Universidade de Brasília
Faculdade de Tecnologia

Sistema de Internet das Coisas para Monitoração de Motores Elétricos Industriais

Alexandre Abrahami Pinto da Cunha

PROJETO FINAL DE CURSO

ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília

2023

Universidade de Brasília
Faculdade de Tecnologia

Sistema de Internet das Coisas para Monitoração de Motores Elétricos Industriais

Alexandre Abrahami Pinto da Cunha

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação

Orientador: Prof. Dr. Jones Yudi Mori Alves da Silva

Brasília
2023

Abrahami Pinto da Cunha, Alexandre.
A159s Sistema de Internet das Coisas para Monitoração de Motores Elétricos Industriais / Alexandre Abrahami Pinto da Cunha; orientador Jones Yudi Mori Alves da Silva. –– Brasília, 2023.
85 p.

Projeto Final de Curso (Engenharia de Controle e Automação) –– Universidade de Brasília, 2023.

1. Gêmeos Digitais. 2. Motores Elétricos. 3. Manutenção Preditiva. 4. Sistemas SCADA. I. Mori Alves da Silva, Jones Yudi, orient. II. Título

Universidade de Brasília
Faculdade de Tecnologia

Sistema de Internet das Coisas para Monitoração de Motores Elétricos Industriais

Alexandre Abrahami Pinto da Cunha

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação

Trabalho aprovado. Brasília, 20 de julho de 2023:

Prof. Dr. Jones Yudi Mori Alves da Silva, UnB/FT/ENM
Orientador

Prof. Dr. Gerson Henrique Pfitscher, UnB/FT/ENE
Examinador interno

Prof. Dr. Carlos Humberto Llanos Quintero, UnB/FT/ENM
Examinador interno

Brasília
2023

Este trabalho é dedicado à minha família, aos meus amigos e meus colegas de faculdade que sempre me apoiaram no sofrimento que foi este curso. Dedico exclusivamente ao Marcos Eduardo por ter me guiado em 50% das matérias.

Agradecimentos

Honestamente, eu não sei como eu consegui chegar tão longe, e com certeza não seria sem o apoio das pessoas que eu amo. Primeiramente agradeço à minha família, meus pais, minha irmã e a Tilde por sempre me incentivarem, me amarem e estarem comigo no dia-a-dia. À minha tia Dahlia, ao Kiko e à minha avó Mimi, que faleceu em 2021, pelo amor incondicional enviado sobre águas estrangeiras. Tenho certeza que onde quer que minha avó esteja, ela deve estar orgulhosa. À minha avó Acileia e à Maria por fazerem eu ter uma segunda casa no RJ, um lugar de amor, conforto e muita comida boa. À minha tia Rosane e meus primos Guilherme, Isabela e Gabriela pelo apoio e amor. E ao meu avó Aldo, que faleceu em 2014, que me inspirou a ser uma pessoa trabalhadora e sei que deve ter me acompanhado de onde quer que esteja.

Gostaria de agradecer imensamente aos meu amigos e colegas de curso David Fanchic Chatelard, Luís Humberto, Gabriel Tambara, Marcos Eduardo e Emanuel Couto Brenag pelo suporte absurdo que me deram ao longo destes últimos anos e por fazer da faculdade um ambiente infinitamente mais suportável. É impossível descrever o quão importante vocês são para mim, mas acredito que consigo resumir em algumas palavras: uh uh ah ah. Não menos importante, gostaria de agradecer ao professor Jones e a todos os demais amigos e colegas de curso que fiz ao longo desses anos, como diria o Vicente: "O verdadeiro curso são os amigos que a gente faz ao longo do caminho". O sentimento de insanidade, desespero e personalidade degenerativa nos uniu e fez com que nossas conexões fossem forjadas a ferro e fogo. Foi uma verdadeira experiência de um manicômio e eu não teria conseguido sem os loucos: Liz, Fernanda, Thiago, Valente, Isabela, Vicente, Lucas, Robson, Montanha, Richard, Rebeschinni, Arthur, Vigna, Adriano, Maria, Luan, Eduardo, Nikolas, Sabrina, Sarah, Lemos, Renatinho, Anderson, Alexandre, Judeu, Fernandes, Rodrigo e todos os muitos outros aqui não citados.

Por fim, gostaria de agradecer aos meus amigos além da faculdade que sempre me apoiaram, me amaram e estiveram comigo nos momentos bons e nos momentos ruins. Ao Valter, Felipe, João, Lucas e Gabriel que me fizeram rir muito ao longo desses anos e estiveram comigo em todo tipo de ocasião. À Laurinha que ouviu incontáveis vezes eu reclamando da faculdade e sempre me deu suporte nos meus surtos. À Brena por todo o apoio no período em que namoramos e posteriormente como uma excelente amiga/parceira de crossfit/salsicha psicóloga. Igualmente à Giovanna pelo apoio e por ter me ensinado a apreciar um solzinho e uma cafeteria. Ao Pedim, Bea, Rafa, Jão, Helena, JP, JV, Janderson, Marco, Titi e Bruno por simplesmente existirem e eu amar vocês. A todas estas pessoas e a todos aqueles que me apoiaram e auxiliaram por este caminho árduo que foi minha graduação só posso lhes dizer com toda a sinceridade e gratidão: Muito obrigado.

Alexandre Abrahami Pinto da Cunha

“Eu procurava homens grandes,
nunca encontrei mais do que
«macacos de imitação»
do seu próprio ideal.”
(Friedrich Nietzsche)

Resumo

O presente artigo tem como objetivo apresentar o estudo realizado nas dependências do GRACO (Grupo de Automação e Controle) da Universidade de Brasília, explorando a tecnologia de Gêmeos Digitais e como ela pode ser aplicada para melhorar a gestão de motores elétricos, incluindo a monitorização em tempo real de suas condições de operação, a previsão de falhas e a otimização do desempenho. Uma breve contextualização do assunto é realizada, apresentando as tendências da Indústria 4.0, a importância da manutenção preditiva e o papel dos motores elétricos no cenário atual. Somado a isso, os conceitos teóricos e técnicos relacionados aos principais temas deste projeto são detalhados de forma a esclarecer qual a proposta da tecnologia aplicada. Em ambos, alguns trabalhos relacionados são citados a fim de destacar a importância do assunto no meio acadêmico e o crescente desenvolvimento científico na área. Partindo de um contexto inexistente, ou seja um trabalho totalmente novo, a construção do Gêmeo Digital envolveu definir os materiais necessários, fossem eles físicos ou virtuais, desenvolver um sistema para a obtenção de dados e outro para a monitoração e controle de motores elétricos. Os materiais utilizados foram o inversor de frequência WEG CFW-08, o microcontrolador ESP32, o sensor de temperatura DS18B20, o sensor de som KY-038, o sensor acelerômetro e giroscópio MPU-6050 e um conversor USB/Serial RS485. Em conjunto, os equipamentos se demonstraram adequados para o seu propósito de obter informações. Aliado a eles, os softwares Arduino IDE e ScadaBR foram utilizados para a programação dos dispositivos e o desenvolvimento de um sistema SCADA responsável pelo papel de monitorar e controlar os dados. A base do modelo digital composta por estes sistemas foi integrada à um trabalho de mestrado para verticalizar a arquitetura do Gêmeo Digital, trazendo inteligência e acesso à nuvem para o trabalho desenvolvido. Ao final, realizou-se a verificação dos objetivos alcançados e formulação de trabalhos futuros.

Palavras-chave: Gêmeos Digitais. Motores Elétricos. Manutenção Preditiva. Sistemas SCADA.

Abstract

This article aims to present the study carried out at GRACO (Automation and Control Group) at the University of Brasilia, exploring the Digital Twins technology and how it can be applied to improve the management of electric motors, including monitoring in real-time of your operating conditions, failure prediction and performance optimization. A brief contextualization of the subject is carried out, presenting the trends of Industry 4.0, the importance of predictive maintenance and the role of electric motors in the current scenario. Added to this, the theoretical and technical concepts related to the main themes of this project are detailed in order to clarify the purpose of the applied technology. In both, some related works are cited in order to highlight the importance of the subject in academia and the growing scientific development in the area. Starting from a non-existent context, in other words, a completely new job, the construction of the Digital Twin involved defining the necessary materials, whether physical or virtual, developing a system for obtaining data from electric motors and another for monitoring and controlling them. The materials used were the WEG CFW-08 frequency inverter, the ESP32 microcontroller, the DS18B20 temperature sensor, the KY-038 sound sensor, the MPU-6050 accelerometer and gyroscope sensor and a USB/Serial RS485 converter. Together, the equipments proved to be adequate for its purpose of obtaining information. Allied to them, the Arduino IDE and ScadaBR software were used for the programming of the devices and the development of a SCADA system responsible for the role of monitoring and controlling the data. The basis of the digital model composed of these systems was integrated into a master's work to verticalize the architecture of the Digital Twin, bringing intelligence and access to the cloud to the work developed. At the end, there was a verification of the achieved objectives and formulation of future works.

Keywords: Digital Twins. Eletric Motors. Predictive Maintenance. SCADA systems.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

1.1 Contextualização

A inovação tecnológica é uma das principais forças motrizes do progresso econômico e social. O mercado de trabalho procura se reinventar diariamente em busca de novas soluções para um mercado mais produtivo e lucrativo, o que resulta em profundas transformações na sociedade, nas indústrias e na forma como nos relacionamos com as atividades do dia-a-dia. Partindo do ponto de vista de que o avanço da tecnologia é inevitável, as empresas precisam inovar para proteger sua posição e ganhar mais espaço. Do contrário, elas são superadas por concorrentes com produtos melhores e processos mais eficientes. O mercado é altamente competitivo e é importante ter em mente que a inovação tecnológica é uma questão estratégica.

Independente do segmento de atuação das empresas, a transformação digital já não é mais novidade no mercado. Inteligência artificial, realidade aumentada, robótica, nuvem e Internet das Coisas (IoT). Termos que há alguns anos não eram nada conhecidos, hoje já fazem parte do cotidiano. Com a crescente adoção de tecnologias mais sofisticadas e inteligentes, a inovação na indústria não para de avançar. Esse fenômeno que está ocorrendo em larga escala nos últimos anos foi batizado de Indústria 4.0, ou também chamado de Quarta Revolução Industrial. Resumidamente, o enfoque são as fábricas inteligentes, as quais tornam os processos produtivos autônomos, eficientes e customizáveis.

Detalhadamente, a Indústria 4.0 é uma tendência mundial de digitalização da indústria, que visa integrar tecnologias avançadas, como inteligência artificial, Internet das Coisas, automação avançada e outras, para transformar a forma como a indústria é conduzida. Esta nova era da indústria busca aumentar a eficiência, flexibilidade e agilidade nas operações, melhorando a qualidade dos produtos e a eficiência dos processos. A Indústria 4.0 é considerada uma evolução da manufatura e é vista como uma oportunidade para aprimorar a competitividade global e criar novas formas de produção. (MARTINS, M. S.; PAULA; BOTELHO, 2021) Esse conceito surgiu para transformar a vida das pessoas, por meio do crescimento econômico, da geração de empregos qualificados e da elevação da qualidade de vida.

A Quarta Revolução Industrial vem reforçar o pensamento de que a cadeia de produção também é suscetível à automação e digitalização. Existem diversas máquinas e equipamentos industriais que podem usufruir das vantagens que a indústria 4.0 tem a oferecer. Máquinas CNC podem ser controladas remotamente e fornecer dados em tempo real sobre suas operações (OLIVEIRA; ÁLVARES, 2018), robôs industriais podem ser programados

para se adaptar a diferentes tarefas e operar de forma autônoma com a incorporação de inteligência artificial (CORREA; RODRÍGUEZ CÓRDOBA; LEYTON, 2014), equipamentos conectados, como sensores, podem fornecer dados em tempo real sobre o desempenho e o estado dos equipamentos, sistemas de automação, como CLPs, podem ser conectados à Internet e se integrar com outros sistemas de informação da indústria. As possibilidades são diversas e, dentre elas, fazemos um destaque para as oportunidades em aprimorar a eficiência dos motores elétricos, componentes amplamente utilizados em muitos setores industriais e em equipamentos domésticos, devido à sua eficiência e versatilidade.

O gerenciamento da operação dos motores elétricos e de sua manutenção ainda é um desafio para fabricantes e usuários. Neste contexto, a Indústria 4.0 é uma grande oportunidade para otimizar o desempenho destes equipamentos. Com a utilização de tecnologias avançadas, é possível monitorar em tempo real a performance dos motores e identificar oportunidades de melhoria. Especificamente, podemos utilizar o conceito de Gêmeos Digitais, que são modelos virtuais de motores elétricos que permitem simular e monitorar suas operações de forma eficiente. Somado a isso, a integração de sensores e sistemas de controle permite ajustar automaticamente as configurações dos motores para otimizar sua eficiência e evitar falhas. A utilização de um sistema SCADA permite monitorar, supervisionar e controlar as informações obtidas para facilitar o gerenciamento dos dados e tornar o processo intuitivo. Baseado nestas ideias que o presente trabalho foi realizado.

1.2 Trabalhos Relacionados

Os principais temas deste trabalho estão atrelados à manutenção, otimização e monitoração de máquinas industriais utilizando recursos disponíveis da indústria 4.0. Neste sentido, foram selecionados algumas pesquisas e alguns artigos que abordam estes tópicos recorrentes, de modo a destacar a importância do assunto no meio acadêmico e o crescente desenvolvimento científico na área.

1.2.1 Manutenção Preditiva

A manutenção preditiva é um método de manutenção que utiliza técnicas de monitoramento e análise de dados para prever possíveis falhas em equipamentos antes que elas ocorram. Ela envolve a coleta de dados e informações sobre o desempenho dos equipamentos, utilizando ferramentas de monitoramento como medidores, sensores e softwares de análise de dados e, com a implementação de ferramentas da indústria 4.0, é possível monitorar os equipamentos de forma remota e online. A partir destes dados, é possível prever quando uma falha pode ocorrer e realizar a manutenção antes que o equipamento pare de funcionar. Pensando nisso, o trabalho de Luciano Baldissarelli e Elton Fabro propôs uma solução em manutenção preditiva de equipamentos rotativos industriais (BALDISSARELLI; FABRO,

2019), onde foi possível eliminar manutenções corretivas em um exaustor industrial após a implantação de um sistema de monitoramento online que identifica vibrações no motor e mancais do conjunto do ventilador. Seus dados podem ser acompanhados em qualquer terminal da fábrica ou por meio de um celular que esteja conectado a rede *wifi* da fábrica, apenas acessando o endereço IP do servidor.

Outros estudos também se preocuparam em detectar falhas e criar sistemas de monitoração para precaver e proteger os equipamentos industriais. Flavia Justina Martins e Elton Fabro propuseram a utilização de um sensor inteligente (WEG Smart Sensor) para identificar, a partir de medições contínuas, o estado de saúde de um equipamento industrial, atuando como instrumento de manutenção preditiva (MARTINS, F. J.; FABRO, 2020). O equipamento escolhido em seu trabalho foi um motor de extrusora industrial e eles conseguiram atender às demandas propostas, coletando dados de vibração e temperatura do motor por meio de um celular com conexão Bluetooth, além de verificar o software de monitoração e o custo-benefício deste sensor. Já autores de outra pesquisa procuraram desenvolver um protótipo de aplicativo mobile nativo Android para visualização de dados de um sistema de monitoramento de máquinas e equipamentos de uma empresa no setor energético no Brasil (FERREIRA; SERUFFO; PIRES, 2022). O aplicativo fornece dados e gráficos em tempo real dos equipamentos da empresa comprovando, por meio da análise de especialistas, ser uma ferramenta de acesso que pode dar amplo suporte na manutenção preventiva.

1.2.2 Motores elétricos

Dentre as diversas máquinas industriais que compõem uma fábrica, os motores elétricos não são exceção na pesquisa acadêmica e na procura de sua integração com a indústria 4.0. Especificamente, diversos artigos foram publicados com estudos sobre detecção de falhas e sistemas de supervisionamento nestes equipamentos. O trabalho apresentado por (NÓBREGA SOBRINHO; SENA; LIMA FILHO, 2021) procura desenvolver estratégias a partir de análises de variáveis mecânicas e elétricas para a detecção de avarias na pista externa dos rolamentos de um motor de indução trifásico. As três técnicas utilizadas se comprovaram eficazes (com seus devidos comentários), sendo elas a análise do som, de vibração mecânica e de eficiência energética.

Em outro artigo (SOARES et al., 2020), desenvolveram um sistema de proteção microcontrolado para detectar falhas e proteger motores assíncronos trifásicos na ocorrência de perturbações produzidas por falta de fases, inversão de fases ou sobrecorrentes. Este sistema utiliza o microcontrolador ATmega328p e por meio de um algoritmo, o sistema de detecção de passagem por zero de cada fase provoca o desligamento do motor trifásico havendo discordância na sequência estabelecida como padrão da rede elétrica da concessionária, a partir da análise da corrente. Somado a estes estudos, diferentes abordagens criativas foram utilizadas para realizar a detecção de falhas em motores elétricos. Por exemplo, utilizando uma rede

neural treinada com o algoritmo ELM, o artigo apresentado por (RAMALHO et al., 2014) utiliza características extraídas de um sinal de vibração obtido com acelerômetros MEMS para aplicação na detecção e identificação de falhas mecânicas e elétricas. Os resultados obtidos registraram uma alta taxa de falsos positivos, porém isso não comprometeu o avanço obtido no desenvolvimento deste sistema.

1.3 Objetivos

Motivado pelas ideias e tecnologias da indústria 4.0 e pela grande quantidade de trabalhos relacionados à manutenção preditiva e ao supervisionamento em diferentes maquinários industriais, com destaque aos motores elétricos, utilizando diversas abordagens diferentes e criativas, este trabalho propõe projetar e inicializar o desenvolvimento de um sistema de manutenção preditiva, otimização e monitoração de motores elétricos por meio do conceito de Gêmeos Digitais. Para isso, tem como objetivo definir as especificações necessárias para o projeto, realizar a coleta de dados dos motores utilizando sensores de temperatura, vibração e som junto com um inversor de frequência e possibilitar a criação dos seus modelos virtuais a partir dos dados obtidos. Por meio destes dados, também será construído um sistema supervisório para a leitura e comando das variáveis, afim de se facilitar o acesso às informações para uma pessoa leiga no assunto, e pretende-se incorporar uma inteligência artificial com a ajuda de outro trabalho para a análise do modelo virtual.

1.4 Estrutura do Documento

O presente manuscrito foi organizado em cinco capítulos de forma a facilitar o entendimento. A estrutura do documento foi dividida da seguinte forma:

- ***Capítulo 1 - Introdução:*** Breve contextualização do assunto, apresenta alguns trabalhos relacionados, os objetivos e a estrutura do documento.
- ***Capítulo 2 - Fundamentação Teórica:*** Aborda os conceitos teóricos e técnicos relacionados a motores elétricos, gêmeos digitais e sistemas de supervisão e controle.
- ***Capítulo 3 - Materiais e Métodos:*** Serão descritos os materiais utilizados, os motivos destes terem sido escolhidos e os métodos de pesquisa empregados.
- ***Capítulo 4 - Desenvolvimento:*** Apresenta o processo no qual o trabalho foi submetido, desde a sua concepção até os resultados finais, onde pode ser observado a implementação física e computacional do sistema proposto.
- ***Capítulo 5 - Conclusões:*** Observações finais do trabalho e propostas de trabalhos futuros.

2 Fundamentação Teórica

2.1 Motores Elétricos

Motores são dispositivos com papel imprescindível para a sociedade, amplamente utilizados em uma variedade de aplicações domésticas e industriais. Desde os primórdios da humanidade, o homem utiliza fontes motoras para realizar trabalho e desenvolver-se, seja na produção rural das primeiras comunidades agrícolas para alimentar-se e realizar comércio ou até em espaçonaves nos tempos modernos para ir ao espaço. Dentre os diferentes tipos de motores existentes (a combustão, ar comprimido), o motor elétrico é o mais utilizado nos equipamentos em geral devido à versatilidade de adaptação a diferentes cargas somado às vantagens da energia elétrica, por ser mais fácil de manusear, ecológica e barata. Eles são dispositivos que convertem energia elétrica em energia mecânica e é possível encontrar este tipo de motor no cotidiano de uma pessoa através de equipamentos como portões eletrônicos, elevadores, ventiladores, escadas rolantes e até em suas residências nos eletrodomésticos: geladeiras, micro-ondas, liquidificadores, aspiradores, máquinas de lavar.

Em relação à industria, motores elétricos são componentes fundamentais, pois são usados para alimentar uma ampla gama de equipamentos e máquinas como: compressores, bombas, veículos elétricos, misturadores, discos de corte. Eles fornecem a força motriz necessária para a produção de bens e serviços em vários setores industriais, incluindo alimentos, químicos, têxtil, papel e celulose, mineração, entre outros. Em países industrializados, por exemplo, 65% do consumo de eletricidade das indústrias são responsáveis pelos motores elétricos, e 90% deste consumo é em função dos motores de indução com uma configuração específica chamada de gaiola de esquilo (MARUTHI; PANDURANGA VITTAL, 2005). É possível visualizar na Fig. 2.1 uma visão em corte deste tipo de motor.

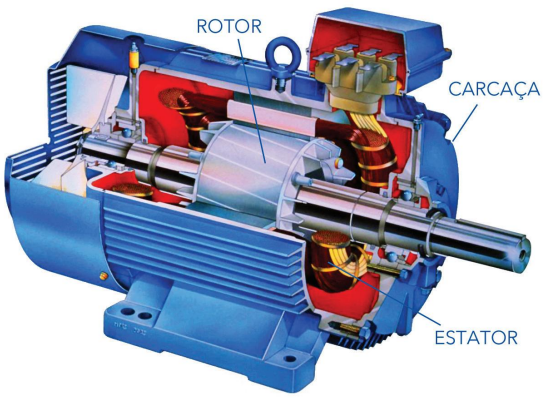


Figura 2.1 – Visão em corte de um motor elétrico
Fonte: https://issuu.com/sobratema/docs/mt_248_alta_sem_marca_de_corte_final/s/11158369 em 08/07/2023

O motor elétrico é dividido resumidamente em três partes principais: a carcaça, o estator e o rotor (Fig. 2.1). A carcaça é responsável por proteger as peças internas e dissipar o calor que é gerado pelo trabalho do motor. Já o estator é a parte estática fixa na carcaça e possui a função de fornecer o fluxo magnético que vai interagir com as partes girantes. Por fim, o rotor é a parte que gira do motor. Ele sofre influência de todo o processo eletromagnético e o transforma em movimento mecânico. O modo de funcionamento básico de um motor elétrico é baseado na interação entre o campo magnético gerado pelo fluxo de corrente elétrica através dos enrolamentos do rotor e o campo magnético gerado pelo ímã permanente ou pela corrente elétrica através dos enrolamentos de excitação do estator. As forças de atração ou repulsão geram torque e consequentemente puxam ou empurram os polos móveis, definindo dessa forma a velocidade de giro a partir da intensidade do campo magnético. Este, por sua vez, é controlado pela frequência da corrente elétrica que pode ser ajustada por meio de um inversor de frequência (FRANCHI, s.d.). De forma ilustrativa e com mais detalhes, a Fig. 2.2 explicita o princípio de funcionamento do motor elétrico.

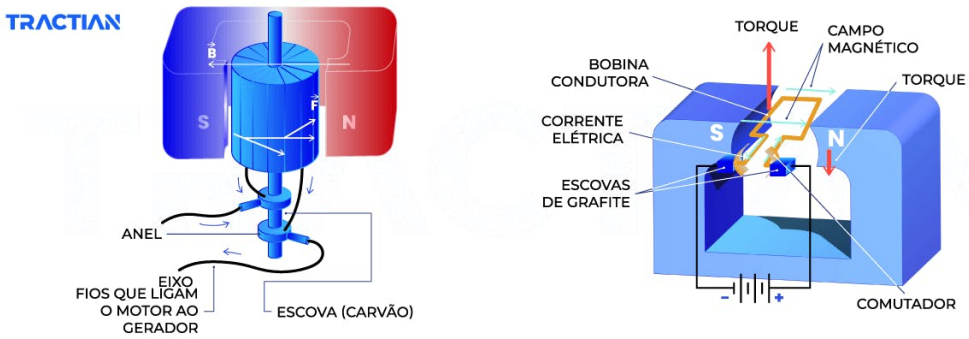


Figura 2.2 – Princípio de funcionamento do motor elétrico
Fonte: https://tractian.com/blog/motores-eletricos-entenda-a-funcionalidade-desse-ativo em 08/07/2023

Uma característica importante dos motores elétricos é a sua eficiência, visto que tem impacto direto na qualidade dos produtos, na segurança dos trabalhadores e na eficiência energética da planta. Uma melhora nela pode resultar em redução de custos operacionais, aumento da vida útil dos equipamentos e até redução de emissões de CO2. Para isso, existem vários tipos de motores elétricos, os quais podem ser classificados como motores de corrente contínua (CC), corrente alternada (CA) ou motores universais (CC/CA). A partir destas categorias derivam-se outros tipos de motores em função de suas características, como pode ser observado na Fig. 2.3. Os motores CC são alimentados por uma fonte de energia contínua, como o próprio nome sugere, que é geralmente uma bateria ou um retificador. Fisicamente, os elétrons percorrem um único sentido neste tipo de alimentação. Já os motores CA são alimentados por uma fonte de corrente alternada, geralmente fornecida pela rede elétrica de energia, onde o fluxo de elétrons alterna o sentido em função da mudança de polaridade do gerador da corrente. Por fim, os motores universais trabalham com as duas formas de corrente elétrica.

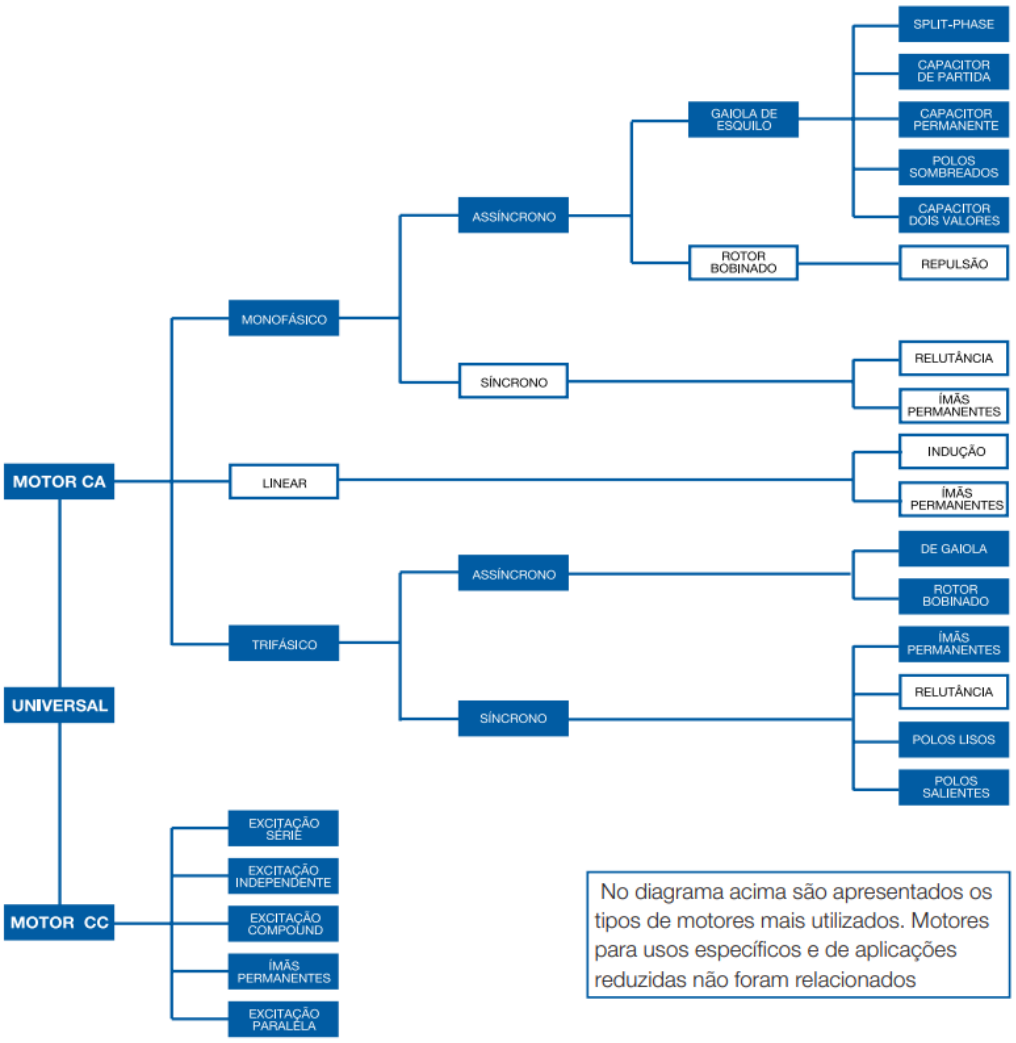


Figura 2.3 – Fluxograma de especificações para motores elétricos
Fonte: (WEG, 2023b)

Especificamente em relação aos motores AC, podemos subdividí-los em motores trifásicos e monofásicos. Motores trifásicos são alimentados por três fases de corrente elétrica alternada, o que lhes permite operar de forma mais eficiente e suave em comparação com motores monofásicos, que são alimentados por apenas uma fase. Devido à sua eficiência, os motores trifásicos são amplamente utilizados em aplicações industriais de grande porte, como sistemas de produção de energia e bombas de grande capacidade. Por outro lado, os motores monofásicos são mais adequados para aplicações domésticas e pequenas, como ventiladores e ferramentas elétricas, pois são mais fáceis de instalar e requerem uma fonte de alimentação mais simples. Em resumo, a escolha entre os diferentes tipos de motores elétricos, trifásicos ou monofásicos, depende da necessidade de potência, eficiência e facilidade de instalação para uma determinada aplicação. O uso de um inversor de frequência é importante em aplicações que requerem velocidades variadas, visto que ele permite controlar a velocidade e a eficiência do motor, garantindo o funcionamento otimizado em todos os momentos.

2.1.1 Inversores de Frequência

Inversores de frequência, como já citado anteriormente, são dispositivos elétricos que fornecem controle preciso da velocidade e do torque de motores elétricos. Eles funcionam convertendo a corrente elétrica fornecida pela rede em corrente alternada de alta frequência, e em seguida regulando a frequência da corrente fornecida ao motor elétrico para diferentes valores. Isso permite aos usuários variar a velocidade de giro e controlar o torque do motor com precisão, aumentando a eficiência energética e o desempenho do sistema (FRANCHI, s.d.). A combinação dessas características, somado a outras vantagens como redução do pico de corrente e aumento da vida útil das máquinas, torna os inversores de frequência uma escolha atraente para diferentes aplicações, especialmente no setor de automação industrial em que desempenha um papel essencial.

Dentre as diversas funcionalidades dos inversores de frequência, eles são capazes de fornecer uma ampla gama de informações úteis para os usuários, os permitindo identificar possíveis problemas de funcionamento, bem como otimizar o desempenho e a eficiência energética de um motor. Os inversores podem monitorar e exibir a tensão, corrente, potência e frequência da corrente elétrica fornecida ao motor elétrico. Também podem fornecer informações sobre o estado de funcionamento do motor, como velocidade, torque e temperatura. Alguns inversores de frequência também fornecem recursos avançados de diagnóstico, como a capacidade de detectar falhas no motor e fornecer alertas para correções. Estas informações também são úteis para manter a integridade e segurança do sistema, pois permitem detectar problemas antes que eles se tornem mais graves e causem danos ao motor ou ao equipamento circundante. Além disso, os inversores de frequência também oferecem flexibilidade de instalação e integração com outros sistemas. Eles podem ser instalados de forma remota, e muitos modelos suportam comunicação com outros dispositivos através de protocolos padrão, como Modbus e Ethernet. (FRANCHI, s.d.)

Todas as informações citadas, as parametrizações do motor e os comandos de controle dependem da programação do inversor que permite aos usuários personalizar as configurações para atender às suas necessidades específicas. A programação de inversores de frequência é realizada através de uma interface de usuário, como um display gráfico ou um software de computador. Essas interfaces permitem aos usuários configurar e ajustar as configurações do inversor de frequência, como a velocidade máxima do motor, a curva de aceleração/desaceleração, entre outros, e visualizar as informações referentes ao estado do motor. Alguns inversores de frequência também podem ser programados através de dispositivos externos, como um controlador lógico programável (CLP), microcontroladores (Arduino, Raspberry Pi) ou um sistema de supervisão remoto.

2.2 Gêmeos Digitais

Gêmeos digitais referem-se a réplicas digitais de objetos, sistemas ou processos físicos. São modelos virtuais criados por meio de dados históricos, de sensores ou futuros (*machine learning*) para simular o comportamento e o desempenho de suas contrapartes físicas em tempo real. Os gêmeos digitais são projetados para fornecer uma compreensão detalhada de como um objeto ou sistema opera e podem ser usados para monitorar, analisar e otimizar o desempenho de sistemas e processos físicos (ATTARAN; CELIK, 2023). Como resultado, a implementação de gêmeos digitais levou a uma maior eficiência, segurança e sustentabilidade em vários setores. Eles podem replicar muitos itens, desde peças únicas de equipamento em uma fábrica a instalações completas, como turbinas eólicas e até cidades inteiras. Eles também podem ajudar as empresas a criar planos de negócios de longo prazo, identificar novas invenções e até simular cenários muito demorados ou caros para serem testados com objetos físicos.

A capacidade dos gêmeos digitais de se integrar entre os espaços físico e digital permite que o sistema físico ajuste seu comportamento em tempo real com base no feedback fornecido pelo gêmeo digital. Em outras palavras, a tecnologia de gêmeos digitais cria uma ponte entre os espaços físico e digital, permitindo que o gêmeo digital imite com precisão o estado do mundo real do gêmeo físico. Para criar essa ponte, é necessário desenvolver um forte vínculo entre os gêmeos físicos e digitais, o que pode ser alcançado pela circulação de dados sensoriais *online* entre eles e só pode ser desenvolvido depois de replicar com precisão o ambiente dinâmico do sistema físico e todos os fatores que impactam as operações desse sistema no espaço digital. Este processo, por outro lado, necessita de outras tecnologias aliadas aos gêmeos digitais para ser desempenhado corretamente.

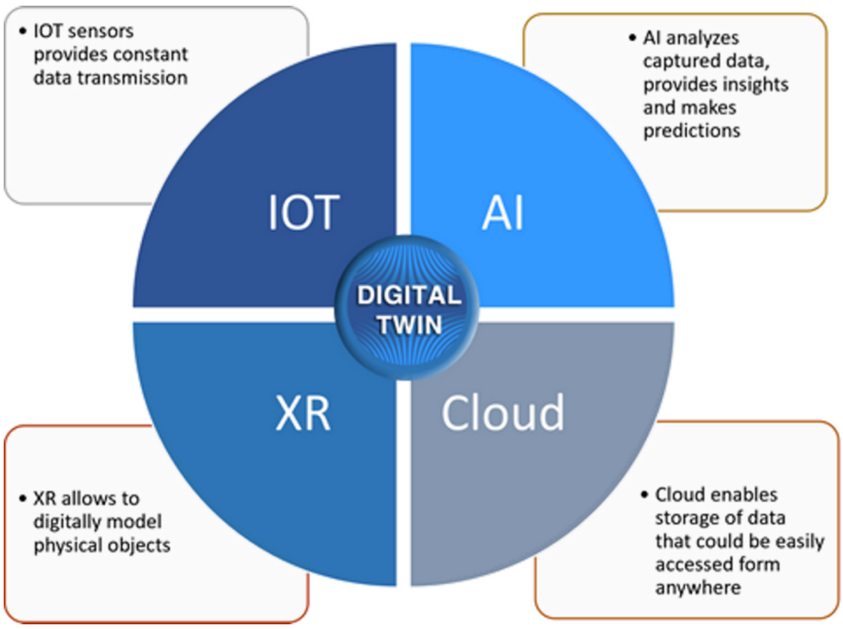


Figura 2.4 – Vertentes dos Gêmeos Digitais
Fonte: (ATTARAN; CELIK, 2023)

De forma resumida, a tecnologia de gêmeos digitais se baseia em 4 outras vertentes tecnológicas para capturar e armazenar dados em tempo real, analisar o valor das informações obtidas e criar representações virtuais coexistentes e interativas com seus objetos físicos: Internet das Coisas (IoT), Inteligência Artificial (AI), Realidade Estendida (XR) e Nuvem (Fig. 2.4). A IoT é responsável pela conexão de uma rede de dispositivos conectados via internet que trocam dados entre si. Dessa forma, diferentes sensores, softwares e equipamentos comunicam-se para prover dados constantes e confiáveis para as aplicações de Gêmeos Digitais construírem seu modelos virtuais. A Inteligência Artificial pode auxiliar a analisar de forma avançada os dados obtidos, determinando valores e prioridades para cada dado, além de prover previsões e sugestões a partir das condições atuais do sistema. Já a Realidade Extendida é um termo genérico usado para descrever tecnologias que mesclam os mundos físico e virtual como Realidade Virtual (VR), Realidade Aumentada (AR) e Realidade Mista (MR). Ela é utilizada para criar os modelos digitais e permitir a interação com o conteúdo digital. Por fim, a Nuvem guarda todas as informações obtidas e geradas em larga escala em um local seguro e de fácil acesso em qualquer lugar, reduzindo no processo efetivamente o tempo de computação de sistemas complexos (ATTARAN; CELIK, 2023).

A tecnologia de gêmeos digitais ganhou força significativa em uma ampla gama de setores nos últimos anos. Dentre eles, podemos citar as áreas da saúde, aeroespacial, robótica, arquitetura, transporte, energia (ATTARAN; CELIK, 2023). Especificamente, na indústria da manufatura, por exemplo, os gêmeos digitais estão sendo usados para otimizar os processos de produção e melhorar a qualidade do produto. Em um estudo, o foco da pesquisa recaiu sobre a geração de um substituto digital de uma fábrica baseada em objetos reais em um ambiente construído com planos e documentação inexistentes ou insuficientes e com cenas complicadas (por exemplo, ambientes internos com objetos repetitivos de formato irregular e dados de medição ruidosos como entrada) para melhorar os processos de produção e logística, levando à redução de custos. Os resultados mostram que mais de 80% dos objetos da fábrica puderam ser reconhecidos no modelo gerado com diferentes configurações arquitetônicas para otimizar a fábrica a depender dos requisitos (SOMMER et al., 2023).

Na indústria aeroespacial, os gêmeos digitais estão sendo usados para melhorar o design e o desempenho das aeronaves. Eles podem simular a aerodinâmica de uma aeronave e otimizar seu desempenho. Por exemplo, o estudo de um caso industrial derivado da colaboração com três fabricantes envolvidos no desenvolvimento de um sistema de propulsão elétrica (EPSs) baseado na tecnologia *Hall Effect Thruster* (HET), propõe usar ativamente gêmeos digitais nas fases iniciais, onde o design pode ser amplamente alterado (PANAROTTO; ISAKSSON; VIAL, 2023). O estudo realiza uma abordagem econômica para identificar um conjunto de módulos gêmeos digitais confiáveis e robustos que podem ser reutilizados e recompostos para criar variantes de gêmeos digitais, seguindo uma abordagem de plataforma modular. Esses resultados apontam para o custo-benefício da aplicação de uma abordagem modular à criação de gêmeos digitais, como um meio de reutilizar os resultados dos testes

físicos para validar novos projetos e seus intervalos de validade, podendo ser eficaz e útil nos estágios iniciais em que o design pode ser amplamente alterado, especialmente no contexto de sistemas avançados, como produtos aeroespaciais e espaciais.

Já na área da saúde, os gêmeos digitais estão sendo usados para personalizar tratamentos médicos e melhorar os resultados dos pacientes. Eles podem simular o comportamento do corpo de um paciente e prever como ele responderá a diferentes tratamentos. Um esboço de alto nível de um roteiro para a construção de um gêmeo digital de uma pessoa é representado na Fig. 2.5. Para criar um modelo digital que possa ser testado e simulado, é preciso reunir enormes volumes de dados de saúde por meio de vários dispositivos IoT e usar modelos baseados em Inteligência Artificial. Com base no estilo de vida do paciente, hábitos alimentares regulares e dados de açúcar no sangue, essa tecnologia ajuda a alertar o paciente sobre prescrições, ajustes dietéticos, consultas médicas e outras situações. O gêmeo digital do próprio paciente usa *insights* de dados anteriores para ajudar a selecionar o medicamento mais apropriado, prever os resultados de uma cirurgia específica e controlar doenças crônicas (HALEEM et al., 2023).

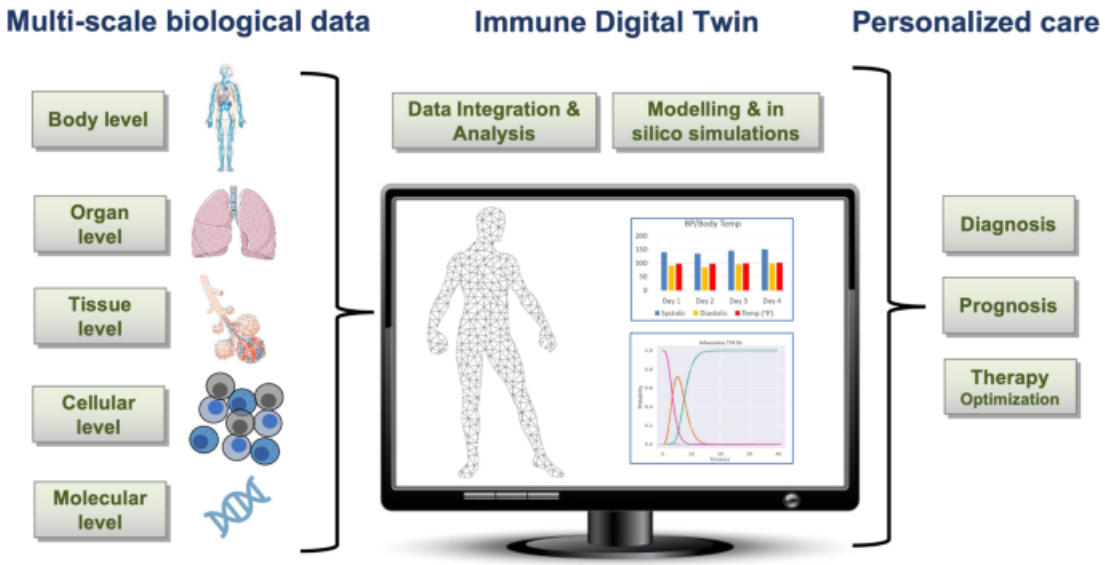


Figura 2.5 – Uso de Gêmeos Digitais na área da saúde
Fonte: https://rdcu.be/c9uHR em 08/07/2023

Da mesma forma, no setor de energia, os gêmeos digitais estão sendo usados para otimizar o desempenho de usinas elétricas e sistemas de energia renovável. Eles surgem como uma técnica potencial para melhorar o desempenho, reduzir os custos de manutenção e operação e garantir uma operação mais segura para qualquer sistema associado (SEMERARO et al., 2023). Na área de transporte, os gêmeos digitais estão sendo usados para na simulação de tráfegos de rodovias e sistemas de transporte. Utilizando softwares específicos somados a dados de tráfego em tempo real recém-disponíveis recebidos a cada minuto dos contadores de tráfego na rodovia de Genebra via ODPMS, a pesquisa realizada por (KUŠIĆ; SCHUMANN; IVANJKO, 2023) apresenta e testa pela primeira vez uma réplica digital de autoestrada ao

vivo no simulador microscópico SUMO, refletindo com precisão a dinâmica real do tráfego com uma resolução temporal muito fina durante o tempo de execução do sistema. Este sistema forma a base para o desenvolvimento da análise preditiva em tempo real como suporte para decisões críticas de segurança no gerenciamento de tráfego. Espera-se que o uso de gêmeos digitais desempenhe um papel crítico no desenvolvimento de cidades inteligentes.

É importante destacar que, em todas as áreas citadas, a manutenção preditiva baseada em gêmeos digitais se faz presente e é um foco de pesquisa nos respectivos campos de estudo (ZHONG et al., 2023). Diante da falha de um equipamento, a manutenção preditiva tradicional enfatiza a análise do estado do equipamento, a previsão de falha do equipamento e a tomada de decisão auxiliar. Essas funções não consideram a interação complexa e o mecanismo de evolução dos atributos estruturais e físicos no equipamento, e não podem considerar os parâmetros do equipamento (Fig. 2.6) em tempo real durante o processamento de dados. A manutenção preditiva tradicional só pode descrever o processo dinâmico sob uma certa escala espacial e de tempo, e não pode realizar manutenção dinâmica de longo prazo. A tecnologia de gêmeos digitais simula quando o equipamento está funcionando, coleta os dados do equipamento que mudam dinamicamente com o tempo e pode fornecer um serviço mais inteligente do que a manutenção preditiva tradicional, visto que pode realizar a manutenção dinâmica de longo prazo do equipamento.

Figura 2.6 – Parâmetros tradicionais de um equipamento
Fonte: https://www.iclass.com.br/blog/manutencao-corretiva/ em 08/07/2023

É possível perceber como o uso de gêmeos digitais está se tornando cada vez mais popular em vários setores. Existem infinitas outras aplicações que poderiam ser citadas e inclusive existem estudos somente para analisar artigos que mencionam o uso de gêmeos digitais afim de caracterizá-lo, identificando lacunas no conhecimento e áreas necessárias para pesquisas futuras (JONES et al., 2020). Os gêmeos digitais têm o potencial de revolucionar a maneira como projetamos, construímos e operamos sistemas físicos. Com sua capacidade de simular o comportamento e o desempenho de sistemas físicos em tempo real, espera-se que os gêmeos digitais continuem a desempenhar um papel significativo na melhoria da performance, segurança, manutenção preditiva, redução de custos, monitoração remota, aceleração do tempo de produção e sustentabilidade em todos os setores.

2.3 Sistemas de Supervisão e Aquisição de Dados

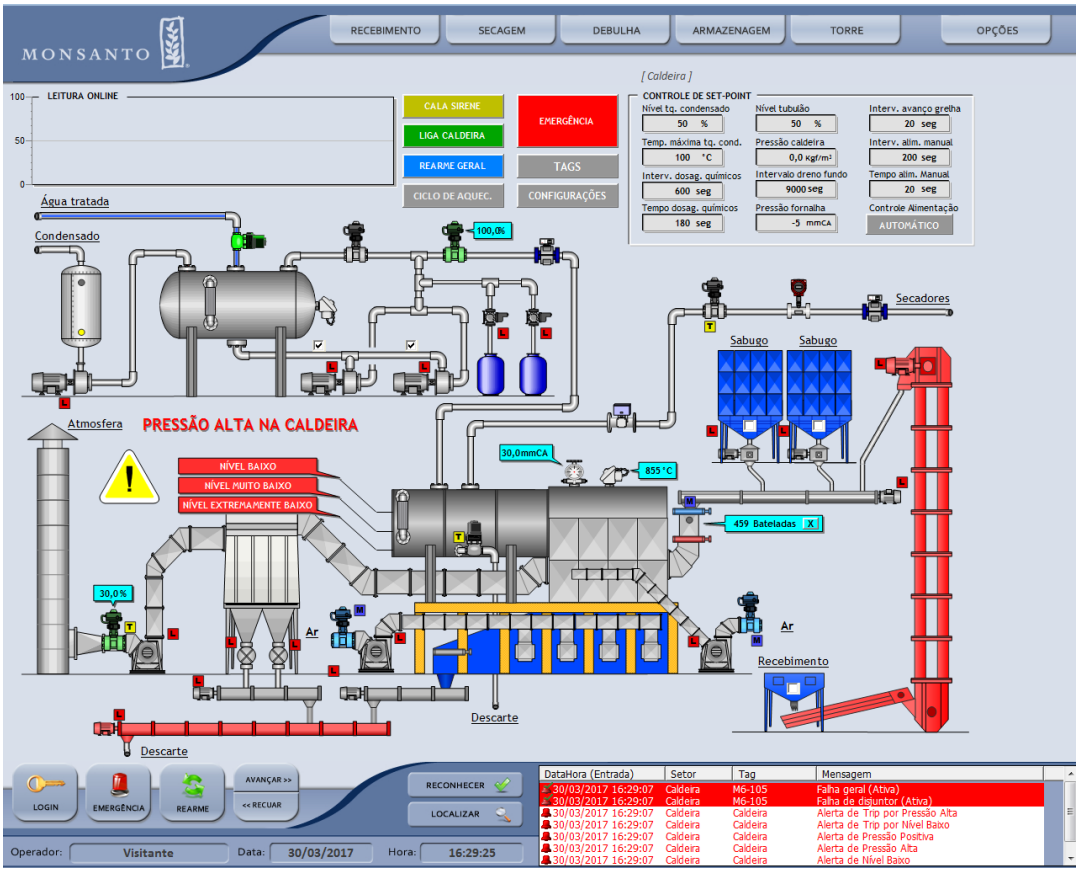


Figura 2.7 – Tela Sinótica de um sistema supervisório para caldeira
Fonte: https://www.soluzionetecnologia.com.br/bws-gallery/revisao-de-sistema-de-controle-para-caldeira/caldeira-visao-geral/ em 08/07/2023

Sistemas de Supervisão e Aquisição de Dados (SCADA), ou simplesmente chamados de sistemas supervisórios, são sistemas de controle que utilizam software para monitorar, supervisionar e controlar as variáveis e os dispositivos de processos e infraestruturas industriais. De forma simplificada, os sistemas SCADA servem para acompanhar, configurar e armazenar dados e disponibilizar recursos para intervenção automática ou manual, tanto pelo operador como pelo gerente, no processo, quando necessário. Através dos dados armazenados, o sistema supervisório permite a visualização e supervisão dessas informações em tempo real por meio de relatórios estatísticos, relatórios gerenciais para controle de qualidade, consultas a dados históricos e sinalização de alarmes e falhas usando telas sinóticas amigáveis ao cliente, como a tela apresentada na Fig. 2.7. Neste exemplo de um processo industrial de uma caldeira, é possível perceber visualmente diversas variáveis dispostas de forma de fácil acesso como pressão, temperatura, nível, tempos de dosagem e alimentação. Somado a isso, também há os alarmes e mensagens de alerta para informar ao usuário alguma falha ou risco no processo e botões para manualmente ligar a caldeira, rearmar, desativar sirene, configurar, entre outros.

A função principal de um sistema SCADA é visualizar e operar uma planta em partes ou em sua totalidade, muitas vezes remotamente, de uma maneira fácil de entender e interagir. Dessa forma, instalações de grande escala podem ser completamente supervisionadas por um grupo reduzido de pessoas em um local central, como uma sala de controle, permitindo uma melhor visão do processo. A aplicação deste sistema evita a presença constante de operadores em locais remotos durante situações normais, tornando mais seguro e econômico, porém a operação local nas unidades da planta permanece importante e necessária. Estes sistemas são comumente usados em operações automatizadas de grande e médio porte, como refinarias de petróleo e gás e usinas de energia, porém também podem ser utilizados em sistemas de serviços públicos, como de tratamento de água e esgoto, em sistemas de automação prediais, para monitoração de caixas de água, gás, incêndio, e para implementar interfaces de sistemas IoT (MORAES; CASTRUCCI, 2007).

No geral, os sistemas supervisórios são críticos para a operação eficaz de muitos processos e desempenham um papel vital na garantia de segurança, eficiência, economia e confiabilidade nessas operações. Um dos principais benefícios é a melhoria da eficiência, visto que o monitoramento e controle em tempo real do processo podem ajudar a identificar e resolver problemas rapidamente, minimizando o tempo de inatividade e maximizando a produtividade. Os sistemas SCADA também aumentam a segurança ao fornecerem aos operadores dados e alarmes em tempo real quando certas condições são atendidas, permitindo que possíveis riscos de segurança sejam identificados e tratados antes que ocorra um incidente. Além disso, estes sistemas podem coletar e analisar grandes quantidades de dados em tempo real e armazená-los em banco de dados, fornecendo informações valiosas. Estes dados podem ser utilizados como base para outros sistemas como MES e ERP, conforme necessário, e podem ajudar os operadores a visualizar tendências de longo prazo para identificar oportunidades de melhoria e otimizar o processo para máxima eficiência. Todas essas vantagens ajudam a reduzir os custos operacionais em termos de tempo e de dinheiro por meio da melhora da eficiência e qualidade, redução do tempo de inatividade e redução da necessidade de pessoal no local (MORAES; CASTRUCCI, 2007).

Os sistemas SCADA são formados por três elementos principais: a Central de Operações, o Sistema de Comunicação e a Estação Remota (GARCIA, 2019). Cada um desses elementos são formados por uma combinação de elementos de software e de hardware. A Estação Remota é onde ficam os equipamentos de instrumentação, as máquinas, disjuntores, válvulas, todos os equipamentos que geram dados e os controladores que realizam toda a coleta e o processamento de dados. A Central de Operações é o local onde todas as informações coletadas são centralizadas, analisadas e manipuladas por meio de uma interface fácil com o usuário. Já o Sistema de Comunicação interliga a Estação Remota à Central de Operações e vice-versa utilizando diversos meios de comunicação. Aprofundando no funcionamento deste tipo de sistema, podemos classificar a distribuição operacional dele em 5 etapas (GARCIA, 2019):

FUNCIONAMENTO DE UM SISTEMA SCADA

1. Medição e Atuação: coleta de dados por equipamentos de instrumentação.
2. Aquisição e Controle: tratamento das variáveis pelos controladores.
3. Comunicação: gerenciamento do meio físico, codificação e protocolos.
4. Supervisão: operação, monitoração e apresentação por meio da interface homem-máquina (IHM).
5. Gerenciamento: controle e comandos do processo, manipulação de variáveis e simulações.

A relação entre a distribuição operacional e os 3 elementos principais de um sistema SCADA pode ser observado abaixo na Fig. 2.8.

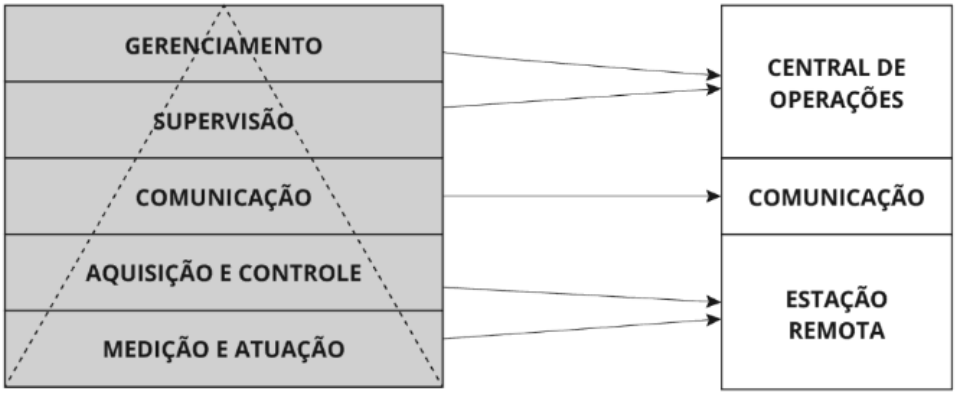


Figura 2.8 – Distribuição operacional de um sistema SCADA
Fonte: (GARCIA, 2019)

O estudo da distribuição operacional de um sistema supervisório recorre à uma tradicional pirâmide de automação, sutilmente demonstrado na Fig. 2.8 por meio das linhas pontilhadas. A pirâmide de automação, resumidamente, é uma representação visual que visa organizar e ilustrar cinco níveis de controle e de trabalho. Os níveis mais baixos comportam uma quantidade maior de itens e informações e relacionam-se com os equipamentos utilizados “em campo”, que tem contato direto com o ambiente de produção. Em contrapartida, os níveis superiores tratam do gerenciamento dos processos e da planta, onde os dados são melhor trabalhados e aumentam em qualidade. Neste caso, a base possui as etapas de Medição e Atuação e Aquisição e Controle, ambas ocorridas ainda na Estação Remota. A Medição e Atuação refere-se aos sensores e atuadores que coletam dados e executam instruções, já a Aquisição e Controle refere-se aos controladores que analisam dados recebidos pelos sensores e enviam comandos para os atuadores. Estes processos ocorrem concomitantemente e podemos resumi-los como um processo único de aquisição de dados.

O processo de aquisição de dados envolve a coleta de dados de sensores, máquinas e dispositivos usando hardware especializado, como unidades terminais remotas (RTUs), controladores lógicos programáveis (CLPs) ou até microcontroladores. Os equipamentos de instrumentação instalados no chão de fábrica estão conectados fisicamente a estes hardwares que processam os dados, os enviam para a estação central de monitoramento e aciona saídas para os atuadores realizarem suas ações com base em parâmetros pré-programados, instruções armazenadas e funções de lógica específicas. Dependendo das entradas e saídas, o controlador pode monitorar e registrar dados em tempo real, como produtividade da máquina ou temperatura operacional, iniciar e parar processos automaticamente e gerar alarmes a depender da alteração do estado de uma variável. Em alguns casos específicos, a depender do tipo de aplicação, o sistema pode contar com sensores/atuadores inteligentes que podem dispensar controladores por terem capacidade de processamento própria e se comunicam diretamente com o sistema supervisório.

Subindo a pirâmide, temos a etapa de Comunicação. O sistema de comunicação é um elemento essencial de um sistema supervisório. A transmissão entre o equipamento hospedeiro de aplicações SCADA e os dispositivos de controle e aquisição de dados baseia-se na troca de dados apropriada entre eles, podendo usar tecnologias de comunicação com ou sem fio. Os modos de comunicação variam, podendo ser por consulta ou por interrupção. O primeiro caso é comum em arquiteturas mestre-escravo, onde um ponto do sistema é definido como mestre (Central de Operações) e os demais como escravos (Estações Remotas). Neste caso, o mestre possui controle absoluto das comunicações, sempre as iniciando para requisitar dados de leitura desejados de um escravo, que apenas responde após a recepção de um pedido. Já no segundo caso, os dispositivos da rede de supervisão se comunicam entre si somente quando os dados que estão monitorando mudam significativamente ou quando ocorre um evento. Normalmente os controladores enviam as informações relacionadas às variáveis monitoradas para o sistema SCADA (MORAES; CASTRUCCI, 2007).

A diversidade de aplicações, modelos e fabricantes dos equipamentos industriais implica em inúmeras possibilidades, seja utilizando diferentes meios físicos, bem como empregando diferentes protocolos de comunicação. Em geral, cada dispositivo adota algum protocolo específico, que pode ser proprietário ou aberto, para padronizar essa troca de dados. Isso pode ser um problema para estes sistemas, que necessitam dessa forma ter uma grande base de componentes de software para lidar com cada protocolo de cada equipamento. Para evitar a sobrecarga dos sistemas com suporte a protocolos pouco utilizados, existe o conceito de driver. Ele consiste em um componente de software responsável por gerenciar a troca de dados utilizando um protocolo/equipamento específico (MORAES; CASTRUCCI, 2007). Assim, apenas as funcionalidades pertinentes ao sistema são armazenadas, economizando memória e evitando desperdício de recursos computacionais. Apesar disso, existem esforços de padronização por parte de órgãos oficiais e associações de fabricantes, como é o caso dos protocolos Modbus (MODBUS, 2014), Fieldbus (VIEGAS; PEREIRA, 2013) e Profibus.

Nos níveis mais altos temos as etapas de Supervisão e Gerenciamento, ambas realizadas na Central de Operações. Em relação à Supervisão, o servidor SCADA processa os dados recebidos de um controlador central responsável pela coleta de dados provenientes dos demais dispositivos de campo para criar exibições gráficas em tempo real do processo industrial. Essa exibição gráfica são telas sinóticas, ou seja, telas contendo uma imagem esquemática do processo e elementos gráficos que reproduzem um painel de controle, e é conhecida como interface homem-máquina (IHM), que como o próprio nome sugere, serve como mediador da interação entre uma pessoa e um sistema de automação (GARCIA, 2019). A IHM é construída para facilitar o trabalho do operador ou usuário, apresentando uma imensa quantidade de dados complexos de forma intuitiva por meio de *widgets*, vinculados às variáveis monitoradas, como visto anteriormente na Fig. 2.7. A IHM permite analisar e supervisionar em tempo real a situação das máquinas ou dos processos e interagir com elas remotamente usando vários controles e comandos. Ela geralmente inclui uma variedade de recursos, como sistemas de vídeo, alarmes, gráficos de tendências, histórico de dados, registros de eventos e exibições de status, e podem ser apresentadas em hardware comuns, como monitores, ou em equipamentos mais robustos para ambientes com mais riscos de danificar a IHM, como a apresentada na Fig. 2.9.

Figura 2.9 – Exemplo de uma Interface Homem-Máquina (IHM)
Fonte: https://blog.kalatec.com.br/o-que-e-ihm/ em 08/07/2023

Por último, o Gerenciamento está atrelado ao sistema SCADA associado à IHM que pode permitir que os operadores controlem todo o processo industrial em um só lugar, como já citado. Após o operador tomar alguma decisão a partir de suas análises, ou por meio de ações pré-programadas, os comandos de controle são transmitidos de volta para as RTUs ou controladores no campo, que então ajustam o processo de acordo. As mudanças podem ser locais ou remotas e podem ajustar operações no nível de instalações completas, processos individuais ou particularmente em algumas máquinas. O supervisório também deve ser capaz de combinar dados de diversas fontes, processá-los e exportá-los para outros softwares, permitindo maior manipulação e gerenciamento das informações (GARCIA, 2019).

2.4 Especificações do projeto

O objetivo primordial do projeto a ser desenvolvido é promover uma gestão inteligente e eficiente dos motores elétricos utilizados em processos industriais, visando otimizar a produtividade, reduzir custos operacionais, prevenir falhas e aprimorar a manutenção preditiva. O monitoramento contínuo e em tempo real dos motores possibilita a detecção precoce de anomalias, permitindo uma rápida intervenção e minimizando potenciais paradas não programadas. As três tecnologias apresentadas possuem características específicas e aplicações distintas sendo necessário especificar alguns elementos em comum para a implementação do sistema:

1. Seleção do Hardware
2. Seleção do Software
3. Seleção do Sistema SCADA
4. Implementação de Gêmeos Digitais
5. Conectividade e Comunicação
6. Visualização e Análise de Dados
7. Implementação de Alertas e Alarmes
8. Integração com Sistemas de Gestão
9. Monitoramento Remoto e Acesso Seguro
10. Testes em bancada
11. Treinamento e Capacitação

Em detalhes, é fundamental selecionar sensores adequados capazes de coletar dados relevantes do desempenho dos motores elétricos. Dentre os principais sensores a serem considerados estão os de temperatura, vibração, corrente elétrica e velocidade. Esses sensores podem ser conectados diretamente aos motores ou aos seus controladores, viabilizando a aquisição dos dados físicos do processo. Cada hardware específico possui um software diferente que pode atendê-lo para programá-lo e calibrá-lo, sendo importante definir também estes elementos. Os motores elétricos em si possuem as suas especificações técnicas que podem variar dependendo do seu tipo e aplicação. É importante se atentar aos seguintes elementos para garantir que o motor atenda aos requisitos operacionais e de desempenho desejados: potência nominal, tensão nominal, corrente nominal, eficiência, torque nominal, fator de potência, velocidade nominal, classe de isolamento e grau de proteção.

O componente central do sistema é o software SCADA, o qual desempenha um papel crucial na integração e visualização dos dados coletados. É importante selecionar um software SCADA com uma interface gráfica amigável, capaz de se comunicar com dispositivos de campo, CLPs, sensores, equipamentos e outros sistemas, capaz de coletar dados de várias fontes, capaz de armazenar e gerenciar um histórico de dados e que permite que os operadores controlem remotamente o sistema. Ele deve ser programado e construído, configurando alarmes e notificações. Segurança, suporte a múltiplas plataformas e integração a outros sistemas de gestão são características desejáveis.

Adicionalmente, o sistema proposto emprega o conceito de Gêmeos Digitais. A criação do Gêmeo Digital possibilita a simulação do comportamento do motor sob diferentes condições operacionais, facilitando a identificação de desvios entre o desempenho real e o comportamento esperado conforme previsto pelo modelo virtual. É necessário que ele seja alimentado com dados relevantes em tempo real ou históricos, integrado aos sensores definidos anteriormente, incorporar modelos analíticos, como modelos matemáticos ou modelos de simulação, conectado a sistemas de controle e possuir uma interface de usuário que permite aos operadores e engenheiros visualizar, interagir e analisar os dados e resultados, geralmente em um modelo 3D que representa geometricamente o objeto físico ou sistema.

A comunicação entre os dispositivos de coleta de dados, o software SCADA e o Gêmeo Digital ocorre por meio de protocolos industriais consagrados, tais como Modbus, OPC UA ou MQTT. Essa integração possibilita a troca eficiente de informações e contribui para uma visão abrangente e holística do processo industrial monitorado. Para a garantia da segurança cibernética do sistema, é imprescindível implementar mecanismos de autenticação e criptografia nas comunicações, bem como adotar boas práticas de segurança da informação para proteger os dados sensíveis e evitar acessos não autorizados.

Ao finalizar toda a montagem e configuração do sistema, é hora de partir para os testes em bancada utilizando diferentes cenários e condições, analisando o comportamento do sistema em reação às situações apresentadas e se ele se encaixa dentro do esperado, fazendo os ajustes necessários e alimentando o banco de dados no processo. A implementação bem-sucedida desse sistema requer uma equipe multidisciplinar, composta por engenheiros especializados em automação industrial, engenharia elétrica e tecnologias da informação. É fundamental fornecer treinamentos adequados para a equipe responsável pelo monitoramento e operação do sistema, a fim de maximizar o potencial dessa solução tecnológica avançada.

3 Materiais e Métodos

Neste capítulo, serão apresentados os materiais utilizados, sejam eles físicos ou virtuais, e os métodos de pesquisa escolhidos para a implementação do Gêmeo Digital. O foco principal na escolha dos materiais é na confecção do sistema de monitoração e aquisição de dados, visto que se trata de uma parte mais física e necessita-se colocar os motores elétricos em funcionamento para desenvolver-se todo o resto. Em relação aos métodos, será mencionado de forma descritiva o planejamento do projeto, os motivos das escolhas dos materiais, alguns problemas encontrados e se houveram estratégias para coleta de dados.

O presente trabalho foi realizado nas dependências do GRACO da Universidade de Brasília, e todos os materiais físicos implementados foram adquiridos por meio dele, sendo que todo o sistema construído pode ser encontrado no laboratório. Em resumo, este capítulo será dividido em 3 seções referentes a 3 aspectos diferentes do sistema implementado: uma seção abordando tudo referente aos motores elétricos utilizados e seu funcionamento, outra sobre os dispositivos restantes do sistema (controladores e atuadores) e por fim uma sobre os softwares implementados.

3.1 Máquinas Industriais

3.1.1 Motores Elétricos

Os motores elétricos utilizados para a execução deste trabalho já se encontravam presentes no GRACO antes mesmo da idealização deste projeto. Se tratam de 2 motores trifásicos, ambos utilizados anteriormente por outros alunos para estudos de vibração, com opções de tensão de entrada variando de 220V até 440V. Eles se encontravam parafusados à uma mesa, com seus cabos de alimentação soltos e preparados para encaixe. Abaixo na Fig. 3.10 é possível visualizar panoramicamente os motores.

Figura 3.10 – Motores elétricos do GRACO

O motor verde é um motor trifásico alemão da empresa VEM, uma produtora clássica de motores elétricos envolvida no desenvolvimento do motor elétrico desde 1886, que fornece equipamentos de acionamento elétrico em todo o mundo, compreendendo transformadores, conversores de frequência e conversores de potência juntamente com motores com desempenho variando de 0,06 kW a 42 MW para baixa e média tensão. As especificações técnicas referentes a características nominais e de desempenho deste motor, como tensão/corrente de operação, fator de rendimento, frequência, entre outros, podem ser visualizadas em sua placa de identificação, apresentada na Fig. 3.11.

Figura 3.11 – Especificações técnicas do motor da VEM

Já o motor azul é um motor trifásico da empresa WEG, uma das maiores fabricantes de equipamentos elétricos do mundo, especializada na fabricação e comercialização de motores elétricos, inversores de frequência, soft starters, controles, painéis, transformadores, geradores e tintas. Igualmente ao outro motor, suas especificações técnicas referentes a características nominais e de desempenho podem ser visualizadas em sua placa de identificação, apresentada na Fig. 3.12.

Figura 3.12 – Especificações técnicas do motor da WEG

3.1.2 Inversor de frequência CFW-08

O inversor de frequência utilizado neste trabalho é o WEG CFW-08, um inversor destinado para o acionamento de motores trifásicos na faixa de potência de 0,25 a 20cv. Ele é um inversor de frequência bastante consolidado no mercado com diversas aplicações e funcionalidades, devido à grande gama de funções deste produto. Para este contexto, o inversor será utilizado para realizar o controle preciso e adequado dos motores apresentados na Fig 3.10 e obter dados elétricos em tempo real via protocolo Modbus. Por padrão, o CFW-08 não possui formas de realizar comunicação serial utilizando o protocolo Modbus, sendo necessário adicionar um dispositivo opcional (WEG, 2009). Para realizar essa comunicação, é inserido na parte frontal do inversor o módulo de comunicação serial KRS-485, permitindo a conexão RS-485, como pode ser observado na Fig. 3.13.

Figura 3.13 – Inversor CFW-08 com módulo KRS-485
Fonte: https://www.commak.com.br/2020/10/02/weg-cfw08-1cv/ em 08/07/2023

A comunicação do protocolo é realizada por meio de mensagens (telegramas) contendo bits e registradores. É importante compreender primeiramente como o protocolo de comunicação Modbus RTU funciona para utilizá-lo, sendo que o mesmo é descrito em detalhes no Apêndice A, recomendando-se a sua leitura para maior entendimento. No entanto, o foco aqui será nas configurações de comunicação do CFW-08 e os métodos utilizados para acessar seus parâmetros, variáveis, registradores e bits internos. De forma simplificada, são disponibilizadas 7 funções para a elaboração dos telegramas, como pode ser visto na tabela 3.1 (WEG, 2009).

Função	Descrição	Código da Função	Tempo de Resposta
Read Coils	Leitura de bloco de bits internos ou bobinas.	01	10 a 20ms
Read Holding Registers	Leitura de bloco de registradores do tipo holding.	03	10 a 20ms
Write Single Coil	Escrita em um único bit interno ou bobina.	05	10 a 20ms
Write Single Register	Escrita em um único registrador do tipo holding.	06	10 a 50ms
Write Multiple Coils	Escrita em bloco de bits internos ou bobinas.	15	10 a 20ms
Write Multiple Registers	Escrita em bloco de registradores do tipo holding.	16	10 a 50ms
Read Device Identification	Identificação do modelo do inversor.	43	10 a 20ms

Tabela 3.1 – Funções Modbus disponíveis no CFW-08

No CFW-08, tantos parâmetros quanto variáveis básicas são definidos como registradores do tipo holding, dado que no geral se tratam de valores inteiros e não binários. São diversos registradores e bits existentes, cujo o significado de cada um pode ser identificado no manual do inversor (WEG, 2009), sendo necessário somente detalhar os que interessam este trabalho no capítulo de desenvolvimento. Em relação ao endereçamento dos dados, o manual do inversor explica detalhadamente como definir os seus valores, ilustrados nas tabelas 3.2 para os parâmetros, 3.3 para as variáveis básicas e 3.4 para os bits de estado e de comando:

> (WEG, 2009) O endereçamento dos dados no CFW-08 é feito com offset igual a zero, o que significa que o número do endereço equivale ao número dado. Os parâmetros são disponibilizados a partir do endereço 0 (zero), enquanto que as variáveis básicas são disponibilizadas a partir do endereço 5000. Da mesma forma, os bits de estado são disponibilizados a partir do endereço 0 (zero) e os bits de comando são disponibilizados a partir do endereço 100.

Parâmetros		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
P000	0	0000h
P001	1	0001h
⋮	⋮	⋮
P100	100	0064h
⋮	⋮	⋮

Tabela 3.2 – Endereçamento dos Parâmetros

Variáveis Básicas		
Número da Variável Básica	Endereço Modbus	
	Decimal	Hexadecimal
V00	5000	1388h
V01	5001	1389h
⋮	⋮	⋮
V05	5005	138Dh

Tabela 3.3 – Endereçamento das Variáveis Básicas

Bits de Estado		
Número do Bit	Endereço Modbus	
	Decimal	Hexadecimal
Bit 0	00	00h
Bit 1	01	01h
⋮	⋮	⋮
Bit 7	07	07h
Bits de Comando		
Número do Bit	Endereço Modbus	
	Decimal	Hexadecimal
Bit 100	100	64h
Bit 101	101	65h
⋮	⋮	⋮
Bit 107	107	6Bh

Tabela 3.4 – Endereçamento dos Bits de Estado e de Comando

3.1.3 Conversor USB/Serial RS-485

Como pode ser visualizado na Fig. 3.13, a saída dos dados via RS-485 não possui conexão direta com computadores ou controladores, visto que se trata de uma conexão de 2 fios (um fio de dados e outro fio com dados invertidos). É necessário então a utilização de um conversor para a leitura e tradução das informações provenientes do inversor. Para isso, um simples conversor USB para RS-485 (Fig. 3.14) foi adquirido para permitir a comunicação do CFW-08 com o computador.

Figura 3.14 – Módulo Conversor USB para RS-485
Fonte: https://www.marinostore.com/arduino/modulos/modulo-conversor-usb-para-rs485/ em 08/07/2023

3.2 Controladores e Atuadores

Apesar do inversor de frequência conseguir disponibilizar diversas informações dos motores, existem outros dados que não são possíveis de serem obtidos por meio do CFW-08. A próxima etapa do projeto foi a de definir quais seriam os controladores e atuadores que seriam utilizados para capturar estes valores. Para isso, primeiramente, era necessário determinar quais informações além das que o inversor fornece seriam necessárias para a construção do modelo virtual de um motor. Após diversas pesquisas, baseando-se nas normas da ABNT para máquinas elétricas girantes (ABNT, 2018) e no Guia de Especificação de Motores Elétricos da WEG (WEG, 2023b), chegou-se à conclusão de que 3 informações importantes para a modelagem do motor estavam faltando, sendo elas a temperatura externa da carcaça, o ruído emitido pelo motor e a vibração nos 3 eixos. Com base nisso, os materiais a seguir foram definidos para a aquisição desses dados.

3.2.1 Microcontrolador ESP32

O microcontrolador ESP32 (Fig. 3.15) criado e desenvolvido pela empresa chinesa Espressif Systems é o controlador escolhido para o desenvolvimento do sistema de aquisição de dados dos motores. Optou-se por utilizá-lo devido à facilidade de uso, o baixo custo, baixo consumo de energia, pequeno porte, grande eficiência e a possibilidade de comunicação sem fio através do Wifi e do próprio sistema Bluetooth, somado ao fato de já existir uma familiaridade com o dispositivo e exemplares físicos à disposição para início de testes. O ESP32 é um dispositivo muito versátil que pode ser usado em uma variedade de aplicações, como IoT, robótica, automação residencial/industrial, tecnologias vestíveis, e que suporta diversas linguagens de programação, como C, C++, Python e MicroPython. A sua programação foi realizada na IDE (Integrated Development Enviroment) do Arduino, a qual será mais detalhada na seção de softwares.

Figura 3.15 – Microcontrolador ESP32
Fonte: www.vidadesilicio.com.br/produto/esp32-esp-wroom-32-nodemcu em 08/07/2023

Vários outros controladores foram considerados e avaliados para essa escolha, como os CLPs de diversas marcas e outros microcontroladores (a citar o RaspBerry Pi3 e o Arduino). Os CLPs foram descartados devido à complexidade do sistema. Levando em conta que há somente 3 informações a serem capturadas, sem a necessidade de utilizar muitos sensores, um simples microcontrolador é capaz de obter esses dados e transmití-los ao sistema supervisório. Utilizar um CLP neste caso aumentaria a complexidade e principalmente o custo do sistema de forma desnecessária.

Já em relação aos outros microcontroladores disponíveis no mercado, o RaspBerry Pi3 analogamente aos CLPs possui um custo e uma complexidade elevados, onde sua única vantagem seria a de poder implementar a programação dos softwares diretamente em seu sistema operacional. Entretanto, presumindo-se que em qualquer chão de fábrica já possua computadores para realizar comandos, a comunicação dos microcontroladores localizados nos motores e o computador é uma solução mais viável que utilizar um RaspBerry Pi3 para executar a programação e disponibilizar os dados em uma tela à parte. Outra opção seria utilizar uma versão de um Arduino (Nano, UNO, Mega). Essa seria uma opção mais vantajosa financeiramente e em questão à complexidade do sistema não haveria muitas mudanças. Porém, uma das vantagens do ESP32 é exatamente a possibilidade de comunicação sem fio através do Wifi e do próprio sistema Bluetooth, o que já não é possível no Arduino sem a utilização de módulos à parte, aumentando o custo para o equivalente a um ESP32 com uma versão bem menos compacta e mais propensa a problemas físicos (cabos, protoboard). Igualmente outros microcontroladores foram avaliados, porém optou-se pelo ESP32 ao final.

3.2.2 Sensores de temperatura, ruído e vibração

Os atuadores utilizados no trabalho para a obtenção da temperatura externa da carcaça, do ruído emitido pelo motor e da vibração nos 3 eixos foram, respectivamente, o sensor de temperatura DS18B20, o sensor de som KY-038 e o acelerômetro e giroscópio MPU-6050. Os motivos de suas escolhas e mais detalhes são apresentados nos itens a seguir.

3.2.2.1 Sensor de Temperatura DS18B20

A medição da temperatura externa da carcaça é de extrema importância para o motor visto que o isolamento do seu enrolamento possui um grande impacto em sua expectativa de vida e confiabilidade. De acordo com a norma NBR 17094-1 (ABNT, 2018), todo motor elétrico deve possuir uma classificação térmica atribuída ao seu sistema de isolação conforme as classes térmicas designadas na Tabela 1 da norma IEC 60085 (ABNT, 2017). O objetivo dessas classificações é descrever a capacidade do isolamento do enrolamento do motor para lidar com o calor e essa informação normalmente é apresentada na placa de identificação do motor. Seguindo as normas e analisando as placas das Figs. 3.11 e 3.12, identificou-se que ambos os motores possuem classe de isolamento do tipo B.

A partir das especificações dos subitens 6.3 e 6.4 da norma NBR 17094-1 (ABNT, 2018), a faixa de valores permitido para a temperatura do ar ambiente no local de funcionamento é de 0°C a 40°C. A classe de isolamento do tipo B possui algumas características, como aumento de temperatura máxima de 80°C e permissão de temperatura excessiva de ponto quente de 10°C. Dado que a temperatura máxima do enrolamento é a soma da temperatura máxima ambiente e o aumento de temperatura permitido (aumento de temperatura máxima mais um ponto quente sobre a permissão de temperatura), a temperatura máxima permitida para esta classe é de 130°C, sendo ideal atingir somente no máximo 120°C . Com base em todas essas informações, escolheu-se o sensor de temperatura DS18B20 (Fig. 3.16), um sensor digital que realiza medições na faixa de -55° a 125°C, para realizar a medição da temperatura. Ele trabalha na faixa de temperatura que procura-se analisar, onde ao se atingir 120°C já se pode considerar com um valor alarmante e ele possui uma capacidade de fornecer leituras com resolução de 9-bit a 12-bit, sendo bastante preciso com seus dados.

Figura 3.16 – Sensor de Temperatura DS18B20
Fonte: www.huinfinito.com.br/sensores/931-sensor-de-temperatura-ds18b20-prova-d-agua.html em 08/02/2023

3.2.2.2 Módulo Sensor de Som KY-038

A norma NBR 7565 (ABNT, 1982) especifica os níveis máximos de potência sonora para motores elétricos em decibéis. Esta norma se aplica para máquinas de uma única rotação, com velocidade não superior a 3750 RPM, potência inferior a 6300 kW e método de resfriamento IC01 e IC41, características as quais ambos os motores utilizados atendem. Seguindo a norma, podemos definir o limite de ruído a partir do valor da potência e da velocidade nominais do motor em análise. Primeiramente em relação ao motor da VEM, dado que ele possui uma potência nominal menor do que 1,1kW/1,5cv e uma velocidade nominal de 1696 RPM, o seu limite de ruído é de 78dB ou 80dB a depender do grau de proteção (IP22 ou IP44). Já para o motor da WEG, visto que ele possui valores nominais maiores de 3cv e 1700 RPM, o seu limite de ruído também é um pouco maior, igual a 81dB ou 83dB.

Conhecendo-se o maior valor de ruído possível (83dB), escolheu-se o sensor de som KY-038 (Fig. 3.17) para medir a intensidade sonora do motor. Este sensor é um pequeno módulo eletrônico composto por um chip LM393, trimpot e microfone, capaz de determinar níveis de ruído dentro de 100dB a 3 kHz e faixa de frequências de 6 kHz, aproximadamente. Quando detecta som, altera o sinal de saída, informando ao microcontrolador conectado o que está acontecendo e agindo conforme pré-programado pelo usuário. O trimpot permite ajustar a sensibilidade do trigger e pode ser usado tanto com as portas digitais quanto analógicas. Nas portas digitais, a saída digital será de nível alto quando nenhum som for detectado e será de nível baixo quando o som for detectado. Já a porta analógica do sensor é usada com um microcontrolador que incorpora um conversor analógico-digital para medir mudanças de som.

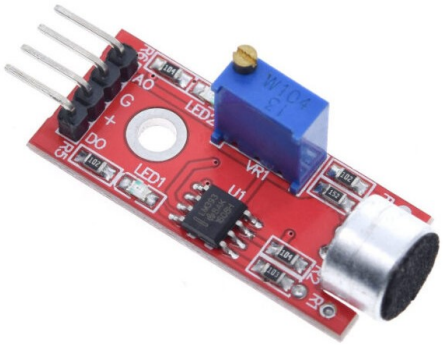


Figura 3.17 – Módulo Sensor de Som KY-038
Fonte: https://dipollo.com.br/loja/product/modulo-sensor-som-ky-038-2/ em 08/07/2023

3.2.2.3 Acelerômetro e Giroscópio MPU-6050 GY-521

As normas NBR 11.390 (ABNT, 1990) e IEC 60.034-14 (ABNT, 2011) especificam os limites recomendados para severidade de vibração. Esta norma se aplica para máquinas de CC ou trifásicas CA com altura de eixo igual ou superior a 56mm e potência nominal de saída até 50MW, características as quais ambos os motores utilizados atendem. Seguindo a norma, podemos definir os limites de vibração nos 3 eixos em função da altura do eixo, do tipo de montagem do motor em análise e do grau de exigência de vibração. Analisando os motores, quanto ao tipo de montagem e ao grau de exigência de vibração, ambos possuem uma montagem de suspensão livre e grau normal. Já quanto à altura do eixo, o motor da VEM possui uma altura de 120mm e o motor da WEG possui uma altura de 135mm. Baseado nesses valores, a seguinte tabela apresenta os limites de magnitude de vibração em deslocamento, velocidade e aceleração eficaz (ou r.m.s):

Motor	Deslocamento	Velocidade	Aceleração
VEM	25 μm	1,6 mm/s	2,5 m/s^2
WEG	35 μm	2,2 mm/s	3,5 m/s^2

Tabela 3.5 – Limites de magnitude de vibração em deslocamento, velocidade e aceleração eficaz

Baseando-se nos valores apresentados na tabela 3.5, escolheu-se o acelerômetro e giroscópio MPU-6050 GY-521 (Fig. 3.18) para medir a aceleração e rotação nos 3 eixos do motor. O módulo contém em um único chip um acelerômetro e um giroscópio tipo MEMS, cada um com 3 eixos totalizando 6 graus de liberdade. Somado a isso, ele possui alta precisão devido ao conversor analógico digital de 16-bits para cada canal, capturando os canais X, Y e Z ao mesmo tempo. A faixa de valores em que o acelerômetro trabalha é de ±2, ±4, ±8, ±16g, sendo g o valor da aceleração gravitacional (aproximadamente 9,81m/s^2), englobando os valores de aceleração dos limites de magnitude de vibração. Já a faixa de valores que o giroscópio trabalha é de ±250, 500, 1000, 2000°/s. O MPU-6050 possui 8 pinos (como mostra na imagem abaixo), sendo 2 para alimentação e 6 para comunicação.

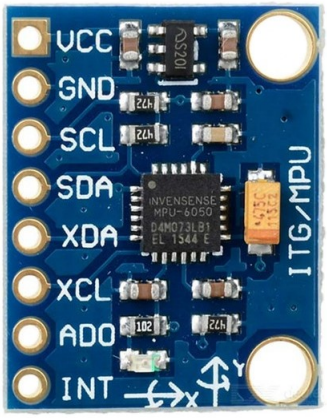


Figura 3.18 – Acelerômetro e Giroscópio 3 Eixos MPU-6050 GY-521
Fonte: https://portuguese.alibaba.com/product-detail/Smart-Bes-MPU-6050-module-triaxial-60495381990.html em 08/07/2023

3.3 Softwares

Todo hardware e instrumentação física necessita de softwares para atuar como intermediário entre os controladores e os atuadores, realizando a comunicação entre esses dispositivos e permitindo coordenação perfeita/controle preciso do sistema de aquisição de dados. Os softwares são responsáveis por receber sinais de entrada do controlador e os traduzir em comandos que os atuadores podem entender e executar, determinando suas ações ou movimentos desejados. De forma a facilitar o acesso às informação obtidas pelo inversor e pelos sensores, também foi desenvolvido um sistema SCADA para a monitoração dos dados, implementando no processo alguns recursos úteis como registro de eventos, alarmes e gráficos. O software deste sistema é uma ferramenta das estações de monitoração central para a análise e o controle dos processos. No caso deste trabalho, a IDE do Arduino foi o software escolhido para a programação dos dispositivos aqui citados e o ScadaBR para o desenvolvimento do sistema supervisório. Os motivos de suas escolhas e mais detalhes são apresentados nos itens a seguir.

3.3.1 Arduino IDE

A IDE (Integrated Development Enviroment) do Arduino é um software de código aberto que serve para auxiliar desenvolvedores que trabalham em projetos Arduino na escrita de códigos e no upload para qualquer placa, desempenhando um papel crucial na simplificação do desenvolvimento do Arduino, tornando-o acessível tanto para iniciantes quanto para desenvolvedores experientes. Ele possui uma interface amigável e uma ampla gama de recursos que agilizam o fluxo de trabalho de desenvolvimento, incluindo recursos de edição e compilação de código, recursos de gerenciamento de placa e biblioteca, ferramenta de monitor serial e funcionalidade de upload de código sem esforço, podendo ser utilizado com as linguagens C/C++ e simplificando significativamente o processo de criação, compilação e upload de códigos na placa (ARDUINO, 2023).

Aprofundando a respeito dos recursos que este software oferece, os seus recursos de edição de código incluem destaque de sintaxe, preenchimento automático e destaque de erro, aprimorando a legibilidade do código e auxiliando na detecção de erros durante a fase de desenvolvimento, reduzindo assim o tempo de depuração. Em relação à compilação de códigos, o IDE permite que os desenvolvedores garantam a integridade de seu código antes de carregá-lo ao compilar código de forma eficiente, transformando-o em um formato binário que pode ser entendido pelas placas Arduino. Na hora de carregar o código compilado, basta conectar o dispositivo ao computador, selecionar a placa correta e as configurações de porta e clicar no botão de upload. O software também conta com um gerenciador de placa integrado que simplifica a instalação e o gerenciamento de configurações específicas da placa, e uma ferramenta de monitor serial que permite a comunicação em tempo real com placas Arduino através da porta serial. Os programadores podem usar esse monitor serial para receber dados da placa, monitorar as leituras de sensores e depurar seu código imprimindo informações relevantes. Além disso, o IDE fornece um gerenciador de bibliotecas que facilita o processo de incorporação de módulos de código pré-escritos, conhecidos como bibliotecas. É possível facilmente pesquisar, instalá-las com alguns cliques e mantê-las atualizadas. Caso necessário, também é possível integrar manualmente bibliotecas externas (ARDUINO, 2023).

Apesar de ser um software dedicado para o Arduino, ele também funciona com outros hardwares incluindo o ESP32. Para isso, basta adicionar o suporte para hardware de terceiros no diretório de hardware do seu diretório de sketchbook. As plataformas instaladas podem incluir definições de placa, bibliotecas principais, gerenciadores de inicialização e definições de programador. Algumas bibliotecas no próprio IDE já possuem esse suporte configurado, sendo somente necessário instalá-las e alterar a placa no gerenciador de placas, como é o caso do ESP32. Somado a isso, este software possui bibliotecas que permitem a comunicação dele com outros dispositivos via Modbus e que facilitam a leitura dos sensores mencionados no item 3.2.2. Devido a este fato e a todos os recursos e facilidades mencionados acima, optou-se por utilizar este software para a programação do microcontrolador e de seus atuadores.

3.3.2 ScadaBR

O ScadaBR é um software livre, gratuito e de código-fonte aberto, para desenvolvimento de aplicações de Automação, Aquisição de Dados e Controle Supervisório. Ele foi desenvolvido em 2006 por iniciativa da MCA Sistemas (atualmente Sensorweb), apoiado por outras empresas e fundações, com o objetivo de fornecer todas as funcionalidades de um sistema SCADA tradicional, entre elas recursos de aquisição de dados, painéis personalizáveis e exibições gráficas, visualização em tempo real, gerenciamento de alarmes, análise de dados históricos e acesso remoto. Seu público-alvo inclui profissionais de automação, universidades, escolas técnicas e empresas de todos os tamanhos, que precisam controlar máquinas por meio de um computador, implementar lógicas de automação ou simplesmente visualizar dados de sensores, ambientes e processos industriais. O software também é adequado para entusiastas e inventores permitindo que eles controlem uma ampla variedade de instrumentos e acessórios em um único sistema, de qualquer lugar (SCADABR, 2023).

A natureza de código aberto do ScadaBR é uma vantagem importante que o diferencia dos sistemas SCADA proprietários. Tradicionalmente, estes softwares custam na faixa de algumas dezenas ou centenas de milhares de reais, valores inviáveis para pequenas automatizações, profissionais autônomos, micro-empresas e utilização acadêmica em geral. Ao oferecer um software completo gratuitamente, de fácil utilização e praticamente com as mesmas funcionalidades, o ScadaBR se destaca entre os demais sistemas do mercado. Somado a isso, o software livre pode ser adaptado e estendido para atender a requisitos específicos e integrar-se perfeitamente aos sistemas existentes. As indústrias têm a liberdade de modificar o software, desenvolver funcionalidades adicionais e compartilhar melhorias com a comunidade. Essa flexibilidade permite uma solução altamente adaptável e escalável que se alinha precisamente com as necessidades do setor.

O ScadaBR possui hoje um público estimado em mais de 10 mil usuários. Isso se deve não somente ao fato de ser um software aberto, mas sim pela continuidade em seu desenvolvimento e pelo investimento na comunidade do projeto ao longo de todos estes anos. O site oficial (SCADABR, 2023) oferece manuais, cursos, informações, cases e um fórum público de discussão com mais de 4.000 posts e mais de 6500 usuários, tudo em português. Especificamente em relação a ferramentas, além das funções típicas de softwares SCADA, com destaque para o protocolo Modbus, o ScadaBR também possui suporte para outros protocolos de comunicação, como OPC, IEC101 e DNP3, e possui uma API em web-services que interage com ele em qualquer linguagem de programação. O software roda em qualquer sistema operacional e possui uma interface 100% web que permite acessar e controlar dispositivos e processos através de computadores, tablets e smartphones, sendo bem versátil. Devido a todas essas características e visando perpetuar o trabalho aqui desenvolvido no Windows para outros sistemas operacionais, devido a sua versatilidade de exportação e manipulação de dados, optou-se por utilizar este software para o sistema supervisório.

4 Desenvolvimento

O desenvolvimento deste trabalho se baseia na composição das 4 vertentes tecnológicas dos gêmeos digitais apresentadas na Fig. 2.4. Dentre estas 4 vertentes, 3 foram estudadas e trabalhadas para a construção do modelo digital dos motores elétricos: a IoT, AI e Nuvem. Primeiramente focou-se na obtenção de dados e comunicação entre os dispositivos e o usuário para análise destas informações. Em seguida, em conjunto com um trabalho de mestrado desenvolvido por Douglas Andrade, procurou-se alimentar o banco de dados relacional de sua plataforma de inteligência com os dados obtidos pelos equipamentos. Ela permite que o motor seja um agente inteligente dentro do sistema de manufatura, compartilhando ativamente dados e buscando informações. Dinamicamente, os dados coletados podem ser armazenados localmente ou remotamente na Nuvem, de acordo com sua criticidade. O fluxograma geral do sistema desenvolvido segue demonstrado na Fig. 4.19. Ele é subdividido em 3 sistemas: o sistema de aquisição de dados, sistema supervisório e o sistema de inteligência.

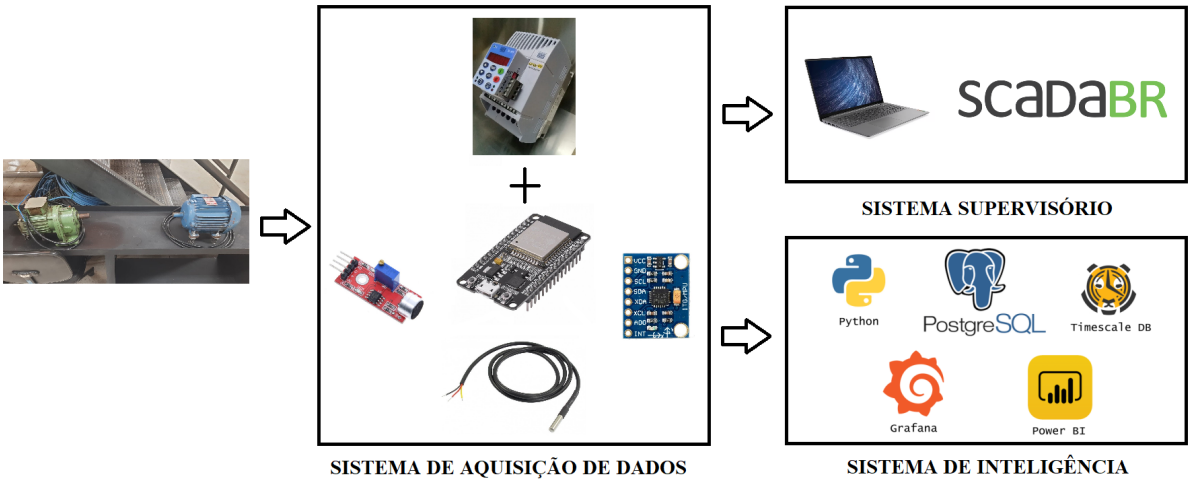


Figura 4.19 – Fluxograma geral do sistema desenvolvido

Todos os passos realizados para o desenvolvimento do Gêmeo Digital, desde a sua concepção até os resultados obtidos, serão descritos a seguir de forma a facilitar a compreensão/análise do sistema criado e possibilitar a perpetuação ou reprodução do projeto aqui disposto. Este capítulo será igualmente dividido em 3 seções referentes aos 3 subsistemas diferentes na confecção do sistema deste trabalho: uma seção abordando o sistema de aquisição de dados, sua configuração, montagem e programação; outra sobre o sistema supervisório desenvolvido para análise e monitoramento das variáveis dos motores, apresentando uma visão geral sua, a definição dos *data sources* e alarmes e a representação gráfica das informações; e por fim um sobre um sistema de inteligência desenvolvido pelo Douglas, explicando a sua integração com os outros sistemas e mais detalhes sobre a plataforma/os softwares utilizados.

4.1 Sistema de Aquisição de Dados

O sistema de aquisição de dados é composto pelo inversor de frequência CFW-08 e o microcontrolador ESP32 conectado aos sensores DS18B20, KY-038 e MPU-6050 que captam diversas informações relacionadas ao status do motor e dos próprios dispositivos, incluindo corrente de saída, tensão de saída, velocidade, temperatura do dissipador, parâmetros nominais do motor, temperatura da carcaça, ruído, vibração, entre outros. Primeiramente, o foco deste sistema foi o entendimento do modo de funcionamento do inversor de frequência, seguindo o manual (WEG, 2009), e a parametrização do próprio. Posteriormente, com o inversor de frequência já configurado e funcional, preocupou-se em desenvolver o circuito conectando os sensores ao microcontrolador e a programação deles. Os resultados obtidos são apresentados a seguir.

4.1.1 Acionamento via Inversor de Frequência

Os motores elétricos, primordialmente, devem estar energizados e acionados para ser possível realizar qualquer outra ação sobre eles com o objetivo de analisá-los. O primeiro passo, então, é alimentar e conectar corretamente o inversor de frequência e os motores de forma a permitir que ele realize partida do motor. Os procedimentos de instalação elétrica e mecânica do CFW-08 para o acionamento dos motores elétricos seguem as instruções apresentadas no seu manual. De acordo com ele, a rede de alimentação CA e a conexão de alimentação do motor são realizadas nos bornes de potência L/L1, N/L2, L3, U, V e W, apresentados na Fig. 4.20. Os três primeiros bornes são referentes à tensão de entrada e os três últimos à tensão de saída. Outros bornes podem ser encontrados em outros modelos para alimentar o inversor com tensão CC, porém não convém ao caso.

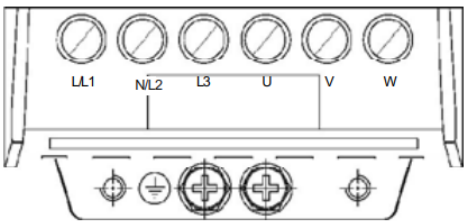


Figura 4.20 – Bornes de potência do modelo utilizado do CFW-08
Fonte: (WEG, 2009)

Ainda de acordo com o manual, os modelos da linha de tensão 200-240V, exceto alguns, podem operar em 2 fases (monofásico) sem redução da corrente nominal. Neste caso, a tensão de alimentação CA poderia ser conectada em 2 quaisquer dos 3 terminais de entrada (idealmente nas entradas L-fase e N-neutro). Entretanto, o modelo do CFW-08 utilizado no Graco é o 2,6A/380-480V, sendo necessário alimentar o inversor com uma tensão de entrada trifásica de 380V. Portanto, a conexão dos bornes de potência e do aterramento para o acionamento dos motores via o inversor de frequência é apresentada na Fig. 4.21.

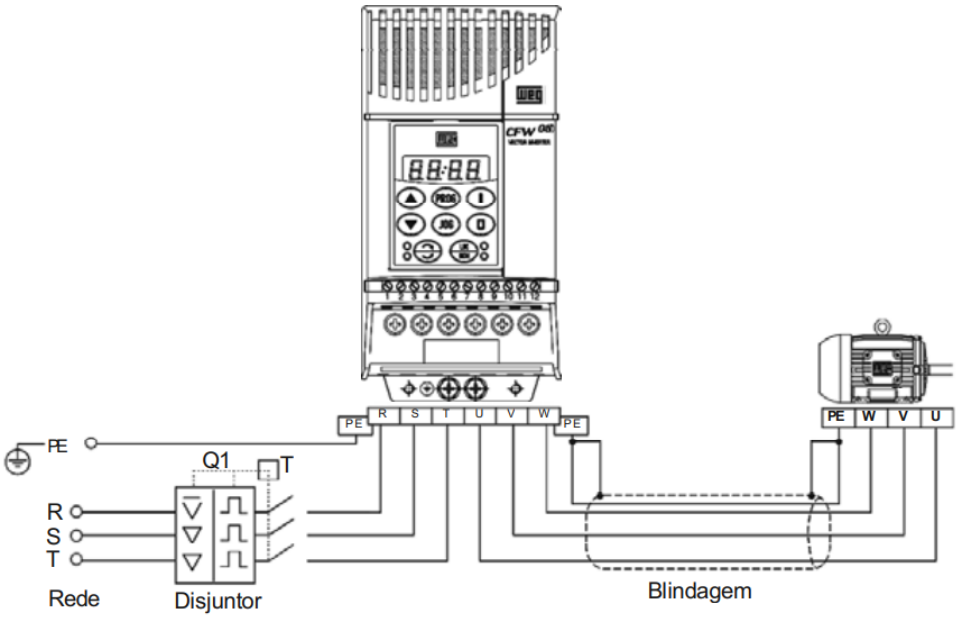
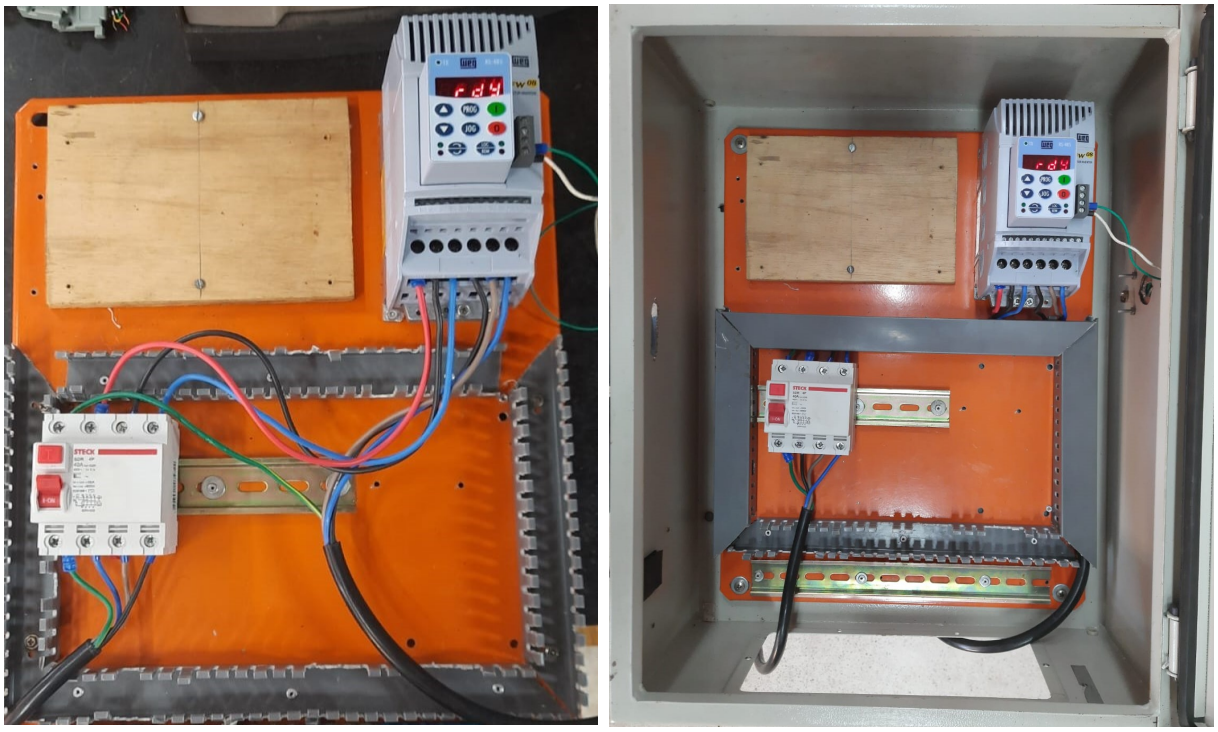


Figura 4.21 – Conexões de potência e aterramento do modelo utilizado do CFW-08
Fonte: (WEG, 2009)

Seguindo as instruções apresentadas e a orientação visual apresentada na Fig. 4.21, o resultado final e funcional obtido da alimentação de toda esta parte do sistema pode ser observado nas Figs. 4.22a e 4.22b. A Fig. 4.22a mostra de forma clara a conexão dos cabos a título de demonstração. Depois de conectados, o procedimento correto é organizar os cabos e não deixá-los expostos, como apresentado na Fig. 4.22b.



(a) Cabos à mostra

(b) Instalado e organizado

Figura 4.22 – Conexão final e funcional do inversor de frequência

4.1.2 Parametrização do CFW-08

O inversor de frequência CFW-08 deve estar corretamente parametrizado para realizar o acionamento e o controle do motor específico conectado a ele, seguindo as suas especificações técnicas, e para funcionar na rede Modbus RTU. Outras configurações de controle de velocidade e de torque e configurações do próprio inversor também podem ser alteradas e personalizadas por meio da parametrização. Inicialmente, todos os parâmetros foram ajustados manualmente por meio da IHM frontal do inversor seguindo as orientações e os valores apresentados no manual do CFW-08. Porém, antes de realizar qualquer ajuste, é necessário colocar o valor do parâmetro P000 em 5 para liberar o acesso à alteração do conteúdo dos restantes. Também é importante colocar o inversor no modo de controle vetorial, ajustando o valor do parâmetro P202. A sua vantagem é que a operação do motor é otimizada, pois permite um melhor desempenho em termos de torque e regulação de velocidade.

Em um primeiro momento, procurou-se habilitar e ajustar o protocolo de comunicação serial para fazer o inversor operar na rede Modbus. Os parâmetros P308, P312, P313 e P314 são os responsáveis pela configuração da Interface Serial, apresentados na tabela 4.6 com os ajustes realizados. O parâmetro P308 possui a função de determinar o endereço serial do CFW-08, ou também conhecido como ID do dispositivo na rede de comunicação. Ele pode assumir os valores de 1 a 247 para determinar o endereço do inversor caso a comunicação seja via o protocolo Modbus RTU (modelo de comunicação do tipo mestre-escravo, citado no Apêndice A), ou de 1 a 30 caso seja via o protocolo Serial WEG. Já o parâmetro P312 possui a função de determinar o protocolo de comunicação utilizado no inversor. Ele pode assumir os valores de 0 a 9 a depender do tipo de rede que se deseja configurar. O protocolo de comunicação pode ser uma das duas citadas, e caso seja Modbus RTU, a sua taxa de transmissão e paridade podem assumir diferentes valores. A rede Modbus do inversor foi configurada utilizando uma taxa de transmissão de 9600 bps sem paridade, equivalente ao valor de número 1 na parametrização do inversor. Por fim, os parâmetros P313 e P314 são relacionados ao *Watchdog*, uma função que alguns nós Modbus possuem e tipicamente é um contador que faz uma contagem regressiva em um determinado intervalo de tempo. Se o *Watchdog* puder chegar a 0, as saídas físicas do nó serão definidas para um valor predefinido. Esse mecanismo pode ajudar a garantir que, quando a comunicação com o equipamento físico for perdida, o equipamento pare com segurança, porém não é uma prioridade deste trabalho e não foram realizados nenhum ajuste neles.

Parâmetro	Função	Ajuste
P308	Endereço Serial	1
P312	Protocolo da Interface Serial	1
P313	Ação do Watchdog	-
P314	Tempo de Atuação do Watchdog	-

Tabela 4.6 – Parâmetros da Comunicação Serial

Posteriormente, procurou-se regular a aceleração e a velocidade dos motores por questões de segurança. Os parâmetros P100, P101, P133 e P134 são os responsáveis pela limitação dos valores máximos e mínimos de rotação do motor, apresentados na Tabela 4.7 com os ajustes realizados. Os parâmetros P100 e P101 possuem respectivamente a função de determinar o tempo de aceleração e desaceleração do motor, ou seja, o tempo em que o inversor demora para atingir a velocidade máxima na saída ou retornar para a velocidade mínima. Já os parâmetros P133 e P134 possuem respectivamente a função de determinar o valor mínimo e máximo de operação da frequência. Ao definir estes limites, consequentemente também se define para a velocidade visto que ela depende da frequência multiplicada por um fator de escala, garantindo que o motor opere a uma velocidade mínima para evitar falhas e opere a uma velocidade máxima para evitar desgastes e diminuição da vida útil.

Parâmetro	Função	Ajuste
P100	Tempo de Aceleração	5 s
P101	Tempo de Desaceleração	10 s
P133	Freqüência Mínima (F_{min})	3 Hz
P134	Freqüência Máxima (F_{max})	66 Hz

Tabela 4.7 – Parâmetros de Regulação

Por fim, procurou-se ajustar os parâmetros relacionados ao motor elétrico em uso, utilizando as informações contidas nos dados de placa do motor e aqueles obtidos pela rotina de Auto-Ajuste. Os parâmetros P399, P400, P401, P402, P403, P404 e P407 são os responsáveis pela configuração dos parâmetros nominais do motor, apresentados na Tabela 4.8 com os ajustes realizados. Esta parametrização baseou-se nos dados do motor da VEM, de acordo com a Fig. 3.11 para uma tensão nominal igual a 380V. Entretanto, caso se deseje realizar o controle do motor da WEG, basta alterar os valores de acordo com a Fig. 3.12. A função e o ajuste de cada um destes parâmetros são auto-explicativos observando-se a Tabela 4.8, com exceção do parâmetro P404. Este parâmetro, especificamente, pode assumir valores de 0 a 17 a depender da potência nominal do motor em CV, HP ou kW. Como na placa do motor da VEM os valores apresentados são 0,5kW/0,65HP, fora da faixa de opções do inversor, optou-se pela opção respectiva mais próxima possível. No caso, colocou-se o valor do parâmetro P404 em 3, equivalente à uma potência nominal de 0,5CV/0,5HP/0,37kW. Vale destacar que esses parâmetros só podem ser alterados com o motor parado.

Parâmetro	Função	Ajuste
P399	Rendimento Nominal do Motor	71,4%
P400	Tensão Nominal do Motor	380 V
P401	Corrente Nominal do Motor	1,43 A
P402	Velocidade Nominal do Motor	1696 RPM
P403	Frequência Nominal do Motor	60 Hz
P404	Potência Nominal do Motor	3
P407	Fator de Potência Nominal do Motor	0,72

Tabela 4.8 – Parâmetros do Motor

4.1.3 Conexão entre o ESP32 e os sensores

A respeito da leitura de dados realizada pelo ESP32 e os sensores DS18B20, KY-038 e MPU-6050, é necessário, a princípio, que todos os dispositivos estejam integrados e corretamente conectados para funcionarem e serem reconhecidos pelo sistema. O processo de construção do circuito consistiu em analisar as especificações dos pinos do microcontrolador e dos sensores, adquirir outros materiais necessários, realizar a fiação entre os componentes e instalá-los na posição adequada dos motores. Idealmente, cada motor deve possuir um circuito independente, visto que a monitoração dos dados é realizada remotamente e os motores se localizariam em posições distintas dentro de uma fábrica, impossibilitando compartilharem o mesmo microcontrolador.

Começando pelo ESP32, como já citado na seção 3.2.1, ele é um módulo NodeMcu com Wi-Fi, Bluetooth e Bluetooth de baixo consumo. De tal forma, a placa possui uma ampla variedade de aplicações e oferece grande desempenho para integração eletrônica, alcance, consumo de energia e conectividade. Ela possui um rico conjunto de periféricos, que inclui sensor de toque capacitivo, sensor de Efeito Hall, interface de cartão SD, Ethernet, SPI de alta velocidade, UART, I2S e I2C. Aprofundando na pinagem em si, ela possui 25 pinos GPIOs (*General Purpose Input Output*) que podem funcionar em sua maioria como entrada ou saída digital simples, canais analógicos de 12 bits ou como algum dos periféricos citados. Todas as possíveis funções e a pinagem completa do ESP32 pode ser observada na Fig. 4.23.

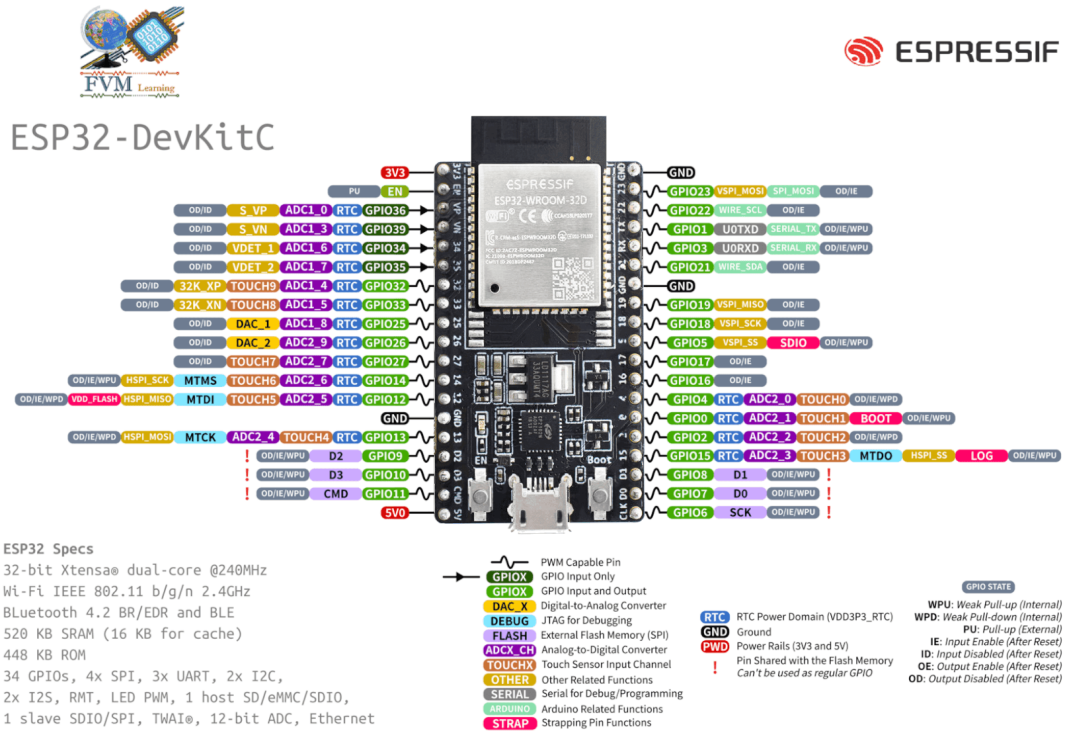


Figura 4.23 – Pinagem do ESP32
Fonte: www.fvml.com.br/2022/04/pinagem-pinout-esp32-devkitc.html em 08/07/2023

Prosseguindo para os sensores, o DS18B20 é um sensor de temperatura digital de um fio, ou seja, que requer apenas uma linha de dados (além do neutro) para se comunicar com o ESP32. Ele possui 3 terminais (Fig. 4.24) e fornece medições de temperatura em graus Celsius. O sensor pode ser alimentado por uma fonte de alimentação externa pelo pino VCC (chamado modo normal) ou pode derivar energia da linha de dados (chamado "modo parasita"), o que elimina a necessidade de uma fonte de alimentação externa. Para a comunicação de um fio, é necessário adicionar um resistor *pull-up* de 4,7kΩ entre o pino DATA e o pino VCC. No circuito montado, seguiu-se o modo normal e conectou-se o pino de dados ao GPIO 5 do ESP32, com o resistor pull-up adicionado.

Figura 4.24 – Pinagem do DS18B20
Fonte: https://www.electronicwings.com/esp32/ds18b20-sensor-interfacing-with-esp32/ em 08/07/2023

Já o KY-038 é um módulo eletrônico que possui 4 terminais (VCC, GND, A0 e D0), como pode ser observado na Fig. 4.25, desenvolvido com a finalidade de detectar/medir variações de som em um ambiente. O pino AO é usado como saída para leitura analógica e o o pino DO como saída para leitura digital. Ao detectar som, o microfone varia a tensão na saída analógica, e também aciona a saída digital conforme a regulagem do seu potenciômetro. Como é de interesse medir a variação do som do motor (ruído por ele gerado), foi-se utilizado no circuito somente a saída analógica, conectando o pino A0 ao GPIO 4 do ESP32.

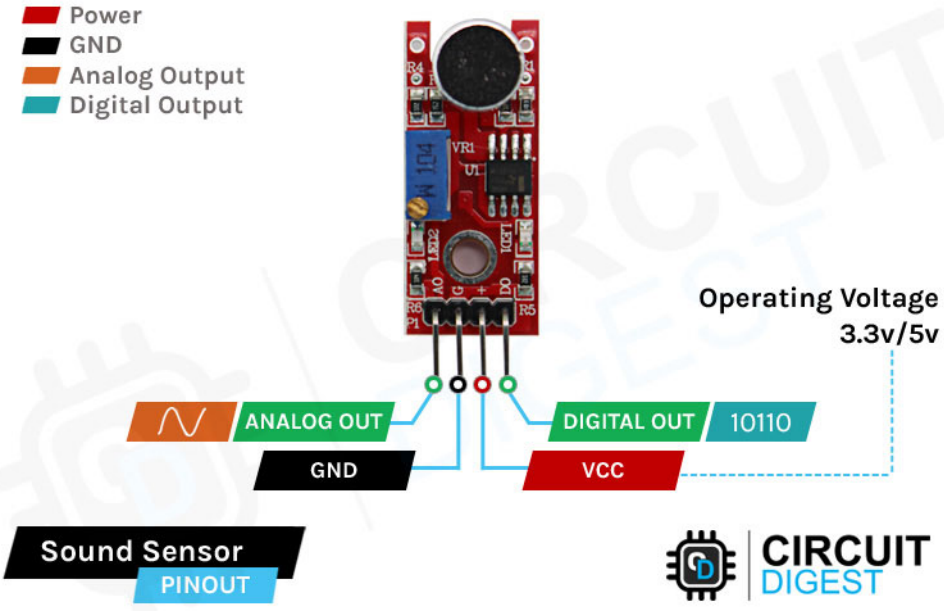


Figura 4.25 – Pinagem do KY-038
Fonte: https://circuitdigest.com/microcontroller-projects/interface-ky038-sound-sensor-with-esp32 em 08/07/2023

Finalizando com o MPU-6050, como já citado na seção 3.2.2.3, ele possui um giroscópio de 3 eixos e um acelerômetro de 3 eixos integrados em um único chip. Sua comunicação com o microcontrolador é realizado por meio do protocolo I2C, um protocolo de comunicação serial cuja transmissão de dados utiliza apenas duas linhas, o Serial Data (SDA) e o Serial Clock (SCL). O SDA é responsável de enviar e receber os dados e o SCL por criar um clock que sincroniza os sistemas. Somado a isso, o sensor possui um barramento mestre I2C auxiliar, permitindo acessar magnetômetros externos ou outros sensores e que eles coletem um conjunto completo de dados do sensor sem intervenção do processador do sistema. A pinagem completa do MPU-6050 contemplando essas funcionalidades pode ser observada na Tabela 4.9, que descreve o significado de cada pino. Não havendo a necessidade de interfacear outros sensores I2C com o MPU-6050, os únicos pinos conectados ao ESP32 são os de alimentação (VCC e GND) e transmissão de dados (SCL e SDA). Para a montagem do circuito, antes de mais nada foi-se necessário soldar a barra de pinos na placa do CI. Posteriormente, conectou-se o pino SCL ao GPIO 22 e o pino SDA ao GPIO 21, as entradas do microcontrolador responsáveis pela comunicação I2C.

VCC	Fase (3,3V ou 5V)
GND	Neutro
SCL	Serial Clock pin para comunicação I2C
SDA	Serial Data pin para comunicação I2C
XCA	Usado para fazer a interface de outros sensores I2C com o MPU-6050
XCL	Usado para fazer a interface de outros sensores I2C com o MPU-6050
AD0	Use este pino para alterar o endereço I2C
INT	Pino de interrupção Pode ser usado para indicar que novos dados de medição estão disponíveis

Tabela 4.9 – Pinagem do acelerômetro MPU-6050

A partir de todos os dados apresentados, modelou-se o circuito do ESP32 conectado aos sensores DS18B20, KY-038 e MPU-6050 por meio do software Fritzing. O resultado final com o diagrama esquemático da montagem do circuito pode ser observado na Fig. 4.26.

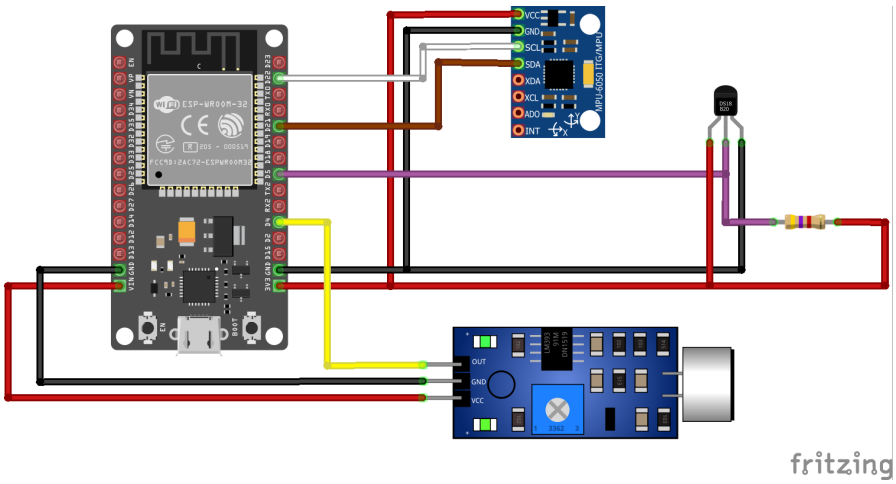


Figura 4.26 – Diagrama esquemático da montagem do circuito

4.1.4 Programação dos dispositivos no Arduino IDE

O código desenvolvido para captar os dados dos sensores e se comunicar com o sistema supervisório é bem intuitivo e fácil de compreender. O código completo pode ser encontrado no link do Anexo A, sendo o foco desta subseção explicar com maiores detalhes alguns trechos. Como qualquer outro programa de Arduino, ele é divido em duas funções principais: *setup* e *loop*. A anatomia do código Arduino define que todas as configurações iniciais sejam executadas somente uma vez no *setup()*, enquanto as operações principais para a execução de uma tarefa sejam executadas em repetição no *loop()*. Os trechos detalhados a seguir apresentam a mesma ordem e estrutura do código original, somente retirando alguns comentários redundantes e reduzindo algumas linhas repetitivas.

As bibliotecas ***modbus-esp8266***, ***Adafruit_MPU6050***, ***Adafruit_Sensor*** e ***DS18B20*** são necessárias para a execução do código. Elas foram instaladas usando o Library Manager no Arduino IDE ou baixando a versão mais recente diretamente do GitHub. A biblioteca ***modbus-esp8266*** permite que a placa Arduino (com suporte para ESP8266/ESP32) se comunique via protocolo Modbus por meio de transporte de rede (Modbus TCP) ou linha serial/RS-485 (Modbus RTU), atuando como mestre, escravo ou ambos. Já as bibliotecas da Adafruit são utilizadas para a leitura do sensor MPU-6050. A ***Adafruit_Sensor*** é uma classe base que define um sistema de driver unificado para garantir que qualquer sensor possa utilizar as mesmas funções pré-definidas e obter de volta os resultados nas unidades e escalas esperadas. A ***Adafruit_MPU6050*** é um biblioteca específica do próprio sensor para a manipulação dos dados recebidos e realizar a comunicação I2C. Por fim, a biblioteca ***DS18B20*** é utilizada, como o próprio nome sugere, para a leitura do sensor de temperatura, oferecendo suporte à descoberta automática de sensores ou endereçamento manual de sensores individuais. Ela usa a biblioteca ***OneWire***, portanto também é necessário instalá-la.

Prosseguindo para o código em si, a priori foram criadas as macros *SLAVE_ID*, *KY038_PIN* e *DS18B20_PIN*, respectivas ao endereço modbus do ESP32, o pino de entrada do KY-038 e o do DS18B20. A ID do microcontrolador é 1 e os sensores estão conectados ao GPIO4 e GPIO5, respectivamente. Um objeto *Adafruit_MPU6050* (definido pela biblioteca) foi criado para lidar com o sensor MPU-6050 e o mesmo ocorreu para a comunicação Modbus. Especificamente para o DS18B20, criou-se uma instância utilizando uma função de sua biblioteca, onde o valor do pino de conexão é usado como argumento.

```
#define SLAVE_ID 1
#define KY038_PIN 4
#define DS18B20_PIN 5

Adafruit_MPU6050 mpu;
ModbusRTU mb;
DS18B20 temp(DS18B20_PIN);
```

4.1.4.1 Função setup()

A função *setup* inicializa os dispositivos e a rede Modbus, declarando alguns parâmetros necessários. Logo no início, o pino onde o sensor de som está conectado foi definido como entrada de dados e a comunicação serial foi configurada com uma taxa de transmissão de 115200 bits/s, sem paridade, 8 bits de dados e 1 stop bit.

```
pinMode(KY038_PIN, INPUT);
Serial.begin(115200, SERIAL_8N1);
```

Depois, inicializou-se o protocolo Modbus declarando o ESP32 como um escravo com endereço 1 e adicionando todos os registradores, responsáveis por armazenar os dados. As funções da biblioteca ***modbus-esp8266*** foram necessárias para realizar estes procedimentos. Os registradores foram adicionados por meio da função *addHreg*, onde o primeiro argumento é o seu offset e o segundo argumento o seu valor inicial. Foram ao total 15 registradores do tipo *holding register* (para armazenar valores analógicos), todos iniciados com valor 0.

```
mb.begin(&Serial);
mb.slave(SLAVE_ID);
mb.addHreg(offset, valor inicial); //Meramente demonstrativo
```

Ao final, inicializou-se o sensor MPU-6050 utilizando as funções da biblioteca ***Adafruit_MPU6050***. Definiu-se a faixa de medição do acelerômetro para 2G (-19,62~19,62m/s^2), a faixa do giroscópio para ±250°/s e a a largura de banda do filtro para 21Hz.

```
mpu.begin();
mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
mpu.setGyroRange(MPU6050_RANGE_250_DEG);
mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
```

4.1.4.2 Função loop()

A função *loop* realiza a leitura dos dados dos 3 sensores e os armazena nos registradores da rede Modbus. Começando pelo sensor de temperatura, utiliza-se a função *getTempC* da biblioteca ***DS18B20*** para ler o seu valor e armazená-lo na variável *temperatura*. Entretanto, seus dados são floats de 32 bits e a rede Modbus trabalha com registradores de 16 bits, sendo necessário fazer uma transformação no valor lido para conseguir armazená-lo. A solução encontrada foi acessar o endereço da variável *temperatura* para obter o seu valor em binário e armazená-lo em outra variável do tipo inteiro sem sinal de 32 bits chamada *valor_temp*. Posteriormente, quebrou-se este inteiro de 32 bits em 2 inteiros sem sinal de 16 bits: *reg_temp1* e *reg_temp2*. De tal forma, é possível alocar 2 registradores para guardar cada um dos inteiros e, na hora de acessá-los, ler como uma variável só do tipo float de 4 bytes.

```
float temperatura = temp.getTempC();
uint32_t valor_temp = *((uint32_t*) &temperatura);
uint16_t reg_temp1 = valor_temp >> 16;
uint16_t reg_temp2 = valor_temp & reg_temp1;
```

Prosseguindo para o sensor de som, os dados do ADC se encontram na faixa de valores entre 0 e 1023. É realizada uma amostragem durante 50ms por meio da função *millis()* dos dados lidos pela função *analogRead* com o pino de conexão do sensor como argumento. Os resultados, então, são armazenados nas variáveis *sinal_min* e *sinal_max*. Agora, para descobrir a relação do sinal pico a pico, subtrai-se o valor mínimo do valor máximo e armazenou-se o resultado na variável *sinal_do_som*. Este valor é utilizado para mapear o sinal do som em decibéis, com uma faixa de valores de 0dB a 90dB. O resultado do ruído lido pelo sensor em decibéis é armazenado na variável *ruído*.

```
unsigned long startMillis = millis();
unsigned int sinal_max = 0;
unsigned int sinal_min = 1024;
float sinal_do_som = 0;

while (millis() - startMillis < 50) {
   valor_anl = analogRead(KY038_PIN);
   if (valor_anl < 1024) {
      if (valor_anl > sinal_max) {
         sinal_max = valor_anl;
      } else if (valor_anl < sinal_min) {
         sinal_min = valor_anl;
      }
   }
}

sinal_do_som = sinal_max - sinal_min;
int ruido = map(sinal_do_som, 0, 1023, 0, 90);
```

A respeito do sensor de vibração e rotação, para ler os seus dados é necessário obter novos eventos de sensor utilizando funções da biblioteca ***Adafruit_Sensor***. O sistema de driver de sensor unificado usa dois objetos básicos para realizar a leitura do sensor, o *sensors_event_t* e o *sensor_t*. Não é necessário utilizar o objeto *sensor_t*. Já o *sensors_event_t* é usado para encapsular uma leitura de sensor específica, chamada de evento, e conter dados de um momento específico no tempo. Portanto, criaram-se os objetos **a**, **g** e **temp** para a leitura respectiva do acelerômetro, giroscópio e da temperatura.

A função *getEvent* lê um novo conjunto de valores do sensor, os converte para as unidades SI e escala apropriadas e atribui os resultados aos objetos específicos criados do *sensor_event_t*. Os valores do acelerômetro foram armazenados nas variáveis ***a.acceleration.x***, ***a.acceleration.y*** e ***a.acceleration.z***, respectivas aos 3 eixos. Os valores do giroscópio foram armazenados nas variáveis ***g.gyro.x***, ***g.gyro.y*** e ***g.gyro.z***, também respectivas aos 3 eixos. E o valor da temperatura na variável ***temp.temperature***.

Igualmente ao dados do sensor de temperatura, os dados do sensor MPU-6050 são floats de 32 bits. O mesmo procedimento de conversão do float em 2 inteiros de 16 bits foi realizado para cada uma das variáveis do sensor. Vale destacar que o acelerômetro mede a aceleração ao longo dos eixos e detecta forças estáticas como gravidade ou forças dinâmicas como vibrações ou movimento. Idealmente, em um objeto estático, a aceleração no eixo Z é igual à força gravitacional e deve ser zero nos eixos X e Y. Portanto, a depender da posição do sensor, os valores de aceleração não serão necessariamente nulos.

```
sensors_event_t a, g, temp;
mpu.getEvent(&a, &g, &temp);

//O mesmo procedimento abaixo foi realizado nas outras variáveis
float acelerometro_x = a.acceleration.x;
uint32_t valor_acel_x = *((uint32_t*) &acelerometro_x);
uint16_t reg_acel_x1 = valor_acel_x >> 16;
uint16_t reg_acel_x2 = valor_acel_x & reg_acel_x1;
```

Por fim, armazenou-se em ordem os valores de todas as variáveis nos registradores da rede Modbus. A função *Hreg* recebe como primeiro argumento o offset do registrador e o segundo argumento a variável que vai ser armazenada nele.

```
mb.task();
mb.Hreg(offset, variável); //Meramente demonstrativo
yield();
```

4.1.5 Instalação completa

Juntando o ESP32, os sensores e o inversor de frequência, tudo devidamente conectado e programado, obtém-se um panorama completo do sistema de aquisição de dados. Uma caixa de automação 40X50cm foi utilizada para organizar os dispositivos, sendo necessário materiais e ferramentas adequados para perfurar, cortar e encaixar os componentes. A ideia é colocar todo o sistema dentro da caixa, com somente entradas e saídas para cabos. Entretanto, não foi possível colocar o ESP32 dada a forma de conexão atual dele, sendo necessário para tal prover alimentação de energia externa à ele e cabos manga para conexão aos sensores.

Somado aos dispositivos, os motores elétricos se encontram instalados na bancada do GRACO e estão conectados aos sistema de aquisição de dados de duas maneiras distintas. A primeira é pelo inversor de frequência individual de cada motor (no trabalho atual há somente 1 porém a ideia é ter 2) localizados dentro da caixa de automação, conectados via os cabos de energização do motor como apresentado na Fig. 4.22b. A segunda é por meio da instalação dos sensores em pontos críticos de temperatura, ruído e vibração dos motores. As Figs. 4.27a e 4.27b apresentam os locais adequados e críticos para verificação da vibração e temperatura, respectivamente, segundo as especificações da WEG.

(a) Fonte: (WEG, 2023a) (b) Fonte: (WEG, 2023b)

Figura 4.27 – Locais para verificação de vibração (a) e temperatura (b) no motor elétrico

Sobre o sensor de som, apesar de nenhum dos documentos citados mencionar um local recomendado de instalação para verificação de ruído, segundo o Guia de Especificação da WEG (WEG, 2023b) os motores elétricos possuem basicamente três fontes de ruído: o sistema de ventilação, os rolamentos e a interação entre ondas eletromagnéticas. Dado que os rolamentos estão em boas condições e sabendo que o número de polos destes motores são mais baixos, havendo menos interferência eletromagnética e rotações mais rápidas, assumiu-se que a principal fonte de ruído proveria do sistema de ventilação. Portanto, instalou-se o sensor de som o mais próximo possível das ventoinhas. O resultado final com o sistema de aquisição de dados completo pode ser observado na Fig. 4.28 abaixo.

Figura 4.28 – Sistema de Aquisição de Dados completo

4.2 Sistema Supervisório

O sistema supervisório foi elaborado no ScadaBR 1.2 (SCADABR, 2023) alinhado às variáveis fornecidas pelo sistema de aquisição de dados, permitindo controlar e monitorar os parâmetros de interesse do CFW-08 e os valores dos sensores DS18B20, MPU-6050 e KY-038. Foram criados *data sources* e *data points* responsáveis por se comunicarem aos dispositivos via Modbus RTU e armazenar as variáveis do sistema, alarmes com dados fora do recomendado para informar o usuário sobre a necessidade de uma devida atenção ao motor e telas gráficas de simples compreensão com informações condensadas em um único ambiente. Será apresentado a seguir uma visão geral e detalhes sobre as propriedades dos componentes do sistema desenvolvido.

4.2.1 Visão Geral

O ScadaBR, como citado na subseção 3.3.2, possui uma interface 100% web, ou seja, ao inicializar o software é necessário escolher um navegador para executá-lo. Um login com um usuário e uma senha vai ser exigido para acessar o sistema, visto que o ScadaBR permite criar múltiplos usuários com diferentes permissões para diferentes propósitos (clientes, operadores, admnistradores). Não foi criado nenhum outro usuário a princípio dado que este não era o foco, bastando utilizar o usuário padrão ***admin*** para entrar no sistema. Ele possui acesso e permissão a todo o projeto desenvolvido e sua senha é igual ao usuário.

Ao entrar será possível observar uma tela parecida com a da Fig. 4.30 contendo o valor de todas as variáveis em tempo real. A página inicial padrão é a de ajuda, porém configurou-se para que o sistema desenvolvido entrasse diretamente na *Watch list*. Uma *Watch list* é uma lista de pontos de dados atualizados automaticamente nos quais o usuário tem interesse em observar simultaneamente. Dessa forma, a necessidade de atualizar a página manualmente (pressionar F5) é dispensada. É possível também criar múltiplas listas, porém só pode observar uma por vez. No canto superior esquerdo da página, existe uma barra de ícones junto ao logotipo do ScadaBR. Cada ícone permite acessar uma funcionalidade diferente do sistema supervisório, sendo que os principais tanto na barra, como na aplicação em geral, são apresentados na Fig. 4.29. Os *data sources* foram ajustados acessando o ícone correspondente da barra, assim como as representações gráficas. Já os alarmes das variáveis podem ser acessadas por meio das configurações individuais de cada *data point*.

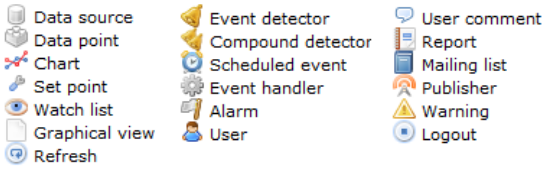


Figura 4.29 – Ícones do ScadaBR
Fonte: https://sites.google.com/a/certi.org.br/certi_scadabr/home/minicursos/iniciando-scadabr em 08/07/2023

Os parâmetros do inversor de frequência e as variáveis básicas escolhidas para fazer parte do sistema supervisório constam nas Figs. 4.30 e 4.31. São ao todo 25 *data points* que serão melhor explicados na subseção 4.2.2. Foram criadas 3 *watch lists* para o projeto: uma chamada "**Status do Motor**" contendo somente as variáveis relacionadas à leitura de dados do motor, outra chamada "**Parametrização**" contendo os parâmetros de configuração do inversor e a última chamada "**Todos**" juntando as outras duas em somente uma *watch list* com todas as variáveis do sistema. É importante realizar esta divisão para separar as variáveis de controle (parametrização) e monitoramento (status do motor), deixando o processo mais organizado e menos suscetível a erros. A Fig. 4.30 apresenta a primeira lista, com dados lidos em tempo real onde o motor foi acionado para rotacionar a uma velocidade de 1800 RPM.

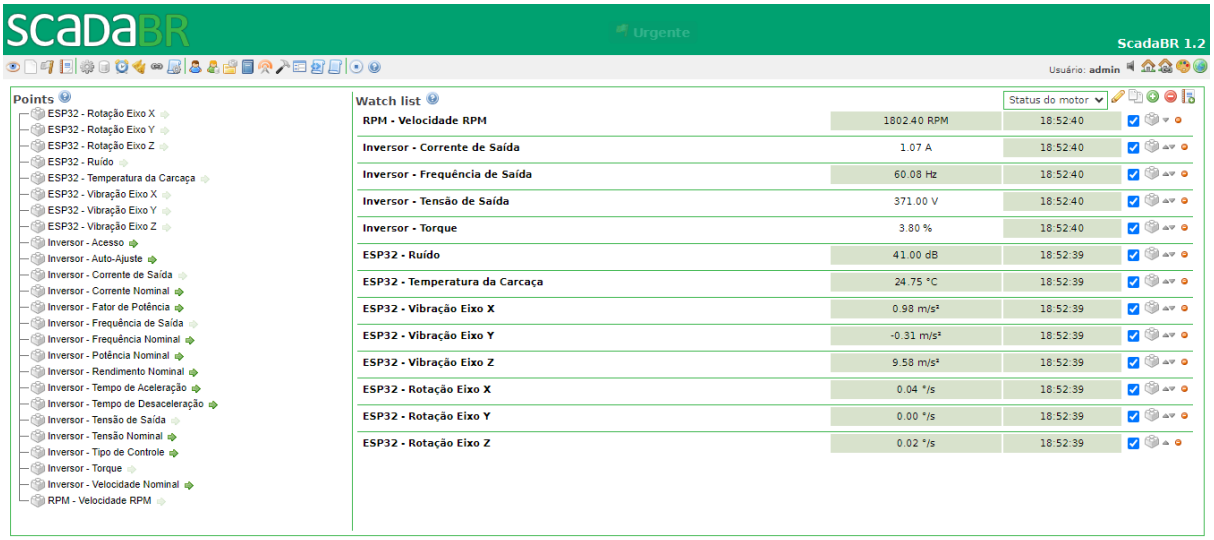


Figura 4.30 – *Watch list* Status do Motor

A Fig. 4.31 apresenta a segunda lista, com os parâmetros definidos nas Tabelas 4.7 e 4.8 somado a alguns outros já mencionados que serão explicados na subseção 4.2.2.2.

Figura 4.31 – *Watch list* Parametrização

4.2.2 Data sources

A lista de *data sources* consiste em um conjunto de informações que apresenta todas as fontes de dados definidas dentro de um sistema, abrangendo seus respectivos nomes, tipos, atributos de conexão e status. Os valores de conexão correspondem a configurações específicas, variando de acordo com o tipo de fonte de dados em questão. O status, por sua vez, indica se uma determinada fonte de dados está habilitada ou desabilitada. No contexto analisado, foram criadas três *data sources* (Fig. 4.32): duas do tipo Modbus Serial, utilizadas para estabelecer comunicação com os dispositivos ESP32 e CFW-08, e uma fonte de dados denominada Meta, a qual tem como finalidade a manipulação de variáveis visando a apresentação do valor da velocidade em rotações por minuto (RPM).

Figura 4.32 – Data sources do projeto

O data source **Modbus Serial** é empregado com o intuito de adquirir dados provenientes de uma rede Modbus local, cujo acesso pode ser efetuado por meio de comunicação RS232 ou RS485. Vale ressaltar que tal fonte de dados opera por meio de consultas periódicas. Já o **Meta** data source recebe essa denominação devido à sua habilidade em combinar pontos de dados existentes para criar novos. Ao invés de obter dados de uma fonte externa, ele utiliza os valores de outros pontos de dados e permite que o usuário os manipule de forma arbitrária.

4.2.2.1 ESP32

Aprofundando nos aspectos de cada um dos data sources, primeiro precisa-se compreender as propriedades e configurações deles. Toda fonte de dados requer um **nome**, podendo tal denominação ser atribuída de forma livre, a critério do usuário. Especificamente em relação ao tipo Modbus Serial, o **período de atualização** determina a frequência com a qual a rede Modbus é consultada para a obtenção de dados. Por outro lado, os campos denominados **Timeout** e **Retries** regulam o comportamento do sistema em situações em que ocorre falha durante a realização de uma consulta. Em tais casos, a fonte de dados permanece em espera por um intervalo de tempo específico, aguardando uma resposta da rede. Caso não seja obtida qualquer resposta, a requisição é repetida por um número determinado de vezes.

A opção denominada "**Apenas quantidades contíguas**" permite especificar que a implementação do protocolo Modbus não deve otimizar diferentes solicitações de valores em uma única requisição. Ao selecionar essa opção, a implementação realizará requisições somente para múltiplos valores quando os mesmos estiverem em uma sequência de registros contínuos. A comunicação serial é controlada através de parâmetros como **taxa de transmissão** (baud rate), **controle de fluxo de entrada**, **controle de fluxo de saída**, número de **bits de dados** (Data bits), número de **bits de parada** (Stop bits) e configuração de **paridade**. Vale destacar que o recurso de **eco** (echo) pode ser empregado em redes RS485, conforme apropriado para a situação.

As propriedades do data source ESP32 podem ser visualizadas na Fig. 4.33. Ele recebeu esse nome para identificar que se trata da comunicação com o microcontrolador, um período de atualização de 5 segundos para manter os dados em tempo real, timeout também de 5 segundos dado que há um delay na comunicação com o ESP32, 2 retentativas e os parâmetros de comunicação serial programados pela IDE do Arduino (4.1.4.1).

Figura 4.33 – Propriedades do data source ESP32

Os *data points*, também conhecidos como *tags* em outros contextos, referem-se às variáveis numéricas ou alfanuméricas utilizadas em uma determinada aplicação. Tanto as redes serial quanto as redes IP utilizam os mesmos atributos para localizar os valores correspondentes. Foram criados 8 *data points* para o data source ESP32 (Fig. 4.34), correspondente a cada um dos dados lidos pelos sensores DS18B20, MPU-6050 e KY-038. É possível perceber que são todos *Holding Registers* com os offsets ajustados seguindo a mesma ordem dos registradores na programação dos dispositivos (4.1.4.2). Com exceção do data point Ruído, todos os outros são do tipo *float* e por isso lêem o valor armazenado no seu offset indicado juntamente com o próximo para determinar o valor da variável.

Data points

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)	
Rotação Eixo X	Numérico		1	Registrador holding	6	
Rotação Eixo Y	Numérico		1	Registrador holding	8	
Rotação Eixo Z	Numérico		1	Registrador holding	10	
Ruído	Numérico		1	Registrador holding	12	
Temperatura da Carcaça	Numérico		1	Registrador holding	13	
Vibração Eixo X	Numérico		1	Registrador holding	0	
Vibração Eixo Y	Numérico		1	Registrador holding	2	
Vibração Eixo Z	Numérico		1	Registrador holding	4	

Figura 4.34 – Data points do ESP32

Aprofundando nas propriedades dos data points, a Fig. 4.35 apresenta os detalhes do data point "Rotação Eixo X" acessíveis ao editá-lo, a fim de exemplificar. O **Id** do escravo é o identificador atribuído a um nó Modbus específico. A **Faixa de registro** é responsável por determinar em qual das quatro faixas os valores devem ser procurados: *Coil status* (bit único gravável), *Input status* (bit único de somente leitura), *Holding register* (2 bytes ou "palavra" gravável) e *Input register* (2 bytes ou "palavra" de somente leitura). O campo **Tipo de dado Modbus** reflete as diferentes formas de codificação dos dados. O campo **Offset** é um valor numérico que segue a indexação baseada em 0. O campo **Configurável** permite definir um ponto que normalmente seria configurável como não configurável. Por fim, os campos **Multiplicador** e **Aditivo** são utilizados para realizar conversões simples de valores, quando necessário. Quando se trata de valores numéricos lidos da rede, esses são calculados seguindo a fórmula: (valor bruto) * multiplicador + aditivo.

Detalhes do data point

Nome	Rotação Eixo X
Export ID (XID)	DP_938554
Id do escravo	1
Faixa do registro	Registrador holding
Tipo de dados modbus	Float de 4 bytes
Offset (baseado em 0)	6
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	
Multiplicador	1
Aditivo	0

Figura 4.35 – Propriedades do data point Rotação Eixo X

4.2.2.2 Inversor

Igualmente ao data source ESP32, os mesmos procedimentos foram realizados para o data source Inversor. As suas propriedades podem ser visualizadas na Fig. 4.36: este nome para identificar que se trata da comunicação com o CFW-08, um período de atualização de 5 segundos para manter os dados em tempo real, timeout de 2.5 segundos dado que a comunicação é mais rápida, 0 retentativas e os parâmetros de comunicação serial definidos de acordo com a Tabela 4.6. Uma característica importante a se destacar é a marcação da opção **Apenas quantidades contíguas** para evitar o erro “Illegal data address”.

Figura 4.36 – Propriedades inversor

Foram criados 16 *data points* para o data source Inversor (Fig. 4.37), correspondente aos parâmetros das Tabelas 4.7 e 4.8, o parâmetro de Acesso P000, os parâmetros de leitura P003, P005, P007 e P009 e o parâmetro de configuração do Tipo de Controle P202, todos seguindo as instruções do manual do CFW-08 (WEG, 2009). Todos os data points são do tipo *Inteiro de 2 bytes sem sinal*.

Data points

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)	
Acesso	Numérico		1	Registrador holding	0	
Auto-Ajuste	Numérico		1	Registrador holding	408	
Corrente de Saída	Numérico		1	Registrador holding	3	
Corrente Nominal	Numérico		1	Registrador holding	401	
Fator de Potência	Numérico		1	Registrador holding	407	
Frequência de Saída	Numérico		1	Registrador holding	5	
Frequência Nominal	Numérico		1	Registrador holding	403	
Potência Nominal	Numérico		1	Registrador holding	404	
Rendimento Nominal	Numérico		1	Registrador holding	399	
Tempo de Aceleração	Numérico		1	Registrador holding	100	
Tempo de Desaceleração	Numérico		1	Registrador holding	101	
Tensão de Saída	Numérico		1	Registrador holding	7	
Tensão Nominal	Numérico		1	Registrador holding	400	
Tipo de Controle	Numérico		1	Registrador holding	202	
Torque	Numérico		1	Registrador holding	9	
Velocidade Nominal	Numérico		1	Registrador holding	402	

Figura 4.37 – Data points do Inversor

4.2.2.3 RPM

Por fim, data source RPM não possui nenhuma configuração de propriedade para ser ajustada. Criou-se o data point "**Velocidade RPM**" correspondente ao parâmetro P002 do inversor de frequência. Devido à limitação do número de bits na comunicação Modbus, o maior número inteiro que pode ser lido em uma variável é 65536, igual a 2^{16}. Como os números inteiros com os decimais no CFW-08 são considerados como um só na comunicação Modbus (por exemplo uma corrente de 1,02A seria armazenada no registrador como 102), o maior valor real que pode ser lido neste parâmetro é 655,36 RPM. Entretanto, são registrados valores muito superiores a este na utilização do motor. Portanto, dado que o próprio manual (WEG, 2009) indica que este parâmetro é um valor proporcional à frequência (P208 X P005), realizou-se esta multiplicação diretamente no ScadaBR como pode ser visto na Fig. 4.38.

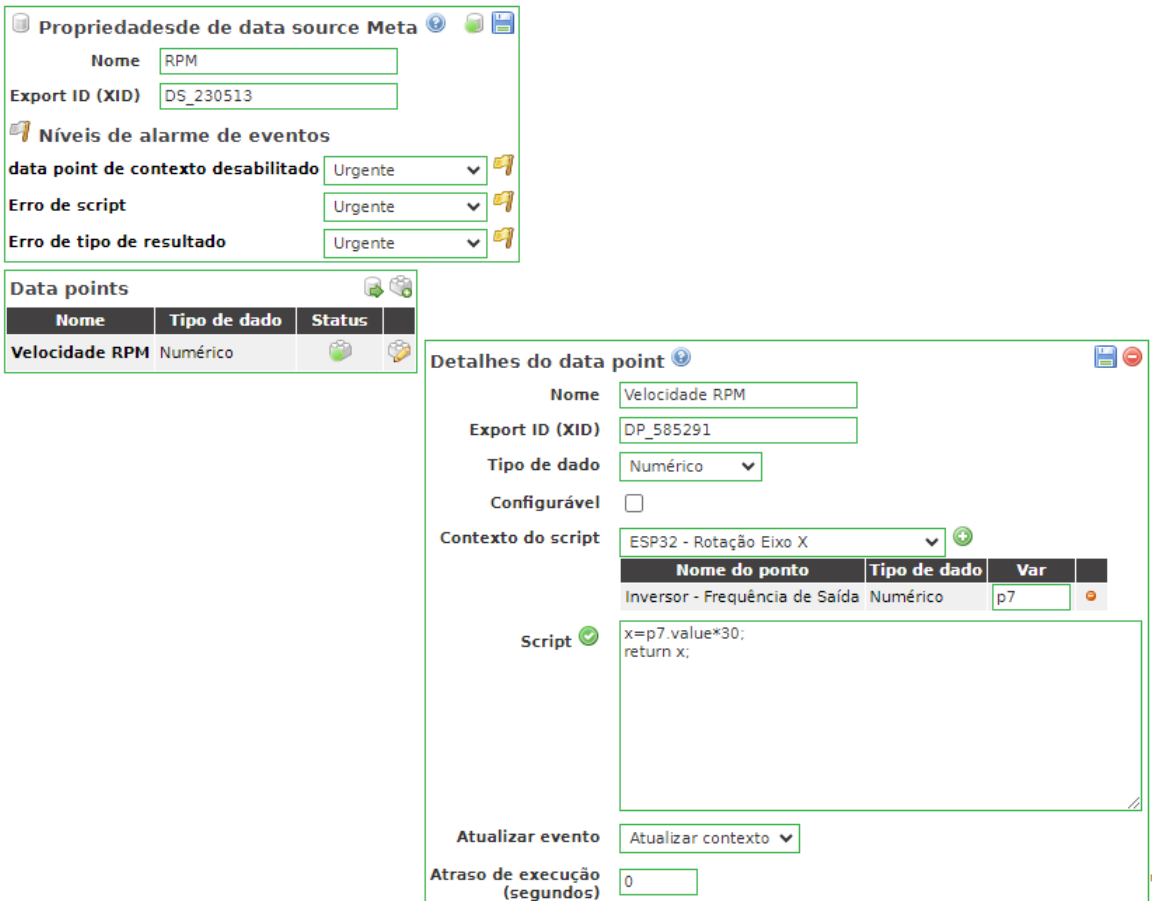


Figura 4.38 – Data Source RPM

O parâmetro P208 só é utilizado quando o motor está no modo de controle vetorial, não havendo a necessidade de realizar a medição da velocidade em RPM em outro modo. A conversão de Hz para rpm é feita em função do número de polos do motor utilizado. Um motor de 4 polos, como o caso deste projeto, indica que o valor a multiplicar pela corrente é 30. Na área de script da Fig. 4.38 é possível observar a fórmula para definir o valor da velocidade. Qualquer script realizado nesse data source é realizado na linguagem JavaScript.

4.2.3 Alarmes

No ScadaBR, existem dois tipos de alarmes: os definidos pelo sistema em casos de erros operacionais, como falha de comunicação e acesso ilegal de endereço, e os definidos pelo usuário nos detectores de eventos. O propósito dos detectores é verificar se o valor de um data point atende a uma ou mais condições específicas e, caso isso ocorra, acionar um evento que pode ser tratado posteriormente. Eles podem identificar diferentes tipos de condições, porém são aplicáveis apenas a um único ponto de dados. Caso seja necessário estabelecer condições para vários pontos ou condições agendadas, é possível utilizar detectores de eventos compostos. Cada data point pode ter um número ilimitado de detectores de eventos, sendo que cada detector possui um nível de alarme associado. Esse nível determina se o detector ativará um alarme quando acionado, bem como a sua gravidade.

Os detectores de eventos criados para o projeto podem ser acessados por meio da *Watch list*, ao clicar no ícone "**Detalhes do data point**" correspondente do data point de interesse, e em seguida na opção "**Editar data point**", direcionando para a página de edição da variável. Foram criados alarmes para os valores limites de vibração, ruído e temperatura nos respectivos data points, conforme especificado na subseção 3.2.2. Os limites superiores e inferiores de vibração foram definidos como 2,5 m/s^2 e -2,5 m/s^2, respectivamente. A Fig. 4.39 apresenta os detectores de eventos criados para o data point Vibração X, sendo que o mesmo foi realizado para o Vibração Y e Vibração Z. Vale destacar que como o eixo Z possui influência da gravidade, os limites superior e inferior são 12,3 m/s^2 e 7,3 m/s^2.

Figura 4.39 – Detectores de eventos do data point Vibração X

O limite superior de ruído foi definido como 78dB, e não há limite inferior. A Fig. 4.40 apresenta os detector de eventos criados para o data point Ruído.

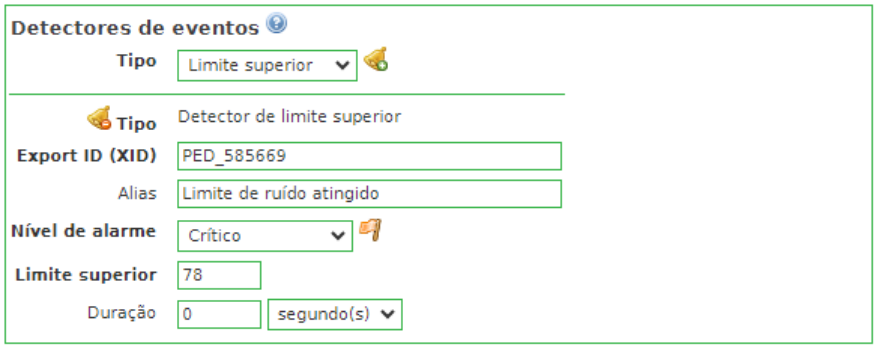


Figura 4.40 – Detector de eventos do data point Ruído

O limite superior de temperatura foi definido como 120°C, e não há limite inferior. A Fig. 4.41 apresenta o detector de eventos criados para o data point Temperatura da Carcaça.

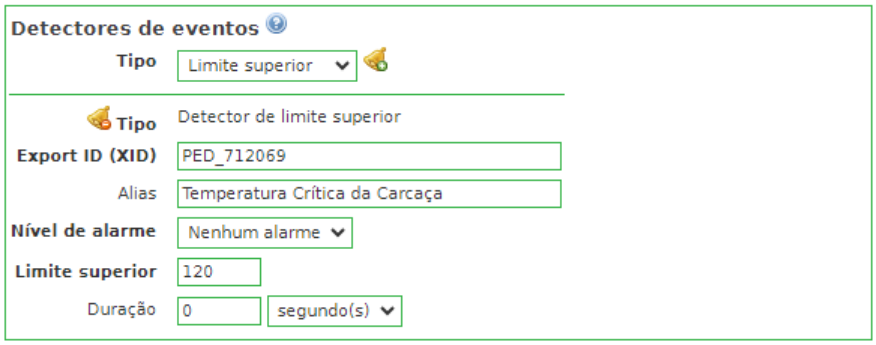


Figura 4.41 – Detector de eventos do data point Temperatura da Carcaça

4.2.4 Representações Gráficas

A representação gráfica é uma visualização gráfica dos data points selecionados, atualizados automaticamente com valores e timestamps em tempo real. Os usuários podem ter várias representações gráficas, mas podem visualizar apenas uma por vez. O ScadaBR oferece uma variedade de componentes gráficos para visualizações e supervisão de sistemas, que incluem gráficos em tempo real, tabelas, indicadores, imagens sobrepostas com dados dinâmicos, botões e controles interativos, animações e histórico com tendências. Ele também permite a criação de interfaces personalizadas de acordo com as necessidades específicas de cada sistema, proporcionando uma experiência de supervisão completa e eficiente.

A página de edição da representação gráfica permite a edição de todos os atributos da representação e seus pontos associados. As visualizações consistem em componentes de visualização, que podem ser conteúdo estático ou dinâmico, dependendo dos pontos envolvidos. Para adicionar conteúdo estático, pode-se selecionar um componente do tipo HTML, que permite a inclusão de trechos HTML válidos, incluindo JavaScript, e referenciar classes de estilo de arquivos CSS. Outros componentes na lista dependem de um ou mais pontos para processamento, sendo o **Script para o servidor** o mais utilizado no trabalho, permitindo definir lógica de processamento específica usando JavaScript.

As telas do projeto foram planejadas para agrupar informações contextualizadas de forma clara e concisa. Um menu de navegação foi criado em todas as telas do projeto para facilitar a transição entre elas. Os layouts das telas do supervisório foram desenhados usando símbolos e cores de maneira agradável e intuitiva, e os botões foram nomeados de forma a simplificar o entendimento. O projeto conta com um total de cinco telas: a página inicial, a tela do motor, a tela de parametrização, a tela de alarmes e a tela de gráficos.

4.2.4.1 Página Inicial

A página inicial é mostrada em detalhes pela Fig. 4.42. O seu objetivo é contextualizar o usuário sobre o que se trata o sistema supervisório e os conteúdos das demais telas do sistema. É uma página bem simples somente para introduzir progressivamente o operador ao projeto. Ao lado esquerdo, é possível observar o menu de navegação com as cinco telas desenvolvidas e a identificação da tela atual. À direita está o conteúdo da tela em si, com as instruções e explicações. No canto direito inferior da tela é possível ver o logo da UnB, responsável por incentivar e patrocinar este projeto.

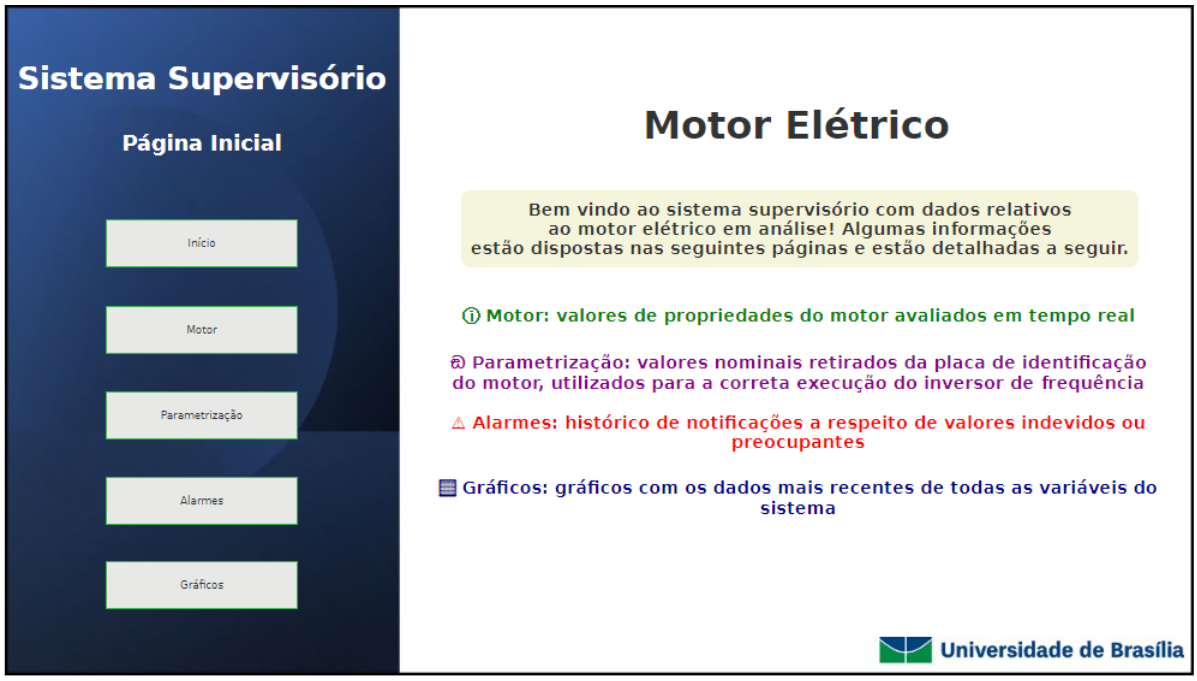


Figura 4.42 – Tela da Página Inicial

4.2.4.2 Motor

A tela do motor é mostrada em detalhes pela Fig. 4.43. O seu objetivo é apresentar as mesmas informações visualizadas na Watch list **Status do motor** (Fig. 4.30), porém de uma forma gráfica, objetiva e clara dos data points responsáveis pela propriedades do motor avaliadas em tempo real. A tela segue o mesmo molde da página inicial (menu de navegação à esquerda e conteúdo da tela à direita) e apresenta as informações centralizadas à imagem de um motor elétrico disposta como exemplo, cuja fonte é o seguinte site.

Figura 4.43 – Tela do Motor

4.2.4.3 Parametrização

A tela de parametrização é mostrada em detalhes pela Fig. 4.44. O seu objetivo é igualmente apresentar as informações de uma forma gráfica, objetiva e clara, contudo dessa vez visualizadas na Watch list **Parametrização** (Fig. 4.31), dos data points responsáveis pela parametrização do inversor de frequência. Ela também segue os mesmos moldes e introduz uma imagem do CFW-08, obtida deste site, para deixar claro que se trata de informações do inversor. Os valores destes parâmetros são modificáveis (o motor deve estar desligado).

Figura 4.44 – Tela de Parametrização

4.2.4.4 Alarmes

A tela de alarmes é mostrada em detalhes pela Fig. 4.45 e segue os mesmos moldes das outras telas. O seu objetivo é apresentar os alarmes mais recentes do sistema, ao total 12 cabíveis na tela, que podem ser reconhecidos ou silenciados.

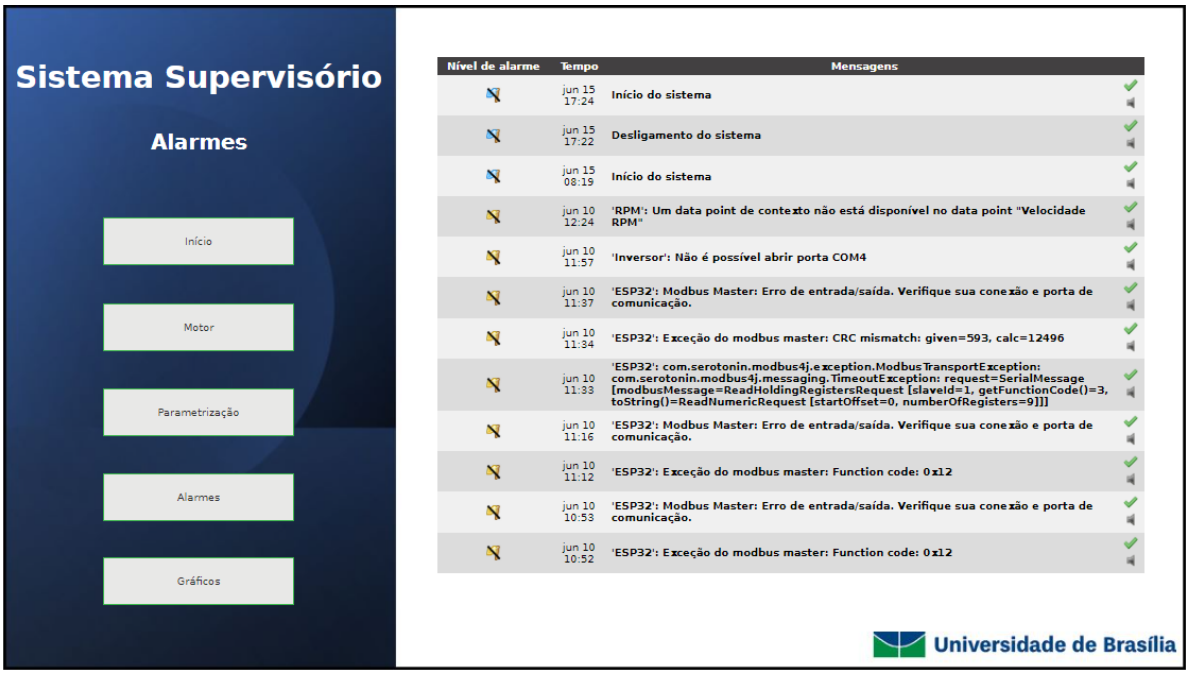


Figura 4.45 – Tela de Alarmes

4.2.4.5 Gráficos

Por fim, a tela de gráficos é mostrada em detalhes pela Fig. 4.46 e também segue os mesmos moldes. O seu objetivo é apresentar os gráficos de tendência do status do motor, oferecendo ao operador a capacidade de monitorar a evolução das variáveis ao longo dos últimos instantes. Dividiu-se os dados em 2 gráficos relacionados às suas naturezas elétrica ou mecânica. O primeiro gráfico mostra os valores mais recentes de saída do CFW-08 para controle de velocidade e torque do motor, apresentando outros dados elétricos relacionados. Já o segundo gráfico mostra os valores mais recentes lidos pelo ESP32 de temperatura, ruído, vibração e rotação do motor. O período de tempo de ambos os gráficos é de 5 minutos, porém isto pode ser alterado nas configurações para apresentar um período maior, ou até todo o histórico de dados em uma futura auditoria. As unidades de cada dado nos gráficos também não são necessariamente compatíveis, como é possível perceber pela Fig. 4.46 ao se analisar por exemplo o primeiro gráfico com valores destoantes de velocidade de giro na casa dos 1800 RPM em comparação com a corrente de saída em torno de 1A. Entretanto, isto não é um problema visto que consegue-se ocultar a informação de qualquer dado ao clicar no botão com cor do lado do seu nome, permitindo analisar cada variável individualmente (ocultando todas as outras) ou comparar 2 ou mais de interesse.

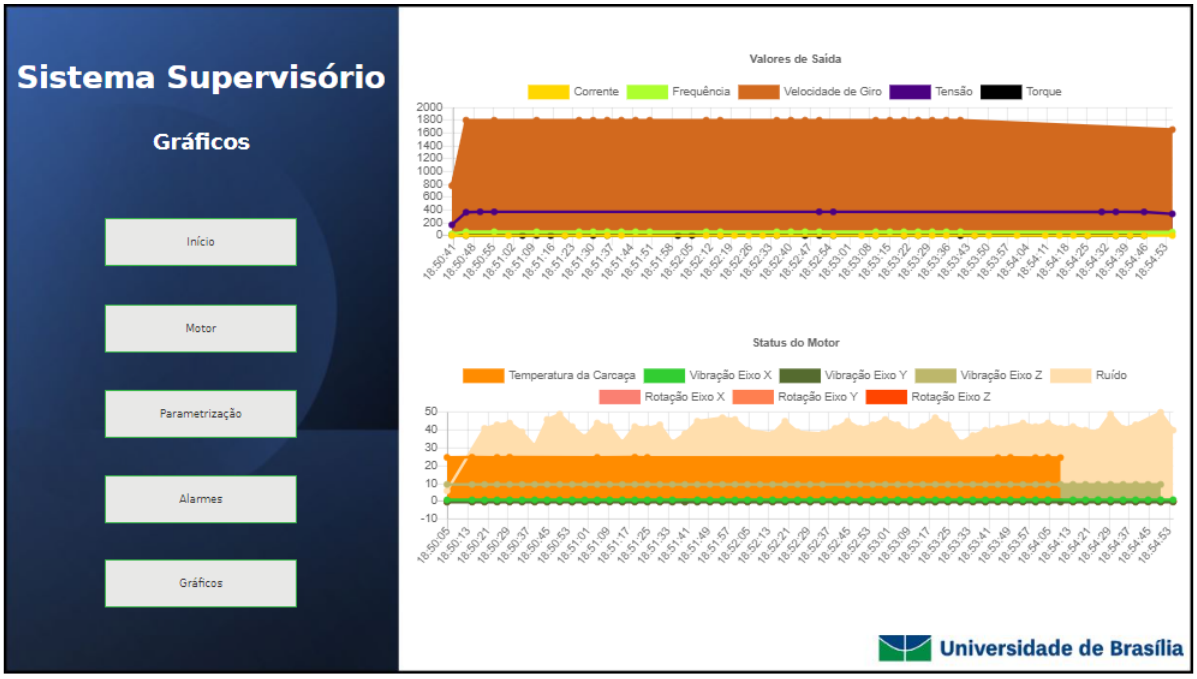


Figura 4.46 – Tela de Gráficos

4.3 Sistema de Inteligência

O trabalho de pesquisa de mestrado conduzido por Douglas Paula de Andrade tem como objetivo a incorporação de inteligência à chamada "*borda*" da Indústria 4.0, utilizando motores elétricos como estudo de caso. A proposta visa viabilizar a leitura e o armazenamento de dados industriais com ênfase na visualização em tempo real e no histórico de dados o mais próximo possível da fonte de dados/do limite da rede. Ao se posicionar parte da inteligência na borda, as redes industriais podem ser aliviadas, proporcionando um aumento na eficiência, segurança e robustez, além de facilitar a prevenção de falhas por causa da distribuição do monitoramento. A implantação desse sistema tem como meta aprimorar a eficiência do funcionamento do motor e prolongar sua vida útil, reduzindo os custos de manutenção e as paradas não programadas.

O trabalho aborda quatro aspectos distintos: o registro de tags por meio de uma interface web conectada a um **banco de dados *PostgreSQL***, em que as tags não estão vinculadas ao código, possibilitando a expansão fácil do equipamento sem a necessidade de interrupções ou modificações no código; um controlador/dispositivo que serve como **fonte de dados** do motor elétrico; um **código Python** estruturado como um gateway orientado a objetos, simplificando a adição de novos protocolos de comunicação sem afetar os protocolos existentes, onde as informações do banco de dados são lidas e a comunicação é feita com o controlador e endereços das tags cadastradas; e **painéis de visualização** em tempo real para controle de processos do motor e de relatório gerencial (ANDRADE, 2023). O fluxograma geral do sistema de inteligência pode ser observado na Fig. 4.47.

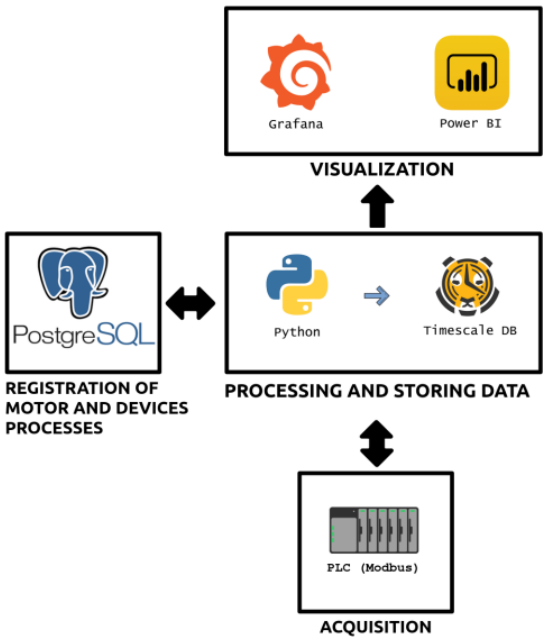


Figura 4.47 – Fluxograma geral da metodologia do sistema de inteligência
Fonte: (ANDRADE, 2023)

De maneira concisa, o sistema opera da seguinte maneira: o código é responsável por ler as tags de interesse provenientes da fonte de dados, seguindo as instruções armazenadas no banco de dados relacional, e em seguida, os valores lidos são salvos em um banco de dados de séries temporais. Esses dados são posteriormente transmitidos e apresentados nos painéis de visualização. Detalhadamente, a fonte de dados do sistema de inteligência é o sistema de aquisição de dados aqui desenvolvido. O ESP32 e o inversor CFW-08 fornecem as informações necessárias sobre o motor elétrico para o código em Python realizar a sua análise e apresentar as suas conclusões sobre o estado do motor.

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional utilizado para cadastrar as tags dos motores e estabelecer a estrutura relacional no sistema de aquisição de dados. A biblioteca psycopg, utilizada no código, possibilita a leitura e escrita de dados no banco de dados PostgreSQL, incluindo o TimescaleDB. O TimescaleDB é uma extensão do PostgreSQL projetada para escalabilidade de dados de séries temporais, oferecendo particionamento automático por data e chave. No projeto, é necessário configurar o servidor TimescaleDB local e criar um banco de dados, além de definir usuário e senha para a interação com o Grafana e Power BI.

O Grafana é uma ferramenta de visualização e análise de código aberto que transforma dados de banco de dados de séries temporais em gráficos e visualizações. É necessário configurar o Grafana adicionando o banco de dados TimescaleDB, fornecendo informações como a URL, porta, nome do banco de dados e autenticação do usuário. O Power BI é usado para criar dashboards de gestão, oferecendo insights sobre os processos de um setor. No projeto, o Power BI é configurado para acessar o banco de dados TimescaleDB e permite a organização e visualização de dados em relatórios.

Somado à isso, o código em Python utiliza as bibliotecas Flask e Pymodbus. O Flask é um framework leve para desenvolvimento de aplicações web e foi utilizado para registrar informações no PostgreSQL e visualizar dados no TimescaleDB. A biblioteca Pymodbus permite a comunicação via Modbus e, nesse caso, é utilizada para leitura de dados do sistema de aquisição de dados. O código em questão realiza o monitoramento das condições e operação dos motores com base em dados históricos, incluindo a análise de falhas passadas e sua importância. Os dados são obtidos por meio da telemetria dos dispositivos, coletados por sensores, e podem ser utilizados para criar um modelo digital do equipamento. Esses dados são agregados e compilados para gerar informações relevantes.

O monitoramento regular da telemetria dos equipamentos possibilita avaliar o potencial de degradação dos motores e calcular Indicadores-Chave de Desempenho (KPIs) relacionados à manutenção. Isso é feito ao combinar dados históricos relacionados a fatores de risco, falhas e cenários operacionais. Essa abordagem proporciona inteligência contínua e permite estimar o momento adequado para a próxima manutenção, que pode ser agendada pelo sistema de manutenção. Ela também confere ao motor uma função de agente inteligente dentro do sistema de manufatura, compartilhando ativamente dados e buscando informações relevantes. Os dados coletados podem ser armazenados localmente ou remotamente na nuvem, dependendo de sua criticidade.

5 Conclusões

A proposta do presente trabalho foi desenvolver um sistema de manutenção preditiva, otimização e monitoração de motores elétricos por meio do conceito de Gêmeos Digitais. Baseado na composição das 4 vertentes tecnológicas deste conceito, foi possível implementar 3 delas na construção do modelo digital dos motores: a Internet das Coisas (IoT), Inteligência Artifical (AI) e Nuvem. O projeto é constituído por um Sistema de Aquisição de Dados, um Sistema Supervisório e um Sistema de Inteligência desenvolvido em um trabalho de mestrado por Douglas, responsáveis pela obtenção, monitoração e análise das informações referentes aos motores presentes nas dependências do GRACO da Universidade de Brasília. Os resultados obtidos evidenciam uma aplicação bem-sucedida dos objetivos iniciais, com grande potencial de crescimento e de melhorias.

Em um primeiro momento, foram definidos os materiais e métodos de pesquisa utilizados com foco principal na confecção do sistema de aquisição de dados, visto que se trata de uma parte mais física e necessitava-se colocar os motores elétricos em funcionamento para desenvolver-se todo o resto. O inversor de frequência WEG CFW-08 foi utilizado para realizar o controle preciso e adequado dos motores e obter dados elétricos em tempo real via protocolo Modbus com um conversor USB/Serial RS-485. O microcontrolador ESP32 foi o controlador escolhido para o desenvolvimento do sistema de aquisição de dados juntamente com os atuadores DS18B20, KY-038 e MPU-6050, responsáveis pela obtenção da temperatura externa da carcaça, do ruído emitido pelo motor e da vibração nos 3 eixos, respectivamente. A programação destes dispositivos foi realizada na IDE do Arduino e o ScadaBR, um software livre, gratuito e de código-fonte aberto, foi escolhido para o sistema supervisório.

Em um segundo momento, focou-se em desenvolver os sistemas para a monitoração dos motores. Sobre o sistema de aquisição de dados, o inversor de frequência foi corretamente alimentado, parametrizado e conectado a eles para realizar as suas partidas e o seu controle, e para funcionar na rede Modbus RTU. Já o ESP32 e os sensores DS18B20, KY-038 e MPU-6050 foram corretamente integrados, programados e conectados para funcionarem e se reconhecerem, por meio da análise das especificações dos pinos, fiação entre os componentes e instalação na posição adequada dos motores. O código completo da programação na IDE do Arduíno pode ser encontrado no link do Anexo A. O resultado final com o sistema completo pode ser observado na Fig. 4.28. Sobre o sistema supervisório, foram criados data sources para se comunicarem via Modbus RTU, seus respectivos data points para armazenarem variáveis, alarmes e telas gráficas de simples compreensão que realizam todo o interfaceamento com o usuário. O projeto conta com um total de cinco telas, apresentadas nas Figs. 4.42, 4.43, 4.44, 4.45 e 4.46. Um menu de navegação foi criado para facilitar a transição entre elas e os layouts foram planejados para serem agradáveis e intuitivos.

Por último, procurou-se incorporar inteligência às informações monitoradas integrando o sistema de aquisição de dados ao trabalho de mestrado conduzido por Douglas. O então chamado Sistema de Inteligência, apesar de ser um trabalho em andamento, possui um modelo funcional, testado e aprovado em um congresso (ANDRADE, 2023) com toda a sua estrutura básica construída. A comunicação via Modbus entre um código em Python, um banco de dados relacional e o sistema de aquisição de dados permite a manipulação e identificação das variáveis. O código lê as tags de interesse dos controladores seguindo as instruções do banco de dados e salva os valores lidos em um novo banco de dados de séries temporais, os dispondo em painéis de visualização em tempo real e de relatório gerencial.

Analisando todas as informações apresentadas, torna-se evidente como os sistemas desenvolvidos alcançam seus objetivos e contribuem significativamente para a concretização do conceito de Gêmeo Digital. Nesse sentido, o presente trabalho desempenha um papel crucial, servindo como um passo inicial e instrucional promissor para a utilização na indústria dessa tecnologia ainda pouco explorada. Contudo, é importante destacar que há uma demanda significativa de trabalho a ser realizado e muitas possibilidades de aprimoramento nos sistemas já construídos. As perspectivas futuras abrangem a implementação de conexão remota entre os sistemas e a exploração da interação tridimensional com o conteúdo digital, aspectos que ampliarão ainda mais a eficiência e a utilidade dessas soluções.

Existem algumas abordagens interessantes a se considerar para aprimorar o sistema atual. Sugere-se adquirir mais materiais para obter dados e monitorar ambos os motores simultaneamente. Além disso, propõe-se conectar os inversores de frequência aos microcontroladores por meio dos pinos RX/TX do ESP32, unificando ambos os dispositivos em uma única rede Modbus para facilitar a comunicação externa com os softwares. Para possibilitar a conexão remota entre o sistema de aquisição de dados e os sistemas supervisório ou de inteligência, sugere-se utilizar o protocolo Modbus TCP/IP ou a comunicação via LoRa, com o auxílio do *ESP32 LoRa LilyGo TTGO*, realizando a comunicação entre ESP32s posicionados estrategicamente nos motores e um ESP32 conectado ao computador como gateway.

Em relação à análise dos dados, o trabalho de mestrado ainda está em andamento para deixar o sistema de inteligência completo. É essencial realizar diversos testes em cenários inadequados para os motores, a fim de verificar a capacidade do sistema em identificar erros e alimentar o banco de dados para futuras referências. Somado a isso, a utilização de softwares de modelagem 3D é recomendada para possibilitar a interação tridimensional com o conteúdo digital, abordando a quarta vertente tecnológica do conceito. Após a construção do sistema de realidade estendida, é necessário integrá-lo ao trabalho já feito para completar o modelo funcional do Gêmeo Digital dos motores elétricos. Dessa forma, o caminho a ser percorrido revela-se repleto de oportunidades para pesquisa e desenvolvimento, e espera-se que os avanços nessa área proporcionem maior sofisticação e funcionalidade aos sistemas em questão, estabelecendo novos padrões de excelência no contexto do Gêmeo Digital.

Referências

ABNT. **IEC 60.034-14, Máquinas elétricas girantes - Parte 14: Medição, avaliação e limites de vibração mecânica de máquinas de altura de eixo igual ou superior a 56 mm**. 2011. Citado na p. 41.

ABNT. **IEC 60085, Isolação elétrica - Avaliação e designação térmicas**. 2017. Citado na p. 39.

ABNT. **NBR 11390, Máquinas elétricas girantes - Medição - Avaliação e limites da severidade de vibração mecânica de máquinas de altura de eixo igual ou superior a 56mm**. 1990. Citado na p. 41.

ABNT. **NBR 17094-1, Máquinas elétricas girantes Parte 1: Motores de indução trifásicos — Requisitos**. 2018. Citado nas pp. 38–40.

ABNT. **NBR 7565, Máquinas elétricas girantes - Limites de ruído**. 1982. Citado na p. 40.

ANDRADE, D. P. de. Telemetry System in Electric Motors using Industry 4.0 Technologies. **12th Brazilian Congress of Manufacturing Engineering**, 2023. Citado nas pp. 70, 71, 74.

ARDUINO. **Tutorial do site oficial do software Arduino IDE**. 2023. Acesso em: 20/05/2023. Disponível em: <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics#libraries>. Citado na p. 43.

ATTARAN, M.; CELIK, B. G. Digital Twin: Benefits, use cases, challenges, and opportunities. **Decision Analytics Journal**, v. 6, p. 100165, 2023. ISSN 2772-6622. DOI: https://doi.org/10.1016/j.dajour.2023.100165. Disponível em: <https://www.sciencedirect.com/science/article/pii/S277266222300005X>. Citado nas pp. 22, 23.

BALDISSARELLI, L.; FABRO, E. Manutenção Preditiva na indústria 4.0. eng. **Scientia cum industria**, Universidade de Caxias do Sul, v. 7, n. 2, p. 12–22, 2019. ISSN 2318-5279. Citado na p. 15.

CORREA, J. S.; RODRÍGUEZ CÓRDOBA, M. d. P.; LEYTON, J. D. Efectos laborales vinculados al uso de técnicas de inteligencia artificial. spa. **Revista Universidad y Empresa**, v. 16, n. 26, p. 211–249, 2014. ISSN 0124-4639. Citado na p. 15.

FERREIRA, B. A. P.; SERUFFO, M. C. D. R.; PIRES, Y. P. Planejamento e construção de um protótipo de aplicativo mobile para visualização de dados de sistema de monitoramento de máquinas e equipamentos. eng. **Revista Principia**, Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, v. 59, n. 3, p. 947–966, 2022. ISSN 1517-0306. Citado na p. 16.

FRANCHI, C. **Inversores de Frequência: Teoria e Aplicações**. Saraiva Educação S.A. ISBN 9788536511658. Disponível em: <https://books.google.com.br/books?id=QYqwDwAAQBAJ>. Citado nas pp. 19, 21.

GARCIA, E. **Introdução a Sistemas de Supervisão, Controle e Aquisição de Dados: SCADA**. Alta Books, 2019. ISBN 9788550809175. Disponível em: <https://books.google.com.br/books?id=vgOcDwAAQBAJ>. Citado nas pp. 27, 28, 30.

HALEEM, A.; JAVAID, M.; PRATAP SINGH, R.; SUMAN, R. Exploring the revolution in healthcare systems through the applications of digital twin technology. **Biomedical Technology**, v. 4, p. 28–38, 2023. ISSN 2949-723X. DOI: https://doi.org/10.1016/j.bmt.2023.02.001. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2949723X23000065>. Citado na p. 24.

JONES, D.; SNIDER, C.; NASSEHI, A.; YON, J.; HICKS, B. Characterising the Digital Twin: A systematic literature review. **CIRP Journal of Manufacturing Science and Technology**, v. 29, p. 36–52, 2020. ISSN 1755-5817. DOI: https://doi.org/10.1016/j.cirpj.2020.02.002. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1755581720300110>. Citado na p. 25.

KUŠIĆ, K.; SCHUMANN, R.; IVANJKO, E. A digital twin in transportation: Real-time synergy of traffic data streams and simulation for virtualizing motorway dynamics. **Advanced Engineering Informatics**, v. 55, p. 101858, 2023. ISSN 1474-0346. DOI: https://doi.org/10.1016/j.aei.2022.101858. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1474034622003160>. Citado na p. 24.

MARTINS, F. J.; FABRO, E. Uso do sensor inteligente na manutenção preditiva do motor de uma extrusora. eng. **Scientia cum industria**, v. 8, n. 2, p. 1–9, 2020. ISSN 2318-5279. Citado na p. 16.

MARTINS, M. S.; PAULA, G. M. d.; BOTELHO, M. d. R. A. Inovações tecnológicas e indústria 4.0 na siderurgia: difusão, estrutura de mercado e heterogeneidade intrassetorial. eng. **Revista brasileira de inovação**, v. 20, e021006, 2021. ISSN 1677-2504. Citado na p. 14.

MARUTHI, G.; PANDURANGA VITTAL, K. Electrical Fault Detection in Three Phase Squirrel Cage Induction Motor by Vibration Analysis using MEMS Accelerometer. In: 2005 International Conference on Power Electronics and Drives Systems. 2005. v. 2, p. 838–843. DOI: 10.1109/PEDS.2005.1619804. Citado na p. 18.

MODBUS. **The Modbus Organization**. 2014. Acesso em: 16/04/2023. Disponível em: <https://modbus.org/>. Citado nas pp. 29, 80.

MORAES, C. C.; CASTRUCCI, P. L. **Engenharia de Automação Industrial 2º ed.** Rio de Janeiro: LTC, 2007. Citado nas pp. 27, 29, 80.

NÓBREGA SOBRINHO, C. A.; SENA, A. P. C. d.; LIMA FILHO, A. C. Estratégias para Detecção de Falhas em Rolamentos de Motores de Indução Trifásicos. eng. **Revista Principia**, Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, v. 1, n. 55, p. 108–118, 2021. ISSN 1517-0306. Citado na p. 16.

OLIVEIRA, L. E. S. de; ÁLVARES, A. J. DESENVOLVIMENTO DE UM SISTEMA PARA MONITORAMENTO E TELEOPERAÇÃO DE MÁQUINAS-FERRAMENTA CNC VIA INTERNET ADERENTE À INDÚSTRIA 4.0. por. **Revista Produção e Desenvolvimento**, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, v. 4, n. 1, p. 133–151, 2018. ISSN 2446-9580. Citado na p. 14.

PANAROTTO, M.; ISAKSSON, O.; VIAL, V. Cost-efficient digital twins for design space exploration: A modular platform approach. **Computers in Industry**, v. 145, p. 103813, 2023. ISSN 0166-3615. DOI: https://doi.org/10.1016/j.compind.2022.103813. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0166361522002093>. Citado na p. 23.

RAMALHO, G. L. B.; PEREIRA, A. H.; REBOUÇAS FILHO, P. P.; MEDEIROS, C. M. d. S. DETEÇÃO DE FALHAS EM MOTORES ELÉTRICOS ATRAVÉS DA CLASSIFICAÇÃO DE PADRÕES DE VIBRAÇÃO UTILIZANDO UMA REDE NEURAL ELM. eng. **Holos (Natal, RN)**, Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, v. 4, p. 185–194, 2014. ISSN 1807-1600. Citado na p. 17.

SCADABR. **Site oficial do software ScadaBR**. 2023. Acesso em: 18/05/2023. Disponível em: <http://www.scadabr.com.br/>. Citado nas pp. 44, 58.

SEMERARO, C.; OLABI, A.; ALJAGHOUB, H.; ALAMI, A. H.; AL RADI, M.; DASSISTI, M.; ABDELKAREEM, M. A. Digital twin application in energy storage: Trends and challenges. **Journal of Energy Storage**, v. 58, p. 106347, 2023. ISSN 2352-152X. DOI: https://doi.org/10.1016/j.est.2022.106347. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352152X22023362>. Citado na p. 24.

SOARES, G. G. P.; TORRES, K. B.; MELO, A. A. d.; MEIRA, M. C.; BATISTA, T. C.; MACIEL, A. D. M. Sistema de proteção microcontrolado para motores elétricos de indução trifásicos. eng. **Revista Principia**, Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, v. 1, n. 50, p. 34–46, 2020. ISSN 1517-0306. Citado na p. 16.

SOMMER, M.; STJEPANDIĆ, J.; STOBRAWA, S.; SODEN, M. von. Automated Generation of Digital Twin for a Built Environment Using Scan and Object Detection as Input for Production Planning. **Journal of Industrial Information Integration**, p. 100462, 2023. ISSN 2452-414X. DOI: https://doi.org/10.1016/j.jii.2023.100

462. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2452414X23000353>. Citado na p. 23.

VIEGAS, V.; PEREIRA, J. M. D. Foundation Fieldbus: From Theory To Practice. **International Journal of Computing**, v. 12, n. 2, p. 115–124, jun. 2013. ISSN 1727-6209. Citado na p. 29.

WEG. **Catálogo Motores Elétricos**. 2023a. Acesso em: 18/05/2023. Disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/catalogo_weg_motores-eletricos_4-44.pdf>. Citado na p. 57.

WEG. **Guia de Especificação de Motores Elétricos**. 2023b. Acesso em: 29/04/2023. Disponível em: <https://static2.weg.net/medias/downloadcenter/h32/hc5/WEG-motores-eletricos-guia-de-especificacao-50032749-brochure-portuguese-web.pdf>. Citado nas pp. 20, 38, 57.

WEG. **Manual do Inversor de Frequência CFW-08**. 2009. Acesso em: 29/04/2023. Disponível em: <https://meloaut.com.br/wp-content/uploads/2021/02/Manual-do-Usuario-CFW08.pdf>. Citado nas pp. 35, 36, 46, 47, 63, 64, 80, 81.

ZHONG, D.; XIA, Z.; ZHU, Y.; DUAN, J. Overview of predictive maintenance based on digital twin technology. **Heliyon**, v. 9, n. 4, e14534, 2023. ISSN 2405-8440. DOI: https://doi.org/10.1016/j.heliyon.2023.e14534. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405844023017413>. Citado na p. 25.

Apêndices

Apêndice A – Protocolo Modbus RTU

O protocolo Modbus inicialmente desenvolvido em 1979 pela fabricante de equipamentos Modicon, atualmente adquirida pela Schneider Electric, é um dos protocolos de comunicação mais antigos e até hoje mais utilizados em redes de CLPs para aquisição de sinais de instrumentos e comando de atuadores. Os direitos do protocolo foram transferidos para a Modbus Organization (MODBUS, 2014) em 2004, de forma que qualquer um pudesse utilizá-lo sem pagar royalties. Pelo fato de ser um protocolo aberto e por se adequar facilmente a diversos meios físicos, é um protocolo bastante famoso e amplamente difundido no ambiente industrial.

Em relação ao seu funcionamento, o protocolo transmite dados analógicos ou digitais por meio de redes seriais que conectam dispositivos eletrônicos, baseando-se em uma estrutura de mensagens compostas por bytes no modelo de comunicação do tipo mestre-escravo (ou servidor-cliente), onde pode haver até 247 escravos, mas somente um mestre, e nunca dois ou mais escravos podem ter o mesmo endereço. Estas mensagens são codificadas e existem dois modos de transmissão de dados especificados pelo protocolo que definem como os bits são transmitidos ao longo da rede e como os dados precisam ser empacotados e decodificados: o ASCII e o RTU. Os dois modos de transmissão não podem ser utilizados concomitantemente em uma mesma rede e, junto a eles, podem ser selecionados outros parâmetros da porta de comunicação serial, como taxa de transmissão, paridade, stop bits (MORAES; CASTRUCCI, 2007).

O CFW-08 utiliza somente o modo RTU para comunicação, não possuindo portanto, comunicação no modo ASCII. Devido a este fato, será focado e apresentado somente este modo de transmissão, onde o ASCII fica a título de curiosidade do leitor. No modo RTU, cada byte de dados é transmitido como sendo uma única palavra com seu valor diretamente em hexadecimal, onde cada palavra transmitida possui 1 start bit, oito bits de dados, 1 bit de paridade (opcional) e 1 stop bit (2 stop bits caso não se use bit de paridade) (WEG, 2009). Desta forma, a seqüência de bits para transmissão de um byte é apresentado na Fig. A.48.

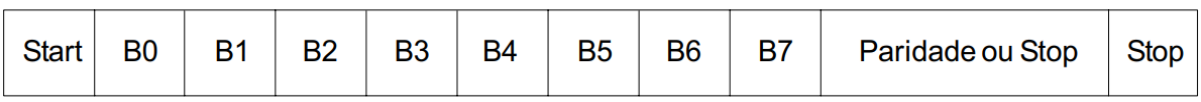


Figura A.48 – Transmissão de dados modo RTU
Fonte: (WEG, 2009)

Os telegramas são organizados utilizando uma mesma estrutura para a comunicação entre o mestre e o escravo em ambos os sentidos, sendo ela: Endereço, Código da Função, Dados e CRC. Apenas o campo de dados poderá ter tamanho variável, dependendo do que está sendo solicitado (WEG, 2009). Esta estrutura é ilustrada na Fig. A.49.

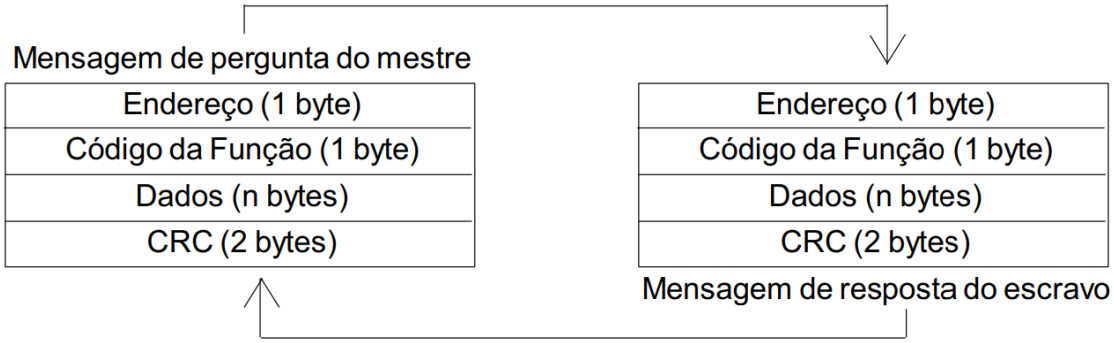


Figura A.49 – Estrutura das mensagens
Fonte: (WEG, 2009)

O campo do endereço contém a identificação do destinatário ou remetente da mensagem transmitida, visto que o endereço é sempre de um dispositivo escravo. O mestre envia um byte com o endereço do escravo que deve receber a mensagem ou pode enviar uma mensagem destinada ao endereço 0 (zero) para todos os escravos da rede. Já o escravo inicia o telegrama com o seu próprio endereço. O campo código da função especifica o tipo de serviço ou função solicitada ao escravo, no caso do CFW-08 apresentados na tabela 3.1. O campo de dados contém a informação em si do telegrama, onde o formato e o conteúdo deste dependem da função utilizada e dos valores transmitidos. Por fim, o CRC é o campo para checagem de erros de transmissão, formado por dois bytes, sendo que o primeiro transmitido é o byte menos significativo (CRC-), e em seguida o mais significativo (CRC+).

Não existe um carácter específico que indique o início ou o fim de um telegrama. O tempo entre as mensagens é utilizado para identificar o começo e o término, onde a ausência de transmissão de dados na rede por um tempo mínimo de 3,5 vezes o tempo de transmissão de uma palavra de dados indica uma nova mensagem, sendo que após esse período a mensagem anterior chegou ao fim. A velocidade da comunicação pode ser ajustada a depender da aplicação, variando-se o *Baud Rate* da porta serial. O valor padrão utilizado quando não se tem um requisito ou especificação é 9600 bit/s, porém existem vários outros valores que podem ser utilizados. A tabela A.10 apresenta alguns valores comuns utilizados e disponíveis para configuração no CFW-08, relacionando-os com o tempo de transmissão de uma palavra de dados T_{11bits} e o intervalo mínimo de tempo entre uma mensagem e outra $T_{3,5x}$. É importante lembrar que todos dispositivos ligados a uma mesma rede devem estar com a mesma taxa de comunicação, visto que não havendo sincronia as mensagens possivelmente irão entrar em conflito e os dispositivos não receberão os dados corretamente.

Taxa de Comunicação	T_{11bits}	$T_{3,5x}$
9600 bits/s	1,146 ms	4,010 ms
19200 bits/s	573 μs	2,005 ms
38400 bits/s	285 μs	1,003 ms

Tabela A.10 – Tempos relacionados com a transferência de telegramas

Apêndice B – Configuração do Conversor Serial na Rede Modbus RTU

A comunicação entre dois dispositivos ou mais dentro de uma única rede Modbus necessita que todos os equipamentos e softwares possuam os mesmos parâmetros de transmissão. O inversor de frequência, o microcontrolador, o sistema supervisório e o sistema de inteligência foram corretamente configurados para se comunicarem, sendo também necessário que o conversor USB/serial RS-485 esteja configurado com os mesmos parâmetros. Como o trabalho foi desenvolvido no Windows 11, será apresentado o passo-a-passo para a configuração da porta serial neste sistema operacional. Caso se deseje fazer no Linux ou Mac, recomenda-se pesquisar os procedimentos corretos para o sistema do usuário.

O primeiro passo para configurar a porta serial é acessar o Gerenciador de Dispositivos e verificar todas as portas COM conectadas ao computador. Se o conversor USB/serial RS-485 estiver conectado e for reconhecido pelo Windows, será possível visualizá-lo como na Fig. B.50, possivelmente com um nome diferente devido ao driver instalado não ser o mesmo. O importante é que ele seja reconhecido como um dispositivo USB-Serial. Entretanto, caso o conversor não seja reconhecido (aparecendo geralmente como "Dispositivo desconhecido"), é possível instalar o driver manualmente diretamente deste site.

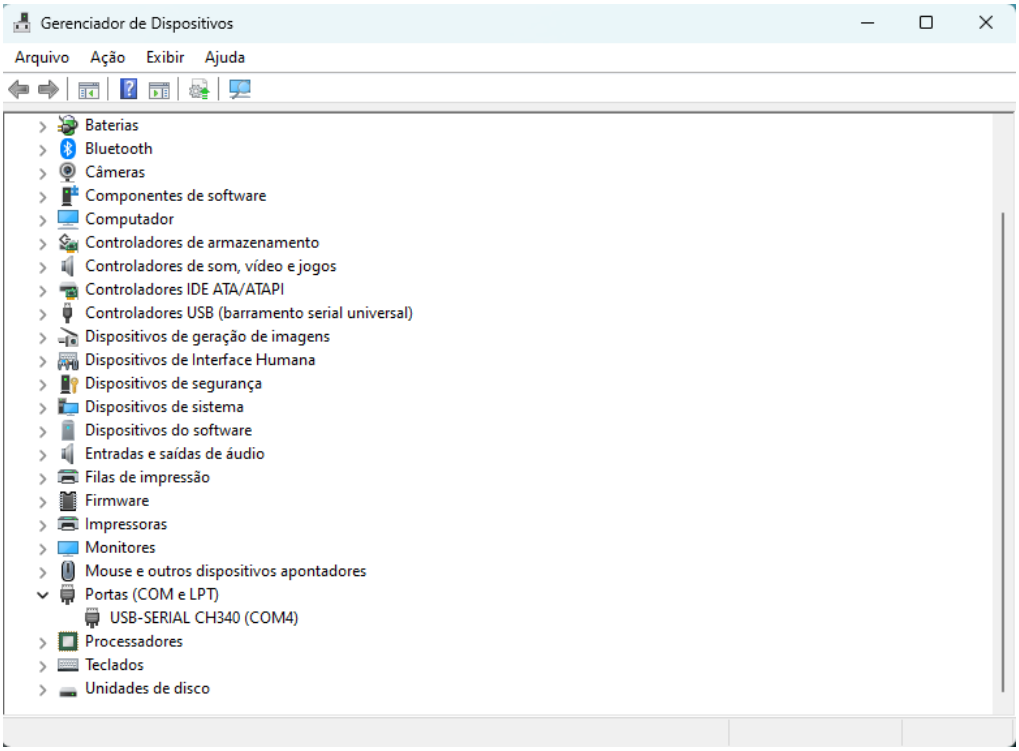


Figura B.50 – Gerenciador de Dispositivos do Windows

Em seguida, sendo reconhecido o conversor, para configurá-lo basta clicar com o botão direito do mouse e acessar as propriedades do dispositivo. Uma janela será aberta e acessando a aba **Configurações de Porta** é possível alterar os campos **Bits por segundo**, **Bits de dados**, **Bits de parada**, **Paridade** e **Controle de Fluxo**, como ilustra a Fig. B.51. Neste caso, como o inversor de frequência trabalha com 2 bits de parada, somente alterou-se este campo.

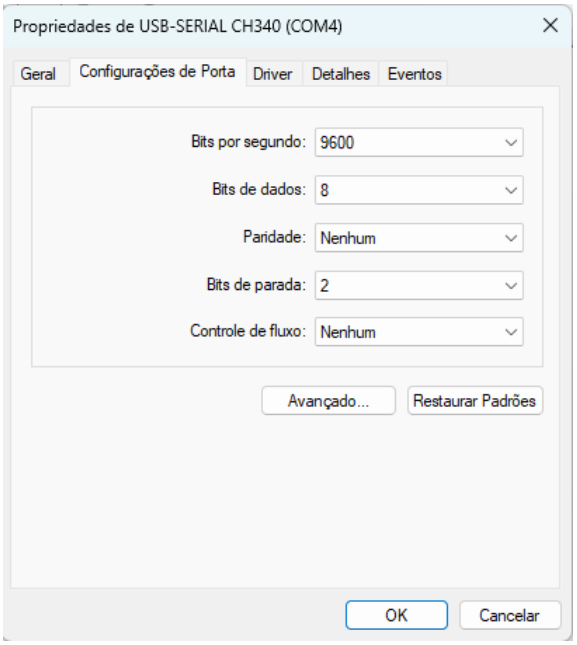


Figura B.51 – Propriedades do Conversor Serial

Anexos

Anexo A – Códigos de programação

Os códigos desenvolvidos para a leitura dos sensores na IDE do Arduino, o projeto do ScadaBR para a reconstrução do sistema supervisório em qualquer máquina, outras informações relacionadas a programação dos dispositivos e documentação para auxiliar na reprodução do trabalho estão disponíveis no *link* abaixo:

- **Repositório do Github:**
  https://github.com/AleXXCunha/Gemeos_Digitais_para_motores_eletricos