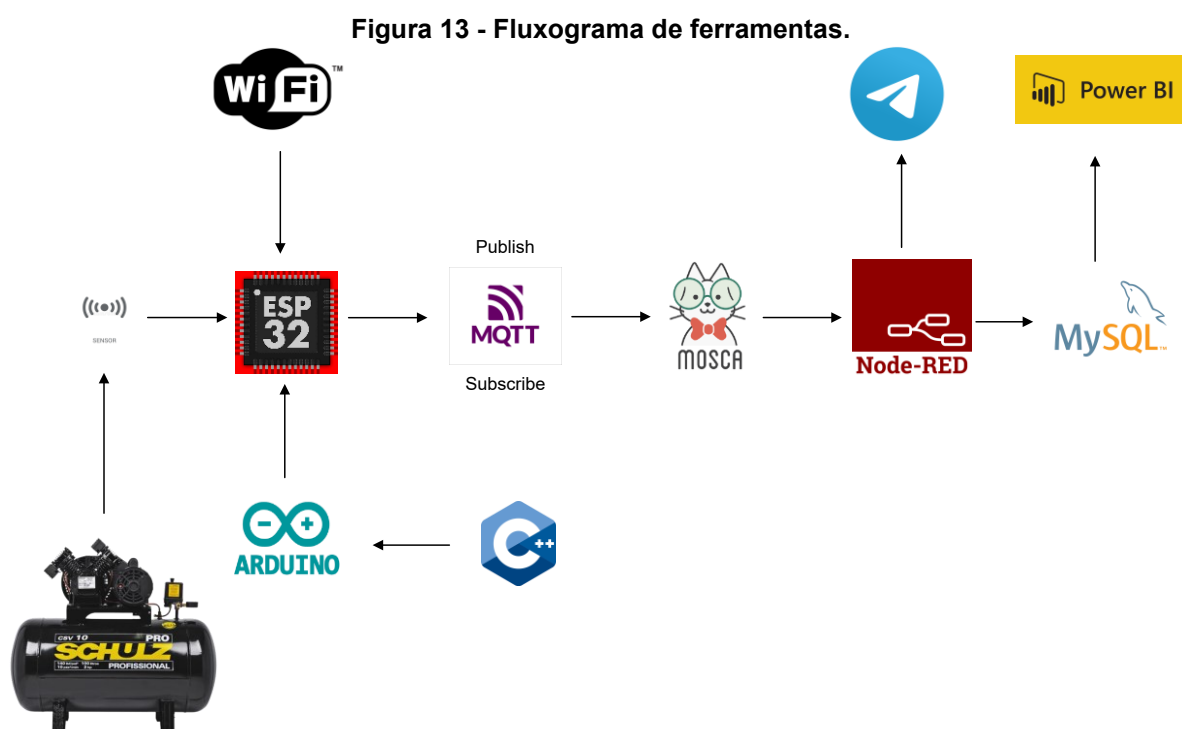
Fonte: Adaptado de (WERLANG).

4 MATERIAIS E MÉTODOS

4.1 PLANEJAMENTO DE PESQUISA

Este capítulo tem como objetivo a apresentação dos materiais e dos métodos, os quais foram feitos dentro do presente estudo, tal como, a interação entre diferentes tipos programas, e sensores utilizados na implementação do sistema de monitoramento, além da validação de seus resultados na prática.

Para o desenvolvimento, será utilizada a composição de 4 placas embarcadas do modelo ESP 32, as quais terão as funções de conectar-se a uma rede local, receber os dados dos sensores acoplados ao compressor, e publicá-los em um broker através do protocolo MQTT. Cada ESP32 ficará responsável pelo acompanhamento de um sensor específico do experimento.



Fonte: Autoria própria.

O projeto tem como ponto inicial, a instalação de sensores ao compressor, os quais serão conectados à placas ESP32, que por sua vez, através da programação em linguagem C++ feita previamente com utilização da IDE Arduino, conseguem estabelecer conexão com a internet local por meio da rede WI-FI. Os dados recebidos

pelas placas, são publicadas no broker Mosca que funciona localmente em um aparelho notebook, por meio da função *Publisher*. Em seguida o Node-RED, conecta-se ao Mosca e se inscreve no tópico para receber as informações através da função *Subscribe*. Os dados recebidos no Node-RED possuem três principais destinos, sendo eles; o acionamento de um BOT, o qual envia a mensagem em um grupo do aplicativo de mensagens Telegram, informando aos usuários deste grupo, que algum sensor registrou um valor fora dos parâmetros pré-estabelecidos. A geração de gráficos de acompanhamento em tempo real por meio do recurso Node-RED Dashboards, o qual pode ser aberto tanto em Desktop quanto em acesso por Smartphone através do IP de rede local. Também é feita o armazenamento das variáveis coletadas pelos sensores através do recurso do Node RED MySQL, o qual guarda as informações recebidas, e as disponibiliza para a criação de relatórios de acompanhamento em PowerBI.

4.2 MATERIAIS

Na presente seção, será abordado os materiais utilizados para o desenvolvimento do protótipo. A Tabela 4 seguir, lista os materiais e suas respectivas quantidades.

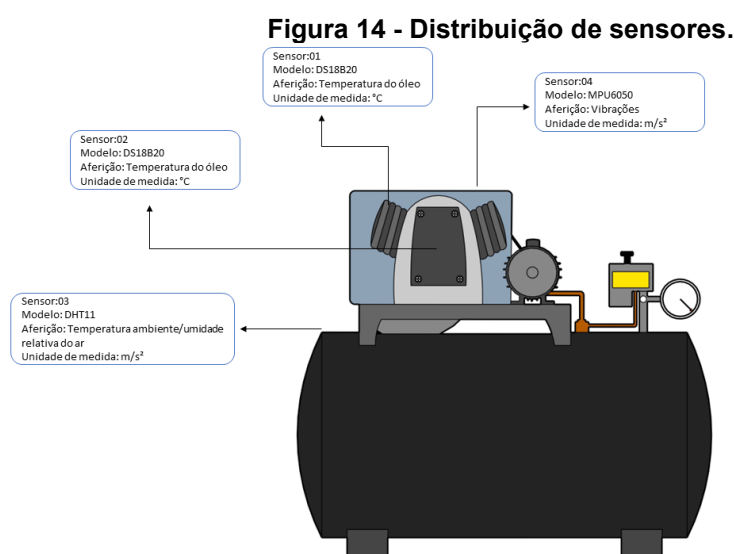
Tabela 4 - Lista de materiais e orçamento.

Materiais	Quantidade	Preço	Total
Placa embarcada esp32	4	R\$ 25,50	R\$ 102,00
Sensor de temperatura DS18b20 capsula	1	R\$ 16,70	R\$ 16,70
Sensor de temperatura DS18b20 chip	1	R\$ 20,90	R\$ 20,90
Sensor acelerômetro 3 eixos MPU6050	1	R\$ 25,90	R\$ 25,90
Sensor de umidade e temperatura DHT11	1	R\$ 16,80	R\$ 16,80
Protoboard 400 pontos	2	R\$ 19,90	R\$ 39,80
Cabo USB	4	R\$ 10,00	R\$ 40,00
Total			R\$ 262,10

Fonte: Autoria própria.

O sensor de temperatura DS18b20 cápsula tem por finalidade monitorar o comportamento do óleo, seu aquecimento e resfriamento durante os acionamentos do

pressostato no processo de compressão de ar. Já o DS18B20 chip, capta a temperatura superficial da carcaça do bloco. O sensor DHT11 busca monitorar as condições ambientais em que a máquina se encontra, a fim de verificar a interferência das condições do ambiente dentro dos parâmetros de funcionamento do compressor. Por fim, o acelerômetro MMA7361 efetua a inspeção das variações de vibração nos eixos X, Y e Z, ocasionadas pelo funcionamento do equipamento, buscando encontrar possíveis desbalanceamentos no sistema mecânico do compressor, na imagem a seguir ilustramos a posição de fixação de cada sensor.



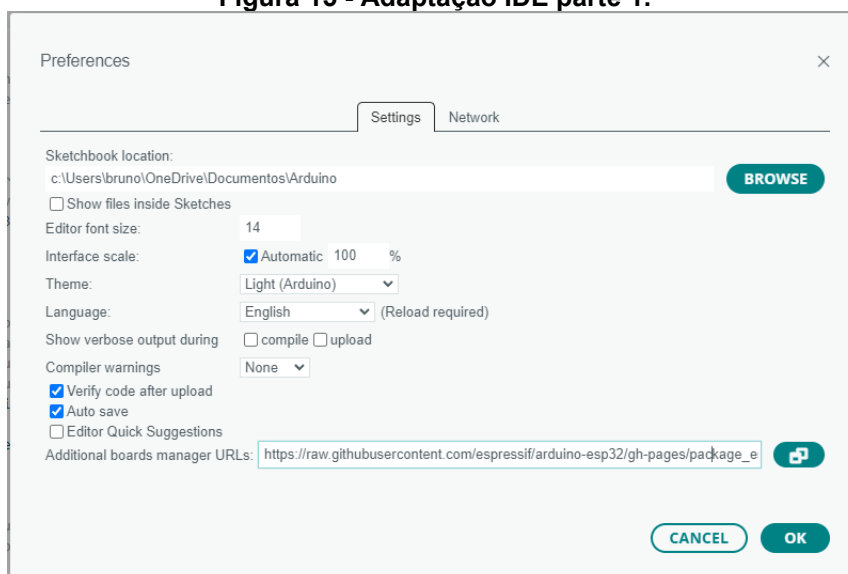
Fonte: Autoria própria.

4.3 IDE ARDUINO

A IDE da Arduino, tem o propósito de editar, compilar e carregar códigos para dispositivos Arduino. A Arduino IDE é um Software de código aberto, encontrada e disponível no próprio site da Arduino, e com algumas adaptações pode ser utilizada para o carregamento de códigos na ESP32.

Para possibilitar o carregamento de códigos no ESP32, basta ir até “Preferences” e na opção “Additional boards manager” “colar a URL presente no Apêndice “x” e clicar em OK.

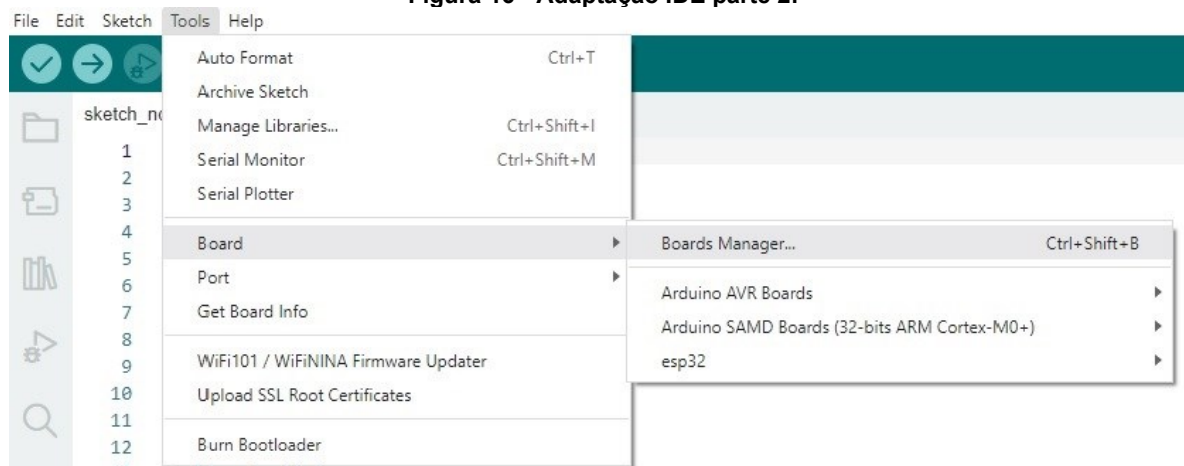
Figura 15 - Adaptação IDE parte 1.



Fonte: Autoria própria.

Depois deste passo, deve-se ir até “Tools”, e na opção “Boards” selecionar “Boards Manager”.

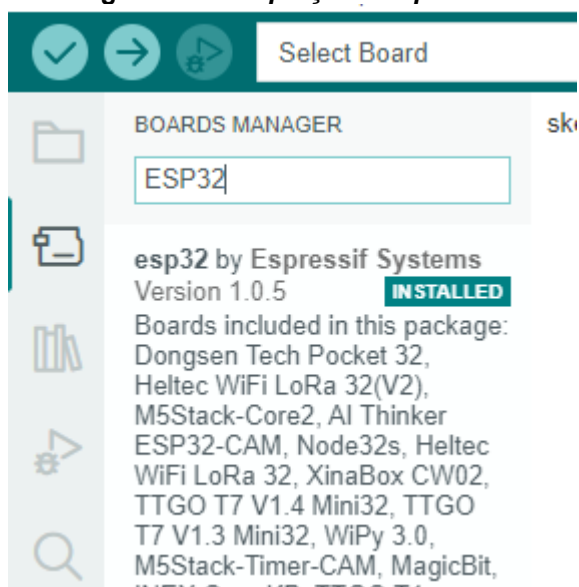
Figura 16 - Adaptação IDE parte 2.



Fonte: Autoria própria.

Por fim, na barra de pesquisa, basta buscar por ESP32 e clicar em “Install” para a versão mais recente disponível, e assim já será possível carregar códigos ao ESP32 com a utilização da IDE Arduino.

Figura 17 - Adaptação IDE parte 3.



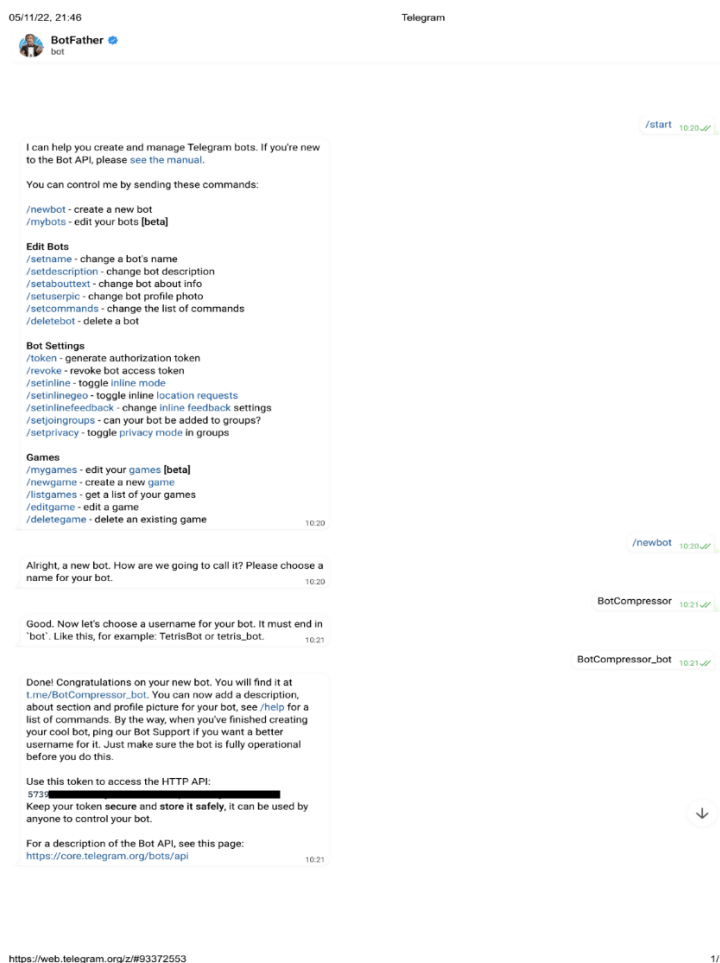
Fonte: Autoria própria.

4.4 TELEGRAM

O Telegram, é uma plataforma no formato de aplicativo para smartphones, a qual busca permitir que usuários se comuniquem e compartilhem arquivos de formatos variados. Dentre as várias possibilidades dentro da plataforma Telegram, uma delas é a criação e utilização de BOT, que podem ser entendidos como robôs, que buscam realizar funções semelhantes a que usuários humanos, fazendo uso dos recursos de inteligência artificial. Tal tecnologia permite a troca de mensagens entre homem e máquina, podendo habilitar comandos de ação e avisos de alertas (OLIVEIRA; SANTOS; NETO, 2016)

A escolha da plataforma Telegram para a criação de um BOT, dentre outras disponíveis, se deu pela facilidade que o processo de criação do BOT, quando comparado a outras plataformas. Para criar um BOT basta iniciar uma conversa com um BOT do próprio sistema Telegram, chamado BotFather, e seguir as instruções de modelagem do BOT, até que se tenha em mãos o seu Token, que servirá para ser utilizado como referência de usuário para configuração e integração com a plataforma Node-RED através do ID da conversa.

Figura 18 - Criação de Telegram BOT.



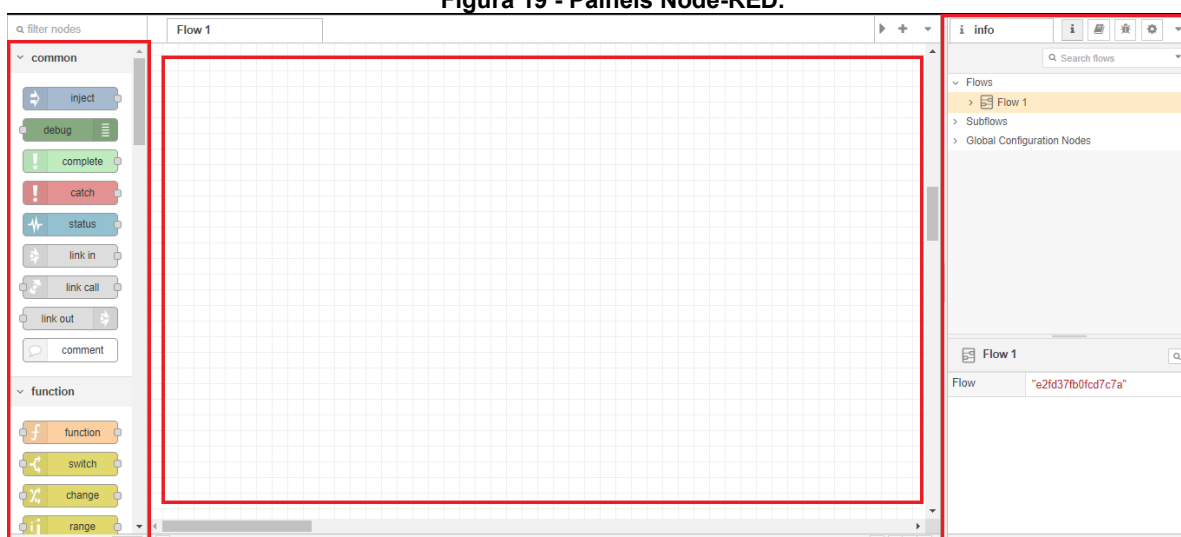
Fonte: Aatoria própria.

4.5 NODE RED

4.5.1 Plataforma Node-RED

Node-RED trata-se de uma plataforma de código aberto baseada em JavaScript, construída pela através do Node.js, desenvolvida pela IBM Emerging Technology, focada em facilitar a utilização de dispositivos IOT. Seu funcionamento é baseado na utilização de nós de funções variadas, podendo ser de entrada ou de saída, os quais podem ser configurados e conectados entre si, podendo propagar mensagens e informações entre eles. Seu layout é dividido em 3 principais blocos, visualizados na figura a seguir (LEKIĆ; GARDAŠEVIĆ, 2018).

Figura 19 - Painéis Node-RED.



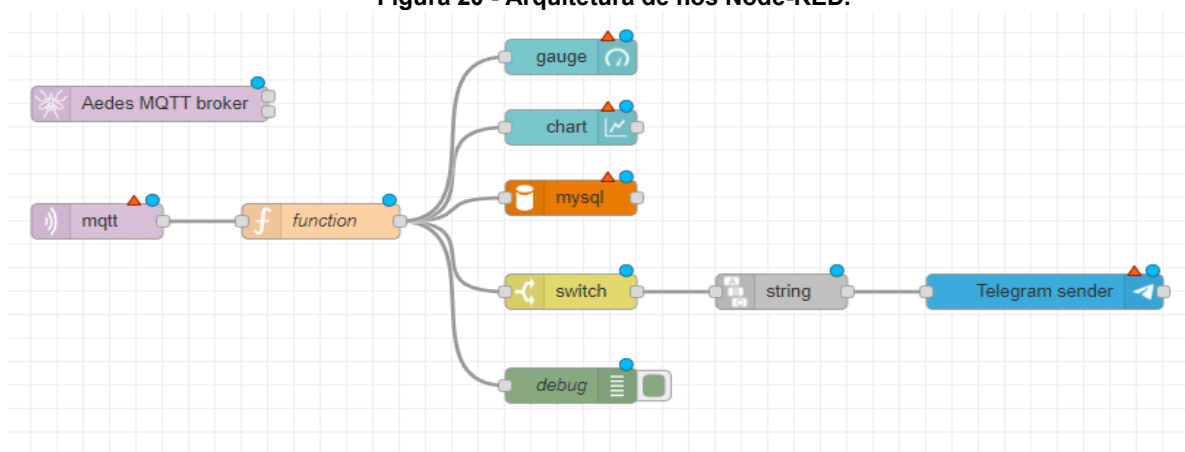
Fonte: Autoria própria.

Seu funcionamento acontece em três principais painéis destacados pelo retângulo vermelho na imagem a acima. O painel à esquerda refere-se ao painel de nós, os quais podem ser instalados na própria plataforma, abrangendo uma grande gama de funções e conexões com os mais variados softwares, API's, ambientes computacionais web, dentre outros. No retângulo central, temos o painel de fluxos, onde os nós podem ser arrastados para o quadro, para que possam estabelecer conexões e criar fluxos de informações. Por fim, à direita, temos o painel de informações e depurações, onde será possível efetuar a configuração de cada um dos nós do painel de fluxo e monitorar o funcionamento deles.

4.5.2 Estrutura de nós

O presente projeto utiliza a configuração de dez nós por sensor, sendo nove deles, nós conectados uns aos outros, e um nó solto. O nó solto trata-se do nó do Aedes MQTT broker, que não tem a necessidade de se repetir nas configurações de todos os sensores utilizados.

Figura 20 - Arquitetura de nós Node-RED.



Fonte: Autoria própria.

O nó Aedes, é o nó ao qual tem por objetivo efetuar a configuração do broker, elemento primordial para utilização do protocolo MQTT. Através da inicialização no Node-RED por meio do Prompt de comando, mediante o comando “node-red”, ao final da inicialização, o Node-RED indicará o endereço de IP local, o qual terá todos os códigos de desenvolvimentos carregados na ESP32 apontados para este endereço de IP local, através da porta 1883. Possibilitando assim, a comunicação local entre as placas e o computador.

Figura 21 - Inicialização Node-RED

```

node-red
Microsoft Windows [versão 10.0.19043.2130]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\bruno>node-red
5 Nov 14:33:41 - [info]

Welcome to Node-RED
=====

5 Nov 14:33:41 - [info] Node-RED version: v3.0.2
5 Nov 14:33:41 - [info] Node.js version: v16.17.0
5 Nov 14:33:41 - [info] Windows_NT 10.0.19043 x64 LE
5 Nov 14:33:43 - [info] Loading palette nodes
5 Nov 14:33:46 - [info] Dashboard version 3.1.7 started at /ui
5 Nov 14:33:46 - [info] Settings file : C:\Users\bruno\.node-red\settings.js
5 Nov 14:33:46 - [info] Context store : 'default' [module-memory]
5 Nov 14:33:46 - [info] User directory : \Users\bruno\.node-red
5 Nov 14:33:46 - [warn] Projects disabled : editorTheme.projects.enabled=false
5 Nov 14:33:46 - [info] Flows file : \Users\bruno\.node-red\flows.json
5 Nov 14:33:46 - [info] Server now running at http://127.0.0.1:1888/
5 Nov 14:33:46 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

5 Nov 14:33:46 - [info] Starting flows
5 Nov 14:33:46 - [info] Started flows
5 Nov 14:33:46 - [info] [aedes broker:Mosca broker] Binding aedes mqtt server on port: 1883
5 Nov 14:33:46 - [info] [mqtt-broker:1486bc7babc6ba7f] Connected to broker: mqtt://192.168. :1883
  
```

Fonte: Autoria própria.

O nó com o nome de “mqtt” é o nó responsável por realizar os recebimentos de mensagens dos tópicos das funções Publish e Subscribe, atuando no projeto unicamente como Subscribe onde o agente publicador é a ESP32. Na sequência temos o nó “function” que tem por objetivo receber a mensagem e endereçá-la a uma coluna específica do banco de dados MySQL. O mesmo nó pode ser endereçado para os nós “Dash boards” que são os nós “Gauge” e Chart, os quais fornecem recursos gráficos de acompanhamento por meio de gráficos de linhas e indicadores de valores. Também temos o nó “Debug”, o qual retorna a saída do nó, e serve como monitor do funcionamento do nó ao qual ele está conectado, podendo retornar o valor de saída do nó, ou algum possível erro.

A sequência de nós “switch”, “String” e “Telegram sender”, fazem parte de um sistema de alerta para ocorrência de valores recebidos fora da margem tolerada, como por exemplo um superaquecimento do sistema. O nó “String” funciona com a função de filtro, então sempre que algum valor for superior ou inferior, a um determinado valor pré-estabelecido, ele dá sequência ao fluxo, passando para o nó “String”, o qual formula um texto com informações do valor recebido e passa como mensagem para o nó “Telegram sender”, que por sua vez, utilizado os recursos de um BOT. O BOT enviará mensagens podendo ser em um grupo ou em uma mensagem privada, alertando o estado atual do sensor, enquanto o valor em questão estiver fora dos parâmetros.

O sistema de nós, é implementado então com um fluxo idêntico ao descrito, para cada parâmetro coletado.

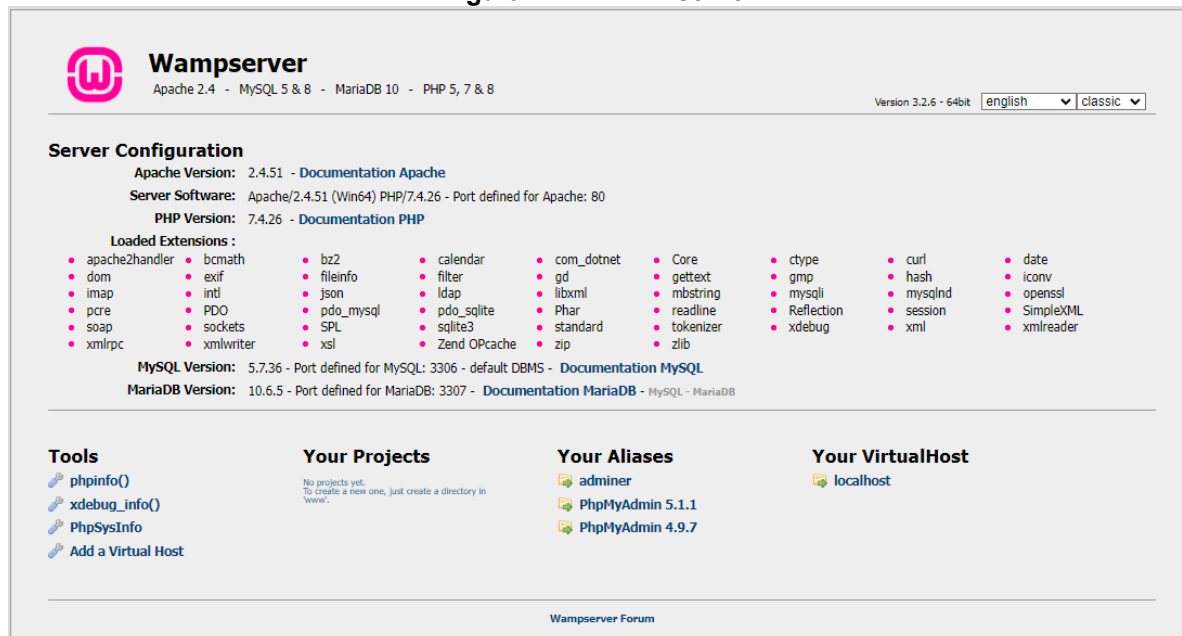
4.6 BANCO DE DADOS MYSQL

Para o armazenamento de dados, buscasse um sistema que trabalhe com eficiência quanto a big dates e interação com outros recursos e softwares, dentre estes, o MySQL possui disponibilidade de nós que interagem com o Node-RED e fazem a ponte entre recebimento de dados e armazenamento, ideal para utilização de históricos de funcionamento dentro do protótipo.

Para a utilização do MySQL, foi utilizado pacote de recursos chamado WAMP, que se trata de um acrônimo para as palavras Windows, Apache, MySQL e PHP. Com estes recursos é possível armazenar os dados recebidos por meio do MQTT,

armazená-los e possibilitar a sua consulta através do Power Bi para criação de dashboards.

Figura 22 - WAMP server.



Fonte: Autoria própria.

4.7 POWER BI

O Power BI faz parte do pacote de softwares que a Microsoft comercializa, com a finalidade de trazer um apanhado de dados para uma visão mais analítica e interativa. A sua grande versatilidade permite trabalhar com diversos modelos de banco de dados, tanto locais quanto em nuvem, sem muitas restrições. Outro ponto de destaque do Power Bi, é a sua possibilidade de compartilhamento de conteúdo. O grande foco da ferramenta, está na criação de Dashboards e relatórios, dos mais variados seguimentos (MIHART, 2022).

4.8 SENSOR DE TEMPERATURA DS18B20

Sensores de temperatura buscam identificar o aquecimento de determinado equipamento ou processo. No presente estudo, foi escolhido o sensor DS18B20 para medir a temperatura de trabalho do óleo lubrificante e dos cabeçotes do bloco do

compressor. Sua escolha foi feita, por trazer características propícias a sua utilização como;

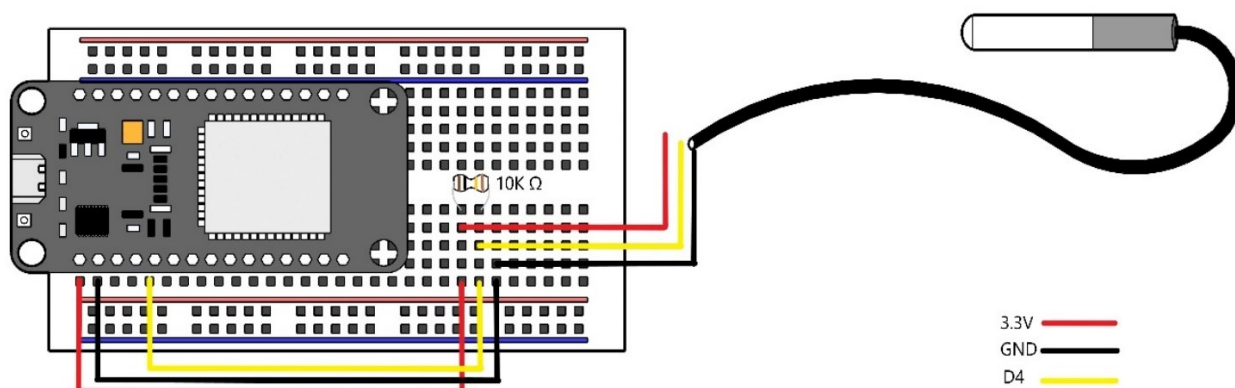
Tabela 5 - Características DS18B20

Parâmetro	Descrição
Condição de funcionamento	Submersa(cápsula), Ambiente (chip)
Resolução	9 a 12 bits
Alimentação	3 a 5.5 V
Faixa de medição	-55 a 125°C
Precisão	±0.5°C na faixa de -10°C a 125°C

Fonte: Adaptado (“DS18B20 - Programmable Resolution 1-Wire Digital Thermometer”, 2019).

O sensor DS18B20 possui 3 fios em seu sistema, nas cores; vermelho, preto e amarelo. O Fio vermelho é utilizado para alimentação VCC de 3.3 V. O fio preto trata-se do GND. Por fim, o fio amarelo atua em sistema One-Wire, trazendo as informações de variação de tensão ocasionadas pela oscilação da temperatura.

Figura 23 - Circuito de montagem do sensor DS18b20 cápsula.



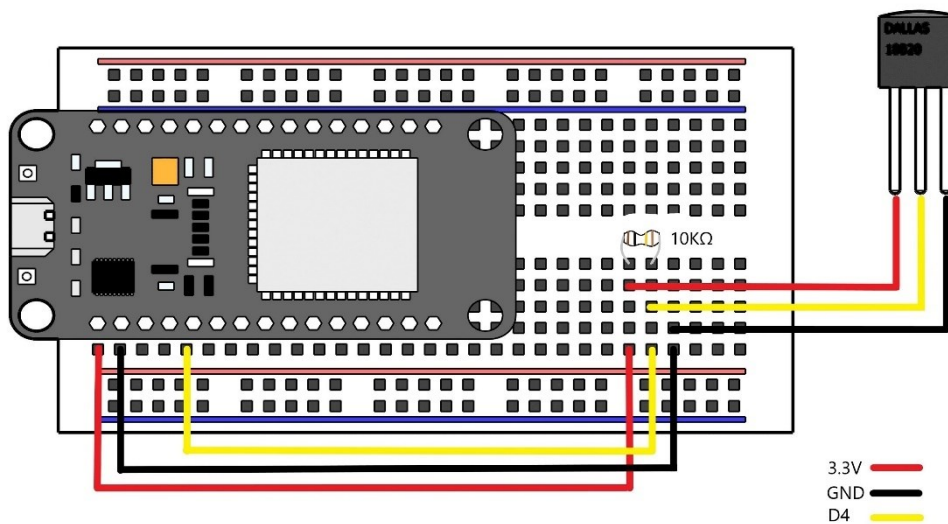
Fonte: Autoria própria.

Figura 24 - Montagem de sensor e termômetro no cárter do compressor.



Fonte: Autoria própria.

Figura 25 - Circuito de montagem sensor DS18B20 chip.



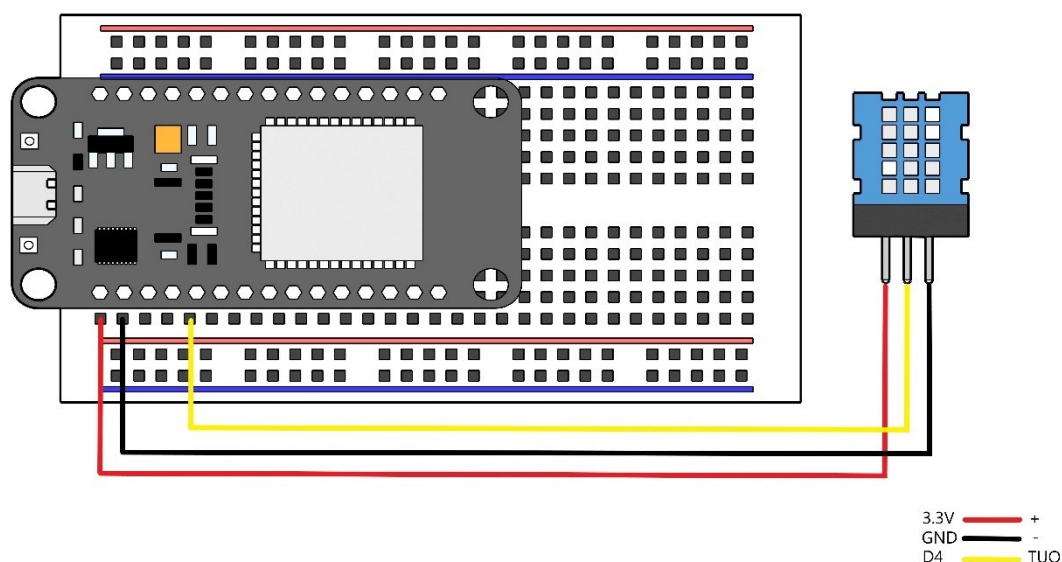
Fonte: Autoria própria.

4.9 SENSOR DE TEMPERATURA E UMIDADE DHT11

O sensor DHT11 é um sensor que efetua medições de temperatura e umidade, podendo medir entre 0 e 50°C com precisão de $\pm 2^\circ\text{C}$., e 20 a 90% da umidade relativa do ar, com precisão de $\pm 5\%$ UR. Seu circuito é de simples

entendimento, contendo um pino VCC que pode ser alimentado com tensão de 3.3V a 5V, com o pino GND para referência de aterramento, e um pino GPIO para utilização 1-Wire, alguns modelos dispensam a utilização de resistores, uma vez que já possuem resistor pull-up acomplado (GAY, 2018).

Figura 26 - Circuito de montagem sensor DHT11.



Fonte: Autoria própria.

Tabela 6 - Características do sensor DHT11.

Parâmetro	Descrição
Resolução	16 bits
Comunicação	1-Wire
Captação	Simultânea
Faixa de medição Umidade relativa	20% a 90%
Alimentação	3V a 5V
Faixa de medição Temperatura	0°C a 50°C

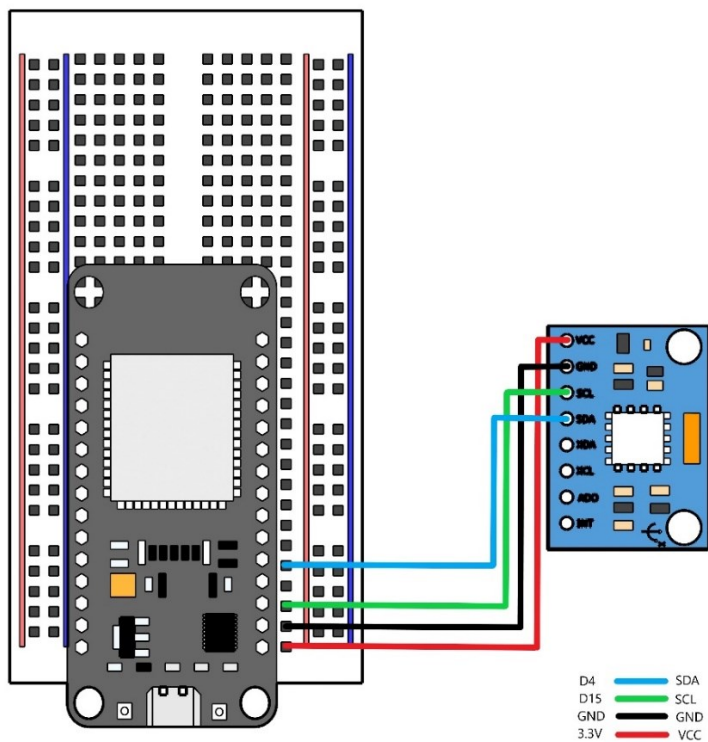
Fonte: (GAY, 2018).

4.10 SENSOR ACELERÔMETRO MPU6050

O acelerômetro é um sensor desenvolvido para detectar e medir a taxa de aceleração de um objeto que lhe é acoplado, assim como a aceleração gravitacional. A sua aplicabilidade acaba sendo possível para a utilização em monitoramentos de vibrações de equipamentos mecânicos, assim como sua posição, uma vez que também possui o recurso de giroscópio. O MPU conta com 8 pinos, sendo o VCC o

qual alimenta o módulo, GND como aterramento, SCL e SDA para comunicação I2C, XDA e XCL para fazer interface de outros sensores, AD0 para alterar endereço I2C e INT para interrupção (FEDOROV, 2015).

Figura 27 - Circuito de montagem sensor MPU6050.



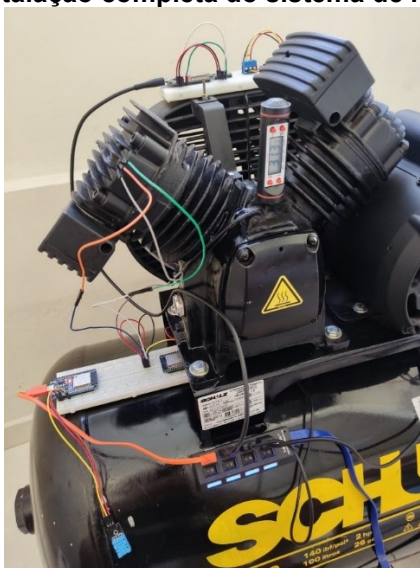
Fonte: Autoria própria.

Tabela 7 - Características MPU 6050.

Parâmetro	Descrição
Resolução	12 bits
Comunicação	I2C
Giroscópio	3 eixos
Acelerômetro	3 eixos
Captação de eixos	Simultânea
Faixa de medição	± 2 , ± 4 , ± 8 , ± 16 g

Fonte: (FEDOROV, 2015).

Figura 28 - Instalação completa do sistema de monitoramento.



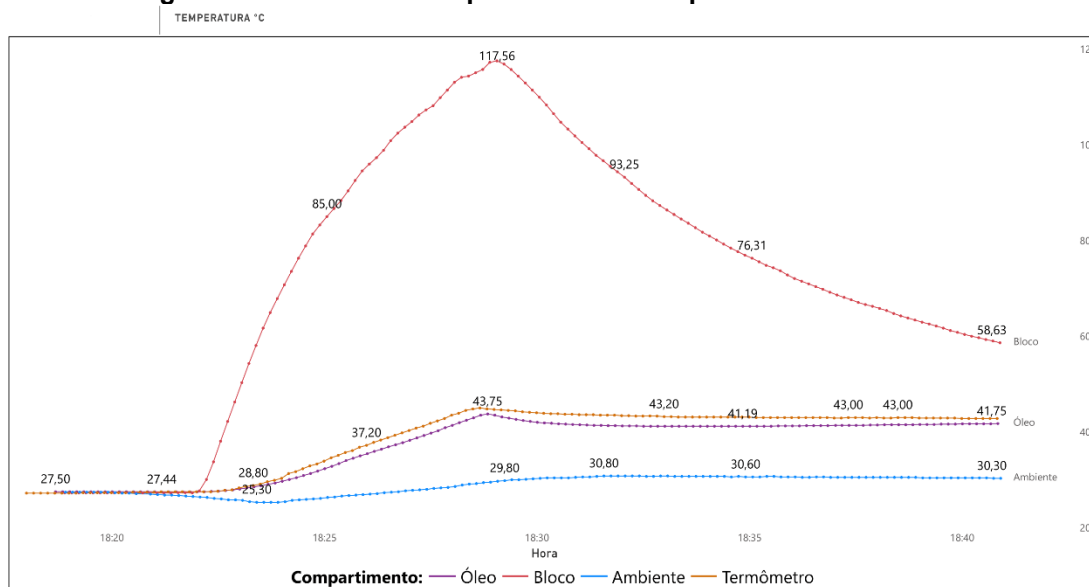
Fonte: Autoria própria.

5 RESULTADOS

5.1 ANÁLISE DE PARTIDA COM RESERVATÓRIO VAZIO

Em um funcionamento geral de compressores alternativos, o trabalho mais árduo que o equipamento pode executar é o encher o reservatório vazio. Partindo do reservatório com a pressão atmosférica registrando o valor de PSI no manômetro até o desarme do pressostato na casa dos 130 PSI. Para esta análise, foi acoplado ao reservatório um termômetro digital com intuito de comparar os valores do comportamento termodinâmico do óleo extraídos do sensor DS18b20 com os registrados pelo termômetro coletados de forma visual com o auxílio de gravação de um smartfone e verificação do horário de início.

Figura 29 - Análise de temperaturas na compressão de 0 a 130 Psi

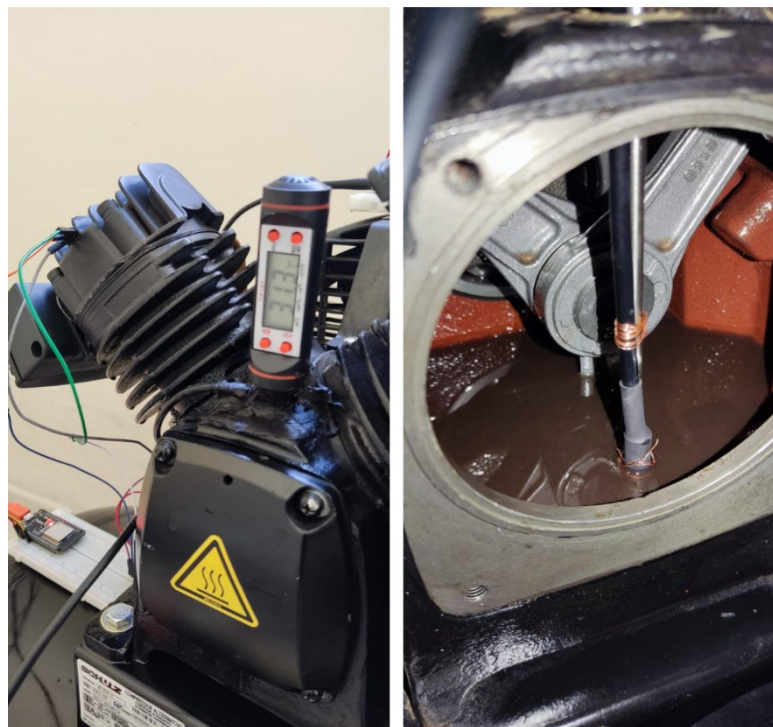


Fonte: Autoria própria.

No teste, o valor que registra a temperatura de aquecimento do bloco, fica bem próximo da sua faixa máxima de medição, que é de 125°C, registrando 117,56°C com aproximadamente 8 minutos e 40 segundos de funcionamento. Conforme o compressor trabalha, os parâmetros de aquecimento do óleo registrados pelo sensor e pelo termômetro analógico aquecem progressivamente e mantem sua temperatura após o desligamento do compressor resfriando em uma taxa lenta. Na temperatura ambiente, nota-se que devido a respiro do compressor demandar fluxo de ar para fora, enquanto o compressor está frio ele tende a resfriar levemente o ambiente que de 27,50°C passa a registrar 25,30°C, um resfriamento de 2,20°C, mas conforme o compressor aquece, ele passa demandar massa de ar quente para o ambiente, e então a temperatura chega na casa de 30,80°C, um aquecimento de 5,5°C, o qual se mantem com um resfriamento lento.

Para o teste de comparações entre o sensor DS18B20, e o termômetro analógico, foi efetuado o acesso de ambos os equipamentos acoplados um ao outro por meio de canal de acesso para a troca de óleo do cárter do compressor, com objetivo de registrar a mesma temperatura na mesma área de aquecimento do líquido de lubrificação e arrefecimento. Para a vedação do espaço do acesso, foi utilizado silicone para juntas de motor, por se tratar de um material removível, podendo ser retirado sem causar prejuízo a integridade do compressor e resistir a altas temperaturas.

Figura 30 - Acoplamento do termômetro ao cárter do óleo

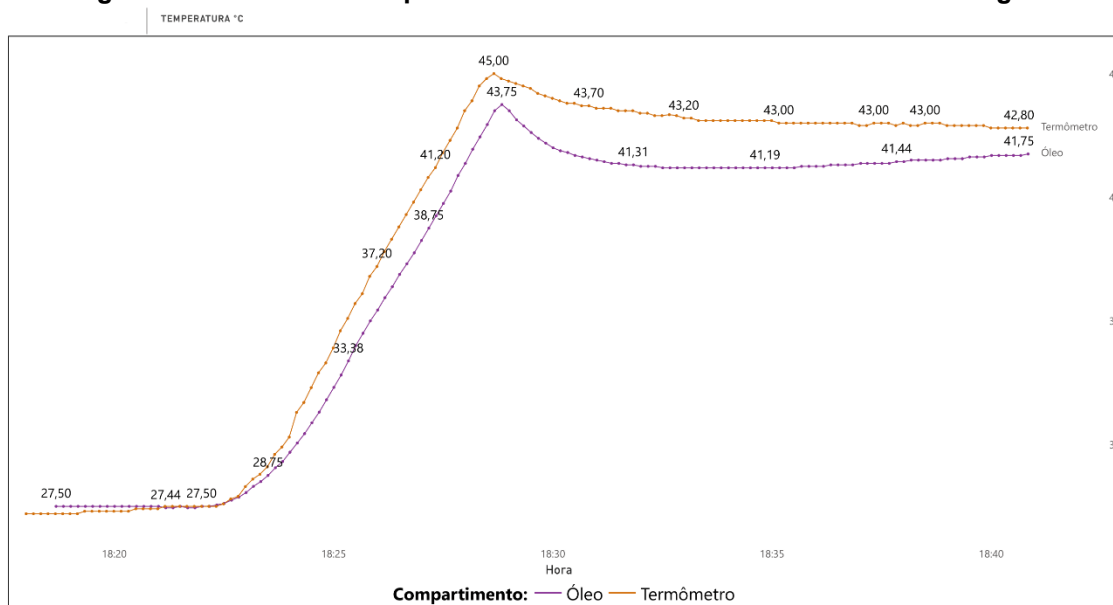


Fonte: Autoria própria.

Durante 20 minutos foi acompanhado com intervalo de 10 segundos, os valores registrados tanto pelo sensor DS18B20 quanto com o termômetro analógico. Ambos os sensores partem de em regime estacionário de valores próximos, o DS18B20 registra 27,50°C enquanto o termômetro analógico registra 27,44°C, uma diferença de 0,06°C. Quando o compressor dá a partida para o enchimento do reservatório, os valores passam a se distanciar progressivamente até atingir o fim do acionamento, registrando no termômetro analógico 45°C e o sensor 43,75°C. No resfriamento do equipamento a diferença entre os equipamentos de medições chega a mais de 2°C, e depois voltam a se aproximar. Tal característica comportamental gráfica pode ser justificada pelo contato da haste do termômetro próxima do bloco do compressor, captando então não só a temperatura do óleo, mas também de parte do aquecimento do bloco. Nota-se uma pequena diferença entre o momento em que os valores são dispostos no gráfico, a diferença é de aproximadamente 5 segundos entre os valores registrados analogicamente e os consultados no banco de dados MySQL, provavelmente ocasionados pelo tempo em que o conjunto leva para reconhecer e armazenar a informação. Conclui-se que os valores apesar de se aproximarem e se afastarem em determinados momentos, descrevem comportamento gráfico

semelhante, e esperados, de aquecimento de resfriamento como pode-se ver no gráfico.

Figura 31 - Análise de temperatura do sensor DS18B20 vs Termômetro digital.



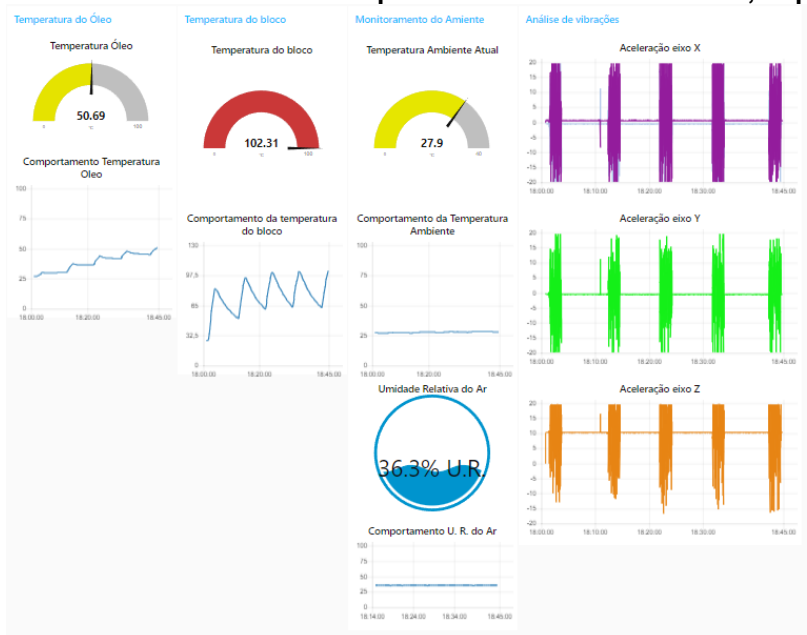
Fonte: (Autoria própria)

Durante este teste foi possível notar a capacidade limitada de se acoplar um sensor um sensor de temperatura com haste para aferição da temperatura do óleo de forma contínua, uma vez que os dados acabam sendo alterados devido ao contato do equipamento não ser restrito ao óleo do compressor.

5.2 ANÁLISE DE PARTIDAS CONSTANTES

Para compreender um pouco do comportamento do compressor monitorado, foram realizadas partidas sincronizadas durante o período de 1 hora.

Figura 32 - Monitoramento em tempo real Node-RED dashboard, 5ª partida.

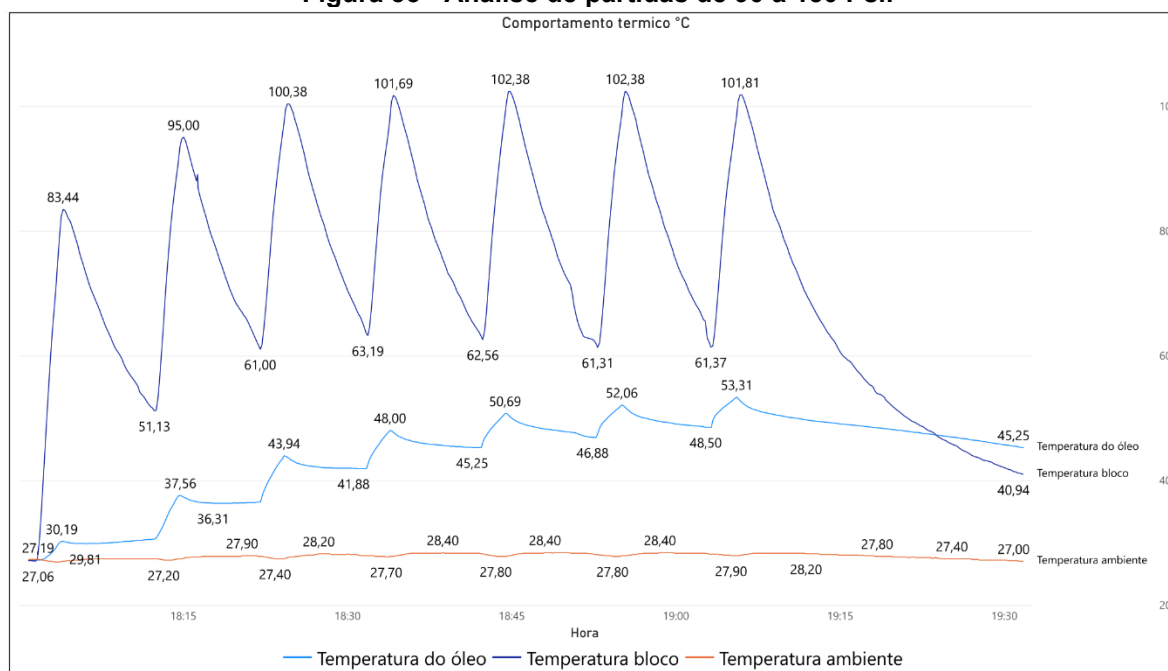


Fonte: Autoria própria.

Dentre os nós disponíveis que temos no Node-RED, os nós do grupo Dashboards nos permitem montar e configurar um layout com indicadores e gráficos que nos permitem visualizar o comportamento recente da máquina bem como seus valores em tempo real. Os indicadores permitem que seja criado um esquema de cores para alertar de forma visual em que estado que estão os parâmetros de máquina. No exemplo acima, os indicadores de temperatura do óleo, temperatura do bloco, e Monitoramento ambiente, oscilam entre as cores verde, amarelo e vermelho, remetendo estados de funcionamento dentro do esperado, alerta, e alerta de perigo.

Para monitorar e entender como funcionaria o monitoramento, quando submetido o compressor pneumático a disparos consecutivos. Durante uma hora foram realizados 7 disparos, com o primeiro disparo no minuto 0, e o último no minuto 60, regulados com acionamento do pressostato, ou seja, de 90 a 130 PSI. O teste foi realizado no início da noite, com temperatura ambiente por volta dos 27°C e baixa umidade relativa do ar.

Figura 33 - Análise de partidas de 90 a 130 Psi.



Fonte: Autoria própria.

Nota-se através da análise gráfica da Figura 33, que no início do experimento, os valores de temperatura encontravam-se próximos, indicando estabilidade da temperatura do sistema; conforme os disparos acontecem, nota-se o crescimento consecutivo nos 3 primeiros disparos, até que a partir do quarto disparo a temperatura passa a se estabilizar para o comportamento da temperatura do bloco. Já para a temperatura do óleo, nota-se que a estabilização não aconteceu durante o monitoramento de uma hora, a cada disparo a tem-se uma nova temperatura máxima. A temperatura ambiente mostra-se oscilante, demonstrando que o comportamento do compressor, apesar de que em pequena escala, consegue alterar o valor da temperatura ambiente com seu disparo.

Tabela 8 - Teste de acionamentos consecutivos.

Acionamentos	Horário	ΔP Pressão [PSI]	Δt Acionamentos [Hrs]	Δt entre partidas [Hrs]	Temperatura do bloco [°C]	ΔT Bloco [°C]	Temperatura do óleo [°C]	ΔT Óleo [°C]
1	18:01:35 18:04:04	40	00:02:29	00:10:59	27,3 83,44	56,14	27,19 30,19	3
2	18:12:34 18:15:04	40	00:02:30	00:09:30	51,13 95	43,87	30,56 37,56	7
3	18:22:04 18:24:34	40	00:02:30	00:09:50	61 100,38	39,38	36,5 43,94	7,44
4	18:31:54 18:34:14	40	00:02:20	00:10:30	63,19 101,69	38,5	41,88 48	6,12
5	18:42:24 18:44:54	40	00:02:30	00:10:30	62,56 102,38	39,82	45,25 50,69	5,44
6	18:52:54 18:55:24	40	00:02:30	00:10:20	61,31 102,38	41,07	46,88 52,06	5,18
7	19:03:14 19:06:04	40	00:02:50		61,37 101,31	39,94	48,5 53,31	4,81

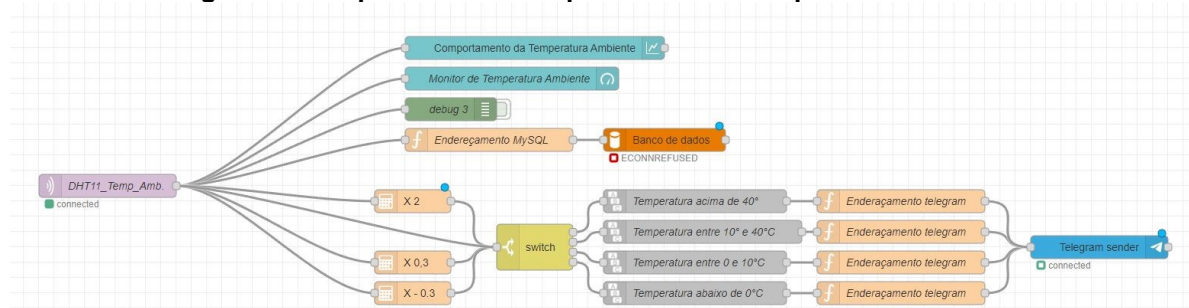
Fonte: Autoria própria.

A partir dos dados extraídos da Tabela 8, conseguimos notar o quanto o pressostato se manteve estável quanto ao tempo de enchimento do compressor, registrando apenas um valor com a diferença de um segundo no primeiro acionamento, dado que no sétimo disparo, onde se tem o enchimento levando 2 minutos e 50 segundos, ouve alteração na válvula de saída de ar. Todas as partidas estiveram sempre próximas do espaço de 10 minutos de intervalo. Na tabela pode-se ver com mais clareza a estabilização do sistema a partir do terceiro disparo, onde ΔT de ganho de temperatura, passam a ficar próximos um do outro. Já para o aquecimento da temperatura do óleo, o ΔT apresenta sinais de estabilidade a partir do quarto disparo, onde passa a reduzir seus valores com uma menor taxa, indicando uma possível estabilidade.

5.3 SISTEMA DE SEGURANÇA EM NODE-RED COM ALERTAS VIA TELEGRAM

Para o teste de alertas de segurança, a arquitetura de nós foi alterada, para que os valores que chegassem dos sensores, remetesse a valores maiores do que os reais, podendo assim disparar os alertas de segurança via Telegram.

Figura 34 - Arquitetura de Nós para alerta de temperatura ambiente.

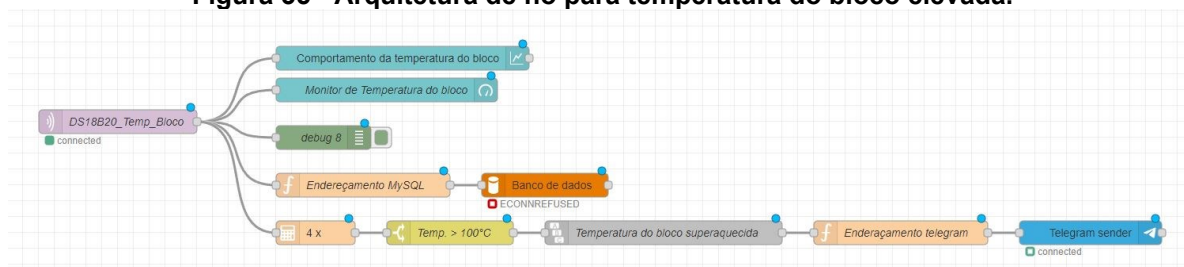


Fonte: Autoria própria.

No sensor DHT11 que capta os valores de temperatura e umidade ambiente, para a variável de temperatura ambiente foram utilizados 4 valores, sendo um real e outros 3 manipulados por expressões algébricas a fim de elevar ou reduzir seus valores podendo assim disparar todos os alarmes possíveis vindos do sensor. Foram definidos 4 parâmetros com 4 tipos de alertas, um deles, alertando quanto a temperatura superior a 40°C onde o fabricante não recomenda a utilização do equipamento. Também foram estabelecidos mais 4 alertas para a recomendação da

utilização específica de óleo lubrificante a partir da temperatura de trabalho, onde o fabricante indica composições de óleos específicos para as faixas de temperatura menor que 0°C, entre 0°C e 10°C e 10 e 40°C.

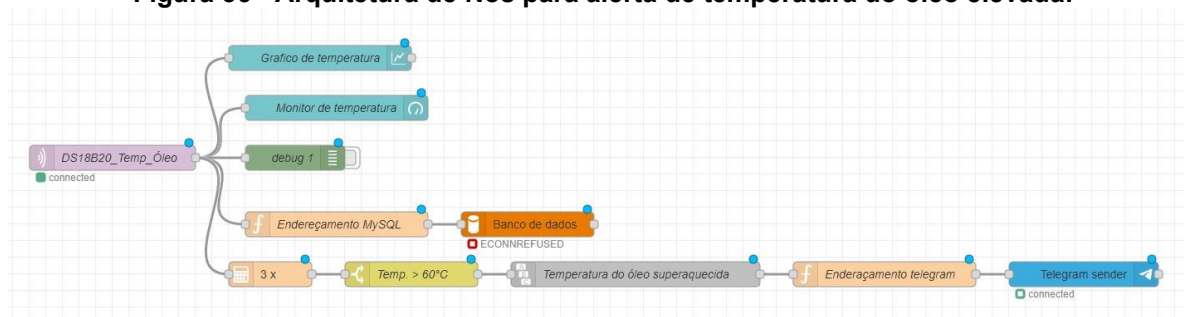
Figura 35 - Arquitetura de nó para temperatura do bloco elevada.



Fonte: Autoria própria.

Para o sensor de temperatura que registra o aquecimento do bloco, foi alterado a arquitetura de nós do sistema para multiplicar por 4 vezes a temperatura que bloco estava registrando quando desligado. Para fins de testes, foi definido um alerta para ser disparado quando a temperatura do bloco superasse a casa dos 100°C.

Figura 36 - Arquitetura de Nós para alerta de temperatura do óleo elevada.



Fonte: Autoria própria.

No sensor que captava a oscilação da temperatura do óleo, foi multiplicado por 3 vez o valor recebido, para assim extrapolar a casa dos 60°C e emitir um alerta de que o óleo do compressor estava trabalhando fora do limite pré-estabelecido;

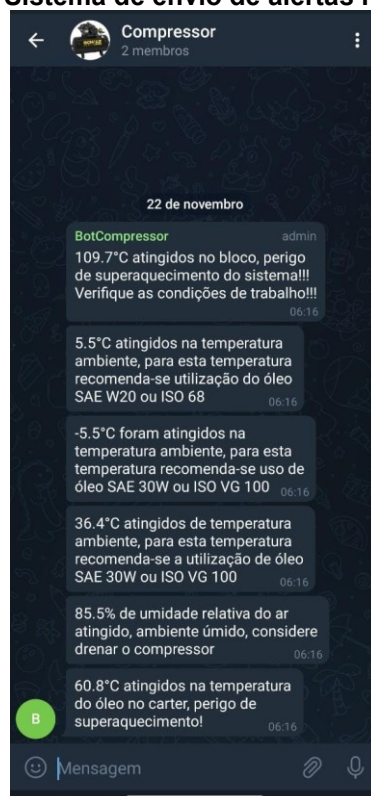
Figura 37 - Arquitetura de nós para alerta da alta umidade.



Fonte: Autoria própria.

Por fim, para o sensor DHT11, coletando o parâmetro de umidade relativa do ar, foi alterado a sua arquitetura para dobrar o valor da umidade, excedendo assim o valor de 80% de umidade relativa do ar, disparando o alarme, e indicado a possibilidade de drenar o compressor uma vez que em ambientes úmidos, o compressor acaba por armazenar água em seu reservatório, devendo ser drenado a partir do expurgador para evitar problemas de funcionamento.

Figura 38 - Sistema de envio de alertas no Telegram.



Fonte: Autoria própria.

Desta forma, foi possível enviar mensagens de alertas de forma instantâneas, com a possibilidade de adicionar mais usuários ao grupo para que também recebam informações do sistema de manutenção. As mensagens mantem seu envio constante,

com frequência de atualização dos parâmetros a cada 10 segundos, informando a situação e o estado atual do equipamento, até que o parâmetro consiga se reestabelecer. Todas as informações dos alertas foram criadas a partir do manual de recomendações do fabricante, porém podem ser adaptas caso julgue-se necessário.

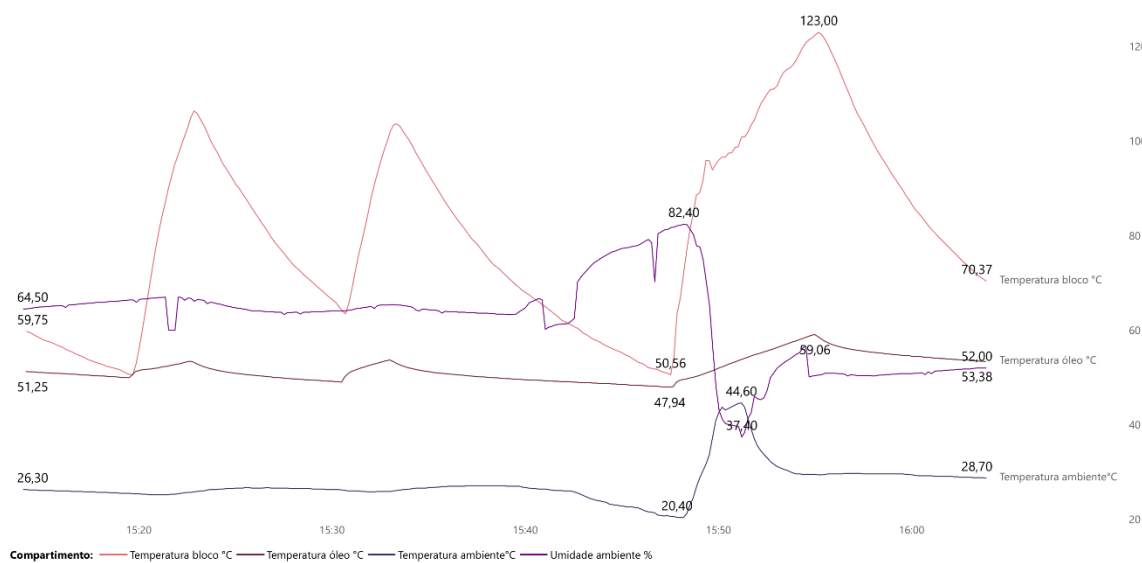
Para testar o funcionamento na prática, foram redefinidos os parâmetros de alerta, para serem emitidos em casos extremos, como; temperatura ambiente superior a 40°C, temperatura do cabeçote superior a 120°C e umidade relativa do ar superior a 80%. Para ajudar indução de acionamento dos alarmes, foram utilizados um soprador térmico e um umidificador, a fim de simular as condições para que o sistema de alerta pudesse ser acionado e informar o usuário dos parâmetros atuais do equipamento em tempo real, dentro das margens de risco.

Figura 39 - Teste prático do sistema de alertas.



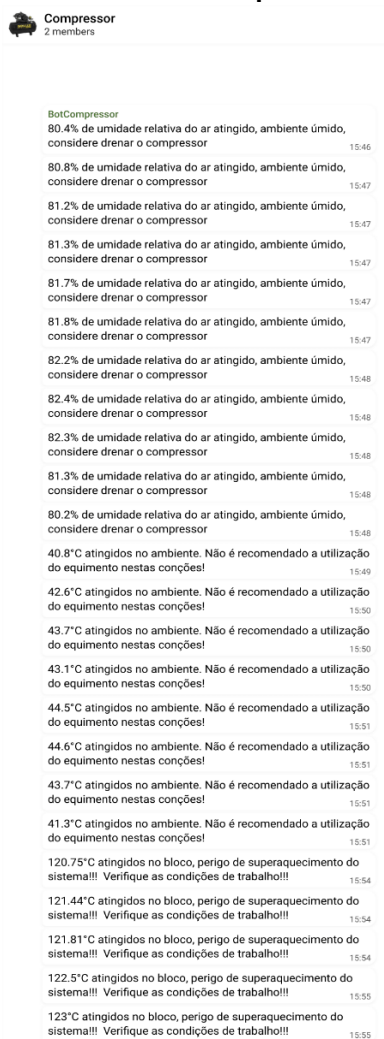
Fonte: Autoria própria.

Figura 40 – Gráfico de variáveis do teste prático do sistema de alertas.



Fonte: Autoria própria.

Figura 41 - Resultado do teste prático de alertas.



Fonte: Autoria própria.

O sistema demonstrou-se eficiente, registrando todos os valores que saíram dos parâmetros ideias de funcionamento previamente definidos, com tempo de resposta rápido, possibilitando o acompanhamento atualizado do comportamento dos parâmetros durante os momentos que esteve dentro da margem de risco definida. Garantindo assim, a entrega de informações mesmo fora do acesso de rede local do equipamento, sendo um recurso com capacidade de auxiliar a tomada de decisões de forma ágil.

5.4 RELATÓRIO EM POWER BI DE DASHBOARDS COM HISTÓRIO DE FUNCIONAMENTO

Para acompanhamento do histórico do equipamento, foi desenvolvido através da ferramenta Power Bi, um dashboard de acompanhamento de parâmetros de máquina, o qual demonstra através de gráficos e indicadores o comportamento dos registros sensoriais de qualquer momento dentro da linha temporal de funcionamento do equipamento, apontando valores de máximo, mínimo, e médio, podendo ser ajustado através de filtros de data, hora e minuto.

Para que os indicadores fossem gerados, foi necessário conectar o Power Bi ao banco de dados MySQL, o qual durante o período de acompanhamento do funcionamento registrou e armazenou um total de ao total 86 mil 446 registros de dados sensoriais do compressor, no dia 20/11/2022 durante o período das 09:00:00 horas às 17:00:00 horas. de forma interrupta. Simulando assim, um funcionamento extensivo do equipamento, o qual trabalhou com um fluxo padrão de vazão de ar comprimido durante o experimento.

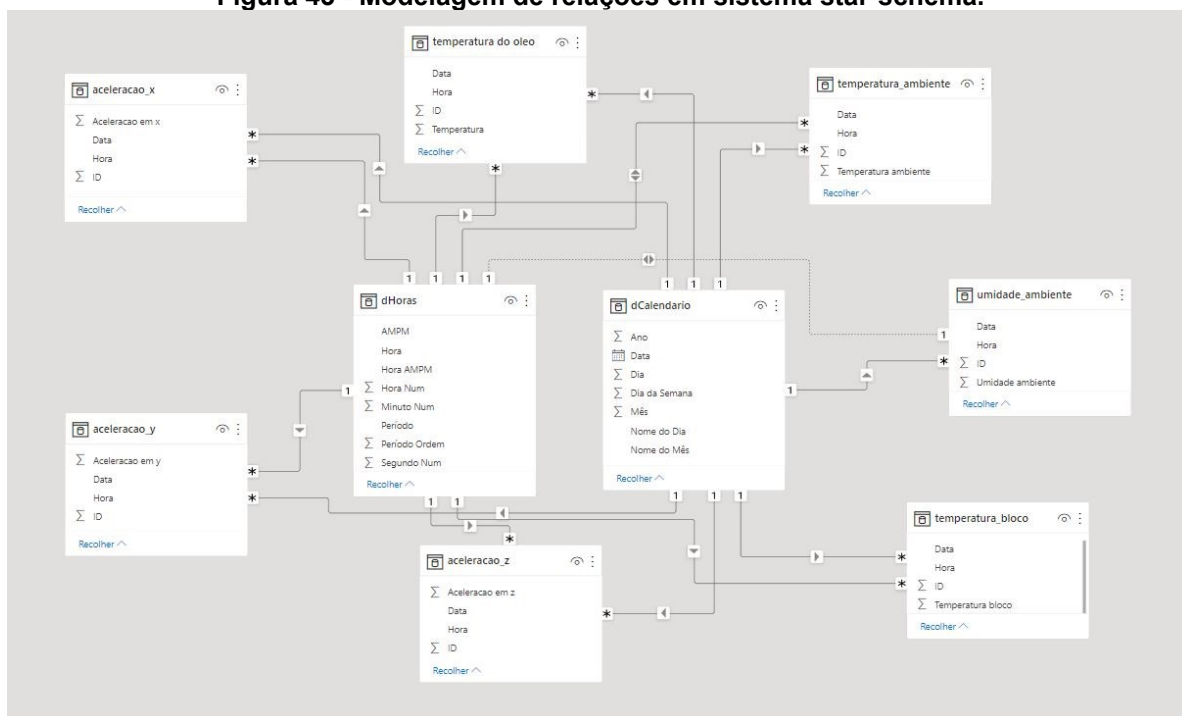
Figura 42 - Armazenamento de dados no banco MySQL.

Tabela	Acções	Registos	Tipo	Agrupamento (Collation)	Tamanho	Suspensão
<input type="checkbox"/> aceleracao_x	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	24,971	MyISAM	latin1_swedish_ci	569.0 KB	-
<input type="checkbox"/> aceleracao_y	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	24,970	MyISAM	latin1_swedish_ci	569.0 KB	-
<input type="checkbox"/> aceleracao_z	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	24,970	MyISAM	latin1_swedish_ci	569.0 KB	-
<input type="checkbox"/> Temperatura_oleo	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	2,900	MyISAM	latin1_swedish_ci	67.8 KB	13 Bytes
<input type="checkbox"/> temperatura_ambiente	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	2,867	MyISAM	latin1_swedish_ci	67.4 KB	-
<input type="checkbox"/> temperatura_bloco	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	2,901	MyISAM	latin1_swedish_ci	67.8 KB	-
<input type="checkbox"/> umidade_ambiente	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	2,867	MyISAM	latin1_swedish_ci	67.4 KB	-
7 tabelas	Soma	86,446	MyISAM	latin1_swedish_ci	1.9 MB	13 Bytes

Fonte: Autoria própria.

Para que o acompanhamento dos parâmetros pudesse ser entendido na mesma linha temporal dos acontecimentos, foi necessário criar uma relação das tabelas fato, as quais registram fatos ocorridos durante o acompanhamento, com tabelas dimensões, pautadas em data e tempo, as quais devem ter todos os dados possíveis para servirem de espelho para as tabelas fato.

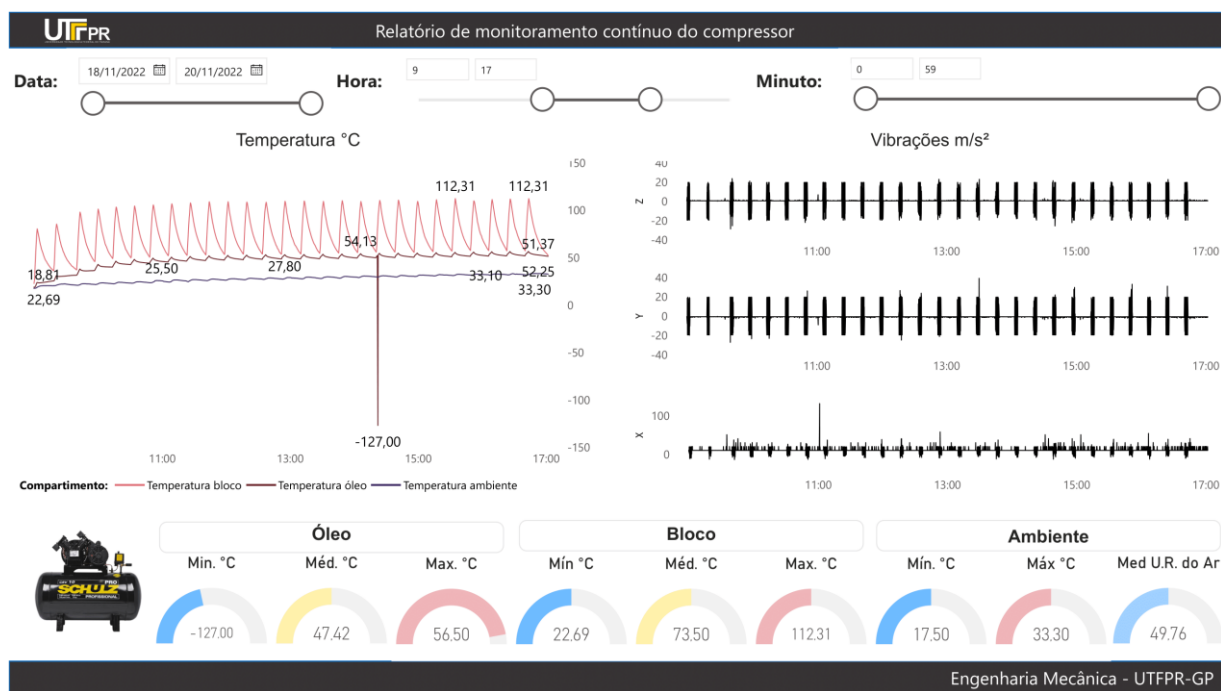
Figura 43 - Modelagem de relações em sistema star-schema.



Fonte: Autoria própria.

A partir dos passos anteriores definidos, foi possível criar o dashboard de acompanhamento do compressor durante o período citado. O sistema manteve-se em pleno funcionamento durante o experimento, registrando um único valor alterado, registrado por uma oscilação brusca no gráfico, a qual registrou -127°C , ficando evidente no dashboard de acompanhamento de temperaturas. Valor este, característico de quando ocorre a interrupção do corrente do fio vermelho VCC de alimentação do sensor.

Figura 44 - Incidência de erro de valor no dashboard.



Fonte: Autoria própria.

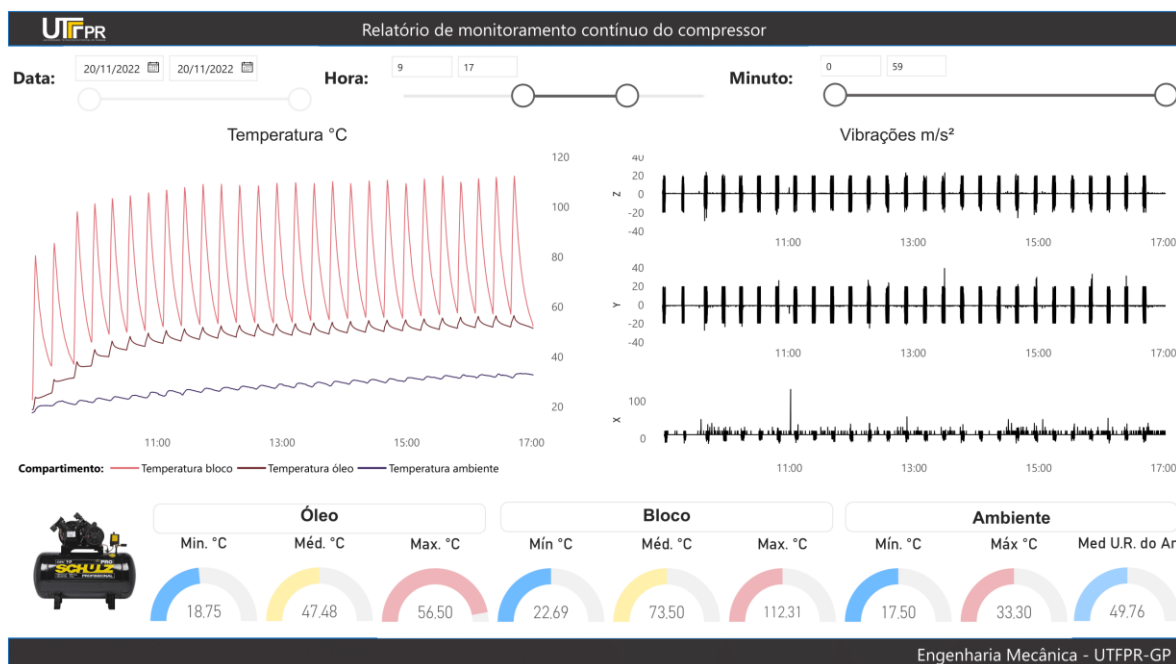
Para solução do problema do dado inconsistente, foi necessário encontrar o dado específico dentro do banco de dados na tabela que armazena os dados temperatura do óleo e apaga-lo, para corrigir o problema. O dado em questão, tratava-se da coleta de ID 1943, e com sua remoção, foi possível normalizar o sistema.

Figura 45 - Dado inconsistente no banco de dados.

<input type="checkbox"/>		Edita		Copiar		Apagar	1940	2022-11-20	14:22:06	52.13
<input type="checkbox"/>		Edita		Copiar		Apagar	1941	2022-11-20	14:22:16	52.25
<input type="checkbox"/>		Edita		Copiar		Apagar	1942	2022-11-20	14:22:26	52.44
<input type="checkbox"/>		Edita		Copiar		Apagar	1943	2022-11-20	14:22:36	-127
<input type="checkbox"/>		Edita		Copiar		Apagar	1944	2022-11-20	14:22:46	52.88
<input type="checkbox"/>		Edita		Copiar		Apagar	1945	2022-11-20	14:22:56	53.06
<input type="checkbox"/>		Edita		Copiar		Apagar	1946	2022-11-20	14:23:06	53.25

Fonte: Autoria própria.

Figura 46 - Dashboard normalizado com amplitude temporal total.

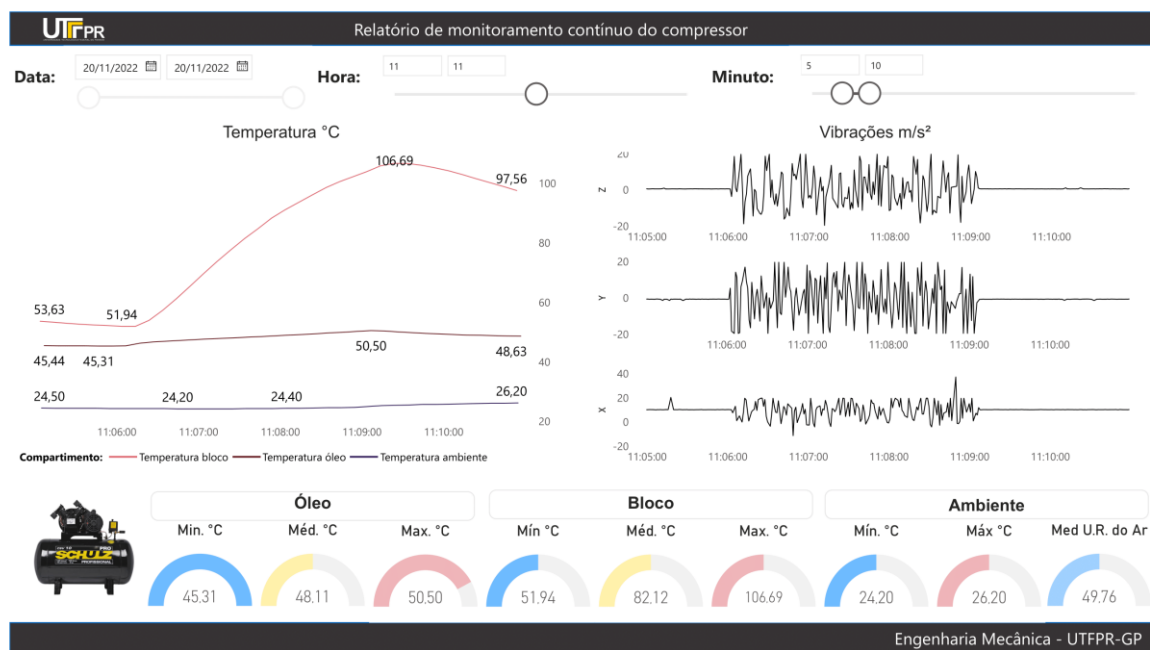


Fonte: Autoria própria.

Através do relatório do histórico de funcionamento com filtros configurados para retornar os dados do período completo do teste, é possível verificar com clareza os 27 acionamentos que ocorreram no sistema, demonstrados tanto pelas oscilações ascendentes de temperatura quanto pelos registros de vibrações, principalmente da direção Z e Y. Também é possível notar todos os valores mínimos e máximos de temperaturas apontados na parte inferior da figura, bem como valores médios de funcionamento.

Os filtros permitem que sejam mostrados com maiores detalhes momentos específicos do funcionamento na ordem de horas e minutos, possibilitando uma análise minuciosa do comportamento do equipamento durante a linha temporal de funcionamento, como por exemplo em uma das 27 partidas de modo específico.

Figura 47 - Dashboard 8º partida com amplitude temporal de 5 minutos.



Fonte: Autoria própria.

Com uma menor amplitude, é possível enxergar com mais clareza o comportamento dos parâmetros durante uma partida específica, como no 8º acionamento do compressor descrito acima. Através de uma resolução de 5 minutos já é possível notar diferenciação nos parâmetros de vibração dos 3 eixos com o comportamento já esperado, onde o posicionamento dos pistões inclinados faz com que as vibrações no eixo Z e Y sejam mais expressivas que no eixo X. Outro ponto visível é a capacidade do equipamento em gerar aquecimento no ambiente, principalmente no estágio final de compressão onde se atingem as maiores temperaturas do bloco.

6 CONCLUSÃO

No presente trabalho, demonstra-se uma pequena parcela da capacidade da tecnologia IOT, a qual está sempre ligada ao conceito de indústria 4.0, no meio fabril relacionadas ao setor de planejamento e controle da manutenção, visando diagnósticos precoces e prevenção de falhas em máquinas.

Nota-se que o protocolo MQTT pode ser considerado uma boa solução para utilização de comunicação de dados dispostos em tempo real, sendo um protocolo leve, com fácil utilização dos recursos de publicação e subscrição de tópicos, organizando e alinhando o fluxo de dados, evitando possíveis tratativas.

A placa ESP32 demonstrou-se uma opção confiável, de baixo custo, baixo consumo energético, e com capacidade de estabelecer uma conectividade estável com a internet via WIFI. Permitindo que se possa efetuar um monitoramento remoto do equipamento através da internet

O Node-RED, demonstrou-se uma ferramenta vasta de recursos para criação de pontes entre softwares. Dispondo de broker próprio, recebendo dados de forma simples, criando fluxos com nós pré-configurados, além de permitir a interação com softwares e aplicativos. O Node-RED permitiu a criação de dashboards com recursos visuais de acompanhamento em tempo real, armazenamento dos dados em um banco para posteriormente serem processados e utilizados para auxiliar a tomada de decisões mais assertivas.

O Telegram, revelou-se uma opção de simples manuseio para a criação de BOTS voltados para a inteligência artificial. O conjunto de recursos phpMyAdmin e banco de dados MySQL, foi eficaz no armazenamento de dados de forma concisa e organizada. Também possibilitou a comunicação direta dos dados com o Power BI, que por sua vez, trouxe uma boa opção para visualização de relatórios de funcionamento de máquina, podendo explorar e entender o comportamento de parâmetros em momentos específicos da linha temporal de funcionamento.

Em linhas gerais, o protótipo se demonstrou eficiente, atendendo a proposta. Foi exemplificado um estudo aplicado a uma máquina real, com testes e simulações indicando possíveis indícios de falhas, gerando alertas eficientes, guardando informações compactadas em um banco de dados robustos, gerando histórico de funcionamento com clareza e objetividade.

Para futuros estudos, poderiam ser aprimorados os recursos em desenvolvimento em linguagem C++ ou MicroPython utilizando comunicação MQTT, para reduzir a quantidade de placas embarcadas no monitoramento. Também explorar outros sensores aplicáveis a placas embarcadas, com novas interações de softwares com a plataforma Node-Red, e desenvolver sistemas de alertas mais complexos.

REFERÊNCIAS

ALMEIDA, PAULO.SAMUEL. **Manutenção Mecânica Industrial - Princípios Técnicos e Operações**. Disponível em:

<<https://integrada.minhabiblioteca.com.br/reader/books/9788536519807/pageid/16>>.

AMP, R. **Public Aladon RCM3 Introductory Course**. Disponível em:

<<https://www.maintenance.org/event/public-aladon-rcm3-introductory-course>>.

BLOCH, H. P.; GEITNER, F. K. **Compressors: How to Achieve High Reliability & Availability**. First Edition ed. [s.l.] McGraw-Hill Education, 2012.

CAPELLI, ALEXANDRE. **Automação Industrial - Controle da Movimento e Processos Contínuos**.

CARREIRA; SILVA; CANEIRA, U. C. DE G. **Manutenção–Evolução e Sua Importância**. 2010.

COPCO, A. **Compressed Air Manual**. 7th. ed.

CREMASCO, MARCO. **CREMASCO, Marco. A. Operações unitárias em sistemas particulados e fluidomecânicos e outros trabalhos**.

DE OLIVEIRA, J. C.; SANTOS, D. H.; NETO, M. P. **Chatting with Arduino platform through Telegram Bot**. 2016 IEEE International Symposium on Consumer Electronics (ISCE). **Anais...** Em: 2016 IEEE INTERNATIONAL SYMPOSIUM ON CONSUMER ELECTRONICS (ISCE). set. 2016.

DS18B20 - Programmable Resolution 1-Wire Digital Thermometer. p. 20, 2019.

ELANGOVAN, U. **Product Lifecycle Management (PLM): A Digital Journey Using Industrial Internet of Things (IIoT)**. 1. ed. [s.l.] CRC Press, 2020.

FEDOROV, D. S. et al. Using of measuring system MPU6050 for the determination of the angular velocities and linear accelerations. **Automatics & Software Enginy**, v. 11, n. 1, p. 75–80, 2015.

FIALHO, ARIVELTO. B. **Automatismos Pneumáticos - Princípios Básicos, Dimensionamentos de Componentes e Aplicações Práticas**. [s.l: s.n.].

FILHO, GUILHERME F. **Máquinas Térmicas Estáticas e Dinâmicas - Fundamentos de Termodinâmica, Características Operacionais e Aplicações**. [s.l: s.n.].

GALLO, T.; SANTOLAMAZZA, A. Industry 4.0 and human factor: How is technology changing the role of the maintenance operator? **Procedia Computer Science**, Proceedings of the 2nd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020). v. 180, p. 388–393, 1 jan. 2021.

GAY, W. DHT11 sensor. Em: **Advanced Raspberry Pi**. [s.l.] Springer, 2018. p. 399–418.

GERTZ, E.; JUSTO, P. D. **Environmental Monitoring with Arduino: Building Simple Devices to Collect Data About the World Around Us**. [s.l.] Make, 2012.

GIMENEZ, D. M. Indústria 4.0, manufatura avançada e seus impactos sobre o trabalho. n. 371, p. 29, 2019.

IBM. Using the Internet of Things for preventive maintenance. 2021.

Interfacing DS18B20 1-Wire Digital Temperature Sensor with Arduino.

Disponível em: <<https://lastminuteengineers.com/ds18b20-arduino-tutorial/>>. Acesso em: 26 nov. 2021.

JAVOID, M. et al. Upgrading the manufacturing sector via applications of Industrial Internet of Things (IIoT). **Sensors International**, v. 2, p. 100129, 1 jan. 2021.

KARDEC; NASCIF. . **Manutenção: função estratégica**. 3° ed. [s.l: s.n.].

KONDAVEETI, H. K. et al. A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. **Computer Science Review**, v. 40, p. 100364, 1 maio 2021.

LEKIĆ, M.; GARDAŠEVIĆ, G. **IoT sensor integration to Node-RED platform**. 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH). **Anais...** Em: 2018

17TH INTERNATIONAL SYMPOSIUM INFOTEH-JAHORINA (INFOTEH). mar. 2018.

LU, Y.; WITHERELL, P.; JONES, A. Standard connections for IIoT empowered smart manufacturing. **Manufacturing Letters**, v. 26, p. 17–20, 1 out. 2020.

MARTINAZZO, C. A.; ORLANDO, T.; DA URI, E. **Comparação entre três tipos de sensores de temperatura em associação com Arduino**. PERSPECTIVA, Erechim, , 2016.

MIHART. **O que é Power BI? - Power BI**. Disponível em: <<https://learn.microsoft.com/pt-pt/power-bi/fundamentals/power-bi-overview>>. Acesso em: 5 nov. 2022.

MITRA, R.; GANIGA, R. A novel approach to sensor implementation for healthcare systems using internet of things. **International Journal of Electrical and Computer Engineering (IJECE)**, v. 9, p. 5031, 1 dez. 2019.

NEHLER, T. Linking energy efficiency measures in industrial compressed air systems with non-energy benefits – A review. **Renewable and Sustainable Energy Reviews**, v. 89, p. 72–87, 1 jun. 2018.

ÖZGÜR-ÜNLÜAKIN, D.; TÜRKALI, B.; AKSEZER, S. Ç. Cost-effective fault diagnosis of a multi-component dynamic system under corrective maintenance. **Applied Soft Computing**, v. 102, p. 107092, 1 abr. 2021.

PROCONCEPT. **Indústria 4.0: A Quarta Revolução Industrial**. ProConcept **Sistemas**, 2019. Disponível em: <<https://www.proconcept.com.br/2019/03/13/o-que-e-industria-4-0/>>. Acesso em: 25 nov. 2021

SACOMANO, JOSÉ. B. **Indústria 4.0 : conceitos e fundamentos**.

SAHBA, R. et al. Development of Industry 4.0 predictive maintenance architecture for broadcasting chain. **Advanced Engineering Informatics**, v. 49, p. 101324, 1 ago. 2021.

TOMAZINI, DANIEL, E PEDRO. **Sensores industriais**.

WEIJIS, JULIA. **7 Practical Applications of IIoT in Industrial Automation.**

Disponível em: <<https://www.ixon.cloud/knowledge-hub/7-practical-applications-of-iiot-in-industrial-automation>>. Acesso em: 23 nov. 2021.

WERLANG, W. **Pro CSV 10/110. Schulz**, [s.d.]. Disponível em:

<<https://www.schulz.com.br/produto/pro-csv-10-110/>>. Acesso em: 9 nov. 2022

WISSKIRCHEN, G. et al. Artificial intelligence and robotics and their impact on the workplace. **IBA Global Employment Institute**, v. 11, n. 5, p. 49–67, 2017.

APÊNDICE A – CÓDIGOS

CÓDIGO DHT11

```
#include "DHT.h"
#include <WiFi.h>
extern "C" {
    #include "freertos/FreeRTOS.h"
    #include "freertos/timers.h"
}
#include <AsyncMqttClient.h>
#define WIFI_SSID "ZazzInternet-782B"
#define WIFI_PASSWORD "87448863"

// Raspberry Pi Mosquitto MQTT Broker
#define MQTT_HOST IPAddress(192, 168, 18, 20)
// For a cloud MQTT broker, type the domain name
// #define MQTT_HOST "example.com"
#define MQTT_PORT 1883

// Temperature MQTT Topics
#define MQTT_PUB_TEMP "esp32/dht/temperature"
#define MQTT_PUB_HUM "esp32/dht/humidity"

// Digital pin connected to the DHT sensor
#define DHTPIN 4

// Uncomment whatever DHT sensor type you're using
#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21 // DHT 21 (AM2301)

// Initialize DHT sensor
DHT dht(DHTPIN, DHTTYPE);
```

```
// Variables to hold sensor readings
float temp;
float hum;

AsyncMqttClient mqttClient;
TimerHandle_t mqttReconnectTimer;
TimerHandle_t wifiReconnectTimer;

unsigned long previousMillis = 0; // Stores last time temperature was published
const long interval = 10000;     // Interval at which to publish sensor readings

void connectToWifi() {
  Serial.println("Connecting to Wi-Fi...");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
}

void connectToMqtt() {
  Serial.println("Connecting to MQTT...");
  mqttClient.connect();
}

void WiFiEvent(WiFiEvent_t event) {
  Serial.printf("[WiFi-event] event: %d\n", event);
  switch(event) {
    case SYSTEM_EVENT_STA_GOT_IP:
      Serial.println("WiFi connected");
      Serial.println("IP address: ");
      Serial.println(WiFi.localIP());
      connectToMqtt();
      break;
    case SYSTEM_EVENT_STA_DISCONNECTED:
      Serial.println("WiFi lost connection");
```

```

    xTimerStop(mqttReconnectTimer, 0); // ensure we don't reconnect to MQTT while
reconnecting to Wi-Fi
    xTimerStart(wifiReconnectTimer, 0);
    break;
}
}

void onMqttConnect(bool sessionPresent) {
    Serial.println("Connected to MQTT.");
    Serial.print("Session present: ");
    Serial.println(sessionPresent);
}

void onMqttDisconnect(AsyncMqttClientDisconnectReason reason) {
    Serial.println("Disconnected from MQTT.");
    if (WiFi.isConnected()) {
        xTimerStart(mqttReconnectTimer, 0);
    }
}

/*void onMqttSubscribe(uint16_t packetId, uint8_t qos) {
    Serial.println("Subscribe acknowledged.");
    Serial.print(" packetId: ");
    Serial.println(packetId);
    Serial.print(" qos: ");
    Serial.println(qos);
}

void onMqttUnsubscribe(uint16_t packetId) {
    Serial.println("Unsubscribe acknowledged.");
    Serial.print(" packetId: ");
    Serial.println(packetId);
}*/

void onMqttPublish(uint16_t packetId) {

```

```

Serial.print("Publish acknowledged.");
Serial.print(" packetId: ");
Serial.println(packetId);
}

void setup() {
  Serial.begin(115200);
  Serial.println();

  dht.begin();

  mqttReconnectTimer = xTimerCreate("mqttTimer", pdMS_TO_TICKS(2000),
pdFALSE, (void*)0, reinterpret_cast<TimerCallbackFunction_t>(connectToMqtt));
  wifiReconnectTimer = xTimerCreate("wifiTimer", pdMS_TO_TICKS(2000),
pdFALSE, (void*)0, reinterpret_cast<TimerCallbackFunction_t>(connectToWifi));

  WiFi.onEvent(WiFiEvent);

  mqttClient.onConnect(onMqttConnect);
  mqttClient.onDisconnect(onMqttDisconnect);
  //mqttClient.onSubscribe(onMqttSubscribe);
  //mqttClient.onUnsubscribe(onMqttUnsubscribe);
  mqttClient.onPublish(onMqttPublish);
  mqttClient.setServer(MQTT_HOST, MQTT_PORT);
  // If your broker requires authentication (username and password), set them below
  //mqttClient.setCredentials("REPLACE_WITH_YOUR_USER",
"REPLACE_WITH_YOUR_PASSWORD");
  connectToWifi();
}

void loop() {
  unsigned long currentMillis = millis();
  // Every X number of seconds (interval = 10 seconds)
  // it publishes a new MQTT message

```

```

if (currentMillis - previousMillis >= interval) {
  // Save the last time a new reading was published
  previousMillis = currentMillis;
  // New DHT sensor readings
  hum = dht.readHumidity();
  // Read temperature as Celsius (the default)
  temp = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  //temp = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(temp) || isnan(hum)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  // Publish an MQTT message on topic esp32/dht/temperature
  uint16_t packetIdPub1 = mqttClient.publish(MQTT_PUB_TEMP, 1, true,
String(temp).c_str());
  Serial.printf("Publishing on topic %s at QoS 1, packetId: %i", MQTT_PUB_TEMP,
packetIdPub1);
  Serial.printf("Message: %.2f \n", temp);

  // Publish an MQTT message on topic esp32/dht/humidity
  uint16_t packetIdPub2 = mqttClient.publish(MQTT_PUB_HUM, 1, true,
String(hum).c_str());
  Serial.printf("Publishing on topic %s at QoS 1, packetId %i: ", MQTT_PUB_HUM,
packetIdPub2);
  Serial.printf("Message: %.2f \n", hum);
}
}

```

CODIGO DS18b20

```

#include <WiFi.h>
extern "C" {
    #include "freertos/FreeRTOS.h"
    #include "freertos/timers.h"
}
#include <AsyncMqttClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define WIFI_SSID "ZazzInternet-782B"
#define WIFI_PASSWORD "87448863"

// Raspberry Pi Mosquitto MQTT Broker
#define MQTT_HOST IPAddress(192, 168, 18, 20)
// For a cloud MQTT broker, type the domain name
// #define MQTT_HOST "example.com"
#define MQTT_PORT 1883

// Temperature MQTT Topic
#define MQTT_PUB_TEMP "esp32/ds18b20/cabecote"

// GPIO where the DS18B20 is connected to
const int oneWireBus = 4;
// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(oneWireBus);
// Pass our oneWire reference to Dallas Temperature sensor
DallasTemperature sensors(&oneWire);
// Temperature value
float temp;

AsyncMqttClient mqttClient;
TimerHandle_t mqttReconnectTimer;

```

```

TimerHandle_t wifiReconnectTimer;

unsigned long previousMillis = 0; // Stores last time temperature was published
const long interval = 10000;     // Interval at which to publish sensor readings

void connectToWifi() {
  Serial.println("Connecting to Wi-Fi...");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
}

void connectToMqtt() {
  Serial.println("Connecting to MQTT...");
  mqttClient.connect();
}

void WiFiEvent(WiFiEvent_t event) {
  Serial.printf("[WiFi-event] event: %d\n", event);
  switch(event) {
    case SYSTEM_EVENT_STA_GOT_IP:
      Serial.println("WiFi connected");
      Serial.println("IP address: ");
      Serial.println(WiFi.localIP());
      connectToMqtt();
      break;
    case SYSTEM_EVENT_STA_DISCONNECTED:
      Serial.println("WiFi lost connection");
      xTimerStop(mqttReconnectTimer, 0); // ensure we don't reconnect to MQTT while
reconnecting to Wi-Fi
      xTimerStart(wifiReconnectTimer, 0);
      break;
  }
}

void onMqttConnect(bool sessionPresent) {

```



```
Serial.println("Connected to MQTT.");
Serial.print("Session present: ");
Serial.println(sessionPresent);
}

void onMqttDisconnect(AsyncMqttClientDisconnectReason reason) {
  Serial.println("Disconnected from MQTT.");
  if (WiFi.isConnected()) {
    xTimerStart(mqttReconnectTimer, 0);
  }
}

/*void onMqttSubscribe(uint16_t packetId, uint8_t qos) {
  Serial.println("Subscribe acknowledged.");
  Serial.print(" packetId: ");
  Serial.println(packetId);
  Serial.print(" qos: ");
  Serial.println(qos);
}

void onMqttUnsubscribe(uint16_t packetId) {
  Serial.println("Unsubscribe acknowledged.");
  Serial.print(" packetId: ");
  Serial.println(packetId);
}*/

void onMqttPublish(uint16_t packetId) {
  Serial.println("Publish acknowledged.");
  Serial.print(" packetId: ");
  Serial.println(packetId);
}

void setup() {
  // Start the DS18B20 sensor
  sensors.begin();
```

```

Serial.begin(115200);
Serial.println();
Serial.println();

mqttReconnectTimer = xTimerCreate("mqttTimer", pdMS_TO_TICKS(2000),
pdFALSE, (void*)0, reinterpret_cast<TimerCallbackFunction_t>(connectToMqtt));
wifiReconnectTimer = xTimerCreate("wifiTimer", pdMS_TO_TICKS(2000),
pdFALSE, (void*)0, reinterpret_cast<TimerCallbackFunction_t>(connectToWifi));

WiFi.onEvent(WiFiEvent);

mqttClient.onConnect(onMqttConnect);
mqttClient.onDisconnect(onMqttDisconnect);
//mqttClient.onSubscribe(onMqttSubscribe);
//mqttClient.onUnsubscribe(onMqttUnsubscribe);
mqttClient.onPublish(onMqttPublish);
mqttClient.setServer(MQTT_HOST, MQTT_PORT);
// If your broker requires authentication (username and password), set them below
//mqttClient.setCredentials("REPIACE_WITH_YOUR_USER",
"REPLACE_WITH_YOUR_PASSWORD");
connectToWifi();
}

void loop() {
  unsigned long currentMillis = millis();
  // Every X number of seconds (interval = 10 seconds)
  // it publishes a new MQTT message
  if (currentMillis - previousMillis >= interval) {
    // Save the last time a new reading was published
    previousMillis = currentMillis;
    // New temperature readings
    sensors.requestTemperatures();
    // Temperature in Celsius degrees

```

```

temp = sensors.getTempCByIndex(0);
// Temperature in Fahrenheit degrees
//temp = sensors.getTempFByIndex(0);

// Publish an MQTT message on topic esp32/ds18b20/temperature
uint16_t packetIdPub1 = mqttClient.publish(MQTT_PUB_TEMP, 1, true,
String(temp).c_str());
Serial.printf("Publishing on topic %s at QoS 1, packetId: ", MQTT_PUB_TEMP);
Serial.println(packetIdPub1);
Serial.printf("Message: %.2f /n", sensors.getTempCByIndex(0));
}
}

```

CODIGO MPU6050

```

#include "Wire.h"
#include "math.h"
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <WiFi.h>
/*-----MPU 6050-----*/
// La dirección del MPU6050 puede ser 0x68 o 0x69, dependiendo
// del estado de AD0. Si no se especifica, 0x68 estará implícito
//MPU6050 sensor_ang;
// Valores RAW (sin procesar) del acelerómetro y giroscopio en los ejes x,y,z
#define SDA 4
#define SCL 15
const int MPU_addr=0x68; // I2C address of the MPU-6050

int16_t ax, ay, az;
int16_t gx, gy, gz, Tmp;
double ax_d, ay_d, az_d;
double ang_x, ang_y;
float ang_x_prev, ang_y_prev;

```

```

long tiempo_prev;
float dt;
int d_max;
float accel_ang_y, accel_ang_x;
#define WLAN_SSID "ZazzInternet-782B"
#define WLAN_PASS "87448863"

//MQTT broker settings https://proyectoinfo.mybluemix.net
#define HOST      "192.168.18.20"
#define PORT      1883
#define USERNAME  "linux"
#define PASSWORD  "linux"

//time out loop count
const int timeout = 200;
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, HOST, PORT, USERNAME, PASSWORD);
Adafruit_MQTT_Publish  accelera_x    =  Adafruit_MQTT_Publish(&mqtt,
"acelerometro/ax");
Adafruit_MQTT_Publish  accelera_y    =  Adafruit_MQTT_Publish(&mqtt,
"acelerometro/ay");
Adafruit_MQTT_Publish  accelera_z    =  Adafruit_MQTT_Publish(&mqtt,
"acelerometro/az");
void MQTT_connect();
void init_MPU_sensor();
void MPU_sensor();
void setup() {
  delay(1);
  WiFi.mode(WIFI_STA);
  Serial.begin(115200);
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);

  WiFi.begin(WLAN_SSID, WLAN_PASS);

```

```

int i = 0;
for (; i < timeout; i++)
{
  if(WiFi.status() == WL_CONNECTED) break;
  delay(100);
  Serial.print(".");
}
if(i == timeout)
  Serial.println("No Conectado");

Serial.println("IP address: "); Serial.println(WiFi.localIP());

MQTT_connect();
init_MPU_sensor();
}
void loop()
{
  delay(150);/*
  int x = random(-10,10);
  int y = random(-10,10);
  int z = random(-10,10);*/

  accelera_x.publish(ax_d);
  delay(500);
  accelera_y.publish(ay_d);
  delay(500);
  accelera_z.publish(az_d);

  Serial.print(ax_d);                               Serial.print("\t");Serial.print(ay_d);
  Serial.print("\t");Serial.print(az_d); Serial.println("\t");
  MPU_sensor();

}
void MQTT_connect() {

```

```

int8_t ret;
if (mqtt.connected()) {
    return;
}

Serial.print("Connecting to MQTT... ");
uint8_t retries = 10;
while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 1 second...");
    mqtt.disconnect();
    delay(1000);
    retries--;
    if (retries == 0)
        Serial.println("No Conectado");
}
Serial.println("MQTT Connected!");
}

void init_MPU_sensor(){
    Wire.begin(SDA, SCL);
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x6B); // PWR_MGMT_1 register
    Wire.write(0); // set to zero (wakes up the MPU-6050)
    Wire.endTransmission(true);
}

void MPU_sensor(){
    /* Leer las aceleraciones y velocidades angulares
    sensor_ang.getAcceleration(&ax, &ay, &az);
    sensor_ang.getRotation(&gx, &gy, &gz);
    // Correccion de Offsets*/
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers

```

```

    ax=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C
    (ACCEL_XOUT_L)
    ay=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E
    (ACCEL_YOUT_L)
    az=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40
    (ACCEL_ZOUT_L)
    Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
    gx=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
    gy=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
    gz=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)

    az_d = az * (9.81/16384.0);
    ay_d = ay * (9.81/16384.0);
    ax_d = ax * (9.81/16384.0);
}

```

CODIGO DAX DIMENSÃO HORA

```

let
    Lista = {0..86399},
    ConverteParaTabela = Table.FromList(Lista, Splitter.SplitByNothing(), null, null,
    ExtraValues.Error),
    AlteraTipo = Table.TransformColumnTypes(ConverteParaTabela,{{"Column1",
    Int64.Type}}),
    AlteraNomeCol = Table.RenameColumns(AlteraTipo,{{"Column1", "Segundo"}}),
    ColDuracao = Table.AddColumn(AlteraNomeCol,"Duração", each
    #duration(0,0,0,[Segundo]), type duration),
    ColHora = Table.AddColumn(ColDuracao, "Hora", each
    #time(Duration.Hours([Duração]),Duration.Minutes([Duração]),Duration.Seconds([Dur
    ação])), type time),
    RemoveCols = Table.RemoveColumns(ColHora,{"Duração", "Segundo"}),
    ColHoraNum = Table.AddColumn(RemoveCols, "Hora Num", each
    Time.Hour([Hora]), Int64.Type),

```

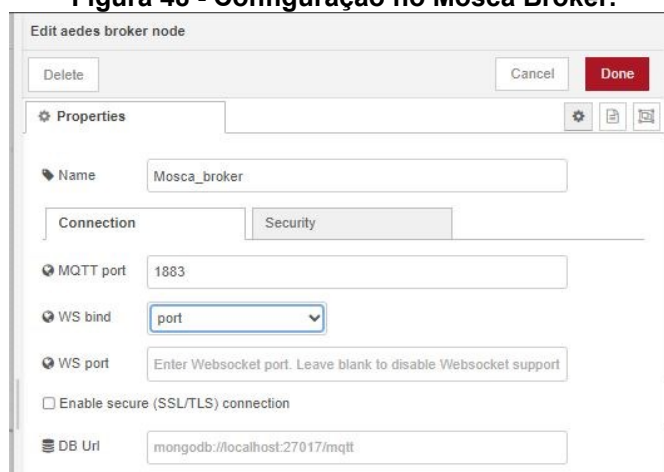
```

ColMinutoNum = Table.AddColumn(ColHoraNum, "Minuto Num", each
Time.Minute([Hora]), Int64.Type),
ColSegundoNum = Table.AddColumn(ColMinutoNum, "Segundo Num", each
Time.Second([Hora]), Int64.Type),
ColAMPM = Table.AddColumn(ColSegundoNum,"AMPM", each if [Hora Num] < 12
then "AM" else "PM", type text),
ColHoraAMPM = Table.AddColumn(ColAMPM, "Hora AMPM", each
if [Hora Num] = 0 then "12 AM"
else if [Hora Num] >= 1 and [Hora Num] <= 11 then Number.ToText([Hora Num])
& " AM"
else if [Hora Num] = 12 then "12 PM"
else if [Hora Num] >= 13 then Number.ToText([Hora Num] - 12) & " PM"
else "Unknown", type text),
ColPeriodo =
Table.AddColumn(ColHoraAMPM, "Período", each
if [Hora Num] >= 0 and [Hora Num] <= 5 then "12AM to 6AM"
else if [Hora Num] >= 6 and [Hora Num] <= 11 then "6AM to 12PM"
else if [Hora Num] >= 12 and [Hora Num] <= 17 then "12PM to 6PM"
else if [Hora Num] >= 18 and [Hora Num] <= 23 then "6PM to 12AM"
else "Unknown", type text
),
ColPeriodoOrdem =
Table.AddColumn(ColPeriodo, "Período Ordem", each
if [Hora Num] >= 0 and [Hora Num] <= 5 then 1
else if [Hora Num] >= 6 and [Hora Num] <= 11 then 2
else if [Hora Num] >= 12 and [Hora Num] <= 17 then 3
else if [Hora Num] >= 18 and [Hora Num] <= 23 then 4
else 5, Int64.Type
)
in
ColPeriodoOrdem

```


APÊNDICE B - CONFIGURAÇÕES

Figura 48 - Configuração nó Mosca Broker.



Edit aedes broker node

Delete Cancel Done

Properties

Name Mosca_broker

Connection Security

MQTT port 1883

WS bind port

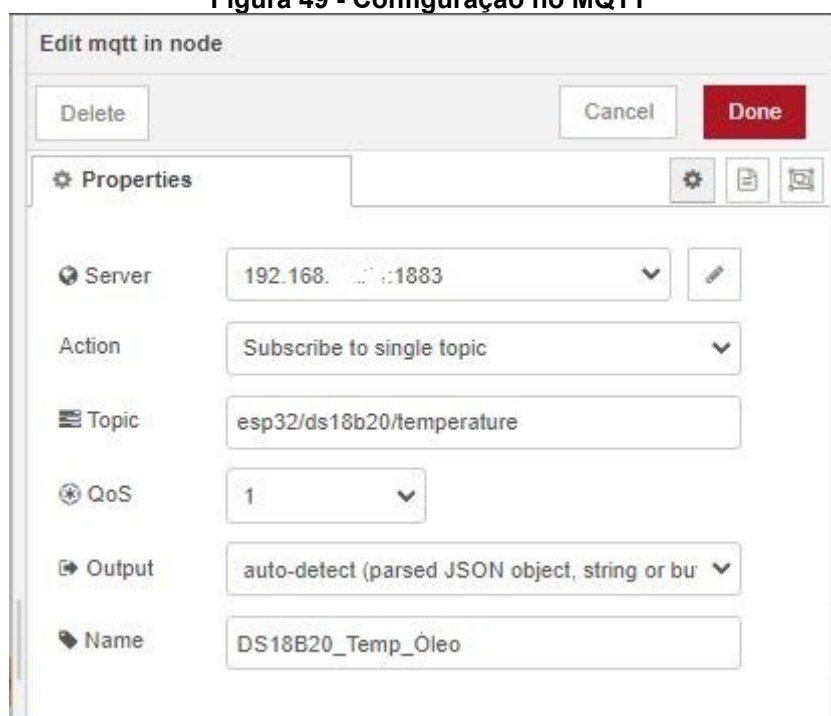
WS port Enter Websocket port. Leave blank to disable Websocket support

Enable secure (SSL/TLS) connection

DB Uri mongodb://localhost:27017/mqtt

Fonte: Autoria própria.

Figura 49 - Configuração nó MQTT



Edit mqtt in node

Delete Cancel Done

Properties

Server 192.168. ... :1883

Action Subscribe to single topic

Topic esp32/ds18b20/temperature

QoS 1

Output auto-detect (parsed JSON object, string or bu

Name DS18B20_Temp_Óleo

Fonte: Autoria própria.

Figura 50 - Configuração nó Dashboard gauge

Edit gauge node
 Delete Cancel Done
 Properties
 Group [Monitoramento do Compressor] Temp
 Size auto
 Type Gauge
 Label Temperatura Óleo
 Value format {{value}}
 Units °C
 Range min 0 max 100
 Colour gradient
 Sectors 0 ... optional ... optional ... 100
 Class Optional CSS class name(s) for widget
 Name Monitor de temperatura

Fonte: Autoria própria.

Figura 51 - Configuração nó Dashboard chart

Edit chart node
 Delete Cancel Done
 Properties
 Group [Monitoramento do Compressor] Ter
 Size 6 x 6
 Label Comportamento Temperatura Oleo
 Type Line chart enlarge points
 X-axis last 1 hours OR 1000 points
 X-axis Label HH:mm:ss as UTC
 Y-axis min 0 max 100
 Legend None Interpolate linear
 Series Colours
 Blank label display this text before valid data arrives
 Class Optional CSS class name(s) for widget
 Name Comportamento Temperatura Oleo
 Enabled

Fonte: Autoria própria.

Figura 52 - Configuração nó String

Delete Cancel Done

Properties

Name Umidade superior a 80%

From msg. payload

Methods

append % de umidade relativa do ar

+ add

To msg. payload

Fonte: Autoria própria.

Figura 53 - Configuração nó Switch

Delete Cancel Done

Properties

Name

Property msg. payload

> 40 → 1

is between 10 and 40 → 2

is between 10 and 0 → 3

< 0 → 4

+ add

checking all rules

recreate message sequences

Fonte: Autoria própria.

Figura 54 - Configuração nó MySQL

Edit mysql node > Edit MySQLdatabase node

Delete Cancel Update

Properties

Host: 127.0.0.1

Port: 3306

User: bruno.oliveira

Password:

Database: banco de dados compressor

Timezone: ±hh:mm

Charset: UTF8

Name: Banco de dados

Tip: The timezone should be specified as ±hh:mm or leave blank for 'local'.

Fonte: (Autoria própria)

Figura 55 - Configuração nó endereçamento do banco MySQL

Edit function node > JavaScript editor

Cancel Done

```

1 var temp = msg.payload;
2
3
4 msg.topic = "INSERT INTO `temperatura do oleo`(`Temperatura`) VALUES (" + temp + ")";
5
6 return msg;

```

Fonte: Autoria própria.

Figura 56 - Configuração nó Telegram

Edit function node

Delete Cancel Done

Properties

Name: Endereçamento telegram

Setup On Start On Message On Stop

```

1 let texto = msg.payload
2 msg.payload = {};
3 msg.payload.chatId = -100161509;
4 msg.payload.type = 'message'
5 msg.payload.content = texto
6 return msg;

```

Fonte: Autoria própria.