



Guilherme Augusto Gubert

**MODERNIZAÇÃO DE BANCADA DE TESTES DE EMPUXO E ELETRÔNICA DE
AERONAVES RADIOCONTROLADAS**

Horizontina-RS

2024

Guilherme Augusto Gubert

**MODERNIZAÇÃO DE BANCADA DE TESTES DE EMPUXO E ELETRÔNICA DE
AERONAVES RADIOCONTROLADAS**

Trabalho Final de Curso apresentado como requisito parcial para a o Trabalho Final de Curso na Engenharia de Controle e Automação da Faculdade Horizontina, sob a orientação do Prof. Dr. Augusto Cesar Huppes da Silva.

Horizontina-RS

2024

FAHOR - FACULDADE HORIZONTINA
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

A Comissão Examinadora, abaixo assinada, aprova o trabalho final de curso

**“Modernização de bancada de testes de empuxo e eletrônica de aeronaves
radiocontroladas”**

**Elaborada por:
Guilherme Augusto Gubert**

Como requisito parcial para a obtenção do grau de Bacharel em
Engenharia de Controle e Automação

Aprovado em: 04/07/2024
Pela Comissão Examinadora

Prof. Dr. Augusto Cesar Huppés da Silva
Presidente da Comissão Examinadora - Orientador

Prof. Dr. Geovane Webler
FAHOR – Faculdade Horizontina

Prof. Dr. Rafael Luciano Dalcin
FAHOR – Faculdade Horizontina

**Horizontina - RS
2024**

RESUMO

O presente trabalho busca resolver os problemas de uma bancada de testes de empuxo de motores já existente, incapaz de atingir a precisão necessária para ter seus dados resultantes utilizados em relatórios. As melhorias implementadas transformaram-na em uma ferramenta muito mais robusta e precisa, ampliando significativamente sua capacidade de análise e contribuindo para o avanço das pesquisas e testes realizados.

Após as modificações, a nova bancada foi capaz de medir o empuxo com a precisão buscada. Em testes realizados com diferentes motores, foram obtidos resultados consistentes, como um empuxo máximo de 4,8 kgf/m e 1074 Watts de potência em um dos motores testados e, já no motor que a equipe utilizará, foram medidos 2,8kgf/m de empuxo com 685 Watts de potência. Estes dados comprovam a eficácia das melhorias implementadas, evidenciando um aumento significativo na confiabilidade das medições.

Este projeto demonstra a importância de aliar tecnologia e inovação para aprimorar equipamentos e processos, resultando em ganhos substanciais de eficiência, precisão e segurança. A nova bancada não apenas atende, mas supera as expectativas iniciais, oferecendo um sistema integrado que facilita a pesquisa e proporciona dados de alta qualidade para análises detalhadas. Estes avanços possibilitam uma análise mais aprofundada dos motores testados, contribuindo significativamente para o avanço das pesquisas na área.

Palavras-chave: *Aerodesign. Brushless. Bancada. Empuxo, ESP32.*

LISTA DE FIGURAS

Figura 1 - Bancada de testes atual	18
Figura 2 - Componentes	21
Figura 3 - Disposição das células de carga	22
Figura 4 - Divisor de tensão.....	23
Figura 5 - Tubo de pitot.....	24
Figura 6 - Programação da dashboard	25
Figura 7 - Dashboard.....	26
Figura 8 – Output da bancada antiga	27

SUMÁRIO

1	INTRODUÇÃO	8
1.1	TEMA.....	8
1.2	DELIMITAÇÃO DO TEMA.....	9
1.3	PROBLEMA DE PESQUISA	9
1.4	HIPÓTESES	9
1.5	OBJETIVOS	10
1.5.1	Objetivo geral	10
1.5.2	Objetivos específicos	10
1.6	JUSTIFICATIVA.....	10
2	REFERENCIAL TEÓRICO.....	12
2.1	A COMPETIÇÃO <i>AERODESIGN</i> SAE	12
2.2	LINGUAGEM DE PROGRAMAÇÃO C++.....	12
2.3	TRANSPORTE DE TELEMETRIA DE ENFILEIRAMENTO DE MENSAGENS (PROTOCOLO MQTT)	13
2.4	PUBLICAÇÃO DE MENSAGENS DO MQTT (MOSQUITTO)	14
2.5	INTERFACE DO USUÁRIO (NODE-RED)	14
2.6	INSTRUMENTAÇÃO	15
3	METODOLOGIA.....	17
3.1	MÉTODOS E TÉCNICAS UTILIZADAS	17
3.1.1	Métodos de procedimentos técnicos	17
3.1.2	Técnicas de coleta de dados	18
3.1.3	Técnicas de análise de dados	19
3.1.4	Fluxograma	19
4	RESULTADOS	21
4.1	MONTAGEM DO ESP32.....	21
4.2	INSTALAÇÃO DO SENSOR DE CORRENTE	21
4.3	INSTALAÇÃO DAS CÉLULAS DE CARGA.....	22
4.4	INSTALAÇÃO DO DIVISOR DE TENSÃO	23
4.5	INSTALAÇÃO DO TUBO DE PITOT	23
4.6	TRANSPORTE DE TELEMETRIA DE ENFILEIRAMENTO DE MENSAGENS (PROTOCOLO MQTT)	24

4.7	BROKER MQTT (MOSQUITTO)	24
4.8	INTERFACE DE USUÁRIO (NODE-RED).....	24
5	ANÁLISE DOS RESULTADOS.....	26
5.1	COLETA DE DADOS DA ANTIGA BANCADA	26
5.2	ALTERAÇÕES APLICADAS	27
5.2.1	COLETA DE DADOS DA BANCADA ATUALIZADA	27
5.2.2	COLETA DE DADOS DE DIFERENTES MOTORES E HÉLICES	28
	CONCLUSÃO	32
	REFERÊNCIAS.....	34
	APÊNDICE A – CÓDIGO FONTE SETUP DO ESP32	36
	APÊNDICE B – CÓDIGO DE INTERAÇÃO DO SENSOR DE CORRENTE	39
	APÊNDICE C - CÓDIGO DE INTERAÇÃO DO SENSOR DE TENSÃO	39
	APÊNDICE D – CÓDIGO DE INTERAÇÃO DE POTENCIA	40
	APÊNDICE E – CÓDIGO DE INTERAÇÃO MQTT.....	40
	APÊNDICE F – CÓDIGO DE INTERAÇÃO DE CÉLULA DE CARGA (EMPUXO)..	40
	APÊNDICE G – CÓDIGO DE INTERAÇÃO DE CÉLULA DE CARGA (TORQUE) .	41
	APÊNDICE H – CÓDIGO DE INTERAÇÃO DO TUBO DE PITOT.....	41
	APÊNDICE I – CÓDIGO DE INTERAÇÃO DA ACELERAÇÃO DO MOTOR	42
	APÊNDICE J – CÓDIGO VBA PARA ATUALIZAÇÃO AUTOMÁTICA DO EXCEL A PARTIR DO ARQUIVO .CSV.....	43
	APÊNDICE K – CÓDIGO VBA DOS BOTÕES DE INICIAR E PARAR COLETA ...	44
	APÊNDICE L – CÓDIGO VBA DO BOTÃO DE LIMPAR LINHAS	46
	APÊNDICE M – CÓDIGO VBA DO BOTÃO DE LEITURA RÁPIDA	46

1 INTRODUÇÃO

O desejo intrínseco do ser humano de voar resultou no desenvolvimento de máquinas mais pesadas que o ar capazes de alçar voos com pouco esforço e percorrer distâncias antes inimagináveis, coisas que só se tornaram possíveis graças à engenhosidade e capacidade humana de resolver problemas. Engenheiros, físicos, matemáticos, qualquer que seja a área relacionada têm um papel importante nesta conquista.

A evolução, trazendo consigo novas problemáticas para o ambiente da aviação, requer estudos precisos e assertivos por parte dos profissionais e, tendo isso como base, a Sociedade dos Engenheiros Automotivos - SAE Brasil, realiza todos os anos, desde 1999, a competição SAE Brasil *Aerodesign* onde alunos das mais diversas instituições de ensino superior do Brasil devem formar equipes que serão desafiadas anualmente com novos regulamentos que contemplam diversas problemáticas reais enfrentadas pela indústria da aviação.

A Equipe MasBáh *Aerodesign* da FAHOR, Faculdade Horizontina, participa destas competições desde 2008 com o intuito de mostrar a capacidade da instituição e dos alunos a trabalharem com problemas não comumente vistos na nossa região. A equipe visa, todos os anos, a melhora de seus processos de engenharia com o desenvolvimento de ferramentas que auxiliem o grupo a se tornar cada vez mais competitivo.

Com a evolução da competição e do surgimento de novas tecnologias, os regulamentos estão cada vez mais rígidos e, com a obrigação de utilização de motores elétricos aos ultrapassados motores a combustão, viu-se a necessidade da modernização da bancada de testes de motores.

1.1 TEMA

O tema deste trabalho é a modernização de uma bancada de testes estáticos que visa a coleta de dados de empuxo do sistema motopropulsor de aeronaves radio controladas e, também, dados relevantes de outros sistemas elétricos da aeronave, como consumo de baterias. A bancada deverá suportar qualquer tipo de motor e bateria disponível no mercado e que sejam permitidos pelos regulamentos da SAE *Aerodesign*.

1.2 DELIMITAÇÃO DO TEMA

O estudo abrangerá o desenvolvimento conceitual e a modelagem de suportes para motores e baterias, juntamente com a programação do sistema de controle da bancada, a criação da interface do usuário e a integração dos sensores essenciais para alcançar os objetivos propostos. Além disso, o trabalho incluirá os testes de bancada realizados pela equipe Masbáh *Aerodesign* em 2024, utilizando os itens previamente adquiridos pela equipe.

1.3 PROBLEMA DE PESQUISA

Na bancada de trabalho atual, os dados são coletados com tacômetros e balanças, sem possibilidade de medir consumo e eficiência de baterias. Além disso, todos os dados precisam ser inseridos manualmente em planilhas de Excel. Portanto, a atualização da bancada deve incluir a coleta automática de todos esses dados e, nas partes móveis, minimizar o atrito para evitar interferências nas medições.

Outro ponto a ser considerado é a segurança de operação do equipamento, onde será necessário abordar formas de prevenir possíveis acidentes de contato com a hélice do grupo motopropulsor girando sem interferir nos fluxos de ar gerados pelos testes.

As informações obtidas desta bancada são fundamentais em todas as fases do desenvolvimento da aeronave, desde definições aerodinâmicas até verificações teóricas de comportamento de voo. Assim, a questão central deste trabalho é: A atualização da bancada de testes garantirá que a aquisição de dados seja precisa o bastante para atender às demandas atuais e futuras da equipe?

1.4 HIPÓTESES

Tendo a pesquisa analisado as possibilidades de atualizações disponíveis, foram levantadas as seguintes hipóteses:

- A atualização da bancada de testes com a inclusão de sensores e aparatos que facilitem a leitura dos dados pelo usuário, trará um ganho de performance para a equipe.
- A utilização da plataforma de desenvolvimento baseada em ESP32, trará um ganho expressivo na coerência e precisão dos dados obtidos,

fazendo com que o desenvolvimento do projeto da aeronave necessite de menos correções pós fabricação.

- A necessidade de uma bancada de testes que funcione independentemente de um computador externo conectado, traz facilidade na utilização visto que todos os componentes para funcionamento estarão embarcados na bancada.

1.5 OBJETIVOS

1.5.1 Objetivo geral

O presente trabalho pretende modernizar as funcionalidades de uma bancada de testes de motores já existente e utilizada pela equipe Masbáh *Aerodesign*.

1.5.2 Objetivos específicos

- Fazer pesquisas na literatura disponível sobre programação da plataforma de desenvolvimento ESP32 e sobre dimensionamento elétrico.
- Resenhar, através de um software CAD, a bancada para que seja possível atualizar componentes mecânicos e acomodar os equipamentos necessários;
- Utilizar a plataforma de desenvolvimento ESP32 e, através da linguagem de programação C++, desenvolver os programas necessários para a integração de todos os sistemas da bancada;
- Criar, através da linguagem de programação *Python*, um software que permita o usuário interagir com a bancada e possa entregar para o usuário todas as informações necessárias;
- Realizar os testes funcionais da bancada;

1.6 JUSTIFICATIVA

A SAE, sendo uma sociedade de engenheiros, busca através de projetos estudantis, como a competição SAE *Aerodesign*, soluções para problemas existentes, por exemplo, na aeronáutica. E a equipe Masbáh, por estar inserida nesta competição, é responsável por solucionar estes problemas e mostrar na prática a solução encontrada.

Este trabalho justifica-se pela importância na precisão da coleta dos dados obtidos através dos testes em bancada e, haja vista que a bancada atual não está mais sendo suficiente para as crescentes demandas de testes e nem trazendo a precisão buscada pela equipe, viu-se a necessidade de uma atualização completa da bancada.

A atualização da bancada de testes da equipe, visa facilitar a obtenção dos dados necessários para tornar possível a solução dos problemas propostos pela competição de *aerodesign*, além de tornar os cálculos, necessários para se obter uma aeronave confiável, manobrável e com o desempenho desejável, mais coerentes.

Considerando que a equipe não dispõe de um túnel de vento, devido ao alto custo, para fazer os testes e pôr a prova os estudos do desenvolvimento da aeronave, a capacidade de se obter dados precisos para contornar essa falta se torna indispensável.

Sendo assim, torna-se necessário a melhoria da bancada para auxiliar a equipe a ter a precisão necessária nos dados para que seja possível o desenvolvimento total do projeto sem a necessidade de efetuar testes em um túnel de vento.

2 REFERENCIAL TEÓRICO

Este capítulo tem por objetivo apresentar conceitos técnicos e aspectos teóricos que baseiam o processo de coleta de dados de motores *brushless* e baterias de LiPo. Também serão apresentados conceitos de linguagem de programação C++ e da instrumentação necessária.

2.1 A COMPETIÇÃO AERODESIGN SAE

A competição SAE *Aerodesign*, concebida pela SAE Internacional, acontece nos Estados Unidos desde 1986 e, desde 1999, esta competição passou a também ser realizada no Brasil, através de programas estudantis da SAE Brasil e tornou-se um evento onde, visivelmente, existe um aumento da qualidade e da quantidade dos projetos participantes. (SAE BRASIL, 2023).

“O programa Competição SAE BRASIL AeroDesign é um desafio lançado aos estudantes de Engenharia que tem como principal objetivo propiciar a difusão e o intercâmbio de técnicas e conhecimentos de Engenharia Aeronáutica, através de aplicações práticas e da competição entre equipes. Ao participar do programa SAE AeroDesign o aluno se envolve com um caso real de desenvolvimento de projeto aeronáutico, desde sua concepção, projeto detalhado, construção e testes.” (SAE Brasil, 2023, Site).

2.2 LINGUAGEM DE PROGRAMAÇÃO C++

A linguagem de programação C++ teve seu desenvolvimento iniciado na década de 80 por Bjarne Stroustrup, e sua versão final oficial foi apresentada em 1985, juntamente com o livro “*The C++ Programming Language*”.

O C++ tem uma enorme variedade de códigos, pois além de seus códigos, pode contar com variedades da linguagem C e estas variedades possibilitam a programação em alto e baixos níveis. O C++ apresenta grande flexibilidade e, embora seja bom, este fato faz com que a programação precise de mais atenção para erros serem evitados. (PACIEVITCH, YURI, 2023).

Alguns fatos sobre o C++:

- O C++ é uma linguagem criada para ser tão eficiente quanto o C, porém com novas funções.
- É uma linguagem que suporta múltiplos paradigmas.

- A linguagem dá liberdade para o programador escolher as opções, mesmo sendo a opção errada.
- Muitos códigos podem ser transferidos para C facilmente, pois o C++ foi criado para ter compatibilidade com o C.
- A linguagem não tem privilégios para alguns grupos de programadores, os comandos são feitos para todas as especialidades de programadores
- Não é necessário um ambiente de desenvolvimento muito potente para o desenvolvimento de C++. (INFOESCOLA, 2023).

2.3 TRANSPORTE DE TELEMETRIA DE ENFILEIRAMENTO DE MENSAGENS (PROTOCOLO MQTT)

Desenvolvido na década de 90, pela IBM, o MQTT (*Message Queuing Telemetry Transport*) é um protocolo baseado em mensagens com o intuito de consumir pouca banda de rede, software e hardware, trabalhando no formato de Cliente/Servidor.

O MQTT tem muita utilidade na área de desenvolvimento de projetos de comunicação entre máquinas e, além disso, pela versatilidade e facilidade de utilização, está muito presente no ambiente IoT (*Internet of Things*).

O Protocolo MQTT opera usando um modelo de *Publish/Subscribe*, no qual os clientes podem publicar e receber informações, enquanto o servidor gerencia a troca de dados. Nesse modelo, um publicador envia mensagens sobre um tópico específico, e um assinante se inscreve nesse tópico para acessar as mensagens. Como não há conexão direta entre publicador e assinante, o MQTT utiliza um gerenciador de mensagens, conhecido como Broker, para mediar essa comunicação. (AUTOMACAO INDUSTRIAL, 2022).

Dessa forma, o Protocolo MQTT foi criado para fornecer uma conexão segura com baixa demanda de recursos e largura de banda. Utilizando o modelo *Publish/Subscribe*, o MQTT permite que os clientes publiquem e recebam informações facilmente, enquanto o servidor gerencia o envio e recebimento desses dados. (SANTOS, GUILHERME, 2022).

2.4 PUBLICAÇÃO DE MENSAGENS DO MQTT (MOSQUITTO)

O Mosquitto é chamado de broker, pois funciona como um intermediário entre máquinas e protocolos para facilitar a comunicação automatizada entre dispositivos. Essa comunicação é especialmente eficaz em aplicações de IoT (*Internet of Things*), permitindo que as máquinas prevejam problemas e operem com alta eficiência, o que é extremamente benéfico para a indústria. (ENGPCESS, 2018).

No presente trabalho, o Mosquitto atuará como um broker MQTT, sendo o responsável por receber as informações do controlador ESP32 e enviá-las para a dashboard.

Dentre os brokers disponíveis, o Mosquitto é um dos mais procurados pela facilidade de instalação e outras vantagens citadas abaixo:

O broker e o protocolo são eficientes em máquinas de alta potência e ideais para sensores residenciais e industriais.

O Mosquitto é conhecido por ser fácil de instalar e pode estar pronto para uso com apenas um clique, em menos de um minuto.

Seu código aberto permite acesso universal e compatibilidade com máquinas antigas e modernas.

2.5 INTERFACE DO USUÁRIO (NODE-RED)

O Node-RED, é uma ferramenta visual de ambiente de código aberto, que inicialmente foi desenvolvida para implementar, criar e/ou conectar dispositivos de IoT (*Internet of Things*), tendo sido estendida posteriormente para hardwares, APIs e *web services*. Assim sendo, por meio dos nodes ou nós, é possível ler arquivos CSV, escutar eventos *http*, *tcp*, *websocket*, *twitter*, *mqtt* entre outros. Ele possui um editor baseado em navegador que além de simples e compatível com

todos os browsers, facilita a conexão de fluxos usando os nós (nodes) do *palette*. (BASÍLIO, SHIRLEY, 2021).

Será utilizado para criar uma interface de usuário intuitiva e eficiente, que servirá também como um *dashboard* para monitoramento em tempo real. Através dessa plataforma, será possível visualizar dados e métricas de forma clara e organizada, facilitando a análise e a tomada de decisões. O Node-RED permite a integração de diversos dispositivos e sensores, proporcionando uma visão abrangente do sistema. Além disso, sua flexibilidade e facilidade de uso tornam a criação e personalização das interfaces rápidas e eficazes. Dessa forma, otimiza-se o gerenciamento e o controle das operações.

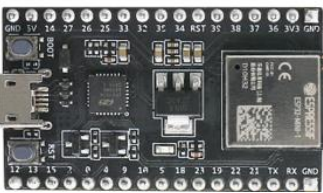
2.6 INSTRUMENTAÇÃO




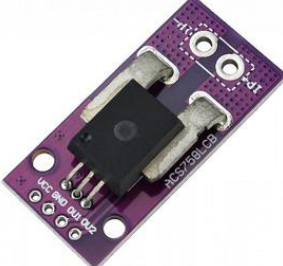



A medição é uma atividade essencial em todas as áreas da tecnologia e da ciência. Se precisamos saber a velocidade de um carro, a temperatura do nosso ambiente de trabalho, a taxa de fluxo de líquido em um tubo, a quantidade de oxigênio dissolvido na água do rio, é importante, portanto, que o estudo da medição faça parte dos cursos de engenharia e ciências no ensino superior. (JOHN P. BENTLEY, 2005).

- Monitoramento: Alguns instrumentos apenas monitoram determinadas informações, mas existem, também, aqueles que recebem e enviam dados brutos para um outro sistema
- Análise de engenharia: Refere-se ao estudo onde serão aplicados os dados recebidos de algum instrumento e são submetidos a análises pós medição.

É importante identificar onde serão utilizados os instrumentos para que se possa escolher a classe correta do equipamento.

Tabela 1: Apresentação dos componentes eletrônicos que compõem a instrumentação do sistema.

<u>Componente</u>	<u>Representação</u>	<u>Função</u>
Placa de desenvolvimento (ESP32)		<u>Controlador principal de bancada de testes. Terá a função de integrar todos os instrumentos, receber a programação para interpretá-los e fornecer os dados para a dashboard.</u>

<p><u>Módulo amplificador HX711</u></p>		<p><u>Conectado a célula de carga, converte os valores de resistência fornecidos por ela em dados digitais.</u></p>
<p><u>Célula de carga</u></p>		<p><u>Sensor eletromecânico que mede força ou peso, utilizando a deformação do material e o fluxo de eletricidade.</u></p>
<p><u>Tubo de pitot</u></p>		<p><u>Ferramenta de medição que calcula a velocidade de vazão de fluídos. Funciona através da pressão dinâmica que é gerada quando o ar da pressão externa penetra no seu tubo, gerando assim a velocidade do fluído.</u></p>
<p><u>Sensor de corrente</u></p>		<p><u>Mede a corrente elétrica utilizando o efeito Hall para detectar o campo magnético gerado pela corrente que passa através dele.</u></p>
<p><u>Sensor reflexivo infravermelho</u></p>		<p><u>Funciona através da detecção do reflexo da luz infravermelha emitida por ele. Como sensor de RPM, interpreta a reflexão da luz em momentos alternados.</u></p>
<p><u>Motor Brushless</u></p>		<p><u>Motor elétrico que opera sem escovas, utilizando um controlador eletrônico para comutar a corrente nas bobinas, resultando em maior eficiência e menor manutenção. Amplamente utilizado em aplicações que exigem alta precisão e durabilidade.</u></p>
<p><u>Controlador eletrônico de velocidade (ESC)</u></p>		<p><u>Tem como função variar a rotação de um motor elétrico DC, através do controle de acionamento das bobinas (motor brushless).</u></p>

<p><u>Bateria LiPo (lítio-polímero)</u></p>		<p><u>É uma bateria recarregável que oferece alta densidade de energia e peso leve, tornando-a ideal para a aplicação.</u></p>
---	---	--

3 METODOLOGIA

A constante busca por melhorias no desenvolvimento do *aerodesign* resultou na necessidade de melhoria da bancada de testes do conjunto moto-propulsor da equipe.

Com base naquilo que foi mencionado, este TFC apresenta-se como um estudo experimental, o qual, conforme indicado por Moresi (2003), constitui uma investigação empírica na qual o pesquisador manipula e regula variáveis independentes, enquanto observa as modificações resultantes dessa manipulação e controle nas variáveis dependentes. Nesse contexto, uma variável se refere a um dado que pode assumir diferentes aspectos, seja em termos de quantidade, qualidade, características ou magnitude, variando em cada situação particular.

A variável independente é aquela que exerce influência, determina ou afeta a variável dependente, sendo reconhecida por surgir previamente como um fator antecedente. Por outro lado, a variável dependente é aquela que será impactada pela variável independente, sendo identificada como uma consequência. A pesquisa experimental oferece a oportunidade de observar e analisar um fenômeno em circunstâncias específicas.

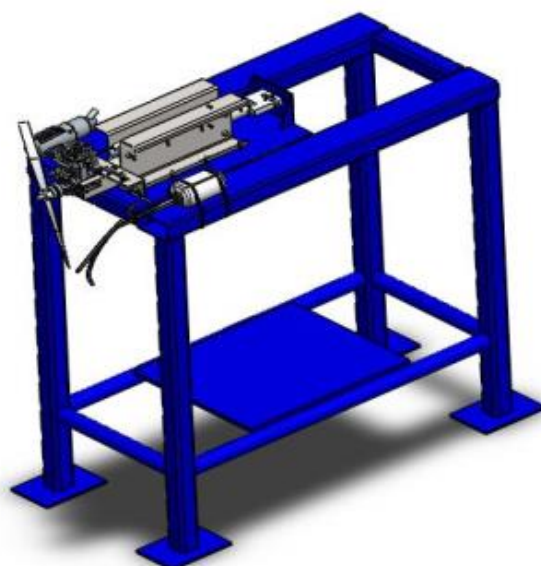
3.1 MÉTODOS E TÉCNICAS UTILIZADAS

3.1.1 Métodos de procedimentos técnicos

Será conduzida uma pesquisa bibliográfica abrangente para identificar pontos cruciais e necessidades relevantes ao projeto, utilizando fontes que tratam diretamente dos temas de interesse. No presente trabalho, adotaremos a pesquisa experimental, conforme descrita por Gil (2017), que envolve a seleção de um objeto de estudo e suas variáveis para realizar experimentos controlados, visando observar os efeitos e obter resultados conclusivos.

Além disso, através de um estudo de caso detalhado, serão analisadas informações da bancada existente, incluindo uma revisão minuciosa do projeto de TFC da bancada original (figura 1), e estudos de TFCs que abordam os equipamentos específicos a serem integrados na bancada atualizada. Essa abordagem permitirá uma compreensão mais profunda das necessidades de atualização e dos desafios técnicos envolvidos, contribuindo para o desenvolvimento eficaz e otimizado da bancada de testes.

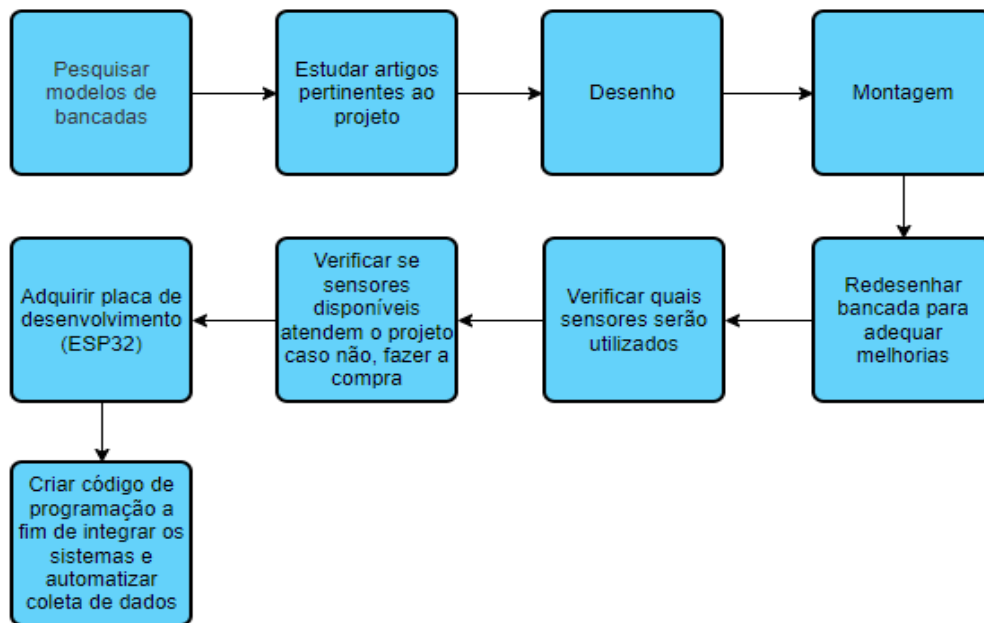
Figura 1 - Bancada de testes atual



Fonte: Leichtweis, 2016.

3.1.2 Técnicas de coleta de dados

Com o objetivo claro de alcançar os objetivos deste trabalho, foi imprescindível estabelecer uma metodologia de pesquisa robusta que delineasse precisamente as técnicas de coleta de dados, conferindo ao projeto uma estrutura clara, organizada e eficaz. Assim, os recursos descritos no fluxograma abaixo serão empregados de forma estratégica para assegurar uma coleta de dados completa e detalhada, essencial para o desenvolvimento e conclusão bem-sucedidos deste estudo:



Fonte: Autor (2024)

3.1.3 Técnicas de análise de dados

Através dos conceitos observados, verificar os que se encaixam de melhor maneira no projeto existente e absorvê-los. Após a bancada ser concluída, realizar os testes para provar sua confiabilidade.

Os dados provenientes dos sensores da bancada serão interpretados pela controladora ESP32 e, através da *WI-FI* e do protocolo de comunicação MQTT, serão inseridos em uma *dashboard* e, posteriormente, no *Excel*, que fará a organização destes dados em tabelas.

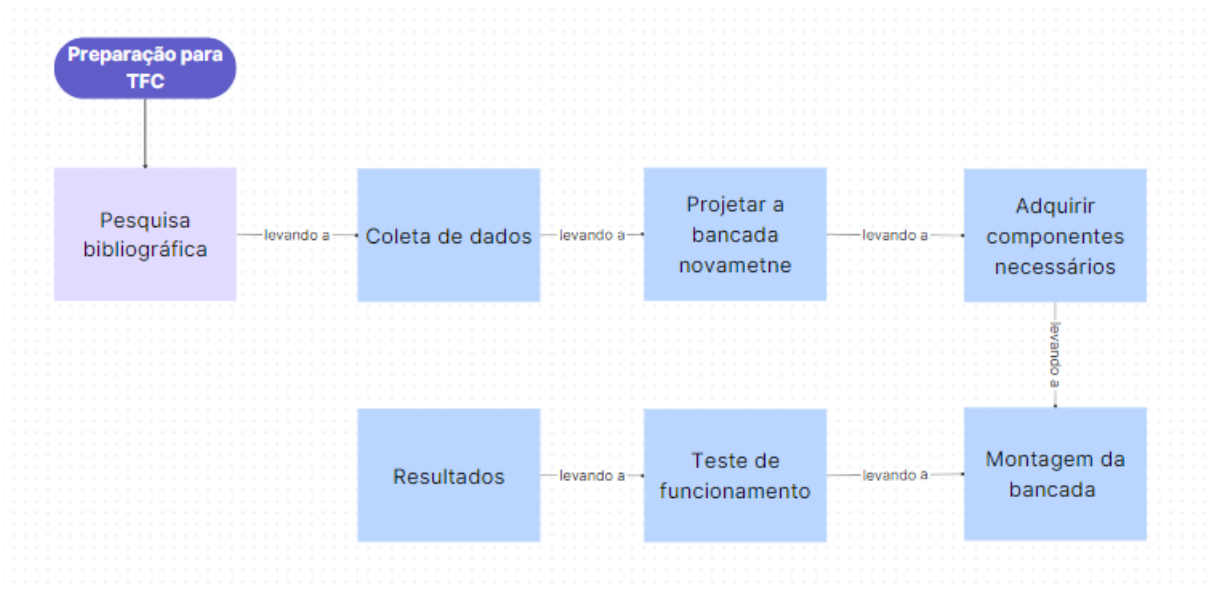
3.1.4 Fluxograma

O fluxograma abaixo delinea as etapas metodológicas do trabalho, começando com a **preparação para TFC** (Trabalho de Conclusão de Curso). Esta etapa inicial envolve organizar e planejar as atividades subsequentes. Em seguida, a **pesquisa bibliográfica** é conduzida para fundamentar teoricamente o projeto, reunindo informações relevantes e atualizadas. A **coleta de dados** vem após, garantindo que todas as informações necessárias para o projeto sejam obtidas de forma precisa.

Com os dados em mãos, a fase de **projetar a bancada novamente** consiste em redesenhar e atualizar a bancada de testes conforme as necessidades identificadas. Segue-se a **aquisição dos componentes necessários**, onde todos os materiais e dispositivos necessários para a montagem da nova bancada são comprados. Na

etapa de **montagem da bancada**, todos os componentes adquiridos são integrados para formar a nova configuração da bancada.

O **teste de funcionamento** é a fase em que a bancada montada é submetida a testes para verificar seu desempenho e eficiência, garantindo que tudo funcione corretamente. Finalmente, a etapa de **resultados** envolve a análise dos dados coletados durante os testes, permitindo tirar conclusões sobre o desempenho da bancada e a eficiência do projeto como um todo.



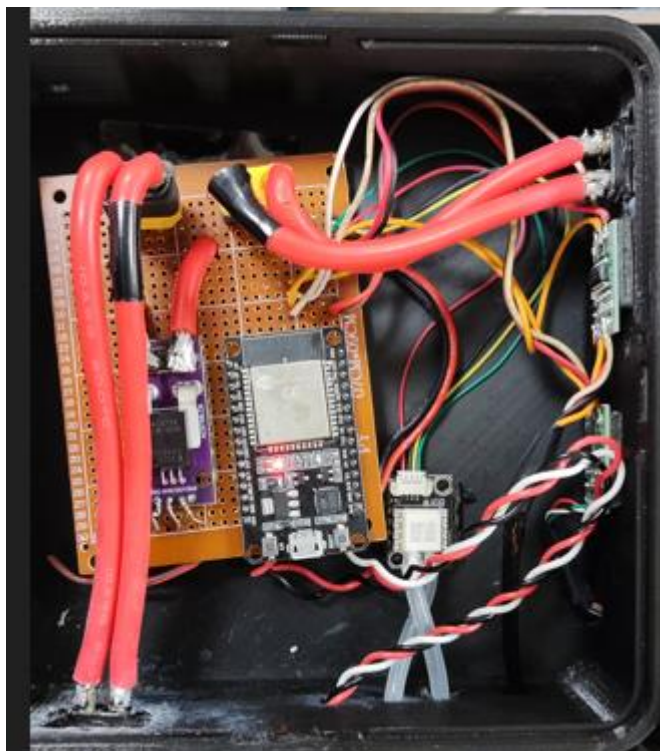
Fonte: Autor (2023)

4 RESULTADOS

4.1 Montagem do ESP32

O ESP32 foi soldado a uma placa de prototipagem de fenolite, o que permite uma melhor organização de todos os componentes e permite a fixação dentro da caixa de acomodação, conforme figura 2.

Figura 2 - Componentes



Fonte: Autor, 2024

4.2 Instalação do sensor de corrente

O sensor de corrente, responsável pela leitura da corrente que circula no sistema, foi instalado ao lado do ESP32 na placa de fenolite, conforme figura 2. É possível a leitura da corrente circulando no sistema, pois o sensor está interrompendo o cabo positivo das baterias, fazendo parte do sistema.

Independente da tensão que circula no sensor, a saída deste, que está ligado a um pino do ESP32, é de 20mv por *Amper*, ou seja, para cada *Amper* circulando no circuito, se obtém uma saída de 20mv para ser interpretada pelo controlador.

O sensor utilizado em questão é o ACS758 de 100A de capacidade, o que é suficiente para os testes necessárias.

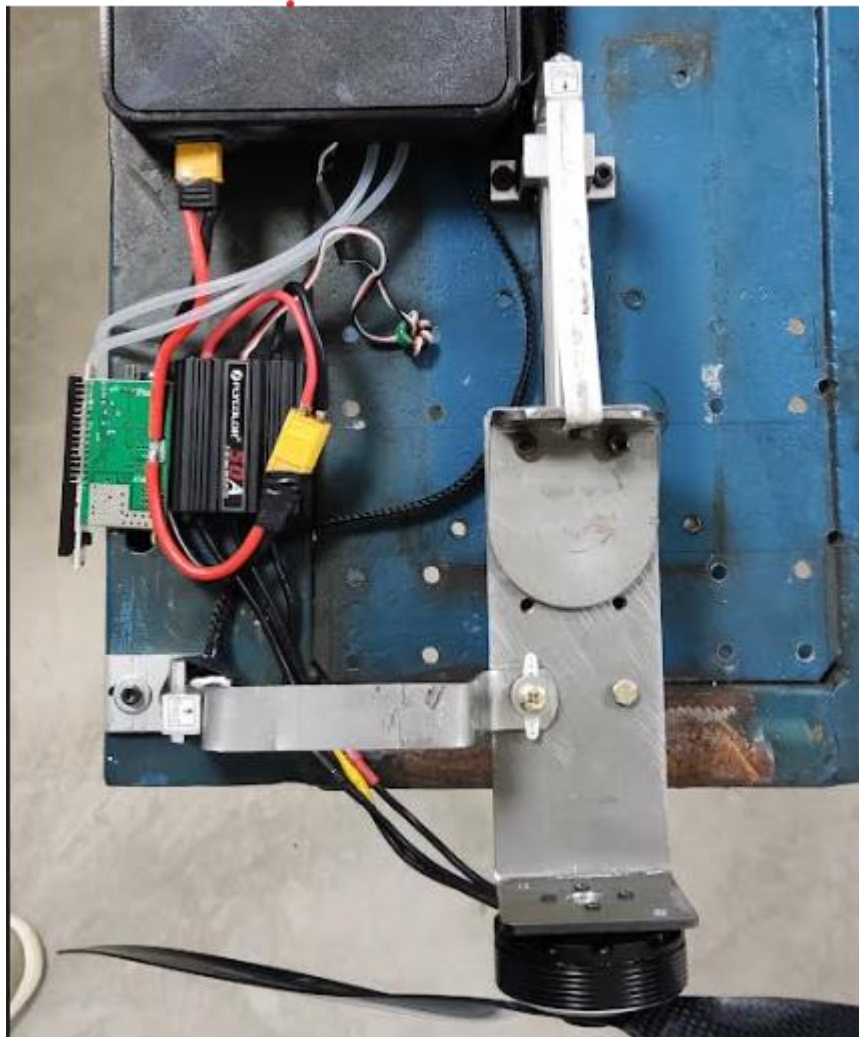
4.3 Instalação das células de carga

As células de carga foram instaladas de acordo com a função que cada uma delas terá. A célula de carga responsável pelo empuxo, foi instalada no mesmo eixo do motor, ou seja, quando o motor traciona, força a célula de carga para frente, fazendo a leitura de empuxo, conforme figura 3.

Já a célula de carga responsável pelo torque, foi instalada a 90° do eixo do motor, conforme figura 3, fazendo uma leitura lateral, que corresponde ao torque produzido pelo motor. Ambas as células de carga necessitam de um módulo HX711, que fornecerá a entrada correta de tensão para o ESP32.

A calibração das células de carga, deu-se através da utilização de uma balança manual, tracionando a célula de carga e comparando os valores de peso entre a balança e o valor resultante da célula.

Figura 3 - Disposição das células de carga

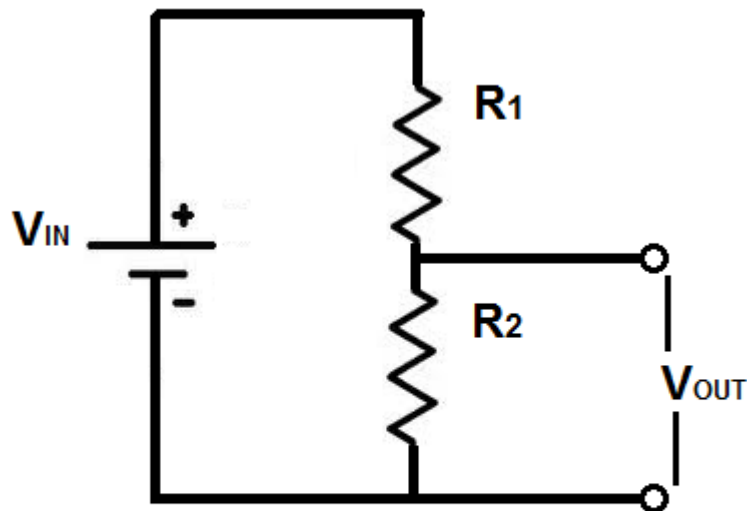


Fonte: Autor, 2024

4.4 Instalação do divisor de tensão

O controlador consegue trabalhar com uma entrada de 3.3v ou menos, então o divisor de tensão construído através de um resistor de 1500 ohms e outro de 220 ohms, tornou os 25 volts máximos das baterias, em 3.20 volts. Os resistores foram soldados juntamente a placa de fenolite, conforme o esquema mostrado na figura 4.

Figura 4 - Divisor de tensão



$$V_{OUT} = V_{IN} \frac{R_2}{(R_1 + R_2)}$$

Fonte: Autor, 2024

4.5 Instalação do tubo de pitot

O tubo de pitot, figura 5, responsável pela referência de velocidade dinâmica do ar, foi colocado em um suporte móvel na bancada pois será necessário movê-lo acima da turbulência de ar que provém da hélice quando testes dinâmicos forem feitos. Estes testes serão efetuados em um momento futuro, ficando este, como melhoria futura para a bancada.

Figura 5 - Tubo de pitot



Fonte: Autor, 2024

4.6 Transporte de telemetria de enfileiramento de mensagens (Protocolo MQTT)

O protocolo MQTT, que trabalha com cliente e servidor, foi o protocolo de comunicação escolhido para ser utilizado neste projeto pois é nativamente suportado pelas bibliotecas do ESP32 e com implementação e programação facilitada.

O apêndice B mostra um exemplo de como o código é implementado para que a variável “CorrenteMQTT” seja publicada no broker.

4.7 Gerenciamento de mensagens MQTT (Mosquitto)

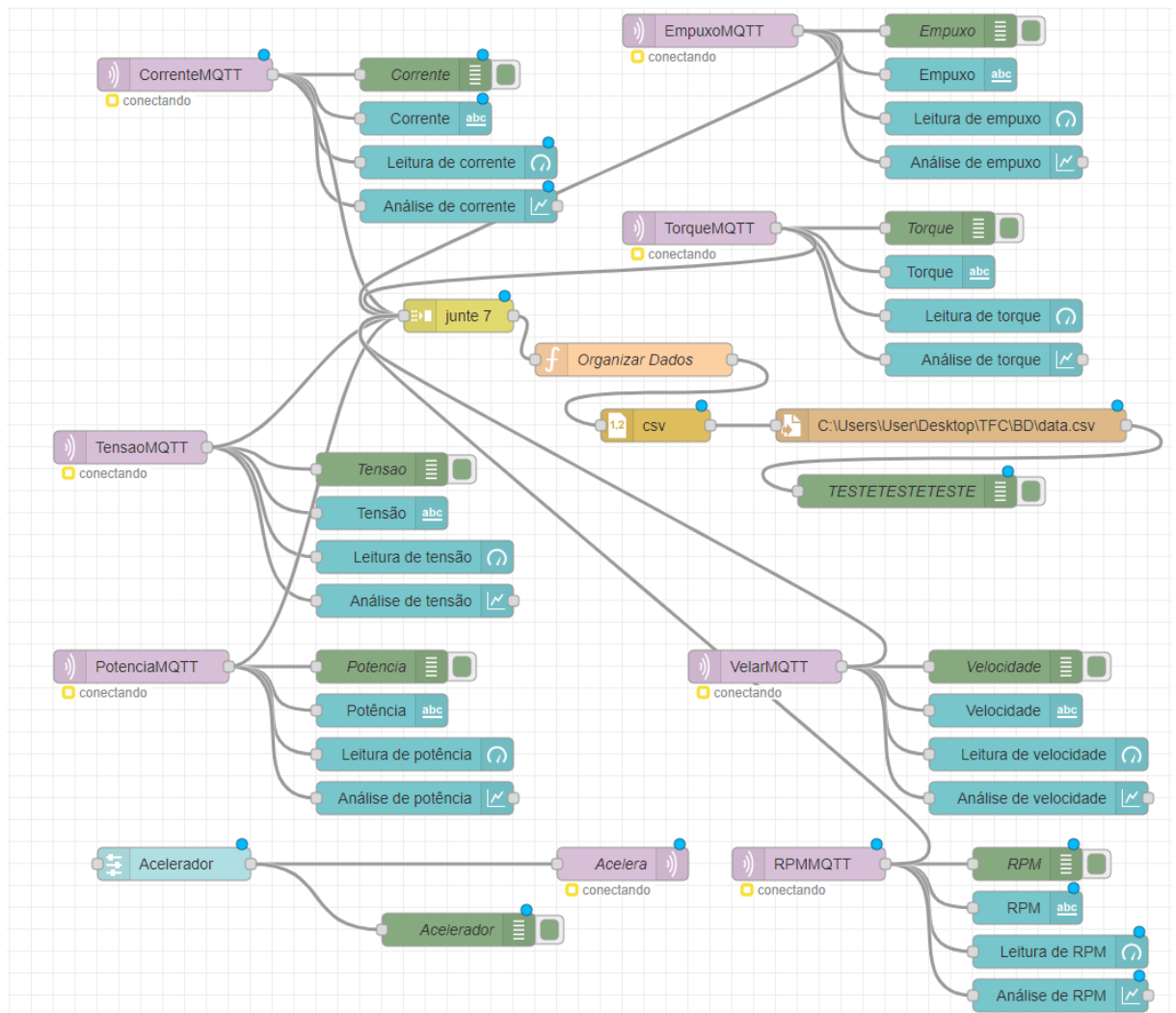
O *Software* Mosquitto, que funciona como o broker MQTT, foi instalado seguindo o manual de operação do mesmo, que consiste no download do *software* e, após a instalação, executá-lo através do *prompt* de comando do *Windows*. Ele já vem previamente configurado para rodar através da porta 1883 e não foi necessário alterá-la.

4.8 Interface de usuário (Node-RED)

A aplicação Node-RED, também foi instalada seguindo seu manual, que consiste no download e instalação do arquivo Node.js. Ele também vem previamente configurado para rodar através da porta 1880, para acessar sua tela de configurações, é necessário abrir um navegador e imputar o endereço <http://127.0.0.1:1880/>. Feito isso, abrirá a tela para criar seus fluxos e, a escolha do Node-RED como interpretador de dados, foi pela sua construção *low-code*, que

permite uma programação fácil e com blocos, conforme é visto na figura 6 a construção de entrada de dados provenientes do *broker* MQTT.

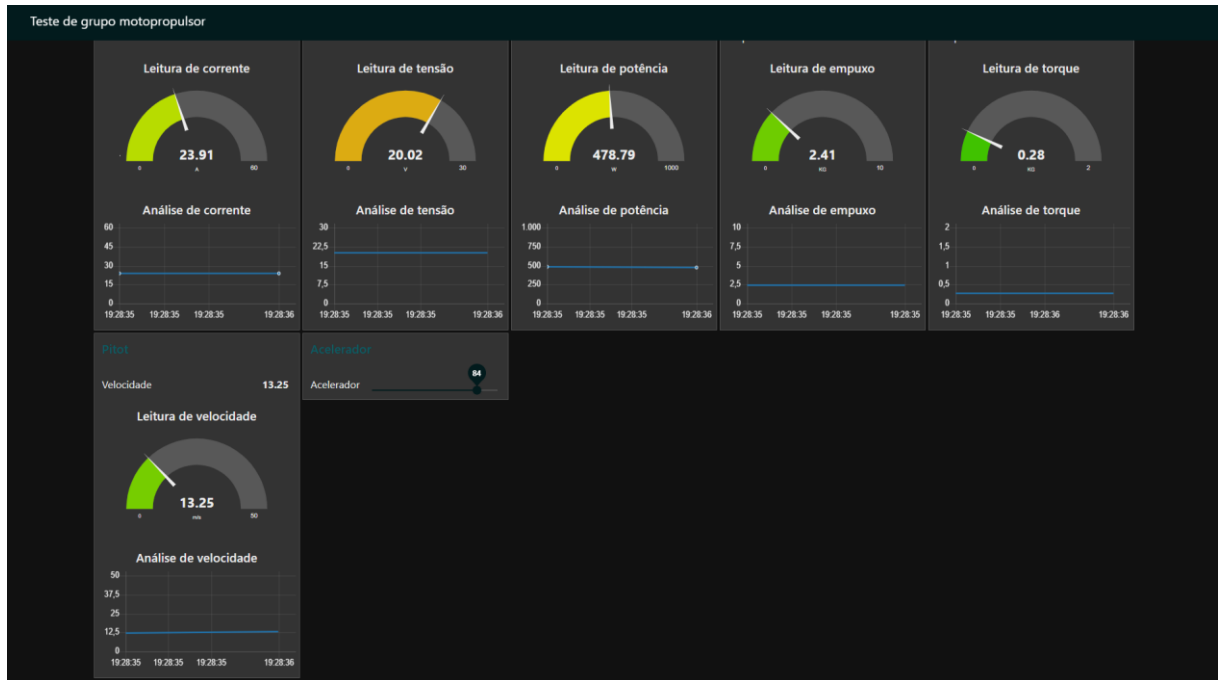
Figura 6 - Programação da dashboard



Fonte: Autor, 2024

Após a programação por blocos ser concluída, é possível visualizar a dashboard, na figura 7, juntamente com o *slider* responsável por fazer a aceleração do motor, trazendo segurança para a operação, pois não é necessária estar perto da bancada para operá-la.

Figura 7 - Dashboard



Fonte: Autor, 2024

A partir do Node-RED, os dados são gravados em um arquivo .csv a cada 200ms.

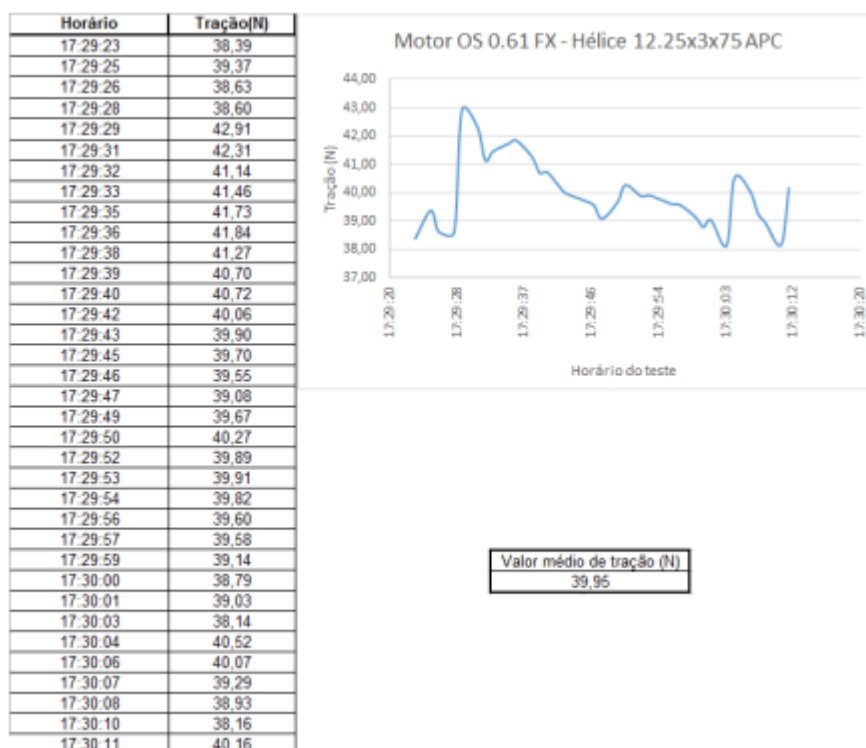
Posteriormente, a cada 1 segundo um arquivo do Excel através de programação VBA, coleta estes dados e separa-os quando encontra uma vírgula e joga cada dado para sua posição correta na tabela.

5 ANÁLISE DOS RESULTADOS

5.1 COLETA DE DADOS DA ANTIGA BANCADA

A bancada anterior, focada na obtenção de dados de motores a combustão interna, buscava coletar informações apenas de empuxo, tendo assim a tração estática como seu principal output. Esse output era coletado através de uma célula de carga conectada a um Arduino, jogando diretamente a informação para uma planilha do Excel, conforme imagem abaixo.

Figura 8 – Output da bancada antiga



Fonte: Leichtweis, 2016

A proposta para a época funcionava, mas com os avanços da tecnologia e as alterações nos regulamentos das competições da SAE, a forma de obtenção e a falta de dados tornou a bancada obsoleta.

5.2 ALTERAÇÕES APLICADAS

O segmento a seguir, tem como objetivo apresentar as alterações e melhorias implementadas na bancada de testes com relação a antiga bancada

5.2.1 Coleta de dados da bancada atualizada

A coleta de dados da bancada atualizada se tornou dinâmica e facilitada, juntamente com uma dashboard (figura 8) onde é possível visualizar o comportamento do teste em tempo real e obter dados precisos.

O quadro 1 mostra a tabela onde os dados são finalmente gravados e, esta tabela, conta com os seguintes botões:

- **Coletar dados:** É responsável por dar o start na coleta de dados, permitindo escolher o melhor momento do teste para dar início a coleta.

- **Parar coleta:** Permite parar a coleta de dados a qualquer momento, se não for pressionado a coleta segue sem interrupções.
- **Limpar dados:** Apaga os valores que foram coletados.
- **Leitura rápida:** Quando pressionado, faz uma leitura rápida de 20 dados, o que permite coletar uma média dos valores em momentos específicos do teste.

Quadro 1 – Output da bancada atualizada

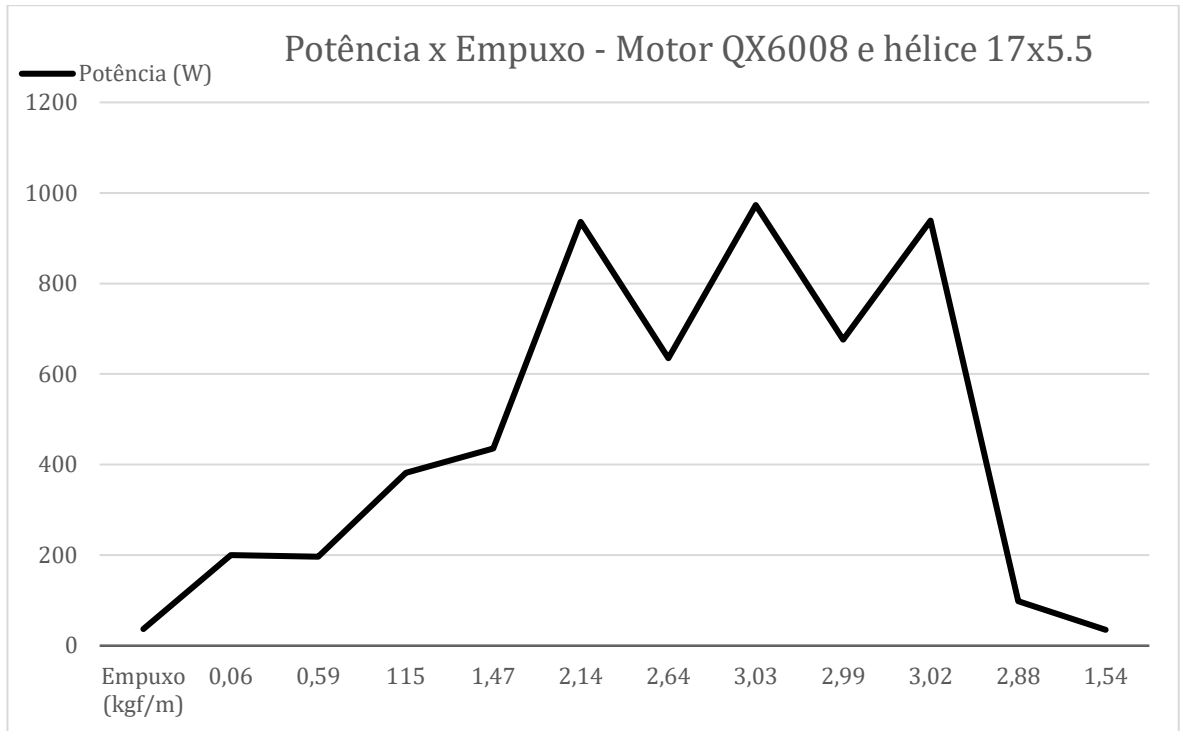
Potencia (W)	Torque (N)	Empuxo (kgf/m)	Corrente (A)	Tensão (V)	Velocidade (m/s)	Rotação (RPM)				
0.17	0.01	0	0.01	21.96	4.64	0				
-2.44	0.02	0	-0.11	21.96	4.93	0	Coletar dados			
0.36	0.02	0	0.02	21.97	4.65	0				
0.79	0.02	0	0.04	21.91	4.45	0				
71.6	0.05	0.23	3.28	21.81	4.69	1452				
99.47	0.06	0.38	4.58	21.74	4.47	1956	Parar coleta			
155.16	0.08	0.68	7.2	21.55	4.39	2230				
262.62	0.09	1.13	12.39	21.19	4.45	2450				
312.58	0.13	1.69	14.92	20.95	4.43	3120				
581.37	0.23	2.35	29.04	20.02	4.95	4510	Limpar dados			
759.87	0.31	2.78	39.29	19.34	4.49	4950				
1147.28	0.37	3.18	61.84	18.55	4.62	5984				
1224.87	0.41	3.4	66.8	18.34	4.31	6523				
1118.02	0.44	3.35	60.76	18.4	4.16	6210	Leitura rápida			
995.42	0.43	3.33	53.44	18.63	4.47	6100				
926.9	0.43	3.28	49.45	18.74	4.3	5489				
884.55	0.42	3.24	47.08	18.79	4.08	5320				
834.53	0.43	3.29	44.27	18.85	4.27	5395				
-0.49	0.21	0.76	-0.02	21	4.35	2116				
-2.32	0.01	0.04	-0.11	21.18	4.56	0				

Fonte: Autor, 2024

5.2.2 Coleta de dados de diferentes motores e hélices

Também foram coletados dados para serem utilizados em relatórios de *aerodesign*, conforme pode ser visto nos gráficos a seguir.

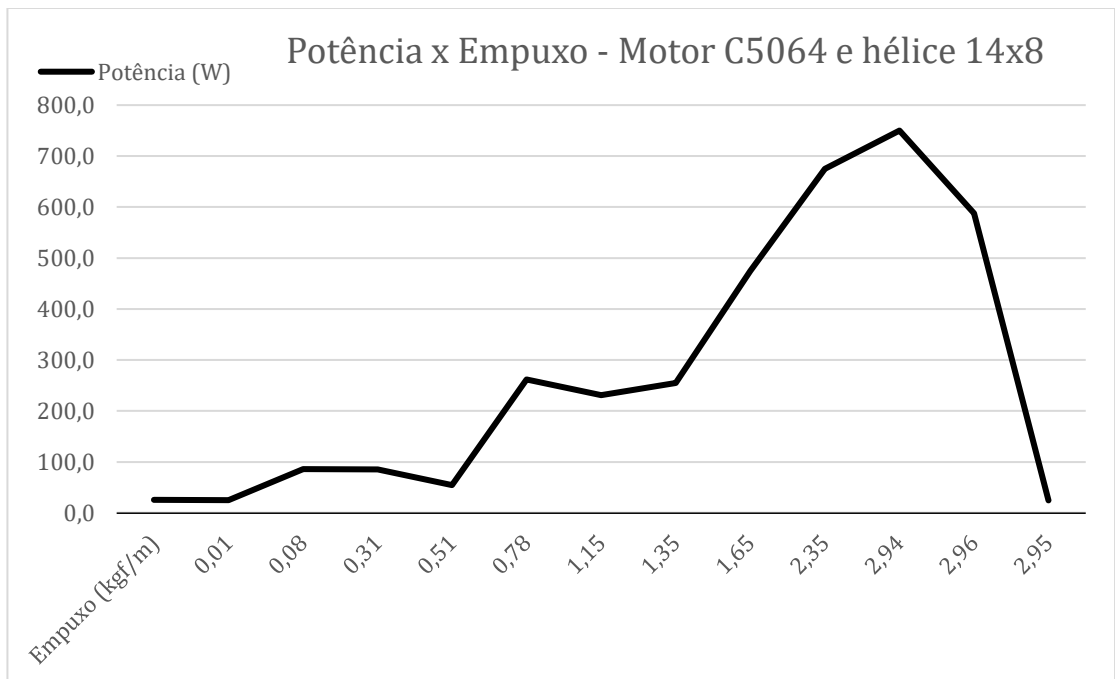
Gráfico 1 - Motor QX6008 e Hélice 17x5.5



Fonte: Autor, 2024

Este motor se mostrou eficiente em todas as suas faixas de rotação, trazendo um empuxo x potência dentro do esperado pela equipe, onde se busca o máximo empuxo possível com no máximo 700 watts de potência.

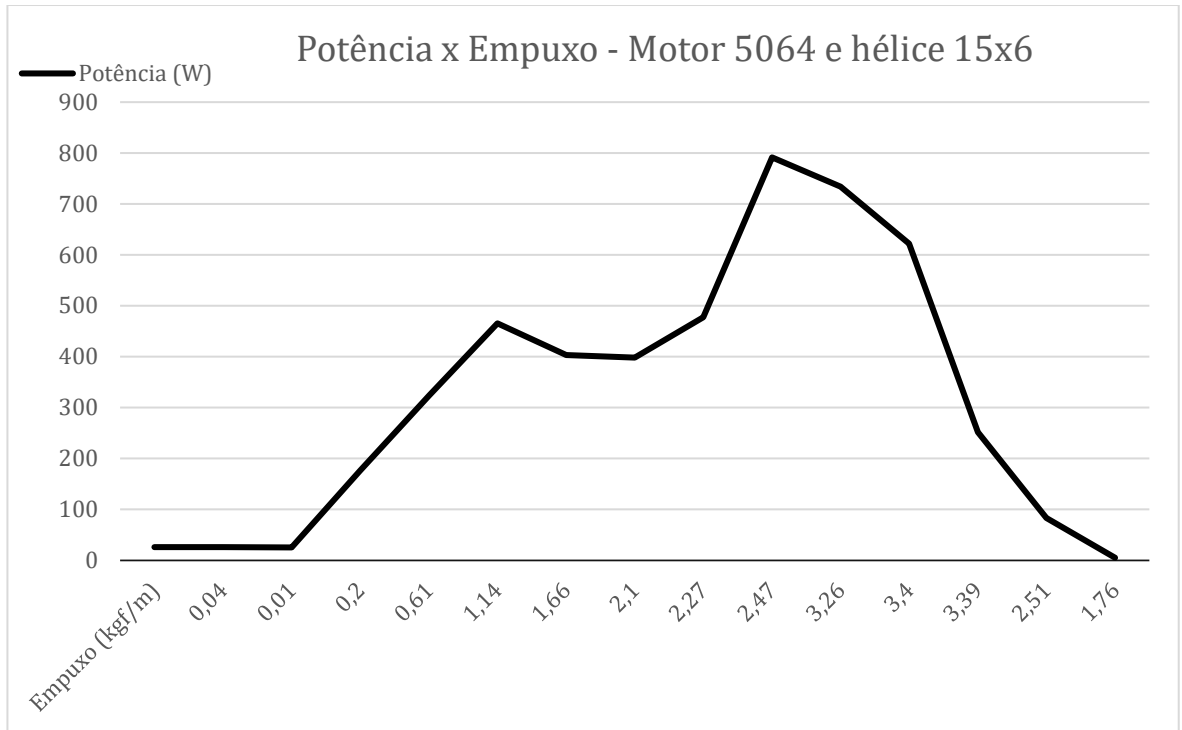
Gráfico 2 - Motor C5064 e hélice 14x8



Fonte: Autor, 2024

Esse motor com a hélice 14x8, mostrou-se eficiente, porém não o suficiente para o desempenho buscado pela equipe.

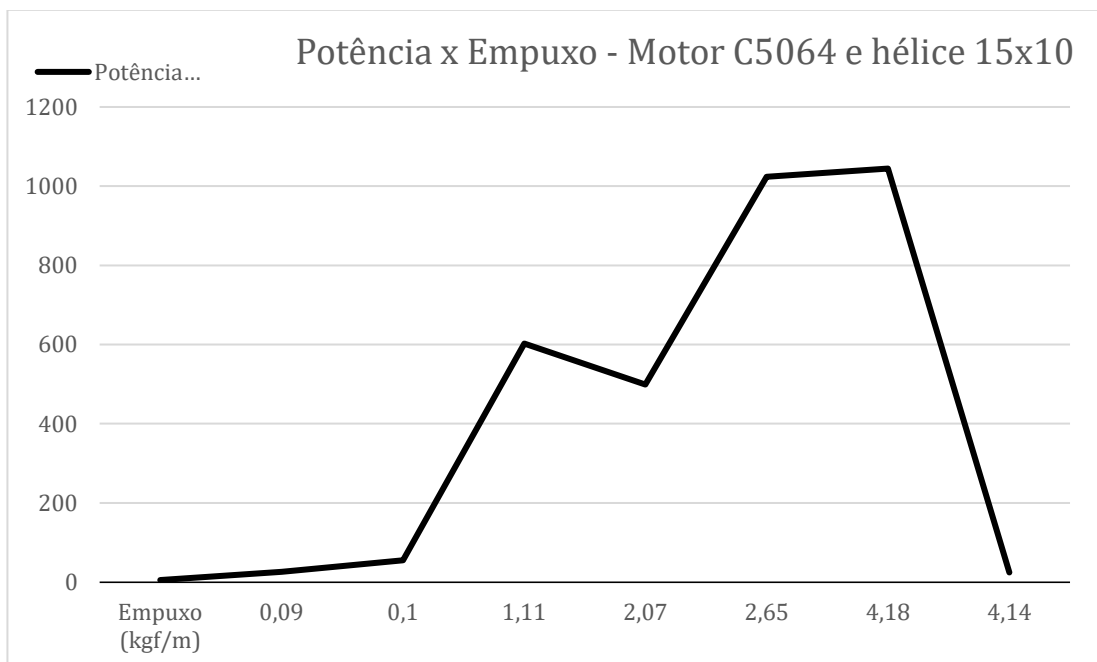
Gráfico 3 - Motor C5064 e Hélice 15x6



Fonte: Autor, 2024

Mantendo o motor do teste anterior e alterando a hélice para uma 15x6, o desempenho melhorou significativamente, tendo um empuxo consideravelmente maior do que os outros testes, conseguindo manter 3.39kgf/m com pouco mais de 600 Watts de potência.

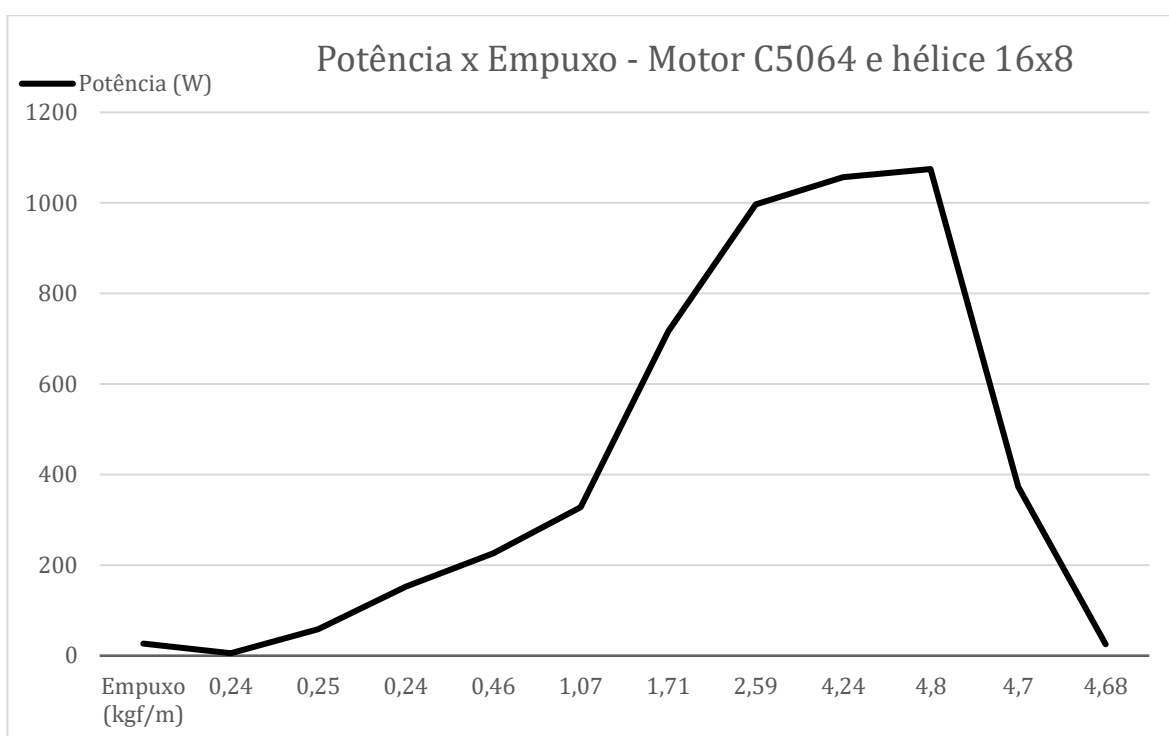
Gráfico 4 - Motor C5064 e Hélice 15x10



Fonte: Autor, 2024

Neste teste, mantendo o motor anterior e aumentando o diâmetro da hélice para 15 polegadas e um passo de 10, verificamos um aumento significativo na potência e, conseqüentemente, um empuxo maior também, porém a potência, 1074 watts, está muito acima do previsto pela equipe, que deve girar em torno dos 700 watts.

Gráfico 5 - Motor C5064 e hélice 16x8



Fonte: Autor, 2024

Ainda mantendo o motor dos testes anteriores e alterando a hélice para uma 16x8, observamos um acentuado aumento no empuxo na potência máxima do motor, 1056 Watts, chegando a 4.8kgf/m, o que seria excelente para o projeto do avião, porém, novamente, a elevada potência não permite sua utilização. Com esse conjunto com a potência girando próximo dos 700 Watts, temos um empuxo de 2.59kgf/m, o que é pouco para o projeto atual de aeronave.

CONCLUSÃO

O desenvolvimento e implementação do Trabalho de Fim de Curso (TFC) proporcionaram significativas melhorias na bancada de empuxo de motor glow existente. A inclusão de diversos sensores e a integração com o ESP32, juntamente com a transmissão de dados via MQTT para o Node-RED e a subsequente organização dos dados em um arquivo Excel por meio de VBA, trouxeram avanços consideráveis em termos de precisão, versatilidade e funcionalidade da bancada.

Entre as melhorias realizadas, destacam-se:

- **Célula de carga para medir empuxo:** A adição de uma célula de carga específica para medir o empuxo aprimorou a precisão e a confiabilidade das medições de força geradas pelo motor.
- **Célula de carga para medir torque:** Esta inclusão permitiu a obtenção de dados detalhados sobre o torque gerado, fornecendo uma visão mais abrangente do desempenho do motor.
- **Divisor de tensão para medir a tensão da bateria:** A integração de um divisor de tensão possibilitou a monitoração contínua e precisa da tensão da bateria, essencial para a avaliação do estado e eficiência energética do sistema.
- **Sensor de corrente:** A medição da corrente consumida pelo motor, realizada por meio de um sensor específico, contribuiu para a análise detalhada do consumo de energia e da performance elétrica.
- **Tubo de Pitot:** Este dispositivo foi fundamental para a medição precisa da velocidade do ar, adicionando uma camada adicional de dados para a análise do comportamento do motor em diferentes condições de operação.
- **Sensor de reflexão para medir o RPM:** A inclusão deste sensor permitiu a monitoração precisa da rotação por minuto do motor, um parâmetro crítico para a avaliação do desempenho.

A integração de todos esses sensores com um ESP32, e a transmissão dos dados para o Node-RED via MQTT, proporcionaram uma solução moderna e eficiente para a coleta e tratamento dos dados. A conexão entre o computador e o ESP32 é feita por WiFi, eliminando a necessidade de conexões físicas e permitindo maior flexibilidade na disposição dos equipamentos. Além disso, a dashboard do

Node-RED permite monitorar todas as informações em tempo real e controlar a aceleração do motor de forma remota, aumentando a segurança da operação ao minimizar a necessidade de intervenção direta durante os testes.

O uso do VBA para organizar os dados em um formato facilmente analisável no Excel facilitou a interpretação dos resultados e a tomada de decisões baseadas em dados concretos.

A transformação da bancada de empuxo de motor glow em uma ferramenta sofisticada e precisa permitiu um avanço significativo na capacidade de análise e na qualidade das pesquisas realizadas. A coleta de dados de empuxo, torque, tensão, corrente, velocidade do ar e RPM de maneira integrada e automatizada representa um progresso substancial em relação ao método anterior, que apenas lia o empuxo diretamente no Excel.

Além de proporcionar dados mais precisos e uma análise mais abrangente, a nova bancada também aumentou a segurança da operação. Com a capacidade de controlar a aceleração do motor via dashboard do Node-RED, os operadores podem realizar testes sem a necessidade de intervenção física direta, reduzindo riscos e melhorando a segurança geral do processo.

Haja vista que, por regulamento da competição, foi definido que a potência do grupo motopropulsor da aeronave não deve superar a potência máxima de 700 Watts, o melhor conjunto de motor e hélice para a equipe será o motor QX6008 e a hélice 17x5.5, pois proporciona um bom empuxo com uma potência ligeiramente menor que o máximo proposto.

Como perspectivas futuras, a bancada será adaptada para testes de empuxo dinâmico, fazendo utilização do tubo de pitot já instalado, reforçando ainda mais a escolha do conjunto motopropulsor da aeronave para a próxima competição.

REFERÊNCIAS

BASÍLIO, SHIRLEY. **O que é Node-RED? Conhecendo e instalando.** Disponível em: <https://blogmasterwalkershop.com.br/outros/o-que-e-node-red-conhecendo-e-instalando>. Acesso em 06 Mar. 2024.

BENTLEY P. JOHN. **Principles of Measurement Systems.** 4. Ed. Inglaterra: Ltc, 2005.

BLOG ASMENG. **Motor Brushless: o que é, como funciona e vantagens.** Disponível em: <https://blog.asmeng.com.br/motor-brushless-o-que-e-como-funciona-vantagens/>. Acessado em 14 Set. 2023.

BLOG ELETROGATE. **ESC: Características e Funcionamento.** Disponível em: <https://blog.eletrogate.com/esc-caracteristicas-e-funcionamento/>. Acessado em 14. Set. 2023.

BLOG DORNENG. **O que você deve saber sobre Baterias Lipo, o combustível do VANT.** Disponível em: <https://blog.droneng.com.br/baterias-lipo/>. Acessado em 15 Set. 2023.

ENGPROCESS. Mosquito – **Entenda suas funcionalidades e vantagens!** 2018. Disponível em: <https://engprocess.com.br/mosquito>. Acesso em 10 Abr. 2024.

ESPRESSIF. **ESP32-DevKitM-1.** Disponível em: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/user-guide-devkitm-1.html>. Acessado em 14 Set. 2023.

ESPRESSIF. **ESP32.** 2023. Disponível em: <https://www.espressif.com/en/products/socs/esp32>. Acesso em 14 Set. 2023.

FLINTEC. **Células de carga.** 2023. Disponível em: <https://www.flintec.com/br/sensores-de-peso/celulas-de-carga/>. Acesso em 10 Set. 2023.

GIL, A. C. **Como elaborar projetos de pesquisa.** 4. ed. São Paulo: Atlas, 2007.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa.** 6. ed. São Paulo: Atlas, 2017.

BERTIN, J. J.; SMITH, M. L. **Aerodynamics for Engineers.** 6. ed. Inglaterra: Pearson Education, 2014.

LEICHTWEIS, AUGUSTO C. DAHMER. **DESENVOLVIMENTO DE UMA BANCADA PARA TESTES DE UM MOTOR DE PISTÃO TIPO GLOW.** 2016.

MARCONI, M. de A.; LAKATOS, E. M. **Metodologia científica.** 7. ed. São Paulo: Atlas, 2012.

MICROSOFT. **Visual Basic for Applications (VBA)**. 2023. Disponível em: <https://learn.microsoft.com/pt-br/office/vba/library-reference/concepts/getting-started-with-vba-in-office>. Acesso em 08 Fev. 2024.

MORESI, EDUARDO. Metodologia da Pesquisa. Brasília, 2003, Disponível em: http://www.unisc.br/portal/upload/com_arquivo/metodologia_da_pesquisa.pdf. Acesso em: 21 Ser. 2013.

MATTEDE, HENRIQUE. **O que é motor brushless? Características e aplicações!** Disponível em: <https://www.mundodaeletrica.com.br/motor-brushless-caracteristicas-aplicacoes/>. Acesso em 14 Set. 2023.

PACIEVITCH, YURI. **C++**. 2023. Disponível em: <https://www.infoescola.com/informatica/cpp/>. Acesso em: 10 set. 2023.

POPPER, Karl S. **A lógica da pesquisa científica**. 2. ed. São Paulo: Cultrix, 1975a.

SAE BASIL. **SAE BRASIL AeroDesign**. 2008. Disponível em: <https://saebrasil.org.br/programas-estudantis/aero-design-sae-brasil/>. Acesso em 10 Set. 2023.

SANTOS, GUILHERME. **Protocolo MQTT: O que é, Como funciona e Vantagens**. Disponível em: <https://www.automacaoindustrial.info/mqtt>. Acesso em 15 Abr. 2024.

TECHTUDO. **Li-Ion vs Li-Po: conheça as diferenças dos tipos de bateria de celular**. Disponível em: <https://www.techtudo.com.br/noticias/2016/08/li-ion-ou-li-po-conheca-diferencas-dos-tipos-de-bateria-de-celular.ghtml>. Acessado em 12 Set. 2023.

USINAINFO. **Módulo Conversor Amplificador HX711**. 2023. Disponível em: <https://www.usinainfo.com.br/amplificadores-de-sinal/modulo-conversor-amplificador-hx711-24bit-2-canais-2818.html>. Acesso em 10 Set. 2023.

USINAINFO. **Sensor de Corrente ACS758**. 2023. Disponível em: <https://www.usinainfo.com.br/sensores-de-corrente/sensor-de-corrente-ac758-50a-bidir-18462.html>. Acesso em 10 Abr. 2024.

USINAINFO. **Divisor de Tensão**. 2023. Disponível em: <https://www.usinainfo.com.br/componentes-eletronicos/divisor-de-tensao-12345.html>. Acesso em 05 Mar. 2024.

APÊNDICE A – CÓDIGO FONTE SETUP DO ESP32

```
#include <WiFi.h>
#include <HX711.h>
#include <PubSubClient.h>
#include <7Wire.h>
#include <math.h>
#include <ms4525do.h>
#include <ESP32_Servo.h>

const char* ssid      = "Bancada_MQTT";
const char* password  = "12345678910";
const char* mqtt_server = "192.168.121.185";
const int  mqtt_Port  = 1883;
const char* mqtt_Topic = "Acelera";

WiFiClient espMQTT;
PubSubClient client(espMQTT);

//Pinos célula de caraga de empuxo
#define pinDT  27
#define pinSCK 14
HX711 scale1;
float medida = 0;

//Pinos célula de caraga de torque
#define pinDTTorque  18
#define pinSCKTorque 19
HX711 scale2;
float medidaTorque = 0;

//////////Leitura de corrente////////////////////////////////////////
#define VIN 34
float Valor = 0;
float Filtro[999];
float Media = 0;
float Valorfinal = 0;
float corr;

//////////Leitura de tensão////////////////////////////////////////
#define TIN 35
float ValorT = 0;
float FiltroT[999];
float MediaT = 0;
float ValorfinalT = 0;
float TensaoT;
```

```

////////////////////Tubo de Pitot////////////////////
bfs::Ms4525do sensor; // Crie uma instância do objeto MS4525DO
const float densidadeAr = 1.158; // Densidade do ar em kg/m³ (valor típico ao
nível do mar)
bool calibrado = false; // Variável para verificar se o sensor foi calibrado
float offset = 29.8; // Variável para armazenar o offset do sensor

#define NUM_READINGS_TO_AVERAGE 20 // Número de leituras para calcular a média

unsigned long previousMillis = 0;
const long interval = 10; // Intervalo de 1 segundo entre leituras

////////////////////Motor////////////////////
Servo Esc30A;
const int pinPWM = 23; // Pino do PWM conectado ao ESC

void setup()
{
//WIFI
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  pinMode(2, OUTPUT); // set the LED pin mode

  delay(10);

  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(5);
    Serial.print(".");

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

//CELULA DE CARGA EMPUXO
    scale1.begin(pinDT, pinSCK); // CONFIGURANDO OS PINOS DA BALANÇA
    scale1.set_scale(89000); // LIMPANDO O VALOR DA ESCALA
    delay(10);
    scale1.tare(); // ZERANDO A BALANÇA PARA DESCONSIDERAR A MASSA DA
ESTRUTURA
    Serial.println("Empuxo Zerado");

//CELULA DE CARGA TORQUE

```

```

    scale2.begin(pinDTTorque, pinSCKTorque); // CONFIGURANDO OS PINOS DA
BALANÇA
    scale2.set_scale(40000); // LIMPANDO O VALOR DA ESCALA
    delay(10);
    scale2.tare(); // ZERANDO A BALANÇA PARA DESCONSIDERAR A MASSA DA
ESTRUTURA
    Serial.println("Torque Zerado");

//LEITURA DE CORRENTE
    analogReadResolution(12); // Definir a resolução do ADC para 12 bits
    pinMode(34, INPUT);

//LEITURA DE TENSÃO
    pinMode(35, INPUT);

//PITOT
    Wire.begin(); // Inicializa a comunicação I2C
    sensor.Config(&Wire, 0x28, 1.0f, -1.0f); // Configura o sensor

//MQTT
client.setServer(mqtt_server, 1883);

//MOTOR
Esc30A.attach(pinPWM); // Associa o pino do ESP32 ligado ao motor na lib Servo
Esc30A.writeMicroseconds(1000); // Envie um sinal neutro para o ESC (pulso de
1500 microssegundos)

}
}

void callback(char *topic, byte * payload, unsigned int length) {
    int sliderValue = 0;
    for (unsigned int i = 0; i < length; i++) {
        sliderValue = sliderValue * 10 + (payload[i] - '0');
    }
    Serial.print("Valor do slider: ");
    Serial.print(sliderValue);
    acelerador(sliderValue);
    Serial.print(" | ");
}

void loop()
{
    //ST_conexao();
    celula_torque();
    celula_empuxo();
}

```

```

corrente();
tensao();
potencia();
reconnect();
pitot();
client.loop();
}

```

APÊNDICE B – CÓDIGO DE INTERAÇÃO DO SENSOR DE CORRENTE

```

void corrente()
{
    for (int i = 0; i <= 999; i++) {
        Valor = analogRead(VIN);
        Valor = map(Valor, 0, 4095, 0, 255);
        Filtro[i] = Valor;
        Media += Filtro[i];
    }
    Valorfinal = Media / 1000;
    Media = 0;
    Filtro[999] = 0;
    Valorfinal = Valorfinal * 0.02972;
    float c_valor = Valorfinal - 2.255;
    corr = c_valor / 0.02;
    Serial.print("V_ESP: ");
    Serial.print(Valorfinal);
    Serial.print(" | ");
    Serial.print("Corrente: ");
    Serial.print(corr);
    client.publish("CorrenteMQTT",String(corr).c_str());
    client.loop();
}

```

APÊNDICE C - CÓDIGO DE INTERAÇÃO DO SENSOR DE TENSÃO

```

void tensao()
{
    for (int v = 0; v <= 999; v++) {
        ValorT = analogRead(TIN);
        ValorT = map(ValorT, 0, 4095, 0, 255);
        FiltroT[v] = ValorT;
        MediaT += FiltroT[v];
    }
    ValorfinalT = MediaT / 1000;
    MediaT = 0;
    FiltroT[999] = 0;
}

```

```

ValorfinalT = ValorfinalT*0.0993;
TensaoT = ValorfinalT;
Serial.print(" | ");
Serial.print("Tensão: ");
Serial.print(TensaoT);
client.publish("TensaoMQTT",String(TensaoT).c_str());
client.loop();
}

```

APÊNDICE D – CÓDIGO DE INTERAÇÃO DE POTENCIA

```

void potencia()
{
float potencia = (corr * TensaoT);
Serial.print(" | ");
Serial.print("Potência: ");
Serial.print(potencia);
client.publish("PotenciaMQTT",String(potencia).c_str());
client.loop();
}

```

APÊNDICE E – CÓDIGO DE INTERAÇÃO MQTT

```

void reconnect()
{
//Serial.print("MQTT Conectado");
if(client.connect("CorrenteMQTT")) {
client.subscribe(mqtt_Topic);
client.setCallback(callback);
//Serial.println("MQTT Conectado");
} else {
Serial.print("Falha na conexão MQTT");
Serial.print(client.state());
Serial.println("Tentando novamente em 3 segundos");
//delay(3000);
client.loop();
}
}

```

APÊNDICE F – CÓDIGO DE INTERAÇÃO DE CÉLULA DE CARGA (EMPUXO)

```

void celula_empuxo() {

medida = scale1.get_units(5); // SALVANDO NA VARIÁVEL O VALOR DA MÉDIA DE 5
MEDIDAS
Serial.print("Empuxo: ");

```



```

Serial.print(medida, 1); // ENVIANDO PARA MONITOR SERIAL A MEDIDA COM 3
CASAS DECIMAIS
Serial.print(" | ");
client.publish("EmpuxoMQTT",String(medida).c_str());
scale1.power_down(); // DESLIGANDO O SENSOR
delay(10); // AGUARDA 5 SEGUNDOS
scale1.power_up(); // LIGANDO O SENSOR
client.loop();
}

```

APÊNDICE G – CÓDIGO DE INTERAÇÃO DE CÉLULA DE CARGA (TORQUE)

```

void celula_torque() {

    medidaTorque = scale2.get_units(5); // SALVANDO NA VARIÁVEL O VALOR DA MÉDIA
DE 5 MEDIDAS
    float absoluto = abs(medidaTorque);
    float torque = (absoluto * 0.11);
    Serial.print("Torque: ");
    Serial.print(torque, 1); // ENVIANDO PARA MONITOR SERIAL A MEDIDA COM 3
CASAS DECIMAIS
    Serial.print(" | ");
    client.publish("TorqueMQTT",String(torque).c_str());
    scale2.power_down(); // DESLIGANDO O SENSOR
    delay(10); // AGUARDA 5 SEGUNDOS
    scale2.power_up(); // LIGANDO O SENSOR
    client.loop();
}

```

APÊNDICE H – CÓDIGO DE INTERAÇÃO DO TUBO DE PITOT

```

#include <Wire.h>

const int sensorAddress = 0x28; // Substitua pelo endereço correto do seu
sensor

void pitot() {
    unsigned long currentMillis = millis();

    // Verifica se passou o intervalo para realizar uma nova leitura
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;

        float velocidadeVento = readWindSpeed(); // Realiza a leitura da
velocidade do vento

        // Imprime o valor da velocidade do vento

```

```

Serial.print("Velocidade do vento: ");
Serial.print(velocidadeVento);
Serial.println(" m/s");
client.publish("VelarMQTT",String(velocidadeVento).c_str());
}

// Outras ações podem ser adicionadas aqui, elas serão executadas a cada
iteração do loop
}

// Função para leitura da velocidade do vento
float readWindSpeed() {
  int numReadings = 0;
  float sumReadings = 0;

  // Loop para realizar múltiplas leituras e calcular a média
  for (int i = 0; i < NUM_READINGS_TO_AVERAGE; i++) {
    // Leitura do sensor de velocidade do ar
    if (sensor.Read()) {
      float pressaoDiferencial = sensor.pres_pa(); // Obtém a pressão
diferencial em pascal (Pa)
      pressaoDiferencial -= offset; // Aplica a calibração
      float velocidadeVento = sqrt(2 * pressaoDiferencial / densidadeAr); //
Calcula a velocidade do vento em m/s

      sumReadings += velocidadeVento;
      numReadings++;
    }
  }

  // Calcula a média das leituras
  return sumReadings / numReadings;
  client.loop();
}

```

APÊNDICE I – CÓDIGO DE INTERAÇÃO DA ACELERAÇÃO DO MOTOR

```

void acelerador(int sliderValue)
{
  int pwmValue = map(sliderValue, 0, 100, 1000, 1800);
  Esc30A.writeMicroseconds(pwmValue);
  //analogWrite(pinPWM, pwmValue);
  client.loop();
}

```

APÊNDICE J – CÓDIGO VBA PARA ATUALIZAÇÃO AUTOMÁTICA DO EXCEL A PARTIR DO ARQUIVO .CSV

```
Option Explicit

Dim atualizacaoAutomatica As Boolean

Sub IniciarAtualizacao()
    atualizacaoAutomatica = True
    AgendarAtualizacao
    ' MsgBox "Atualização automática iniciada."
End Sub

Sub PausarAtualizacao()
    atualizacaoAutomatica = False
    ' MsgBox "Atualização automática pausada."
End Sub

Sub AgendarAtualizacao()
    If atualizacaoAutomatica Then
        ' Agendar a próxima execução do procedimento
        ExecutarAtualizacaoPlanilha em 1 segundo
        Application.OnTime Now + TimeValue("00:00:01"),
        "ExecutarAtualizacaoPlanilha"
    End If
End Sub

Sub ExecutarAtualizacaoPlanilha()
    ' Se a atualização automática estiver ativada, continuar
    If atualizacaoAutomatica Then
        ' Defina o caminho do arquivo CSV
        Dim caminhoArquivo As String
        caminhoArquivo = "C:\Users\User\Desktop\TFC\BD\data.csv"

        ' Verifique se o arquivo existe
        If Dir(caminhoArquivo) = "" Then
            MsgBox "O arquivo CSV não foi encontrado!", vbExclamation
            Exit Sub
        End If

        ' Desative atualizações de tela para melhor desempenho
        Application.ScreenUpdating = False

        ' Verifique se a planilha "Dados" existe
        On Error Resume Next
        Dim planilhaDados As Worksheet
        Set planilhaDados = ThisWorkbook.Sheets("Dados")
        On Error GoTo 0
    End If
End Sub
```

```

If planilhaDados Is Nothing Then
    MsgBox "A planilha 'Dados' não foi encontrada!", vbExclamation
    Exit Sub
End If

' Importe os dados do arquivo CSV
Dim novaLinha As Long
novaLinha = planilhaDados.Cells(planilhaDados.Rows.Count,
1).End(xlUp).Row + 1
With planilhaDados.QueryTables.Add(Connection:="TEXT;" &
caminhoArquivo, Destination:=planilhaDados.Cells(novaLinha, 1))
    .TextFileParseType = xlDelimited
    .TextFileCommaDelimiter = True ' Use True se o delimitador for uma
vírgula, ou False se for ponto e vírgula
    .TextFileStartRow = 1
    .TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1) ' Define o tipo
de dados para todas as colunas como texto
    .Refresh
End With

' Ative as atualizações de tela novamente
Application.ScreenUpdating = True

' Travar a largura das colunas A a F em 15 unidades após a colagem dos
dados
Dim col As Variant
For Each col In Array("A", "B", "C", "D", "E", "F")
    planilhaDados.Columns(col).ColumnWidth = 16
Next col

' Agendar a próxima execução do procedimento
ExecutarAtualizacaoPlanilha em 1 segundo
AgendarAtualizacao
End If
End Sub

```

APÊNDICE K – CÓDIGO VBA DOS BOTÕES DE INICIAR E PARAR COLETA

```

Option Explicit

Dim atualizacaoAutomatica As Boolean

Sub IniciarAtualizacao()
    atualizacaoAutomatica = True
    AgendarAtualizacao
    ' MsgBox "Atualização automática iniciada."
End Sub

```

```

Sub PausarAtualizacao()
    atualizacaoAutomatica = False
    ' MsgBox "Atualização automática pausada."
End Sub

Sub AgendarAtualizacao()
    If atualizacaoAutomatica Then
        ' Agendar a próxima execução do procedimento
        ExecutarAtualizacaoPlanilha em 1 segundo
        Application.OnTime Now + TimeValue("00:00:01"),
        "ExecutarAtualizacaoPlanilha"
    End If
End Sub

Sub ExecutarAtualizacaoPlanilha()
    ' Se a atualização automática estiver ativada, continuar
    If atualizacaoAutomatica Then
        ' Defina o caminho do arquivo CSV
        Dim caminhoArquivo As String
        caminhoArquivo = "C:\Users\User\Desktop\TFC\BD\data.csv"

        ' Verifique se o arquivo existe
        If Dir(caminhoArquivo) = "" Then
            MsgBox "O arquivo CSV não foi encontrado!", vbExclamation
            Exit Sub
        End If

        ' Desative atualizações de tela para melhor desempenho
        Application.ScreenUpdating = False

        ' Verifique se a planilha "Dados" existe
        On Error Resume Next
        Dim planilhaDados As Worksheet
        Set planilhaDados = ThisWorkbook.Sheets("Dados")
        On Error GoTo 0

        If planilhaDados Is Nothing Then
            MsgBox "A planilha 'Dados' não foi encontrada!", vbExclamation
            Exit Sub
        End If

        ' Importe os dados do arquivo CSV
        Dim novaLinha As Long
        novaLinha = planilhaDados.Cells(planilhaDados.Rows.Count,
        1).End(xlUp).Row + 1
        With planilhaDados.QueryTables.Add(Connection:="TEXT;" &
        caminhoArquivo, Destination:=planilhaDados.Cells(novaLinha, 1))
            .TextFileParseType = xlDelimited

```

```

        .TextFileCommaDelimiter = True ' Use True se o delimitador for uma
vírgula, ou False se for ponto e vírgula
        .TextFileStartRow = 1
        .TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1) ' Define o tipo
de dados para todas as colunas como texto
        .Refresh
    End With

    ' Ative as atualizações de tela novamente
    Application.ScreenUpdating = True

    ' Travar a largura das colunas A a F em 15 unidades após a colagem dos
dados
    Dim col As Variant
    For Each col In Array("A", "B", "C", "D", "E", "F")
        planilhaDados.Columns(col).ColumnWidth = 16
    Next col

    ' Agendar a próxima execução do procedimento
    ExecutarAtualizacaoPlanilha em 1 segundo
    AgendarAtualizacao
End If
End Sub

```

APÊNDICE L – CÓDIGO VBA DO BOTÃO DE LIMPAR LINHAS

```

Sub LimparLinhas()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Dados") ' Substitua "Nome_da_Sua_Planilha"
pelo nome da sua planilha

    Dim linha As Long
    Dim coluna As Variant

    ' Percorre todas as linhas de A2 a A100 e F2 a F100
    For linha = 2 To 100
        For Each coluna In Array("A", "B", "C", "D", "E", "F")
            If Not IsEmpty(ws.Cells(linha, coluna)) Then
                ws.Cells(linha, coluna).ClearContents
            End If
        Next coluna
    Next linha
End Sub

```

APÊNDICE M – CÓDIGO VBA DO BOTÃO DE LEITURA RÁPIDA

```

Dim contador As Integer

```

```

Sub IniciarLeituraContinua()
    contador = 0
    LerDadosDoCSV
End Sub

Sub LerDadosDoCSV()
    ' Defina o caminho do arquivo CSV
    Dim caminhoArquivo As String
    caminhoArquivo = "C:\Users\User\Desktop\TFC\BD\data.csv"

    ' Verifique se o arquivo existe
    If Dir(caminhoArquivo) = "" Then
        MsgBox "O arquivo CSV não foi encontrado!", vbExclamation
        Exit Sub
    End If

    ' Abrir o arquivo CSV somente para leitura
    Dim arquivoCSV As Workbook
    Set arquivoCSV = Workbooks.Open(FileName:=caminhoArquivo, ReadOnly:=True)

    ' Defina a planilha "Dados" onde os dados serão colados
    Dim planilhaDados As Worksheet
    Set planilhaDados = ThisWorkbook.Sheets("Dados")

    ' Encontre a próxima linha vazia na coluna A da planilha "Dados"
    Dim proximaLinha As Long
    proximaLinha = planilhaDados.Cells(planilhaDados.Rows.Count,
1).End(xlUp).Row + 1

    ' Determinar a última linha com dados no arquivo CSV
    Dim ultimaLinhaCSV As Long
    ultimaLinhaCSV =
arquivoCSV.Sheets(1).Cells(arquivoCSV.Sheets(1).Rows.Count, 1).End(xlUp).Row

    ' Copiar os dados do arquivo CSV para a planilha "Dados"
    Dim intervaloCSV As Range
    Set intervaloCSV = arquivoCSV.Sheets(1).Range("A1:F" & ultimaLinhaCSV)

    ' Copiar dados diretamente para a planilha "Dados"
    intervaloCSV.Copy Destination:=planilhaDados.Cells(proximaLinha, 1)

    ' Fechar o arquivo CSV sem salvar
    arquivoCSV.Close SaveChanges:=False

    ' Atualizar contador
    contador = contador + 1

    ' Verificar se já foram feitas 20 leituras

```

```
If contador < 20 Then
    ' Realizar a próxima leitura dos dados do CSV
    LerDadosDoCSV
Else
    ' Não há nenhuma mensagem de conclusão aqui
    contador = 0
End If
End Sub
```