

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

BRENO AMARAL E CABRAL

**BRES: Modelagem, simulação e construção de um braço robótico
SCARA para ensino de cinemática**

FLORIANÓPOLIS, 2024.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

BRENO AMARAL E CABRAL

**BRES: Modelagem, simulação e construção de um braço robótico
SCARA para ensino de cinemática**

Trabalho de Conclusão de Curso
submetido ao Instituto Federal de
Educação, Ciência e Tecnologia de Santa
Catarina como parte dos requisitos para
obtenção do título de Engenheiro
Mecatrônico.

Orientador:

Prof. Dr. Eng. Francisco Rafael Moreira da
Mota

Coorientador:

Prof. Dr. Eng. Aurelio da Costa Sabino
Netto

FLORIANÓPOLIS, 2024.

Ficha de identificação da obra elaborada pelo autor.

Cabral, Breno

BRES: Modelagem, simulação e construção de um braço robótico SCARA para ensino de cinemática / Breno Cabral; orientação de Francisco Da Mota; coorientação de Aurelio Netto. - Florianópolis, SC, 2024.

46 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Santa Catarina, Câmpus Florianópolis. Bacharelado em Engenharia Mecatrônica. Departamento Acadêmico de Metal Mecânica.
Inclui Referências.

1. Cinemática. 2. Robótica educacional. 3. ROS2.
4. RaspBerry Pi 4. 5. STEM. I. Da Mota, Francisco. II. Netto, Aurelio. III. Instituto Federal de Santa Catarina.
IV. BRES: Modelagem, simulação e construção de um braço robótico SCARA para ensino de cinemática .

BRES: MODELAGEM, SIMULAÇÃO E CONSTRUÇÃO DE UM BRAÇO ROBÓTICO SCARA PARA ENSINO DE CINEMÁTICA

BRENO AMARAL E CABRAL

Este trabalho foi julgado adequado para obtenção do título de bacharel em Engenharia Mecatrônica, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Florianópolis, 13 de Dezembro de 2023.

Banca Examinadora:

Documento assinado digitalmente
 FRANCISCO RAFAEL MOREIRA DA MOTA
Data: 30/07/2024 10:38:52-0300
Verifique em <https://validar.iti.gov.br>

Francisco Rafael Moreira da Mota, Dr. Eng.

Documento assinado digitalmente
 CASSIANO BONIN
Data: 05/09/2024 17:02:22-0300
Verifique em <https://validar.iti.gov.br>

Cassiano Bonin, Me. Eng.
IFSC

Documento assinado digitalmente
 VITOR FARIAS DE BORBA
Data: 30/07/2024 15:20:02-0300
Verifique em <https://validar.iti.gov.br>

Vitor Farias de Borba, Me. Eng.
IFSC

RESUMO

Este trabalho aborda a importância da robótica, especialmente no contexto do ensino de cinemática em engenharia mecatrônica, e propõe o desenvolvimento de um braço robótico educacional chamado BRES (Braço Robótico Educacional SCARA). O objetivo é criar um dispositivo que permita aos alunos aplicar conceitos de cinemática em um ambiente prático, usando o ROS2 (Robot Operating System 2). A crescente relevância da robótica tanto na indústria quanto na academia, justifica a inclusão de disciplinas específicas no curso de engenharia mecatrônica e do desenvolvimento de materiais didáticos para apoiar o ensino nestas disciplinas. O trabalho foi desenvolvido de forma a incluir a modelagem CAD do protótipo, o desenvolvimento da cinemática, a criação do diagrama elétrico/eletrônico usando o Raspberry Pi 4, e a integração com o ambiente ROS2 para simulação e controle. O trabalho também aborda a pesquisa pré-TCC realizada para orientar o desenvolvimento do BRES, com referências a outros projetos educacionais similares. A apresentação dos resultados inclui testes de medição do protótipo, destacando a capacidade do BRES de reproduzir posições específicas e discutindo as limitações encontradas, como imprecisões devido ao material utilizado na impressão 3D. Por fim, ressalta-se a importância da melhoria contínua do BRES, indicando a possibilidade de aprimorar a precisão do dispositivo, superar limitações, desenvolver com complexidade tanto a parte de cinemática voltada ao ensino quanto a interface com o aluno, abrindo espaço para ampliar o uso do dispositivo para outras pesquisas em diferentes áreas.

Palavras-chave: Cinemática. Robótica educacional. ROS2. RaspBerry Pi 4. STEM.

ABSTRACT

This work addresses the importance of robotics, especially in the context of teaching kinematics in mechatronics engineering, and proposes the development of an educational robotic arm called BRES (Educational SCARA Robotic Arm). The aim is to create a device that allows students to apply kinematics concepts in a practical environment, using ROS2 (Robot Operating System 2). The increasing relevance of robotics in both industry and academia justifies the inclusion of specific disciplines in the mechatronics engineering curriculum and the development of educational materials to support teaching in these disciplines. The work was developed to include CAD modeling of the prototype, kinematics development, creation of the electrical/electronic diagram using Raspberry Pi 4, and integration with the ROS2 environment for simulation and control. The work also addresses pre-thesis research conducted to guide the development of BRES, with references to other similar educational projects. The presentation of results includes prototype measurement tests, highlighting BRES's ability to reproduce specific positions and discussing limitations encountered, such as inaccuracies due to the material used in 3D printing. Finally, the importance of continuous improvement of BRES is emphasized, indicating the possibility of enhancing the device's precision, overcoming limitations, developing both the teaching-oriented kinematics part and the interface with the student with complexity, opening up space to expand the device's use for further research in different areas.

Keywords: Kinematics. Educational Robotics. ROS2. Raspberry Pi 4. STEM.

LISTA DE FIGURAS

Figura 1 – Estrutura cinemática BRES	13
Figura 2 – Matrizes cinemática direta BRES	14
Figura 3 – Tópicos ROS2	15
Figura 4 – ROS_DOMAIN_ID	16
Figura 5 – Visualização RVIZ	17
Figura 6 – Formulário de pesquisa	19
Figura 7 – BRACON	20
Figura 8 – iArm	20
Figura 9 – Laboratory robotic arm	21
Figura 10 – Casa da qualidade inicial	22
Figura 11 – Desenho CAD	23
Figura 12 – Sessões CAD	23
Figura 13 – Desenhos para ilustrar dimensões Estrutura cinemática BRES	24
Figura 14 – GPIOs Raspberry Pi 4 model B	25
Figura 15 – Terminais Minos e Talos	26
Figura 16 – Arquivo Minos launch	27
Figura 17 – Visualização urdf_launch	27
Figura 18 – MotorHandle	28
Figura 19 – MotorHandle funções rotacao e translacao	28
Figura 20 – MinimalSubscriber	29
Figura 21 – Configuração de instâncias MotorHandle	29
Figura 22 – MinimalSubscriber função listener_callback	30
Figura 23 – rqt_graph	31
Figura 24 – BRES recolhido e estendido	31
Figura 25 – BRES gabinete	32
Figura 26 – BRES software desenvolvido	32
Figura 27 – BRES na posição zero	33
Figura 28 – BRES na posição randômica	34
Figura 29 – Casa da qualidade final	35
Figura 30 – CAD peça que apresentou falha	36
Figura 31 – Reparo realizado na peça que apresentou falha	36

LISTA DE ABREVIATURAS E SIGLAS

ABP	Aprendizagem Baseada em Projetos
BRES	Braço Robótico Educacional Scara
CAD	Computer-Aided Design, ou Desenho Assistido por Computador
DDS	Data Distribution Service
DH	Denavit-Hartenberg
DNS	Domain Name System
GPIO	General Purpose Input/Output, ou Entradas e Saídas de Propósito Geral
ROS	Robot Operating System, ou Sistema Operacional Robótico
SCARA	Selective Compliance Assembly Robot Arm, ou Braço Robótico para Montagem de Conformidade Seletiva
STEM	Science, Technology, Engineering and Mathematics, ou Ciência, Tecnologia, Engenharia e Matemática
URDF	Unified Robot Description Format, ou Formato Unificado de Descrição de Robôs
XACRO	Macro XML

SUMÁRIO

1 INTRODUÇÃO.....	9
1.1 Justificativa.....	10
1.2 Objetivos.....	11
1.2.1 Objetivo Geral.....	12
1.2.2 Objetivos Específicos.....	12
2 FUNDAMENTAÇÃO TEÓRICA.....	13
2.1 Cinemática Direta SCARA.....	13
2.2 Ambiente ROS2.....	15
2.2.1 Nós, Tópicos, Inscrições e Publicações.....	15
2.2.2 Comunicação entre dispositivos.....	16
2.2.3 Simulação com URDF, RVIZ e RQT.....	17
3 METODOLOGIA.....	18
3.1 Pesquisa para Casa da Qualidade.....	18
3.2 Modelagem CAD.....	22
3.3 Diagrama elétrico/eletrônico.....	25
3.4 Estrutura ROS.....	26
3.4.1 Comunicação entre hardware.....	26
3.4.2 Minos.....	27
3.4.3 Talos.....	28
4 APRESENTAÇÃO DOS RESULTADOS.....	31
4.1 Análise e discussão dos resultados.....	33
4.2 Realimentando a casa da qualidade.....	35
4.3 Falha mecânica do protótipo.....	35
5 CONSIDERAÇÕES FINAIS.....	37

1 INTRODUÇÃO

A robótica, termo popularizado por Isaac Asimov, é a ciência que estuda as tecnologias associadas ao desenvolvimento e construção de robôs. Como citado por diversos autores, a robótica é uma excelente ferramenta para o ensino (CAÑAS et al., 2014; PLAZA et al., 2017; KARALEKAS et al., 2019). É um fato que o campo da robótica está crescendo tanto em uso industrial quanto acadêmico e está cada vez mais presente no dia a dia. Portanto, desenvolver profissionais qualificados nessa área é essencial para um curso como engenharia mecatrônica, pois um robô é capaz de abranger praticamente todas as disciplinas do curso e possibilitar a aplicação prática do conhecimento dos alunos.

Dada a importância dessa área para um engenheiro mecatrônico, disciplinas fundamentais de robótica e robótica industrial estão incluídas no currículo. Durante a disciplina de fundamentos de robótica, os alunos revisam conceitos de disciplinas anteriores, como álgebra linear, cálculo vetorial e geometria analítica. Esses conceitos são usados no ensino de cinemática direta e inversa. Como apontado por Napolitano (Napolitano, 2001), o ensino precisa integrar diferentes conhecimentos dos alunos para alcançar uma aprendizagem significativa em vez de mecânica. Para auxiliar nessa consolidação do conhecimento, este trabalho propõe o desenvolvimento de um dispositivo que possa ser configurado por um aluno, permitindo que ele veja onde os cálculos de cinemática que estudou podem ser aplicados em um robô real, sem a abstração inerente que um robô industrial possui.

BRES (Braço Robótico Educacional Scara) é o nome que será dado ao protótipo de dispositivo proposto. A ideia é que este trabalho desenvolva o BRES e a documentação necessária para sua utilização. Juntamente com o protótipo, uma simulação do mesmo deve ser criada tanto para auxiliar no desenvolvimento quanto para o uso do BRES. A principal tecnologia escolhida para realizar este protótipo foi o ROS2 (*Robot Operating System 2*), o qual integra todos os recursos necessários e tem se mostrado uma tecnologia ímpar no desenvolvimento da robótica em todo o mundo.

Como mostrado no trabalho do PAHC (PAHC, 2022), o modelo de funcionamento de um braço seguindo a cinemática SCARA (Selective Compliance Assembly Robot Arm) traz inúmeras vantagens sendo bastante utilizado na indústria. O estudo da cinemática do BRES possibilita um aprendizado abrangente, possuindo características únicas, como o uso de uma junta prismática conectada a juntas rotativas e uma relação trigonométrica bem explorada em artigos e documentações. Essas características permitem o desenvolvimento de diferentes conceitos para os alunos que vierem a utilizar o protótipo proposto; outro ponto é que a maioria dos trabalhos com propostas semelhantes buscam construir braços antropomórficos ou

braços que não utilizem juntas prismáticas deixando de lado o aprendizado que um robô SCARA pode oferecer.

O ambiente de desenvolvimento escolhido para aprimorar a aprendizagem proposta pelo BRES se trata do ROS2, este pode ser definido como um “sistema operacional” robótico de código aberto. No entanto, não é um sistema operacional convencional, mas o termo é usado devido às suas semelhanças com um sistema operacional convencional; ou seja, o ambiente ROS2 foi desenvolvido para suportar aplicações de robótica (assim como um sistema operacional suporta *software*) e controlar robôs (como o sistema operacional controla periféricos e gerencia *drivers*), mas não equivale a uma distribuição Linux, Mac ou Windows; ele requer um sistema operacional convencional na máquina onde será instalado para funcionar.

Por ser um *software* de código aberto, o ROS2 ganha vida por meio de sua comunidade, que está em constante crescimento. O ROS2 permite a padronização de projetos, facilita a troca de conhecimento e se concentra em aplicações. Uma aplicação construída no ambiente ROS2 tem um tempo de desenvolvimento mais curto, removendo a necessidade de se dedicar à construção de um ambiente complexo e único, facilitando a reutilização de métodos de desenvolvimento de robôs estabelecidos, possibilitando a criação de integrações inviáveis ou muito complexas de outra forma.

A comunidade ROS2 é uma das grandes vantagens de seu uso. Basicamente, todos os recursos do ROS2 são bem explicados na documentação (ros.org, 2022) e contêm exemplos de como aplicá-los. Esta documentação, juntamente com a capacidade de desenvolver rapidamente uma simulação e controle do BRES, foram os pontos que definiram a escolha do ROS2 neste trabalho.

1.1 Justificativa

A obtenção de materiais didáticos relacionados à robótica no Brasil apresenta desafios, uma vez que os principais centros de desenvolvimento dessas tecnologias estão nos Estados Unidos e na Europa, tornando esses materiais geralmente caros e, em muitos casos, limitados quanto à exploração de suas características internas. Diante desse cenário, várias iniciativas (CRUZ et al., 2022; GENIVAL et al., 2018; LUCIANO et al., 2020) têm surgido para atender à demanda do ensino brasileiro. Este trabalho, por sua vez, propõe o desenvolvimento de uma solução nesse contexto, com foco específico no ensino de cinemática, abrindo também possibilidades para atender demandas de pesquisa que possam surgir no IFSC (ou em outra instituição), como estudos em informática industrial, sistemas de visão computacional, inteligência artificial, entre outros.

Um braço robótico didático pode desempenhar um papel fundamental no curso de engenharia mecatrônica, proporcionando conhecimento e aprendizado aos alunos. A qualidade da formação oferecida pelo IFSC requer materiais didáticos igualmente qualificados. Durante o desenvolvimento deste projeto, tanto alunos quanto professores demonstraram interesse e apoiaram a construção do protótipo, oferecendo sugestões de melhorias e soluções para possíveis desafios encontrados ao criar o BRES.

Além de ser um protótipo funcional de um braço robótico que segue a cinemática de um robô SCARA, este trabalho utilizará o ROS2, uma ferramenta de robótica recente e atualizada e com uma comunidade em crescimento em todo o mundo. O ambiente ROS2 é amplamente explorado por alunos em diversas instituições de ensino e por empresas que realizam pesquisas em robótica, demonstrando ser uma ferramenta versátil de conhecimento. Para utilizar o BRES, o aluno deve compreender conceitos básicos desse ambiente, capacitando-o em uma tecnologia atual e com uma comunidade ativa, o que permite uma exploração independente do aluno, caso haja interesse. O ROS2 também oferece a possibilidade de simulação por meio do RViz e do Gazebo (programas utilizados dentro do ROS2). Essa simulação abre caminho para a criação de outros projetos de robótica, incentivando os alunos a explorarem essa área e atribuindo um nível profissional de utilização ao protótipo.

1.2 Objetivos

A eficiência da Aprendizagem Baseada em Projetos (ABP) pode ser comprovada por meio da aferição dos conceitos apreendidos durante as aulas práticas, na área das engenharias, onde se pode constatar que a aprendizagem é mais efetiva que nas aulas teóricas (MASSON, 2012).

O aprendizado mecânico é caracterizado por uma compreensão superficial na qual o aluno memoriza conceitos e os reproduz sem verdadeiro entendimento, carecendo da curiosidade necessária para explorar os temas em profundidade. Um protótipo voltado para o ensino pode transformar esse processo de aprendizagem mecânica em um aprendizado significativo, à medida que o aluno interage diretamente com as aplicações práticas das teorias discutidas em sala de aula. Este trabalho tem como objetivo estimular a curiosidade dos alunos na resolução de problemas práticos utilizando os princípios matemáticos e científicos estudados, contribuindo para a melhoria do ensino de robótica no Brasil.

1.2.1 Objetivo Geral

Utilizar tecnologias de código aberto e amplamente difundidas para modelar, simular e fabricar um protótipo de braço robótico SCARA aplicado ao ensino de robótica.

1.2.2 Objetivos Específicos

Este trabalho tem como objetivos específicos:

- a) desenhar em CAD a estrutura mecânica do protótipo e fabricar as peças projetadas por meio de impressão 3D e corte a *laser*;
- b) configurar ambiente de desenvolvimento integrando os *hardware* (Computador e Raspberry Pi 4) a ser utilizado com os *software* (Ubuntu 22.04 e ROS2 humble);
- c) realizar os cálculos de cinemática direta necessária para desenvolver a simulação e controlar o protótipo;
- d) programar arquivo URDF (*Unified Robot Description Format*, arquivo contendo a descrição e cinemática do robô);
- e) utilizar as ferramentas RViz (responsável pela visualização do robô) e RQT (responsável pela interface com usuário) para gerar uma simulação do braço robótico, incluindo os arquivos CAD criados;
- f) comunicar o computador com a Raspberry Pi 4 possibilitando o envio de comandos e respostas;
- g) programar a Raspberry Pi 4 para controlar as GPIOs (Entradas e Saídas de Propósito Geral) comandando os *drivers* dos motores;
- h) montar e testar os componentes mecânicos e elétricos do protótipo;
- i) criar documentação e disponibilizar código do projeto necessário para manuseio do BRES.

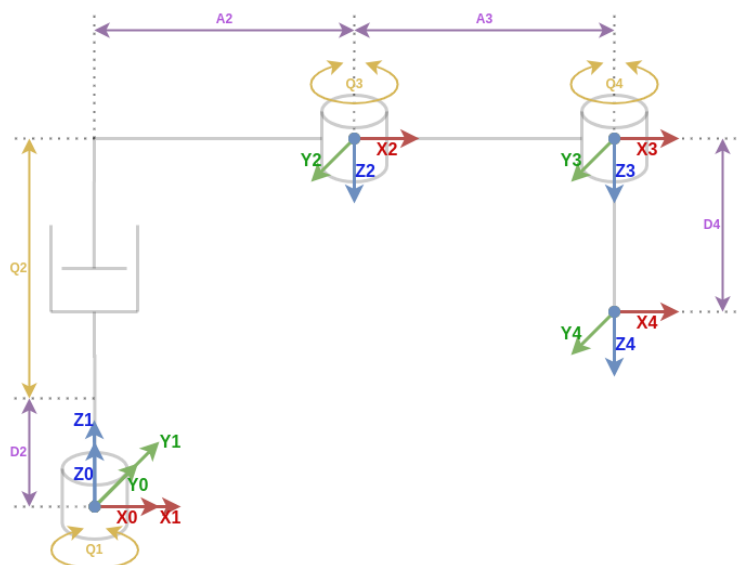
2 FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento do BRES envolve a criação de um protótipo operacional, cuja concretização demandou não apenas a utilização das ferramentas do ambiente ROS2 para o desenvolvimento de sistemas de controle e simulação CAD, mas também a integração de conhecimentos esperados de um engenheiro mecatrônico, como metodologias de desenvolvimento de produtos (casa da qualidade) e conceitos de robótica (cinemática direta e o *design* SCARA). Além da familiarização com a plataforma ROS2, fundamental para o controle e simulação do braço robótico, faz-se necessário compreender a cinemática para a correta modelagem dos movimentos do dispositivo. A utilização do *design* SCARA, com suas características específicas de movimento e estrutura, implicou em personalizações tanto no *software* quanto no *hardware* do protótipo, exigindo uma análise cuidadosa das interações entre seus componentes. Ademais, o emprego do Raspberry Pi 4 como controlador principal demanda uma profunda compreensão de suas capacidades e limitações, bem como a implementação de estratégias de comunicação e processamento de dados, utilizando as ferramentas do ROS2.

2.1 Cinemática Direta SCARA

Os robôs SCARA podem possuir a configuração cinemática apresentada na Figura 1, contando com 4 graus de liberdade, onde a junta 1 e 3 são responsáveis pelas posições X e Y da ferramenta, a junta 2 pela posição Z e a junta 4 pela orientação em torno do eixo Z. Dessa forma o robô terá capacidade de se movimentar nos três eixos (X, Y, Z) e de se orientar em torno do eixo Z.

Figura 1 – Estrutura cinemática BRES



Fonte: Autor, 2023.

Para manipular o robô, é necessário algum algoritmo de cinemática sendo o de DH (Denavit-Hartenberg) estudado na disciplina de Fundamentos de Robótica. A configuração de juntas escolhidas se trata de rotativa, prismática, rotativa e rotativa para atribuir os 4 graus de liberdade esperados. Aplicando o algoritmo de DH, determinam-se as matrizes demonstradas na Figura 2 e possibilita o cálculo da matriz T; esta matriz corresponde a cinemática direta da configuração mencionada anteriormente.

Figura 2 – Matrizes cinemática direta BRES

$$\begin{aligned}
 {}^0\mathbf{A}_1 &= \begin{bmatrix} C(Q1) & -S(Q1) & 0 & 0 \\ S(Q1) & C(Q1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^1\mathbf{A}_2 &= \begin{bmatrix} 1 & 0 & 0 & A2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & D2+Q2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^2\mathbf{A}_3 &= \begin{bmatrix} C(Q3) & -S(Q3) & 0 & A3xC(Q3) \\ S(Q3) & C(Q3) & 0 & A3xS(Q3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^3\mathbf{A}_4 &= \begin{bmatrix} C(Q4) & -S(Q4) & 0 & 0 \\ S(Q4) & C(Q4) & 0 & 0 \\ 0 & 0 & 1 & D4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^0\mathbf{T}_4 &= \begin{bmatrix} C(Q3-Q1+Q4) & -S(Q3-Q1+Q4) & 0 & A2*C(Q1)+A3*C(Q1-Q3) \\ -S(Q3-Q1+Q4) & -C(Q3-Q1+Q4) & 0 & A2*S(Q1)+A3*S(Q1-Q3) \\ 0 & 0 & -1 & D2-D4+Q2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Fonte: Autor, 2023.

2.2 Ambiente ROS2

O ROS2 funciona como um *framework*, criando um ambiente de desenvolvimento o qual fornece recursos necessários para a criação de *softwares* para robôs; uma aplicação ROS2 é chamada de pacote.

2.2.1 Nós, Tópicos, Inscrições e Publicações

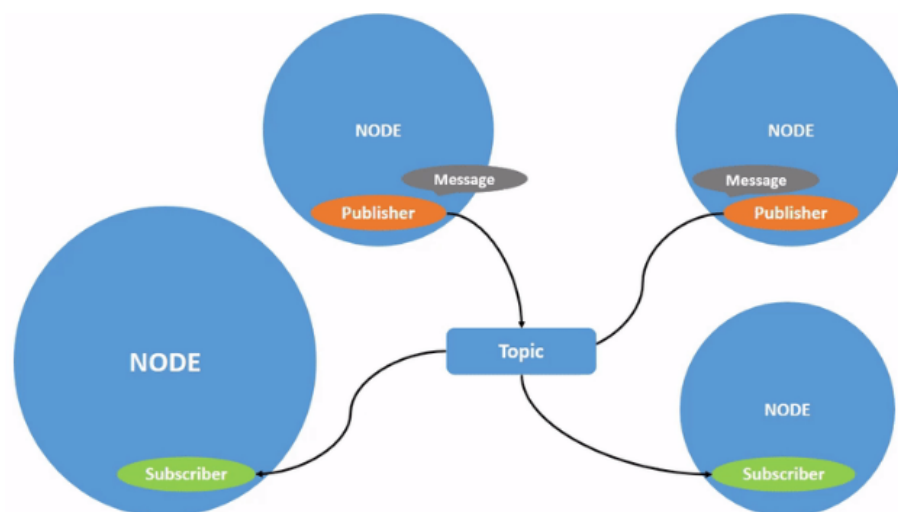
Nós de processamento (ou simplesmente **nós**) são os programas escritos em C++ ou Python que executam algum processamento no ambiente ROS2, o controle de um dispositivo via ROS2 se dará pela programação de diferentes **nós** e das relações entre os mesmos.

O ROS2 utiliza os **tópicos** como seu principal recurso de troca de informações simples entre diferentes **nós**. Cada **nó** pode interagir com os **tópicos** através de duas maneiras: "publicação" e/ou "inscrição". Um tópico pode ter um ou vários **nós** inscritos ou publicando mensagens nele, todos **nós** inscritos recebem todas mensagens publicadas no tópico.

Em termos mais simples, imagine que os tópicos são como canais de comunicação, onde os **nós** podem enviar e receber mensagens. Se um **nó** está interessado em receber mensagens sobre um determinado assunto, ele se "inscreve" no **tópico** relevante. Se um **nó** deseja enviar mensagens sobre um determinado assunto, ele "publica" essas mensagens no **tópico** apropriado.

Para mais detalhes, consultar a documentação oficial do ROS2 (ROS 2 Documentation: Humble, 2023), especialmente a seção de conceitos, que explica os **tópicos** de forma mais detalhada. A Figura 3, foi retirada desta documentação.

Figura 3 – Tópicos ROS2



Fonte: ROS 2 Documentation: Humble (2023).

2.2.2 Comunicação entre dispositivos

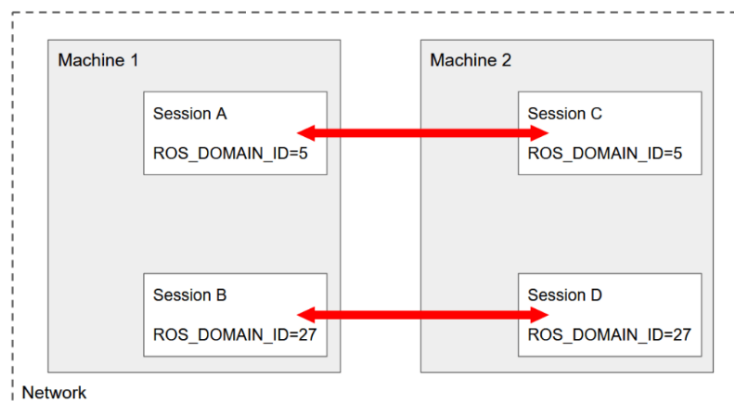
Uma das modificações trazidas pelo ROS2 em comparação com o ROS1 está em como os **nós** localizam outros **nós**. Enquanto no ROS1 existe o *roscore*, um **nó** responsável por armazenar a localização dos outros **nós** funcionando como um servidor DNS, em ROS2 não se utiliza o *roscore* por ser considerado um ponto de falha único e trazer desvantagens no desenvolvimento de aplicações.

No lugar dessa abordagem, o ROS2 se utiliza do protocolo DDS, em uma explicação simplista significa que os **nós** em ROS2 se encontram automaticamente desde que estejam na mesma rede ethernet (não é necessário conexão com a internet) e usando a mesma variável de ambiente **ROS_DOMAIN_ID**.

Através da documentação oficial e de dois artigos, ROS2 vs. ROS1 (Omar Elmofty, 2022) e ROS2 Multiple Machines Tutorial (The Robotic Backend, 2023), torna-se claro como funciona a comunicação entre dispositivos através do recurso de mensagens no ROS2.

A Figura 4 representa o funcionamento do **ROS_DOMAIN_ID**, demonstrado como duas máquinas “Machine 1” e “Machine 2” virtualmente conectadas na mesma rede ethernet “Network” possuem diferentes aplicações ROS2. Note que, a aplicação “Session A” em “Machine 1” utiliza o valor 5 na variável de ambiente **ROS_DOMAIN_ID**, o mesmo valor é utilizado na “Session C” em “Machine 2”, dessa forma as duas aplicações ROS2 “Session A” e “Session C”, compartilham acesso aos mesmos **nós** e **tópicos**, possibilitando que duas máquinas diferentes se comuniquem desde que estejam conectadas à mesma rede ethernet.

Figura 4 – ROS_DOMAIN_ID



Fonte: The Robotic Backend, 2023.

2.2.3 Simulação com URDF, RVIZ e RQT

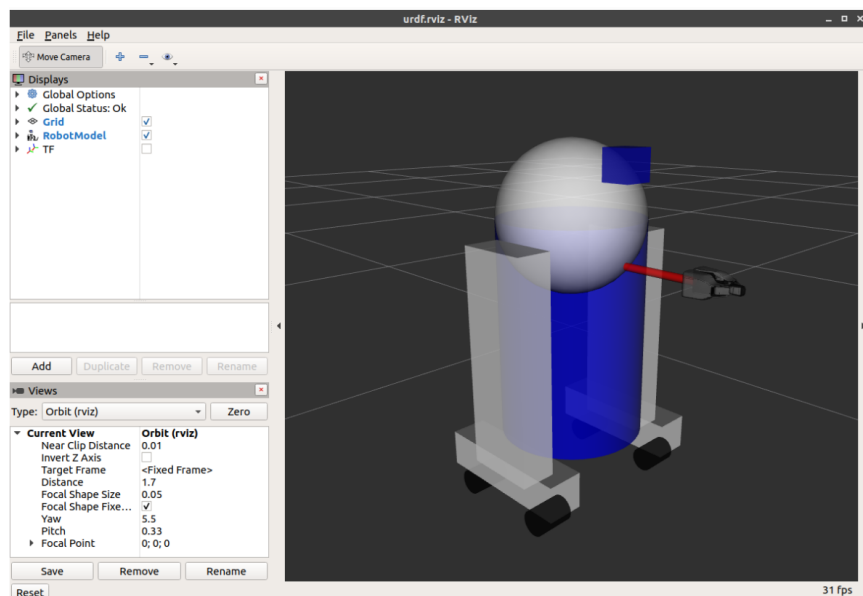
Para criar uma simulação computacional do robô, o ROS2 fornece algumas ferramentas, citando as principais a seguir.

O URDF se trata de um arquivo contendo a descrição do robô, as dimensões e como os elos e juntas se conectam, dessa forma contém a representação cinemática do robô. A URDF recebeu uma implementação de macros chamada XACRO, a qual permite reutilizar valores em diferentes partes do código URDF, em programação e equivalente à implementação de constantes.

O RVIZ se trata do visualizador de tf (este é o nome que o ROS2 utiliza para as funções de transformadas relacionadas diretamente com a cinemática), basicamente é um programa que permite ver o modelo cinemático do robô de forma dinâmica.

Já o RQT é uma ferramenta ROS2 para utilização do QT (*framework* para desenvolvimento de interfaces, muito utilizado em distribuições linux), e permite a criação das interfaces com o usuário. O RVIZ junto ao RQT permite uma simulação interativa como mostrada na Figura 5.

Figura 5 – Visualização RVIZ



Fonte: ROS 2 Documentation: Humble (2023).

3 METODOLOGIA

Neste capítulo são apresentados as ferramentas e procedimentos metodológicos utilizados para desenvolver a parte mecânica do protótipo, realizar os cálculos de sua cinemática, fazer um esquemático elétrico/eletrônico, utilizar do ambiente ROS2 para simular como o robô se comportaria, comunicar o computador com a raspberry, programar a raspberry para controle dos *drivers* e, por fim, fazer com que o BRES copie o posicionamento da simulação.

O BRES utilizou a versão ROS2 humble, sendo esta a versão *long term support* (a versão com suporte de longo prazo) disponível durante o desenvolvimento. O ROS2 humble possui suporte Tier 1 (prioridade de suporte) para o Ubuntu 22.04 sendo assim, essa foi a distribuição Linux escolhida tanto para o computador quanto para a Raspberry Pi 4 utilizados. O computador recebeu o Ubuntu 22.04 desktop com o ROS2 humble full enquanto a Raspberry Pi 4 recebeu o Ubuntu 22.04 server com o ROS2 humble base.

3.1 Pesquisa para Casa da Qualidade

A Casa da Qualidade é uma das ferramentas da qualidade criada na década de 60 pelo japonês Yoji Akao. Essencialmente, faz com que as empresas integrem a Voz do Cliente no desenvolvimento de produtos. Essa metodologia oferece diversos benefícios como entender os clientes/usuários, usar as necessidades do cliente/usuário para desenvolver metas, documentar requisitos e priorizar recursos (Lucidchart, 2023).

Uma pesquisa guiou o desenvolvimento inicial do BRES, sendo esta, composta por um formulário (Figura 6) distribuído aos alunos da engenharia mecatrônica, que cursaram a disciplina de fundamentos de robótica. O formulário questionava a importância de aspectos de um robô educacional para ensino de cinemática.

Figura 6 – Formulário de pesquisa

Pela sua opinião, classifique o grau de importância dos requisitos de um braço robótico voltado ao ensino de cinemática.

	0 - Irrelevante	1 - Não importa tanto	2 - É bom se tiver	3 - É esperado que tenha	4 - Precisa funcionar bem	5 - É fundamental para utilização
Abstração do software na hora da utilização	☐	☐	☐	☐	☐	☐
Movimentação precisa	☐	☐	☐	☐	☐	☐
Possibilidade de implementar recursos ao protótipo como câmeras ou sensores	☐	☐	☐	☐	☐	☐
Acesso ao código e a documentação técnica sobre o braço	☐	☐	☐	☐	☐	☐
Facilidade de utilização	☐	☐	☐	☐	☐	☐
Simulação do braço robótico	☐	☐	☐	☐	☐	☐
Interface do usuário	☐	☐	☐	☐	☐	☐

Fonte: Autor, 2023.

Pelas respostas obtidas, iniciou-se o desenvolvimento de uma casa da qualidade para avaliar especificações e requisitos que o BRES se propõe a atender. Posteriormente, foi feita a busca por projetos com propostas similares, a fim de encontrar “competidores” do projeto que possam ser usados para comparação. Os modelos a seguir (Figuras 7 a 9) foram os escolhidos para nortear o desenvolvimento do protótipo.

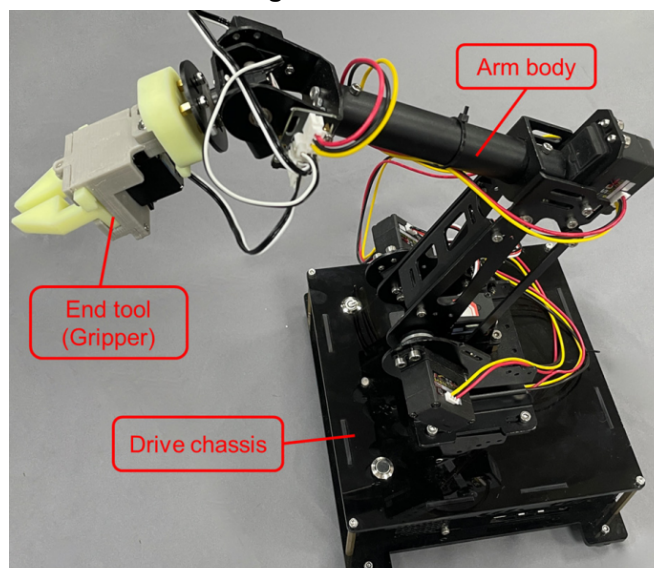
Figura 7 - BRACON



Fonte: RIVAS, 2015.

O BRACON foi um braço robótico de baixo custo utilizado pela *Universidad de las Fuerzas Armadas ESPE* no artigo “BRACON: control system for a robotic arm with 6 degrees of freedom for education systems”. Possui 6 graus de liberdade, formando uma configuração antropomórfica, utiliza python para criação do seu *software* e controla as juntas por meio de servomotores *Dynamixel* (uma marca que produz atuadores voltados à robótica), BRACON se utiliza de um *hardware* comercial voltado a robótica junto de chapas metálicas na estrutura se concentrando na lógica de funcionamento do *software*.

Figura 8 - iArm

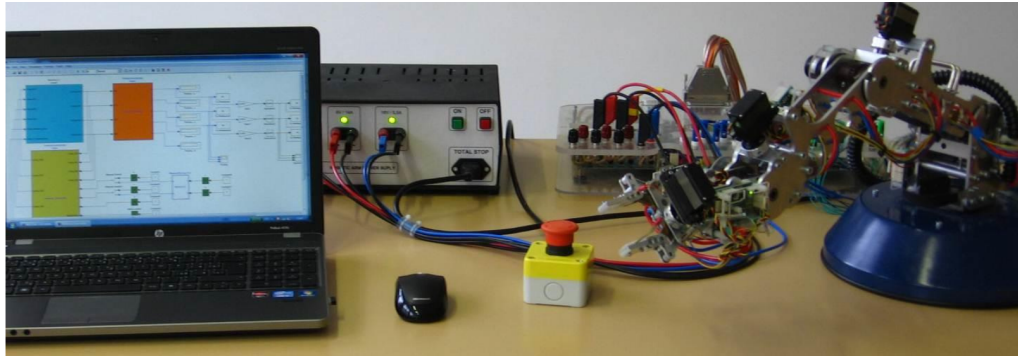


Fonte: ZENG, 2022.

O iArm foi um braço robótico de baixo custo utilizado pela *East China Normal University* no artigo “iArm: Design an Educational Robotic Arm Kit for Inspiring Students’ Computational Thinking”. Foi encontrado o mesmo braço, chamado por

AL5B, no artigo “Modeling and Simulation of 5 DOF Educational Robot Arm”, ambos os braços serão tratados como um, a fim de obter comparações válidas de *software*, *hardware* e funcionalidades. Possui 6 graus de liberdade, formando uma configuração antropomórfica, se utiliza de servomotores simples para as juntas e para o *software* usa uma integração do Autodesk Inventor (*software* CAD, usado por AL5B) com o Matlab por meio da biblioteca CAD2Matlab.

Figura 9 - Laboratory robotic arm



Fonte: KRASŇANSKÝ, 2013.

O Laboratory robotic arm foi um braço robótico de baixo custo utilizado pela *Slovak University of Technology* no artigo “Basic Laboratory Experiments with an Educational Robotic Arm”. Possui 6 graus de liberdade, formando uma configuração antropomórfica, servomotores simples acoplados à conjuntos de engrenagens formam uma mecânica robusta, para o *software*, foi utilizado o Matlab junto ao Simulink criando a interface com o usuário, o braço foi pensado para uso de módulos de comunicação industriais, os quais permitem resposta de tempo real e trazem a confiabilidade exigida pela indústria, porém o trabalho realizado no artigo utilizou-se de um módulo próprio desenvolvido para prova de conceito do robô.

A partir dos dados coletados, a seguinte casa da qualidade (Figura 10) foi montada, a qual orientou o desenvolvimento do BRES. A partir dela obtemos o BRES ideal, o qual teria todos os pontos positivos dos projetos similares, e a importância de cada requisito e especificação para gerar valor aos alunos que utilizarem o projeto, temos que o foco de desenvolvimento deve se concentrar no sistema operacional e interface com usuário seguido de uma boa simulação e documentação do robô.

Figura 10 – Casa da qualidade inicial

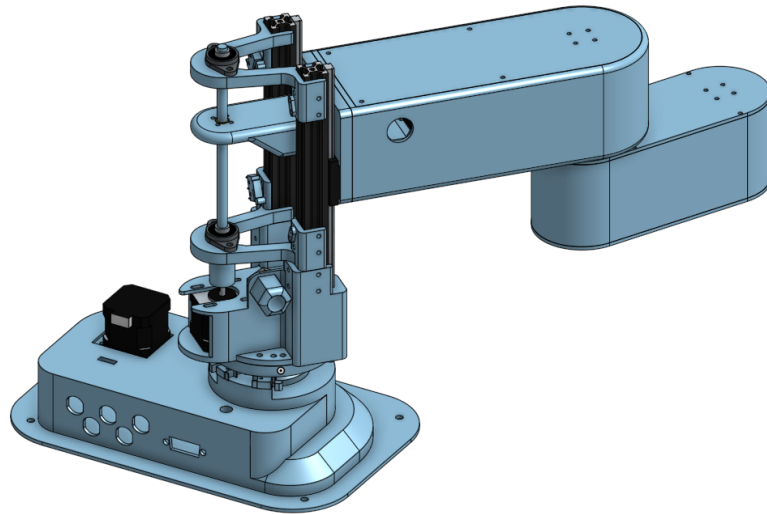
				ESPECIFICAÇÕES								SIMILARES			
				▼	▼	▼	▲	▲	▲	▲					
				Peso	Tamanho	Custo	Sistema Operacional	Documentação	Interface	Simulação		BRACON	iArm	Laboratory Robotic Arm	BRES Ideal
		Importancia Absoluta Para Alunos (Segundo Formulário)	Importancia Relativa Para Alunos												
		5	20,83%	♙	♙		♚	♙	♚	♙		2	4	2	4
REQUISITOS	Facilidade de utilização	4	16,70%				♚	♙	♚	♙		3	5	2	5
	Interface do usuário	3	12,50%	♙	♙	♚			♙			5	5	5	5
	Movimentação precisa	2	8,30%	♙		♙	♚	♙	♙			3	5	4	5
	Possibilidade de implementar recursos ao protótipo como câmeras ou sensores	4	16,70%				♚		♙	♚		3	0	3	3
	Simulação do braço robótico	4	16,70%				♙	♚				1	3	3	3
	Acesso ao código e a documentação técnica sobre o braço	2	8,30%				♙		♚			1	4	1	4
	Abstração do software na hora da utilização														
Importancia Absoluta Para Projeto				26	24	33	145	57	116	55					
Importancia Relativa Para Projeto				5,70%	5,26%	7,24%	31,80%	12,50%	25,44%	12,06%					

Fonte: Autor, 2023.

3.2 Modelagem CAD

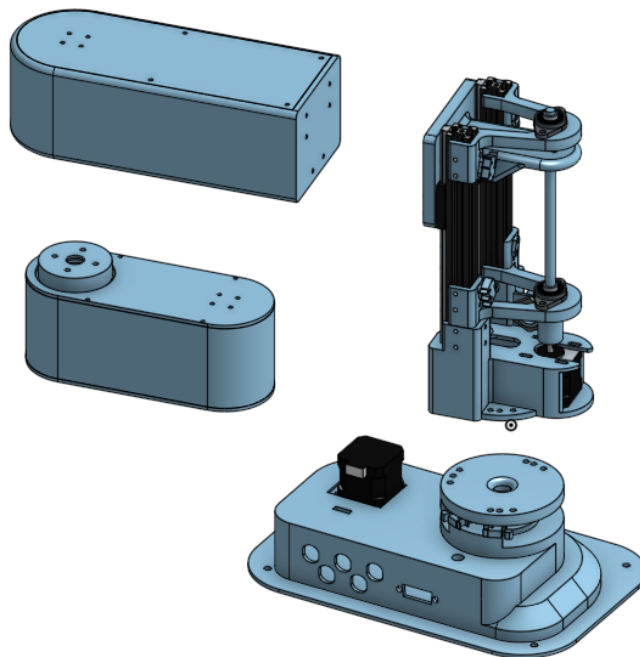
O projeto CAD foi realizado no *software onshape*. Este *software* permite a criação de CAD gratuitamente desde que não tenha uso comercial, além de funcionar de forma *web*, possibilitando que seja desenhado, modificado e visualizado em um sistema operacional Linux. A Figura 11 ilustra o modelo CAD desenvolvido, enquanto a Figura 12 demonstra o projeto dividido em 4 sessões, esta divisão busca facilitar o entendimento e montagem do BRES, o Apêndice B traz as vistas explodidas dessas sessões isoladas. As partes planas do CAD foram pensadas para serem cortadas a *laser* e partes de geometria complexa, para serem impressas em 3D.

Figura 11 – Desenho CAD



Fonte: Autor, 2023.

Figura 12 – Sessões CAD



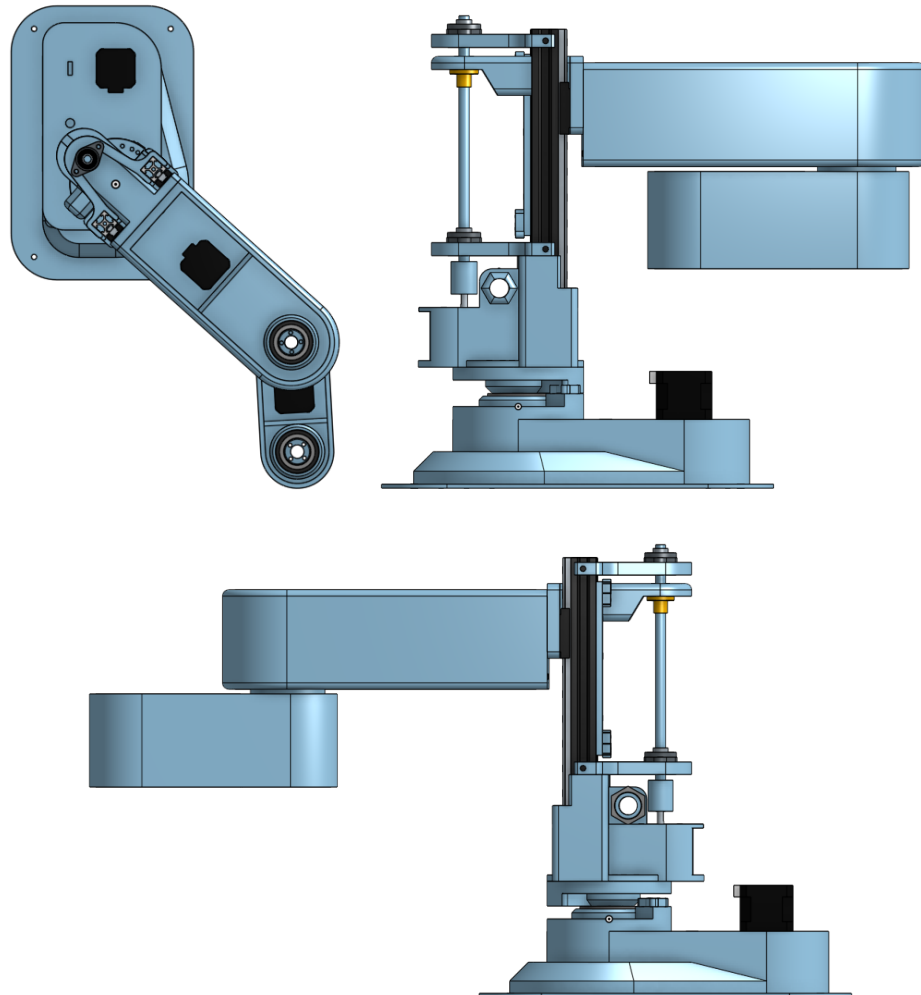
Fonte: Autor, 2023.

As correias utilizadas nas juntas rotativas possuem tamanho fixo de 350mm, 350mm e 220mm; a polia do motor com 20 dentes tem diâmetro de 12mm e a polia da junta com 80 dentes tem diâmetro de 50,50mm. Dessa forma, todas as juntas rotativas tiveram uma redução de $\frac{1}{4}$; já a junta prismática utilizou um sistema de fuso e castanha possuindo um fuso de passo 8mm.

Com essas medidas foi possível localizar as posições ideais dos motores no protótipo. O BRES (Figura 13) pode ficar recolhido possuindo um volume de

365x364x200mm³, sendo que quando na posição zero do braço o comprimento máximo é de 575mm; as juntas rotacionais possuem 270° livres para rotação e a junta prismática possui 110mm livre para deslocamento.

Figura 13 – Desenhos para ilustrar dimensões



Fonte: Autor, 2023.

Para utilizar a cinemática SCARA discutida na fundamentação teórica algumas medidas foram necessárias; as medidas foram retiradas do CAD e são apresentadas abaixo:

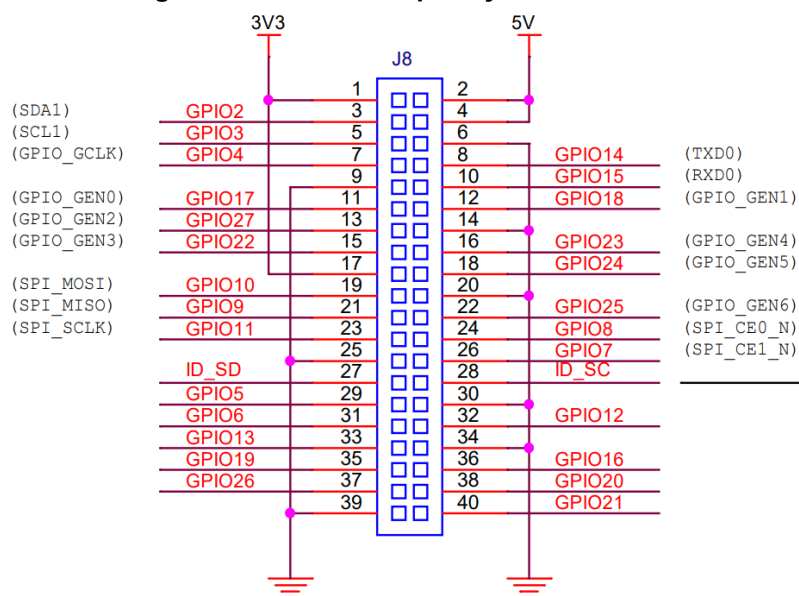
- A2 = 230mm;
- D2 = 150mm;
- A3 = 150mm;
- D4 = 100mm.

3.3 Diagrama elétrico/eletrônico

O Raspberry Pi é um micro-computador de placa única desenvolvido pela Fundação Raspberry Pi. Ele foi projetado para ser utilizado como ferramenta educacional para ensinar programação e computação em escolas. Mas também pode ser utilizado como um computador de baixo custo para outros fins, como media center, servidor web, controle de automações, etc. (Felipe Alencar, 2023). O Raspberry Pi 4B conta com processador Quad Core 1.8GHz 64-bit e 4GB RAM (Raspberry Pi, 2023).

Para a utilização das GPIOs do Raspberry Pi 4 de forma a controlar motores de passo, foi necessário o uso de *drivers* DRV8849 (Last Minute Engineers, 2023), e de placas de expansão (Joshua Hrisiko, 2021) facilitando as conexões eletrônicas. O *datasheet* da Raspberry Pi 4 foi consultado localizando cada pino do barramento de GPIO (Figura 14). Neste ponto o *software* Qeletrotech, recomendado na disciplina de PI V (Projeto Integrador V), foi utilizado para criação do diagrama elétrico/eletrônico (Apêndice A).

Figura 14 – GPIOs Raspberry Pi 4 model B



Fonte: Raspberry Pi, 2023.

O projeto elétrico/eletrônico é composto por um gabinete para armazenar e fornecer suporte mecânico, contendo os seguintes componentes:

- 1 x raspberry Pi 4;
- 4 x *drivers* DRV8825, configurados para 1A;
- 4 x placas de expansão, configuradas para ¼ de passo;
- 1 x fonte 24V@15A;
- 1 x módulo de 2 relés ;
- 1 x botão liga/desliga com led.

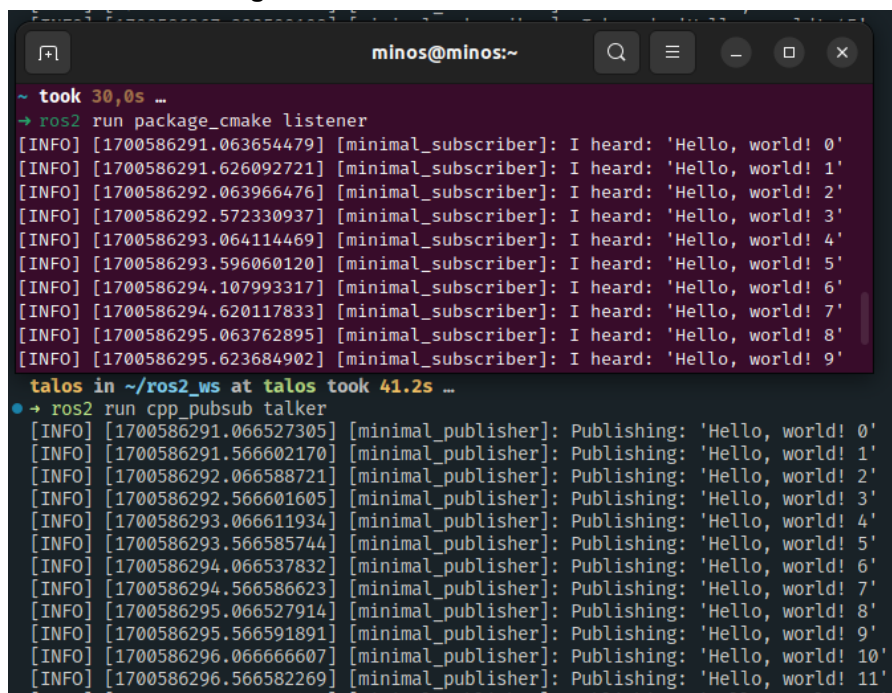
3.4 Estrutura ROS

Os recursos de *software* necessários para funcionamento do BRES foram avaliados separadamente para no final se integrarem e constituírem dois pacotes ROS2, sendo o primeiro uma aplicação dedicada ao computador que controla a simulação chamado de Minos; e o segundo corresponde a aplicação que controla as GPIOs da Raspberry Pi 4 chamado de Talos, o qual busca imitar a simulação. Ambos os pacotes podem ser encontrados no github em duas *branch*, minos e talos (<https://github.com/BreCabral/bres>).

3.4.1 Comunicação entre *hardware*

Como mostrado na fundamentação teórica, para a comunicação entre o Raspberry Pi 4 e o computador se faz necessário colocar os dois na mesma rede (foi utilizada uma rede wifi) e configurar o ROS_DOMAIN_ID = 5 em ambos ambientes. Com essa configuração de ambientes, a partir de dois **nós**, um inscridor e um publicador, a comunicação foi validada entre as máquinas. A Figura 15 ilustra a validação da comunicação entre os terminais Minos (computador) e Talos (Raspberry PI 4). Tempo real não é requisito para o BRES, por se tratar de um equipamento didático, porém o teste possibilita avaliar a diferença de tempo entre a mensagem postada pelo Minos e a leitura pelo Talos através do campo de *timestamp*, em segundos, ao lado direito de “[INFO]”.

Figura 15 – Terminais Minos e Talos



```
minos@minos:~$ ros2 run package_cmake listener
[INFO] [1700586291.063654479] [minimal_subscriber]: I heard: 'Hello, world! 0'
[INFO] [1700586291.626092721] [minimal_subscriber]: I heard: 'Hello, world! 1'
[INFO] [1700586292.063966476] [minimal_subscriber]: I heard: 'Hello, world! 2'
[INFO] [1700586292.572330937] [minimal_subscriber]: I heard: 'Hello, world! 3'
[INFO] [1700586293.064114469] [minimal_subscriber]: I heard: 'Hello, world! 4'
[INFO] [1700586293.596060120] [minimal_subscriber]: I heard: 'Hello, world! 5'
[INFO] [1700586294.107993317] [minimal_subscriber]: I heard: 'Hello, world! 6'
[INFO] [1700586294.620117833] [minimal_subscriber]: I heard: 'Hello, world! 7'
[INFO] [1700586295.063762895] [minimal_subscriber]: I heard: 'Hello, world! 8'
[INFO] [1700586295.623684902] [minimal_subscriber]: I heard: 'Hello, world! 9'

talos in ~/ros2_ws at talos took 41.2s ...
talos@talos:~/ros2_ws$ ros2 run cpp_pubsub talker
[INFO] [1700586291.066527305] [minimal_publisher]: Publishing: 'Hello, world! 0'
[INFO] [1700586291.566602170] [minimal_publisher]: Publishing: 'Hello, world! 1'
[INFO] [1700586292.066588721] [minimal_publisher]: Publishing: 'Hello, world! 2'
[INFO] [1700586292.566601605] [minimal_publisher]: Publishing: 'Hello, world! 3'
[INFO] [1700586293.066611934] [minimal_publisher]: Publishing: 'Hello, world! 4'
[INFO] [1700586293.566585744] [minimal_publisher]: Publishing: 'Hello, world! 5'
[INFO] [1700586294.066537832] [minimal_publisher]: Publishing: 'Hello, world! 6'
[INFO] [1700586294.566586623] [minimal_publisher]: Publishing: 'Hello, world! 7'
[INFO] [1700586295.066527914] [minimal_publisher]: Publishing: 'Hello, world! 8'
[INFO] [1700586295.566591891] [minimal_publisher]: Publishing: 'Hello, world! 9'
[INFO] [1700586296.066666607] [minimal_publisher]: Publishing: 'Hello, world! 10'
[INFO] [1700586296.566582269] [minimal_publisher]: Publishing: 'Hello, world! 11'
```

Fonte: Autor, 2023.

3.4.2 Minos

O pacote Minos ficou responsável pela interface com usuário, a qual se utilizou de outro pacote (urdf_launch), sendo este, disponibilizado pela comunidade ROS2 para a utilização de macros no arquivo URDF (chamadas de XACRO) e traz consigo **nós** responsáveis por uma interface pronta de controle para simulação dos robôs que o utilizarem. Essa interface chama-se “Joint State Publisher” e, basicamente, ela publica valores no tópico que a simulação busca os valores das juntas.

A Figura 16 ilustra como se utiliza o pacote urdf_launch, chamando-o por meio de um arquivo lançador. A Figura 17 ilustra como a interface ficou com a implementação do arquivo URDF do BRES.

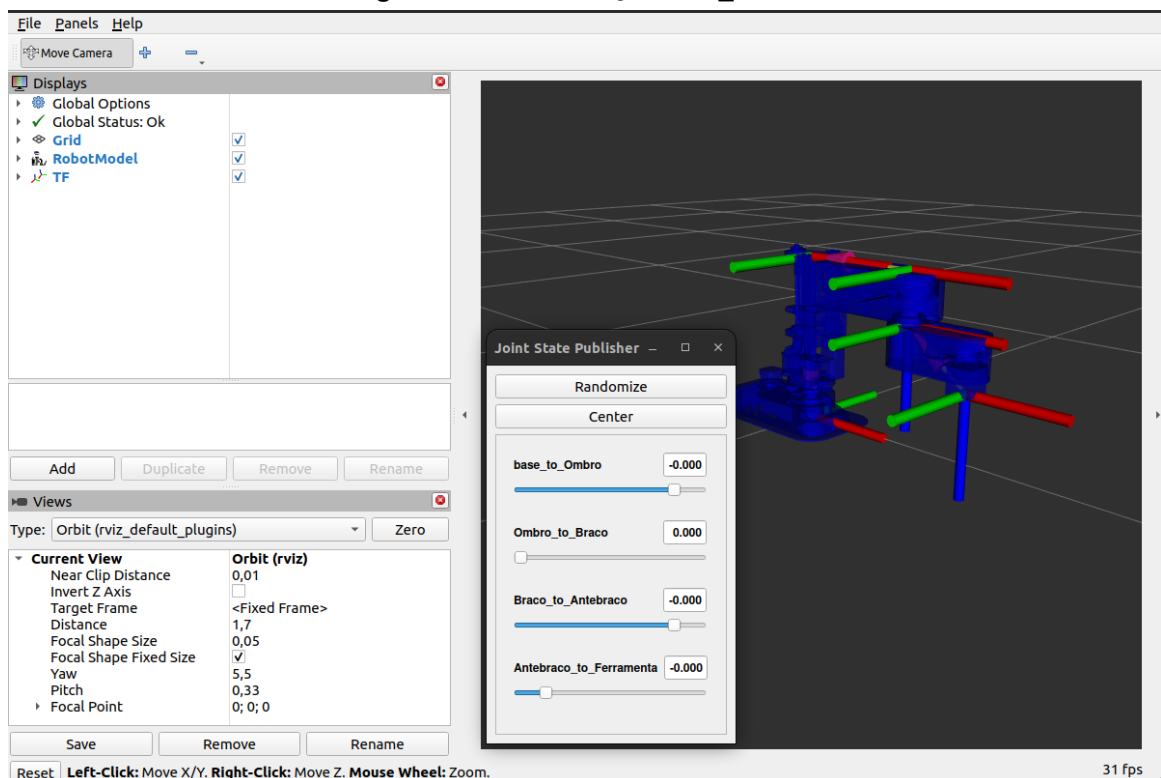
Figura 16 – Arquivo Minos launch

```
def generate_launch_description():
    ld = LaunchDescription()

    ld.add_action(IncludeLaunchDescription(
        PathJoinSubstitution([FindPackageShare('urdf_launch'), 'launch', 'display.launch.py']),
        launch_arguments={
            'urdf_package': 'bres',
            'urdf_package_path': PathJoinSubstitution(['urdf', 'bres.urdf.xacro']).items()
        })
    )
    return ld
```

Fonte: Autor, 2023.

Figura 17 – Visualização urdf_launch



Fonte: Autor, 2023.

3.4.3 Talos

Já o pacote Talos teve a responsabilidade de “ouvir” os comandos do Minos e controlar as GPIOs da Raspberry Pi 4 de forma a imitar a simulação. Para realizar o movimento dos motores, foram utilizadas duas bibliotecas python: a RPi.GPIO (Cleiton Bueno, 2014) e a RpiMotorLib; a primeira para controlar diretamente GPIOs e a segunda para controlar os *drivers* de motores de passo.

Para facilitar o uso da biblioteca RpiMotorLib, uma classe chamada MotorHandle foi criada, agindo como uma camada de abstração. Essa classe é capaz de movimentar um motor escolhido com base em um ângulo ou deslocamento recebido. Na Figura 18, pode-se observar como uma instância dessa classe deve ser criada e, na Figura 19, ilustra as funções rotação e translação que de fato movimentam os motores.

Figura 18 – MotorHandle

```
class MotorHandle():  
  
    def __init__(self, motor=1, angulo_por_passo=0.45, deslocamento_por_passo=0.01):  
        self.angulo_por_passo = angulo_por_passo  
        self.deslocamento_por_passo = deslocamento_por_passo  
        self.step = m1_step  
        self.dir = m1_dir  
        self.conectarGPIO(motor)
```

Fonte: Autor, 2023.

Figura 19 – MotorHandle funções rotação e translação

```
def rotacao(self, angulo):  
    if (abs(angulo) == 0):  
        return 0  
    elif (angulo > 0):  
        direcao = True  
        mul_dir = 1  
    elif (angulo < 0):  
        direcao = False  
        mul_dir = -1  
    qtd_passos_mover = int(abs(angulo)/self.angulo_por_passo)  
    self.motor_conectado.motor_go(direcao,"1/4",qtd_passos_mover,0.005,False,0.05)  
    return (qtd_passos_mover*self.angulo_por_passo*mul_dir)  
  
def translacao(self, deslocamento):  
    if (deslocamento == 0):  
        return 0  
    elif (deslocamento > 0):  
        direcao = True  
    elif (deslocamento < 0):  
        direcao = False  
    qtd_passos_mover = int(abs(deslocamento)/self.deslocamento_por_passo)  
    self.motor_conectado.motor_go(direcao,"1/4",qtd_passos_mover,0.005,True,0.05)  
    return (qtd_passos_mover*self.deslocamento_por_passo)
```

Fonte: Autor, 2023.

Outra classe a MinimalSubscriber foi criada para lidar com a subscrição no tópico “joint_state” (Figura 20), o qual contém os valores que as juntas devem se posicionar, o mesmo tópico que a simulação e que a interface “Joint State Publisher” utilizam.

Figura 20 – MinimalSubscriber

```
class MinimalSubscriber(Node):  
  
    def __init__(self):  
        super().__init__('minimal_subscriber')  
        self.subscription = self.create_subscription(  
            JointState,  
            'joint_states',  
            self.listener_callback,  
            10)
```

Fonte: Autor, 2023.

Dentro dessa classe foi criada uma instância da classe MotorHandle (Figura 21) para cada motor utilizado. Lembrando que os *drivers* foram configurados para 1/4 de passo, que corresponde a uma rotação de 0,45° para cada pulso e as juntas rotativas possuem uma redução de 1/4. Dessa forma, os motores de juntas rotativas rotacionam 0,1125° a cada pulso no *driver*. Já para junta prismática, o *driver* também está configurado para 1/4 de passo e, dessa forma, para uma rotação de 360° do eixo do motor, o *drive* deve receber 800 pulsos. Como o fuso desloca 8mm a cada volta completa do eixo, a cada 800 pulsos no *driver* a castanha se desloca 8mm no eixo da junta, dessa forma, a cada pulso recebido o deslocamento no eixo Z é de 0,01mm.

Figura 21 – Configuração de instâncias MotorHandle

```
self.motor1 = MotorHandle(motor=1, angulo_por_passo=(0.45/4))  
self.motor2 = MotorHandle(motor=2, deslocamento_por_passo=(0.01))  
self.motor3 = MotorHandle(motor=3, angulo_por_passo=(0.45/4))  
self.motor4 = MotorHandle(motor=4, angulo_por_passo=(0.45/4))
```

Fonte: Autor, 2023.

Após a criação das instâncias, a função listener_callback (Figura 22) é chamada repetidamente pelo **nó** criado, recebendo a mensagem do tópico “joint_state”. Com isso os motores recebem os comandos para se posicionarem de acordo com os valores desejados (com a devida correção de radianos para graus e de metros para milímetros).

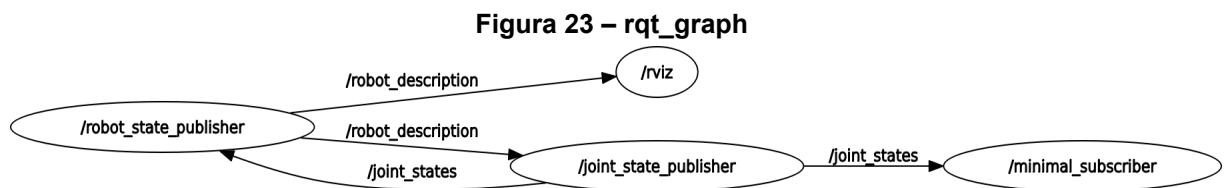
Figura 22 – MinimalSubscriber função listener_callback

```
def listener_callback(self, msg):  
  
    self.m1_pos_goal = self.rad_to_deg(-1*msg.position[0])  
    self.m2_pos_goal = self.m_to_mm(msg.position[1])  
    self.m3_pos_goal = self.rad_to_deg(msg.position[2])  
    self.m4_pos_goal = self.rad_to_deg(msg.position[3])  
  
    diferenca_m1 = (self.m1_pos_goal-self.m1_pos_atual)  
    diferenca_m2 = (self.m2_pos_goal-self.m2_pos_atual)  
    diferenca_m3 = (self.m3_pos_goal-self.m3_pos_atual)  
    diferenca_m4 = (self.m4_pos_goal-self.m4_pos_atual)  
  
    self.m1_pos_atual += self.motor1.rotacao(diferenca_m1)  
    self.m2_pos_atual += self.motor2.translacao(diferenca_m2)  
    self.m3_pos_atual += self.motor3.rotacao(diferenca_m3)  
    self.m4_pos_atual += self.motor4.rotacao(diferenca_m4)
```

Fonte: Autor, 2023.

4 APRESENTAÇÃO DOS RESULTADOS

Por meio de componentes adquiridos, impressões 3D e cortes a *laser*, um protótipo funcional foi confeccionado (com um custo em torno de R\$4000,00). Com o BRES em mãos e os pacotes *minos* e *talos* instalados no computador e raspberry para controle do mesmo, o trabalho seguiu para os testes e análises do projeto desenvolvido. Uma das ferramentas de análise do ambiente ROS2 é o *rqt_graph* o qual resultou na seguinte arquitetura de **nós**, Figura 23.



Fonte: Autor, 2023.

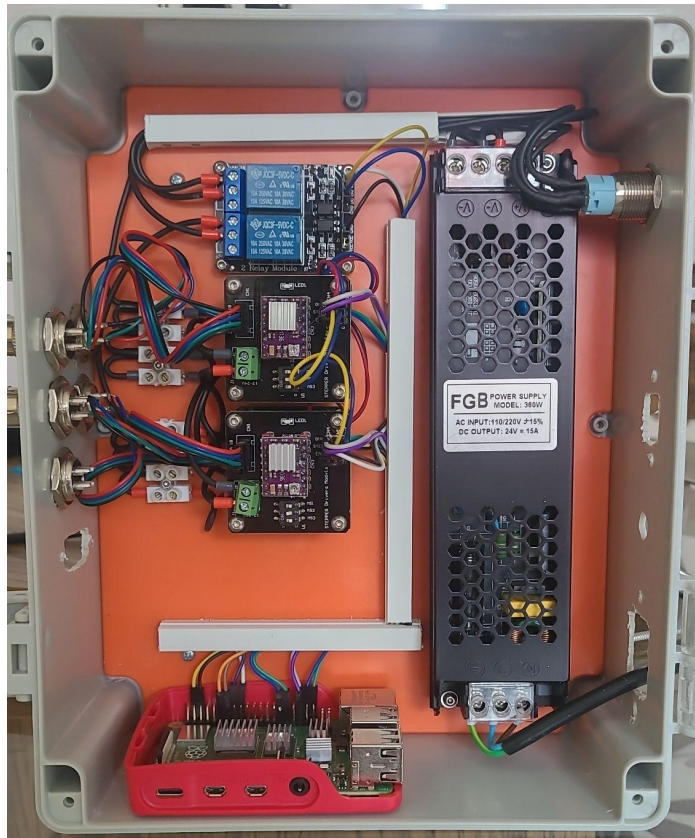
Com ela pode-se entender como os **nós** se relacionam e quais tópicos foram utilizados no funcionamento do BRES. As Figuras 24, 25 e 26 ilustram, respectivamente, o protótipo recolhido e estendido, o gabinete contendo a parte elétrica/eletrônica e, por fim, a visualização do projeto de *software* desenvolvido.

Figura 24 – BRES recolhido e estendido



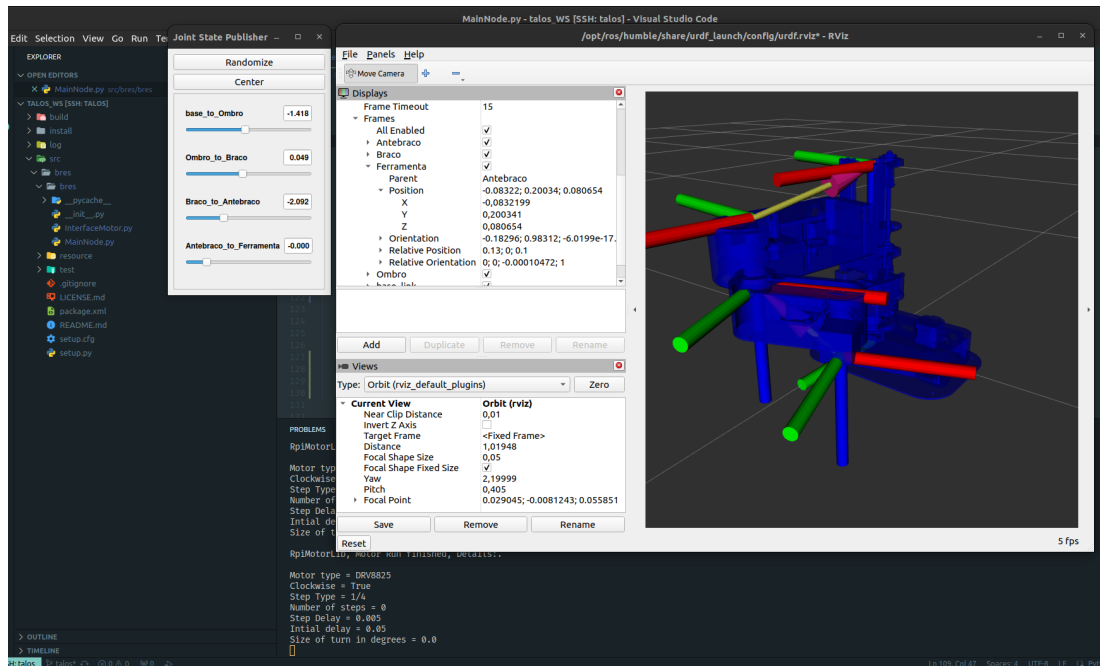
Fonte: Autor, 2023.

Figura 25 – BRES gabinete



Fonte: Autor, 2023.

Figura 26 – BRES software desenvolvido

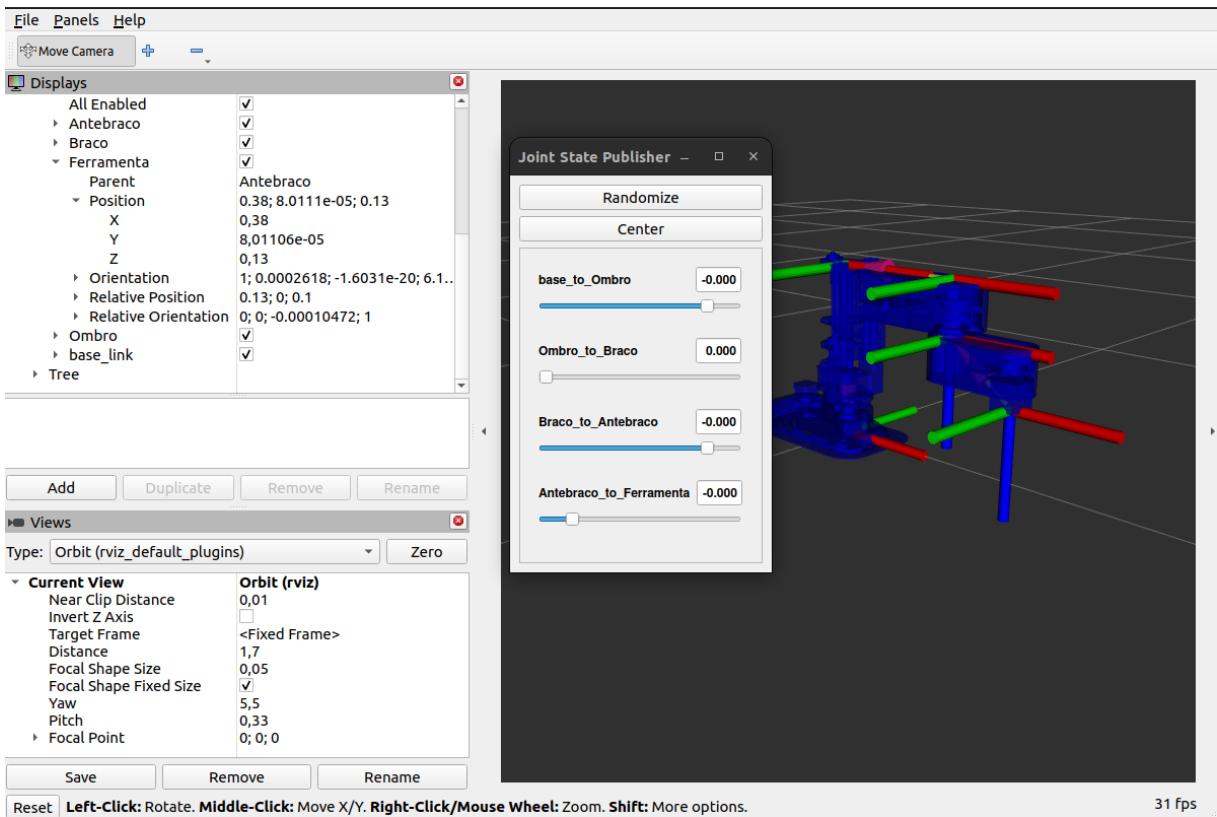


Fonte: Autor, 2023.

4.1 Análise e discussão dos resultados

Foi realizado o seguinte teste para avaliar a capacidade do BRES: o robô foi posicionado em sua posição zero (do ponto de vista ROS2, Figura 27) e medido com uma trena métrica a posição (x, y e z) da última junta, obtendo-se os valores de (400mm, 0 e 150mm). Esses valores foram obtidos a partir da medição até um ponto arbitrário para facilitar a medição com trena. Dessa forma é esperado ser diferente dos valores obtidos pelo modelo cinemático direto, porém consegue ser comparado por possuir na teoria um *offset* fixo, sendo que a cinemática direta desenvolvida prevê esse ponto sendo (380mm, 0 e 160mm) e o próprio ROS2 calcula esse valor por meio da geometria da simulação chegando na posição (380mm, 0 e 130mm).

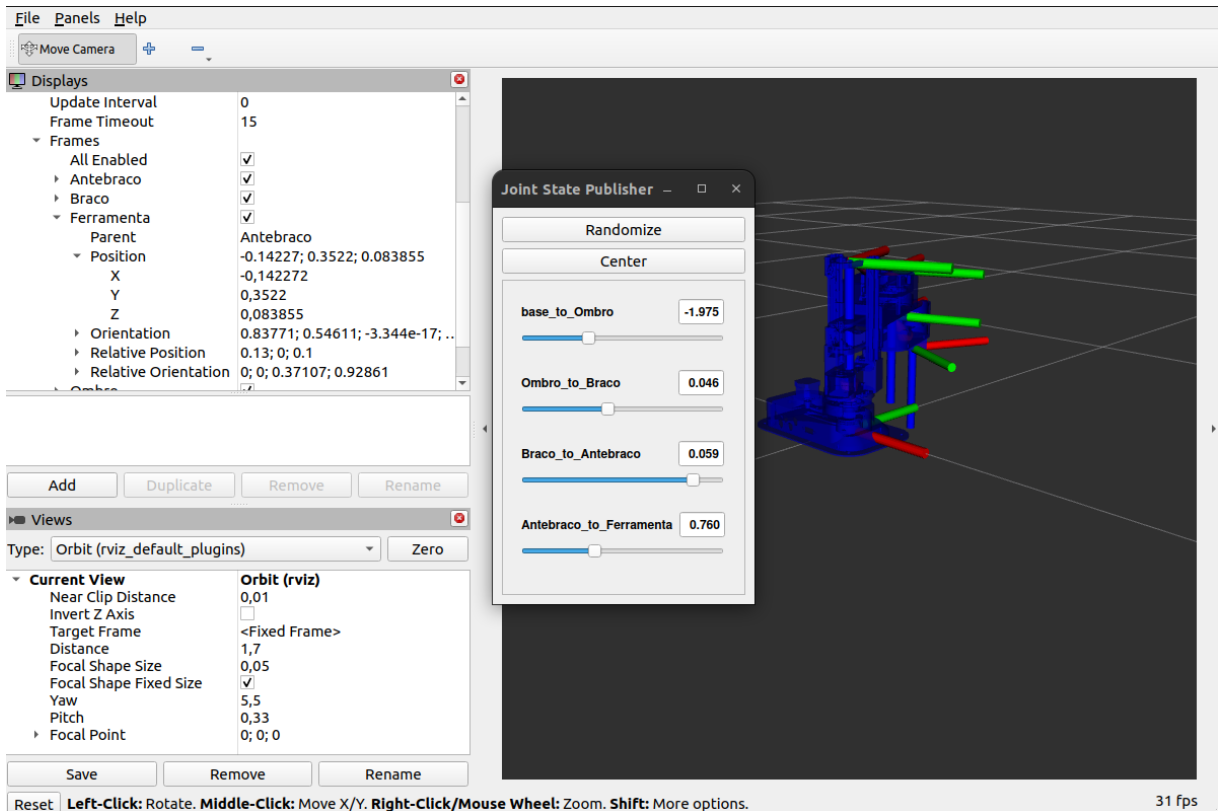
Figura 27 – BRES na posição zero



Fonte: Autor, 2023.

Em seguida, foi utilizado o botão *randomize* (Figura 28) que escolhe valores aleatórios para as juntas e, então, foi novamente medido com a trena obtendo (-180mm, 350mm e 110mm). Para os valores randômicos de junta, o modelo cinemático direto resulta em (-157mm, 346mm e 112mm) e o cálculo do ROS2 em (-142mm, 352mm e 84mm). Na tabela 1, pode-se observar os valores de posição obtidos pelo modelo cinemático direto, pelo ROS e pela medição manual.

Figura 28 – BRES na posição randômica



Fonte: Autor, 2023.

Tabela 1 – Comparação entre os resultados obtidos pelo modelo cinemático direto, pelo ROS e pela medição manual.

	X0	Y0	Z0	X1	Y1	Z1
Medição	400	0	150	-180	350	110
Cinemática	380	0	160	-157	346	112
ROS2	380	0	130	-142	352	84

Fonte: Autor, 2023.

Os resultados encontrados demonstram uma capacidade aproximada de prever o comportamento do protótipo com baixa precisão. É possível que o material do protótipo tenha causado os erros encontrados e o cálculo geométrico feito pelo ROS2 precisa ser ajustado para corresponder melhor com a cinemática direta para um aprimoramento de resultados. No entanto, considera-se que para um primeiro teste do BRES os resultados estão minimamente satisfatórios.

4.2 Realimentando a casa da qualidade

Com base nos resultados e objetivos atingidos foi realizada uma avaliação do BRES para preenchimento da Casa da Qualidade (Figura 29). Dessa forma é possível avaliar o protótipo em relação às características inicialmente desejadas e possibilita um processo de melhoria contínua para o desenvolvimento do BRES que pode ser continuado a partir deste trabalho.

Figura 29 – Casa da qualidade final

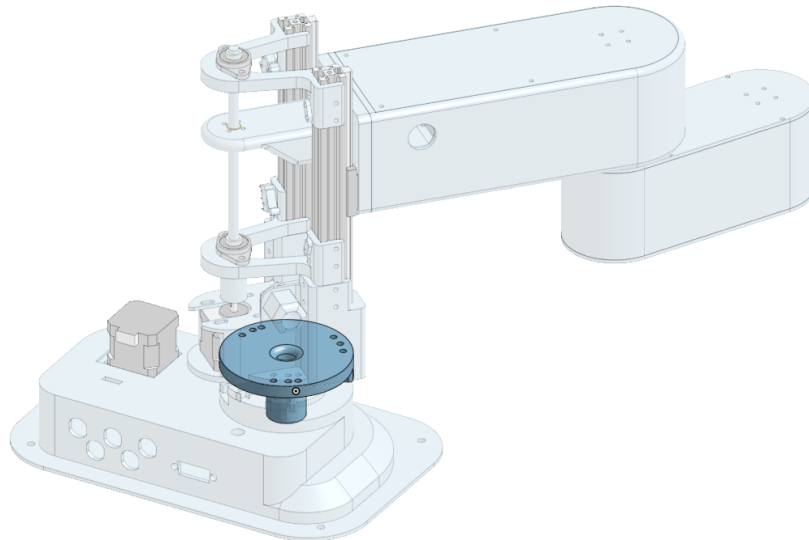
				ESPECIFICAÇÕES								SIMILARES				
				▼	▼	▼	▲	▲	▲	▲						
				Peso	Tamanho	Custo	Sistema Operacional	Documentação	Interface	Simulação		BRACON	iArm	Laboratory Robotic Arm	BRES Ideal	BRES Atingido
REQUISITOS	Facilidade de utilização	5	20,83%	👑	👑		👑	👑	👑	👑		2	4	2	4	2
	Interface do usuário	4	16,70%				👑	👑	👑	👑		3	5	2	5	3
	Movimentação precisa	3	12,50%	👑	👑	👑			👑			5	5	5	5	3
	Possibilidade de implementar recursos ao protótipo como câmeras ou sensores	2	8,30%	👑		👑	👑	👑	👑			3	5	4	5	4
	Simulação do braço robótico	4	16,70%				👑		👑	👑		3	0	3	3	5
	Acesso ao código e a documentação técnica sobre o braço	4	16,70%				👑	👑				1	3	3	3	3
	Abstração do software na hora da utilização	2	8,30%				👑		👑			1	4	1	4	3
Importancia Absoluta Para Projeto				26	24	33	145	57	116	55						
Importancia Relativa Para Projeto				5,70%	5,26%	7,24%	31,80%	12,50%	25,44%	12,06%						

Fonte: Autor, 2023.

4.3 Falha mecânica do protótipo

O BRES se mostrou impreciso para um uso não didático, provavelmente por utilizar componentes impressos com baixa precisão e material insuficientemente rígido. Um problema que aconteceu duas vezes durante o desenvolvimento foi uma peça impressa em 3D (em destaque na Figura 30) que rompeu sem conseguir suportar a força aplicada sobre a mesma.

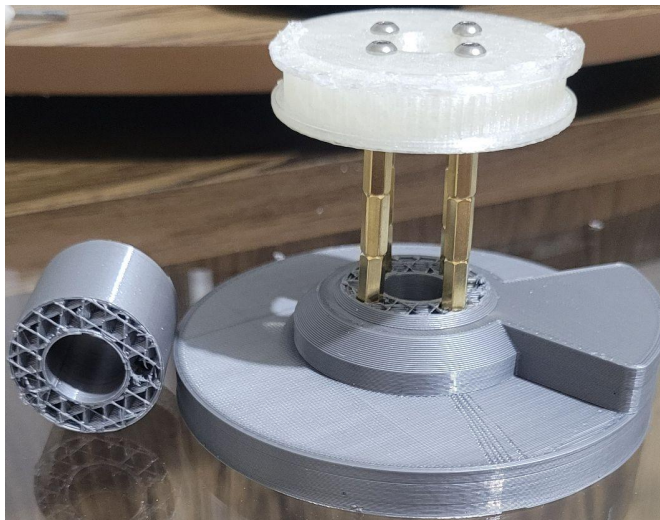
Figura 30 – CAD peça que apresentou falha



Fonte: Autor, 2023.

Por apresentar essa falha, a peça foi reparada com parafusos e espaçadores metálicos para conseguir realizar o teste de medição discutido (Figura 31), porém para um uso real é recomendado a confecção em alumínio dessa peça em especial. Outras peças podem ser também melhoradas em material ou estratégia de impressão para conferir maior precisão ao BRES.

Figura 31 – Reparo realizado na peça que apresentou falha



Fonte: Autor, 2023.

5 CONSIDERAÇÕES FINAIS

O objetivo geral do BRES “Utilizar tecnologias de código aberto e amplamente difundidas para modelar, simular e fabricar um protótipo de braço robótico SCARA aplicado ao ensino de robótica.”, se concretizou pelo uso de ROS2 (tecnologia de código aberto e amplamente difundido) na modelagem do robô com o arquivo URDF e na simulação via RVIZ e pela fabricação do protótipo com impressora 3D, corte a laser e componentes elétricos/eletrônicos utilizados, no entanto, o ensino de robótica por meio do BRES não foi validada durante o trabalho.

Seguindo os objetivos específicos, temos que, o desenho CAD e fabricação do protótipo foram concluídos, obtendo um dispositivo funcional; a configuração do ambiente de desenvolvimento utilizando o ROS2 foi bem sucedida; a modelagem e cinemática direta foram calculadas por meio do algoritmo de DH; esses cálculos foram utilizados no arquivo de descrição do robô URDF; a simulação via RVIZ foi construída por meio do pacote ROS2 urdf_launch; o sistema de comunicação entre computador e Raspberry pi 4 foi baseado nos recursos de tópicos do ROS2, funcionando como esperado no projeto; as GPIOs da Raspberry pi 4 que controlam os motores foram manipuladas pela programação em python, utilizando as bibliotecas RPi.GPIO e RpiMotorLib; com o protótipo construído foi efetuado o teste discutido no tópico 4.1, e encontrado a falha mecânica do tópico 4.3; por fim a documentação criada concentra-se neste trabalho, o código fonte está disponibilizado no site github e o arquivo CAD no site onshape.

Buscando o objetivo principal do BRES, o ensino de cinemática, o próximo passo é o desenvolvimento da cinemática inversa possibilitando que o aluno defina um ponto no espaço e o *software* define quais ângulos as juntas devem ter para atingir tal ponto. Este desenvolvimento também pede uma interface com usuário melhorada. Após o desenvolvimento e implementação da cinemática inversa é interessante o desenvolvimento do cálculo de trajetória sendo este responsável por um deslocamento contínuo do BRES até atingir um ponto no espaço.

A programação da Raspberry Pi 4 pode ser refatorada para enviar os comandos aos *drivers* de forma intercalada, fazendo os movimentos do BRES parecerem simultâneos. Outra sugestão, consiste em trocar a programação em Python por C++ para uma melhor performance do embarcado; essa troca não é necessariamente essencial pois a troca do código para C++ pode dificultar o entendimento do aluno sobre a lógica de funcionamento, porém é uma sugestão ainda válida pois C++ é a linguagem ensinada no curso de engenharia mecatrônica.

No ambiente ROS2 é possível a utilização de *actions* que podem melhorar o funcionamento do protótipo. Existe também a possibilidade de integrar outras máquinas, robôs ou equipamentos e desenvolver lógicas mais complexas, inclusive é aconselhável receber mais sensores como os de fim de curso e para fechamento de malha das juntas possibilitando uma profissionalização do protótipo. Pesquisas em visão computacional e aprendizado de máquinas podem utilizar o

BRES como equipamento auxiliar se aproveitando do ambiente ROS2 para se comunicar com diferentes programas.

REFERÊNCIAS

CAÑAS, J.; MARTÍN, Laura; VEGA, Julio. Innovating in robotics education with gazebo simulator and jderobot framework. In: **Proceedings of XXII Congreso Universitario de Innovación Educativa en Enseñanzas Técnicas, CUIEET**. 2014. p. 1483-1496.

Como preencher a matriz QFD ou Casa da Qualidade. Lucidchart, 2023. Disponível em: <<https://www.lucidchart.com/blog/pt/casa-da-qualidade-matriz-qfd>>. Acesso em: 06/12/2023.

Control Stepper Motor with DRV8825 Driver Module & Arduino. Last Minute Engineers, 2023. Disponível em: <<https://lastminuteengineers.com/drv8825-stepper-motor-driver-arduino-tutorial/>>. Acesso em: 06/12/2023.

CRUZ, Otávio Augusto Rocha da et al. Desenvolvimento de um Robô Móvel Open Source Baseado em ROS 2 para Pesquisa em Robótica Cooperativa. 2022.

GENIVAL, A. et al. O braço robótico como ferramenta interdisciplinar. <http://sistemaolimpico.org/midias/uploads/7cd05ca2e7bf2981ca931606c94c33d9.pdf> Acessado em, v. 13, n. 08, p. 2018, 2018.

How to install Ubuntu Server on your Raspberry Pi. Canonical Ubuntu, 2023. Disponível em: <<https://ubuntu.com/tutorials/how-to-install-ubuntu-on-your-raspberry-pi#1-overview>>. Acesso em: 06/12/2023.

KARALEKAS, Georgios; VOLOGIANNIDIS, Stavros; KALOMIROS, John. EUROPA—A ROS-based Open Platform for Educational Robotics. In: **2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)**. IEEE, 2019. p. 452-457.

KRASŇANSKÝ, Pavol et al. Basic laboratory experiments with an educational robotic arm. In: **2013 International Conference on Process Control (PC)**. IEEE, 2013. p. 510-515.

LUCIANO, Edivan Dagostin; DALEFFE, Anderson; DA ROSA, Magali. PROJETO ELETROELETRÔNICO DE UM BRAÇO ROBÓTICO DO TIPO SCARA UTILIZANDO COMANDO CNC. **Revista Vincci-Periódico Científico do UniSATC**, v. 5, n. 1, p. 237-261, 2020.

MASSON, Terezinha Jocelen et al. Metodologia de ensino: aprendizagem baseada em projetos (pbl). In: **Anais do XL Congresso Brasileiro de Educação em Engenharia (COBENGE), Belém, PA, Brasil**. sn, 2012. p. 13.

NAPOLITANO, Hamilton Barbosa; LARIUCCI, Carlito. Alternativa para o ensino da cinemática. 2001.

O que é e para que serve o Raspberry Pi?. Felipe Alencar, 2023. Disponível em: <<https://www.hardware.com.br/artigos/o-que-e-e-para-que-serve-o-raspberry-pi/>>. Acesso em: 06/12/2023.

PLAZA, Pedro et al. Home-made robotic education, a new way to explore. In: **2017 IEEE Global Engineering Education Conference (EDUCON)**. IEEE, 2017. p. 132-136.

QASSEM, Mohammed Abu; ABUHADROUS, Iyad; ELAYDI, Hatem. Modeling and Simulation of 5 DOF educational robot arm. In: **2010 2nd International Conference on Advanced Computer Control**. IEEE, 2010. p. 569-574.

Raspberry PI – GPIO output com Python. Cleiton Bueno, 2014. Disponível em: <<https://embarcados.com.br/raspberry-pi-ios-python/>>. Acesso em: 06/12/2023.

Raspberry Pi Stepper Motor Control with NEMA 17. Joshua Hrisco, 2021. Disponível em: <<https://makersportal.com/blog/raspberry-pi-stepper-motor-control-with-nema-17>>. Acesso em: 06/12/2023.

Raspberry Pi 4 Tech Specs. Raspberry Pi, 2023. Disponível em: <<https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>>. Acesso em: 06/12/2023.

RIVAS, David et al. BRACON: Control system for a robotic arm with 6 degrees of freedom for education systems. In: **2015 6th International Conference on Automation, Robotics and Applications (ICARA)**. IEEE, 2015. p. 358-363.

ROBÔ SCARA: SAIBA O QUE É E CONHEÇA 13 VANTAGENS. PAHC Soluções em Automação Industrial, 2022. Disponível em: <<https://www.pahcautomacao.com.br/robo-scara-saiba-o-que-e-e-conheca-13-vantagens>>. Acesso em: 29/05/2023.

ROS 2 Documentation: Humble. ROS.org, 2023. Disponível em: <<https://docs.ros.org/en/humble/index.html>>. Acesso em: 06/12/2023.

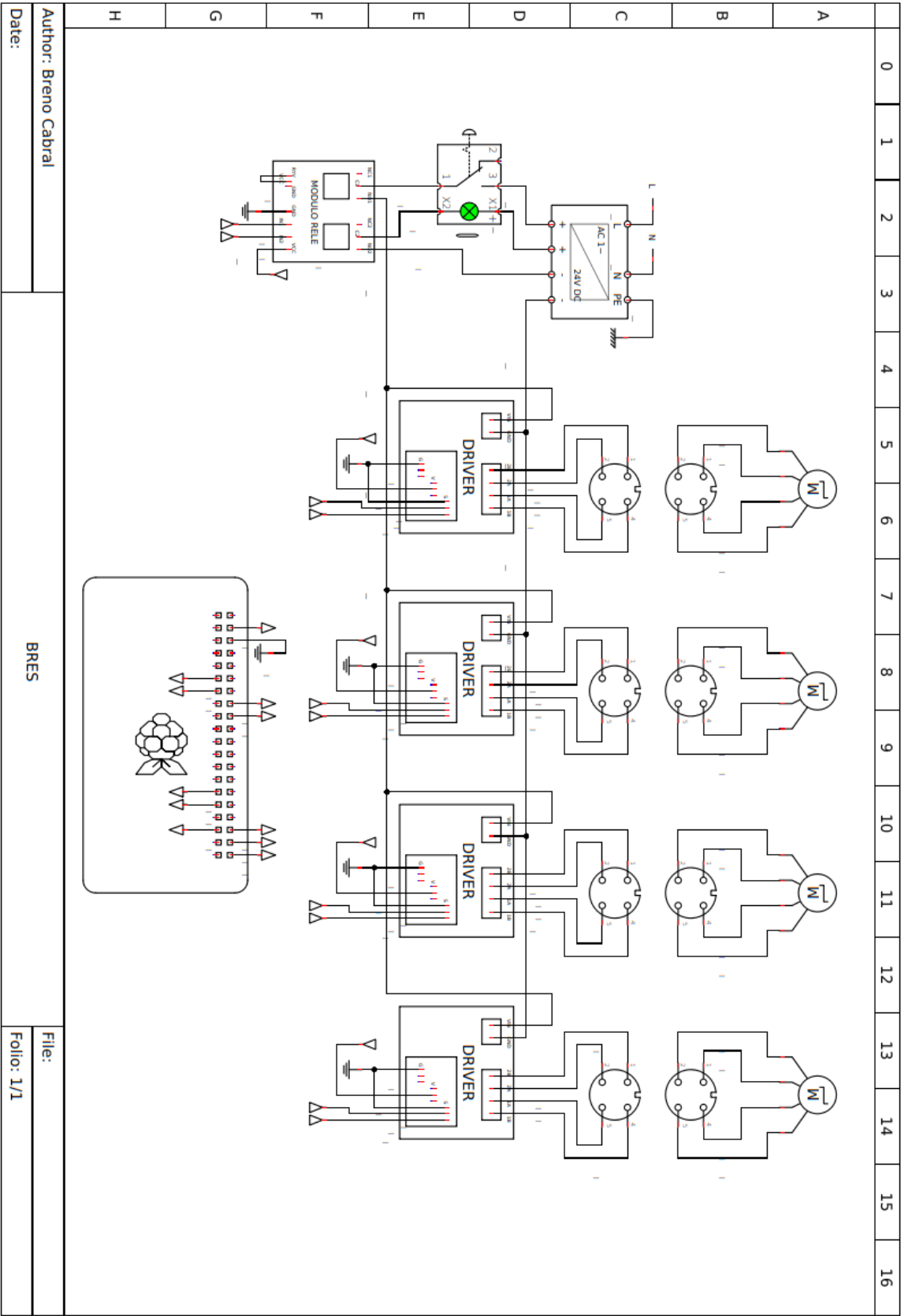
ROS2 Multiple Machines Tutorial (including Raspberry Pi). The Robotic Backend, 2023. Disponível em: <<https://roboticsbackend.com/ros2-multiple-machines-including-raspberry-pi/>>. Acesso em: 06/12/2023.

ROS2 vs. ROS1— key differences and which one is better?. Omar Elmofty. Medium, 2022. Disponível em: <<https://medium.com/@oelmofty/ros2-how-is-it-better-than-ros1-881632e1979a>>. Acesso em: 06/12/2023.

SCARA Robot | How To Build Your Own Arduino Based Robot. Dejan, 2022. Disponível em: <<https://howtomechatronics.com/projects/scara-robot-how-to-build-your-own-arduino-based-robot/>>. Acesso em: 29/05/2023.

ZENG, Chengze et al. iArm: Design an educational robotic arm kit for inspiring students' computational thinking. **Sensors**, v. 22, n. 8, p. 2957, 2022.

APÊNDICE A



APÊNDICE B

