UNIVERSIDADE SÃO JUDAS TADEU

Giovanni de Nicolai

Larissa Galvão Terron

Luccas Henrique Grotto Damasceno de Oliveira

Rodrigo Hortêncio Alves

Wesley Duarte Assumpção

AUTOMATIZAÇÃO NO PROCESSO DE ARMAZENAMENTO E COLETA EM ESTOQUES

Engenharia de Controle e Automação

Orientador: Prof. Dr. Rogério Issamu Yamamoto

São Paulo

2022

RESUMO

A gestão no processo de armazenamento e coleta em estoques é de vital importância, ao longo dos anos a falta de gestão demonstrou um grande problema na indústria acarretando prejuízos desnecessários, desde as pequenas empresas até grandes distribuidoras. Este trabalho de conclusão de curso tem por objetivo construir um manipulador cartesiano responsável pelo armazenamento e coleta de *pallets* através de uma interface homem-máquina, com o intuito de automatizar as movimentações e realizar a gestão pelo *software* Elipse SCADA (supervisório). O protótipo é capaz de executar comandos designados pelo operador e se comunicar com o Arduino via protocolo Modbus, exercendo uma relação mestre-escravo com o supervisório. O sistema trata e armazena as informações de forma a realizar o deslocamento dos eixos XYZ a partir da programação realizada em linguagem C integrada com o sistema de controle, composto pelo Elipse SCADA, Arduino, CNC *Shield* e o *drive* A4988 responsável pela movimentação dos motores de passo NEMA 17. A gestão é realizada por meio de um histórico presente no *software*, onde ficam disponíveis as movimentações realizadas, status de cada posição ("espaço disponível" e "espaço indisponível"), *login* e senha para identificação de usuários. A construção deste protótipo envolveu 3 frentes de pesquisa, sendo estas: mecânica, eletroeletrônica e computação, juntas proporcionaram uma automação precisa e de acordo com os objetivos traçados, desenvolvendo um protótipo com a capacidade de armazenamento e coleta de *pallets*, através do manipulador cartesiano integrado com o *software* elipse SCADA e Arduino.

Palavras-chave: Controle; Gestão; Manipulador Cartesiano.

ABSTRACT

Management in the process of storage and collection in stocks is of vital importance, over the years the lack of management has shown a major problem in the industry causing unnecessary losses, from small companies to large distributors. This course conclusion work aims to build a Cartesian manipulator responsible for the storage and collection of pallets through a human-machine interface, in order to automate the movements and carry out the management by the Elipse SCADA software (supervisory). The prototype is capable of executing commands designated by the operator and communicating with the Arduino via Modbus protocol, exercising a master-slave relationship with the supervisor. The system handles and stores the information in order to perform the displacement of the XYZ axes from the programming carried out in C language integrated with the control system, composed of Elipse SCADA, Arduino, CNC Shield and the A4988 drive responsible for moving the motors of NEMA step 17. Management is carried out through a history present in the software, where the movements performed, status of each position ("space available" and "space unavailable"), login and password for user identification are available. The construction of this prototype involved 3 research fronts, namely: mechanics, electronics and computing, together they provided a precise automation and in accordance with the outlined objectives, developing a prototype with the capacity of storage and collection of pallets, through the Cartesian manipulator integrated with the ellipse SCADA and Arduino software.

Keywords: Control; Management; Cartesian manipulator.

1. Introdução

Um levantamento realizado pela Fundação Dom Cabral junto a Associação Brasileira de Operadores Logísticos (2020/2021) identificou que o setor dos operadores logísticos é um dos que mais crescem e fornecem empregos no Brasil, a análise foi realizada de 2018 a 2020 comprovando um aumento de 23,8%, elevando o faturamento médio por empresa em 21%.

Christopher (1997) define que logística é o processo de gerenciar a aquisição, armazenamento e movimentação de materiais. Além de destacar a importância da implementação de sistemas logísticos para orientar nas movimentações realizadas em cada processo. Novaes (2004) ainda afirma que, sem estoque é impossível uma empresa trabalhar, pois este é definido por bens materiais que são mantidos e armazenados para suprir futuras demandas lucrativas.

É vital que as organizações apresentem qualidade no gerenciamento de seus estoques, pois eles chegam a absorver uma quantia substancial do capital da empresa, cerca de 25 a 40% (BALLOU, 2006). Pozo (2007) salienta que os gestores das organizações devem aplicar uma atenção especial a separação e administração dos materiais, pois esta atividade pode levar a organização ao sucesso ou ao fracasso.

Ao verificar as informações citadas acima, pode-se estabelecer a conexão direta que a logística tem perante o estoque e a relevância de uma boa gestão, monitoramento, armazenamento e correta divisão dos materiais.

Em entrevista ao site Terra, o CEO de uma startup mineira Ronei Marques (2018), diz que “O controle de estoque ainda não é uma preocupação para muitas empresas, o que é um erro, já que a falta de organização e eficiência pode gerar prejuízos importantes para o negócio". Em 2016 o Grupo Pão de Açúcar levou um prejuízo de R$ 512 milhões no lucro líquido, por problemas na má gestão de estoque da *Cnova* Brasil (MENDONÇA, 2016). De acordo com estudos realizados pela empresa *PlanetRetail* RNG (2018), furtos e má gestão de estoque fizeram empresas do varejo perderem 2,3 bilhões do faturamento no Brasil no ano de 2017. A má gestão de estoque foi responsável por 21,93% deste prejuízo.

Na busca por uma solução ao problema apresentado e analisando a área de formação onde Pazos (2002, p.1) afirma que “pode se definir a automação como a tecnologia que se ocupa da utilização de sistemas mecânicos, eletroeletrônicos e computacionais na operação e controle da produção [...] Portanto, a robótica é uma forma de automação de sistemas.” Pode-se considerar o desenvolvimento da automação em estoques e seus processos diretamente ligados.

Conforme a definição de Frank Lamb (2015, p.2) a automação "é o uso de comandos lógicos programáveis e de equipamentos mecanizados para substituir as atividades manuais que envolvem tomadas de decisão e comandos-resposta de seres humanos". Segundo Romano (2002, p.14), os robôs industriais são muito utilizados no processo de automatização, são adaptáveis as necessidades industriais e realizam diversos movimentos programados.

Um estudo realizado pela Associação Brasileira de Automação – GS1 Brasil (2020), visa mensurar o quanto a automação está presente nas indústrias e na vida dos consumidores. De acordo com os dados apresentados, a automação obteve um crescimento de 3% entre 2019 e 2020 no mercado industrial Brasileiro. Dentre os itens estudados, a logística se destaca na indústria e no comércio.

Desta forma, o presente trabalho refere-se ao desenvolvimento da automação no gerenciamento de armazenamento e coleta de *pallets* em um estoque, através da aplicação de um processo automatizado. Trata-se do desenvolvimento de um robô de coordenadas cartesianas, este apresenta um movimento linear e é dividido por três juntas prismáticas, onde os eixos se coincidem com as coordenadas XYZ. Em conjunto, um sistema faz a identificação dos *pallets*, armazena ou o envia para a expedição conforme a necessidade do usuário. O sistema é controlado por um supervisório SCADA (sistema de supervisão e aquisição de dados) que se comunica com o Arduino.

1.1. Objetivos

O objetivo geral do presente trabalho é apresentar a integração de um manipulador cartesiano na gestão de estoque, que permita realizar a movimentação de *pallets* mediante um *software* de sistema de supervisão e aquisição de dados controlado por um Arduino. Visando apresentar o processo de armazenamento e coleta de *pallets* a partir de eixos XYZ acionados por motores de passo.

Objetivos específicos:

1. Desenvolver um protótipo do manipulador cartesiano por eixos XYZ, para realizar a movimentação dos *pallets;*

2. Elaborar uma interface de comunicação entre o operador e o manipulador, através da aplicação de Arduino, comunicação Modbus e o *software elipse* SCADA;

3. Implementar uma programação para controlar os eixos movimentados por motores de passo, de acordo com o comando designado pelo operador;

2. Revisão bibliográfica

Diante dos objetivos apresentados, pesquisas foram realizadas para formular e apresentar o material teórico.

2.1. Robótica

Pozo (2007) classifica os robôs em quatro classe: manipuladores, exploradores, máquinas ferramenta e uso geral, demonstrado na figura 01.

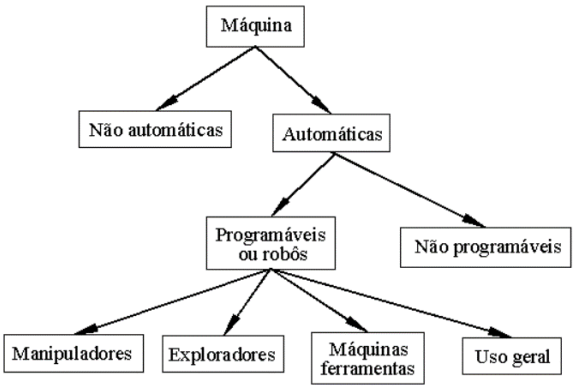

Figura 01 - Classificação de Máquinas (Fonte: Administração de recursos materiais e patrimoniais: uma abordagem logística 4.ed.)

Pozo (2007, p.8) classifica robôs manipuladores como “braços mecânicos, ou qualquer sistema que, em geral, tenha por objetivo deslocar material de um ponto a outro do espaço ou acompanhando uma trajetória dentro de um volume de trabalho”.

Mataric (2014, p.85) alega que “Um manipulador robótico é um efetuador. Pode se constituir de qualquer tipo de pinça, mão, braço ou parte do corpo que é usada para interagir e mover objetos em um dado ambiente.”

Gorgulho Jr. e Santos afirmam que os robôs cartesianos se deslocam de forma translacional. Devido sua ortogonalidade, o conjunto mecânico obtém boa rigidez. Afirmam ainda que “Esta estrutura é aplicável principalmente para manipulação e movimentação em sistemas de armazenagem. Tipicamente, apresenta-se com acionadores de natureza elétrica, podendo, em alguns casos, ser encontrada com acionamentos pneumáticos ou hidráulicos” (2015, p.27).

Segundo Craig (1989), manipuladores cartesianos são compostos de 1 a 3 juntas prismáticas, correspondentes aos vetores XYZ do plano cartesiano, podem manipular ferramentas pequenas ou até aeronaves, sua estrutura é forte e seu tamanho pode ser adaptado. A estrutura pode ser observada na figura 02.

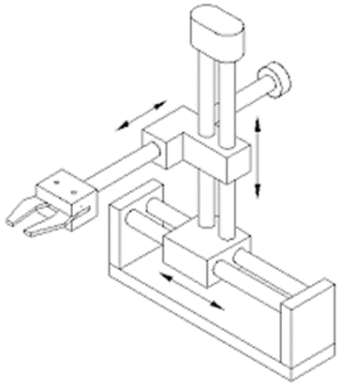

Figura 02– Estrutura de um Robô Cartesiano (Fonte: https://repositorio.ufmg.br)

Mataric (2014) explica que os manipuladores são constituídos por elos conectado por juntas e um efetuador.

- Elos: são componentes individuais do manipulador, como a estrutura, se visualizar o robô cartesiano, os eixos XYZ seriam os elos.

- Juntas: As estruturas (elos) do manipulador são conectadas por juntas, estas podem ser esféricas ou prismáticas;

- Efetuador final: É o componente que executa a interação com o ambiente, pode ser um braço, uma garra ou ferramenta;

2.2. Motor de Passo

O motor de passo funciona através da rotação de seu eixo, no entanto, seu giro não é contínuo como o tipo padrão. O eixo gira conforme os incrementos dos pulsos elétricos que são aplicados em determinada sequência. "Cada revolução é dividida em um certo número de passos e ao motor tem de ser enviado um pulso de tensão para cada passo. A quantidade de rotação é diretamente proporcional ao número de pulsos, e a velocidade de rotação está relacionada à frequência dos pulsos". Um motor de 1 grau por passo vai precisar de 360 pulsos para conseguir completar uma revolução (número de graus por passo). "Quando parado, um motor de passo se mantém inerentemente em sua posição". O movimento produzido pelo motor através do pulso, é preciso e repetível, por este motivo os motores de passo são mais utilizados e eficazes nos sistemas de posicionamento de carga. (PETRUZELLA, 2013, p. 105).

2.3. Arduino

O Arduino é denominado como um microcontrolador e foi desenvolvido na Itália com o intuito de auxiliar no ensino dos estudantes. O microcontrolador de uma placa de Arduino Uno utiliza o modelo ATmega e é especificamente um chip, este contém 28 pinos encaixados

em um soquete no centro da placa. Neste chip estão contidos o processador, a memória e toda a eletrônica necessária aos pinos de entrada e saída. As saídas são analógicas ou digitais e podem ser energizadas com 0 a 5 volts. Sua comunicação é realizada por um computador através de um cabo USB, a linguagem de programação utilizada é C ou C++ e são desenvolvidas no aplicativo ArduinoIDE (MONK, 2017).

De acordo com McRoberts (2010, p.3), "Arduino é um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos que você conecta a ele".

2.4. Programação

A linguagem utilizada neste trabalho será do tipo C. Para desenvolver a linguagem C o programador utiliza um conjunto de dados pré-definidos, classificados em: número inteiro *(int),* número de ponto flutuante *(float* ou *double),* caractere (*char*), desconhecido (*void*) e lógico *(bool*) (MANZANO, 2015).

Para converter uma linguagem de programação de alto nível para linguagem das máquinas (binário), é necessário um compilador. O compilador do Arduino é o IDE, um aplicativo de computador com um ambiente de desenvolvimento e um compilador integrado, nele você escreve a programação, o aplicativo converte para números binário, envia para memória do Arduino e executa o programa (CHAVIER, 2019).

2.5. Protocolo Modbus

Modbus é um protocolo de comunicação de estrutura aberta e flexível, além de ser muito utilizado nas indústrias, principalmente na comunicação entre CLP (controlador lógico programável) e IHM (interface homem máquina). "Publicado pela *Modicon* em 1979, é usado principalmente para estabelecer a comunicação entre dispositivos do tipo cliente-servidor onde um dispositivo, tido como mestre, é responsável por iniciar as transmissões, e os outros, escravos, respondem ao pedido do mestre" (Branquinho *et al.* ,2014, p. 7).

"O protocolo Modbus pode utilizar vários tipos de meio físico. O mais utilizado é o RS485 a dois fios (2W-MODBUS), existindo uma outra opção a quatro fios (4W-MODBUS). A interface serial RS232 deve ser utilizada somente para comunicação ponto a ponto." (Moraes e Castrucci, 2006, p. 169). Para transmissão do protocolo Modbus, existem duas opções: RTU e ASCII. O modo de transmissão deve ser o mesmo para todos os dispositivos, esta transmissão define o conteúdo da mensagem enviada e recebida. (Moraes e Castrucci, 2006)

2.6. Aquisição De Dados E Controle Do Supervisório (SCADA)

"Os sistemas SCADA são uma tecnologia que permite ao usuário coletar dados de uma ou mais instalações distantes e/ou enviar algumas limitadas instruções de controle para essas instalações. Sua utilização ocorre em larga escala em processos industriais" (BRANQUINHO, 2014, p.5). Seu funcionamento parte da comunicação com os equipamentos do processo em que está integrado, onde as informações são enviadas para o núcleo, esta distribui as informações para os módulos até chegarem em forma de gráfico, relatório ou aviso para o operador. (SANTOS, 2014).

3. Materiais e Métodos

Para elaboração do presente trabalho, seu desenvolvimento foi dividido em 3 frentes de pesquisa, sendo estas: mecânica, eletroeletrônica e computação. A construção, testes e aplicação do robô em um estoque fictício foram realizados na instituição de graduação, Universidade São Judas Tadeu.

3.1. Mecânica

3.1.1. Manipulador

O manipulador desenvolvido para locomoção de *pallets* nos processos de armazenamento e coleta dentro de um estoque é do tipo cartesiano com 3 graus de liberdade. O manipulador se locomove por meio de 3 juntas prismáticas lineares, movimentadas através de 4 motores de passo. A escolha do tipo cartesiano foi realizada devido sua elevada rigidez mecânica e alta capacidade de exatidão no posicionamento de materiais.

A estrutura mecânica do manipulador foi desenvolvida a partir de um projeto já realizado pelo professor Marlon Nardi, conforme demonstrado na figura 03. Em seu site, Nardi explica como realizar a construção de um CNC (*Computer Numeric Control,* traduzindo Controle Numérico por Computador) do zero e utilizá-la para desenhos 3D. Sendo assim, seu projeto foi utilizado como base para execução do cartesiano, entretanto, algumas modificações foram realizadas:

- No eixo Z a retífica (item destacado na figura 03) foi substituída por um efetuador do modelo garfo, implementado para retirar os *pallets* da prateleira.
- O manipulador será utilizado na posição horizontal, diferente do projeto demonstrado na figura 03 (posição vertical), o mesmo terá uma rotação e ficará em "pé" conforme figura 04.

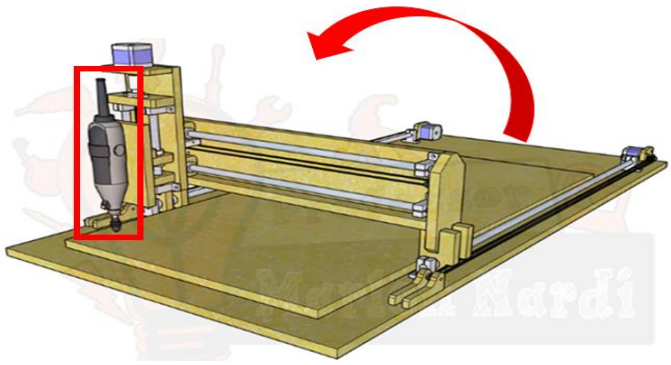

Figura 03 – Controle numérico computadorizado Marlon Nardi (CNC) (Fonte: Alterado https://marlonnardi.com/2020/06/16/construa-sua-propria-cnc-3-0-com-arduino/)

Para construir o manipulador foram utilizadas as peças mecânicas relacionadas na tabela 01. Além destas, 34 moldes em compensado foram produzidos para sustentar a estrutura do protótipo, como: mancais, bases, mesa de apoio, fixadores, suportes e travessas. O compensado foi escolhido por ser uma madeira mais resistente e de baixo custo. Na figura 04 pode-se observar a estrutura já construída.

Tabela 01 – Peças da Parte Mecânica do Manipulador Cartesiano (Fonte: Alterado https://marlonnardi.com/2020/06/16/construa-sua-propria-cnc-3-0-com-arduino/)

Peças	**Quantidade**
Guia Linear 12mm x 1000mm em Aço 1045 Tolerância H7	4
Rolamentos LM12UU	12
Suporte Eixo Guia Linear SK-12 Para guia 12mm	12
(300mm ou 30cm) Fuso Trapezoidal TR8 X Passo 8 Com 1 Flange De Latão	1
Acoplamento Flexível 5mm X 8mm	1
Mancal Modelo KFL8 com rolamento	1
Correia GT2 6mm/ Passo 2mm (em metros)	6
Polia GT2 Passo 2mm 16 Dentes Eixo 5mm	3

Figura 04 – Estrutura Mecânica Pronta (Fonte: Elaborado pelos Autores)

3.1.2. Efetuador

O garfo de empilhadeira é essencial para realizar a carga e descarga dos materiais armazenados. A peça possui este nome, justamente por se assemelhar ao garfo, este formato gera estabilidade ao conduzir o material transportado. O garfo utilizado no manipulador foi desenvolvido pelo grupo conforme figura 05 e será utilizado como efetuador, na ponta do eixo Z foram fixadas duas chapas de alumínio de 10,5 cm de comprimento e 5 mm de espessura, este material proporciona mais rigidez e firmeza, elevando a vida útil do equipamento.

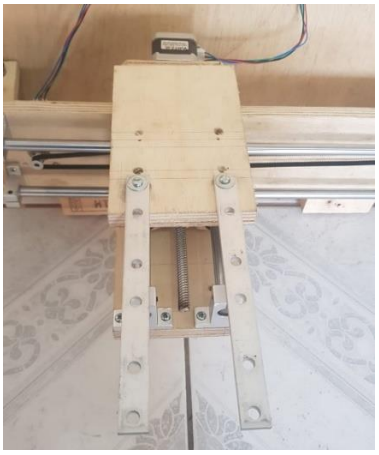

Figura 05 – Efetuador garfo de empilhadeira (Fonte: Elaborada pelos autores)

3.2. Eletroeletrônica

3.2.1. Atuador

O sistema atuador foi desenvolvido a partir de 4 motores de passo do tipo NEMA 17 (especificação descritas na tabela 02), este modelo de motor de passo realiza a conversão de pulsos elétricos em variações angulares rotacionais, de forma precisa. A movimentação angular descrita em suas especificações é de 1,8°, ou seja, a cada pulso elétrico no motor, seu eixo tem uma movimentação rotacional de 1,8°, para que o motor realize uma volta inteira são necessários 200 pulsos. Pode-se observar a estrutura do motor na figura 06.

O controle dos pulsos elétricos é realizado pelo circuito eletrônico, constituído pelo CNC *Shield* V3, *Drivers* A4988 e o Arduino.

O motor de passo foi escolhido por produzir um movimento preciso e repetível, sendo mais eficaz nos sistemas de posicionamento de carga, sua conversão é realizada do movimento rotativo em movimento linear através de um parafuso de potência.

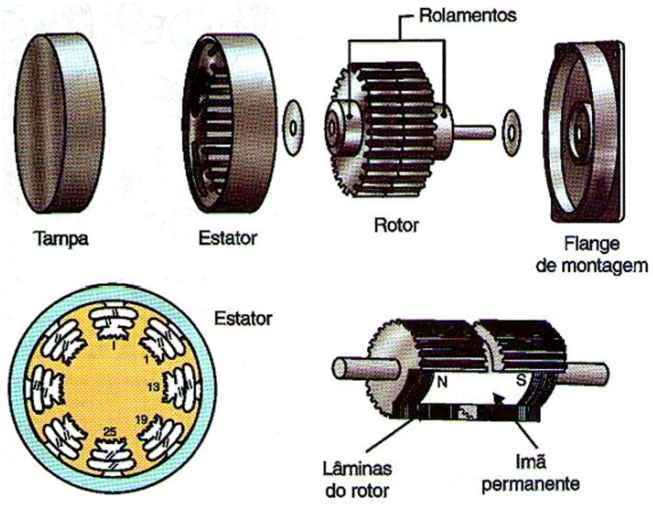

Figura 06 – Estrutura do Motor de Passo (Fonte: *https://www.newtoncbraga.com.br*)

Tabela 02 – Especificação do motor (Fonte: Alterado SPECIFICATIONS, T.; SPECIFICATION, G. Stepper motor nema 17 – 4.8kgf.cm / 1.0a, p.1, 2016)

Padrão	NEMA 17
Modelo	VZS1740-048-1004
Tensão por fase	3,9Vcc
Corrente por fase	1 A
Resistência por fase	3,9 Ω ±10%
Indutância por fase	8,9 mH± 20%
Torque de retenção	4,8 Kgf.cm
Inércia do rotor	54 gf.cm²
Fios	4
Conexão	Bipolar
Força dielétrica	500 VAC/1 min
Força radial	28 N
Força axial	10 N
Dimensões	42,3 x 42,3x 40 mm
Comprimento da Carcaça	40 mm
Peso	0,3 Kg
Quantidade de Fios	4 (2 fases)
Resolução	200 PPR / 1.8º
Temperatura de Trabalho	-20° à 50°

Em conjunto com o sistema atuador, é necessário um controlador para completar o processo de automação.

3.2.2. Controlador

Um controlador pode ser desenvolvido de diversas maneiras, na indústria o mais usual é a aplicação de um CLP (Controlador Lógico Programável) conforme cogitado no início do projeto, no entanto, por ser tratar apenas de um protótipo, optou-se pela utilização de um Arduino devido seu baixo custo e facilidade na manipulação. Sendo assim, o Arduino irá funcionar como o controlador de todo sistema, os programas são executados de acordo com os comandos recebidos, esta ação é realizada pelo microcontrolador, neste caso, o ATmega328. Os programas são armazenados na *Flash Memory* de 32 KB.

A comunicação e alimentação do Arduino é realizada através de um cabo USB (*Universal Serial Bus*, traduzindo barramento serial universal) conectado ao computador. Todas as conexões e especificações do Arduino utilizado estão descritas na Figura 07 e Tabela 03.

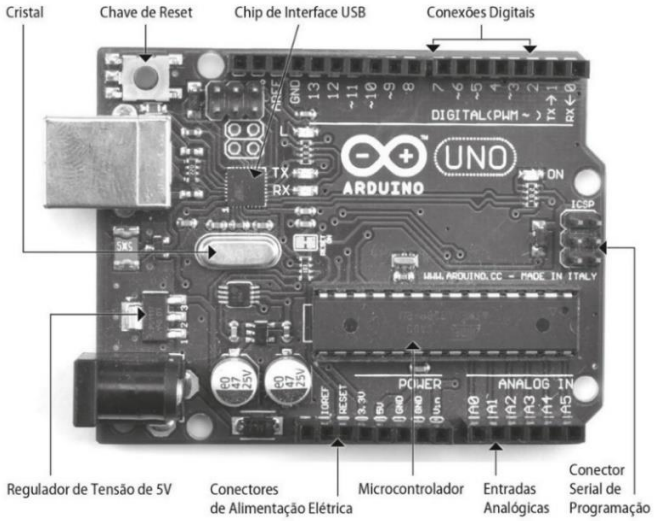

Figura 07 – Conexões do Arduino UNO (Fonte: *Programação com Arduino*, 2. ed.)

Tabela 03 – Modelo e especificação do Arduino (Fonte: Elaborado pelos autores)

Modelo	Arduino UNO
Microcontrolado	ATmega328
Tensão de Operação	5V
Pinos E/S Digitais	14
Pinos E/S Analógicos	6
Corrente DC por Pino E/S	40mA
Corrente DC por Pino 3.3V	50mA
Flash Memory	32KB
SRAM	2KB
EEPROM	1KB
Clock Speed	16MHz

Para realizar o controle dos motores, não basta utilizar apenas o Arduino. A placa CNC *Shield* foi projetada para permitir a integração entre os *drivers* de controle dos motores com o Arduino, possibilitando a extensão de algumas portas do Arduino. É possível observar o *layout* da placa na figura 08. A CNC *Shield* V3 possibilita a utilização de 4 eixos, no protótipo optou-se por utilizar o eixo X, Z e duplicar o eixo Y. Tem suporte para conectar 4 *drivers*, conexão para fim-de-curso e uma alimentação de 12 V a 36 V (Tensão compatível com o *drive*), conforme modelo descrito na tabela 04. Neste projeto a alimentação foi realizada através de uma fonte externa de 24 V.

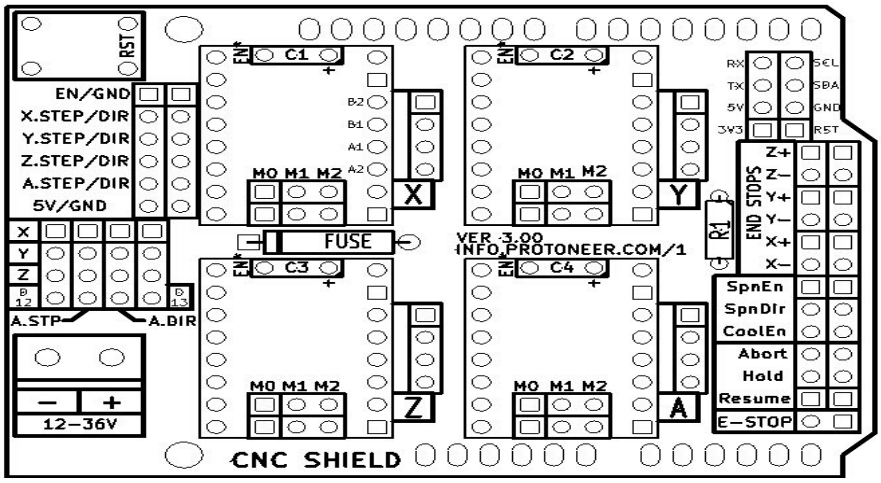

Figura 08 - Placa *CNC Shield V3* (Fonte: https://osoyoo.com/2017/04/07/arduino-uno-cnc-shield-v3)

Tabela 04 – Modelo Placa *CNC Shield V3* (Fonte: Alterado https://blog.eletrogate.com/cnc-shield-guia-pratico-e-introducao-ao-grbl/)

Código	CNC Shield V3
Alimentação	12V a 36V
Suporte	4 eixos
Conexões	I2C /Serial/ fim-de-curso
Compatibilidade	GRBL
Slots	4
Compatibilidade dos slots	A4988 ou DVR8825

A CNC *Shield* precisa de seu complemento, o *drive* que irá comandar cada motor de passo. O drive contém um *chip* A4988, nele possuem duas pontes H com transistor *FET DMOS*, eles realizam o controle de motores de passo bipolares, conforme tabela 05. De um lado do *drive* encontram-se as portas de controle do motor, alimentadas pelo Arduino e do outro lado, as conexões com o motor.

O *drive* realiza o controle de sinais enviados para rotação de cada motor, além de regular a corrente necessárias para alimentação do mesmo. Possui proteção para curto-circuito e excesso de temperatura.

Tabela 05 – Modelo e especificação *Driver A4988* (Fonte: Alterado https://blog.eletrogate.com/driver-a4988-motor-de-passo-usando-o-arduino/)

Código	A4988
Ponte H	2
Transistor	FET DMOS
Controle de saída	Até 35V ± 2A
Compatibilidade	Motor de passo bipolar
Pinos pull down	*M0, M1 e M2 - ENABLE*
Pinos pull up	*SLEEP*

Para utilização dos *drivers* são necessários alguns passos:

1. É necessário configurar o micro passo do motor, basta verificar a figura 09 e conectar os *jumpers* nas posições indicadas em MO, M1 e M2. Neste caso ¼ de passo.

MODE0	MODE1	MODE2	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
High	High	High	1/16 step

Figura 09 – Configuração do micro passo *(*Fonte: https://blog.eletrogate.com/cnc-shield-guia-pratico-e-introducao-ao-grbl/)

2. Colocar um *jumper* na conexão ED/GND, esta conexão fará com que o motor não se movimente até receber um novo comando.

3. A corrente do drive deve ter o mesmo valor da corrente máxima do motor. Para realizar o ajuste é necessário calcular o valor da tensão de referência e regular este valor no *trimpot*. Resistência do sensor do *Driver* é de 100 Ω e a Corrente máxima (Imáx) do motor é de 1 A.

$$I_{máx} = \frac{V_{REF}}{8 \times Rs} \quad (1)$$

$$V_{REF} = 8 \times 1 \times 100$$

$$V_{REF} = 800\ mV$$

3.2.3. Fonte

A fonte utilizada é uma fonte de alimentação chaveada estabilizada, tabela 06. Ela alimenta a placa CNC *Shield* e o *Cooler*. Tem uma entrada da rede elétrica de 110 V ou 220 V com uma saída de corrente contínua de 24 V.

Tabela 06 – Especificação Fonte de Alimentação (Fonte: https://teccontroll.com.br/products/fonte-chaveada-24vdc-10%2C41a-%252d-ms%252d250%252d24.html

Modelo	MS-250-24
Entrada	110 ~ 220 VAC - 50/60HZ
Saída	24 VDC
Corrente	5 A
Potência	120 W

3.2.4. Circuito Eletrônico

Um protótipo do circuito foi elaborado no Protheus para testes iniciais e verificar possíveis mudanças antes da parte prática. Conforme demonstrado na figura 10, pode-se observar: a alimentação do Arduino por meio do cabo USB e suas ligações com a CNC *Shield* através das entradas e saídas digitais; os *drivers* conectados na CNC *Shield* e suas ligações com os motores; e a fonte de alimentação fornecendo tensão para CNC *Shield* e para o *cooler*, utilizado para resfriar o circuito.

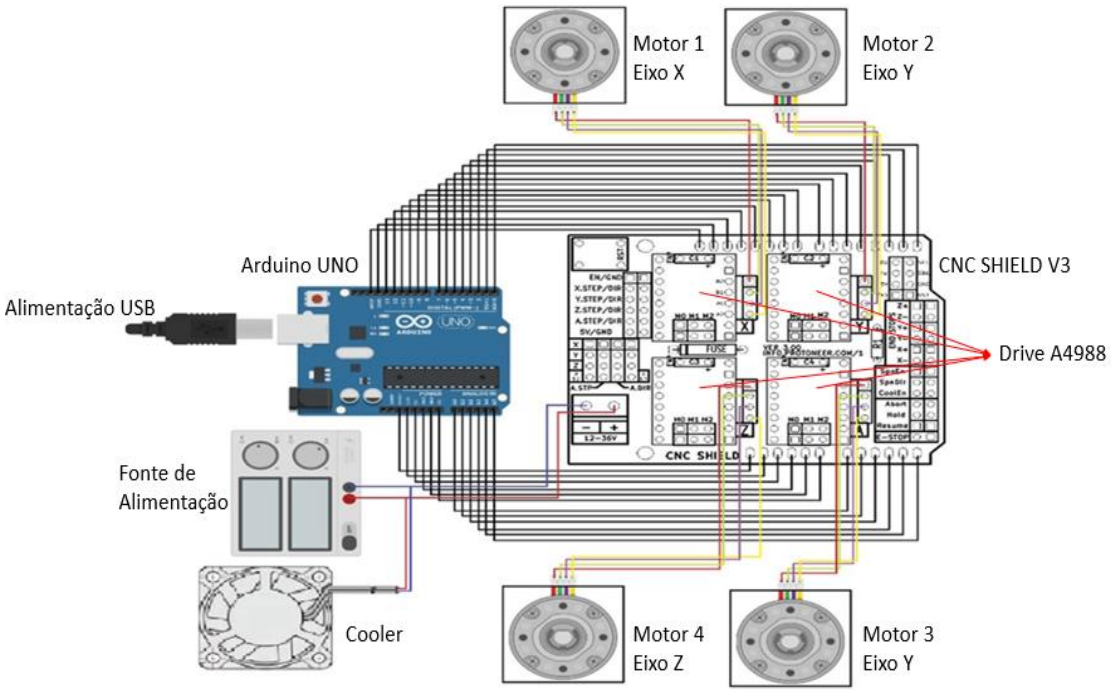

Figura 10 – Circuito Eletrônico (Fonte: Alterado https://blog.protoneer.co.nz/arduino-cnc-shield/)

3.3. Computação

3.3.1. Protocolo Modbus

Conforme descrito anteriormente, o protocolo Modbus é responsável pela comunicação entre o cliente e o servidor, mais conhecida como mestre-escravo. Será utilizado o padrão RS-485 (*Recommendad Standart-485*), este permite que o mestre comande até 32 escravos. O supervisório SCADA irá funcionar como mestre em uma rede Modbus RTU, portando como escravo o Arduino.

3.3.2. SCADA

A interface do supervisório SCADA foi desenvolvida na plataforma Elipse SCADA, adquirida no site: **https://www.elipse.com.br/produto/elipse-e3/.** Além de realizar uma interface homem-máquina, o *software* Elipse SCADA também exerce a função de gestão de estoque com um histórico das movimentações realizadas, status de cada posição (“espaço disponível” e “espaço indisponível”) e *login* para identificação de usuários.

Ao entrar no *software* o operador deverá se identificar realizando um *login*, conforme tela demonstrada na figura 11.

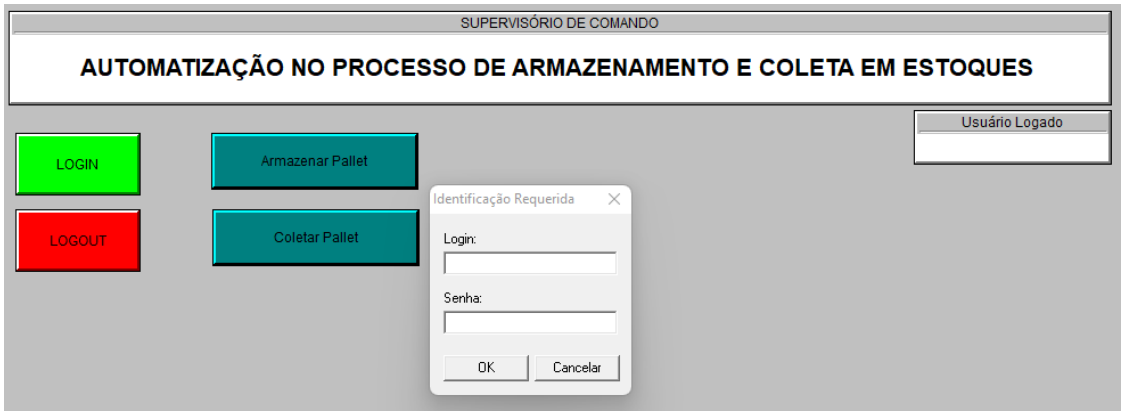

Figura 11 – Interface SCADA *login* (Fonte: Elaborado pelos autores)

Após realizar o *login*, o operador deve designar as funções para o manipulador, selecionando se deseja armazenar ou coletar um *pallet* , conforme figura 12.

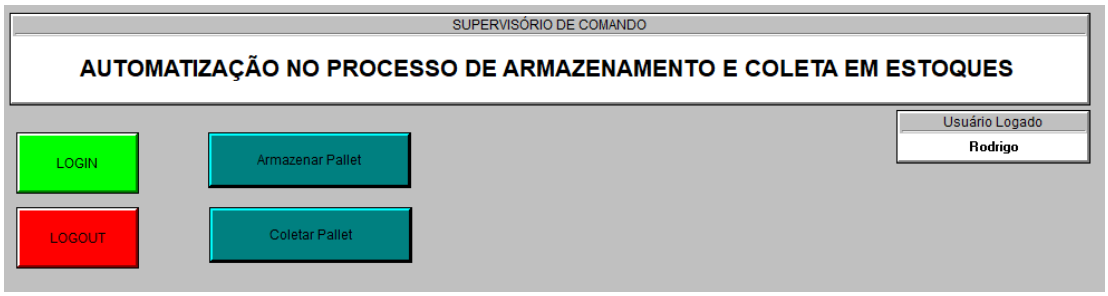

Figura 12 – Interface SCADA Tela Inicial (Fonte: Elaborado pelos autores)

Na figura 13, é possível observar um painel de controle com as posições disponíveis para disposição dos *pallets* ao lado esquerdo, no centro verificar a temperatura dos dissipadores e ao lado direito o histórico das movimentações. Após selecionar a posição desejada, basta voltar e realizar o *logout* ou coletar/armazenar outro *pallet.*

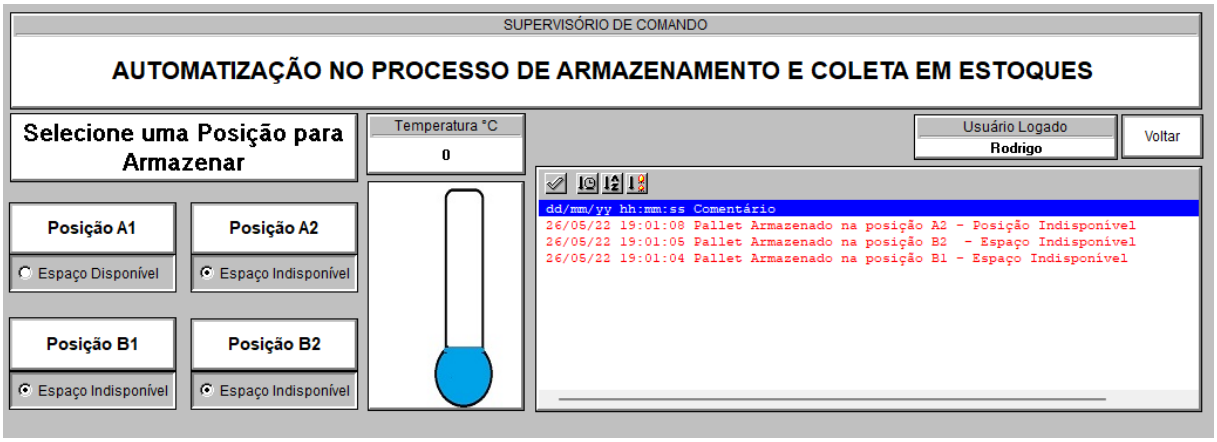

Figura 13 – Interface SCADA Tela de Controle (Fonte: Elaborado pelos autores)

3.3.3. Sistema de Controle

O sistema de controle exibido na figura 14 foi elaborado a partir da integração do Elipse SCADA utilizando o Protocolo Modbus para se comunicar com o Arduino conectado a *CNC Shield* e aos *drivers* A4988 que movimentam os motores de passo NEMA 17.

Para realizar a integração é necessário comunicar o Arduino com o *software* Elipse SCADA, sendo necessário um *hardware* com configuração RTU, ou seja, o protocolo Modbus que irá receber e transmitir as informações recebidas pelos dispositivos ao *Software.*

A configuração do protocolo é realizada tanto no Arduino IDE, em formato de biblioteca **"ArduinoModbus"**, quanto no *software* Elipse SCADA em formato de *driver* ***"Driver Modicon Modbus Master* (ASC/RTU/TCP)"**. Após ambos instalados, foi necessário configurar o modo RTU, *Serial e Timeout* para 1000ms e escolher a mesma porta COM.

Os sinais que alimentam o sistema de controle são os designados pelo operador no *software* Elipse SCADA (mestre), este é transmitido através do protocolo Modbus em uma porta serial USB. O Arduino junto com a *CNC Shield* (escravos) recebe o sinal e converte em controle aos *drivers*, responsáveis por movimentar os motores.

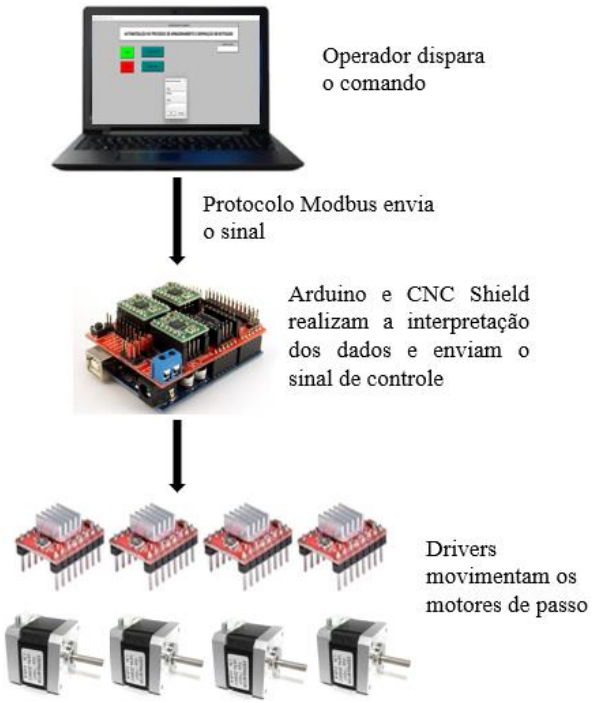

Figura 14 – Sistema de Controle (Fonte: Elaborado pelos autores)

4. Resultados e Discussões

Conforme descrito, o manipulador foi desenvolvido com base no CNC do Marlon Nardi, as alterações necessárias para modificar sua função foram realizadas, o manipulador será utilizado em pé, com efetuador do tipo garfo e uma gôndola para armazenamento dos *pallets.* Nas figuras 15 e 16 é possível visualizar o manipulador e a gôndola prontos.

Figura 15 – Manipulador (Fonte: Elaborado pelos autores)

Figura 16 – Manipulador com a gôndola (Fonte: Elaborado pelos autores)

O manipulador tem um espaço de trabalho em X, Y e Z, conforme descrito na tabela 07. É uma área de trabalho pequena se for pensado em uma indústria, porém por se tratar de um protótipo o espaço utilizado é adequado para a demonstração do sistema.

Tabela 07 – Espaço de trabalho do manipulador (Fonte: Elaborado pelos autores)

Eixo	Espaço de Trabalho em Centímetros
X	44,5 cm
Y	97,5 cm
Z	14,5 cm

Cada motor é responsável por movimentar um eixo, com exceção do eixo Y que utiliza dois motores por ser o mais extenso e pesado. Segue relação de cada eixo e motor na tabela 08.

Tabela 08 – Definição de motor e eixo (Fonte: Elaborado pelos autores)

Eixo	Motor que realiza o movimento
X	Motor 1
Y	Motor 2 e Motor 3
Z	Motor 4

O sistema de controle foi desenvolvido conforme já descrito, a linguagem de programação foi desenvolvida e alterada conforme o progresso dos testes, as posições designadas na gôndola foram alinhadas de acordo com a quantidade de passos do motor. Esta foi a parte mais delicada do desenvolvimento, pois a posição deve ser precisa para que não ocorram acidentes no momento de deixar ou retirar algum *pallet* da prateleira. Na tabela 09 estão descritos a quantidade máxima de passos que cada motor realiza para chegar em determinada posição, segundo os passos pré-estabelecidos na programação.

Tabela 09 – Quantidade de passos máxima que cada motor realiza para cada posição (Fonte: Elaborado pelos autores)

Posição na gôndola	Motor 1 (passos)	Motor 2 e Motor 3 (passos)	Motor 4 (passos)
A1	0	113.000	248.000
A2	132.000	116.000	248.000
B1	0	224.000	248.000
B2	132.000	216.800	248.000

O circuito foi montado fisicamente (figura 17) conforme o protótipo de testes elaborado no Protheus, na figura 11. O mesmo foi instalado na parte traseira do manipulador com o intuito de não atrapalhar a movimentação dos eixos.

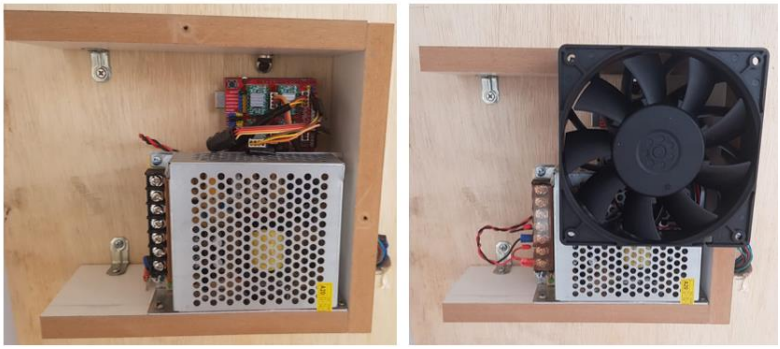

Figura 17 – Circuito Eletrônico (Fonte: Elaborado pelos autores)

4.1. Programação

As bibliotecas utilizadas na programação são subprogramas que auxiliam no desenvolvimento do *software* com funções pré-compiladas. Conforme demonstrado na figura 18, ao todo foram utilizadas 6 bibliotecas para auxiliar no desenvolvimento da programação deste protótipo.

Figura 18 – Bibliotecas programação (Fonte: Elaborado pelos autores)

- <Modbusino.h> a biblioteca implementa o protocolo Modbus na programação utilizando a comunicação serial RS485 com RTU (*Remote Terminal Unit*) e possibilita a definição do Arduino como mestre ou escravo, código demonstrado na figura 19:

```
ModbusinoSlave modbusino_slave(1);// define Arduino como escravo 1
uint16_t tab_reg[10];
```

Figura 19 – Definição do Arduino como escravo (Fonte: Elaborado pelos autores)

- "A4988.h" e "*BasicStepperDriver.h*" estas bibliotecas trabalham em conjunto, controlando os pulsos PWM para definição de passos do motor e revoluções (RPM-Rotação por Minuto), conforme demonstrado na figura 20.

```
//Define 200 passos por revolução
#define MOTOR_STEPS 200

// RPM para eixo X,Y e Z
#define MOTOR_X_RPM 30
#define MOTOR_Y_RPM 20
#define MOTOR_Z_RPM 30
```

Figura 20 – Definição passos e Revolução dos motores na programação (Fonte: Elaborado pelos autores)

Na figura 21, é realizada a instância de três varáveis/objetos a serem controlados: *Motor_STEPS* é variável dos motores, *dir* é a variável direção e *Step* são os passos.

```
BasicStepperDriver stepperX(MOTOR_STEPS, DIR_X, STEP_X);
BasicStepperDriver stepperY(MOTOR_STEPS, DIR_Y, STEP_Y);
BasicStepperDriver stepperZ(MOTOR_STEPS, DIR_Z, STEP_Z);
```

Figura 21 – Instâncias de variáveis da programação (Fonte: Elaborado pelos autores)

- "*MultiDriver.h"* esta biblioteca possibilita a utilização de mais de um *drive* e o controle de mais de um eixo, na figura 22 o código demonstra as variáveis utilizadas de acordo com a biblioteca para utilizar 3 eixos.

```
MultiDriver controller(stepperX, stepperY, stepperZ);
```

Figura 22– Definição de eixos na programação (Fonte: Elaborado pelos autores)

- "*SyncDriver.h*" sincroniza o tempo para que todos os motores atinjam seu alvo ao mesmo tempo.

- "DHT.h" esta é uma biblioteca para sensores de temperatura da série DHT (*digital temperature and humidity sensor,* traduzindo sensor digital de temperatura e umidade), monitora a temperatura dos dissipadores, de acordo com a figura 23.

```
#define DHTPIN A3
#define DHTTYPE DHT11
DHT dht(DHTPIN,DHTTYPE);
float temperatura;
```

Figura 23 – Definição sensor de temperatura (Fonte: Elaborado pelos autores)

4.2. Custo

Na tabela 10 pode-se observar o custo de cada produto utilizado no projeto, o valor total foi de R$1264,90 dentro do esperado pelo grupo. Conforme previsto, os itens mais caros foram os motores, os rolamentos obtiveram um custo elevado devido a quantidade utilizada.

Tabela 10 – Custo do Projeto (Elaborado pelos autores)

Peça / Descrição	Quantidade	Valor unitário	Valor
Guia Linear 12mm x 300mm em Aço 1045 Tolerância H7	3	R$ 26,00	R$ 78,00
Rolamentos LM12UU	12	R$ 12,50	R$ 150,00
Suporte Eixo Guia Linear SK-12 Para guia 12mm	12	R$ 15,00	R$ 180,00
300mm Fuso Trapezoidal TR8 X Passo 8 Com 1 Flange De Latão	1	R$ 54,00	R$ 54,00
Acoplamento Flexível 5mm X 8mm	1	R$ 20,00	R$ 20,00
Mancal Modelo KFL8 com rolamento	1	R$ 19,00	R$ 19,00
Correia GT2 6mm/ Passo 2mm (em metros)	6	R$ 17,33	R$ 104,00
Polia GT2 Passo 2mm 16 Dentes Eixo 5mm	3	R$ 16,63	R$ 49,90
Driver Motor de Passo A4988	4	R$ 12,50	R$ 50,00
Arduino Uno	1	R$ 50,00	R$ 50,00
Motor de passo Nema 17	4	R$ 70,00	R$ 280,00
CNC *Shield* V3	1	R$ 30,00	R$ 30,00
Madeira/ Compensado	1	R$ 200,00	R$ 200,00
Total			**R$ 1264,90**

4.3. Funcionamento

Para demonstrar o passo-a-passo do funcionamento sistêmico, foi elaborado o fluxograma demonstrado na figura 24. O sistema inicia com a tela do *software* Elipse SCADA solicitando *login* e senha do usuário. Após o *login* realizado, o sistema identifica o usuário e disponibiliza na tela a opção de armazenamento e coleta, assim o usuário seleciona a operação e posição desejadas. Neste momento é estabelecida a comunicação Modbus com o Arduino que recebe as informações de operação e posição e envia para os *drivers*, estes estão acoplados no manipulador e realizam as movimentações designando o *pallet* para a posição escolhida e

finalizam o comando solicitado. Por fim, o usuário decide se deseja realizar um novo serviço, retornando para tela inicial ou finaliza suas ações realizando *logout.*

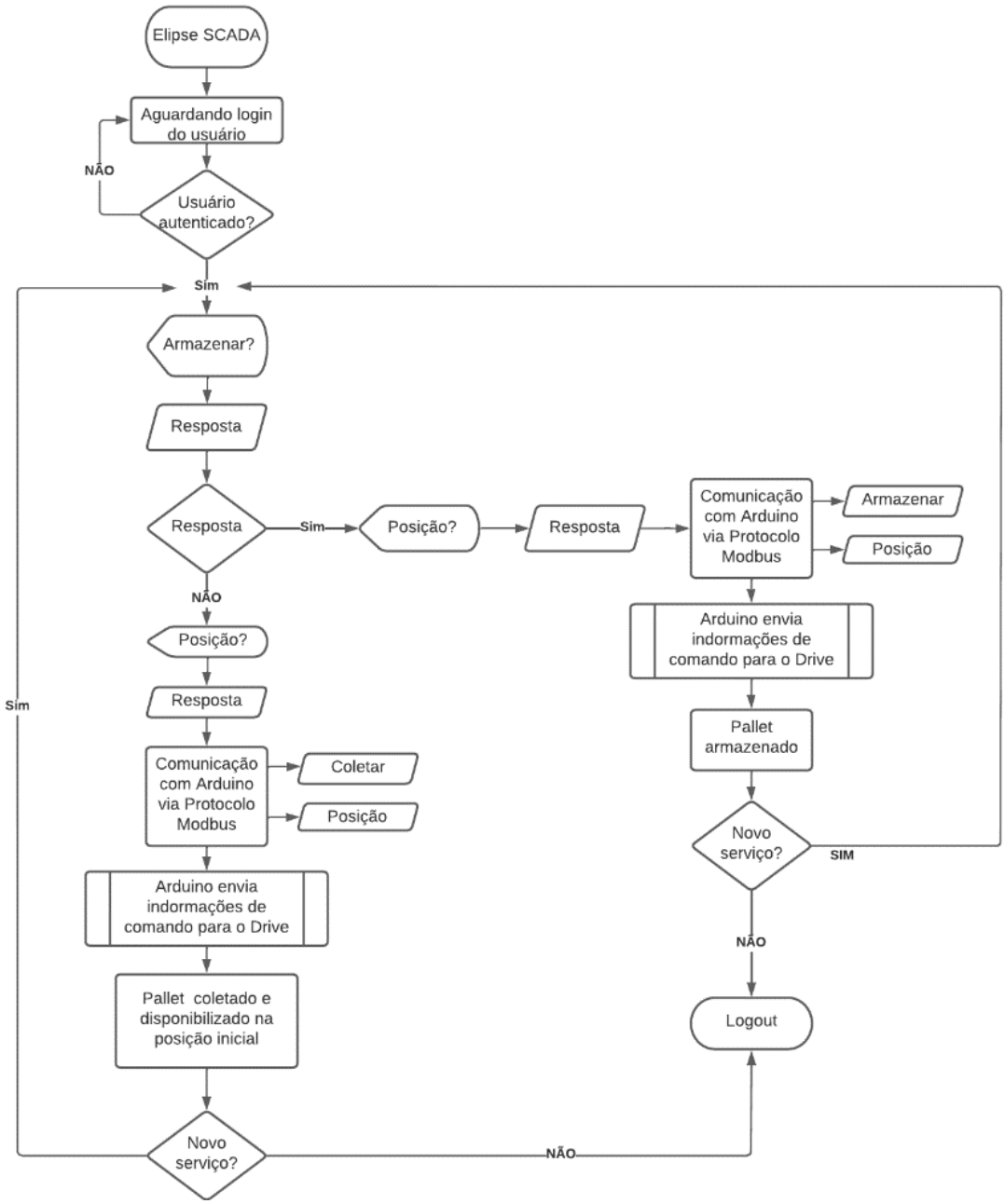

Figura 24 – Fluxograma de funcionamento (Fonte: Elaborado pelos autores)

Com o propósito de analisar a precisão do manipulador, foram realizados testes de armazenamento e coleta de *pallets* para validação, os pontos levantados para observação foram:

1. Se o manipulador posicionou o *pallet* na posição correta;
2. Se o manipulador retirou o *pallet* na posição correta;
3. Se não ocorreram instabilidades no movimento;
4. Se não houve acidente durante os movimentos;

As figuras 25, 26, 27, 28, 29, 30 e 31 visam demonstrar o funcionamento do protótipo do início ao fim através de fotos, demonstrando uma hipótese do que o operador pode fazer, de acordo com os passos a seguir: (a) operador realizando *login*, (b) operador designando operação de armazenamento, (c) operador escolhendo posição desejada – B2, (d) manipulador na posição inicial, (e) manipulador pegando o *pallet*, (f) manipulador seguindo para posição desejada, (g) manipulador armazenando o *pallet*, (h) manipulador retornando para posição inicial, (i) manipulador na posição inicial, (j) operador designa nova operação – coleta, (k) operador escolhendo a posição – A1, (l) manipulador na posição inicial, (m) manipulador seguindo para posição desejada, (n) manipulador coletando *pallet*, (o) manipulador entregando *pallet*.

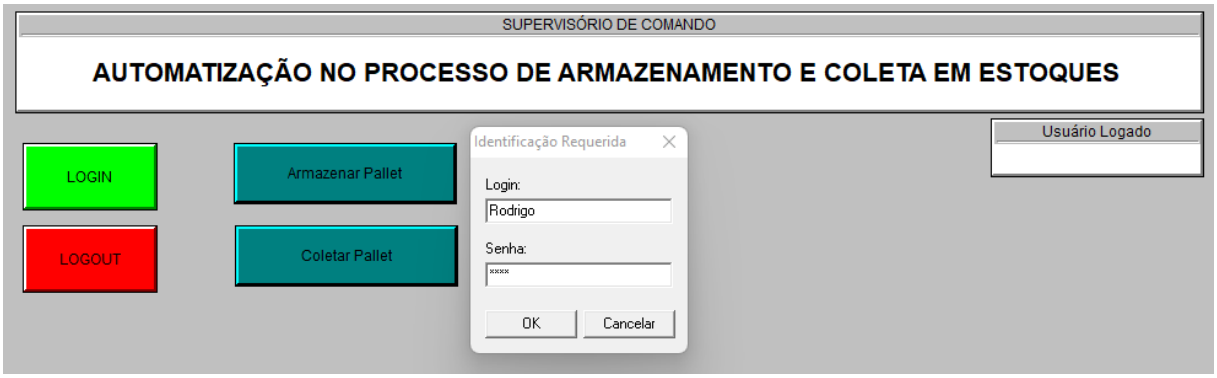

Figura 25 – (a) – Tela *Login Software* (Fonte: Elaborado pelos autores)

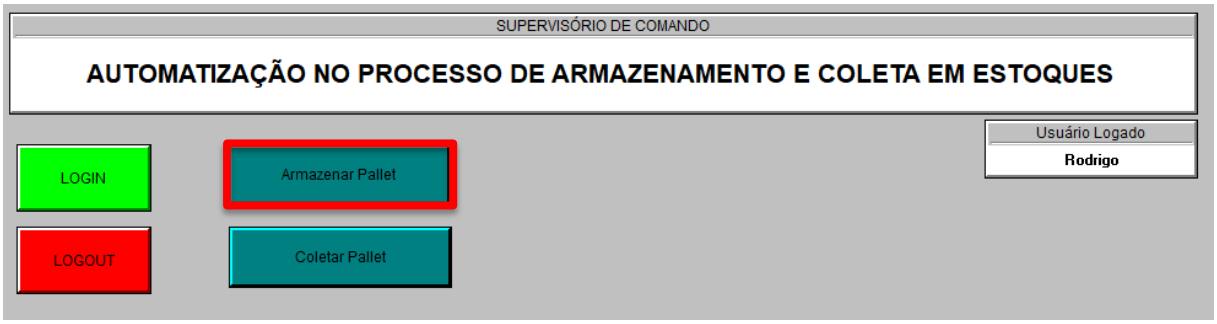

Figura 26 – (b) – Tela Inicial do *Software* (Fonte: Elaborado pelos autores)

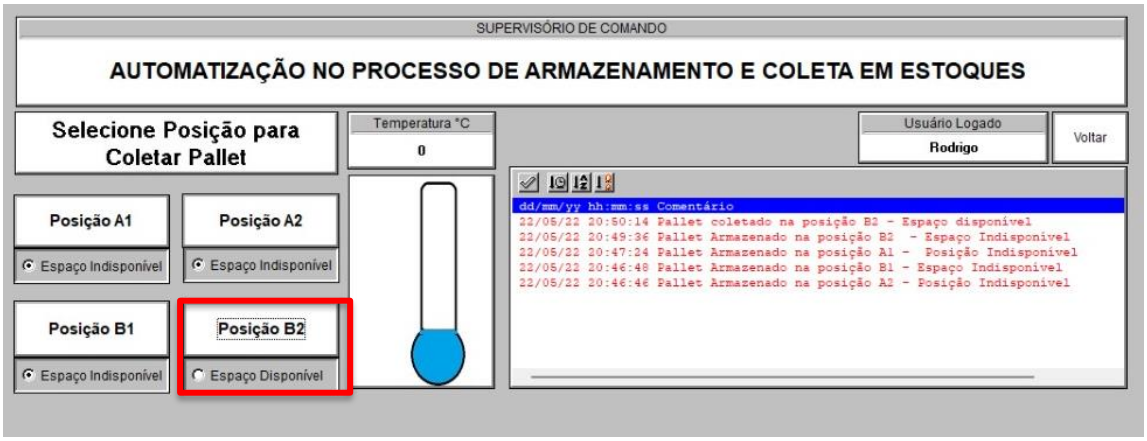

Figura 27 – (c) – Tela de Comando do *Software* (Fonte: Elaborado pelos autores)

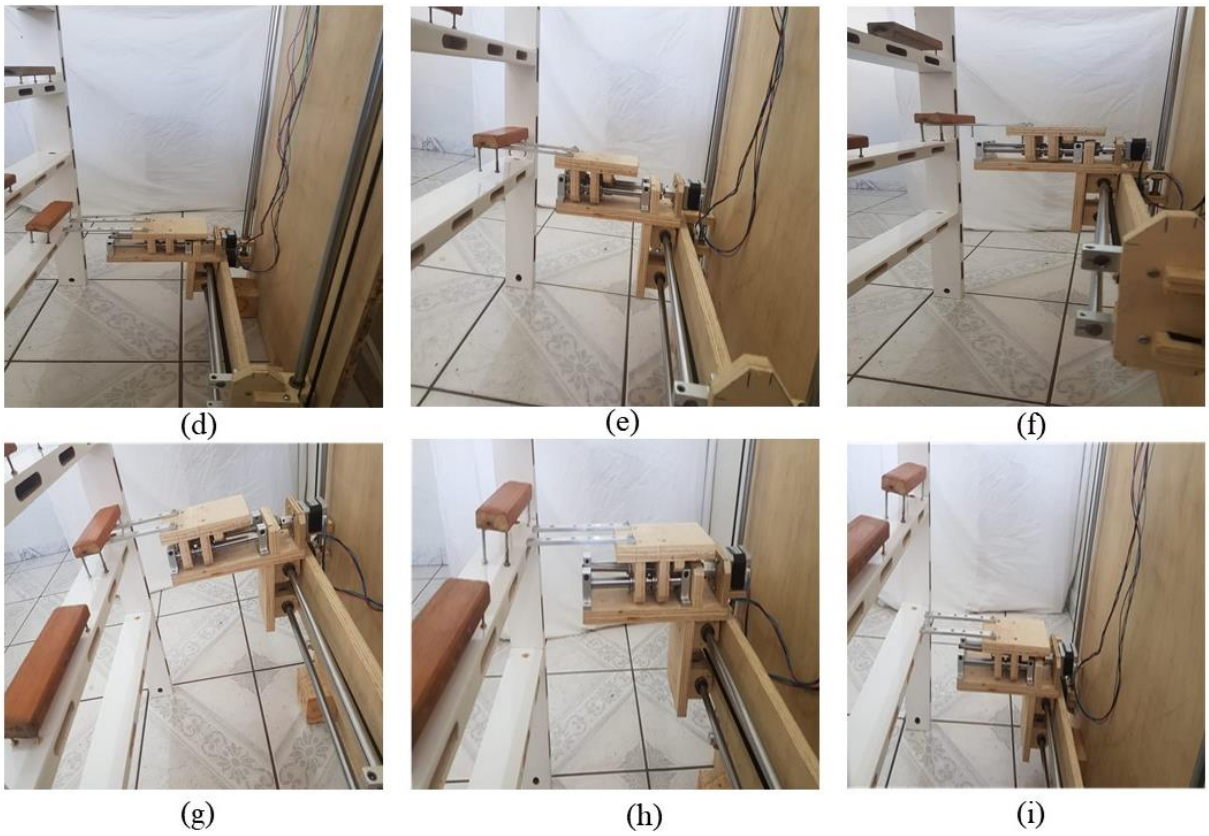

Figura 28 – Funcionamento – Movimentações de armazenamento do manipulador (Fonte: Elaborado pelos autores)

Figura 29 – (j) - Tela inicial para nova escolha (Fonte: Elaborado pelos autores)

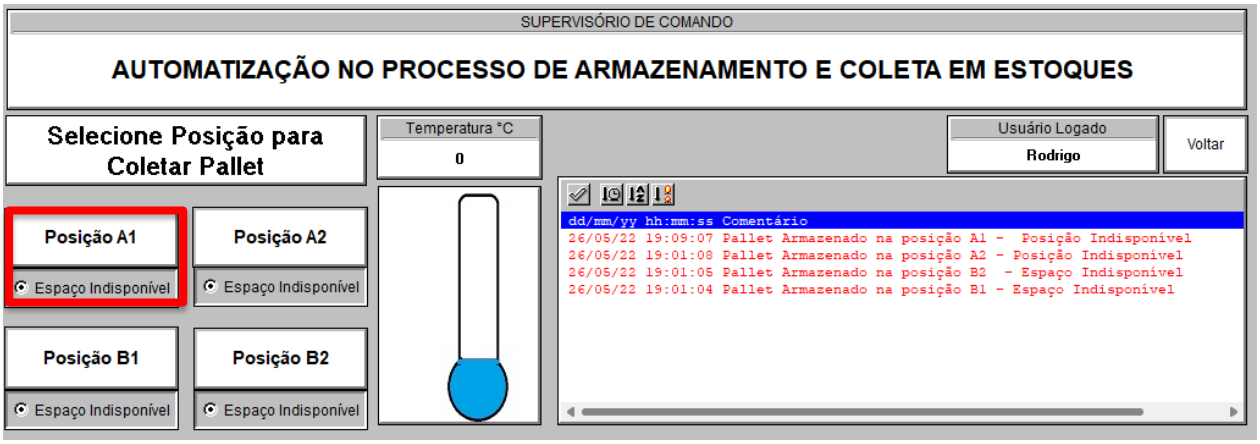

Figura 30 – (k) - Tela de controle para nova escolha (Fonte: Elaborado pelos autores)

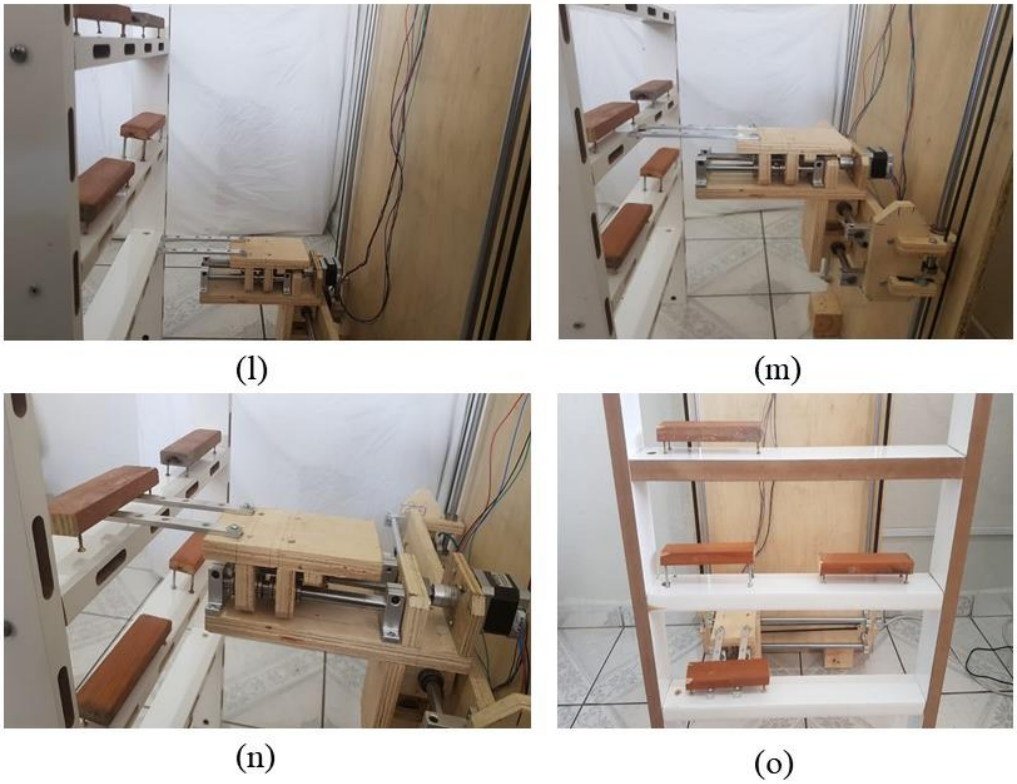

Figura 31 – Funcionamento – movimentações de coleta do manipulador (Fonte: Elaborado pelos autores)

5. Considerações Finais

A partir dos resultados obtidos nesta pesquisa, conclui-se que é exequível a integração de um manipulador cartesiano na gestão de estoque a partir de um *software* de supervisão e aquisição de dados e um Arduino.

A interface de comunicação entre o operador e o manipulador foi elaborada através do sistema de controle composto pelo *software* Elipse SCADA que é o principal meio de interação homem-máquina, o protocolo Modbus que realiza a comunicação entre o Elipse SCADA e o Arduino e o próprio Arduino que trabalha como controlador junto com a CNC *Shield* e os *drivers*, responsáveis por movimentar os motores de passo.

A gestão se obteve por meio do histórico de movimentações realizadas e quais operadores as fizeram, estas informações ficam disponíveis na tela do *software* Elipse SCADA.

A programação foi produzida em linguagem C e implementada no sistema, controlando os eixos movimentados pelos motores de passo, de acordo com os comandos designados pelo operador no *software* Elipse SCADA.

O protótipo de um manipulador cartesiano com 3 graus de liberdade, apresentando os eixos XYZ foi desenvolvido é inteiramente capaz de realizar a movimentação de *pallets*, sendo aplicado para a coleta e armazenamento.

Para aplicação real do projeto, algumas melhorias podem ser realizadas. Como o presente trabalho se trata de um protótipo, o *software* Elipse SCADA foi utilizado na versão DEMO, por isso, algumas funções ficaram limitadas para o desenvolvimento. No entanto, é possível melhorar a interface e os recursos de gestão, disponibilizando mais posições, usuários e histórico mais detalhado. Entre outros recursos, como o botão de emergência, sensor fim de curso, sensores de presença e sensor *laser* para posicionamento.

6. Referências

ASSOCIAÇÃO BRASILEIRA DE AUTOMAÇÃO – **GS1 BRASIL. Índice de Automação – Mercado Brasileiro**. 2020. Disponível em: <https://www.gs1br.org/indices-e-pesquisas/indices/Documents/Índice de Automação do Mercado Brasileiro> Acesso em: 23 set. 2021.

ASSOCIAÇÃO PAULISTA DE SUPERMERCADOS**. Furto e má gestão fazem varejo perder US$ 2,3 bi no Brasil**. 2018 Disponível em: <https://portalapas.org.br/furto-e-ma-gestao-fazem-varejo-perder-us-23-bi-no-brasil/>. Acesso em: 23 set. 2021.

BALLOU, R. H. Gerenciamento da Cadeia de Suprimentos: Logística Empresarial. 5. ed. Porto Alegre: Bookman, 2006.

BRANQUINHO, Marcelo. A. **Segurança de Automação Industrial e SCADA .** 1. ed. Rio de Janeiro: Elsevier, 2014. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788595153233/. Acesso em: 05 out. 2021.

CHAVIER, L. F. **Programação para Arduino - Primeiros Passos**. 2019.

CHRISTOPHER, M. Logística e gerenciamento da cadeia de suprimentos: estratégias para redução de custos e melhorias de serviços. São Paulo: Pioneira, 1997.

CRAIG, Jhon J.. **Introduction of robotics: Mechanics and control.** 2 ed. California: Addidon-Wesley Publishing Company, 1989.

DAMAS, Luís. **Linguagem C**, 10. ed. Rio de Janeiro: Grupo GEN, 2006. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788521632474/. Acesso em: 24 out. 2021

ELETROGATE. **Tudo sobre DRIVER A4988 e Motor de Passo**. Disponível em: <https://blog.eletrogate.com/driver-a4988-motor-de-passo-usando-o-arduino/>. Acesso em: 4 fev. 2022.

ELETROGATE. **CNC Shield: Guia prático e introdução ao GRBL**. Disponível em: <https://blog.eletrogate.com/cnc-shield-guia-pratico-e-introducao-ao-grbl/>. Acesso em: 4 fev. 2022.

FDC; ABOL. **Perfil dos Operadores Logísticos**. Disponível em: <https://abolbrasil.org.br/projects/perfil-dos-operadores-logisticos/>. Acesso em: 5 set. 2021.

FERNANDES, M. O. **Cartesiano De 3 Graus De Liberdade**. 2016.

LAMB, F. **Automação Industrial na Prática**. [s.l.] Bookman, 2015.

MANZANO, José.Augusto.**NG Linguagem C - Acompanhada de uma Xícara de Café** 1. ed. São Paulo: Editora Saraiva, 2015. 9788536519364. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788536519364/. Acesso em: 24 out. 2021.

MARQUES, RONEI. **Problemas na gestão de estoque podem causar falência de empresas**. [Entrevista concedida a] Terra, fevereiro de 2018. Disponível em: < https://www.terra.com.br/noticias/problemas-na-gestao-de-estoque-podem-causar-falencia-de-empresas,8c6c62f4e072c33eddee38d829cefa06scptms92.html>. Acesso em: 05 de setembro de 2021

MATARIC, Maja. J. **Introdução á robótica** . São Paulo: Editora Blucher, 2014. 9788521208549. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788521208549/. Acesso em: 21 out. 2021

MCROBERTS, M. Beginning **Arduino .** Apress, 2010.

MENDONÇA, C. **Problemas com estoque da Cnova geraram prejuízos de R$ 512 milhões ao GPA**. Disponível em: <https://www.consumidormoderno.com.br/2016/07/27/problemas-com-estoque-da-cnova-geraram-prejuizos-de-r-512-milhoes-ao-gpa/>. Acesso em: 5 set. 2021.

MONK, Simon. **Programação com Arduino,** 2. ed. Porto Alegre : Bookman, 2017. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604472/. Acesso em: 22 out. 2021.

MORAES, Cícero, C. e CASTRUCCI, Plínio de Lauro. **Engenharia de Automação Industrial**, 2 ed, Grupo GEN, 2006.

NARDI, M. **Construa sua própria CNC 3.0 com ArduinoConstrua sua própria CNC 3.0 com Arduino.** Disponível em: <https://marlonnardi.com/2020/06/16/construa-sua-propria-cnc-3-0-com-arduino/>. Acesso em: 2 ago. 2021.

NOVAES, A. G. N. **Logística e Gerenciamento da cadeia de distribuição**. 2.ed. São Paulo: Atlas, 2004.

PAZOS, F. **Automação de Sistemas e Robótica**. Rio de Janeiro: Axcel Books do Brasil, 2002.

PETRUZELLA, F. D. **Motores elétricos e acionamentos**. Porto Alegre: 2013.

POZO, H. Administração de recursos materiais e patrimoniais: uma abordagem logística. 4. ed. São Paulo: Atlas, 2007.

RIVIN, E. **Mechanical Design of Robots**, 1 ed., MacGraw-Hill Inc., New York, 1988

ROMANO, V. F. **Robótica Industrial: Aplicação na indústria de manufatura e de processos**. São Paulo: Editora Edgard Blücher LTDA, 2002. Disponível em: https://docplayer.com.br/18225324-Prologo-vitor-ferreira-romano.html Acesso em: 27 set. 2021

SANTOS, Winderson.Eugenio. D.; JR., José.Hamilton.Chaves. G. **Robótica Industrial - Fundamentos, Tecnologias, Programação e Simulação.** 1. ed. São Paulo: Saraiva, 2015. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788536520254/. Acesso em: 27 set. 2021

SCHULTZ, F. **Conheça as 7 vantagens da automatização de processos**. Disponível em: <https://blog.bomcontrole.com.br/vantagens-automatizacao-de-processos/>. Acesso em: 5 set. 2021.

VIEIRA, Á. V. D. O. **Desenvolvimento de um manipulador robótico de baixo custo para uso didático.** P. 1–4, 2011.