

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA
DEPARTAMENTO DE ELETROELETRÔNICA
CURSO DE ENGENHARIA ELÉTRICA

ISABELLI SASDELLI TAVARES

**ANÁLISE DOS INDICADORES DE DESEMPENHO DA MÁQUINA
REACH STACKER APÓS A IMPLEMENTAÇÃO DE *SOFTWARE*
TESTADOR DE *JOYSTICK* NO SETOR DE MANUTENÇÃO
PORTUÁRIA**

TRABALHO DE CONCLUSÃO DE CURSO

ITAJAÍ
2022

ISABELLI SASDELLI TAVARES

**ANÁLISE DOS INDICADORES DE DESEMPENHO DA MÁQUINA
REACH STACKER APÓS A IMPLEMENTAÇÃO DE *SOFTWARE*
TESTADOR DE *JOYSTICK* NO SETOR DE MANUTENÇÃO
PORTUÁRIA**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica da Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina - IFSC Campus Itajaí, como requisito parcial para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Sergio Augusto Bitencourt Petrovcic, Dr. Eng.

ITAJAÍ
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca do IFSC.

Sasdelli Tavares, Isabeli

Análise dos indicadores de desempenho da máquina reach
stacker após a implementação de software testador de
joystick no setor de manutenção portuária / Isabeli
Sasdelli Tavares ; orientador, Sergio Augusto Bitencourt
Petrovcic, 2022.

70 p.

Trabalho de Conclusão de Curso (graduação) - Instituto
Federal de Santa Catarina, Campus Itajaí, Graduação em
Engenharia Elétrica , Itajaí, 2022.

Inclui referências.

1. Engenharia Elétrica . 2. Testador de joystick . 3.
Análise de indicadores de desempenho. 4. Teste de
usabilidade. 5. Reach stacker. 6. Desenvolvimento de
software. I. Petrovcic, Sergio Augusto Bitencourt. II.
Instituto Federal de Santa Catarina. Graduação em
Engenharia Elétrica . III. Título.



Ministério da Educação
Instituto Federal de Educação, Ciência e Tecnologia
de Santa Catarina
Câmpus Itajaí
Coordenação do Curso de Engenharia Elétrica



TERMO DE APROVAÇÃO

**ANÁLISE DOS INDICADORES DE DESEMPENHO DA MÁQUINA
REACH STACKER APÓS A IMPLEMENTAÇÃO DE SOFTWARE
TESTADOR DE JOYSTICK NO SETOR DE MANUTENÇÃO
PORTUÁRIA**

por

ISABELLI SASDELLI TAVARES

Este trabalho foi julgado adequado para obtenção do Título de Engenheiro Eletricista e aprovado na sua forma final pela banca examinadora do curso de engenharia elétrica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

BANCA EXAMINADORA:



Documento assinado digitalmente
GUILHERME RANZOLIN PIAZZETTA
Data: 21/09/2022 23:34:08-0300
CPF: 048.696.069-25
Verifique as assinaturas em <https://v.ifsc.edu.br>

Prof. Guilherme Ranzolin Piazzetta, Msc.
Instituto Federal de Santa Catarina



Documento assinado digitalmente
LUIS FERNANDO POZAS
Data: 22/09/2022 15:25:11-0300
CPF: 278.823.298-51
Verifique as assinaturas em <https://v.ifsc.edu.br>

Prof. Luis Fernando Pozas, Dr. Eng.
Instituto Federal de Santa Catarina



Documento assinado digitalmente
SERGIO AUGUSTO BITENCOURT PETROVIC
Data: 22/09/2022 09:18:41-0300
CPF: 026.649.159-67
Verifique as assinaturas em <https://v.ifsc.edu.br>

Prof. Sergio Augusto Bitencourt Petrovcic, Dr. Eng.
Instituto Federal de Santa Catarina

Itajaí, 6 de setembro de 2022.

Este trabalho é, especialmente, dedicado ao meu pai Valdemir Lauro Tavares e minha mãe Silvana Maria Sasdelli Tavares.

AGRADECIMENTOS

Ao meu professor orientador, Sergio Augusto Bitencourt Petrovcic, por aceitar o desafio da orientação, pela paciência, instrução e sabedoria, desde muito antes deste trabalho de conclusão de curso.

Ao meu supervisor, Thiago Augusto Pereira, pelo apoio, incentivo e oportunidade cedida ainda durante a realização do estágio curricular.

Aos meus colegas de trabalho, especialmente Leandro Garcia Manhas e André Araújo dos Santos, pelos conhecimentos compartilhados e suportes prestados.

Ao Instituto Federal de Santa Catarina, pela educação pública, gratuita e de qualidade prestada à sociedade brasileira, que através de editais de projetos de pesquisa, como o projeto PVITJ2439/2021 do edital interno 38/2021/PROPPI, propiciam oportunidades aos diversos alunos da comunidade.

Aos meus amigos e familiares que, durante toda minha trajetória acadêmica, incentivaram-me, encorajaram-me e não mediram esforços para eu alcançar meus objetivos.

“Uma mente que se abre a uma nova ideia jamais volta ao seu tamanho original.”

(Albert Einstein)

RESUMO

Um dos equipamentos mais utilizados em ambientes portuários é a empilhadeira tipo *reach stacker*, responsável pela movimentação de contêineres por curtas distâncias. Ela é dividida em alguns principais conjuntos, como lança, *spreader* e cabine do operador. Dentro da cabine do operador, há o componente eletrônico *joystick*, responsável por realizar todas as funções de movimentação da lança (esticar, recolher, subir e descer), do *spreader* (abrir, fechar, deslocar, girar, inclinar e apalpar), além de função de comunicação via rádio e buzina. O *joystick*, ao apresentar defeito, passa por uma série de passos até encontrar a causa raiz, desde retirar do equipamento e instalar em outro para confirmação do problema, até o envio para reparo externo. Esse procedimento, além de demorado, é muito oneroso. Assim, é de extrema relevância o desenvolvimento de um testador de *joystick*, a fim de detectar rapidamente as falhas e permitir o conserto do componente de forma assertiva. Este trabalho visa, então, aprimorar o *software* e *hardware* inicialmente desenvolvido, analisar o impacto da implementação no setor de manutenção através dos principais indicadores de desempenho (confiabilidade, disponibilidade, custo, MTTR, MTBF) e avaliar o uso do testador com os usuários finais com a aplicação de um teste de usabilidade. No que diz respeito às funcionalidades desenvolvidas, destaca-se a geração de relatório em PDF para controle de histórico de falhas. Em relação aos indicadores de desempenho, observou-se significativa melhora na confiabilidade, custo, MTTR e MTBF. A disponibilidade devido a exclusivamente a falhas de *joystick* aumentou, porém diminuiu de um modo geral devido a outras falhas. Os testes de usabilidade com os usuários finais foram realizados através de formulário digital, em que o software e funções como a geração de relatórios em PDF foram avaliadas, além da coleta de comentários e sugestões de melhorias, obtendo-se excelentes resultados e *feedbacks* interessantes para o desenvolvimento de trabalhos futuros.

Palavras-chave: *Joystick. Reach Stacker. KPI. Testador. Teste de Usabilidade.*

ABSTRACT

One of the most used equipment in port environments is the reach stacker, responsible for moving containers over short distances. It is divided into a few main parts such as boom, spreader and operator's cab. Inside the operator's cabin, there is the joystick, an electronic component, responsible for carrying out all the functions of moving the boom (stretching, retracting, raising and lowering), the spreader (opening, closing, moving, rotating, tilting and probing), in addition to radio communication function and horn. When the joystick is defective, it goes through a series of steps to find the root cause, from removing it from the equipment and installing it in another one to confirm the problem, to send it for external repair. This procedure is not only time-consuming but also very costly. Thus, it is extremely important to develop a joystick tester to quickly detect failures and allow the component to be repaired assertively. This work aims to improve software and hardware initially developed, analyze the impact of the implementation in the maintenance sector through the leading performance indicators (KPI) (reliability, availability, cost, MTTR, MTBF) and evaluate the use of the tester with the end users with the application of a usability test. Concerning the features developed, the generation of a PDF report to control the failure history stands out. Regarding the performance indicators, there was a significant improvement in reliability, cost, MTTR and MTBF. Availability due to joystick failures alone has increased but generally decreased due to other failures. Usability tests with end users were carried out using a digital form, in which the software and functions such as generating PDF reports were evaluated, in addition to collecting comments and suggestions for improvements, obtaining excellent results and interesting feedback for the development of future work.

Keywords: *Joystick. Reach Stacker. KPI. Tester. Usability Testing.*

LISTA DE FIGURAS

Figura 1 – Fluxograma de testes para identificar falha do <i>joystick</i>	2
Figura 2 – Visão geral da RSK.	5
Figura 3 – Movimentos principais da RSK.	6
Figura 4 – A lança da RSK e alguns componentes.	7
Figura 5 – Movimentação da lança da RSK.	7
Figura 6 – Mecanismo <i>twistlock</i> no <i>spreader</i> para acoplamento de carga.	8
Figura 7 – Pinos <i>twistlocks</i> para acoplamento nas bolsas do contâiner.	8
Figura 8 – <i>Spreader</i> da RSK fechado e aberto.	9
Figura 9 – Deslocamento lateral do <i>spreader</i>	9
Figura 10 – Inclinação do <i>spreader</i>	10
Figura 11 – Giro do <i>spreader</i>	10
Figura 12 – Modelo do <i>joystick</i> da RSK.	10
Figura 13 – Funções do <i>joystick</i>	11
Figura 14 – <i>Joystick</i> internamente.	12
Figura 15 – <i>Windows Tester</i>	16
Figura 16 – <i>Joystick</i> da RSK fora e dentro da máquina.	18
Figura 17 – Conexão para codificação de sinais do <i>joystick</i>	19
Figura 18 – Testes com <i>NI Max</i>	20
Figura 19 – Painel de testes.	20
Figura 20 – <i>Layout</i> inicial do testador.	21
Figura 21 – Tabela com resultado dos testes.	22
Figura 22 – Erro apresentado caso não seja criado o caminho da tabela de testes.	22
Figura 23 – Processo de desenvolvimento do <i>hardware</i>	24
Figura 24 – Relação entre benefício e número de participantes.	29
Figura 25 – Novo <i>layout</i> do testador.	31
Figura 26 – Organização do código para criação do relatório em PDF.	33
Figura 27 – <i>Hardware</i> do testador após modificações.	34
Figura 28 – Projeto do testador na plataforma <i>GitHub</i>	35
Figura 29 – Fluxograma de testes para identificar falha do <i>joystick</i> após implementação do testador.	37
Figura 30 – Número absoluto de falhas por período.	40
Figura 31 – Tela inicial do aplicativo.	47
Figura 32 – Tela do aplicativo após iniciar a simulação.	47
Figura 33 – Botão de controle de simulação.	48
Figura 34 – Caminho para salvar o relatório em PDF.	48
Figura 35 – Seta indicativa de referência.	49

Figura 36 – Gráfico da função de subir e descer lança.	51
Figura 37 – Gráfico da função de estender e recolher lança.	51
Figura 38 – Gráfico da função de girar <i>spreader</i>	51

LISTA DE QUADROS

Quadro 1 – Divisão das etapas de desenvolvimento do testador de <i>joystick</i>	17
Quadro 2 – Características dos participantes.	29
Quadro 3 – Perguntas do questionário de teste de usabilidade.	30

LISTA DE TABELAS

Tabela 1 – Conexão elétrica DB25 por função.	19
Tabela 2 – Comparativo entre as métricas de código das versões.	38
Tabela 3 – Impacto das paradas por <i>joystick</i> na confiabilidade das RSKs.	39
Tabela 4 – Impacto das paradas por <i>joystick</i> na disponibilidade das RSKs.	39
Tabela 5 – Custo médio de parada por <i>joystick</i>	39
Tabela 6 – MTBF e MTTR por falhas de <i>joystick</i>	40
Tabela 7 – Média das avaliações referente às funcionalidades desenvolvidas.	41
Tabela 8 – Junção das avaliações referentes à geração de relatório em PDF.	41
Tabela 9 – Valores ideais das funções testadas.	48
Tabela 10 – Instrução para conexão do conector DB25 e placa NI 6001.	49
Tabela 11 – Resultado dos potenciômetros testados.	50

LISTA DE ABREVIATURAS E SIGLAS

BBC	<i>Bomb Cart</i>
BD	<i>Breakdown</i>
CMMS	<i>Computerized Maintenance Management System</i>
CSV	<i>Comma-Separated Values</i>
DAQ	<i>Data Acquisition</i>
IDE	<i>Integrated Development Environment</i>
KPI	<i>Key Performance Indicator</i>
LED	<i>Light Emitting Diode</i>
MHC	<i>Mobile Harbor Crane</i>
MTBF	<i>Mean Time Between Failures</i>
MTTR	<i>Mean Time To Repair</i>
PCM	Planejamento e Controle de Manutenção
PDF	<i>Portable Document Format</i>
POO	Programação Orientada a Objetos
OS	Ordem de Serviço
RSK	<i>Reach Stacker</i>
STS	<i>Ship To Shore</i>
SIGM	Sistema Informatizado para Gestão da Manutenção
TCC	Trabalho de Conclusão de Curso
TTR	<i>Terminal Tractor</i>
TEU	<i>Twenty-foot Equivalent Unit</i>
USB	<i>Universal Serial Bus</i>
VS	<i>Visual Studio</i>
XML	<i>eXtensible Markup Language</i>

LISTA DE SÍMBOLOS

C	Confiabilidade
D	Disponibilidade
n_f	Número absoluto de falhas
t_{BD}	Tempo total de <i>breakdown</i>
t_m	Tempo total em manutenção
t_t	Tempo total de funcionamento
v	Volume abastecido

SUMÁRIO

1	INTRODUÇÃO	1
1.1	JUSTIFICATIVA	1
1.2	OBJETIVOS	3
1.2.1	Objetivo geral	3
1.2.2	Objetivos específicos	3
1.3	ORGANIZAÇÃO DO TRABALHO	4
2	REVISÃO DA LITERATURA	5
2.1	EMPILHADEIRA TIPO <i>REACH STACKER</i>	5
2.1.1	Lança	6
2.1.2	<i>Spreader</i>	8
2.1.3	<i>Joystick</i>	10
2.2	INDICADORES DE DESEMPENHO	12
2.2.1	Confiabilidade	12
2.2.2	Disponibilidade	13
2.2.3	Custo de manutenção	13
2.2.4	MTBF	14
2.2.5	MTTR	14
2.3	CONCEITOS COMPUTACIONAIS DE DESENVOLVIMENTO DE <i>SOFTWARE</i>	14
2.3.1	<i>Integrated Development Environment</i>	14
2.3.2	Biblioteca	15
2.3.3	Repositório de <i>software</i>	15
2.3.4	Placa de aquisição de dados	15
2.4	TESTE DE USABILIDADE	15
2.5	TRABALHOS RELACIONADOS	16
2.5.1	Controladores de jogo	16
2.5.2	Aplicação de <i>software</i> com teste de usabilidade	17
2.6	VERSÃO INICIAL DO TESTADOR DE <i>JOYSTICK</i>	17
2.6.1	Estudo do <i>joystick</i>	18
2.6.2	Criação do <i>layout</i>	21
2.6.3	Armazenamento de dados em tabela	21
2.6.4	<i>Hardware</i> inicial	23
3	MATERIAIS E MÉTODOS	25
3.1	CARACTERIZAÇÃO DA PESQUISA	25
3.1.1	Abordagem	25

3.1.2	Natureza	25
3.1.3	Objetivos	25
3.1.4	Procedimentos técnicos	25
3.2	SOFTWARES DE DESENVOLVIMENTO E PLATAFORMAS DIGITAIS DE APOIO	26
3.2.1	<i>Visual Studio Community</i>	26
3.2.2	<i>GitHub</i>	26
3.2.3	<i>NI Max</i>	26
3.3	HARDWARE	26
3.4	AQUISIÇÃO E ANÁLISE DE DADOS	27
3.5	INSTRUÇÃO E TREINAMENTO DA EQUIPE	27
3.6	TESTE DE USABILIDADE	28
3.6.1	Desenvolvimento do plano de teste	28
3.6.1.1	Objetivos do teste	28
3.6.1.2	Dados coletados	28
3.6.2	Seleção e recrutamento dos participantes	28
3.6.3	Preparação do material e ambiente de teste	29
3.6.4	Condução do teste	29
3.6.5	Questionário pós teste	30
3.6.6	Procedimento de análise dos dados	30
4	DESENVOLVIMENTO	31
4.1	DESENVOLVIMENTO DO TESTADOR	31
4.1.1	Alteração do layout	31
4.1.2	Controle de usuário	32
4.1.3	Controle de simulação	32
4.1.4	Geração de relatório PDF	32
4.1.5	Melhoria no <i>hardware</i>	33
4.1.6	Repositório no <i>GitHub</i>	34
4.2	DESENVOLVIMENTO DO MANUAL DE INSTRUÇÕES	35
4.3	LEVANTAMENTO DE DADOS	36
4.3.1	Antes da implementação do <i>software</i>	36
4.3.2	Depois da implementação do <i>software</i>	36
5	ANÁLISE E DISCUSSÃO DOS RESULTADOS	37
5.1	SOFTWARE TESTADOR DE <i>JOYSTICK</i>	37
5.1.1	Métricas de código da IDE	38
5.2	INDICADORES DE DESEMPENHO	38
5.2.1	Confiabilidade	38
5.2.2	Disponibilidade	39

5.2.3	Custo	39
5.2.4	MTBF e MTTR	40
5.3	TESTE DE USABILIDADE DE USUÁRIO	40
5.3.1	Funcionalidades desenvolvidas	40
5.3.2	Geração de relatório em PDF	41
5.3.3	Feedback geral	41
6	CONCLUSÃO	42
6.1	TRABALHOS FUTUROS	43
6.2	CONSIDERAÇÕES FINAIS	43
	Referências	44
	APÊNDICE A INSTRUÇÃO DE USO DO TESTADOR DE JOYSTICK	47
A.1	INFORMAÇÕES	47
A.2	PASSOS PARA TESTE	48
A.3	RELATÓRIO DE TESTES	48
A.4	VALORES IDEAIS	48
A.5	CONEXÃO ELÉTRICA DO TESTADOR	49
	APÊNDICE B RELATÓRIO DE TESTE AUTOMATIZADO	50
B.1	INFORMAÇÕES GERAIS	50
B.2	RESULTADO DO TESTE	50
B.2.1	Botões digitais	50
B.2.2	Potenciômetros	50
B.3	OBSERVAÇÕES DO TESTE	50
B.4	GRÁFICOS DOS POTENCIÔMETROS	51
	ANEXO A DATASHEET DO JOYSTICK	52

1 INTRODUÇÃO

O porto organizado é um bem público construído e aparelhado para atender necessidades de navegação, movimentação de passageiros ou movimentação e armazenagem de mercadorias, em que o tráfego e as operações portuárias estejam sob a jurisdição de autoridade portuária (BRASIL, 2013). Nesse contexto, no município de Itajaí, Santa Catarina, encontra-se o Porto de Itajaí, um dos portos mais operantes do Brasil, com a gestão municipalizada e a parte operacional sob responsabilidade da empresa arrendatária *APM Terminals*, sendo encarregada de 100% da movimentação de contêineres (OLIVEIRA, 2011).

Para realizar a movimentação das cargas no Porto de Itajaí, existem diversos equipamentos portuários, como os guindastes para retirada de carga dos navios (portêineres), *Ship To Shore* (STS) e *Mobile Harbor Crane* (MHC), a máquina *Reach Stacker* (RSK) para carregamento e descarregamento dos contêineres, o *Terminal Tractor* (TTR) e *Bomb Cart* (BBC) para movimentação das cargas, além de outros equipamentos e empilhadeiras menores para auxílio à rotina operacional (TERMINALS, 2022).

A RSK é um veículo tipo empilhadeira de grande porte utilizado no empilhamento e movimentação de contêineres por curta distância em terminais portuários, logísticos, retroportuários (áreas adjacentes ao terminal portuário) e até em aplicações militares (SKAF S. LAMROUS; MANIER, 2019; KALMAR, 2022).

Vários movimentos da máquina RSK são realizados pelo operador através do dispositivo eletrônico *joystick* que, ao apresentar falha, pode interromper o funcionamento da máquina, gerando impacto na operação e nos indicadores de desempenho, além do impacto financeiro para reparo externo do componente (THIAGO, 2013).

Os indicadores de manutenção adotados no cenário portuário (ou *Key Performance Indicator* – KPI) medem a performance das máquinas de forma quantitativa, com base nas horas rodadas do equipamento e horas indisponíveis, sejam essas por falhas ou por paradas para realização de manutenção preventiva e preditiva (manutenções programadas) (THIAGO, 2013). Já o custo operacional da máquina abrange os valores destinados à manutenção programada, às peças e aos reparos realizados devido às quebras não planejadas a fim de obter o custo médio de manutenção e operação por equipamento.

1.1 JUSTIFICATIVA

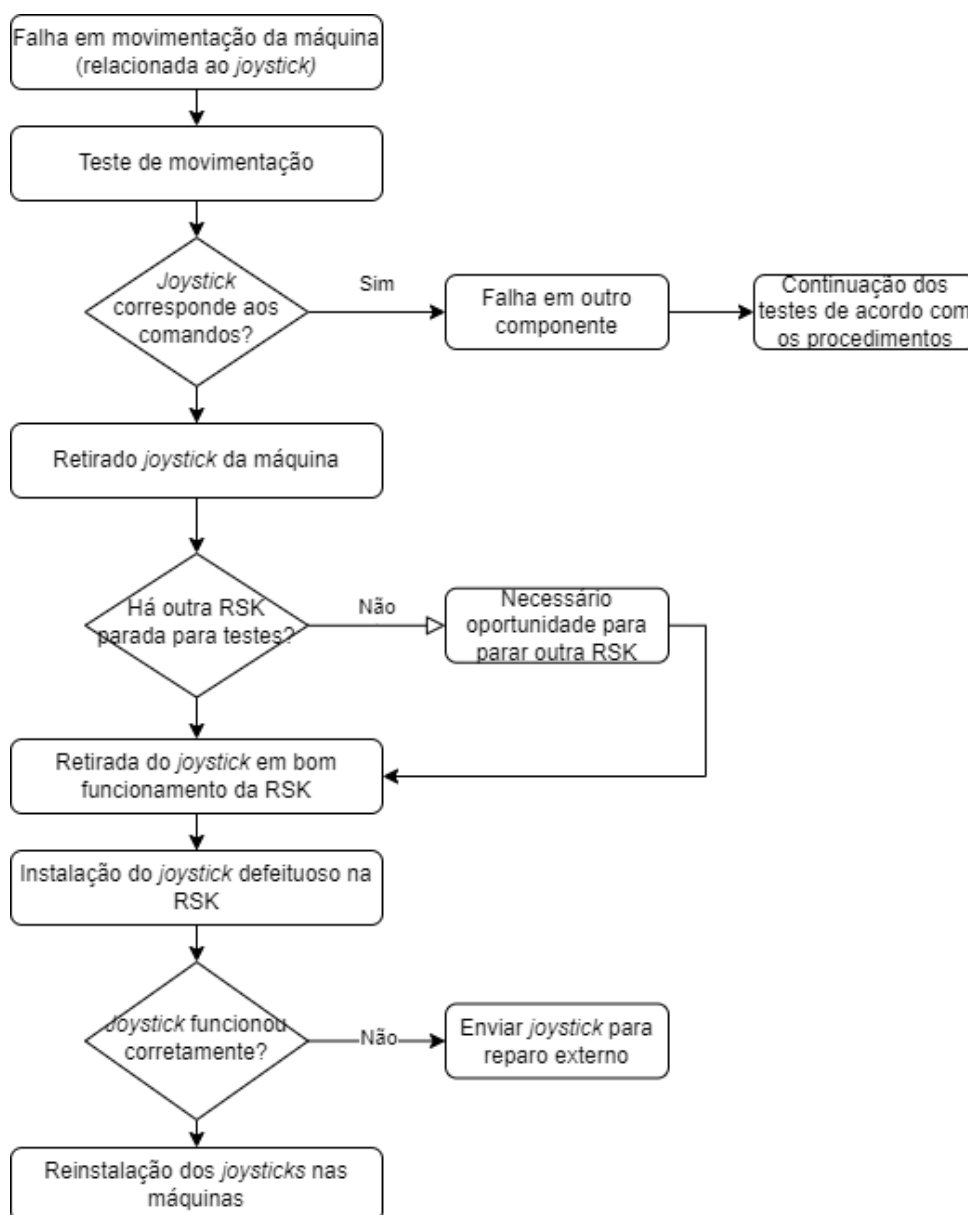
Quando há falha especificamente do *joystick* na RSK, o procedimento a ser realizado é tirá-lo da máquina em que se encontra instalado, e então é inserido em outra RSK que esteja parada (para manutenção preventiva, por exemplo) para confirmar a falha do componente. Porém, nem sempre o equipamento pode ser ligado para ser testado (por questões impeditivas ou de bloqueio de segurança durante a realização de manutenção), gerando mais atrasos para verificação

do problema. Se não há outra máquina parada para realizar o teste do *joystick*, o equipamento originalmente com a falha fica fora de operação até haver disponibilidade de outro equipamento.

Esse procedimento pode demorar algumas horas, dependendo do cenário de operação portuária. Essa verificação é apenas a fim de confirmar se a falha persiste em outro equipamento, para certificar que é uma falha inerente ao *joystick* e não à máquina. Após realizar todo o processo de retirada do *joystick*, instalação e testagem em outro equipamento e verificado que de fato existe uma falha, o componente é então levado para reparo externo, por falta de uma correta identificação do problema raiz na própria empresa.

A fim de melhor demonstrar e ilustrar o procedimento realizado em caso de suspeita ou falha do *joystick*, é apresentado o fluxograma do processo na Figura 1.

Figura 1 – Fluxograma de testes para identificar falha do *joystick*.



Assim, é de extrema relevância o desenvolvimento de um *software* testador dos parâmetros do *joystick* juntamente com o *hardware*, pois não há um *software* comercial capaz de realizar esse tipo de teste. O *software* desenvolvido deve ser capaz de identificar se existe resposta do sinal aplicado nos componentes analógicos e digitais, e verificar se essa resposta é compatível com o movimento esperado na máquina.

Anteriormente, na disciplina de Estágio Obrigatório, foi desenvolvido um protótipo do *software*, com funções básicas de teste dos parâmetros do *joystick*. A partir dessa versão, foram identificadas oportunidades de melhoria.

Com o aprimoramento e implementação desse *software* no ambiente empresarial, será possível identificar corretamente a falha e realizar o reparo na própria empresa, quando possível, sem a necessidade de levá-lo para reparo externo. Com isso, pretende-se diminuir o tempo destinado aos testes (quando comparado ao tempo de teste na própria máquina), o custo operacional da máquina e o custo relacionado ao conserto.

Essa ferramenta também abre espaço para a criação de procedimentos internos de reparo, ampliação do laboratório de elétrica já existente na empresa e a valorização e capacitação do quadro técnico especializado.

1.2 OBJETIVOS

1.2.1 Objetivo geral

O objetivo geral do presente trabalho é aprimorar o *software* testador de parâmetros do dispositivo *joystick* de máquinas tipo *reach stacker* e avaliar o impacto de sua implementação no setor de manutenção portuária.

1.2.2 Objetivos específicos

- Aprimorar o protótipo de *hardware* para acoplar o *joystick*;
- Identificar e implementar novas funções no *software* testador de parâmetros, como função de registro de usuário, data de teste e função de geração de relatório;
- Implementar o *software* testador em bancada/laboratório apropriado para uso e teste;
- Desenvolver procedimento e instrução de uso (manual) para capacitação interna;
- Levantar os custos relacionados ao reparo do *joystick*;
- Definir e analisar os indicadores de manutenção antes e depois da implementação do *software*;
- Aplicar teste de usabilidade com os usuários do testador para avaliar a facilidade e relevância do uso.

1.3 ORGANIZAÇÃO DO TRABALHO

O presente trabalho está dividido em 6 capítulos na sua totalidade. O capítulo 2 consta com a revisão de literatura, abordando princípios importantes e fundamentais para a compreensão do trabalho, como a definição e explicação da máquina RSK, seus movimentos e o dispositivo eletrônico *joystick*. Traz também o conceito de indicadores de desempenho e os principais utilizados, além da contextualização dos *softwares* e plataformas de apoio durante o trabalho, definição de teste de usabilidade e referência aos trabalhos relacionados.

No capítulo 3 constam os materiais e métodos utilizados no trabalho, que tratam sobre a descrição concisa do que foi utilizado para o desenvolvimento do trabalho.

O capítulo 4 traz todas as etapas de desenvolvimento do testador de parâmetros do *joystick*, as alterações sofridas em relação à versão anterior do programa, o desenvolvimento do manual de instruções e procedimento interno de trabalho, além do levantamento de dados de indicadores de manutenção e o teste de usabilidade aplicado aos usuários.

Já o capítulo 5 aborda a análise e discussão dos resultados antes e após a implementação do *software* testador de *joystick* no local de pesquisa, um setor de manutenção portuária em Itajaí.

Por fim, o último capítulo traz as principais conclusões do trabalho, propostas de trabalhos futuros e considerações finais.

2 REVISÃO DA LITERATURA

O presente capítulo tem como objetivo introduzir e referenciar conceitos importantes para a compreensão deste trabalho. Inicialmente, será apresentada a máquina *Reach Stacker* para melhor entendimento de suas principais partes e funções e sua relação com o dispositivo eletrônico *joystick*.

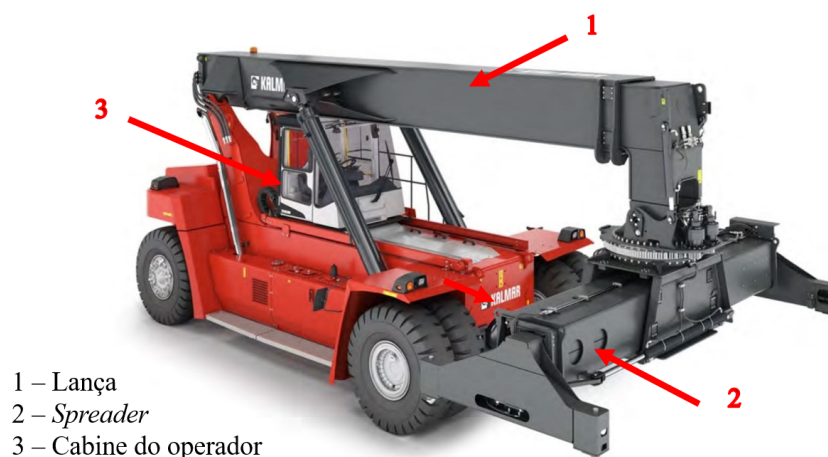
Na sequência, são apresentadas definições de alguns indicadores de desempenho e conceitos computacionais relacionados ao desenvolvimento de *software* e à aquisição de dados. Também é abordada a definição e aplicação dos testes de usabilidade de usuário e, por último, alguns trabalhos relacionados.

2.1 EMPILHADEIRA TIPO *REACH STACKER*

A máquina tipo *Reach Stacker* (RSK) é uma empilhadeira de grande porte utilizada em terminais portuários e logísticos para a movimentação e transporte de contêineres no pátio. De acordo com seu porte e capacidade de carga, consegue acoplar contêineres de diferentes tamanhos, como de 20ft e 40ft¹, e em diversas disposições de altura nas pilhas de contêineres. Os modelos DRF-450 e DRU-450 da marca *Kalmar* alcançam até 15,1 metros de altura, o suficiente para empilhar cinco contêineres (KALMAR, 2014).

A máquina é dividida em conjuntos, de acordo com sua funcionalidade, e os de principal interesse nesse trabalho são a lança, *spreader* e cabine do operador (onde se localiza o *joystick*), conforme Figura 2.

Figura 2 – Visão geral da RSK.



Fonte: Adaptado de Kalmar (2014).

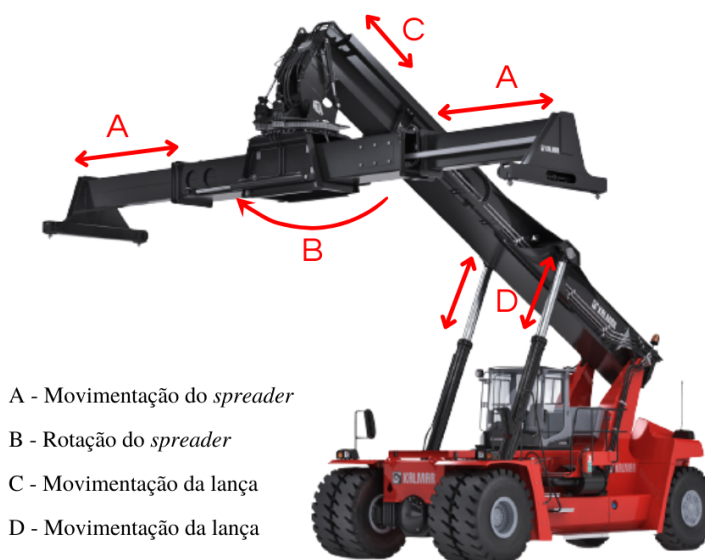
A parte responsável pelo acoplamento do contêiner é chamada de *spreader*, que contempla as asas do *spreader* (lado direito e esquerdo), capaz de deslocar e expandir/recolher,

¹A unidade padrão para medição de contêineres é em pés.

para pegar diferentes tamanhos de contêineres (20ft e 40ft). Já o *boom* (lança), faz a função de expandir e recolher (movimentação no eixo longitudinal da lança) e levantar e abaixar, a fim de conseguir levantar o contêiner.

A cabine do operador é onde se localiza o *joystick*, responsável por realizar algumas funções da máquina, como os movimentos principais da Figura 3.

Figura 3 – Movimentos principais da RSK.



Fonte: Adaptado de Iljukkjin (2015).

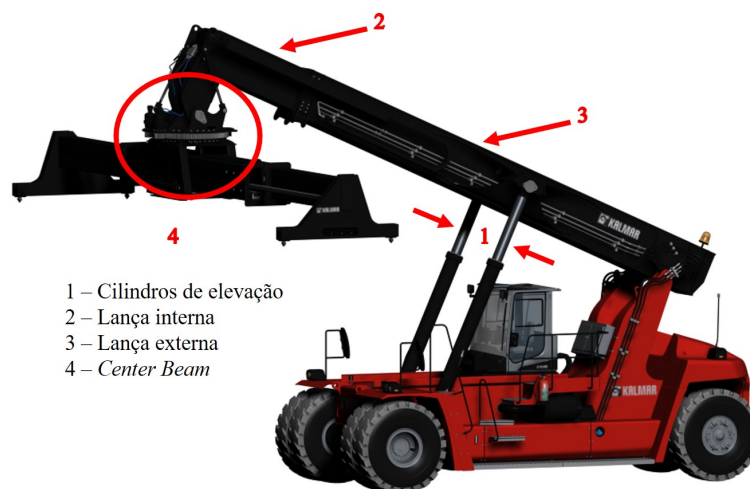
2.1.1 Lança

A lança, ou *boom*, é a parte mecânica da máquina composta por cilindros de elevação e extensão, e sua extremidade é conectada ao *spreader* através do *center beam*, ponto central do *spreader* em que se encontram alguns motores hidráulicos para realizar funções do *spreader*, conforme Figura 4 (KALMAR, 2014).

Através dos cilindros de elevação (lado esquerdo e lado direito) é possível rotacionar a lança em torno do seu ponto de fixação localizado atrás da cabine do operador, elevando-a ou abaixando-a. Além disso, o cilindro de extensão localizado dentro da lança interna permite avançá-la ou recuá-la no sentido longitudinal, a fim de realizar os movimentos necessários para acoplamento e movimentação dos contêineres.

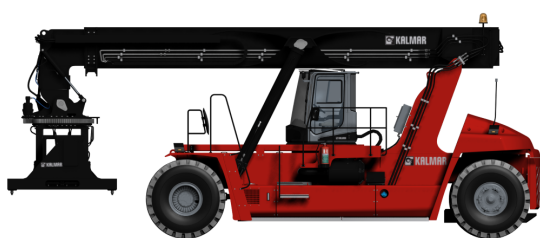
Todas essas funções (Figura 5) são comandadas através da resposta do sinal emitido pelo dispositivo *joystick*, manuseado pelo operador dentro da cabine.

Figura 4 – A lança da RSK e alguns componentes.



Fonte: Autora.

Figura 5 – Movimentação da lança da RSK.



(a) Lança totalmente recolhida.



(b) Lança totalmente estendida.



(c) Lança totalmente erguida.



(d) Lança totalmente erguida e estendida.

Fonte: Autora.

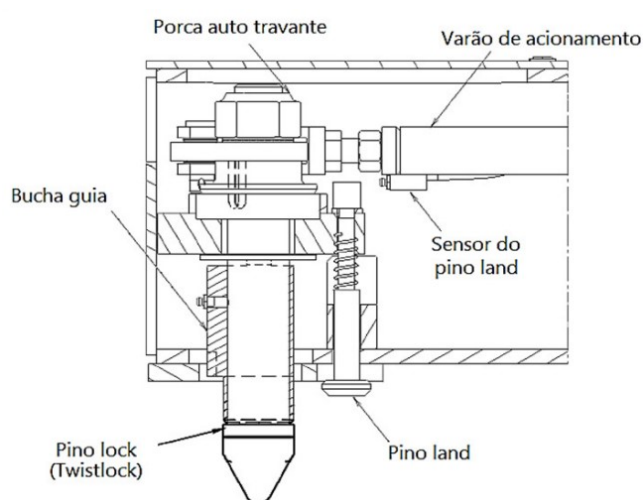
2.1.2 Spreader

O *spreader* é a parte responsável por acoplar e carregar o contêiner e carga unitizada². Na versão do *spreader* para contêineres encontram-se componentes mecânicos e sensores eletrônicos que compõem o mecanismo de travamento, denominado *twistlock*, ou pino *lock* (Figura 6).

Esse mecanismo, em formato de pino, está localizado nas quatro extremidades do *spreader* e é acoplado nas quatro bolsas (cantos) do contêiner para então sinalizar o travamento, conforme Figura 7 (ITAJAÍ, 2022a).

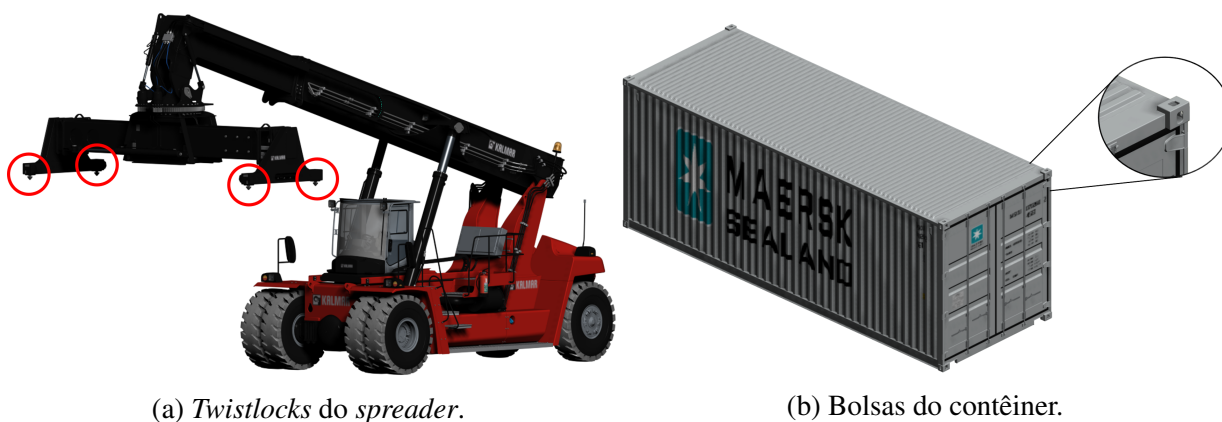
O *spreader* é um elemento presente tanto na RSK quanto em portêineres (guindastes portuários, como STS e MHC).

Figura 6 – Mecanismo *twistlock* no *spreader* para acoplamento de carga.



Fonte: Kalmar (2014).

Figura 7 – Pinos *twistlocks* para acoplamento nas bolsas do contêiner.



(a) *Twistlocks* do *spreader*.

(b) Bolsas do contêiner.

Fonte: Autora.

²Carga unitizada é um conceito logístico baseado em agregar volumes fracionados em uma única unidade de carga, podendo ser através de palete, lingas, contêineres, etc. Um exemplo de carga unitizada possível de ser carregada pelo *spreader* são pequenas embarcações com o condicionamento da carga com lingas de aço (KEEDI, 2003).

O conjunto do *spreader* possui algumas funções realizadas pelo *joystick* para adaptar a cada contêiner, como:

- Abertura e fechamento, para ajuste de acordo com o tamanho do contêiner (Figura 8);
- Deslocamento lateral, para direita e esquerda³, principalmente em casos de contêiner mal ovado⁴ para ajustar o centro de massa (Figura 9);
- Inclinação, a fim de balancear o peso e encontrar o centro de massa do contêiner e, principalmente, para manuseio de cargas carregadas com grãos, cavacos de madeiras e outros materiais à granel (Figura 10);
- Giro do *spreader*, para operar a máquina em locais de difícil acesso (Figura 11).

Figura 8 – *Spreader* da RSK fechado e aberto.



(a) *Spreader* fechado para carga de 20'.



(b) *Spreader* aberto para carga de 40'.

Fonte: Autora.

Figura 9 – Deslocamento lateral do *spreader*.



(a) *Spreader* deslocado para a direita.



(b) *Spreader* deslocado para a esquerda.

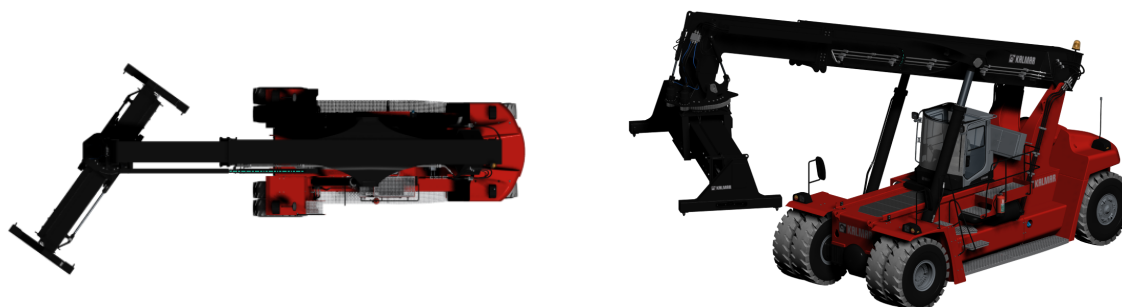
Fonte: Autora.

³A referência de esquerda e direita é com base na visão do operador dentro da cabine.

⁴Estufagem, ou ova, de contêiner é o procedimento de carregar mercadorias e preencher um contêiner.

Figura 10 – Inclinação do *spreader*.

Fonte: Autora.

Figura 11 – Giro do *spreader*.(a) Giro do *spreader* com visão superior.(b) Giro do *spreader* com visão lateral.

Fonte: Autora.

2.1.3 Joystick

O joystick (Figura 12) é um dispositivo eletrônico localizado no interior da cabine do operador da RSK e responsável por inúmeras funções da máquina.

Figura 12 – Modelo do joystick da RSK.

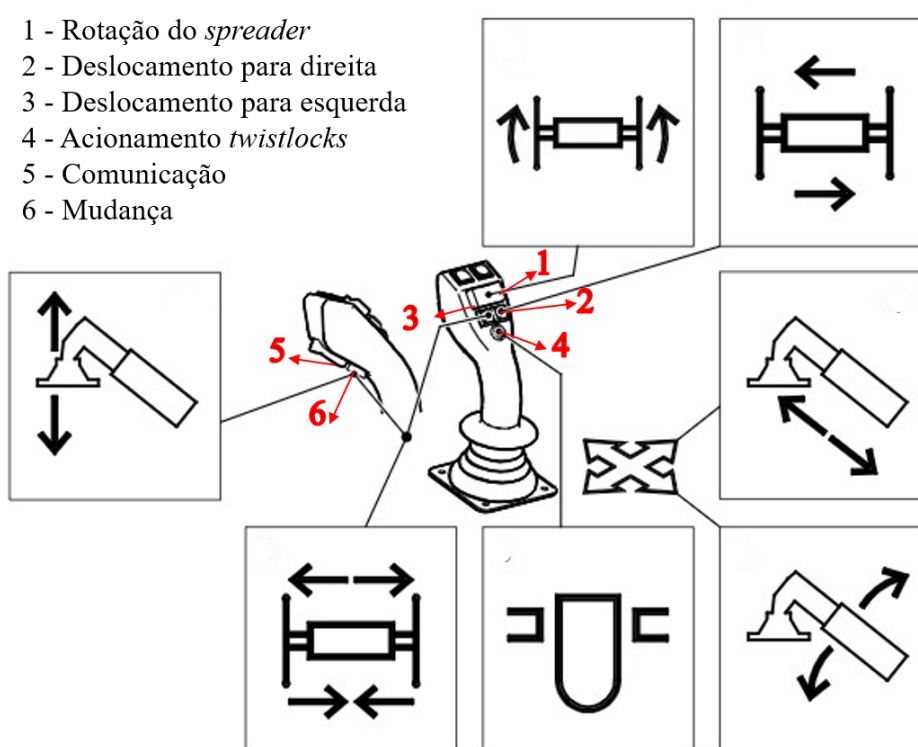


Fonte: Kalmar (2014).

Internamente, o *joystick* é composto por quatro componentes analógicos (potenciômetros), cinco botões digitais, dois *Light Emitting Diode* (LEDs) para sinalização e suas conexões elétricas. A tensão de alimentação do circuito é 24 volts (V) e a tensão dos sinais analógicos é de 5 V (Anexo A).

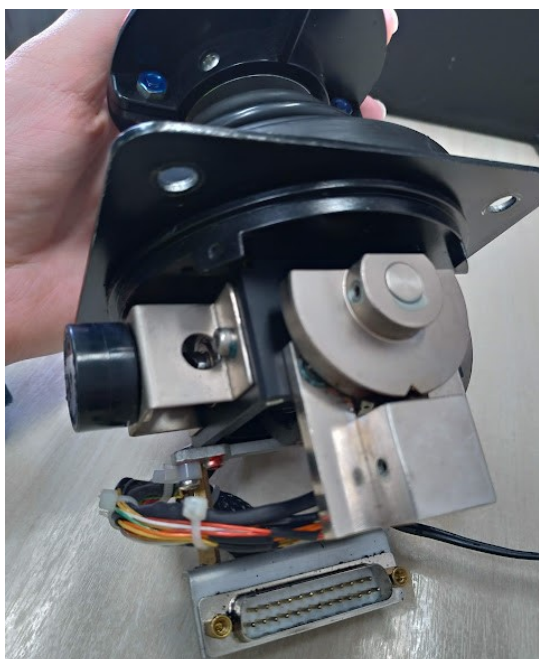
Os componentes eletrônicos responsáveis pelas funções de rotação do *spreader*, extensão, recolhimento, elevação e subida da lança são analógicos, os potenciômetros. Na Figura 13 eles não estão visíveis, porém um deles está localizado na parte superior do *joystick* e é acionado através do manuseio do botão 3. Há outros dois potenciômetros, localizados na coifa do *joystick* (Figura 14) e acionados através da movimentação nos eixos (indicado pelas setas na Figura 13). O quarto potenciômetro também está na coifa, responsável pela função de *tilt* que, por ser uma função opcional, encontra-se desativada para essa aplicação e com o circuito reconfigurado para realizar a função de comunicação via rádio, acionado pelo botão 5 da Figura 13.

Figura 13 – Funções do *joystick*.



Fonte: Adaptado de Kalmar (2014).

Já os demais botões são digitais. Os botões 2 e 3 realizam a função de deslocamento do *spreader* para a direita e para a esquerda, respectivamente. Esses botões, quando acionados juntamente com o botão 6 (botão de mudança ou, popularmente chamado, botão de homem morto) realizam a função de abrir e fechar o *spreader*. O botão 4 realiza o acionamento dos *twistlocks*, acoplando e desacoplando o *spreader* na carga.

Figura 14 – *Joystick internamente.*

Fonte: Autora.

2.2 INDICADORES DE DESEMPENHO

Os indicadores de desempenho, ou indicadores de performance (*Key Performance Indicator* - KPI), são uma forma de representação quantitativa dos dados oriundos de execução de atividades organizacionais, de acordo com a estratégia empresarial e, usualmente, transformados em representações gráficas.

Dentro do contexto de manutenção portuária, os indicadores também são classificados de acordo com sua criticidade, que são diretamente ligados com a estratégia de manutenção e com os requisitos de entrega do departamento de manutenção, sendo estes indicadores de confiabilidade, disponibilidade, *Mean Time Between Failures* (MTBF), ou seja, tempo médio entre falhas e *Mean Time To Repair* (MTTR), tempo médio para reparo (BURGER, 2017).

Para os equipamentos movidos a óleo diesel, há o indicador de consumo de combustível, como é o caso da máquina RSK. Outro indicador também considerado é o custo de manutenção, que é calculado para todos os equipamentos.

2.2.1 Confiabilidade

A confiabilidade indica o percentual de tempo em que o equipamento esteve funcionando em relação ao tempo operacional, que é composto pelo total de horas rodadas do equipamento e as horas devido a paradas para manutenção emergencial, os *breakdowns* (BD).

As paradas do equipamento causadas por manutenção preventiva, avarias, acidentes ou outros motivos operacionais não são consideradas neste indicador, apenas falhas de máquina.

A Equação 1 apresenta o cálculo da confiabilidade e seu resultado é dado em porcentagem (BURGER, 2017).

$$C = \frac{t_t - t_{BD}}{t_t} 100 \quad (1)$$

Onde:

C = Confiabilidade, em percentual;

t_t = Tempo total de funcionamento, em horas;

t_{BD} = Tempo total de BD, em horas.

2.2.2 Disponibilidade

O indicador de disponibilidade indica o percentual de tempo em que o equipamento esteve disponível, em relação ao tempo total do período em questão. A Equação 2 apresenta o cálculo de disponibilidade, também dada em porcentagem (BURGER, 2017).

$$D = \frac{t_t - t_m}{t_t} 100 \quad (2)$$

Onde:

D = Disponibilidade, em percentual;

t_t = Tempo total de funcionamento, em horas;

t_m = Tempo total em manutenção, em horas.

Dentro de “Tempo total em manutenção” devem ser consideradas as horas de equipamento parado por motivo de manutenção preventiva, manutenção corretiva e manutenção emergencial (*breakdown*).

Nesse indicador, as paradas de equipamento causadas por avarias, acidentes ou outros motivos operacionais não devem ser consideradas.

2.2.3 Custo de manutenção

O custo de manutenção indica os custos relacionados com a manutenção de equipamentos, incluindo as peças, materiais e consumíveis (óleos, graxas, mantas absorventes) utilizados em atividades de manutenção, assim como custos com contratação de serviços externos necessários para tais atividades. Este indicador não inclui os custos com mão de obra da equipe de manutenção (folha de pagamento), nem custos com combustíveis utilizados nos equipamentos (consumo de combustíveis) ou energia elétrica utilizada (para o caso do STS) (BURGER, 2017).

O custo de manutenção pode ser apresentado tanto em valores absolutos, como em relação à quantidade de *moves* (medida de movimentação portuária com base na movimentação de um contêiner de 20ft, *Twenty-foot Equivalent Unit* - TEU) realizados no período (ITAJÁI, 2022b).

2.2.4 MTBF

O tempo médio entre as falhas é a relação entre o tempo total de horas trabalhadas de um equipamento ou unidade pelo número de intervenções de manutenção corretivas, em um dado período observado.

Quando não houver falhas no período, o MTBF será igual ao número de horas trabalhadas no período. A forma de cálculo desse indicador pode ser vista na Equação 3 (BURGER, 2017).

$$MTBF = \frac{t_t}{n_f} \quad (3)$$

Onde:

MTBF = Tempo médio entre falhas, em horas;

t_t = Tempo total de funcionamento, em horas;

n_f = Número absoluto de falhas.

2.2.5 MTTR

O tempo médio para reparo é um indicador que analisa o tempo total de manutenção após a ocorrência de uma falha. No cálculo é levado em consideração todo o tempo dedicado à manutenção, desde a concepção do problema, diagnóstico e conserto, até os testes finais para o equipamento se tornar apto à operação novamente.

Assim, é possível medir a capacidade de manutenção de uma equipe em relação ao número de ocorrências de falhas, conforme Equação 4 (BURGER, 2017):

$$MTTR = \frac{t_m}{n_f} \quad (4)$$

Onde:

MTTR = Tempo médio entre falhas, em horas;

t_m = Tempo total em manutenção, em horas;

n_f = Número absoluto de falhas.

2.3 CONCEITOS COMPUTACIONAIS DE DESENVOLVIMENTO DE SOFTWARE

2.3.1 *Integrated Development Environment*

De acordo com Sebesta (2003), *Integrated Development Environment* (IDE) pode ser identificado como um ambiente de desenvolvimento integrado (em tradução literal) que reúne características e ferramentas que dão apoio ao desenvolvimento de *software*, com o objetivo de agilizar o processo.

Algumas características e ferramentas encontradas em um IDE são: a existência de um editor para código fonte, um compilador (*compiler*) para o programa e um depurador (*debugger*) para auxílio no processo de encontrar e corrigir erros no código fonte (CASAVELLA, 2012).

2.3.2 Biblioteca

O conceito de biblioteca para a computação é um arquivo com coleção de subprogramas utilizados no desenvolvimento de *software*, com o intuito de facilitar a programação. Nas bibliotecas encontram-se códigos e dados auxiliares, que organizam o código predefinido para utilizar nas aplicações (WERNER et al., 2007).

2.3.3 Repositório de *software*

Repositórios de *softwares* são responsáveis por armazenar todos os diretórios e arquivos de um projeto, sendo capazes de realizar o controle de versão durante o desenvolvimento do projeto (ROMANI, 2009). Através do repositório, os desenvolvedores podem realizar modificações no código e manter um registro detalhado do progresso, possibilitando a recuperação de algum arquivo, caso seja necessário.

No repositório do projeto, constam informações sobre o trabalho desenvolvido, da linguagem de programação utilizada e as devidas pastas com os códigos. Para o desenvolvimento dos códigos, é possível criar *branches* no projeto, que seriam as ramificações, para que uma equipe consiga trabalhar junto ou então desenvolver partes separadas de um programa. Quando o *branch* é confirmado e validado, pode-se utilizar a função *commit* e dar um *merge* com o *branch* principal, ou seja, mesclar as alterações com o código principal.

2.3.4 Placa de aquisição de dados

Um sistema de aquisição de dados, como uma placa, é um produto destinado a coletar informações para documentar ou analisar um fenômeno. A aquisição de dados, também conhecida como *Data Acquisition* (DAQ), é um processo de conversão de sinais do domínio analógico para o domínio digital para exibição, armazenamento e análise (SMITH, 2020).

Um produto moderno para aquisição de dados é composto internamente de sensores, condicionadores de sinais e conversores analógico-digital e externamente por entradas e saídas digitais, analógicas e entrada diferencial. Possui algumas características como a resolução, em bits, do sinal detectado e o intervalo de amostra, em amostra por segundo (OMEGA, 2014).

Para completar o sistema de aquisição de dados, têm-se os *softwares* programáveis DAQ para armazenamento e análise dos dados, fornecendo recursos visuais e diversas ferramentas para observação e manuseio dos dados.

2.4 TESTE DE USABILIDADE

Teste de usabilidade é uma ferramenta que emprega técnicas para coletar dados empíricos enquanto se observam os usuários finais utilizando o produto, de forma a realizar a tarefa no ambiente mais real possível, para testar se o produto atende os critérios de usabilidade definidos. O teste tem como objetivo, também, eliminar possíveis problemas e frustrações durante o uso do produto/programa (RUBIN; CHISNELL; SPOOL, 2011).

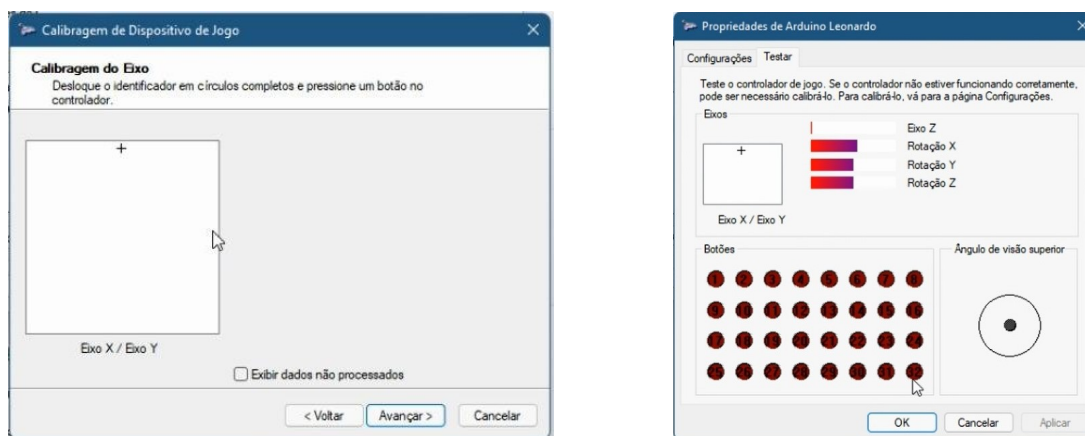
Basicamente, um teste de usabilidade possui seis etapas, sendo elas: desenvolvimento do plano de teste, seleção e recrutamento de participantes, preparação do material do teste, condução do teste em si, conversa com o participante após o teste e análise dos resultados (RAPOSO, 2018).

2.5 TRABALHOS RELACIONADOS

2.5.1 Controladores de jogo

No próprio sistema operacional *Windows*, há a possibilidade de se testar um controlador de jogo, ou seja, um periférico, como é o caso do *joystick*. Para isso, o usuário precisa ter conectado ao computador um dispositivo periférico, e assim, iniciar os testes. Esses testes permitem uma configuração e calibração dos potenciômetros, em função de eixos do sistema cartesiano. Já os botões digitais são testados em níveis lógicos alto e baixo, conforme Figura 15 (MICROSOFT, 2022a).

Figura 15 – *Windows Tester*.



(a) Calibração em função dos eixos cartesianos.

(b) Testes dos botões digitais.

Fonte: Autora.

No caso específico do *joystick* da RSK, a alimentação e transmissão de dados não ocorre através de um cabo USB e, por isso, não é possível realizar essa verificação dos parâmetros pelo próprio *Windows* de forma direta. Para realizar tal feito, seria necessário um microcontrolador conectado ao *joystick*, por exemplo, capaz de ler seus sinais e transformá-lo em um periférico de computador.

A leitura de dados utilizando o controlador de jogo do *Windows* é bastante simples, não permitindo a configuração dos sinais desejados (como adicionar uma variável no nome ou o tempo de leitura desse sinal), nem o armazenamento dessas informações ou exportação como um relatório. Para o usuário final, seriam apenas sinais sendo testados, sem a correta relação com a função do *joystick* ou os valores de referência ideais.

2.5.2 Aplicação de *software* com teste de usabilidade

Em relação aos trabalhos envolvendo teste de usabilidade de usuário, ganha-se especial atenção um *plugin* (leve alteração de *software* para personalizar um programa, aplicativo, etc) desenvolvido para automatização de etapas em projetos de instalações elétricas no *software AutoCad*, desenvolvido por Gomes (2022).

O *plugin* tem como propósito gerar de forma automática quadro de carga, diagrama unifilar, lista de material, e fazer sugestão em relação à quantidade mínima de pontos de tomada e à carga destinada por luminárias em determinado ambiente. O teste de usabilidade teve como objetivo validar o funcionamento do *plugin* através da aplicação em um público alvo desejado, no caso, pessoas relacionadas à área de projetos elétricos com experiência e atuação em instalações elétricas.

O teste aplicado teve resultados interessantes e positivos no que diz respeito ao *feedback* recebido sobre as funcionalidades desenvolvidas e ao funcionamento do *plugin* no geral, tanto em termos de resultados quantitativos quanto qualitativos. Também constatou um ganho significativo no ambiente de trabalho utilizando o *plugin* desenvolvido.

Esse tipo de teste permite que o desenvolvedor de *software* identifique possíveis problemas, erros ou falhas no programa que prejudiquem seu funcionamento, avalie os resultados de acordo com os critérios estabelecidos (facilidade, navegabilidade, etc) e receba também sugestão de aprimoramento e correções para trabalhos futuros (RUBIN; CHISNELL; SPOOL, 2011).

2.6 VERSÃO INICIAL DO TESTADOR DE *JOYSTICK*

O protótipo utilizado foi inicialmente desenvolvido durante o estágio obrigatório do curso de Engenharia Elétrica do Instituto Federal de Santa Catarina - Câmpus Itajaí (TAVARES, 2021). Para melhor ambientar o leitor, serão descritas tanto as novas funcionalidades do *software*, quanto as antigas já implementadas e o *hardware* desenvolvido. As novas alterações se encontram no capítulo 4 Desenvolvimento.

Quadro 1 – Divisão das etapas de desenvolvimento do testador de *joystick*.

Versão	Etapa do Desenvolvimento
Desenvolvido durante o estágio obrigatório	Estudo do <i>joystick</i> e comunicação
	Criação do <i>layout</i>
	Armazenamento de dados em tabela
	<i>Hardware</i> inicial
Desenvolvido durante o Trabalho de Conclusão de Curso (TCC)	Alteração do <i>layout</i>
	Controle de usuário
	Controle de simulação
	Geração de relatório PDF
	<i>Hardware</i> final

Fonte: Autora.

2.6.1 Estudo do joystick

O joystick (Figura 16), componente de foco principal nesse estudo, é um componente eletrônico composto por cinco botões digitais, quatro potenciômetros e dois LEDs, como mencionado anteriormente na seção 2.1.3 e como pode ser visto no Anexo A.

Figura 16 – Joystick da RSK fora e dentro da máquina.

(a) Joystick fora da RSK, com conexão aparente.



(b) Joystick na cabine do operador.



Fonte: Autora.

Porém, nem todos os componentes são utilizados na aplicação estudada, pois um dos potenciômetros que faz a função de inclinação do *spreader* (função de *tilt*, como popularmente chamada no cenário portuário), não está habilitado, assim como o uso dos LEDs, já que servem de auxílio para a função de *tilt*.

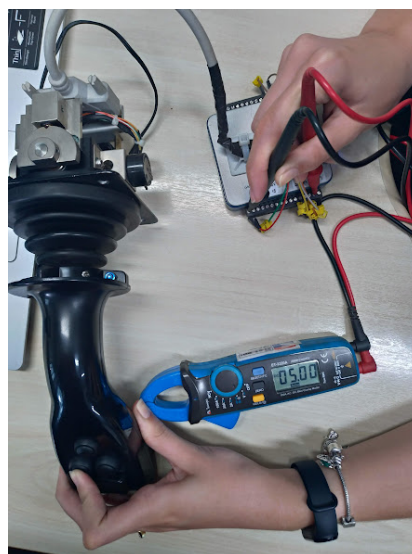
A comunicação dos dados oriundos do joystick deu-se através do cabo com conector fêmea DB25, compatível com a conexão elétrica de saída de dados do componente.

A conexão do cabo com conector DB25 é composta por 25 terminais, sendo que cada terminal transmite um sinal do joystick, e não necessariamente todos os terminais contêm informações. Para identificar os sinais provenientes de cada terminal, foi numerado cada um dos terminais (Figura 17a) e utilizado um multímetro para as medições (Figura 17b).

Posteriormente, após testar todas as saídas do joystick através do cabo com conector DB25 e com multímetro, foram anotados em uma tabela os resultados, e o código de cores foi estabelecido de acordo com o tipo de sinal (analógico, digital, alimentação e referência), conforme Tabela 1.

Figura 17 – Conexão para codificação de sinais do *joystick*.

(a) Cabos numerados para medição.



(b) Medição dos parâmetros utilizando multímetro.

Fonte: Autora.

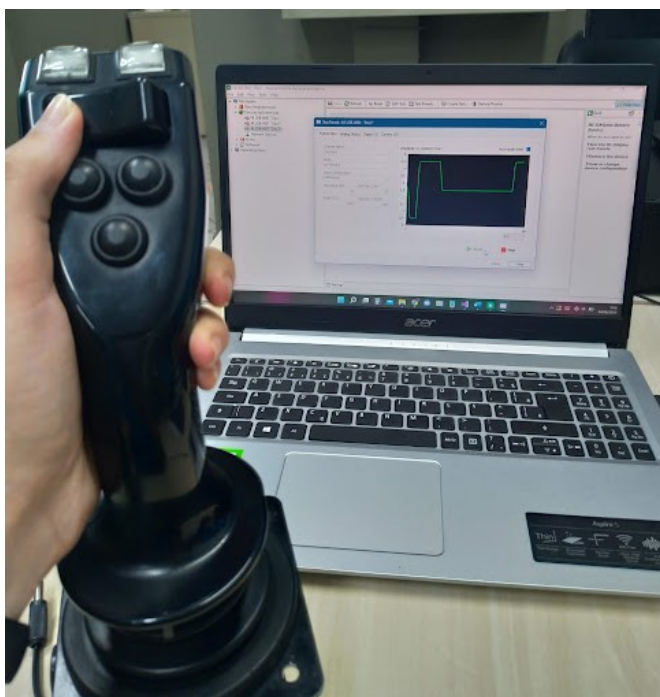
Tabela 1 – Conexão elétrica DB25 por função.

Tipo de conexão	Função	Conector DB25
Alimentação	-	Terminal 9
Alimentação	-	Terminal 15
Referência	-	Terminal 10
Analógico	Recolhe/Estende lança	Terminal 04
Analógico	Sobe/Desce lança	Terminal 08
Analógico	Gira <i>spreader</i>	Terminal 11
Digital	Deslocamento esquerda	Terminal 14
Digital	Deslocamento direita	Terminal 16
Digital	Homem morto	Terminal 19
Digital	Buzina/Acoplamento	Terminal 20

Fonte: Autora.

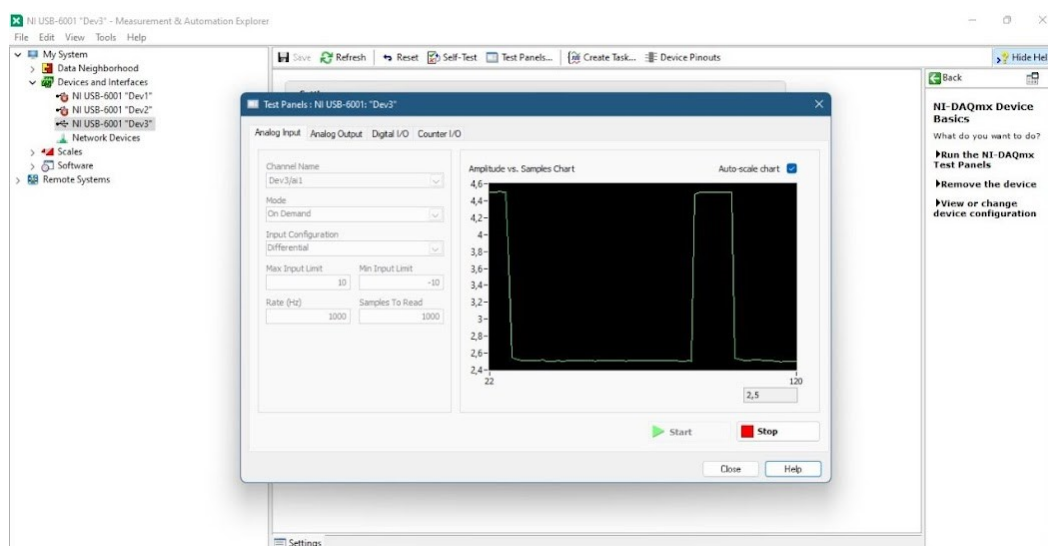
Através da codificação dos sinais e testes com o multímetro, foi possível identificar quais funções eram realizadas por potenciômetros e por botões digitais. Então, foram feitos testes de funcionalidade utilizando a NI USB-6001 (Figura 18), pois o aplicativo *NI Max* da *National Instruments* permite a visualização dos sinais em função do tempo, semelhante a um osciloscópio, e também a configuração dos sinais como sinais de entrada e saída, conforme Figura 19. Também possui função de teste de todos os botões digitais, e permite a configuração dos sinais como entradas e saídas analógicas ou digitais.

Figura 18 – Testes com NI Max.



Fonte: Autora.

Figura 19 – Painel de testes.



Fonte: Autora.

Assim, foi possível criar funções, dentro do aplicativo, para ler determinado sinal do *joystick* como entrada ou saída e configurar os botões digitais como padrão de nível lógico alto ou baixo, para identificar seu funcionamento.

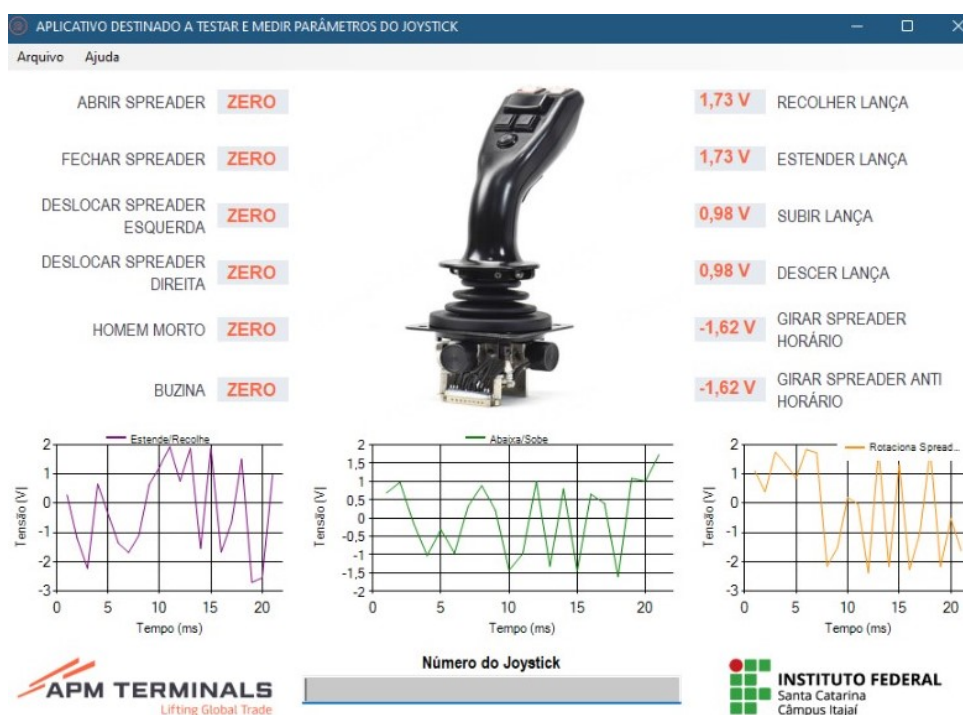
2.6.2 Criação do layout

O programa foi inteiramente desenvolvido na IDE *Visual Studio* na linguagem de programação C# (*C-Sharp*), no formato de *Windows Forms Application*, ou seja, aplicativo de computador.

O *layout* inicial foi pensado em centralizar a imagem do *joystick* no aplicativo, e dividir as funções provenientes de botões digitais e analógicos, portanto, do lado esquerdo do aplicativo ficaram as funções acionadas pelos botões digitais e do lado direito as funções referentes aos potenciômetros. Para referenciar os sinais, foi necessário utilizar bibliotecas da própria *National Instruments*, compatíveis com o VS.

Também foi implementado um campo de texto para armazenamento do número do *joystick*, que consta em cada componente da empresa. O *layout* inicial pode ser visto na Figura 20.

Figura 20 – *Layout* inicial do testador.



Fonte: Autora.

2.6.3 Armazenamento de dados em tabela

Para guardar os dados do teste de acordo com a função realizada, o campo de número do *joystick* precisa estar obrigatoriamente preenchido. Após, o usuário pode clicar na função de salvar o relatório e então visualizar os resultados salvos anteriormente através da função "Arquivo > Salvar".

Esses dados ficam armazenados em uma tabela, criada a partir de uma nova janela do aplicativo (novo *Forms*). Essa versão inicial mostrava-se o resultado dos testes de forma bem

simples, sem muitos recursos de formatação e exploração dos dados, conforme Figura 21. Esse histórico de testes é aberto em uma janela dentro do próprio *software*, o que dificultava seu uso em simultâneo com o menu de testes, além de não ser possível exportar os dados, imprimir ou anexar em um *email*.

Figura 21 – Tabela com resultado dos testes.

Número do Joystick	Estender/Recolher Lança Máximo	Estender/Recolher Lança	Estender/Recolher Lança Mínimo	Subir/Descer Lança Máximo	Subir/Descer Lança	Subir/Descer Lança Mínimo

Fonte: Autora.

Para conseguir tanto salvar o teste pela primeira vez quanto acessar posteriormente essa planilha, o usuário precisava acessar a pasta C: do disco rígido do computador e criar uma pasta com o nome “Teste Joystick” e um arquivo com a extensão .csv⁵ dentro da pasta, ficando com o caminho “@c:\Teste Joystick\Testes.csv”. Esse procedimento, mesmo que seja manual, só era necessário ser executado apenas uma vez, e caso não seja realizado, mostrará o seguinte erro de execução:

Figura 22 – Erro apresentado caso não seja criado o caminho da tabela de testes.

```

1 referencia
private void SaveFile()
{
    var linha =
        numerojoystick.Text + ";" +
        subir_descer.Text + ";" +
        estender_recolher.Text + ";" +
        girar_spr.Text + ";" +
        side_shift_left.Text + ";" +
        side_shift_right.Text + ";" +
        homem_morto.Text + ";" +
        buzina.Text;
        //nefiteste.Text + ";" +
        //nefiteste2.Text;
    using (StreamWriter arquivo = new StreamWriter(@"c:\Teste Joystick\Testes.csv", true))
        arquivo.WriteLine(linha);
}

```

Exceção Sem Tratamento

System.IO.DirectoryNotFoundException: 'Não foi possível localizar uma parte do caminho 'c:\Teste Joystick\Testes.csv'.'

Exibir Detalhes | Copiar Detalhes | Inicie a Sessão do Live Share...

Configurações de Exceção

Fonte: Autora.

⁵Arquivos *Comma-separated Values* (CSV) são arquivos de texto que podem ser utilizados para armazenar estruturas tabulares simples (MÁS et al., 2018) para uso em editores de planilhas externos, como *Microsoft Excel* e *Google Sheets*.

2.6.4 **Hardware inicial**

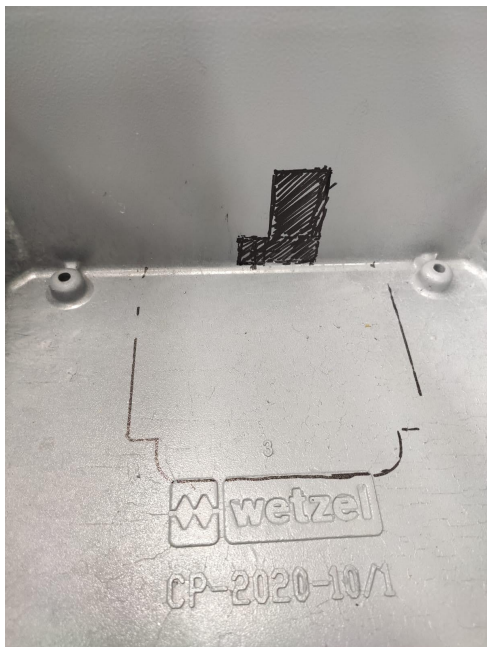
O desenvolvimento do *hardware* foi pensado a fim de tornar a experiência do usuário mais ergonômica para manusear o *joystick* e também para comportar o cabo DB25 e a placa de aquisição de dados NI USB 6001 em um só local.

Foi então reaproveitada uma caixa de passagem elétrica de sobrepor com dimensões de 20x20x15 centímetros de metal. Dentro da caixa fixou-se a placa de aquisição de dados, com auxílio de fita dupla face, juntamente com o cabo DB-25 devidamente conectado. Na placa de aquisição foi inserida uma etiqueta na direção das entradas digitais e analógicas com instrução de ligação.

Para encaixar o *joystick* na tampa, foi necessário serviço de solda para realizar o corte do metal, conforme medidas da base do *joystick*, para acoplar, conectar e prender o componente. Para passar o cabo USB da placa de aquisição de dados, também foi feito um corte na tampa em direção à placa. Os serviços de solda foram realizados com auxílio da equipe de manutenção, com as devidas ferramentas pertinentes.

Com a finalidade de melhorar o acabamento após os serviços de solda, foi feita a pintura do protótipo com um *spray* prata e algumas etapas podem ser vistas na Figura 23.

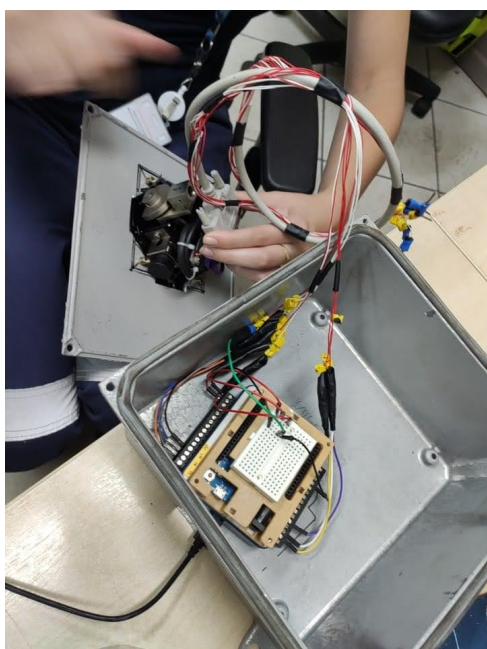
Figura 23 – Processo de desenvolvimento do *hardware*.



(a) Esboço para corte de solda lateral e fixação da placa de aquisição de dados.



(b) Caixa de passagem com corte lateral para conexão dos cabos.



(c) Processo de conexão dos cabos da placa de aquisição de dados e *joystick*.



(d) Protótipo final do *hardware* para acoplamento do *joystick*, placa de aquisição de dados e conexões.

Fonte: Autora.

3 MATERIAIS E MÉTODOS

Este capítulo tem como objetivo definir os principais recursos e materiais aplicados no decorrer do trabalho e está dividido em 6 subseções. Assim, é definido o tipo de trabalho realizado em função da abordagem, natureza, objetivos e procedimentos técnicos aplicados, as principais ferramentas computacionais utilizadas, os materiais e protótipo de *hardware* desenvolvido, a aquisição e análise de dados, a forma de instrução e treinamento da equipe de manutenção e o teste de usabilidade aplicado com os usuários finais.

3.1 CARACTERIZAÇÃO DA PESQUISA

A pesquisa pode ser definida de acordo com alguns critérios, como a abordagem, a natureza, os objetivos e os procedimentos técnicos, conforme definiu Gerhardt e Silveira (2009):

A pesquisa científica é o resultado de um inquérito ou exame minucioso, realizado com o objetivo de resolver um problema, recorrendo a procedimentos científicos (GERHARDT; SILVEIRA, 2009).

3.1.1 Abordagem

Em relação à abordagem da pesquisa, ela tem caráter quantitativo e qualitativo, pois analisa dados de origem matemática e também dados não-métricos (GOLDENBERG, 2004).

3.1.2 Natureza

Do ponto de vista de sua natureza, a pesquisa é do tipo aplicada, com foco em aplicação prática e resolução de problemas (GERHARDT; SILVEIRA, 2009).

3.1.3 Objetivos

Em relação aos objetivos da pesquisa, pode ser classificada como uma pesquisa exploratória-descritiva. Exploratória porque envolve levantamento bibliográfico e entrevista com as partes interessadas e descritiva, pois envolve técnica de coleta de dados e assume a forma de levantamento de informações (GERHARDT; SILVEIRA, 2009).

3.1.4 Procedimentos técnicos

Quanto aos procedimentos técnicos aplicados na pesquisa, pode-se considerar a aquisição e tratamento de dados através de pesquisa bibliográfica, pesquisa documental e pesquisa de levantamento de campo, complementando as duas primeiras pesquisas (GERHARDT; SILVEIRA, 2009).

3.2 SOFTWARES DE DESENVOLVIMENTO E PLATAFORMAS DIGITAIS DE APOIO

Alguns *softwares* e plataformas⁶ foram utilizadas ao longo do desenvolvimento desse trabalho, seja de apoio em alguma parte de teste ou etapa, seja para o desenvolvimento em si.

3.2.1 *Visual Studio Community*

O *Visual Studio* (VS) é um ambiente de desenvolvimento integrado (IDE) utilizado para a criação de aplicativos de computador, *sites*, serviços *web* e aplicativos móveis. Possui editor de código e depurador, e permite 36 linguagens de programação diferentes (MICROSOFT, 2022c), incluindo a linguagem C#. Foi a IDE escolhida para realizar a programação do *software*.

Para o desenvolvimento do *software* testador foi utilizado o VS na linguagem de programação C#. A linguagem foi escolhida com base na familiaridade com a Programação Orientada a Objetos (POO), já inicialmente estudada no curso de Engenharia Elétrica do Instituto Federal de Santa Catarina - Câmpus Itajaí e, principalmente, pela disponibilidade das bibliotecas .NET e DAQ NI utilizadas.

3.2.2 *GitHub*

O *GitHub* é uma plataforma de hospedagem de código-fonte e arquivos em geral. Permite o controle de versão de códigos e projetos, além do compartilhamento e colaboração com outros usuários (GITHUB, 2022).

Essa plataforma possui suporte *online* e programa para ser instalado, além de permitir integração com outras IDEs, como o VS e facilitar o uso durante o desenvolvimento de projetos, e foi utilizada para hospedar o repositório do presente trabalho durante o desenvolvimento.

3.2.3 *NI Max*

A *Nationals Instruments* (NI) possui um programa destinado para configuração dos produtos no âmbito de *software* e *hardware*, em que é possível criar canais, tarefas, interfaces, executar diagnósticos e até atualizar o produto, denominado *NI Measurement & Automation Explorer* (NI MAX). Seu *download* é juntamente com outros serviços da empresa e de forma gratuita (INSTRUMENTS, 2022b). Foi empregado durante os momentos iniciais de teste dos parâmetros para reconhecer as funções do *joystick*.

3.3 HARDWARE

O *hardware* foi desenvolvido com reaproveitamento de materiais disponíveis no setor de manutenção e a utilização de mão de obra interna para auxílio nos serviços de solda. Como havia

⁶Plataformas digitais são estruturas *online* baseada em *softwares*, operando por meio da tecnologia que conecta usuários e empresas em diversos setores (PRIBERAM, 2021).

um protótipo inicial, foram feitas algumas modificações, como pintura e inserção de identificação e instrução para o usuário. Em resumo, os principais materiais utilizados foram:

- Caixa de passagem elétrica metálica de sobrepor 20x20x15 cm para o protótipo de *hardware*;
- Placa de aquisição de dados modelo NI USB 6001;
- Computador com *Visual Studio* e *NI Max*;
- Cabo com conector macho DB25;
- Cabo USB.

3.4 AQUISIÇÃO E ANÁLISE DE DADOS

A comunicação entre o *software* e o *joystick* e a aquisição e leitura de dados se deu através da placa de aquisição de dados NI USB 6001 da *National Instruments* conectada ao computador via *Universal Serial Bus* (USB) (INSTRUMENTS, 2022a), para leitura dos sinais elétricos do *joystick*. A interface entre *joystick* e a NI USB 6001 foi realizada através da conexão dos terminais de um cabo com conector fêmea DB25.

A base de dados para o cálculo dos indicadores são as Ordens de Serviço (OS), que constam as manutenções realizadas. Para o caso de manutenções programadas (preventivas, preditivas e inspeções), há um roteiro de manutenção predefinido. Já para as manutenções não programadas/emergenciais, em especial os BD, tem-se a necessidade de especificar na OS o problema encontrado no equipamento, a causa deste problema e a solução aplicada.

Essas informações alimentam o *Computerized Maintenance Management System* (CMMS), ou Sistema Informatizado para Gestão da Manutenção (SIGM), que no caso da empresa trata-se do *Benner*. Assim, é possível extrair relatórios, calcular indicadores e realizar análises dos dados (BENNER, 2022).

O período para aquisição e análise dos dados foi dividido de acordo com a implementação do *software*. Para o primeiro período, consideraram-se os dados de janeiro de 2020 a agosto de 2021, anterior ao protótipo e, para o segundo período, consideraram-se os dados de setembro de 2021 a junho de 2022, posterior ao protótipo e às novas modificações no *software*.

Os dados adquiridos são apenas de OS de RSK (14 no total) e na forma de planilhas, para serem apurados e filtrados no *Microsoft Excel* posteriormente.

3.5 INSTRUÇÃO E TREINAMENTO DA EQUIPE

Para atingir o propósito de utilização do *software*, foi criado um roteiro de instrução para utilização do testador de forma independente, de acordo com os modelos dos procedimentos internos da empresa, além de treinamento dos profissionais de interesse para manuseio do programa.

O acompanhamento do desenvolvimento do trabalho e auxílio para implementação

do software e melhoria do hardware foi realizado pelo supervisor do setor de Planejamento e Controle de Manutenção (PCM), Thiago Augusto Pereira, que também foi o supervisor interno da empresa APM Terminals durante a realização do estudo

Ao final, também foi aplicado um teste de usabilidade, para avaliação dos resultados e coleta de *feedbacks* dos usuários após o uso.

3.6 TESTE DE USABILIDADE

Conforme mencionado anteriormente, o teste de usabilidade possui seis etapas para melhor andamento e êxito do teste (RUBIN; CHISNELL; SPOOL, 2011; RAPOSO, 2018). As seções seguintes descrevem e detalham essas etapas.

3.6.1 Desenvolvimento do plano de teste

3.6.1.1 Objetivos do teste

- Avaliar a facilidade de uso do testador;
- Identificar possíveis erros;
- Obter sugestões de novas implementações;
- Receber *feedback* acerca do testador.

3.6.1.2 Dados coletados

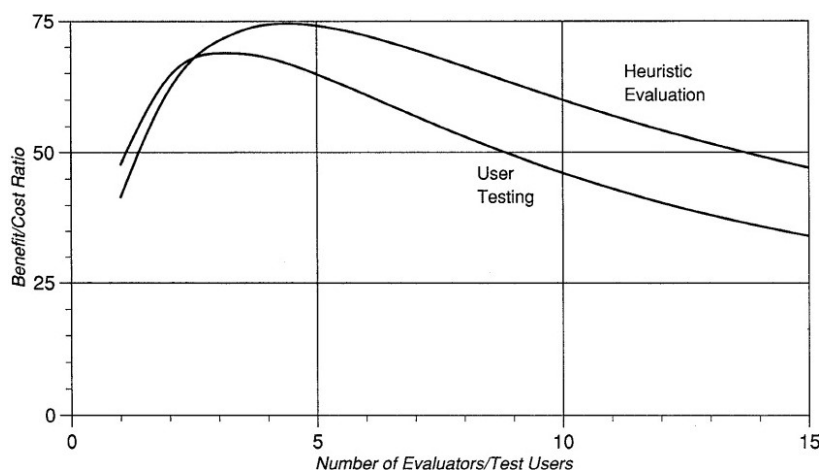
Os dados foram coletados a partir do próprio questionário de testes, desenvolvido no *Microsoft Forms*, e também contabilizado o número de relatórios em *Portable Document Format* (PDF) salvos durante a execução do teste, que representa a finalização de um teste.

3.6.2 Seleção e recrutamento dos participantes

Levando-se em consideração a área de atuação deste trabalho, os participantes são todos do setor de manutenção elétrica. A quantidade de participantes recrutados foi com base no estudo de Nielsen e Landauer (1993), que consta o número de participantes desejado de acordo com a porcentagem de problemas encontrados. De 4 a 5 participantes já apresenta mais de 75% de problemas de usabilidade encontrados, uma quantia bem expressiva, conforme Figura 24. Portanto, foram definidos 5 participantes.

O perfil de usuário de teste define as características principais dos usuários finais a testarem o produto. Os participantes foram classificados de acordo com a formação profissional, cargo que ocupa na empresa, faixa etária e tempo de experiência na área elétrica, conforme o Quadro 2 abaixo. A autora atuou apenas como moderadora dos testes e não como avaliadora.

Figura 24 – Relação entre benefício e número de participantes.



Fonte: Nielsen e Landauer (1993).

Quadro 2 – Características dos participantes.

Participante	Formação Profissional	Cargo	Faixa Etária	Tempo de Experiência
I	Graduando em Engenharia Elétrica	Inspetor de Manutenção Elétrica	31 - 40	>10 anos
II	Graduando em Engenharia Mecatrônica	Técnico Líder	31 - 40	>10 anos
III	Graduando em Engenharia Elétrica	Técnico Eletricista	31 - 40	>10 anos
IV	Superior Completo (Engenharia Elétrica)	Técnico Líder	21 - 30	>5 anos
V	Superior Completo (Engenharia Elétrica)	Supervisor de Manutenção	21 - 30	>5 anos

Fonte: Autora.

3.6.3 Preparação do material e ambiente de teste

O local de testes foi definido como a sala de reuniões da empresa, por maior facilidade logística para os participantes e moderadora. Inicialmente, foi preparado o local do teste, deixando disponível o computador com o programa do testador já instalado, o *hardware* do produto para acoplar o *joystick* e *joysticks* para teste.

Os *joysticks* selecionados foram: um *joystick* com conformidade, já realizado teste anteriormente, um *joystick* defeituoso e um *joystick* cujo estado de operação é desconhecido.

3.6.4 Condução do teste

Os testes aconteceram de forma individual com cada usuário participante e conduzido de forma moderada, havendo engajamento apenas quando necessário entre moderador e usuário.

3.6.5 Questionário pós teste

As perguntas realizadas foram divididas em seções, de acordo com as funcionalidades desenvolvidas, a geração de relatório em PDF e o *feedback* geral do programa. O objetivo é que as perguntas sejam focadas e diretas, de forma que não abra espaço para dúvidas ou respostas muito genéricas e abrangentes.

Quadro 3 – Perguntas do questionário de teste de usabilidade.

Seção	Questão
Funcionalidade desenvolvida	Facilidade de visualização dos resultados das funções testadas (botões digitais e potenciômetros). (Classificação de 1 a 5)
	Facilidade de navegabilidade do programa. (Facilidade de encontrar as informações dentro do menu do programa, como instrução de testes). (Classificação de 1 a 5)
	<i>Layout</i> do testador do <i>joystick</i> . (Se o <i>layout</i> é amigável com o usuário, seja em tamanho dos ícones, fontes, cores, disposição dos elementos na tela). (Classificação de 1 a 5)
Geração de relatório em PDF	Você utilizou o manual de instrução para gerar o relatório de teste em PDF? Se sim, considerando-o suficiente para guiá-lo durante os testes? (Resposta livre)
	Destaque os pontos positivos e negativos do relatório em PDF gerado após o teste. (Resposta livre)
Feedback geral	Você se deparou com algum erro durante o uso do programa? Se sim, mencione brevemente o problema. (Resposta livre)
	Caso deseje, utilize esse campo para deixar sugestões e comentários adicionais sobre o programa. (Resposta livre)

Fonte: Autora.

3.6.6 Procedimento de análise dos dados

A análise dos dados foi dividida em duas seções de acordo com o tipo de resultado analisado, dividindo-os em dados qualitativos e quantitativos (RUBIN; CHISNELL; SPOOL, 2011).

- Qualitativos: Respostas livres e *feedbacks* agrupados de acordo com o teor (positivo e negativo);
- Quantitativos: Respostas sobre funcionalidades medidas através de escala numérica, feita a média aritmética dos valores.

A apresentação desses dados é mostrada na forma de resultados, no Capítulo 5.

4 DESENVOLVIMENTO

O presente capítulo descreve o desenvolvimento do testador quanto às novas funcionalidades e alterações realizadas, o desenvolvimento do manual de instruções do testador para instruir os usuários e as etapas de levantamento de dados para posterior análise.

4.1 DESENVOLVIMENTO DO TESTADOR

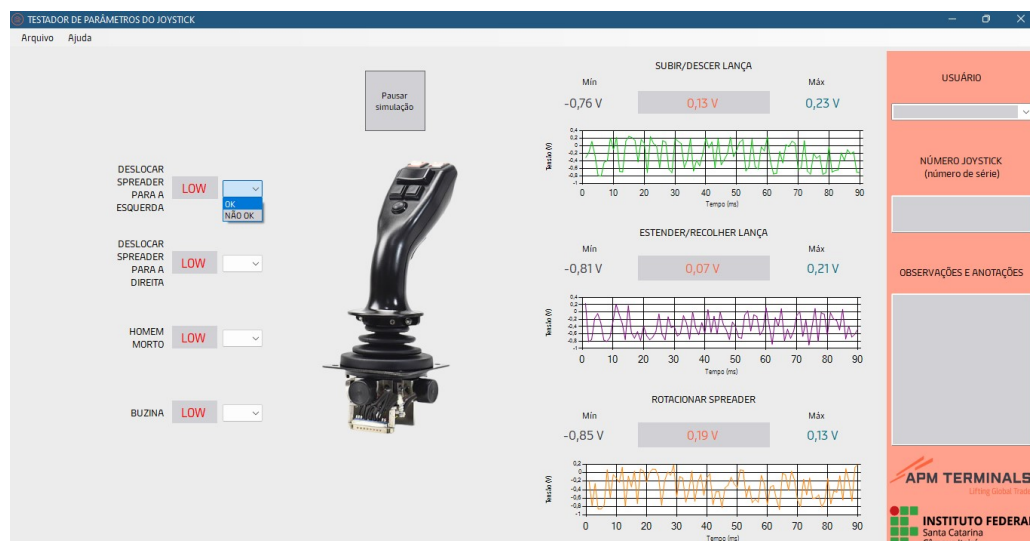
4.1.1 Alteração do layout

Na nova versão do testador, foram feitas algumas alterações no *software*, como a mudança de posição dos gráficos das funções analógicas, modificação do menu do programa, inclusão de campo de usuário, campo de observações e de controle de simulação.

A mudança da posição dos gráficos ocorreu para que a visualização do sinal em função do tempo estivesse mais próxima da função, juntamente com seus valores máximo, mínimo e em tempo real. Ainda sobre a alteração na visualização dos sinais durante o teste, foram acrescentados, ao lado das funções dos botões digitais, opções para escolher se o componente se comportou conforme o esperado ou não, devido às falhas intermitentes que poderiam levar o usuário a um falso resultado do teste, conforme Figura 25.

No menu do programa, foi incluída a função “Sobre”, que traz uma breve descrição sobre a utilidade do programa. Também foi incluído no menu um botão de “Instruções”, que abre um arquivo tipo PDF no visualizador padrão do computador do usuário com as instruções de uso do testador, para que o usuário consiga visualizar esse arquivo simultaneamente ao uso do programa.

Figura 25 – Novo *layout* do testador.



Fonte: Autora.

4.1.2 Controle de usuário

O controle de usuário foi implementado de forma simples, com base em uma lista de pessoas pré-selecionadas, integrantes da equipe de manutenção, que são autorizadas a acessar o computador e manusear o programa. Foi desenvolvido através de lista suspensa, pois não foi identificada a necessidade de criação de *login* e senha ou de banco de dados de usuário, pois o objetivo do testador é que seja amigável com o usuário e acessível, mesmo sem internet.

Dessa forma, o usuário apenas seleciona nessa lista seu nome (campo obrigatório para salvar o resultado posteriormente) e prossegue com os testes.

4.1.3 Controle de simulação

Uma função extra adicionada ao testador foi a do controle de simulação, que basicamente é um botão de “iniciar” e “parar”, com o intuito de que o programa não comece a ler os sinais instantaneamente, antes mesmo do usuário iniciar os testes. Essa função também permite fazer pausas durante o teste, para que o usuário verifique alguma situação inesperada, em especial em relação aos gráficos, e consiga fazer comentários pontuais, visualizando bem o intervalo de tempo, sem que o tempo continue infinitamente.

Para isso, algumas situações de intertravamento foram criadas, como: só é possível realizar a geração de PDF e armazenamento do resultado do teste após a simulação ter sido rodada ao menos uma vez e que esteja pausada, além das outras condições de salvamento de testes.

4.1.4 Geração de relatório PDF

O relatório em PDF foi implementado para tornar mais profissional a apresentação do resultado de teste, trazendo não apenas dados numéricos em uma tabela, como era feito anteriormente. Com a geração em PDF, os relatórios ficam automáticos e padronizados. Para gerar o relatório, o usuário precisa, obrigatoriamente, preencher os campos de usuário, número do *joystick* e condições dos botões digitais. O arquivo gerado em PDF foi dividido em 3 partes: informações gerais, resultados digitais e resultados analógicos.

Na primeira parte, o relatório traz informações acerca do usuário, número do *joystick*, data e horário do teste e observações anotadas. Na parte de resultados, foi dividido em duas formas de visualização. A primeira trazendo o resultado do teste dos botões digitais, na forma de conformidade ou não conformidade. A outra parte diz respeito aos resultados dos testes analógicos, apresentando-os em forma de tabela com os valores mínimo, em repouso (quando não é realizado nenhuma função no *joystick*) e máximo. Em seguida, é apresentado, em forma de imagem, o resultado dos três gráficos analógicos, de cada uma das funções.

Para a implementação do relatório, sua configuração e geração de PDF no final do teste, foi utilizada uma biblioteca, a *iTextSharp*. É uma biblioteca de código aberto capaz de gerar documentos e relatórios com base em dados de um arquivo *eXtensible Markup Language*

(XML)⁷, ou banco de dados, preencher formulários interativos e enviar documentos PDF gerados ou manipulados dinamicamente para um navegador da web (ITEXT, 2022).

Durante o desenvolvimento, foi criada no código a função “GerarRelatorio()” e dividida em regiões, de acordo com as implementações realizada, através da função “#region” antes de cada seção, conforme Figura 26. Assim, mantém o código mais organizado e estruturado.

Na primeira parte, de configuração de data e hora, foi definido para adquirir as informações de data e hora no momento de submissão do teste, e guardado o valor com a correta configuração de fuso horário. Na parte de configurações iniciais do arquivo, foi definido o tamanho do arquivo (no caso, folha A4), margens do arquivo, nome de cada arquivo PDF salvo e seu caminho. Na região de cabeçalho consta o caminho e posição das imagens do cabeçalho do arquivo e o título.

Nas informações gerais, imprime dados de identificação como número do *joystick*, nome do usuário responsável pelo teste, data e horário. Por último, a última região diz respeito aos resultados do teste, responsável por imprimir os dados oriundos de botões digitais e potenciômetros, configurá-los em tabela, pegar informações do campo de observações e a resposta dos potenciômetros em função do tempo. Um exemplo de relatório gerado após um teste de *joystick* é apresentado no Apêndice B.

Figura 26 – Organização do código para criação do relatório em PDF.

```
1 referência
private void GerarRelatorio()
{
    Configuração de data e hora
    Configurações iniciais do arquivo
    Cabeçalho do relatório
    Informações Gerais do Relatório
    Resultado do Teste
}
```

Fonte: Autora.

4.1.5 Melhoria no *hardware*

O protótipo do *hardware* passou por algumas alterações. Uma delas é a cor, que posteriormente foi alterada a cor da pintura para preto e acrescentado instruções de uso na tampa. Também foi feito mais um corte de solda na tampa da caixa de passagem, para passar o cabo USB de conexão ao computador por cima, devido alteração na posição da placa de aquisição de dados. Por fim, a placa passou a ficar na posição vertical, fixada na lateral da caixa, para que as

⁷Tipo de arquivo para criar documentos com dados organizados.

conexões do *joystick* fiquem mais livres e com menor chance de obstrução no interior da caixa. As modificações realizadas podem ser vistas na Figura 27.

Figura 27 – *Hardware* do testador após modificações.



(a) *Hardware* final.



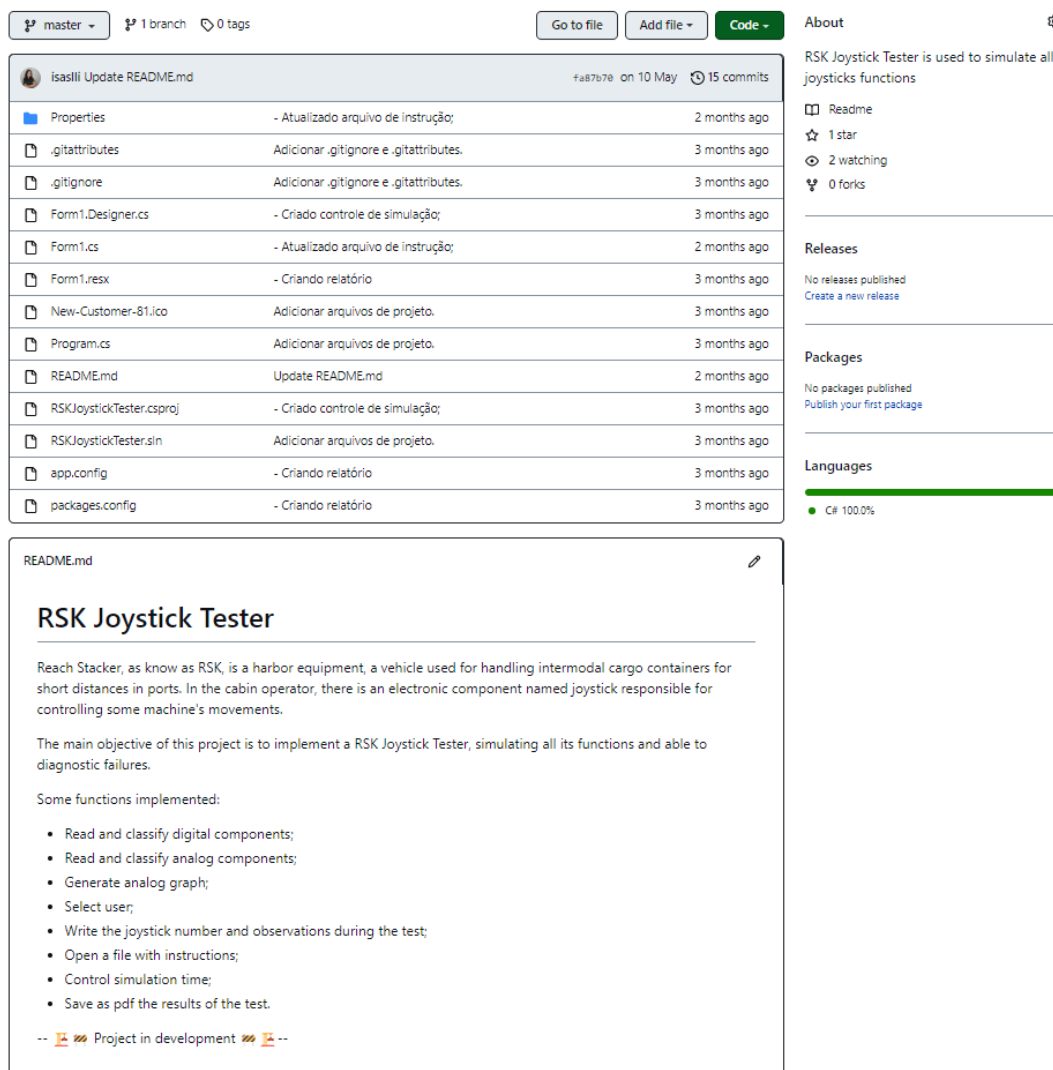
(b) Placa fixada internamente.

Fonte: Autora.

4.1.6 Repositório no *Github*

Foi criado um repositório intitulado de “*RSKJoystickTester*” no *Github* para hospedagem do código (TAVARES, 2022). Na Figura 28 é possível destacar alguns itens: ao lado do nome dos arquivos há uma descrição, que é resultado do *commit* realizado para controle das atualizações realizadas; também há uma breve descrição do projeto, através do arquivo “*README.md*”, localizado logo abaixo dos demais arquivos; ainda traz as linguagens detectadas nos arquivos, no caso do projeto, o C#.

Figura 28 – Projeto do testador na plataforma *GitHub*.



Fonte: Autora.

4.2 DESENVOLVIMENTO DO MANUAL DE INSTRUÇÕES

O manual de instruções de uso do testador foi desenvolvido a fim de orientar o usuário durante os testes, seja para sanar dúvidas de utilização ou até mesmo de conexão elétrica.

Nele, constam informações gerais, como o propósito da criação do testador e traz também imagens do *layout* do *software* para ambientar o usuário. Dispõe de uma parte com explicação sobre como realizar um teste com êxito e como salvá-lo posteriormente, além de demonstrar os valores considerados ideais do *joystick*.

Possui ainda um passo a passo de conectividade, instruindo o usuário como conectar o cabo com conector DB25 no *joystick* e também as conexões entre o cabo e a placa de aquisição de dados NI USB 6001, caso algum cabo desconecte.

Esse manual de instruções está disponível em PDF dentro do próprio programa do testador (Apêndice A).

4.3 LEVANTAMENTO DE DADOS

Os dados extraídos são referentes ao tempo de equipamento parado por falhas no *joystick* de forma não programada, ou seja, que causa impacto no indicador de confiabilidade do equipamento. Esse tipo de parada ocorre quando o operador da máquina aciona a equipe de manutenção, durante o uso do equipamento, relatando algum problema referente às funções realizadas pelo *joystick*. O problema é então diagnosticado e identificado que, de fato, a causa se encontra no mau funcionamento do *joystick*.

A partir do CMMS, foi extraído um relatório, em formato de planilha, com todas as paradas de RSK registradas em OS's. Com o banco de dados, é possível aplicar fórmulas para cálculo das horas paradas e avaliar os impactos das paradas.

De acordo com um levantamento realizado nas OS's, as principais causas desses problemas são: algum botão ou potenciômetro do *joystick* danificado, algum componente interno do *joystick* ou conexão com o módulo da RSK com mau contato e mesa (base) do *joystick* frouxa. Os problemas que causam maior impacto, são mais custosos e demandam mais tempo de reparo, são os que necessitam da troca do *joystick*.

Para melhor aquisição e tratamento dos dados, foram filtrados esses principais problemas para análise posterior.

4.3.1 Antes da implementação do *software*

O período inicial foi considerado de janeiro de 2020 a agosto de 2021, totalizando 20 meses. Esse período foi selecionado pelo fato de ser anterior à implementação inicial do *software* e por ser um período com mais informações acerca das paradas dos equipamentos (anterior a 2020 as informações lançadas no CMMS não tinham no título a descrição completa do problema, dificultando a coleta de dados).

O somatório de todas as paradas por problemas de *joystick* de todas as RSK ao longo desse período totalizou 318,91 horas, uma média de 15,95 horas por mês.

4.3.2 Depois da implementação do *software*

Considerando o período de setembro de 2021 a junho de 2022 (10 meses), o somatório de todas as paradas por problemas de *joystick* de todas as RSK ao longo desse período totalizou 36,17 horas, uma média de 3,62 horas por mês.

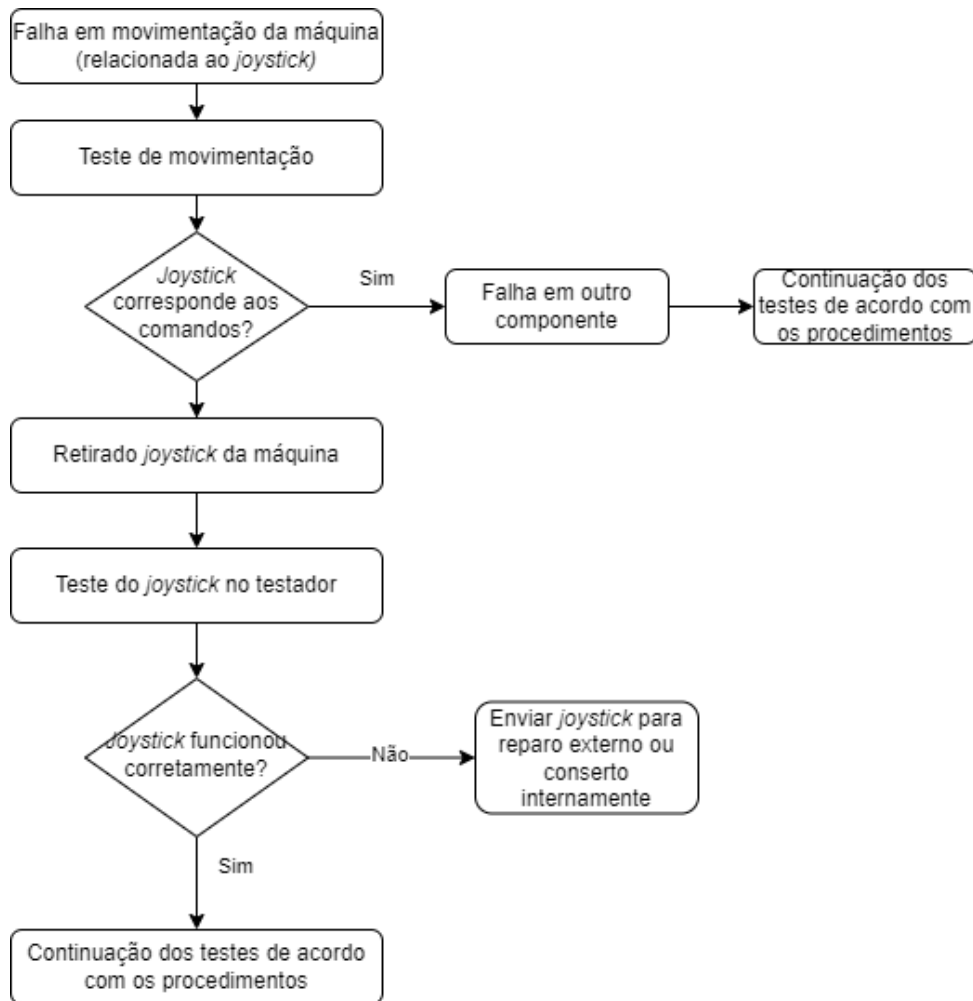
5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Este capítulo sintetiza os resultados obtidos a partir da implementação do *software* testador de *joystick*, através das funcionalidades aplicadas, da análise dos indicadores de desempenho e do teste de usabilidade aplicado.

5.1 SOFTWARE TESTADOR DE JOYSTICK

A aplicação do *software* testador de *joystick* representou uma diminuição de etapas para identificação de falha no *joystick* e, conseqüentemente, diminuição do tempo para solucionar o problema, conforme ilustrado no fluxograma da Figura 29.

Figura 29 – Fluxograma de testes para identificar falha do *joystick* após implementação do testador.



Fonte: Autora.

5.1.1 Métricas de código da IDE

Uma das novas funcionalidades desenvolvidas para o TCC foi a alteração do *layout* do programa, com os gráficos em tempo real das funções realizadas pelo *joystick* próximos dos seus valores adquiridos e do respectivo nome da função. Somada a essa alteração, foi adicionado o botão de controle de simulação e o campo de seleção de usuário responsável pelo teste.

Nesse novo *layout* foi modificado o menu do usuário, trazendo como opção o acesso ao manual de instruções, informações acerca do programa, opção de fechar o programa e salvar relatório em PDF, principal ganho dessa nova versão.

Mesmo com a implementação de novas funcionalidades, a reestruturação do código trouxe um ganho no que diz respeito às métricas de código dentro da IDE. Um indicador importante é o índice de facilidade de manutenção, que varia de 0 a 100, e quanto mais próximo do valor máximo, mais fácil de se manter o código (MICROSOFT, 2022b).

Na versão anterior do *software* já possuía um bom índice e, apesar do aumento de linhas de código-fonte (número de linhas totais) e código-executável (número de linhas desconsiderando comentários, espaços em branco, etc), o indicador aumentou ligeiramente, conforme Tabela 2.

Tabela 2 – Comparativo entre as métricas de código das versões.

Métrica	Versão Estágio	Versão TCC
Índice de Facilidade de Manutenção	67	72
Linhas de Código-Fonte	1481	2135
Linhas de Código-Executável	769	1269

Fonte: Adaptado do VS (2022).

5.2 INDICADORES DE DESEMPENHO

5.2.1 Confiabilidade

A partir da Equação 1 mencionada na Seção 2.2.1, tem-se a confiabilidade das RSK nos dois períodos e a relação entre as horas paradas por falha no *joystick* em comparação com as horas paradas totais por falhas.

A Tabela 3 demonstra a redução do número de horas paradas por falhas no *joystick*. Apesar do período II ser menor em número de meses, proporcionalmente ainda houve redução das horas paradas e aumento na confiabilidade geral das RSKs.

Tabela 3 – Impacto das paradas por *joystick* na confiabilidade das RSKs.

Métrica	Período I	Período II
Tempo total de <i>breakdown</i>	5639,17 horas	2291,06 horas
Somatório de horas paradas por <i>joystick</i>	328,91 horas	36,17 horas
Relação de horas paradas por <i>joystick</i> em relação a horas paradas por BD	6,13 %	1,58 %
Confiabilidade geral	95,11 %	95,21 %

Fonte: Autora.

5.2.2 Disponibilidade

Para a análise do impacto do *joystick* na disponibilidade das RSK's, foi utilizada a Equação 2, além de analisar também o impacto da indisponibilidade do *joystick*, que representou uma redução considerável, conforme Tabela 4.

Tabela 4 – Impacto das paradas por *joystick* na disponibilidade das RSKs.

Métrica	Período I	Período II
Disponibilidade	91,10 %	89,89 %
Indisponibilidade do <i>joystick</i>	0,17 %	0,04 %

Fonte: Autora.

5.2.3 Custo

O custo foi dividido em duas categorias, sendo a primeira o custo absoluto de reparos do *joystick* e a segunda o custo médio de manutenção (preventivas e corretivas), por hora, das RSK's. Tanto o custo absoluto de reparo do *joystick*, quanto o custo médio de manutenção representou uma redução considerável.

Tabela 5 – Custo médio de parada por *joystick*.

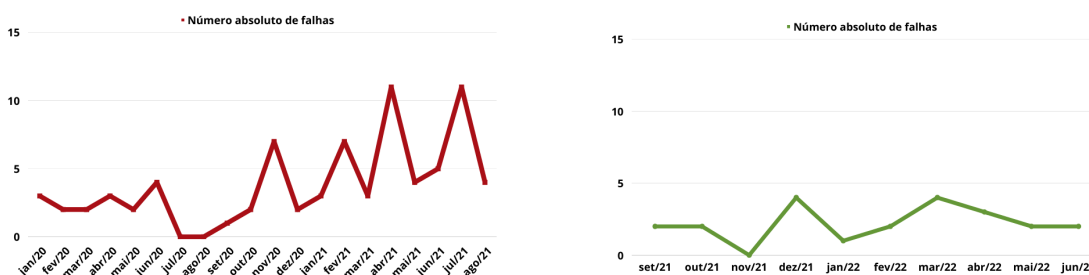
Métrica	Período I	Período II
Reparo do <i>joystick</i>	R\$ 13.233,21	R\$ 2.268,75
Custo médio de manutenção por hora	R\$ 86,00	R\$ 86,00
Somatório de horas paradas por <i>joystick</i>	328,91 horas	36,17 horas
Custo médio de manutenção por <i>joystick</i>	R\$ 28.286,26	R\$ 3.110,62

Fonte: Autora.

5.2.4 MTBF e MTTR

De acordo com as Equações 3 e 4, os indicadores MTBF e MTTR levam em consideração o somatório do número de falhas. O primeiro período de análise totalizou $n_{f1} = 76$ ocorrências e o segundo período $n_{f2} = 22$ ocorrências, conforme Figura 30. O resultado, em horas, dos indicadores, pode ser visto na Tabela 6, ambos representando expressiva variação positiva.

Figura 30 – Número absoluto de falhas por período.



(a) Número absoluto de falhas no período I.

(b) Número absoluto de falhas no período II.

Fonte: Autora.

Tabela 6 – MTBF e MTTR por falhas de joystick.

Métrica	Período I	Período II	Variação
Número de falhas	76	22	- 71,05 %
MTBF	1444,75 horas	2173,32 horas	50,77 %
MTTR	4,33 horas	1,64 horas	- 62,01 %

Fonte: Autora.

5.3 TESTE DE USABILIDADE DE USUÁRIO

5.3.1 Funcionalidades desenvolvidas

Foram realizadas 3 perguntas acerca das funcionalidades desenvolvidas, todas de caráter quantitativo. Na Tabela 7 consta a pergunta realizada no teste e, como resultado, a média aritmética das respostas dos usuários, seguida do desvio padrão.

A escala de avaliação foi de 1 a 5, em que 1 representa pouco fácil/amigável e 5 muito fácil/amigável. Observa-se que todas as perguntas obtiveram ótimos resultados.

Tabela 7 – Média das avaliações referente às funcionalidades desenvolvidas.

Pergunta	Média aritmética dos resultados	Desvio padrão
Facilidade de visualização dos resultados das funções testadas (botões digitais e potenciômetros)	4,80	0,44
Facilidade de navegabilidade do programa	4,80	0,44
Layout do testador do joystick	4,60	0,89

Fonte: Autora.

5.3.2 Geração de relatório em PDF

As duas perguntas realizadas sobre a geração de relatório em PDF (nova funcionalidade desenvolvida) foram perguntas abertas, com respostas livres e de caráter qualitativo e agrupadas conforme o teor (respostas positivas e respostas com críticas construtivas).

Sobre o manual de instruções todos os comentários foram positivos e afirmando que o manual é uma excelente ferramenta para sanar possíveis dúvidas. Já sobre a geração do relatório em PDF, houve apenas um usuário que avaliou com alguns pontos relevantes:

- Informações geradas: úteis e bem dispostas no relatório, porém melhorar a disposição/legenda dos gráficos gerados;
- Geração do PDF: sugestão de ser mais amigável com o usuário e apresentar um botão na tela inicial para gerar o relatório e mostrar os requisitos necessários.

Tabela 8 – Junção das avaliações referentes à geração de relatório em PDF.

Pergunta	Avaliação positiva	Crítica construtiva
Você utilizou o manual de instrução para gerar o relatório de teste em PDF? Se sim, considerando-o suficiente para guiá-lo durante os testes? (Resposta livre)	5/5	0/5
Destaque os pontos positivos e negativos do relatório em PDF gerado após o teste. (Resposta livre)	4/5	1/5

Fonte: Autora.

5.3.3 Feedback geral

Nessa última etapa, também de resposta livre, o usuário, caso desejasse, poderia deixar algum comentário adicional sobre o programa, sugestão de melhoria ou dificuldade encontrada.

Dentre as respostas gerais, destacam-se os comentários acerca da importância do *software* desenvolvido e resultados alcançados e, na sugestão mencionada anteriormente na pergunta sobre geração de relatório em PDF, de deixar ainda mais amigável ao usuário a geração do relatório de teste. Houve um caso de dificuldade de testes, não em relação ao *software*, mas ao *hardware*, pois o conector DB25 apresentou mau contato.

6 CONCLUSÃO

O *software* testador de *joystick* de RSK se mostrou fidedigno ao realizar a simulação e teste das funções desejadas com êxito, correspondendo aos movimentos da máquina. Mesmo com a implementação de novas funcionalidades, aumento de número de linhas de código-fonte, o código teve uma ligeira melhoria no índice de facilidade de manutenção (Tabela 2).

Destaca-se, dentre as alterações realizadas, por propiciar um histórico de testes através do recurso de salvar o relatório de testes em PDF, de forma a agrupar os resultados obtidos, informações de usuário, data, hora e observações de teste. A geração de relatórios em PDF é o principal diferencial em relação à realização de testes com multímetros e osciloscópios de forma tradicional.

O *software* desenvolvido preencheu uma lacuna existente para testes em *joysticks* de máquinas portuárias tipo RSK. Mais ainda, pode ser facilmente adaptado para outros tipos de máquinas. Outro ponto importante que saciou é a dificuldade de encontrar modelos comerciais disponíveis de *software* para se testar esse componente, principalmente ao levar em consideração o contexto que está inserido, portos e áreas retroportuárias, que é de suma importância o diagnóstico rápido de falhas.

No que se diz respeito aos indicadores de desempenho para a manutenção, obteve-se uma significativa redução do número de horas paradas por falhas no *joystick*, passando de 328,91 para 36,17 horas, nos períodos analisados, o que representa 6,13% e 1,58% do tempo total de horas paradas, respectivamente, em cada período. A confiabilidade geral teve um leve aumento, de 0,1%, o que pode ser explicado pela redução do número de horas paradas por falha no *joystick*.

A indisponibilidade das máquinas RSK por falha no *joystick* sofreu uma queda de 0,13%, o que parece pouco, mas que, ao considerar todas as paradas da máquina, e que o maior impacto de máquina indisponível acontece devido à manutenção preventiva, é uma boa melhoria. A disponibilidade geral decaiu 1,21%, o que acredita representar um aumento de outros tipos de paradas da máquina.

Um grande impacto foi referente ao custo de reparo, quase 6 vezes menor que no primeiro período, reduzindo assim o custo e a frequência de envio para reparo externo. Consequentemente, também reduz-se o custo médio de manutenção, que contempla todos os tipos de paradas, materiais e ferramentas aplicados.

A diminuição da frequência de ocorrências de falhas no *joystick* resultou em um aumento do tempo médio entre falhas e uma redução significativa, de 2,64 vezes, do tempo médio de reparo. Todos esses indicadores comprovam que houve um impacto positivo após a implementação do *software* no setor de manutenção estudado.

Os testes de usabilidade com os usuários finais, além de comprovar o êxito da implementação do *software*, também abriu espaço para sugestões de trabalhos futuros. O desenvolvimento deste tipo de trabalho também é importante para a capacitação interna dos trabalhadores, com

enriquecimento do laboratório de testes, oportunidades de colaboração e de reparo interno do *joystick*.

6.1 TRABALHOS FUTUROS

Pretende-se dar continuidade neste trabalho com a atualização do *software*, a fim de implementar as sugestões levantadas durante o teste de usabilidade, como incluir o botão de gerar relatório diretamente na tela inicial do aplicativo para tornar o *layout* ainda mais amigável para o usuário.

Durante o desenvolvimento das novas funcionalidades, também foram levantados alguns pontos de melhoria futuras, por exemplo, a criação de uma lista suspensa em arquivo de texto para carregar os nomes dos usuários, para se tornar personalizada e sem a necessidade de alteração no *software*, e a escolha do local para salvar o relatório de testes.

Abre espaço também para o desenvolvimento de testadores de parâmetros de outros componentes eletrônicos dos equipamentos do setor portuário, além do aproveitamento da poderosa placa de aquisição de dados utilizada.

6.2 CONSIDERAÇÕES FINAIS

O desenvolvimento do trabalho possibilitou o estudo aprofundado do dispositivo eletrônico *joystick*, interações e intercâmbio de conhecimentos dentro do setor. Além de analisar o impacto da implementação do *software* no setor de manutenção, não apenas no que diz respeito aos indicadores de desempenho, mas também na rotina dos trabalhadores envolvidos.

Referências

- BENNER. **Benner**. 2022. Disponível em: <<https://www.benner.com.br/>>. Acesso em: 13 de abril de 2022. Citado na página 27.
- BRASIL. Lei nº 12.815, de 5 de junho de 2013. **Diário Oficial da República Federativa do Brasil**, Brasília, DF, 2013. ISSN 1677-7042. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2013/lei/112815.htm>. Citado na página 1.
- BURGER, J. **MNR Key Performance Indicator Definition**. 2017. Disponível em: <AcervoEmpresarial>. Acesso em: 11 de abril de 2022. Citado 3 vezes nas páginas 12, 13 e 14.
- CASAVELLA, E. **Ambientes Integrados de Desenvolvimento em Linguagem C**. 2012. Disponível em: <<http://linguagemc.com.br/ambientes-integrados-de-desenvolvimento-em-linguagem-c/>>. Acesso em: 09 de julho de 2022. Citado na página 14.
- GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de Pesquisa**. 1. ed. Porto Alegre: Universidade Federal do Rio Grande do Sul, 2009. Citado na página 25.
- GITHUB. **GitHub**. 2022. Disponível em: <<https://github.com>>. Acesso em: 12 de abril de 2022. Citado na página 26.
- GOLDENBERG, M. **A arte de pesquisar**. 8. ed. Rio de Janeiro: Record, 2004. Citado na página 25.
- GOMES, M. D. **Plugin de AutoCad para automatização de etapas do desenvolvimento de projetos de instalações elétricas**. [s.n.], 2022. Disponível em: <https://drive.google.com/file/d/1wxfPg6AYbjk2AdsyIxGqPIIszkZA_P9u/view>. Citado na página 17.
- ILJUJJKIN. **Reachstacker Kalmar DRG450 Gloria K-Motion Modelo 3D**. 2015. Disponível em: <https://www.turbosquid.com/pt_br/3d-models/3dsmax-reachstacker-kalmar/912999>. Acesso em: 21 de fevereiro de 2022. Citado na página 6.
- INSTRUMENTS, N. **USB 6001 - NI**. 2022. Disponível em: <<https://www.ni.com/pt-br/support/model.usb-6001.html>>. Acesso em: 21 de fevereiro de 2022. Citado na página 27.
- INSTRUMENTS, N. **What is NI Measurement Automation Explorer (NI MAX)**. 2022. Disponível em: <<https://www.ni.com/pt-br/support/documentation/supplemental/21/what-is-ni-measurement---automation-explorer--ni-max--.html>>. Acesso em: 24 de junho de 2022. Citado na página 26.
- ITAJAÍ, O. **Spreader**. 2022. Disponível em: <<http://www.ogmoitajai.com.br/portal/info/spread.asp>>. Acesso em: 11 de abril de 2022. Citado na página 8.
- ITAJAÍ, O. **TEU**. 2022. Disponível em: <<http://www.ogmoitajai.com.br/portal/info/teu.asp>>. Acesso em: 13 de abril de 2022. Citado na página 13.
- ITEXT. **iTextSharp**. 2022. Disponível em: <<https://itextpdf.com/products/itextsharp>>. Acesso em: 29 de maio de 2022. Citado na página 33.

KALMAR. **Kalmar Workshop Manual**. Suécia: Lidhult: Cargotec, 2014. 334 p. Citado 5 vezes nas páginas 5, 6, 8, 10 e 11.

KALMAR. **Kalmar Essential Reachstacker**. 2022. Disponível em: <<https://www.kalmarglobal.com/equipment-services/reachstackers/essential-reachstacker/>>. Acesso em: 21 de fevereiro de 2022. Citado na página 1.

KEEDI, S. **Transportes, unitização e seguros internacionais de carga: prática e exercícios**. Aduaneiras, 2003. ISBN 9788571293663. Disponível em: <<https://books.google.com.br/books?id=KODXAAAACAAJ>>. Citado na página 8.

MÄS, S. et al. **Generic Schema Descriptions for Comma-Separated Values Files of Environmental Data**. In: . [S.l.]: International Conference on Geographic Information Science., 2018. Citado na página 22.

MICROSOFT. **How to troubleshoot game controllers in Microsoft games**. 2022. Disponível em: <<https://support.microsoft.com/en-us/topic/how-to-troubleshoot-game-controllers-in-microsoft-games-ee731224-b2d7-f25e-3c2a-338b27dbe1f5>>. Acesso em: 29 de junho de 2022. Citado na página 16.

MICROSOFT. **Métricas de código – intervalo e significado do índice de manutenção**. 2022. Disponível em: <<https://docs.microsoft.com/pt-br/visualstudio/code-quality/code-metrics-maintainability-index-range-and-meaning?view=vs-2022>>. Acesso em: 19 de julho de 2022. Citado na página 38.

MICROSOFT. **Visual Studio Community**. 2022. Disponível em: <<https://visualstudio.microsoft.com/pt-br/vs/community>>. Acesso em: 12 de abril de 2022. Citado na página 26.

NIELSEN, J.; LANDAUER, T. K. **A mathematical model of the finding of usability problems**. 1993. Citado 2 vezes nas páginas 28 e 29.

OLIVEIRA, A. **Itajaí: realizando sonhos, ultrapassando fronteiras e definindo destinos**. 1. ed. Itajaí: Sa Bureau Ltda, 2011. Citado na página 1.

OMEGA. **Aquisição de Dados**. 2014. Disponível em: <<https://br.omega.com/prodinfo/aquisicao-de-dados.html>>. Acesso em: 09 de julho de 2022. Citado na página 15.

PRIBERAM. **plataforma**. 2021. Disponível em: <<https://dicionario.priberam.org/plataforma>>. Acesso em: 09 de julho de 2022. Citado na página 26.

RAPOSO, A. **Testes de Usabilidade**. 2018. Disponível em: <https://web.tecgraf.puc-rio.br/~abraposo/inf1403/INF1403_06_testes_usabilidade.pdf>. Acesso em: 28 de junho de 2022. Citado 2 vezes nas páginas 16 e 28.

ROMANI, L. S. **Análise e Implantação de Repositório Digital utilizando Software Livre DSpace**. 2009. Disponível em: <<http://hdl.handle.net/11077/286>>. Acesso em: 08 de julho de 2022. Citado na página 15.

RUBIN, J.; CHISNELL, D.; SPOOL, J. **Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests**. Wiley, 2011. ISBN 9781118080405. Disponível em: <https://books.google.com.br/books?id=l_e1MmVzMb0C>. Citado 4 vezes nas páginas 15, 17, 28 e 30.

SEBESTA, R. **Conceitos de Linguagens de Programacao**. Bookman, 2003. ISBN 9788536301716. Disponível em: <https://books.google.com.br/books?id=b0tcn_uPLoAC>. Citado na página 14.

SKAF S. LAMROUS, Z. H. A.; MANIER, M. A. **Single quay crane and multiple yard trucks scheduling problem with integration of reach-stacker cranes at port of Tripoli-Lebanon**. **IEEE International Conference on Systems, Man and Cybernetics (SMC)**, 2019. Citado na página 1.

SMITH, G. M. **O que é Aquisição de Dados - DAQ ou DAS?** 2020. Disponível em: <<https://dewesoft.com/br/aquisicao-de-dados/o-que-e-aquisicao-de-dados>>. Acesso em: 07 de julho de 2022. Citado na página 15.

TAVARES, I. S. **Relatório Final de Estágio Obrigatório**. 2021. Disponível em: <<https://drive.google.com/file/d/1pm4YBX7M-Du0bMCwW1S0VA5ubMThyOsJ/view>>. Acesso em: 18 de fevereiro de 2022. Citado na página 17.

TAVARES, I. S. **RSK Joystick Tester**. 2022. Disponível em: <<https://itextpdf.com/products/itextsharp>>. Acesso em: 10 de maio de 2022. Citado na página 34.

TERMINALS, A. **Our Terminal Itajaí**. 2022. Disponível em: <<https://www.apmterminals.com/en/itajai/about/our-terminal>>. Acesso em: 20 de fevereiro de 2022. Citado na página 1.

THIAGO, D. B. **Melhoria da performance nos processos de planejamento, programação e controle da manutenção dos equipamentos portuários: Reach Stacker e Empilhadeiras**. Julho 2013. 83 p. Dissertação (Doutorado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013. Citado na página 1.

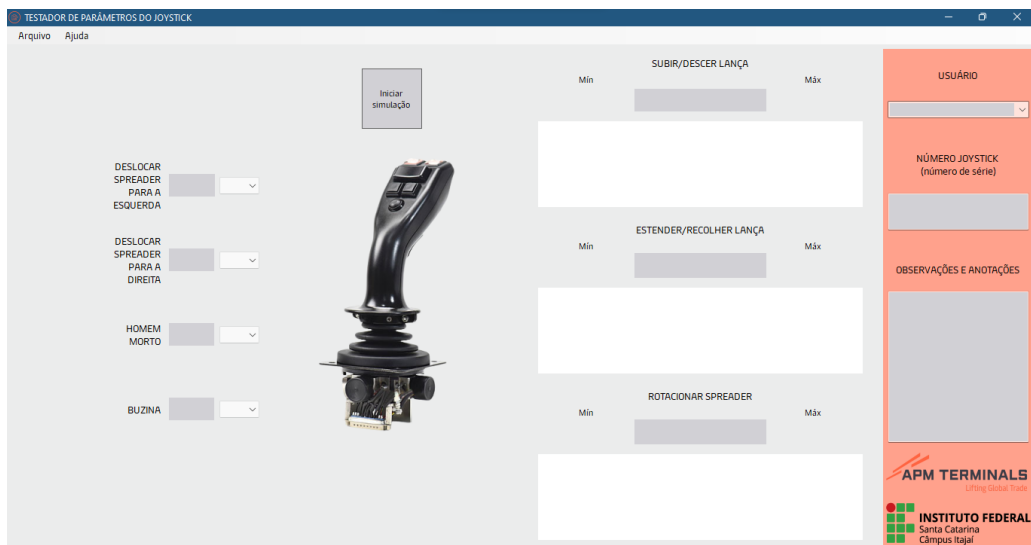
WERNER, C. et al. **Brechó: Catálogo de Componentes e Serviços de Software**. [s.n.], 2007. Disponível em: <<http://www.ic.uff.br/~leomurta/papers/werner2007.pdf>>. Citado na página 15.

APÊNDICE A – INSTRUÇÃO DE USO DO TESTADOR DE *JOYSTICK*

A.1 INFORMAÇÕES

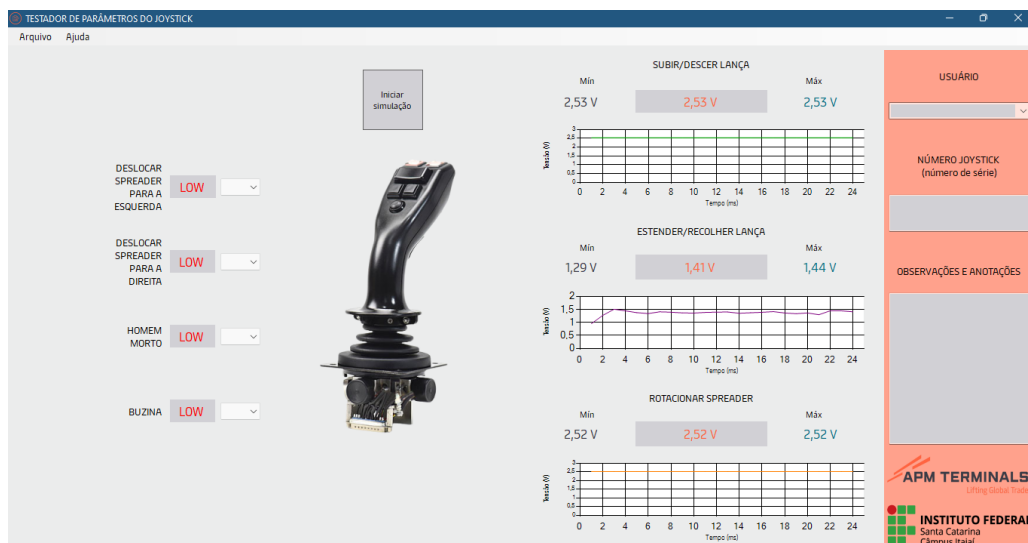
Aplicativo destinado à medição de parâmetros dos potenciômetros e botões digitais do *joystick* da máquina *reach stacker*. Na tela inicial do aplicativo (Figura 31), os botões digitais (buzina, deslocamento de *spreader*, homem morto) são mostrados à esquerda e os potenciômetros à direita (subir e descer lança, estender e recolher lança, girar *spreader*) e nos gráficos.

Figura 31 – Tela inicial do aplicativo.



Fonte: Autora.

Figura 32 – Tela do aplicativo após iniciar a simulação.

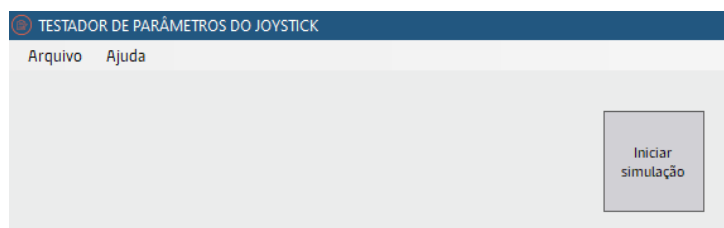


Fonte: Autora.

A.2 PASSOS PARA TESTE

1. Conectar o *joystick* no cabo com conector DB25 disponibilizado dentro da base de apoio;
2. Plugar o cabo USB no computador;
3. Abrir o aplicativo;
4. Clicar em “Iniciar simulação” (Figura 33).

Figura 33 – Botão de controle de simulação.

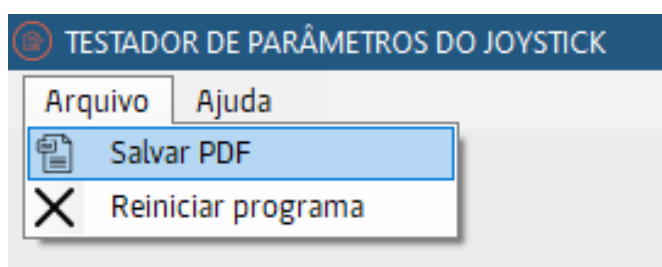


Fonte: Autora.

A.3 RELATÓRIO DE TESTES

1. Campos obrigatórios a serem preenchidos: usuário, número do *joystick* e condições dos botões digitais (OK ou NÃO OK);
2. Se desejar anotar alguma observação, incluir no campo “Observações”;
3. Para salvar o teste realizado, clicar em “Arquivo > Salvar PDF” (Figura 34).

Figura 34 – Caminho para salvar o relatório em PDF.



Fonte: Autora.

A.4 VALORES IDEAIS

Tabela 9 – Valores ideais das funções testadas.

	Valor mínimo	Valor de base	Valor máximo
Potenciômetros	0,4 a 0,6 [V]	2,4 a 2,6 [V]	4,4 a 4,6 [V]
Botões digitais	<i>LOW</i>	-	<i>HIGH</i>

Estado *LOW*: sem pressionar os botões;

Estado *HIGH*: pressionando os botões.

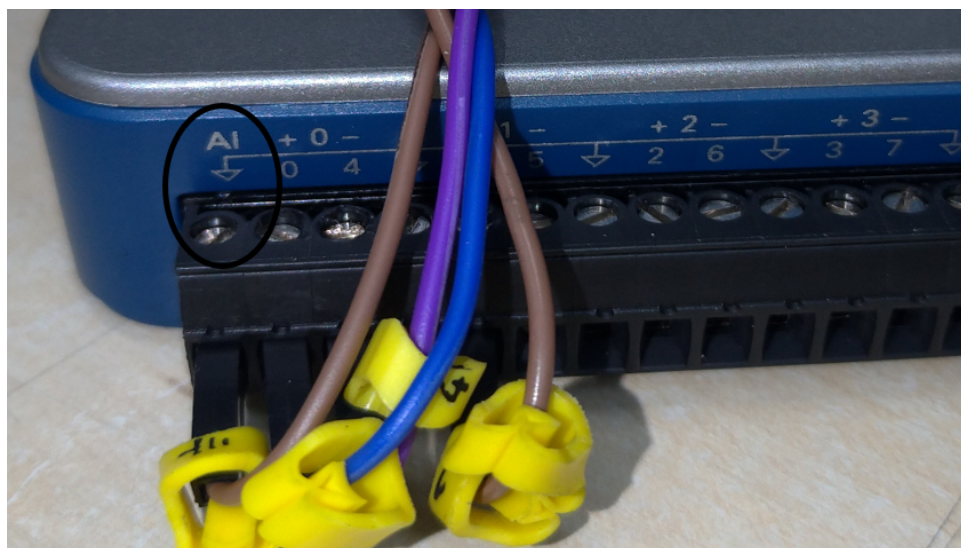
A.5 CONEXÃO ELÉTRICA DO TESTADOR

Tabela 10 – Instrução para conexão do conector DB25 e placa NI 6001.

Tipo de conexão	Função	Conector DB25	Placa NI 6001
Alimentação	-	Terminal 9	5 V
Alimentação	-	Terminal 15	5 V
Referência	-	Terminal 10	Seta à esquerda de 5 V e seta embaixo de AI1 (jump) (Figura 35)
Analógico	Recolhe/Estende lança	Terminal 04	AI4
Analógico	Sobe/Desce lança	Terminal 08	AI0
Analógico	Gira <i>spreader</i>	Terminal 11	AI1
Digital	Deslocamento esquerda	Terminal 14	P0.x 0
Digital	Deslocamento direita	Terminal 16	P0.x 1
Digital	Homem morto	Terminal 19	P0.x 2
Digital	Buzina/Acoplamento	Terminal 20	P0.x 3

Fonte: Autora.

Figura 35 – Seta indicativa de referência.



Fonte: Autora.

APÊNDICE B – RELATÓRIO DE TESTE AUTOMATIZADO

B.1 INFORMAÇÕES GERAIS

Responsável pelo teste: Isabelli Sasdelli Tavares;

Número do *joystick*: 3;

Data do teste: 10/05/2022 14:20.

B.2 RESULTADO DO TESTE

B.2.1 Botões digitais

Deslocamento para esquerda: OK;

Deslocamento para direita: OK;

Homem morto: OK;

Buzina: OK.

B.2.2 Potenciômetros

Tabela 11 – Resultado dos potenciômetros testados.

	Valor mínimo [V]	Valor de base [V]	Valor máximo [V]
Subir e descer lança	0,34	0,93	1,50
Estender e recolher lança	0,14	0,84	1,45
Rotacionar <i>spreader</i>	0,54	2,50	4,48

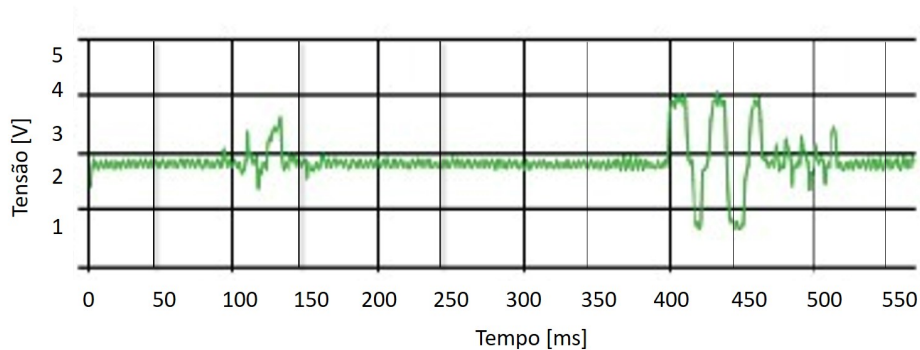
Fonte: Autora.

B.3 OBSERVAÇÕES DO TESTE

- Função de subir/descer lança: ruim;
- Função de estender/recolher lança: ruim;
- Função de rotacionar *spreader*: OK;
- Botões digitais: OK.

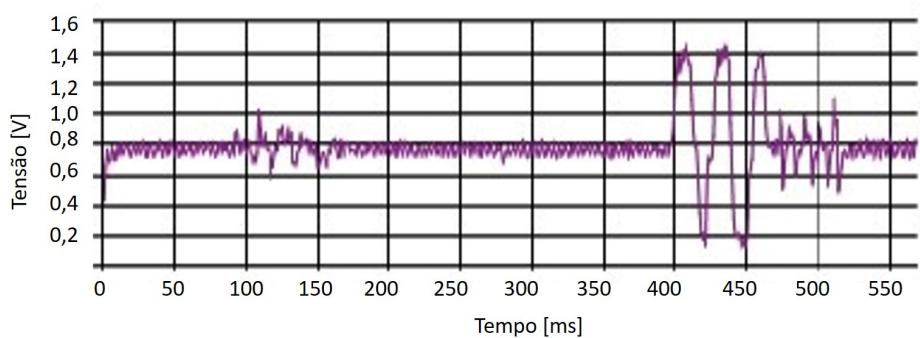
B.4 GRÁFICOS DOS POTENCIÔMETROS

Figura 36 – Gráfico da função de subir e descer lança.



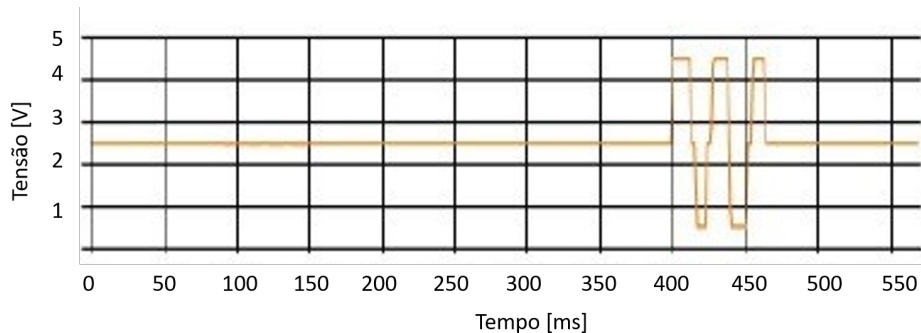
Fonte: Autora.

Figura 37 – Gráfico da função de estender e recolher lança.



Fonte: Autora.

Figura 38 – Gráfico da função de girar *spreader*.



Fonte: Autora.

ANEXO A – DATASHEET DO JOYSTICK

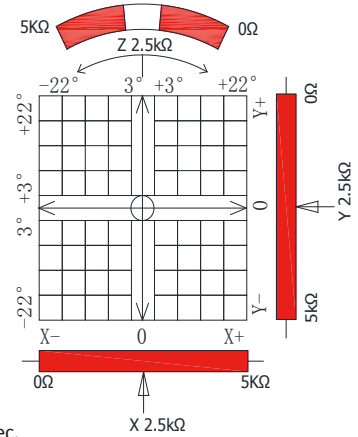
RunnTech Electronics (Changzhou) Corp.

RT90 Series Multi-axis Cobra Shaped Knob Joystick

RT90 Series Multi-axis Cobra Shaped Knob Joystick Controller



- P/N: RT90JAM-4U-46R4GP (920943.0058);
- Three axis, omnidirectional movement;
- Self-centering operation;
- Operation force: 5N;
- Operation temperature: -20...80°C;
- Analog output:
- Input voltage: 5Vdc
- Output voltage: 0.4...2.5...4.6Vdc
- Resistance: 5KΩ
- Resistance tolerance: ±5%
- Independent linear tolerance: ±1%, ±0.5%
- Resolution ratio: infinite
- Temperature coefficient: ±400 ppm/°C
- X/Y axis deflection angle: 32°
- Insulated resistance: 50 MΩ@ 500 VDC, 60 sec.



Wire diagram

